

OpenCV를 이용한 PTZ 카메라 기반의 실시간 객체 추적 시스템

조지훈*, 이상구*

요약

본 논문에서는 PTZ(Pan-Tilt-Zoom) 카메라 기반의 영상 객체 추적을 위한 강인한 실시간 보안 시스템을 구현한다. 이를 위해 영상 객체 추적을 위하여 탐지된 이동 객체의 HSI 컬러 영상 분포에 기반을 둔 CAM shift 추적 알고리즘을 사용한다. CAM shift 알고리즘은 속도가 빠르고 강인한 성능을 가지므로 실시간 추적에 매우 효율적이다. 제안된 시스템에서는 영상처리와 PTZ 프로토콜의 접근 및 이동객체의 영상을 모니터 스크린의 가운데 부분에 배치하도록 PTZ 카메라의 위치를 제어하기 위한 RS-485 통신을 위하여 C++ 과 OpenCV 라이브러리를 사용한다. 본 시스템은 보다 큰 범위에서 연속적으로 움직이는 객체 추적을 위한 효율적이고 빠른 영상 감시 시스템에 활용될 수 있다.

PTZ Camera Based Real-time Object Tracking System Using OpenCV

Ji-Hoon Jo*, Sang-Gu Lee*

ABSTRACT

In this paper, we implement a robust real-time surveillance system of image object tracking for Pan-Tilt-Zoom(PTZ) cameras. For image object tracking, we use the CAM shift tracking algorithm based on the HSI color image distribution of detected moving objects. CAM shift algorithm is very efficient for real-time tracking because of its fast and robust performance. In the proposed system, C++ and OpenCV libraries are used for image processing and accessing the PTZ protocol and RS-485 communication for controlling the positions of PTZ camera in order to arrange the moving object images in the middle part of the monitor screen. This system can be applied to an effective and fast image surveillance system for continuous object tracking in a wider area.

Key Words : PTZ camera, Object tracking, Mean shift, CAM shift, OpenCV

* 한남대학교 컴퓨터공학과(✉inver83@nate.com)

· 제1저자(First Author) : 조지훈 · 교신저자(Correspondent Author) : 이상구

· 접수일(2012년 1월 10일), 수정일(1차 : 2012년 2월 8일), 게재 확정일(2012년 2월 10일)

I. 서론

CCTV 기술은 영상보안 분야에서 가장 큰 시장을 차지하고 있으며 2000년도부터 급속한 성장을 거듭하고 있는 분야이다. 기존의 CCTV 영상보안 기술은 고정형 카메라를 통하여 모니터링하고 저장된 DVR을 통해 관리를 하는 시스템이었고 이러한 방식은 비용, 시간적인 측면에서 효과적이지 못한 시스템이었다. 최근에 보안에 대한 관심이 늘어남에 따라 CCTV의 설치가 점차 보편화 되고, 나아가 인공지능, 객체인식 등의 기술을 결합한 영상보안이 요구되고 있다. 특히 복잡한 환경에서도 객체를 놓치지 않고 연속적으로 추적할 수 있는 기술이 많이 연구되고 있다.

기존의 CCTV 시스템은 고정형 카메라를 사용하기 때문에 객체의 이동, 가려짐, 조명과 같은 주변환경에 매우 민감하다는 단점이 있다.[1][2] 최근 카메라 및 디바이스의 발전으로 고성능, HD급의 화질을 갖춘 카메라들이 출시되고 있고 그중에서도 PTZ 카메라는 360도 회전 및 줌 인-아웃(in-out)이 가능하기 때문에 기존의 고정형 CCTV가 갖는 단점인 단방향 감시를 보완하여 카메라 주변의 모든 영역에 대해서 모니터링이 가능하다는 장점이 있다. 본 논문에서는 이동 객체를 추출하는 방법으로는 CAM shift 알고리즘을 사용한다. CAM shift 알고리즘을 이용한 연구들은 꾸준히 진행되어 왔다. '컬러 확률 분포를 이용한 객체 추적 알고리즘[3]'의 경우 고정형 카메라를 사용하였기 때문에 객체가 카메라의 범위를 벗어날 경우 객체를 놓쳐 버린다는 단점이 존재하였으며, '정보융합을 위한 객체 추적[4]'에서는 모션 템플릿과 CAM shift 알고리즘을 융합하여 객체의 인식률은 높아졌지만 계산이 느리다는 단점이 존재하였다. '배경 생성 기법을 이용한 다중 카메라 객체 추적 시스템[5]'의 경우 다중 카메라를 사용하여 넓은 범위의 구역을 감시 할 수 있다는 점이 있었으나 초기비용 및 입력영상에서의 개별 물체들을 분할해야 한다는 점이 존재한다. 본 논문에서는

