메디치소프트 기술연구소

CH45. OpenCV: 비디오에서 객체 추적하기1

MEDICISOFT

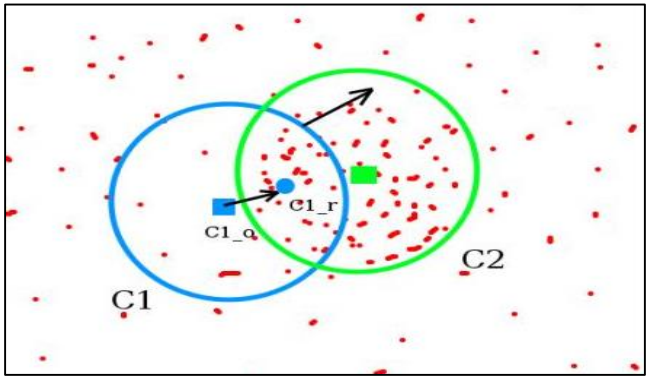
45. 비디오에서 객체 추적하기1

- 45.1 테스트 환경

순 번	제 목	설치 버전
1	운영체제	Windows 10 64bit
2	프로그래밍언어	Python3.6.5

▪ 45.2 Meanshift 알고리즘

- 점들의 집합을 생각해 보았을 때, 점들의 집합은 히스토그램 배경투사와 같이 확률적으로 비슷한 픽셀 분포가 될 수도 있다. 원 모양의 작은 영역을 취하고 이 영역을 픽셀 밀도가 가장 높은 영역으로 이동시킨다. 아래 그림에서 초기에 잡은 원 영역을 C1으로 부르기로 하자. C1의 중심을 C1_o라고 하고 C1에 분포된 점들의 무게 중심을 C1_r 이라고 하면, 아마도 99%는 C1_o와 C1_r은 일치하지 않을 것이다.



[그림] 픽셀 분포에 따른 원 이동

▪ 45.2 Meanshift 알고리즘

- C1_o를 C1_r과 일치되도록 C1을 이동 시킨다. 여기서 다시 새로운 C1_o와 C1_r을 계산한다. 이런 과정을 반복하여 C1_o와 C1_r을 최소차로 일치하게 하면 이 원 영역의 점들의 밀도는 가장 높게 된다.

- 실제로 MeanShift를 적용하려면 히스토그램 배경투사가 된 이미지와 이 이미지에서 객체의 최초 위치를 지정하여 전달해야 한다. 객체가 움직이기 시작하면 히스토그램 배경투사된 이미지에 객체의 움직임이 반영된다. Meanshift는 객체 영역의 픽셀 밀도와 가장 일치하는 영역을 찾아주게 되어 최초 지정된 객체 영역을 추적할 수 있게 해준다.

▪ 45.2 Meanshift 알고리즘

- OpenCV에서 Meanshift는 다음과 같은 절차로 구현한다.

1) 타겟을 설정한다.
2) 설정된 타겟의 히스토그램을 얻는다. 이는 Meanshift 계산을 위해 비디오의 각 프레임에서 타겟의 배경투사를 할 수 있도록 한다.
3) 초기 타겟 영역의 위치를 지정한다.
4) 히스토그램은 색공간에서 Hue(색상)만 고려한다.

45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
import numpy as np
import cv2


col, width, row, height = -1, -1, -1, -1
frame = None
frame2 = None
inputmode = False
rectangle = False
trackWindow = None
roi_hist = None
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
def onMouse(event, x, y, flags, param):
    global col, width, row, height, frame, frame2, inputmode
    global rectangle, roi_hist, trackWindow

    if inputmode:
        if event == cv2.EVENT_LBUTTONDOWN:
            rectangle = True
            col, row = x,y

        elif event == cv2.EVENT_MOUSEMOVE:
            if rectangle:
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
            frame = frame2.copy()
            cv2.rectangle(frame, (col, row), (x,y), (0,255,0), 2)
            cv2.imshow('frame', frame)


        elif event == cv2.EVENT_LBUTTONUP:
            inputmode = False
            rectangle = False
            cv2.rectangle(frame, (col, row), (x,y), (0,255,0), 2)
            height, width = abs(row-y), abs(col-x)
            trackWindow = (col, row, width, height)
            roi = frame[row:row+height, col:col+width]
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
            roi = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
            roi_hist = cv2.calcHist([roi], [0], None, [180], [0, 180])
            cv2.normalize(roi_hist, roi_hist, 0, 255, cv2.NORM_MINMAX)
        return
def meanShift():
    global frame, frame2, inputmode, trackWindow, roi_hist


    try:
        cap = cv2.VideoCapture(0)
    except Exception as e:
        print(e)
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
    return

ret, frame = cap.read()

cv2.namedWindow('frame')
cv2.setMouseCallback('frame', onMouse, param=(frame, frame2))

