

# PTZ 제어에 의한 이중차영상 기반의 움직임 추적 시스템의 설계 및 구현

양금석<sup>†</sup>·양승민<sup>††</sup>

## 요 약

이동물체 추적은 카메라와 물체사이의 이동관계에 따라 카메라는 고정되어 있고 물체가 이동하는 경우, 물체는 고정되어 있고 카메라가 움직이는 경우와 물체가 이동하고 카메라도 움직이는 경우로 분류된다. 이동 물체를 추적하는 기존의 방법으로는 차영상 기반의 방법과 광류(Optical Flow)를 이용한 방법이 있다.

광류를 이용한 방법은 카메라와 물체 중에서 하나가 고정되어 있는 경우에 주로 쓰인다. 이 방법은 이전 프레임에서 구한 물체가 다음 프레임의 한 위치에 있을 때 시공간 벡터를 이용하여 물체의 이동을 인식하고 추적한다. 그러나 이 방법은 계산량이 많아 처리속도가 느리기 때문에, DVR(Digital Video Recorder)과 같이 실시간 처리가 필요한 보안 시스템에서는 사용할 수 없다.

차영상을 이용한 방법은 카메라는 고정되어 있고 물체가 이동하는 경우에 적용 가능하며 배경영상과 차영상을 이용하여 물체를 추적한다. 이 방법은 계산량이 적어 처리속도가 빠르기 때문에, 실시간 처리가 필요한 보안 시스템에 적합하다. 하지만, 카메라가 움직일 경우에는 배경영상을 구하기가 어려우므로 물체 검출을 할 수 없는 단점이 있다.

본 논문에서 설계, 구현한 PTZ(Pan-Tilt-Zoom) 제어에 의한 차영상 기반의 이동물체 추적 시스템은 카메라와 물체가 모두 이동하는 상황에서 사용할 수 있다. 이 이동물체 추적 시스템은 차영상 기반방식의 장점을 이용, 실시간 처리가 가능하기 때문에 DVR과 같은 실시간 처리가 필요한 보안 시스템 구축을 용이하게 해 줄 것이다.

키워드 : 움직임 추적, 차영상, PTZ, 실시간처리

## Design and implementation of motion tracking based no double difference with PTZ control

Yang Geum-Seok<sup>†</sup> · Yang Seung Min<sup>††</sup>

### ABSTRACT

Three different cases should be considered for motion tracking: moving object with fixed camera, fixed object with moving camera, and moving object with moving camera. Two methods are widely used for motion tracking: the optical flow method and the difference frame method. The optical flow method is mainly used when either one, object or camera, is fixed. This method tracks object using time-space vector which compares object position frame by frame. This method requires heavy computation, and is not suitable for real-time monitoring system such as DVR(Digital Video Recorder). The different frame method is used for moving object with fixed camera. This method tracks object by comparing the difference between background images. This method is good for real-time applications because computation is small. However, it is not applicable if the camera is moving.

This thesis proposes and implements the motion tracking system using the difference frame method with PTZ(Pan-Tilt-Zoom) control. This system can be used for moving object with moving camera. Since the difference frame method is used, the system is suitable for real-time applications such as DVR.

Key Words : Motion Tracking, Difference Frame, PTZ, Realtime Tracking

## 1. 서 론

이동물체 추적은 컴퓨터 비전 및 여러 응용분야에서 많은

연구가 진행되고 있는 분야이다. 특히 움직이는 카메라에서 이동물체를 추적하는 것은 배경의 움직임이 있기 때문에 컴퓨터 비전 분야에서 대단히 도전적인 과제중 하나이다. 이동물체 검출 및 추적 시스템은 물체 인식, 로봇, 감시시스템, 무인 차량 시스템과 같은 다양한 응용 분야에 적용된다.

이동물체 추적은 카메라와 물체사이의 이동 관계에 따라 카메라는 고정되어 있고 물체가 이동하는 경우, 물체는 고

※ 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌습니다.

† 정 회 원 : (주)엠스톤 부설연구소 연구원

†† 정 회 원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2004년 12월 31일, 심사완료 : 2005년 4월 4일

정되어 있고 카메라가 움직이는 경우와 물체가 이동하고 카메라도 움직이는 경우로 분류된다. 이동물체를 추적하는 기존의 방법으로 차영상을 이용한 방법과 광류(optical flow)를 이용한 방법이 있다.

광류를 이용한 방법은 카메라와 물체 중에서 하나가 고정되어 있는 경우에 주로 쓰인다. 이 방법은 이전 프레임에서 구한 물체가 다음 프레임의 한 위치에 있을 때 시공간벡터를 이용하여 물체의 이동을 인식하고 추적한다. 그러나 이 방법은 계산량이 많아 처리속도가 느리기 때문에, DVR(Digital Video Recorder)과 같이 실시간 처리가 필요한 보안 시스템에서는 사용할 수 없다.

차영상을 이용한 추적 방법은 카메라는 고정되어 있고 물체가 이동하는 경우에 적용가능하며 배경영상과 차영상을 이용하여 물체를 추적한다. 이 방법은 계산량이 적어 처리속도가 빠르기 때문에, 실시간 처리가 필요한 보안 시스템에 적합하다. 하지만 주위 환경의 변화에 많은 제약을 받으며, 카메라 움직임 경우 배경영상을 구하기가 쉽지 않기 때문에 전처리 과정에서 잘못된 결과를 얻게 되면 이동물체 추적시 추적하고자 하는 물체를 정확하게 추적하지 못하는 경우가 발생한다.

물체가 이동하고 카메라도 움직이는 경우, 기존의 방법을 이용하여 물체를 추적하게 되면 배경에서 추적 물체를 분할 및 정합하고 추적하기가 쉽지 않다. 또한, DVR과 같은 보안 시스템에 적용하려면 색상을 가지고 나타나는 물체를 실시간으로 추적하도록 본질적인 문제인 계산시간을 줄여야 한다.

본 논문에서는 움직이는 카메라로부터 획득한 연속영상에서 이동물체를 자동으로 검출하고 PTZ 제어를 통해 추적하는 시스템을 제안하였다. 제안된 시스템은 크게 영상캡처, 이동물체 분할과 정합 및 추적부분으로 구성되어 있다. 영상포맷은 디지털 영상처리와 관련된 계산량을 최소화하기 위해 RGB에 비해 화소 당 비트수가 적은 YUV 계열의 YV12 포맷을 사용한다. 영상캡처시 RGB 영상포맷을 YV12로 변환하며, 이후 처리되는 모든 영상처리를 YV12를 기준으로 설계 및 구현하였다.

이동물체 분할은 차영상을 이중화하는 방법을 사용한다. 이동물체 분할시 크기가 작은 잡음은 로컬 히스토그램 분석과 모폴로지 필터링으로 크기가 큰 잡음은 라벨링을 통해 제거한다. 이동물체의 움직임이 감지되었을 경우에만 이동물체 정합을 수행하도록 이중차영상을 8x8 블록 영상으로 등분하고 블록 내에서 로컬 움직임 감지 및 블록 간의 전역 움직임 감지를 검사하도록 하였다.

이동물체 정합은 이동물체 분할에서 이동물체의 움직임이 감지되었을 경우에만 수행하며 추적중인 이동물체가 있을 경우 추적중인 이동물체의 히스토그램과 라벨링된 이동물체의 히스토그램간의 유사도 분석을 통해 정합하도록 하였다. 추적중인 이동물체가 없을 경우에는 라벨링된 물체들 중에서 가장 큰 것을 추적중인 이동물체로 자동 선택되도록 하였다.

이동물체 추적은 영상 프레임 내에서 추적중인 이동물체의 이동거리가 한계이동거리를 넘었을 경우에만 PTZ 제어를 통해 추적하도록 하였다. PTZ 제어를 통해 추적하는 동안에는 플래그를 두어 이동물체 분할 및 정합을 수행하지 않고 영상캡처만 하도록 하였다.

PTZ 장비가 이동물체를 추적하는 동안 영상캡처를 하며 대기해야 하는 시간 산출시 PTZ 각속도, 장비 구동 및 신호 처리 지연시간의 PTZ 장비의 성능과 가변 영상캡처 주기와 크기를 고려하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 이동물체 추적을 위한 영상처리 연구와 이를 이용한 기존의 이동물체 추적 관련 연구들을 살펴본다. 3장에서는 본 논문에서 제안한 이동물체 추적시스템의 각 단계를 설계하고 구현한다. 4장에서 결론 및 앞으로의 연구 방향을 제시하며 논문을 맺는다.

