

라즈베리 파이를 이용한 OpenCV 기반 CMT 알고리즘의 객체 추적 기법

송기범*, 양용준*, 이상구*

*한남대학교 컴퓨터공학과

e-mail:ssonggibum@gmail.com

An Object Tracking Technique on OpenCV-based CMT Algorithm Using Raspberry Pi

Gi-Beom Song*, Yong-Jun Yang*, Sang-Gu Lee*

*Dept of Computer Engineering, Hannam University

요 약

동영상 플랫폼 확산에 따라 영상 콘텐츠 수요와 공급이 폭발적으로 성장하여 영상 콘텐츠 확산에 큰 영향을 미치고 있다. 이와 같은 콘텐츠 확산으로 인해 전문가의 영역에 해당하는 영상 제작 기술이 일반적인 기술의 범주로 인식될 만큼 영상 처리 및 제작에 대한 기술의 발전이 나날이 급증하고 있으며 이러한 기술의 발전에 따라 사람의 수작업을 통해서만 영상을 조절하던 과정 또한 객체 추적 기술을 활용함에 따라 자동화 과정이 가능하게 되었다. 본 연구에서는 객체 추적 알고리즘 중 특징 점을 키 포인트로 나눠 객체를 추적하는 OpenCV 기반 CMT 알고리즘으로 라즈베리파이를 활용하여 객체를 추적하는 기법을 소개한다. 본 시스템은 방송용 카메라의 객체 추적에 잘 활용될 수 있다.

1. 서론

최근 Youtube 등 동영상 플랫폼 확산에 따라 영상 콘텐츠 수요와 공급이 폭발적으로 성장하고 문자와 사진으로 구성되어 표현되던 기존 콘텐츠들이 영상으로의 전환이 가속화됨에 따라 영상 콘텐츠 생태계의 성장 또한 가속화 되고 있다. 이와 같이 가속된 발전에 따라 전문가의 영역으로 인식되던 영상 촬영 및 편집 기술이 전공지식 없이 일반인도 쉽게 접할 수 있는 일반적인 기술의 범주로 인식되고 있는 추세이며 전문 장비가 없더라도 스마트폰, 디지털 카메라 같은 동영상 촬영장비의 보급에 따라 누구나 쉽게 동영상 콘텐츠의 제작과 편집이 가능한 실정이다.

하지만 이러한 발전에도 불구하고, 아직까지 영상 콘텐츠 제작을 위해 사람의 개입이 많이 요구되고 있다. 영상을 통해 전달하고자 하는 메시지의 중요 포인트가 되는 부분을 강조하고 시청자들에게 잘 전달하기 위해서는 중요 포인트를 영상에 잘 담아내는 것이 중요하기에 일반적으로 메시지의 포인트가 되는 부분은 영상의 중앙에 위치하게 된다. 이러한 작업을 위해 영상 촬영 담당자의 많은 노력과 개입이 요구되며 이러한 요구는 비효율성과 부정확한 결과를 불러일으킬 수 가능성이 존재한다.

CMT 알고리즘은 객체를 추적하기 위해 지정된 관심 대상에 대한 범위 내에서 존재하는 특징 점을 키 포인트로 나눠 객체를 추적하는 기술로 객체를 인식할 경우 지정된 컬러의 박스를 이용하여 인식한 객체를 나타내며 객

체를 인식하지 못할 경우에는 영상에 박스를 표현하지 않고 Terminal에 인식하지 못하고 있는 상황을 의미하는 텍스트를 제공함으로써 객체의 인식이 이루어지고 있지 않는 상황을 표현한다.

본 연구에서는 객체 추적 기술로 알려진 CMT 알고리즘을 기반으로 객체를 추적하고 자동으로 영상장비를 제어하기 위해 Raspberry Pi를 이용하여 실시간으로 영상을 추적하면서 추적된 객체의 중앙좌표와 프레임의 중앙좌표를 비교하여 영상장비가 이동하여야 하는 수치에 대한 정보를 도출하고 도출된 정보를 기반으로 방송 영상 장비가 지정된 객체를 따라 추적할 수 있음을 검증하였고 이러한 검증 사례를 기반으로 영상 처리 분야에 기여하고자 한다.