PTZ 카메라 환경에서 객체를 검출하고 추적하는 방법을 OpenCV로 구현하며, PTZ 프로토콜과 RS-485 통신을 이용하여 카메라를 제어하는 방법으로 화면상의 객체의 위치를 화면 가운데에 위치하게 하는 실시간 추적 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 CAM shift와 PTZ 카메라를 이용한 객체 추적방법에 대하여 알아본다. 3장에서는 실험 결과 및 분석에 대해 알아 보며, 4장에서는 결론을 다룬다.

II. 객체 추적 방법

움직이는 객체를 추적하는 방법은 추적하고자 하는 객체의 특징 값의 선택에 따라 나뉜다[3]. 첫 번째 방법으로 컬러 기반 방법(color based method)은 추적 객체의 컬러 분포를 바탕으로 특정 컬러를 프레임에 따라 연속 추적하는 것으로 알고리즘이 간단하면서도 속도가 다른 방법에 비해 월등히 빠르기 때문에 실시간 응용분야에 많이 사용된다. 영역 기반(area based method) 방법은 현재의 프레임과 다음 프레임에서 특정 영역에서의 픽셀차이에 따라 물체를 검출하게 된다. 또한, 에지(edge) 필터링, 허프 변환(Hough transform) 등을 통해 경계 윤곽선을 표현하고 이를 동적으로 갱신하면서 추적하는 능동 윤곽선 기반(active contour based method) 방법이 있으며, 물체 전체를 추적하는 것이 아닌 특징점 또는 그 점들의 집합, 일정한 방향성을 갖는 벡터 등을 기반으로 한 특징 기반(feature based method) 방식도 대표적인 객체 추적의 방법으로 손꼽힌다.

2.1 CAM shift 알고리즘

기존의 객체를 추적하는 알고리즘은 칼만 필터(Kalman filter), Mean shift 알고리즘, Particle filter 등

여러 가지 방법이 존재하지만, 계산량이 많아진다는 단점이 존재 하였으며, 객체를 추적하는데 많이 쓰이는 Mean shift 알고리즘의 경우 추적 객체의 회전 각도, 윈도우의 크기 등에 민감하다는 단점이 존재한다.

본 논문에서는 객체를 추적하는 방법에 있어서 색 정보를 바탕으로 추적하는 알고리즘인 CAM shift 알고리즘을 사용하였다. CAM shift 알고리즘은 HSV 모델을 바탕으로 원하는 부분의 색 정보를 추출하여 연속되는 프레임에서 색 정보를 비교하여 추적하고자 하는 객체를 연속적으로 추적할 수 있는 알고리즘이다. CAM shift 알고리즘은 기본적으로 Mean shift 알고리즘을 사용하면서도 객체의 크기 및 각도를 같이 계산한다. 본 논문에서 사용하는 PTZ 카메라로 추적할 때 객체의 크기에 영향받지 않고 추적하기 위해서는 CAM shift 알고리즘이 적합한 알고리즘이라고 할 수 있다.

CAM shift 알고리즘은 다음과 같은 단계로 나뉘어 수행된다.[6][7]

step.1 초기 탐색 윈도우의 위치 및 크기를 정한다.

step.2 탐색 영역 내에서 컬러 분포의 중심을 구한다.

step.3 탐색 영역에 해당하는 Hue 컬러 공간의 Hue의 히스토그램을 계산하고 컬러 확률 분포를 계산한다. 이때 컬러 확률 분포 기반으로 중심점을 찾아 탐색 윈도우를 위치시키고 영역을 찾는다. 만약에 수렴하지 않으면 step.3을 반복하며, 수렴하면 step.4로 이동한다.

step.4 이어지는 프레임에서 탐색 윈도우의 위치와 크기, 회전 각도를 알아내고 탐색 영역 크기를 재조정하여 중심 위치를 결정한다.

step.5 step.2에서 step.4의 과정을 반복한다.