## 2. 관련 연구

### 2.1 이동물체 추적 시스템의 실시간 지원

이동물체를 탐지하고 추적하는 기법들은 동영상 분석의 핵심인 처리시간을 개선하고자 하는 노력으로 대변될 수 있다. 다양한 모양과 크기, 색상을 가지고 나타나는 물체를 실시간으로 추적하도록 하기 위해서는 계산시간을 줄여야 한다[2]. 이와 관련된 연구들로는 제한된 영역의 창을 사용한 방법[12], 시공간을 적응탐색에 의한 방법[13][14], 이동물체의 탐지 및 추적을 수행하는 모듈들을 하드웨어 화하는 방법[8] 등이 있다.

### 2.2 이동물체 추적 연구

배경의 움직임이 발생하는 이동 카메라로부터 획득한 연속영상에서의 물체 추적에 대한 연구는 다음과 같이 몇 개의 유형으로 분류할 수 있다. 첫 번째 유형은 3차원 모델 기반의 방법이다. 이 방식은 3차원 공간상의 좌표를 정사투영(orthographic) 혹은 원근 투영(perspective projection)에 의하여 2차원 영상 평면상의 좌표로 표현하는데 사용된다. 이 모델은 3차원계의 분석적인 정보를 제공하기는 하지만, 3차원 강체 움직임(rigid motion)의 해석에만 적용할 수 있다는 단점이 있다.[3][4]. 개별 영상 단위로는 비교적 정확한 움직임의 해석이 가능하지만 이를 영상 취득 환경이 시시각각 변하는 영상 시퀀스에 적용했을 경우 영상간의 관계를 규명하는데 많은 계산을 필요로 한다. 또한 비강체 움직임(non-rigid motion)의 해석에도 한계가 있다.

두 번째 유형은 광류 기반 추정방법[5][6]이다. 이 방법은 이전 프레임에서 구해진 물체가 시간과 공간의 변화를 거쳐서 다음 프레임의 한 위치에 있을 때 시공간 벡터를 이용하여 물체의 이동을 인식하고 추적한다. Horn 등[11]에 의해 제안된 방법은 연속적인 영상에서 완만한 밝기를 갖는 경우에 정확한 광류를 얻을 수 있으나 밝기가 급변하는 영상의

경우에는 정확한 해를 구하기가 어렵다. 그리고 각 화소에 대해 반복적인 계산이 요구되어서 많은 계산적 부담을 필요로 한다. 광류를 이용한 방법은 여러 요소에 의해 영상의 밝기 값이 변화하는 외부에서의 적용의 어려움과 계산 량이 많기 때문에 처리속도가 느린 단점이 있다.[12]

세 번째 유형은 특징 기반 추정방법[7][8]이다. 이 방법은 움직이는 카메라에서 획득한 연속영상에서 영상의 특징을 이용하여 이동물체와 배경의 서로 다른 모션 정보를 구하고 배경으로부터 물체를 분리 하는 방법이다. 영상의 특징이란 면과 면 사이의 경계선, 선의 교차점, 물체의 모서리 부분, 곡선의 극대/극소점 등을 말한다. 인간의 시각 체계(human visual system)에서 배경 속 물체의 존재는 먼저 이러한 영상의 특징에 의해 배경과 물체를 구분함으로써 판단되며, 구분된 물체의 형태는 물체의 외곽 경계 속에 존재하는 특징 점들의 연결 관계에 의하여 결정된다. 물체의 이동 여부의 판단 및 추적은 이러한 특징 점들의 이동 관계를 규명함으로써 가능하다. 이동물체의 검출은 특징 점의 시간에 따른 이동을 나타내는 모션 경로를 구한 후, 특징 점의 위치, 특징 기반 모션 경로를 이루는 특징 벡터의 각과 크기를 매개 변수로 하는 특징 공간상의 군집화를 함으로써 가능하다. 특징 경로 기반 정보로 사용되는 모서리 점의 검출은 SUSAN (smallest univalue segment assimilating nucleus)[13], KLT (Kanade-Lucas-Tomasi)[14], Harris[15] 등의 방법이 있다. 이러한 특징 기반 방식은 특징 점들이 잘 나타나지 않는 영상에 적용하기가 힘들고, 배경의 움직임이 일정한 하나의 방향을 갖지 않는 이동물체를 검출하기 위한 영상구조에서는 균일한 배경 정보를 얻기가 힘든 단점이 있다[14].

네 번째 유형은 블록 기반 추정방법[9][10]이다. 이 방법은 비디오 압축(video compression) 분야에 주로 사용되는 방법으로, 영상을 정방형 블록 단위로 나누어, 연속적인 프레임  $I(x, y, t)$ 와  $I(x, y, t + \Delta t)$  사이의 모션을 추정한다. 개별 모션 벡터의 추정은 대개 프레임  $I(x, y, t)$ 의 개별 블록과 프레임  $I(x, y, t + \Delta t)$ 의 미리 정의된 검색 윈도우 내부에서 전 영역을 비교하는 과정으로 수행된다. 따라서 수행 시간이 많이 걸리게 되는 단점이 있다.

마지막 유형은 차영상 기반 방법이다. 이 방법은 현재 프레임에서 이전 프레임을 빼는 것으로 움직임이 없으면 검정색, 움직임이 있으면 어떤 대상이 표현된다. 인접한 두 프레임 간에 차이가 발생하는 영역이 움직이는 영역이다. 차영상 기반 방법은 움직이는 물체를 실시간으로 추적 할 수 있지만 물체가 정지하는 경우에는 물체가 존재하지 않는 형태로 나타나서, 정확한 물체의 추적이 어렵다.[3]

### 2.3 영상처리

#### 2.3.1 영상포맷

YUV는 ITU-R BT601에서 제정한 세계적인 디지털 비디오 표현 방식으로서 JPEG 및 MPEG 표준에서 사용되고 있다. YUV는 색상을 하나의 휘도 신호(Y)와 두 개의 색상 신

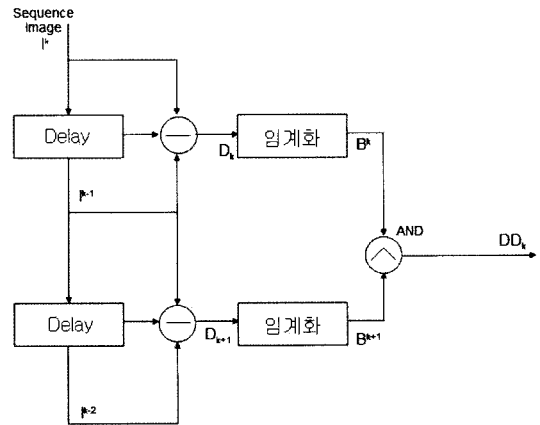
호(U, V)로 표현한다. Y는 휘도(luminance)로서 전 색상에 걸쳐 표현되는 단색 영상의 밝기로서 흑백 수상기에 나타나는 영상이고, U, V 신호는 각각 파랑색과 붉은색을 표현한다. Y와 U, V의 표본 비율과 구성방식에 따라 여러 개의 형식이 있다. [1]

YV12는 YUV 4:2:0 플레나(Planar) 형식의 관용적인 줄임말이다. 4:2:0 이라는 의미는 Y와 U, V의 표본 비율로서 U와 V의 표본 비율이 Y의 1/2라는 의미이다. YUV 표본의 순서를 구성하는 방식에 따라 플레나(Planar) 방식과 인터레이스(Packed) 방식이 있다. 플레나(Planar) 형식은 Y, U, V 색상을 각각의 하나의 프레임으로 구성된다. [1]

### 2.4 이동물체 분할

#### 2.4.1 이중차영상

자연환경에서 획득된 영상은 하늘, 아스팔트 도로와 같이 균일한 질감을 가지는 배경들이 많은 부분을 차지한다. 차영상 기법으로 2개의 연속된 영상에서 이러한 균일한 질감을 가지는 성분들은 제거하고 움직임이 있는 영역만을 검출 가능하다. 이중차영상 방법은 연속 영상으로부터 균일한 질감을 가지는 배경 영역을 효율적으로 제거하고 영상 내의 움직임 정보를 얻기 위해 사용하며 움직임 정보를 해당 영역에 누적시킨다. 이중차영상 방법을 이용하여 연속 영상에서 움직임 정보를 획득하는 과정은 (그림 1)과 같다.



