Digital Image Processing

LabWork #CV5

Professor : Jin-Woo Jung

* Using the given CImageAnalysis class and additional reference material, Design a program that can perform the followings:

1. Using USB camera and a template images for template matching
   1. Capture the video from USB camera using cv::VideoCapture to access the camera device and cv::Mat image to get the camera image

|  |  |
| --- | --- |
| **Input** | **Output** |
| (Camera Image) |  |
| code | |
| #include <opencv2/opencv.hpp>  using namespace cv;  int main()  {  Mat image; // 웹캠으로 캡쳐한 이미지를 저장하기 위한 변수를 선언한다.  VideoCapture cap; //비디오 캡처를 위한 클래스의 인스턴스를 선언한다.  cap.open(0); //open 함수를 통해 카메라를 불러오는 작업을 수행한다.  //파라미터로 index가 올 수 있다.  // index는 이미지를 캡처하는데 사용할 카메라를 지정하는 파라미터로, 장치관리자에 등록되어 있는 카메라 순서대로 index가 설정되어 있다.  // 기본적으로 index는 0으로 시작된다. (0 = 기본 카메라)  //또한, 이 함수는 open(const String & filename, int apiPreference)과 오버로딩 되어있다.  //filename 파라미터는 자신이 불러올 비디오, 이미지 파일의 파일명을 전달한다.  //apiPreference 파라미터는 선호하는 capture api를 지정하는 파라미터다. CAP\_FFMPEG, CAP\_IMAGES 등등이 올 수 있다.  cap.read(image); //카메라를 통해 얻은 이미지를 파라미터로 들어온 변수에 저장한다.  imshow("image", image); // 웹캠을 통해 캡쳐한 image를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | |

* 1. Template matching using GS\_templateMatching

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |
| code | |
| #include <opencv2/opencv.hpp>  #include "ImageAnalysis.h"  using namespace cv;  int main()  {  Mat image; // 웹캠으로 캡쳐한 이미지를 저장하기 위한 변수를 선언한다.  Mat temp = imread("template.png", IMREAD\_COLOR); // template.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  CImageAnalysis analysis; //이미지 분석을 위한 클래스의 인스턴스를 선언한다.  VideoCapture cap; //비디오 캡처를 위한 클래스의 인스턴스를 선언한다.  cap.open(0); //open 함수를 통해 카메라를 불러오는 작업을 수행한다.  //파라미터로 index가 올 수 있다.  // index는 이미지를 캡처하는데 사용할 카메라를 지정하는 파라미터로, 장치관리자에 등록되어 있는 카메라 순서대로 index가 설정되어 있다.  // 기본적으로 index는 0으로 시작된다. (0 = 기본 카메라)    //또한, 이 함수는 open(const String & filename, int apiPreference)과 오버로딩 되어있다.  // filename 파라미터는 자신이 불러올 비디오, 이미지 파일의 파일명을 전달한다..  //apiPreference 파라미터는 선호하는 capture api를 지정하는 파라미터다. CAP\_FFMPEG, CAP\_IMAGES 등등이 올 수 있다.  cap.read(image); //카메라를 통해 얻은 이미지를 파라미터로 들어온 변수에 저장한다.  Mat template\_matching = analysis.GS\_templateMatching(image, temp); //웹캠으로 불러운 이미지를 템플릿 이미지를 이용하여 template matching을 수행한다.  imshow("image", template\_matching); // template matching을 완료하여 템플릿 이미지가 존재하는 위치가 표현된 image를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | |