
Implementing one of difference-keying algorithms

- PC 카메라 영상의 배경을 임의의 다른 배경으로 교체 -

- ▷ 이름: 손 수 지
- ▷ 학번: 202011933
- ▷ 학과: 전자정보통신공학부 전자공학 전공
- ▷ 과목(분반): 객체지향프로그래밍 (108)
- ▷ 담당교수: 박 한 훈 교수님
- ▷ 제출일: 2022.06.18

- 목 차 -

| | |
|---------------------------|----|
| I. 서론 | 3 |
| 1. 과제 수행 배경(동기)과 목적 | 3 |
| 2. 과제 수행 방향 및 방법 | |
| II. 본론(구현) | 4 |
| 1. 구현 | 4 |
| 2. 전체 코드 | 6 |
| III. 결과 및 분석 | 7 |
| 1. 실험(시뮬레이션) 환경 | 7 |
| 2. 실험 결과 및 분석 | 7 |
| 3. 한계 및 개선 방향 | 8 |
| IV. 결론 | 9 |
| 1. 과제 수행 결과 요약 | 9 |
| 2. 과제 수행을 통해 느낀 점 | 9 |
| V. 참고문헌 | 10 |

I. 서론

1. 과제 동기 및 목적

PC 카메라 영상의 배경을 임의의 다른 배경으로 교체한 것을 ‘가상 배경’이라고 할 수 있습니다. 제가 COVID-19로 인해 사용하게 된 zoom과 같은 서비스에서 배경을 흐리게 하거나, 변경하는 기법도 가상 배경을 사용한 것임을 알게 되었습니다.

객체지향프로그래밍의 첫 번째 과제 ‘PC 카메라 영상의 배경을 임의의 다른 배경으로 교체해보기’를 수업시간에 배운 Python과 OpenCV를 활용하여 구현해보고자 하였습니다.

2. 과제 수행 방향 및 방법

이 과제의 목표는 PC 카메라 영상의 배경을 임의의 다른 배경으로 교체할 수 있도록 프로그래밍 하는 것입니다. 간단하게 두가지 큰 틀로 정리해보겠습니다. 먼저 캠을 키고 입력되는 영상에서 전경(사람=나)과 배경으로 나눠서 볼 수 있습니다. 여기서 배경을 지운 전경만 남겨둔 영상을 준비합니다. 다음단계는 준비해 둔 영상을 입력하고 전경을 캡처하여 전경 영상을 넣는것입니다. 이때, 입력된 전경의 영상 사이즈에 바꿔줄 가상환경 영상의 사이즈를 맞춰주는 과정도 고려합니다. 이런 방식으로 opencv에서 제공한 소스를 활용해 프로그래밍을 진행합니다.

II. 본론(구현)

1. 구현

1) 필요한 라이브러리 가져오기

```
import cv2
import numpy as np
import sys
```

2) resize 함수 정의

전경과 배경에 맞게 크기 조정

```
def resize(dst,img):
    width = img.shape[1] #너비
    height = img.shape[0] #높이
    dim = (width, height) #튜플로 묶기
    resized = cv2.resize(dst, dim, interpolation = cv2.INTER_AREA)
    return resized #resized 반환
```

3) camera를 켜고, background로 사용할 video 가져오기

카메라 켜기

```
video = cv2.VideoCapture(0)
```

opencv를 사용하여 background로 사용할 비디오 파일 열기

```
backgroundVideo = cv2.VideoCapture("background.mp4")
```

카메라가 배경을 캡처하고, 참조로 저장하도록 하기

```
success, ref_img = video.read()
```

flag 초기화 (flag : 프로그램을 수동으로 제어하고 배경을 다음 줄에서
이해할 수 있는 이미지 또는 비디오로 대체하는 데 사용)

```
flag = 0
```