(그림 1) 이중차영상 기법의 구성

차영상, 이중차영상을 만들기 위한 수식은 각각 수식 (2-1), (2-2)와 같다.

$$D_k(x, y) = |I_k(x, y) - I_{k-1}(x, y)| \quad (2-1)$$

$$DD_k(x, y) = \begin{cases} 1 & \text{if } (D_{k+1}(x, y) > T) \wedge (D_k(x, y) > T) \\ 0 & \text{otherwise} \end{cases} \quad (2-2)$$

#### 2.4.2 이진화

일반적으로 영상을 이진 영상으로 변환하는 처리를 이진

화라고 한다. 이진화는 기초적이지만 영상 인식을 위한 매우 중요한 처리 중의 하나이며, 이진에 의한 표현으로 데이터양을 적게 할 수 있다. 이진화의 장점으로는 대상 영상의 성질, 특징을 보존할 수 있으며, 이진 영상을 처리 대상으로 하게 되면 처리에 필요한 시간과 기억 용량이 대폭적으로 절감된다. 이진화의 방법으로는 여러 가지 방법이 있으나 가장 간단한 방법은 어떤 임계값 보다 작은 화소 값을 갖는 화소를 0(흑), 큰 화소 값을 갖는 화소를 255(백)를 할당하는 처리이다.[7]

2.4.3 모폴로지 영상처리

모폴로지 기법의 기본 아이디어는 미리 기하학적 형태를 알고 있는 대상 물체의 정보를 반영하여 영상 내에서 원하는 부분만을 추출하는 것이다. 일반적인 영상의 경우 영상 내에는 다양한 물체들이 혼합되어 있으며 우리가 관심을 가지는 나머지 물체들은 노이즈 성분이라고 볼 수 있다. 이러한 경우, 원하는 물체만을 추출하기 위해서 모폴로지 기법을 이용한다. 모폴로지 기법을 구성하는 가장 기본적인 연산에는 다음의 두 가지가 있다.

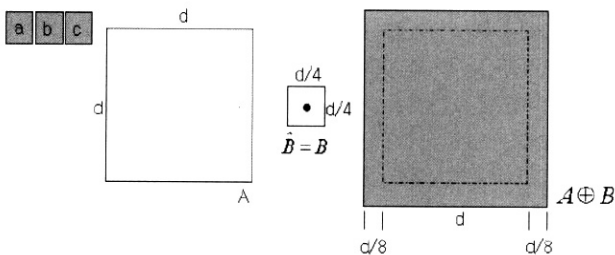
- 침식(erosion) 연산
- 팽창(dilation) 연산

위의 2가지 연산을 복합적으로 사용한 연산에는 다음의 두 가지가 있다.

- 제거(opening) 연산 : 팽창(dilation) 연산 수행 후 침식(erosion) 연산 수행
- 채움(closing) 연산 : 침식(erosion) 연산 수행 후 팽창(dilation) 연산 수행

팽창 연산은 물체의 최외각 화소를 확장시키는 역할을 한다. B에 의한 A의 팽창을 수식 (2-3)과 같이 표기한다.

$$A \oplus B = \{z | (\widehat{B}_z \cap A \neq \emptyset)\} \tag{2-3}$$

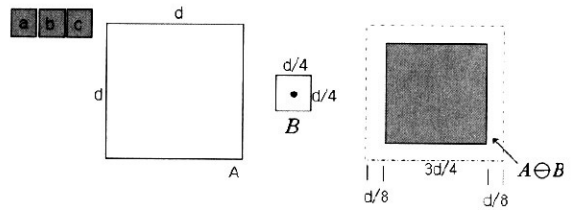


(그림 2) 팽창 연산

(그림 2)는 (a)의 원 영상 둘레에 점을 중심으로 한 정사각형 성분 (b)가 돌면서 지나가는 지역에 팽창한 영역을 (그림 2)의 영상 (c)에 보여주고 있다.

침식 연산은 물체의 최외각 화소를 축소시키는 역할을 한다. B에 의한 A의 팽창을 수식 (2-4)와 같이 표기한다.

$$A \ominus B = \{z | (B)_z \subseteq A\} \tag{2-4}$$



(그림 3) 침식 연산

(그림 3)은 영상 A의 둘레(a)에 점을 중심으로 한 정사각형 (b)을 돌면서 지나가는 지역을 제거하여 나타낸 영상이 (c)이다. B에 의한 A의 제거 연산을 침식 연산 후에 팽창연산을 하는 것으로 수식 (2-5)와 같이 정의한다.

$$A \cdot B = (A \ominus B) \oplus B \tag{2-5}$$

이와 같이 제거 연산을 하면 일반적으로 물체의 윤곽을 부드럽게 만들고, 가는 돌출부나 좁은 협곡 같은 부분을 없애준다. B에 의한 A의 채움 연산을 팽창 연산 후에 침식 연산을 하는 것으로 수식 (2-6)과 같이 정의한다.

$$A \cdot B = (A \oplus B) \ominus B \tag{2-6}$$

채움 연산 역시 윤곽을 부드럽게 하지만, 가는 협곡 같은 부분을 융합시키고, 작은 구멍을 제거하며 윤곽 내에서 간격을 채워준다[17].

2.4.4 라벨링

라벨링은 연결되어 있는 모든 화소(연결 성분)에 같은 라벨(번호)을 붙이고, 다른 연결 성분에는 다른 번호를 붙이는 처리로써, 이진 영상처리의 한 방법이다. 물체의 영역 구분은 라벨링을 통해 연결 성분으로 분리할 수 있으며, 각 연결 성분의 특징을 조사할 수 있다. 영역 분할은 이진 영상에서 구분된 영역이 어떤 형태인지 구별하기 위해 영역 성장 범으로 라벨링을 한다.

영역 성장범이란 주목하고 있는 화소와 그것에 인접한 화소들이 서로 같은 특징을 갖고 있는 경우에, 그들을 하나의 영역으로 통합하는 처리를 순차적으로 실행하고, 특징이 같은 영역을 조금씩 성장시켜 최종적으로 영상 전체에 대해 영역분할을 하는 방법이다[9]. 라벨링을 수행하면, 물체와 잡음들에 각각의 라벨이 설정된다.

2.5 이동물체 정합

분할된 이동물체의 정합과 제안된 시스템에서 사용하는 YV12 색상 모델을 위해 밝기 값을 특징으로 하는 패턴 인식의 방법인 히스토그램 인터섹션 방법을 사용한다[12].

2.5.1 히스토그램 인터섹션

히스토그램 인터섹션은 히스토그램 정합에 관한 연구법으로써 고전적인 패턴인식의 일종이라고 할 수 있다. 이 방법은 히스토그램의 각 밝기 값을 특징 점으로 사용한다.

인식하고자 하는 물체가 동일한 물체인지를 판단하기 위해서 히스토그램 인터섹션을 이용하는데, 결과 값이 클수록 인식하고자 하는 물체와 유사하다고 할 수 있다. 히스토그램 인터섹션은 수식 (2-7)과 같다.

$$\sum_{j=1}^n \text{Min}(H_{I_i}(j), H_M(j)) \quad (2-7)$$

여기서, 입력 영상  $I$ 의 한 점  $k$ 을 중심점으로 가지는 모델 영상과 같은 크기의 정합후보 영상을 만든다. 정합후보 영상의 히스토그램은  $H_I$ 이고, 모델 영상의 히스토그램은  $H_M$ 이다. 정합후보 영상과 모델 영상은 동일한  $n$ 개의 밝기 값을 갖고 있다. 모델 영상의 히스토그램과 정합후보 영상의 히스토그램 인터섹션의 결과는 정합후보 영상에서 동일한 밝기 값을 갖는 화소에 대응되는 모델 영상의 화소의 개수다.

0과 1사이의 정합 값을 갖도록 하기 위해 히스토그램 인터섹션 값을 정규화 시킨다. 모델 영상의 히스토그램에 대응되는 화소의 개수를 적용시키는데 정규화 된 값은 다음과 같이 나타낼 수 있다.

$$H(I_i, M) = \frac{\sum_{j=1}^n \text{Min}(H_{I_i}(j), H_M(j))}{\sum_{j=1}^n H_M(j)} \quad (2-8)$$

히스토그램 인터섹션 결과 값은 배경과의 분할이 쉽게 되지 않는 상황에서도 그 값이 감소되지 않는다. 배경에서 물체를 완벽하게 분할해 내기란 쉽지 않은데 만약 모델 내부에 있는 화소가 동일한 밝기 값을 갖는다면 정합 값이 커지게 된다. 그리고 정합을 하려는 물체 내부의 특정 밝기 값을 갖는 화소의 개수가 모델 영상에서 동일한 값을 갖는 화소 개수보다 적으면 정합 값이 높아진다.

