영상속의 그림판

1. 주제 선정의 이유

내가 주변에서 다룰 수 있는 영상처리 기능을 고민하던 중, 요즘 나오는 카메라 기능에 촬영하고 있는 동영상 위에 그림을 그리거나 스티커를 위에 붙이는 것을 보며 해보고 싶다는 생각이 들었다.

수업 중 배웠던 것 중 가장 흥미로웠던 hsv나 역투영을 활용하여 원하는 색상을 영상에서 뽑아낼 수 있는 것을 활용하고 싶었다.

이후, 영상의 후처리 과정에서, hsv의 s,v를 따로 조절하여 색상을 찾아내는 효율을 높이고, 컨투어와 기하학적변환을 이용하여 물체인식을 높이는 등 후처리 관련 기능도 사용 가능하고.

위로 얻어 낸 마스크 영상을 이용 물체의 중심점의 이동 좌표를 활용하여 선을 그리면 그림을 그리는 효과를 나타낼 수 있다고 생각하였다.

2. 주제요약 및 목표

1. 영상속에서 원하는 색 만을 뽑아낼 수 있는가.
2. 색의 컬러모델을 활용하여 색상을 더 잘 찾아낼 수 있는가.
3. 인식률 향상을 위한 후처리를 활용할 수 있는가.
4. 해당 색의 영역 중 가장 큰 넓이를 가진 영역을 찾아낼 수 있는가.
5. 원하는 영역의 중심 값을 얻은 수 있는가.
6. 해당 좌표를 이용하여 영상위에 그림을 그릴 수 있는가.

3. 개발 내용

1. 영상속에서 원하는 색상을 뽑아내기.

가장 먼저 화면의 일정 영역을 색상을 얻어내는 구역으로 설정을 한다

이후, space bar가 눌리면 해당영역안의 색상 모델을 hsv로 바꾸어 준다.

원하는 색상 값의 근사값을 얻기 위해 h의 평균값으로 기준 값을 설정한다.

H의 평균값을 기준으로 +-20의 범위의 색상을 inRange 함수로 설정을 한다.

* 이때에 h의 평균값이 0~180의 범위임으로 +-계산중 오버 혹은 음수가 되지 않도록 3개의 구간으로 나누어 계산을 해주어야 한다.
* H.avr + 20 이 180이 넘을 경우 h.avr~180 , 0~h.avr+20-180, h.avr-20~h.avr
* H.avr -20이 0보다 작을 경우 h.avr-20+180~180, 0~h.avr, h.avr~h.avr+20
* 그 외 h.avr-20~h.avr+20

1. 색의 컬러모델을 활용하여 색상 검출의 효율을 높이기.

이후 위의 hsv색의 모델 값을 이용하여 색상을 검출하면, 조명등의 차이로 인해 색상 검출이 잘 되지 않을 수 도 있다.

* 위를 해결하고 자 트랙바를 이용하여 h,v값을 조절 할 수 있도록 한다.

위를 이용하면 내가 원하는 색상의 위치만 하얀색으로 마스크 이미지를 만들 수 있다.

1. 인식률 향상을 위한 후처리

여기선 총 2가지의 후처리 기술을 이용하였다.

* 첫째로 morphology
* Open과 close를 반복 실행 시켜 물체안의 검은 점을 없애고, 물체 밖의 하얀색 점들을 제거해준다.
* 둘째로 돌아가거나 삐뚤어진 영역을 사각형의 형태로 바꾸어 주는 기하학적 변환이다.

우선 위 후처리를 위해선 원하는 물체의 영역의 테두리 부분이 필요하다.

위를 얻기 위해 우선 Canny함수를 이용하여 엣지 검출을 해주고, 또 이를 이용하여 컨투어 함수를 이용하여 윤곽선 부분만 얻어낸다.

이후, 내가 얻어내고자 하는 부분을 영역의 가장 큰 부분이라고 가정하고, 컨투어 영역의 크기 중 가장 큰 곳을 찾아낸다.

이후 해당영역에서 4개의 점을 기준점으로 두는데, 이 점은 각각

왼쪽 위 -> x,y 두 좌표의 합이 가장 작은 점

오른쪽 위 -> x,y 두 좌표의 차가 제일 작은 점

오른쪽 밑 -> x,y 두 좌표의 차가 제일 큰 점

왼쪽 밑 -> x,y 두 좌표의합이 제일 큰 점 이다.

이후 변경할 가로 세로길이는 두 변의 길이 중 큰 변을 기준으로 잡아준다.

마지막으로 필요한 변경 후 좌표를 구해야 하는데

[topleft,

[topleft[0]+widthc-1, topleft[1]]

,[topleft[0], topleft[1]+heightc-1]

,[topleft[0]+widthc-1, heightc-1+topleft[1]]]

로 변경점을 지정한다.

* 이후 warpPerspective함수를 이용하여 기하학적 변환을 하면 돌아가거나 위아래로 눌리지 않은 사각형 모양의 영역 마스크 영상이 생성이 된다.

1. 다음으로 그림을 그릴 기준점을 얻기 위해선 위 작업으로 얻어온 영역의 정보들을 활용하여야 한다.

이때에 connectedComponentsWithStats함수를 이용하는데 이는 index, 라벨링 값, 중심 값 등을 반환해 준다.