앞의 CAM shift 트래킹 알고리즘을 프로시저 형태로 표현하면 다음과 같다.

```

CAM - shift tracking()
kernel = kernel_initialize();
for image_frame = 1: n
features = object_features(image_frame);
repeat
[M00M01M10] = moments(kernel, features);
Cw = centroid(kernel);
Cg = [M01/M00M10/M00];
window_shift(kernel, Cg - Cw);
kernel_size(kernel, width(M00),height(M00));
until |Cg - Cw| <= e;
end
    
```

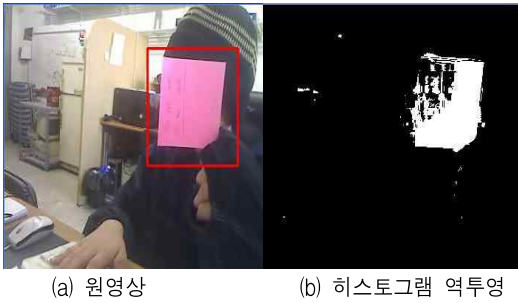
여기서 M_{00} 는 0차 모멘트 이고, M_{01} 과 M_{10} 은 각각 1차 모멘트(x)와 1차 모멘트(y)이다.

CAM shift 알고리즘은 객체의 컬러 정보, 즉, 컬러 확률 분포를 이용해 위치뿐만 아니라 회전 각도, 크기까지 빠르게 계산해서 추적할 수 있다.

객체의 추출은 기본적으로 배경영상과 입력영상의 차를 구함으로써 이루어진다. 차 영상을 통하여 객체가 추출이 되면 미디언 필터링 및 모폴로지를 통하여 전처리 과정을 거친 후 클러스터링을 통하여 객체를 추출한다. 다음 과정으로 step.4의 단계와 같이 다음 프레임에서의 객체간 유사도 검사를 통해 추적 객체로 선정한다. 객체간 유사도 검사는 최소 편차값을 통해 구한다. 이때 검출된 객체의 크기가 추적 객체의 크기와 다르다면 크기 조정을 수행한 후 유사도 검사를 수행한다.

그림 1은 CAM shift를 사용하여 실제 객체를 추적하는 것을 보여준다.

(a)는 원 영상을, (b)는 색상(Hue), 채도(Saturation)의 히스토그램 유사도 역투영(Backprojection)을 나타낸다.



(a) 원영상 (b) 히스토그램 역투영

그림 1. CAM shift 추적
Fig. 1 CAM shift tracking

CAM shift 알고리즘의 경우 기존 Mean shift 알고리즘이 갖는 단점인 크기, 각도에 취약하다는 단점을 보완하여 추적하고자 하는 객체의 크기가 변하거나 각도가 달라진다 하더라도 연속적인 추적이 가능하다. 그림 2는 객체를 다양한 크기, 각도에서 추적한 것이다.



그림 2. 다양한 객체 크기에서의 CAM shift 추적
Fig.2 CAM shift tracking in various object sizes

2.2 PTZ 카메라의 제어

PTZ 카메라의 제어는 PC의 메인 프로그램에서 이

루어진다. 원 배경에서 추출한 객체가 이동하였을 경우 프로그램에서는 이를 판단, 객체를 연속적으로 추적하기 위해 명령 패킷을 RS-485 통신을 통하여 카메라에게 전송한다. 카메라는 객체의 움직임에 따라 스텝모터를 작동시키며 객체를 화면상의 가운데에 위치하게 함으로써 객체의 연속적인 추적이 가능하게 된다.

본 논문에서 사용된 PTZ 카메라는 삼성테크윈사의 SPD-1000 모델이다. 이 PTZ 카메라의 명령어 패킷 포맷은 표 1과 같다.

표 1. 명령어 패킷 포맷
Table 1. Command packet format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
STX	C A M ID	Host ID	Cmd-1	Cmd-2	Data1
Byte7	Byte8	Byte9	Byte10	Byte11	
Data2	Data3	Data4	ETX	C h e c k -sum	

STX 바이트는 A0H이고, ETX 바이트는 AFH이다. 에러판별을 위한 Check-sum은 다음과 같이 계산된다.

$$\text{Check-sum} = \sim((\text{byte2} + \text{byte3} + \dots + \text{byte9}) \& \text{FFH})$$

실제 카메라를 제어하기 위해서는 RS-485 통신을 통해 표 1의 11 바이트를 보내주어야 한다. 예로 speed 34로 Pan을 오른쪽으로 제어하기 위해서는 A0 01 00 00 02 22 00 00 00 AF DA를 보내주면 된다. 본 논문에서는 원하는 P, T, Z 값의 제어를 위해 앞의 프로토콜 데이터들을 lookup table의 형태로 만들어 제어하였다.

그림 3은 PTZ 카메라를 제어하기 위해 패킷을 전송하는 PTZ 제어 프로그램의 화면이다.