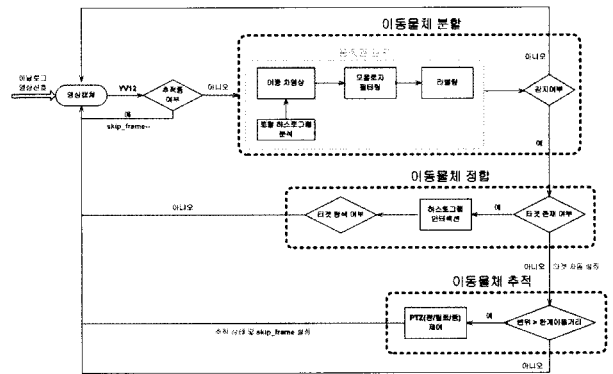
3. 이동물체 추적 시스템의 설계 및 구현

본 장에서는 카메라와 물체가 모두 이동하는 상황에서 PTZ (팬/틸트/줌) 제어에 의한 이종차영상 기반의 이동물체 추적 시스템을 설계하고 구현한다. 먼저 전체 구조를 설명하고, 이어서 각 구성 요소에 대한 개념과 세부적인 구현 내용을 자세히 설명한다.

3.1 전체구조

본 연구에서 설계, 구현한 이동물체 추적 시스템은 (그림 4)와 같이 크게 영상캡처부분과 이동물체 분할 및 정합 부

분, 이동물체 추적부분으로 구성된다. 영상캡처부분은 카메라에서 입력된 아날로그 영상신호를 YV12 신호로 변환한다. 현재 추적 상태가 활성화 되어 있으면 PTZ 제어를 통해 이동물체를 추적 중이므로 이동물체 분할 및 정합, 추적부분을 수행하지 않고 다음 영상을 캡처한다. 이동물체 분할 부분은 영상 평면 내의 이동물체를 블록 기반 이종차영상 기법을 사용하여 검출하고 정확한 이동물체의 영역을 배경으로부터 분리한다. 분리된 이동물체의 영역들을 모폴로지 필터링을 수행한 후 다중이동물체 추적을 위해 라벨링을 한다. 이동물체 정합 부분에서는 라벨링 된 영역 중에서 히스토그램 인터섹션을 사용하여 정합한다. 이동물체 추적부분은 분할단계에서 추출된 이동물체의 정보를 가지고서 이동물체가 화면상에서 사라지려 할 경우 PTZ 제어를 통해 이동물체를 추적하여 화면상에서 사라지지 않도록 한다. 서로 다른 PTZ 프로토콜을 가지는 장비를 지원하기 위해 이동물체 추적부분으로부터 전달받은 PTZ 프로토콜을 각 장치의 프로토콜에 맞도록 변환하여 해당 장치에 전달한다.



(그림 4) 이동물체 추적 시스템 구조

3.2 영상캡처

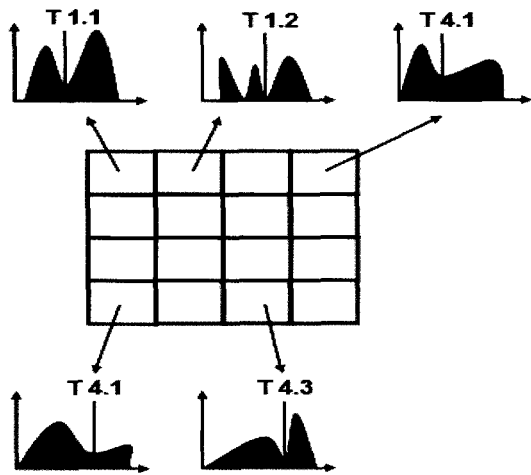
본 논문에서는 영상처리 시간 단축과 최적화를 위해 영상 포맷으로 YUV 계열의 YV12 형식을 사용한다. 영상캡처시 카메라로부터 입력받은 아날로그 영상신호를 YV12 형식으로 변환한다. 로컬 히스토그램 분석, 이종차영상, 모폴로지 필터링, 라벨링, 히스토그램 인터섹션, 디스플레이 부분들을 영상처리시 지연이 발생할 수 있다. 이러한 지연을 RGB 영상형식에 비해 화소 당 비트수가 적으면서도 화질 열화가 비교적 적은 YV12 형식을 사용하여 줄였고, YV12 형식을 기준으로 최적화가 가능하게끔 설계하였다.

3.3 이동물체 분할

3.3.1 로컬 히스토그램 분석

차영상 생성시 임계값의 선정은 (그림 5)와 같이 원영상을 일정 블록 영상(8x8)으로 등분하고, 등분된 각 국부 영역에 대해 로컬 히스토그램(local histogram) 분석을 하여 얻어낸다. 이렇게 얻어진 각각의 임계값으로 이진화를 한다.

이진화를 위한 임계값 선정은 전역 히스토그램(global histogram) 분석이 아닌 국부 히스토그램(local histogram) 분석을 해야 한다. 이것은 움직이는 물체가 배경의 밝기값 분포와 유사한 경우, 차영상 전 영역에 히스토그램 분석을 하여 임계값을 선정하게 되면, 배경 밝기와 유사한 부분은 이진화가 제대로 되지 않게 되기 때문이다. 따라서 원 영상을 일정 블록 영상으로 나눈 뒤 히스토그램 분석에서 얻어진 각각의 임계값으로 이진화를 한다.



(그림 5) 로컬 히스토그램 분석

3.3.2 모폴로지 필터링

이진화 된 영상은 물체와 잡음이 함께 존재하므로, 이진화 된 영상에 존재하는 잡음을 제거하기 위해 형태학적 필터를 수행한다. 적용한 형태학적 필터는 제거 연산이며, 제거 연산을 통해서 물체의 윤곽을 부드럽게 만들고, 가는 돌출부나 좁은 협곡 같은 부분을 없애며 크기가 작은 잡음을 효과적으로 제거한다.

3.3.3 라벨링

라벨링은 이중차영상과 형태학적 필터링으로 잡음이 제거 되지 않은, 크기가 큰 잡음을 효과적으로 제거하기 위해 사용한다. 본 논문에서는 영역 성장 법으로 라벨링을 한다.

영역 성장법이란 주목하고 있는 화소와 그것에 인접한 화소들이 서로 같은 특징을 갖고 있는 경우에, 그들을 하나의 영역으로 통합하는 처리를 순차적으로 실행하여, 특징이 같은 영역을 조금씩 성장시켜 최종적으로 영상 전체에 대해 영역분할을 하는 방법이다.

라벨링을 수행하면, 물체와 잡음들에 각각의 라벨이 설정된다. 라벨링 된 영상에서 수식 (3-1)에 의해 구해진 라벨의 크기가 수식 (3-2)에 의해 일정크기 이하의 라벨은 잡음으로 판단하고 제거한다.

$$Value_i = \frac{\sum_{i=0}^n Labelcnt_i}{Allcnt} (i: 1, 2, 3, \dots) \quad (3-1)$$

$$f(x) = \begin{cases} 1 & \text{for Value}_i \geq T_{2i} \\ 0 & \text{for Value}_i < T_{2i} \end{cases} \quad (3-2)$$

( $T_{2i}$ : 임계값,  $Value_i$ : 각 라벨의 크기,  $Labelcnt_i$ : 각 라벨의 개수,  $Allcnt$ : 원영상의 크기)

3.3.4 움직임 감지

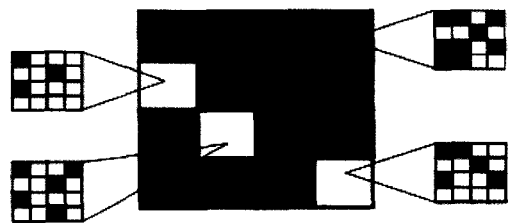
이동물체 분할의 마지막 단계는 움직임 감지 여부를 검사하는 것이다. 움직임 감지는 이중차영상을 가지고 수행하며, 이동물체 분할 동안 로컬 움직임 감지(local motion detection)와 전역 움직임 감지(global motion detection)의 두 단계로 나누어서 처리한다. 움직임 감지 여부에 따라 이동물체를 정합할 지 여부를 결정한다. 움직임이 감지되었을 경우에는 이동물체가 존재하는 것이므로 이동물체 정합 부분을 수행한다. 움직임 감지가 되지 않았을 경우에는 이동물체 정합단계로 진행하지 않고, 영상캡처를 수행한다. 이동물체 정합을 영상캡처시마다 하는 것이 아니라, 움직임 감지가 되었을 경우에만 이동물체 정합을 함으로써 영상처리 소요시간 및 계산 량을 줄였다.