2. 관련연구

2.1 OpenCV

OpenCV(Open Source Computer Vision)는 오픈 소스 컴퓨터 비전 라이브러리로 실시간 이미지 프로세싱에 중점을 둔 영상 처리 라이브러리이다[3]. 영상 처리란 컴퓨터가 알고리즘을 사용하여 물체 또는 글씨를 인식하기 위한 컴퓨터 과학(Computer Science) 분야 기술로 2진화 되어 있는 영상 또는 사진들을 알고리즘을 통해 정보를 추출하거나 가공하는 것을 뜻한다. 하지만 영상이나 사진에 있는 정보를 추출하기 위해서는 대략 심화과정 이상의 수학적 접근이 요구되기에 OpenCV가 나오기 전까지는 컴퓨터 비전 학문에 대한 연구가 많이 진행되지 않았다.

이러한 문제점을 극복하고자 2000년 Intel社에서 다각적인 수학적 기능들을 라이브러리(Library)로 만들어 개발자들이 손쉽게 컴퓨터 비전이라는 학문에 접근하고 연구하도록 OpenCV 라이브러리를 개발하였다[4]. OpenCV는 얼굴 검출과 인식, 객체 인식, 객체의 3D 모델 추출, 영상 검색, 적목 현상 제거 등 수 많은 응용 분야에 이용되고 있으며 본 연구에 기반이 되는 CMT 알고리즘 또한 OpenCV 라이브러리에 기반 한다.

2.2 CMT

CMT(Consensus-based Matching and Tracking) 알고리즘이란 관심 대상의 특징 점을 찾아 키포인트로 분해한 뒤 각 프레임에서 일치하는 키포인트를 추적하는 알고리즘이다. 객체를 추적하는 기법으로는 CMT 알고리즘 외에도 mean shift, CAMshift 등의 방식이 있다. mean shift 알고리즘의 경우 local minimum에 빠지기 쉽다는 문제점이 존재하며 대상의 크기 변화 및 형태 변화에 반응할 수 없는 단점이 존재한다. CAMshift 알고리즘의 경우에는 탐색 윈도우 크기를 조절할 수 있는 장점이 있지만 조도변화가 심하거나 잡음이 심한 배경에 있는 곳에서는 성능이 떨어지는 단점이 있다. 하지만 CMT 알고리즘은 추적하고자 하는 객체를 키 포인트로 분해하여 초기에 있었던 키포인트를 찾는 방식으로 optical flow 알고리즘을 사용하여 이전 프레임으로부터 지금 프레임까지의 키포인트를 추적하거나 OpenCV에서 제공해주는 descriptors를 활용하여 전체 프레임을 대상으로 비교한다. 그리고 이 두 가지 방식을 조합하여 추적하고자 하는 객체 타겟과 일치도가 가장 높은 객체를 찾는 방법을 사용한다[1].

본 연구에서는 CMT 알고리즘에 기반 하여 객체추적 실험을 진행하기 위해 Python 언어를 기반으로 제작되어진 스크립트를 이용하였으며[2], 방송용 카메라에서의 객체 추적에 이용하기 위해 프로그램의 일부분을 수정하여 구현하였다.

3. CMT 기반 라즈베리 파이를 활용한 객체추적 기법

3.1 환경구성

우리는 본 연구에서 라즈베리 파이를 활용하여 객체추적 기법의 실험을 진행하기 위해 라즈베리 파이2와 파이 카메라를 사용하였으며 운영체제는 NOOBS 2.4.3 버전을 통해 진행하였다. CMT 알고리즘의 기반이 되는 컴퓨터 비전(Computer Vision) 라이브러라인 OpenCV의 경우에는 2.4.13 버전을 설치하여 사용하였다.

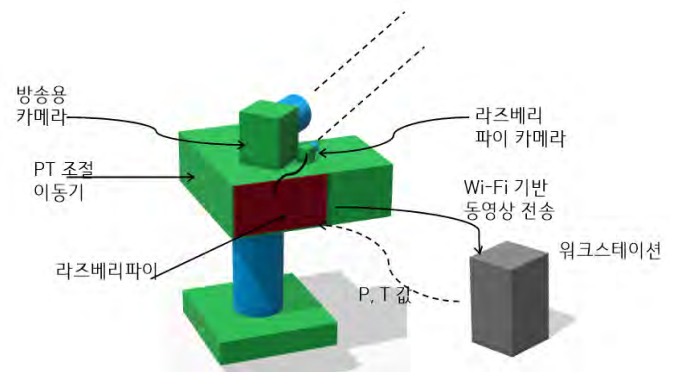
또한 본 논문에서 진행하려고 하는 방송용 카메라의 자동 객체추적 시스템의 전체적인 구조는 [그림 1]과 같다 먼저 방송용 카메라를 “PT 조정 이동기”(PT adjustment translator) 라고 하는 부분 위의 카메라 고정 나사 부분에