termination = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)

while True:
    ret, frame = cap.read()
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
if not ret:
    break


if trackWindow is not None:
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    dst = cv2.calcBackProject([hsv], [0], roi_hist, [0, 180], 1)
    ret, trackWindow = cv2.meanShift(dst, trackWindow, termination)


    x,y,w,h = trackWindow
    cv2.rectangle(frame, (x,y), (x+w, y+h), (0,255,0),2)
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 code이다.

```
cv2.imshow('frame', frame)


k = cv2.waitKey(60) & 0xFF


if k == 27:
    break
if k == ord('i'):
    print('Select Area for CamShift and Enter a key')
    inputmode = True
    frame2 = frame.copy()
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

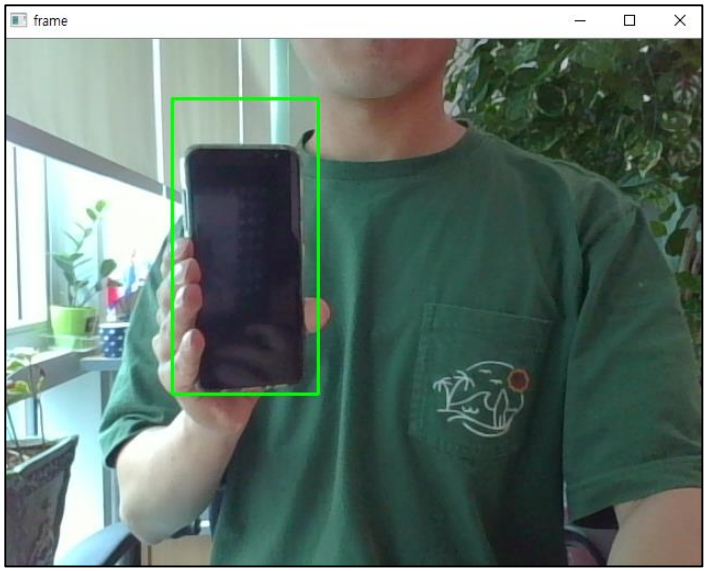
- 다음은 45-01 example.py 예제의 code이다.

```
        while inputmode:
            cv2.imshow('frame', frame)
            cv2.waitKey(0)
    cap.release()
    cv2.destroyAllWindows()

meanShift()
```

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

- 다음은 45-01 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 45-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 45-01 example 실행 결과

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

>> LINE 12) def onMouse(event, x, y, flags, param):

>> LINE 17) event == cv2.EVENT_LBUTTONDOWN:

>> LINE 21) elif event == cv2.EVENT_MOUSEMOVE:

>> LINE 27) elif event == cv2.EVENT_LBUTTONUP::

- meanShift() 함수에서 키보드 'i'가 입력되면 마우스로 추적할 객체를 지정할 수 있도록 코드를 구현한다. 마우스왼쪽이 클릭되었을 때, 마우스가 왼쪽 버튼을 누른채 움직일 때, 마우스 왼쪽 버튼을 땠을 때 처리할 루틴을 작성한다.
- LINE 27) 에서 추적할 객체 영역을 계산한다. 또한 ROI를 tracking window를 이용하여 설정하다. ROI를 HSV 색공간으로 변경하며 HSV 색공간으로 변경한 히스토그램을 계산하고 계산된 히스토그램을 노멀라이즈 한다.

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

>> LINE 39) def meanShift():

>> LINE 43) cap = cv2.VideoCapture(0)

>> LINE 48) ret, frame = cap.read()

>> LINE 50) cv2.namedWindow('frame')

>> LINE 51) cv2.setMouseCallback('frame', onMouse, param=(frame, frame2))

- 비디오 장치를 통해 입력되는 영상 객체를 생성한다.

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

>> LINE 53) termination = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)

- meanShift의 3번째 인자를 정의한다. meanShift의 iteration 회수를 10 또는 C1_o 와 C1_r의 차가 1pt 일 때까지 알고리즘을 구동하라는 의미이다.

>> LINE 62) dst = cv2.calcBackProject([hsv], [0], roi_hist, [0, 180], 1)

- roi_hist는 normalized ROI 히스토그램이다.

>> LINE 66) cv2.rectangle(frame, (x,y), (x+w, y+h), (0,255,0),2)

- 추적된 물체를 녹색 사각형으로 표시한다.

▪ 45.3 [45-01 example.py] – meanShift를 이용한 객체추적

>> LINE 74~77)

```
if k == ord('i'):
        print('Select Area for CamShift and Enter a key')
        inputmode = True
        frame2 = frame.copy()
```

- 'i'키를 눌렀을 때 onMouse에 구현된 로직이 활성화 되도록 한다. 현재 화면을 키보드를 누를 때 까지 일시 멈춘다.