로컬 움직임 감지는 (그림 6)과 같이 이중차영상을 일정 블록 영상(8x8)으로 등분하고, 등분된 각 국부 영역에서 수행한다. 이중차영상 전 영역을 순환하면서 수식 (3-3)에 의해 구해진 국부 영역에서 수식 (3-4)에 의해 차이가 있는 화소의 수가 일 정수 이하일 경우에는 잡음으로 판단하여 해당 국부 영역을 0으로 만든다.

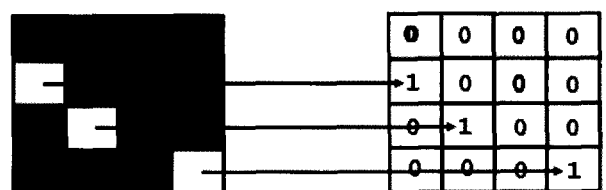
$$Block_k = \sum_y^{y_{max}} \sum_x^{x_{max}} DD_k(x, y) \quad (3-3)$$

$$f(x) = \begin{cases} 1 & \text{for Block}_i \geq T_{3i} \\ 0 & \text{for Block}_i < T_{3i} \end{cases} \quad (3-4)$$

( $Block_k$ : 국부 영역(8x8),  $y_{max}$ : 블록의 세로크기(8),  $x_{max}$ : 블록의 가로크기(8),  $DD_k(x, y)$ : 이중차영상)



(그림 6) 로컬 움직임 감지



(그림 7) 로컬 움직임 감지 수행결과 비트맵 정보

로컬 움직임 감지 결과는 블록 당 1비트로 기록하며, 생성된 블록 당 비트맵 정보는 (그림 7)과 같다.

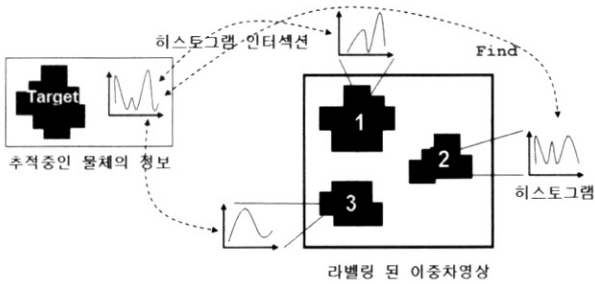
전역 움직임 감지는 로컬 움직임 감지 결과 생성된 비트맵 정보를 이용하여 수행한다. 전역 움직임 감지 값은 수식 (3-5)에 의해 구해지며, 수식 (3-6)에 의해 이 수치가 일정 크기 이상이면 이동물체가 이동한 것으로 결정한다.

$$GlobalMotion = \sum_{i=0}^j LocalMotionBitmap_i \quad (3-5)$$

$$MotionDetection = \begin{cases} 1 & \text{for } GlobalMotion \geq Sensitivity \\ 0 & \text{for } GlobalMotion < Sensitivity \end{cases} \quad (3-6)$$

### 3.4 이동물체 정합

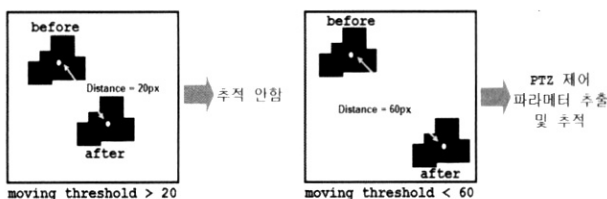
이동물체 정합 단계에서는 현재 추적중인 이동물체의 존재 여부에 따라 수행하는 방식이 달라진다. 추적중인 이동물체가 있을 경우, (그림 8)과 같이 추적중인 이동물체의 히스토그램과 이동물체 분할단계에서 생성된 라벨들의 히스토그램과 히스토그램 인터섹션을 수행하여 추적중인 이동물체가 현재 영상에서 존재하는지 여부를 검사한다. 추적중인 이동물체를 찾는데 실패하였을 경우에는 이동물체가 카메라의 시계에서 벗어난 것이므로 추적중인 이동물체의 정보를 초기화한다. 추적 중이던 이동물체가 없을 경우에는, 이동물체 분할단계에서 생성된 라벨들 중에서 영역의 크기가 가장 큰 이동물체를 추적할 이동물체로 설정한다.



(그림 8) 이동물체 정합 과정

### 3.5 이동물체 추적

이동물체 추적은 추적중인 이동물체의 이동거리가 한계이동거리를 넘었을 경우에만 PTZ 제어를 통해 추적하도록 한다. 이동물체의 이동거리가 한계이동거리를 넘지 않았을 경우에는 영상캡처단계로 돌아간다.



(그림 9) 이동물체 이동거리와 한계이동거리

한계이동거리란 PTZ 제어를 하기 위한 이동물체의 최소 이동거리로써 캡처 된 영상 프레임에서 추적중인 이동물체의 중심좌표 사이의 거리이다. 단위는 화소단위이며 응용에서 설정해야 하는 값이다.

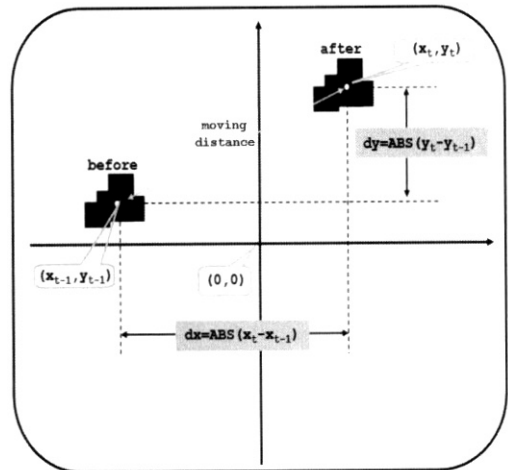
(그림 9)는 한계이동거리가 50으로 설정되었을 경우 이동물체 추적 여부를 보여준다. 이동물체의 이동거리가 20일 경우 한계이동거리 50보다 작기 때문에 추적하지 않는다. 그러나 이동물체의 이동거리가 60일 경우에는 한계이동거리 50보다 크기 때문에 PTZ 제어 정보를 추적하고 이를 이용하여 PTZ 제어를 통해 이동물체를 추적하게 된다.

#### 3.5.1 PTZ 제어정보 추출

본 단계는 이동 중인 물체에 대한 추적 작업을 수행하기 위해서 추적 물체의 중심위치사이의 거리와 속도를 이용하여 PTZ 제어정보를 추출하는 방법에 대하여 기술한다.

이동 중인 물체를 PTZ 제어를 통해 추적하기 위해 먼저 (그림 10)과 같이 영상 프레임 좌표계의 원점  $(X_0, Y_0)$ 와 추적중인 이동물체의 현재 영상 프레임에서의 중심좌표  $(X_t, Y_t)$ , 이전 영상 프레임에서의 중심좌표  $(X_{t-1}, Y_{t-1})$ , 중심좌표간의 거리  $dx, dy$ 을 정의한다.

PTZ에 대한 제어는 수직방향의 이동정도를 나타내는 수직각(tilt angle)과 수평방향의 이동정도를 나타내는 수평각(pan angle)에 의하여 정의할 수 있다. (그림 10)에서 추적중인 이동물체의 이전 프레임에서의 중심을 현재 영상 프레임에서의 중심으로 이동하기 위하여, 즉  $X$ 축으로  $dx$ 만큼 그리고  $Y$ 축으로  $dy$ 만큼 이동하기 위하여 필요한 수평각과 수직각은 수식 (3-7)과 수식 (3-8)에 의하여 구할 수 있다.