연결한다. 그리고 라즈베리 파이 카메라도 방송용 카메라와 똑같은 방향을 보도록 위치를 고정시킨다. 파이 카메라의 역할은 똑딱이 카메라처럼 view finder의 역할을 한다. 이때 파이 카메라에서 얻은 실시간 동영상에 객체를 지정하는 프레임 마킹을 하여 객체를 추적하는 과정을 거친다. PT 조정 이동기는 내부에 스텝 모터가 있어 pan과 tilt의 기능을 수행하는 액추에이터의 역할을 한다. 즉 파이 카메라에서 영상을 얻은 후 객체 추적 알고리즘을 수행하여 pan, tilt가 조정되어 방송용 카메라에서 원하는 객체를 항상 출력터미널의 중앙부분에 배치할 수 있도록 영상을 촬영할 수 있다.

[그림 1]의 시스템 구조에서 다음과 같이 2가지의 방법으로 객체를 추적할 수 있다. 첫째로, 라즈베리 카메라에서 얻은 영상을 stand alone 방식으로 라즈베리 파이 자체에 OpenCV 기반 CMT 알고리즘을 구동하여, 라즈베리 파이의 GPIO 포트를 거쳐 PT 조정 이동기의 스텝 모터를 제어하여 pan, tilt를 조정하는 방법이다.

둘째로, 파이 카메라에서 얻은 영상을 Wi-Fi로 워크스테이션으로 스트림 형태로 보내서, CMT 알고리즘을 수행하여 객체를 추적하고, 다시 무선으로 라즈베리 파이의 GPIO 포트로 pan, tilt의 이동에 필요한 데이터를 보내주는 방식이다. 이 방식에서는 라즈베리 파이를 영상을 보내주는 기능(Wi-Fi로)과 PT 조정 이동기를 위한 컨트롤러로써 사용하는 방법이다.

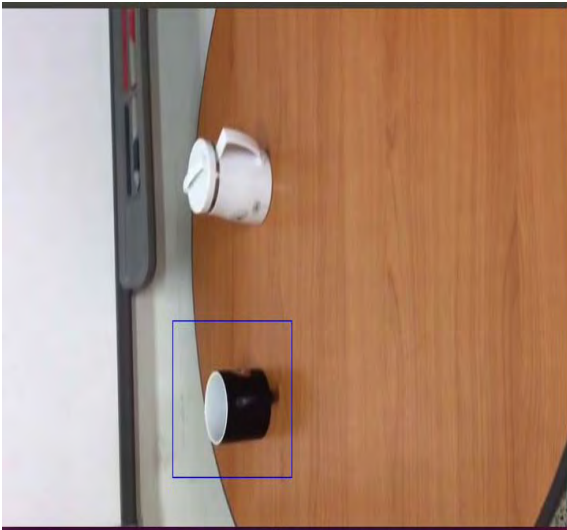
본 논문에서는 첫 번째 방식인 라즈베리 파이를 stand alone 으로 사용하는 방법을 사용하여 구현하는 것을 보인다.



(그림 1) System Architecture

3.2 객체추적 기법

CMT 알고리즘을 기반으로 객체를 추적하는 방법은 크게 2가지로 나뉜다. 사전에 촬영되어 파일로 존재하는 동영상을 대상으로 추적하는 방법과 연결되어 있는 카메라 디바이스를 이용하여 실시간으로 추적하는 방법이다. 우리는 본 연구에서 사전에 동영상 촬영을 통해 얻은 동영상 파일을 이용하여 실험을 진행 하였으며 스크립트를 시작하는 순간 아래 [그림 2]과 같이 동영상의 첫 frame에서 추적하고자 하는 객체를 설정할 수 있다.



(그림 2) 객체 추적 설정

또한 우리는 룩업 테이블을 만들어 영상장비가 각 축에 대해서 이동해야하는 거리를 1차 함수식을 통해서 변환하였다. 1차 함수식의 계수는 장비의 종류나 상태에 따라 변경될 수 있기 때문에 변수 수치 조절을 통해 각 장비에 맞는 이동 좌표를 도출할 수 있다.

```
# look up table
x_weight = 0.29
x_bias = 0.5
y_weight = 0.43
y_bias = 0.8

status, im0 = cap.read()
full_size = im0.shape

x_arr = [i * x_weight + x_bias for i in xrange(full_size[1])]
y_arr = [i * y_weight + y_bias for i in xrange(full_size[0])]
```