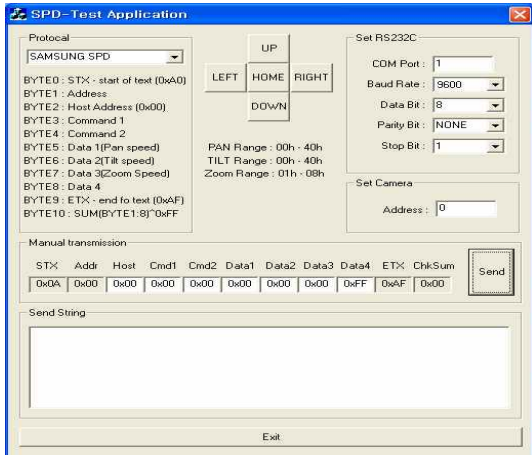


그림 3. PTZ 제어 프로그램
Fig. 3 PTZ control program

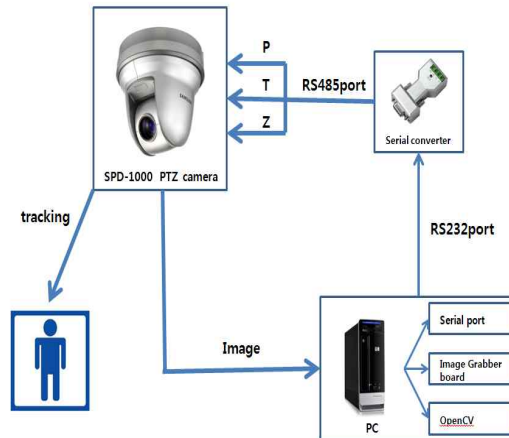


그림 4. 제안된 하드웨어 시스템
Fig. 4 Proposed hardware system

III. 실험결과 및 분석

제안하는 움직이는 객체 추적 시스템을 실험하기 위한 환경은 표 2와 같다.

표 2. PC 개발환경
Table 2. PC development environment

	개발환경
H/W	CPU : Intel(R)Core(TM)2 2.0GHZ Memory : 2GB RAM
OS	MS Windows XP Professional SP3
실험도구	MS Visual Studio 2008, OpenCV
실험 카메라	SAMSUNG Techwin SPD-1000

본 논문에서의 제안된 전체적인 하드웨어 시스템의 구성도는 그림 4와 같다.

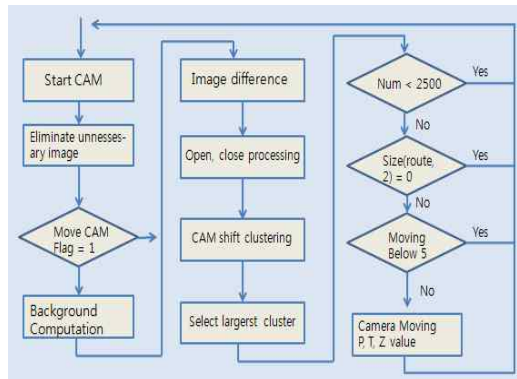


그림 5. 전체 시스템의 개요도
Fig. 5 System flow chart

그림 5와 같이 먼저 카메라가 객체를 인식하게 되면 배경을 제거 직전 프레임과의 비교, 배경을 제거한 후 모폴로지 및 미디언 필터링을 통한 전처리 과정을 거친 후, CAM shift 알고리즘을 통한 클러스터를 추출한다. 클러스터 추출 과정에서 클러스터의 크기는 2500을 반드시 넘어야 한다. 클러스터 과정에서 2500보다 작은 클러스터들은 무시한다. 객체가 추출이 되면 객체의 움직임에 따라서 미리 지정된 lookup table을 통

해 PTZ 값을 RS-485 통신을 통해 전달한 후, PTZ 카메라는 수신된 PTZ 값들을 통해서 움직이게 된다.

실험을 위해 삼성테크윈사의 SPD-1000 PTZ 돔 카메라를 사용하였다. 해상도는 320 × 200의 동영상상을 전송받는다. 제안 알고리즘의 성능 평가를 위해 실험 대상은 여러 실험 객체 대상을 바꾸어 실험하였다. 그림 6은 PTZ 카메라 상에서의 객체 추적 실험 영상이다.

왼쪽은 히스토그램 유사도 역투영 (Backprojection) 을 나타내며 오른쪽 영상은 PTZ 카메라가 객체를 추적하는 영상이다.