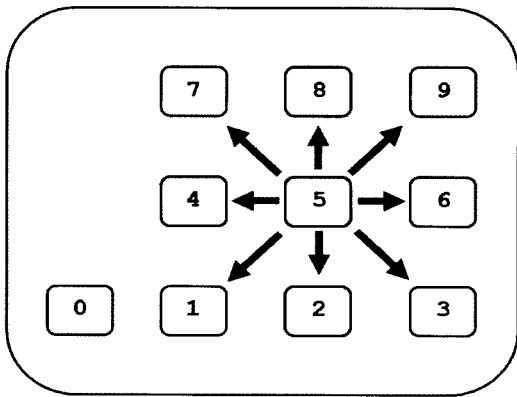


(그림 10) 이동거리 산출을 위한 영상 프레임 좌표계

$$\begin{aligned} dx &= abs(X_t - X_{t-1}) \\ \alpha &= \tan^{-1}(dx/f) \\ &\quad \text{if } (\alpha > 0) \\ &\quad PanRight(abs(\alpha)) \\ &\quad \text{if } (\alpha < 0) \\ &\quad PanLeft(abs(\alpha)) \end{aligned} \quad (3-7)$$

$$\begin{aligned}
 dy &= \text{abs}(y_t - y_{t-1}) \\
 \alpha &= \tan^{-1}(dy/f) \\
 &\quad \text{if } (\beta > 0) \\
 &\quad \text{TiltUp}(\text{abs}(\beta)) \\
 &\quad \text{if } (\beta < 0) \\
 &\quad \text{TiltDown}(\text{abs}(\beta))
 \end{aligned}
 \tag{3-8}$$

수식 (3-7)과 (3-8)의  $\alpha = \tan^{-1}(dy/f)$ 와  $\beta = \tan^{-1}(dx/f)$ 는 이전 영상 프레임에서 추적중인 이동물체를 현재 영상 프레임에서의 중심에 위치하도록 하기 위해 독립적으로 이동물체의 영상 프레임에서의 중심좌표와 영상 프레임사이에서의 중심좌표간 거리 및 초점거리( $f$ )를 사용하여 얻어진다. 수식 (3-7)과 (3-8)에 의하여 수직각과 수평각은 상호 독립적으로 산출이 가능하다. 그래서 (그림 11)과 같이 본 논문의 시스템은 8방향으로 제어가 가능하다. 즉, 수직 및 수평 방향뿐만 아니라 대각선방향으로도 제어할 수 있다.



(그림 11) PTZ 제어 방향

<표 1>  $\alpha$ 와  $\beta$ 에 의한 PTZ 제어 방향

$\alpha = \tan^{-1}(dx/f)$ $\beta = \tan^{-1}(dy/f)$	Control direction
$\alpha > 0, \beta = 0$	Pan_Right
$\alpha < 0, \beta = 0$	Pan_Left
$\alpha = 0, \beta > 0$	Tilt_Up
$\alpha = 0, \beta < 0$	Tilt_Down
$\alpha > 0, \beta > 0$	Pan_Right & Tilt_Up
$\alpha > 0, \beta < 0$	Pan_Right & Tilt_Down
$\alpha < 0, \beta > 0$	Pan_Left & Tilt_Up
$\alpha < 0, \beta < 0$	Pan_Left & Tilt_Down
$\alpha = 0, \beta = 0$	Stop
$\alpha = 0, \beta = 0$	Default

<표 1>은  $\alpha = \tan^{-1}(dx/f)$ 와  $\beta = \tan^{-1}(dy/f)$ 의 부호에 따른 각 제어방향을 나타낸다.

일반적으로 추적중인 이동물체는 추적하기 위한 PTZ 제어신호를 산출하는 중에도 이동을 계속한다. 그래서 추적중인 이동물체의 중심위치에 대한 정보뿐 아니라, 추적중인

이동물체의 이동 속도와 처리시간 그리고 PTZ 장비의 성능을 고려하여 제어신호를 산출해야 한다.

추적중인 이동물체가 현재 영상 프레임에서의 중심좌표에 위치하도록 하기 위해서 제어신호를 산출하기 위하여 소요되는 처리시간 동안에 예상되는 추적중인 이동물체의 이동거리와 PTZ 장비가 이동하는 동안에 예상되는 표적의 이동거리를 합한 값을 변위로 사용하여 계산해야 한다.

이동물체가 영상의 중심에 일치하도록 PTZ를 제어한 후 새로 산출된 표적의 중심과 영상의 중심이 일치하지 않는 정도만큼 증가 또는 감소시키는 크기가 현시점의 표적 속도가 된다. 이동물체를 추적하기 위한 PTZ 제어방향을 산출한 후 현재 프레임에서의 추적중인 이동물체의 중심에 일치하도록 PTZ 제어방향에 맞는 PTZ 제어신호를 PTZ 장비에 보낸다. 본 논문에서 사용하는 PTZ 제어신호는 <표 2>와 같다.

<표 2> PTZ 제어 방향과 PTZ 제어 신호

Control direction	Control Code
Pan_Right	6
Pan_Left	4
Tilt_Up	8
Tilt_Down	2
Pan_Right & Tilt_Up	9
Pan_Right & Tilt_Down	3
Pan_Left & Tilt_Up	7
Pan_Left & Tilt_Down	1
Stop	5
Default	0

### 3.5.2 PTZ 제어

PTZ 장비가 이동물체를 추적하는 동안 이동물체 분할은 수행하지 않는다. 그 이유는 본 논문에서 사용한 이동물체 분할 방식이 이중차영상 기반이기 때문이다. 이중차영상 기반의 단점은 배경과 이동물체가 같이 이동하는 상황에서는 이동물체를 배경으로부터 정확히 분리 할 수 없기 때문이다. 따라서 이동물체를 배경으로부터 정확히 분리해 내기 위해서 PTZ 장비가 이동물체를 추적하는 동안에는 이동물체 분할과 정합을 수행하지 않는다. 즉, PTZ 장비가 이동물체를 추적하는 동안에는 이동물체 분할과 정합을 하지 않고 추적이 완료될 때까지 일정 시간동안 영상캡처만 수행하게 되며, 일정 시간동안 영상캡처만 수행하는 시간이 PTZ 장비가 이동물체를 추적하는데 걸리는 시간이 된다.

일정시간 동안 영상캡처만 수행하면서 대기하기 위해 PTZ 장비가 이동물체의 추적을 완료하는데 소요되는 시간을 산출해야 한다. 이동물체의 추적에 소요되는 시간에 영향을 줄 수 있는 요소는 많이 있다. 우선 PTZ 장비마다 구동 및 신호 처리 지연시간, PTZ 각속도 성능이 모두 다르다. 두 번째로 카메라 렌즈의 초점거리에 따라 볼 수 있는 시야가 모두 다르다. 마지막으로 가변적으로 변할 수 있는 영상캡처 주기와 영상캡처 크기가 있다. 본 논문에서는 다양한 성능을 가지는 임의의 PTZ 장비와 가변적인 영상캡처



주기 및 크기를 가지더라도 지원할 수 있도록 설계하였다. 소요되는 시간은 수식 (3-9)와 같이 산출이 가능하다.

$$SkipFrames = PanningTime + TiltingTime \quad (3-9)$$

이동물체를 추적하는데 걸리는 시간은 PTZ 장비의 Pan Left/Right 시간과 Tilt Up/Down 시간의 합이다. 최소 SkipFrames 수만큼 영상캡처만 수행해야 한다. SkipFrames은 PTZ 장비가 이동물체를 추적하기 위해 대기해야 하는 최소한의 시간이다.

$$\begin{aligned}
 &fps = 1000 / FPS \\
 PanningTime &= (MovingWidthThreshold / MovingPanUnit) \times fps \\
 &\quad PanPreDelay + PanPostDelay \\
 MovingPanUnit &= fps \times PanSpeed \times MovingPanUnitPerPixel \\
 PanSpeed &= PanMaxSpd / 1000 \\
 MovingPanUnitPerPixel &= \partial(WidthImage / PanAngleFocus)
 \end{aligned} \quad (3-10)$$

수식 (3-10)은 추적중인 이동물체의 PTZ 추적을 위해 PTZ 장비의 panning 각속도, panning 구동 및 신호 처리 지연시간, 가변 영상캡처 주기 및 크기를 고려하여 panning 하는데 걸리는 시간을 계산하기 위한 수식이다.

FPS : 영상캡처 주기  
 fps : ms 단위의 영상캡처 시간  
 panning : PTZ 장비가 왼쪽/오른쪽으로 이동하는 것  
 PanPreDelay : PTZ 장비가 panning 명령을 처리하기 전에 걸리는 시간  
 PanPostDelay : PTZ 장비에 panning 명령을 처리한 후 걸리는 지연 시간  
 PanSpeed : 1° panning 하는데 걸리는 밀리세컨드 단위의 시간  
 PanMaxSpeed : 초당 panning 가능한 수평각  
 PanAngleFocus : 초점거리에 따른 가로 시계 각도  
 MovingPanUnit : 영상캡처 주기 동안 panning 하여 이동 가능한 화소 간격  
 MovingPanUnitPerPixel : panning 하여 영상 프레임에서 초당 이동 가능한 세로 길이  
 PanningTime = panning 각속도, panning 구동 및 신호 처리 지연시간, 가변 영상캡처 주기 및 크기를 고려하여 panning 하는데 걸리는 시간

(그림 12) Panning 소요 시간 산출 수식의 단위

(그림 12)는 수식 (3-10)에서 사용된 panning과 관련된 단위를 상세히 설명한 것이다.

$$\begin{aligned}
 &fps = 1000 / FPS \\
 TiltingTime &= (MovingHeightThreshold / MovingTiltUnit) \times fps \\
 &\quad TiltPreDelay + TiltPostDelay \\
 MovingTiltUnit &= fps \times TiltSpeed \times MovingTiltUnitPerPixel \\
 TiltSpeed &= TiltMaxSpd / 1000 \\
 MovingTiltUnitPerPixel &= \partial(HeightImage / TiltAngleFocus)
 \end{aligned} \quad (3-11)$$

수식 (3-11)은 추적중인 이동물체의 PTZ 추적을 위해 PTZ 장비의 tilting 각속도, tilting 구동 및 신호 처리 지연시간, 가변 영상캡처 주기 및 크기를 고려하여 tilting 하는데 걸리는 시간을 계산하기 위한 수식이다.