(그림 3) 룩업 테이블 코드

객체 추적 시에 축에 대해서 이동해야하는 거리 값을 생성해 놓은 룩업 테이블에서 가져올 수 있어 계산 없이 빠르게 이동거리를 구할 수 있으며 x, y 축에 대해서 개별적으로 도출되는 수치를 기반으로 영상 장비의 위치를 조절할 수 있도록 관련 디바이스에 데이터를 전송하여 추적된 객체를 영상의 중앙에 오도록 할 수 있다.

```
Current center point in frame : (360, 202)
Current center point in recognized box : (301, 163)
To locate the tracked box to center, please move 59 point to left and 39 point t
o down

look up table
x value :left 17.61
y value :down 26.17

Current center point in frame : (360, 202)
Current center point in recognized box : (298, 161)
To locate the tracked box to center, please move 62 point to left and 41 point t
o down

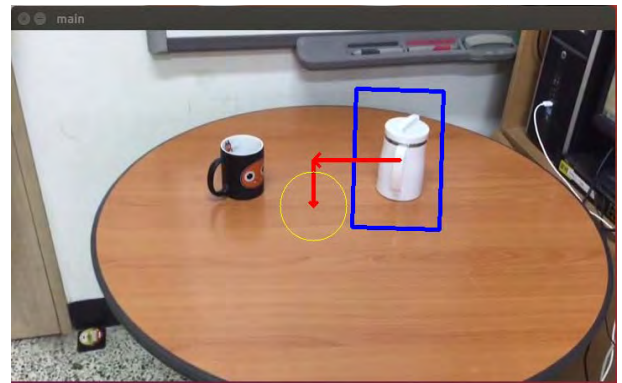
look up table
x value :left 18.48
y value :down 27.46
```

(그림 4) 룩업 테이블 기반 이동 좌표 도출

아래 [그림 5]의 화면을 통해 추적되고 있는 객체의 주변에는 파란색 사각형 박스가 표현되고 있고 사각형 박스의 정중앙을 기준으로 가운데 있는 노란색 원의 정중앙과

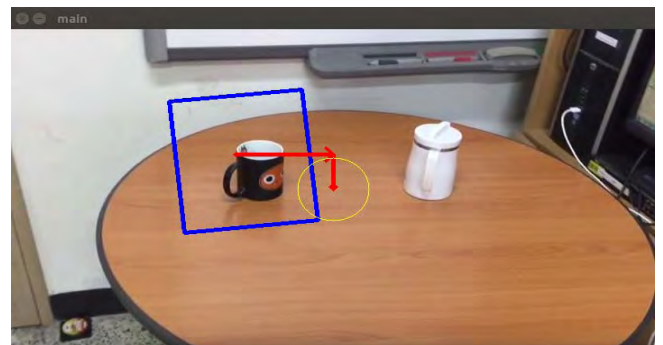
선으로 이어져 있는 것을 확인할 수 있다. 우리는 사용자가 추적하고자 하는 객체가 추적될 때 마다 나타나는 파란색 사각형 박스를 기준으로 가운데의 중심 값을 추출하여 프레임의 정중앙 좌표와 비교하여 객체가 화면의 중앙에 오기 위한 이동 경로를 표시하는 기능 또한 추가적으로 구현하였다.

이를 통해 기계가 아닌 사람이 장비를 움직이게 될 경우 이동 경로에 대한 정보를 통해 손 쉽게 확인할 수 있다.

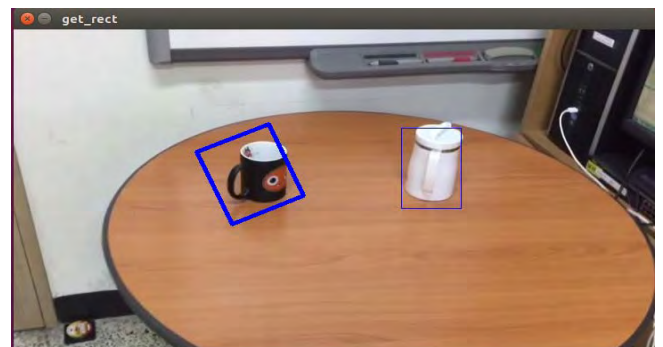


(그림 5) 객체 이동경로 정보 제공

또한 우리는 아래 [그림 6,7]과 같이 객체를 추적하는 도 중 추적하고자 하는 객체를 변경할 수 있는 사항을 고려하여 추적 객체 변경 기능을 추가하였다.