이중 영역의 크기 중 가장 큰 (위 컨투어 에서 한번 걸러주었지만 한번 더 걸러 줌) 영역을 사용하는 영역으로 설정하고 해당 index값의 영역의 중심 좌표, 왼쪽 위 좌표, 가로 세로변의 길이를 얻어온다.

* 위 정보들을 이용하여 기본 영상위에 가시성을 위해 영역표시와, 중심좌표를 표시하여 준다.
* 왼쪽 위 좌표와 가로 세로길이로는 해당 물체의 후처리 이후의 영역 표시.
* 중심 좌표는 그림이 그려지는 연필심 부분의 영역 표시
* 전체 지우기는 저장된 배열들을 전부 초기화 해줌으로 활용.

1. 해당 기준점을 이용하여 그림 그리기.

* 이후, r 버튼이 눌려 그림 그리기 상태가 True가 되면, 중심 좌표 2개를 이용하여 위에서 받아온 h값의 평균값을 가진 색의 선을 line함수를 이용하여 그린다.

그림을 그리기, 멈추기, 색상 바꾸기 기능 활용을 위하여 좌표저장 배열을 두개로 구분해준다

현재 그리고 있는 부분, 과거에 그려진 부분, 이 두가지를 모두 draw해주면 된다.

* 두 배열은 전부, 기준 좌표의 x, y좌표와 당시의 색상 값을 저장한다.

마지막으로 스티커를 만들 때 에 필요한 과정 두가지이다

1. 스티커를 그리는 부분이다.

m키를 누르면 스티커를 만들 수 있는 영역이 나온다.

영상위에 그림을 그리는 부분과 다른 점을 두기 위하여 차이점 두가지를 주었다.

1. 마우스 콜백함수를 활용하여 마우스의 상태, 좌표를 이용하여

해당구역의 색상을 얻어 오기, 그림을 그리기

* 2. 내가 그린 부분을 지우기
  + 1. 이후 스티커 후처리 과정에서 나오지만 검은색 부분을 처리하지 않는 부분으로 지정하기 때문에 지우개를 좀 큰 검은색 원을 그리는 행위로 지우기를 실행.

이 후 그림을 그리는 부분에 관한 알고리즘은 위와 동일하다.

* 마지막으로 스티커는 그린 뒤 트렉바와 warpAffine 를 활용하여 크기, 회전 을 결정하여 준다.

1. 만들어진 스티커를 원본 이미지에 붙이기

마지막으로 완성된 스티커를 이미지에 붙이기 위해선 배경부분을 지워 주어야한다.

이를 위해 스티커 그림판의 기존 이미지와 현재 그린 스티커 그림판의 다른 부분만을 받아온다. (즉 그려진 부분만을 받아 옴 absdiff함수 이용)

* 이후 해당 영역을 grayscale로 바꾸어 준 뒤, Threshold를 활용하여 마스크 이미지를 생성하여 준다. 이는 배경만 지워진 이미지.

위 이미지를 마우스 콜백 함수를 활용하여 s를 누른뒤 원하는 지점에 클릭을 하면

스티커가 원본 영상위에 그려지게 된다.

실행 결과

https://www.youtube.com/watch?v=xV7hKFyit30

사용방법

원본 화면에서 space bar를 눌러 빨간 사각형 안에 색을 인식 시켜줌

이후 img\_result 에서 트랙바를 활용 색을 확실하게 인식시켜줌

R 버튼을 눌러서 물체의 중심점을 활용하여, 그림을 그리거나 일시 멈춤

다른 색을 찾고 싶으면 spacebar를 눌러 색찾기 모드로 바꾸어줌

m키를 누르면 스티커 만들기 화면으로 넘어감

클릭으로 원하는 색을 찾은 뒤, 드래그로 스티커를 그림

e버튼을 누르면 지우개 모드로 바뀌고 드래그 하면 지워짐

이후 트랙바로 크기나, 각도를 조정해줌

Esc를 누른뒤 img\_draw 윈도우를 클릭후 s버튼을 누르면 스티커 붙이기 모드가 된다

이후 원하는 영역에 클릭을 하면 스티커가 붙는다.

스티커는 x를 누르면 전부 지워진다.

향후 계획

1. 각종 필터효과, 블러 , 모자이크 , 색변환
2. 좀 더 깔끔한 물체인식 요망
3. 얼굴 인식을 통한 얼굴을 따라다니는 스티커 구현

총 목표를 말하자면, 좀더 포토샵에 가까운 기능들을 구현해 보고싶다.

\*\*

색 모델 값 들을 활용한 인식능력

기하학적 변환을 통한 인식능력

후처리 과정을 통환 지우기 기능

개발 후기

목표가 배운 것에 기반으로, 였기에 얼굴 인식이나 글자 인식 등 색다른 기능들을 써보지 못한 것이 조금은 아쉬움으로 남는다. 그래서 인지 이후 추가적인 기능들을 넣고 싶은 마음이 더욱 크다. 또한 중간 보고서 피드백을 통해 게임기능을 포기, 인식률 향상이라는 중간 단계를 추가 하였는데, 어떻게 뭘 해야 하지? 라는 생각으로 전공서적만 열 번 이상 다시 정독 한 것 같다. 덕분에 활용한 기능들과 더불어, 앞에서 배운 내용들에 관하여 더욱 확실히 알게 되었다.

이 프로젝트 이후 물체 추적을 기반으로 하는 다양한 프로젝트에 더욱 자신감을 가지고 임할 수 있을 것이다.