FPS : 영상캡처 주기  
 fps : ms 단위의 영상캡처 시간  
 tilting : PTZ 장비가 위/아래로 이동하는 것  
 TiltPreDelay : PTZ 장비가 tilting 명령을 처리하기 전에 걸리는 시간  
 TiltPostDelay : PTZ 장비에 tilting 명령을 처리한 후 걸리는 지연 시간  
 TiltSpeed : 1° tilting 하는데 걸리는 밀리세컨드 단위의 시간  
 TiltMaxSpeed : 초당 tilting 가능한 수직각  
 TiltAngleFocus : 초점거리에 따른 세로 시계 각도  
 MovingTiltUnit : 영상캡처 주기 동안 tilting 하여 이동 가능한 화소 간격  
 MovingTiltUnitPerPixel : tilting 하여 영상 프레임에서 초당 이동 가능한 세로 길이  
 TiltingTime = tilting 각속도, tilting 구동 및 신호 처리 지연시간, 가변 영상캡처 주기 및 크기를 고려하여 tilting 하는데 걸리는 시간

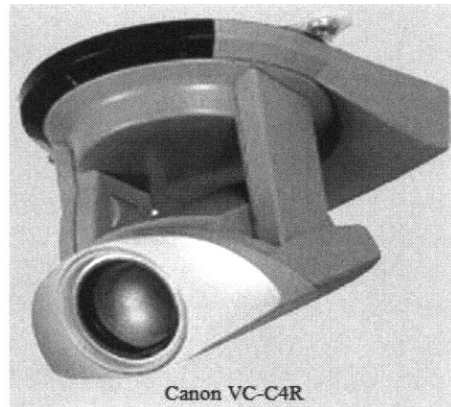
(그림 13) Tilting 소요 시간 산출 수식의 단위

(그림 13)은 수식 (3-11)에서 사용된 tilting과 관련된 단위를 상세히 설명한 것이다.

#### 4. 실험결과

##### 4.1 실험환경

본 논문에서 제안된 방법을 실험하기 위한 환경은 일반적으로 조도의 변화가 심하고, 잡음이 많이 포함된 실내 환경을 대상으로 하였다. (그림 14)는 본 시스템에서 사용된 PTZ 일체형 카메라이고 주요 스펙은 <표 3>과 같다.



(그림 14) PTZ 일체형 카메라

<표 3> PTZ 일체형 카메라 주요 스펙

Function	Value
Zoom	16x Power Zoom
F-number	f/1.4 to 2.8
View angle	47.5° to 3°
Pan range	+ or - 170°
Pan Speed	1 to 90°
Pan Pre/Post Delay	1ms
Tilt range	+10° to -90°
Tilt Speed	1° to 70°
Tilt Pre/Post Delay	1ms

본 논문에서는 이동물체가 이동하다 정지했을 경우 이동물체의 확대/축소 기능은 없기 때문에 카메라의 주요 스펙중에서 Zoom, F-number, View angle을 각각 1x, 2.8, 47.5°로 제한한다. 실험에 사용한 영상의 크기는 352 x 240 화소 크기이며, 영상캡처 주기는 30FPS로 설정하여 33ms 마다 영상이 캡처가 되도록 설정하였다. 한계이동거리는 80화소로 설정하였다. 실험 환경 설정과 카메라 성능으로부터 이동물체 추적에 필요한 최소 추적 시간을 구하기 위해 *PanningTimes*와 *TiltingTimes*을 수식 (3-10), (3-11)을 이용하여 산출하면 다음과 같다.

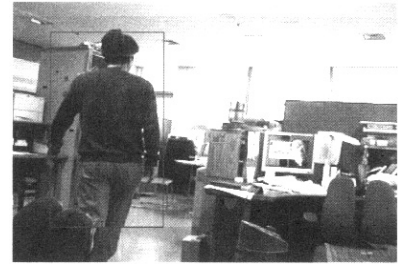
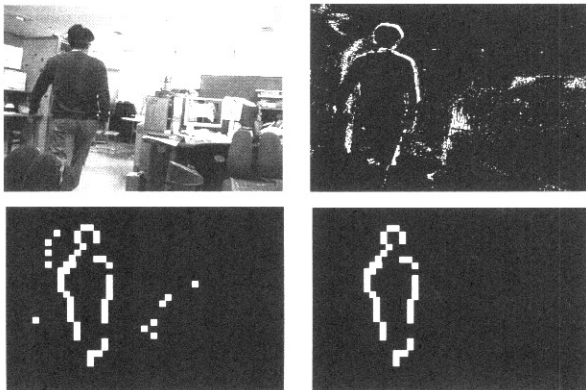
$$\begin{aligned}
 fps &= 1000 / FPS = 33 \text{ ms} \\
 PanSpeed &= 90 / 1000 = 0.09 \text{ ms} \\
 MovingUnit &= \theta(352 / 47.5) = 7 \text{ px} \\
 MovingPanUnit &= 33 \times 0.09 \times 7 = 20 \text{ px} \\
 PanningTimes &= (80 / 20) \times 33 + 1 + 1 = 134 = 4 \text{ frames}
 \end{aligned} \tag{4-1}$$

$$\begin{aligned}
 fps &= 1000 / FPS = 33 \text{ ms} \\
 TiltingSpeed &= 70 / 1000 = 0.07 \text{ ms} \\
 MovingUnit &= \theta(240 / 47.5) = 5 \text{ px} \\
 MovingTiltingUnit &= 33 \times 0.07 \times 5 = 12 \text{ px} \\
 TiltingTimes &= (80 / 12) \times 33 + 1 + 1 = 222 = 7 \text{ frames}
 \end{aligned} \tag{4-2}$$

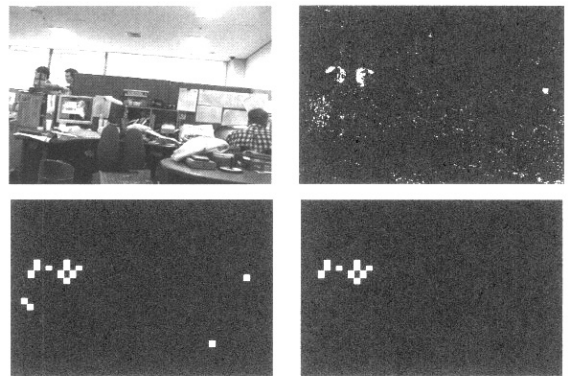
수식 (4-1)에서 이동물체가 좌우로 이동시 PTZ가 추적하는데 걸리는 시간은 134ms이며, 이를 영상캡처 주기로 변환하면 4frame이 된다. 즉, PTZ가 추적하는 4프레임 동안에는 영상캡처만 수행해야 한다. 수식 (4-2)에서 이동물체가 상하로 이동시 PTZ가 추적하는데 걸리는 시간은 222ms이며, 이를 영상캡처 주기로 변환하면 7 frame이 된다. 즉, PTZ가 추적하는 7프레임 동안에는 영상캡처만 수행해야 한다.

#### 4.2 실험결과

(그림 15)와 (그림 16)에서 (a)는 입력영상이다. (b)는 이중차영상으로써 배경영역을 제거한 화면이다. 하지만 조도의 변화가 심하기 때문에 잡음이 많이 포함된 것을 알 수 있다. (c)는 잡음이 많이 포함된 이중차영상에서 로컬움직임 감지를 수행하여 크기가 작은 잡음을 제거한 영상이다. (d)는 로컬움직임 감지를 수행한 영상에서 제거되지 않은 크기가 큰 잡음을 제거한 영상이다. (e)는 최종적으로 추적할 이동물체 후보의 영역이 표시된 것을 나타낸다.