그림 6. PTZ카메라를 이용한 객체 추적
Fig. 6 Object tracking using PTZ camera

그림 6에서 보는 것과 같이 객체의 움직임에 따라 카메라가 따라서 움직여 추적하고자 하는 객체가 화면의 중앙 부분에 위치되는 것을 볼 수 있다. 만일 추출한 이동 객체가 이동하였을 경우 프로그램 상에서 이를 판단하고 카메라의 모터를 움직이기 위하여 명령 패킷을 RS-485 통신을 통하여 카메라에게 전송하며, 카메라의 모터가 움직여서 객체를 화면상의 가운데에 위치하게 함으로써 객체의 추적이 가능하게 된다.

표 3은 그림 6의 실험에 사용된 PTZ 값들을 나타낸 것이다.

표 3. PTZ 값
Table 3. PTZ values

Frame #	Pan	Tilt	Zoom
115th	271	15	1.0
168th	246	19	1.0
199th	192	24	1.0
231th	159	22	1.0
276th	114	18	1.0

IV. 결론

본 논문에서는 PTZ 카메라 기반의 CAM shift 알고리즘을 이용하여 감시 영역 내의 객체 움직임을 판별하여 실시간으로 동작하는 영상 감시 시스템을 구현

하였다. CAM shift 추적 알고리즘은 빠르고 안정적인 성능으로 실시간 추적에 적합하며 추적하고자 하는 객체의 이동이나 회전 등에도 강건하게 추적을 할 수 있었다.

그러나 CAM Shift 알고리즘은 배경 및 객체 간의 컬러 분포에 민감하고 배경 물체에 의한 가려짐 및 객체의 크기 변화 수용에 어려움 등이 있고, 잡음과 조명의 영향으로 인하여 편차가 발생하는 경우도 있다. 향후의 연구로서는 배경 설정 등의 개선을 통한 주변 환경 및 조명에 더욱 강한 커널 함수의 선택과 이에 따른 효율적인 추적 기법에 대한 연구의 필요성이 있다.

참고문헌

- [1] D. Comaniciu, V. Ramesh, P. Meer, "Real-time tracking of non-grid objects using mean shift", Computer Vision and Pattern Recognition, 2000.
- [2] D. Comaniciu, and P. Meer "Mean Shift: A Robust Approach Toward Feature Space Analysis", IEEE. Transaction on Pattern Analysis and Machine Intelligence, Vol.24, No.5, 603-619, 2002
- [3] Hye-Youn Lim, Dae-sung Kang "A Study of Advanced Tracking Algorithm using Color Probability Distribution for Objects Tracking System", Korean Institute of Information Technology, Vol.8, No.2 pp.153-158, 2010.
- [4] Jinhyung Lee, Seongwon, Jeamin Kim and Sun-Tae Chung, "Object Tracking Using Information Fusion", Korea institute of intelligent systems, Vol.18, No.5, pp.666-671, 2008
- [5] Hyuntae Jo, Jaenee Jang, Namoh Kang, Joonki Paik, "Implementation of Object Tracking System with Multi Camera by Using Background Generation Technique", The institute of Electronics Engineers of Korea, Vol.31 No.1, pp.947-948, 2008
- [6] Gary R, Bradski, "Computer Vision Face Tracking For use in

a perceptual User Interface", Microcomputer Reseach Lab, 2002

- [7] J. G. Allen, R.YD Xu, J. S. Jin "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces," Pan-Sydney Area Workshop on Visual Information Processing, pp. 3-7, 2003
- [8] C. Stauffer and W. Grimson, "Adaptive background mixture. models for real-time tracking", IEEE. Computer Vision and Pattern Recognition, 1999.
- [9] Nicole M. Artner, "A Comparison of Mean Shift Tracking Methods", CESC2008, pp.197-204, 2008.
- [10] Sang Gu Lee, Ji hoon Jo, "An Object Tracking System for PTZ Camera Using CAM shift", Applied Mathematics in Electrical and Computer Engineering, pp. 191-194, Jan, 2012.

감사의 글

본 논문은 2011년도 한남대학교 교비연구비 지원에 의해 수행되었음.

저자소개



조지훈 (Ji Hoon Jo)

2011년 8월 : 한남대학교 컴퓨터공학과 졸업

2011년~현재 한남대학교 컴퓨터공학과 석사과정
 ※ 관심분야: 객체추적, 영상처리, 영상 인식



이상구 (Sang Gu Lee)

1983년~현재 한남대학교 컴퓨터공학과 교수
 ※ 관심분야: 영상처리, 임베디드 시스템, 패턴인식