4) While 루프

비디오 카메라에서 입력을 읽을 때까지 while 루프 실행

```
while(1):
    success, img = video.read()
    success2, bg = backgroundVideo.read()
    bg = resize(bg,ref_img)
    if flag==0:
        ref_img = img
# mask 만들기
    diff1=cv2.subtract(img,ref_img)
    diff2=cv2.subtract(ref_img,img)
    diff = diff1+diff2
```

```

        diff[abs(diff)<10.0]=0    #임계값 만들기
# opencv 내장함수 cvtColor를 사용하여 추출 된 이미지를 회색조로 변환
        gray = cv2.cvtColor(diff.astype(np.uint8), cv2.COLOR_BGR2GRAY)
#회색의 절댓값을 0으로 변환
        gray[np.abs(gray) < 10] = 0
# 다시 흑백으로 변환
        fgmask = gray.astype(np.uint8)
        fgmask[fgmask>0]=255
# mask 반전시키기
        fgmask_inv = cv2.bitwise_not(fgmask)
# mask를 사용하여 ForeGround에서 관련 부분 추출
        fgimg = cv2.bitwise_and(img,img,mask = fgmask)
# mask를 사용하여 BackGround에서 관련 부분 추출
        bgimg = cv2.bitwise_and(bg,bg,mask =~ fgmask)
# BackGround images와 ForeGround images 결합
        dst = cv2.add(bgimg,fgimg)
#cv2.imshow를 사용하여 image 표시
        cv2.imshow('Background Removal',dst)

        key = cv2.waitKey(5) & 0xFF
        if ord('q') == key:    #“q”키를 눌러 종료
            print("End")
            break
        elif ord('d') == key:    #“d”키를 눌러 가상 배경을 만들고
            flag = 1
            print("Background Captured")
        elif ord('r') == key:    #“r”키를 눌러 배경 비디오를 표시
            flag = 0
            print("Ready to Capture new Background")
cv2.destroyAllWindows()
video.release()

```

2. 전체 코드

```
backgroundRemoval_video.py x README.md x

import cv2
import numpy as np
import sys

def resize(dst,img):
    width = img.shape[1]
    height = img.shape[0]
    dim = (width, height)
    resized = cv2.resize(dst, dim, interpolation = cv2.INTER_AREA)
    return resized

video = cv2.VideoCapture(0)
oceanVideo = cv2.VideoCapture("background.mp4")
success, ref_img = video.read()
flag = 0

while(1):
    success, img = video.read()
    success2, bg = oceanVideo.read()
    bg = resize(bg,ref_img)
    if flag==0:
        ref_img = img
    # create a mask
    diff1=cv2.subtract(img,ref_img)
    diff2=cv2.subtract(ref_img,img)
    diff = diff1+diff2
    diff[abs(diff)<10.0]=0
    gray = cv2.cvtColor(diff.astype(np.uint8), cv2.COLOR_BGR2GRAY)
    gray[np.abs(gray) < 10] = 0
    fgmask = gray.astype(np.uint8)
    fgmask[fgmask>0]=255
    #invert the mask
    #fgmask_inv = cv2.bitwise_not(fgmask)
    #use the masks to extract the relevant parts from FG and BG
    fgimg = cv2.bitwise_and(img,img,mask = fgmask)
    bgimg = cv2.bitwise_and(bg,bg,mask = ~ fgmask)
    #combine both the BG and the FG images
    dst = cv2.add(bgimg,fgimg)
    cv2.imshow('Background Removal',dst)
    key = cv2.waitKey(5) & 0xFF
    if ord('q') == key:
        print("End")
        break
    elif ord('d') == key:
        flag = 1
        print("Background Captured")
    elif ord('r') == key:
        flag = 0
        print("Ready to Capture new Background")

cv2.destroyAllWindows()
video.release()
#return jpeg.tobytes()
```

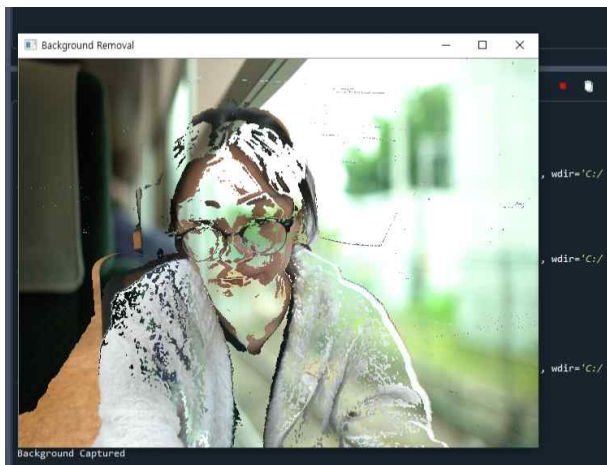
Ⅲ. 결과 및 분석

Python 3.6

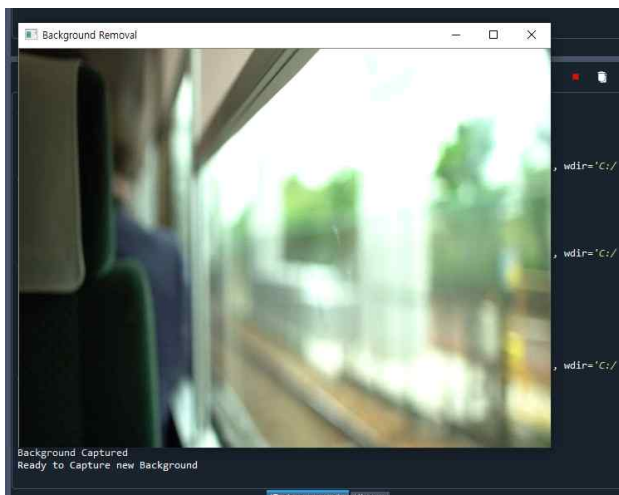
spyder

프로그램을 실행시키면 입력해둔 background 영상이 켜집니다.