(그림 15) 실험결과 영상. (a) 입력영상, (b) 이중차영상, (c) 크기가 작은 잡음을 제거한 영상, (d) 크기가 큰 잡음을 제거한 영상, (e) 추적할 이동물체



(그림 16) 실험결과 영상 II, (a) 입력영상, (b) 이중차영상, (c) 크기가 작은 잡음을 제거한 영상, (d) 크기가 큰 잡음을 제거한 영상, (e) 추적할 이동물체

#### 5. 결 론

본 논문에서는 CCD 카메라, 캡처 보드, PTZ 장비와 성능이 다양한 PTZ 장비를 사용하고 영상캡처 주기 및 크기가 가변적인 DVR과 같은 보안 시스템에서 실시간 처리가 가능한 이동물체 추적 시스템을 설계하고 구현했다.

기존의 연구들은 차영상 기반 방식과 움직임 기반 방식이 있다. 차영상 기반 방식은 실시간 처리가 가능하지만, 물체와 배경이 같이 변하는 상황에서 사용이 불가능하다. 움직임 기반 방식은 물체와 배경이 같이 변하는 상황에서 사용이 가능하지만, 많은 계산량으로 인해 관련 모듈을 하드웨어로 만들지 않고서는 실시간 처리가 어렵다. 반면에, 본 논문의 방식은 실시간 처리가 가능한 차영상 기반이고 물체와 배경이 같이 변하는 상황에서 사용이 가능하다.

다양한 모양과 크기, 색상을 가지고 이동하는 물체를 실시간으로 추적하고 다양한 성능을 가진 PTZ 장비와 영상캡

처 주기 및 크기가 가변적인 경우에도 추적이 가능하도록 다음과 같은 방법을 제안하였다.

- 영상포맷은 디지털 영상처리와 관련된 계산량을 최소화하기 위해 RGB에 비해 화소 당 비트수가 적은 YUV 계열의 YV12 포맷을 사용한다. 영상캡처시 RGB 영상포맷을 YV12로 변환하며, 이후 처리되는 모든 영상처리를 YV12를 기준으로 설계 및 구현하였다.
- 이동물체 분할은 차영상을 이중화하는 방법을 사용하였다. 이동물체 분할시 크기가 작은 잡음은 로컬 히스토그램 분석과 모폴로지 필터링으로 크기가 큰 잡음은 라벨링을 통해 제거하였다. 이동물체의 움직임이 감지되었을 경우에만 이동물체 정합을 수행하도록 하였고, 이중차영상을 8x8 블록 영상으로 등분하고 블록 내에서 로컬 움직임 감지 및 블록 간의 전역 움직임을 검사하도록 하였다.
- 이동물체 정합은 이동물체 분할에서 물체의 움직임이 감지되었을 경우에만 수행하며 추적중인 이동물체가 있을 경우 추적중인 이동물체의 히스토그램과 라벨링된 이동물체의 히스토그램간의 유사도 분석을 통해 정합하도록 하였다. 추적중인 이동물체가 없을 경우에는 라벨링된 물체들 중에서 가장 큰 것을 추적중인 이동물체로 자동선택하도록 하였다.
- 이동물체 추적은 영상 프레임 내에서 추적중인 이동물체의 이동거리의 차가 한계이동거리를 넘었을 경우에만 PTZ 제어를 통해 추적하도록 하였다. PTZ 제어를 통해 추적하는 동안에는 플래그를 두어 이동물체 분할 및 정합을 수행하지 않고 영상캡처만 하도록 하였다.
- PTZ 장비가 이동물체를 추적하는 동안 영상캡처를 하며 대기해야 하는 시간 산출시 PTZ 각속도, 장비 구동 및 신호 처리 지연시간의 PTZ 장비의 성능과 가변 영상캡처 주기와 크기를 고려하여 산출하도록 하였다. 이를 통해 다양한 PTZ 장비를 사용할 수 있고, 가변적인 영상캡처 주기 및 크기에서도 이동물체 추적이 가능하다.

위의 방법들로 인해 기존 이동물체 추적시스템에 비해 다음과 같은 장점을 갖는다.

- 차영상 기반이지만 배경과 물체가 이동하는 환경에서 이동물체 추적이 가능하다.
- 성능이 제각각인 PTZ 장비를 사용할 수 있다.
- 영상캡처 주기 및 크기가 가변적일 경우에도 사용이 가능하다.
- 효율적인 영상포맷 선정과 관련 모듈의 최적화, 선택적인 이동물체 추적방식으로 인해 실시간 지원이 가능하다.

본 논문에서 설계, 구현한 PTZ 제어에 의한 차영상 기반의 이동물체 추적 시스템은 카메라와 물체가 모두 이동하는

상황에서 사용할 수 있다. 이 이동물체 추적 시스템은 차영상 기반의 방식의 장점을 이용, 실시간 처리가 가능하기 때문에 DVR과 같은 실시간 처리가 필요한 보안 시스템 구축을 용이하게 해 줄 것이다.

## 참 고 문 헌

- [1] FOURCC, video codec and pixel format definitions, "http://www.fourcc.org/yuv.php"
- [2] 천인국, 윤영택, "영상처리", 기한재, 1999.
- [3] Gregory A. Baxes, "Digital Image Processing(Principles and Applications)", John Wiley & Sons, 1994.
- [4] Randy Crane, "A Sumplified Approach to Image Processing", Prentice-Hall, 1997.
- [5] N. Moghadam Charkari Hideo Mori, "A new approach for real time moving vehicle detection", Proceedings of the 1993 IEEE/RSJ Internaltional Conference on Intelligent Robots and System, pp.273-278, 1993.
- [6] N. Hoose and L. G. Willumsen, "Automatically extracting traffic data from video tape using the CLIP4 parallel image processor", Pattern Recognition Letters, Vol. 6, pp.199-213, 1987.
- [7] Toshinari Nonaka, Shigeru Kimura and Shinji Ojawa, "A moving object extraction using adapted frame integration", First Korea-Japan Joint Conference on Computer Vision, pp. 186-191, 1991.
- [8] Alton L. Gilbert, Michal K. Giles, Gerald M. Flachs, Robert B. Rogers and Yee Hsun U, "A real-time video tracking system", IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-2, No.1, pp.47-56, January, 1980.
- [9] R. Schalkoff and E. S. Mcvey, "A model and tracking algorithm for a class of video targets", IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-4, No. 1, pp.2-10, 1982.
- [10] T. Uno, M. Ejiri and T. Tokunaga, "A method of real-time recognition of moving objects and its application", Pattern Recognition, Vol.8, pp.201-208, 1976.
- [11] W. N. Martin and J. K. Aggarwal, "Survey: dynamic scene analysis", Computer Graphics and Image Processing, Vol. 7, pp.356-374, 1978.
- [12] 황신환, 이성욱, "자동 영상 추적기 개발 연구", 대한전자공학회 논문지, 제25권 제4호, pp.437-446, 1988.
- [13] 김계영, 최형일, "시공간 적응탐색에 의한 실시간 이동물체 추적", 대한전자공학회 논문지, 제31권 B편 제11호, pp.63-77, 1994.
- [14] Hyung-II Choi, Gye-Young Kim and Bong-Keun Kim, "Real-time tracking of moving object", The 3rd pacific Rim

International Conference on Artificial Intelligence, pp. 843-849, 1994.

- [15] William B. Thompson, Pamela Lechleider and Elizabeth R. Stuck, "Detecting moving objects using the rigidity constraint", IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-15, No.2, pp.162-166, 1993.
- [16] Vincent S. S. Hwang, "Tracking feature points in time-varying images using an opportunistic selection approach", Pattern Recognition, Vol.22, No.3, pp.247-256, 1989.
- [17] 강동중, 하중은, "Visual C++을 이용한 디지털 영상처리", 사이텍미디어, 2004.

### 양 금 석



e-mail : goldstone@emstone.com  
2002년 숭실대학교 컴퓨터학과(학사)  
2005년 숭실대학교 컴퓨터학과(석사)  
2002년~현재 (주)엠스톤 부설연구소 연구원  
관심분야: 실시간 멀티미디어 처리, 운영체제

### 양 승 민



e-mail : yang@realtime.ssu.ac.kr  
1978년 서울대학교 공과대학 전자공학과(학사)  
1978년~1981년 삼성전자(주) 연구원  
1983년 미국Univ. of South Florida 전산학(석사)  
1986년 미국Univ. of South Florida 전산학(박사)

1987년 미국Univ. of South Florida 조교수  
1988년~1993년 미국 Univ. of Texas at Arlington 조교수  
1993년~현재 숭실대학교 컴퓨터학부 교수  
1996년~1998년 국회도서관 정보처리국장 파견  
관심분야: Real-Time System, Operating System, Fault-Tolerant System