(그림 6) 객체 추적 재설정 과정-1



(그림 7) 객체 추적 재설정 과정-2

객체를 재 추적하기 위해 우리는 사용자가 단축키로

“R”을 누르게 되면 재생되고 있는 프레임을 정지시키고 정지된 프레임을 기준으로 다시 재 추적하고자 하는 마크(mark)를 재 입력받는다. 그 후 재 입력받은 객체를 기반으로 과거의 추적하고 있는 객체는 추적하지 않고 새로 입력받은 객체를 추적한다.

4. 결론

본 연구에서는 라즈베리 파이와 파이 카메라 모듈을 이용하여 사용자가 추적하길 원하는 객체 정보를 입력받고 해당 정보를 이용하여 CMT 알고리즘 기반 객체 추적 외에도 객체 위치에 따른 촬영 장비가 이동하여야 하는 이동 정보 도출, 추적 객체 변경 기능 등의 기능을 구현하기 위해 CMT 스크립트의 일부를 수정하였다.

촬영 장비의 이동 정보 도출과 가시화를 함께 제공하기 위해 OpenCV 라이브러리에서 선과 원형을 그릴 때 사용되는 arrowedLine, circle 이라는 함수들을 활용하였고 추적 객체 변경 기능을 위해 매 프레임마다 사용자의 입력을 대기하다 키보드의 “R”키에 해당하는 입력을 수신하게 되면 재생되고 있는 Window를 종료하고 종료된 프레임을 기준으로 추적하고자 하는 객체를 입력받을 수 있도록 정지된 프레임이 담겨있는 Window를 다시 제공하고 사용자로부터 추적하고자 하는 객체를 입력받도록 하였다.

이와 같이 우리는 본 연구에서 라즈베리 파이를 [그림 1]과 같은 시스템 구조를 동일하게 표현하기 위해 stand alone 방식으로 구현하고 실험을 통해 검증하였다.

5. 향후 연구

우리는 향후연구로 [그림 1]과 같은 구조를 위해 라즈베리 파이를 stand alone 으로 사용하는 것과 컨트롤러로써 사용하는 방법을 비교하여 더 좋은 성능을 나타내는 구조를 사용하고자 한다.

라즈베리 파이를 stand alone으로 운용하는 경우 일반 컴퓨터보다 컴퓨팅 파워가 부족하여 영상을 처리하고 추적하는 과정은 일반 컴퓨터보다 처리과정이 다소 느린 부분이 존재하지만 네트워크에 의존하지 않아 통신장애와 네트워크로 인한 지연과 관련된 문제들을 염려하지 않아도 되는 장점이 존재한다. 반면 컨트롤러로써 라즈베리 파이를 운용할 경우 객체 추적 과정을 워크스테이션에서 진행함으로써 라즈베리 파이보다 빠른 연산을 할 수 있는 장점이 존재하지만 무선 네트워크 환경을 통해 데이터를 전송하는 과정으로 인해 전체적으로 객체추적 수행 시간이 더 길리게 되는 단점이 존재한다.

라즈베리 파이에서 객체를 추적하고 이동하여야 하는 좌표를 도출하기까지 약 0.22 ~ 0.36 정도의 시간이 소요된 반면 라즈베리 파이에서 촬영한 영상을 데스크톱 PC로 전송하고 객체를 추적하여 이동좌표를 얻기까지 걸리는 시간은 약 0.53 ~ 0.84 초로 stand alone 방식보다 더 많은 소요시간이 발생되었다.

이에 따라, 우리는 각 구조의 비교된 결과에 따라 더

나은 성능을 도출하는 구조를 기반으로 촬영 담당자가 이동하면서 객체 추적을 해야만 하는 경우를 고려하여 라즈베리 파이뿐만이 아닌 모바일 환경에서도 애플리케이션(Application)을 통해 추적할 객체를 지정할 수 있도록 연구를 진행할 계획이다.

6. Acknowledgement

본 연구는 미래창조과학부 연구개발 특구 육성사업의 연구비 지원에 의해 수행되었습니다.

참고문헌

- [1] Nebehay, Georg and Pflugfelder, Roman "Clustering of Static-Adaptive Correspondences for Deformable Object Tracking" Computer Vision and Pattern Recognition
- [2] GitHub, "CMT" "<https://github.com/gnebehay/CMT>"
- [3] Wiki "OpenCV" "<https://ko.wikipedia.org/wiki/OpenCV>"
- [4] OpenCV team, "OpenCV" "<http://opencv.org/>"
- [5] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp.564 - 577, May 2003.
- [6] Kari Pulli, Anatoly Baksheev, Kirill Korniyakov and Victor Eruhimov, "Real-time computer vision with OpenCV", Communications of the ACM, 2012, 55, 61 - 69