“d”키를 누르면 가상 배경이 만들어지고, 입력해둔 문구 “Background Captured”가 출력됩니다.



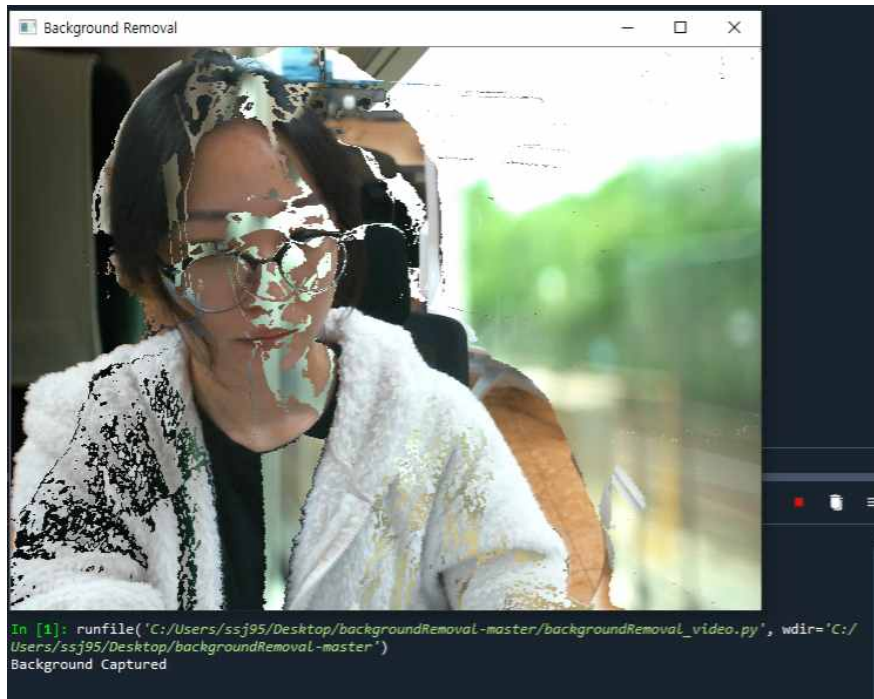
“r”키를 누르면 다시 입력해둔 background 영상이 켜지고, “Ready to Capture new Background” 문구가 출력됩니다.



마지막으로 “q”키를 누르면 창이 종료되고, “End”문구가 출력 되는 것을 확인할 수 있었습니다.

```
In [1]: runfile('C:/Users/ssj95/Desktop/backgroundRemoval-master/backgroundRemoval_video.py', wdir='C:/Users/ssj95/Desktop/backgroundRemoval-master')
Background Captured
Ready to Capture new Background
End
```

제가 구성한 코드 대로 프로그램이 실행되는 것을 확인할 수 있었습니다. 하지만 첫 번째 결과 이미지를 보면, 제가 예상한 가상배경처럼 완벽하고 깔끔하지 못한 결과를 나타내는 것을 확인할 수 있습니다.



또한, “d” 키를 누른 뒤 옆으로 이동하면 뒤의 배경이 보이는 공백이 생성된 것을 확인할 수 있었습니다. 이는 capture된 background에서 제 모습(전경 이미지)가 움직였기 때문에 제가 있던곳 즉, capture되지 않았던 위치가 노출되며 그 부분은 공백으로 남게 된 것입니다.

IV. 결론

1. 요약

입력시켜둔 영상에 캠에 입력되는 영상의 배경을 제거하고 남은 전경(제모습)을 넣는 과정을 프로그래밍 후 실험을 실시 하였습니다.

프로그래밍 해둔대로 잘 진행이 되었지만 결과적으로 깔끔하지 못한 결과를 보이는 것을 볼 수 있었습니다.

2. 느낀점

구글과 깃허브를 통해 소스코드와 배경 제거와 영상 저장, 영상 크기 조절 함수등을 직접 프로그래밍 해보았습니다. 부족한 부분이 많았고, 실험 결과가 매우 좋지는 않았지만, 수업시간에 배운 파이썬으로 직접 설계를 해 보았다는 점에서 좋은 경험이 되었습니다.

V. 참고문헌 & 사이트

- [1] <http://www.gisdeveloper.co.kr/?p=6862> #background subtraction
- [2] <https://zzsza.github.io/data/2018/01/23/opencv-1/> #영상 저장, 읽기