HW8 Report

Computer Vision
Section 02
21800181 Kim Jisu

HW 8

- Purpose

This program is that read “background.mp4” and set background image by average of the first 10 frames. find moving object draw bounding rectangle on the original image and print out the number of moving object (whose size is bigger than 400 pixels).

- Principle

void cv::absdiff (InputArray src1, InputArray src2, OutputArray dst)

Calculates the per-element absolute difference between two arrays or between an array and a scalar.

void cv::findContours (InputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset = Point())

Finds contours in a binary image.

- Process

1. background image

set background image by average of the first 10 frames.

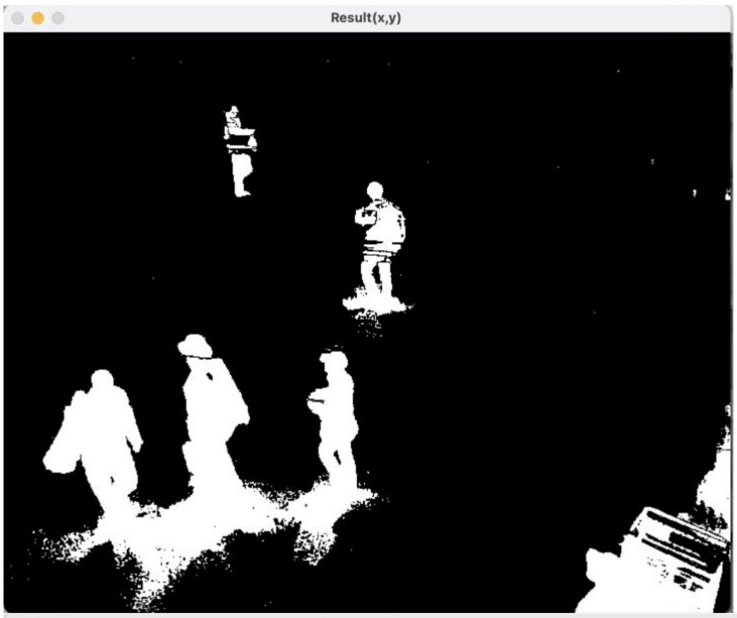
2. Generate a binary image by using the following equation

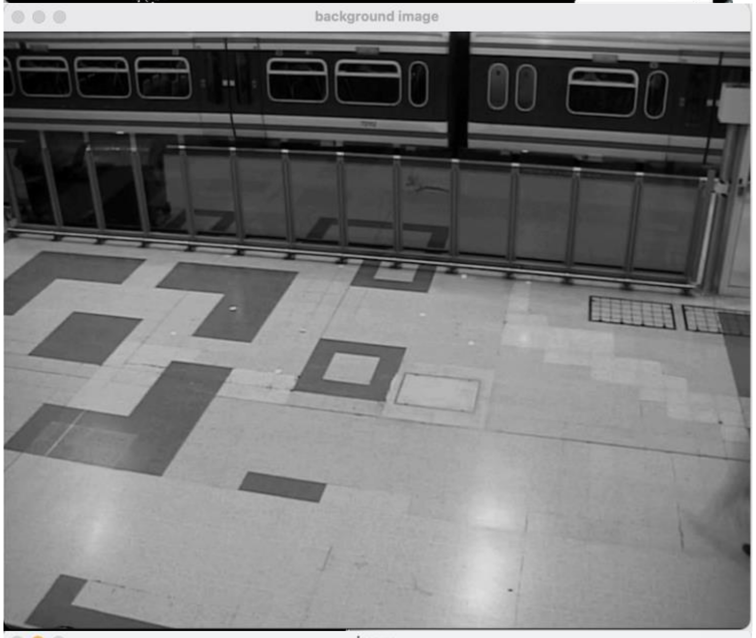
$$\text{Result}(x,y) = \begin{cases} 255 & |\text{Current frame}(x,y) - \text{Background}(x,y)| > 20 \\ 0 & \text{otherwise} \end{cases}$$

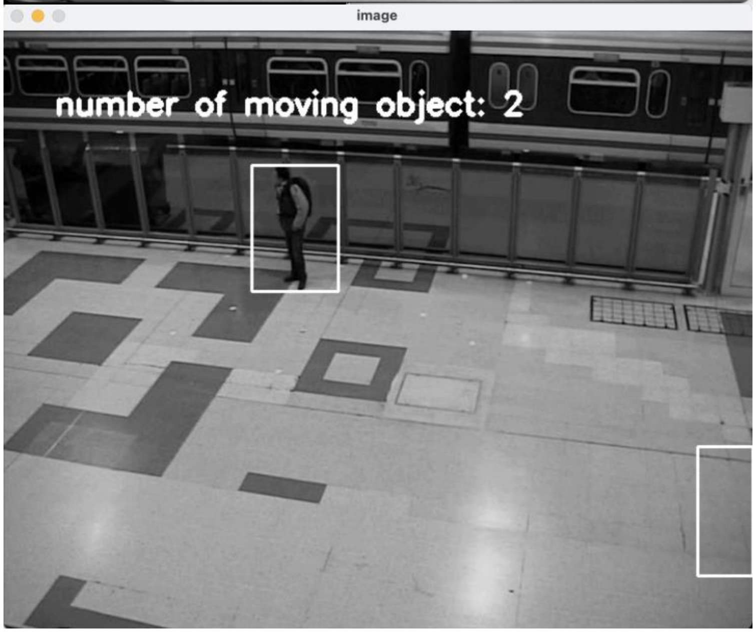
3. Draw bounding rectangle on the original input video each moving object whose size is bigger than 400 pixels.

4. Print out the number of moving objects on the image whose size is bigger than 400 pixels.

- Result







number of moving object: 2

- Code

```cpp
#include "opencv2/opencv.hpp"
#include <iostream>


using namespace std;
using namespace cv;


int main(){
    VideoCapture cap("background.mp4");
    Mat sum, avg,img;
    Mat frame;
    int cnt = 2;
    cap >> avg;
    int current_frame = 0;
    while (true) {
        if(!cap.read(img)) break;
        if (current_frame>=10) break;
        add(img/cnt,avg*(cnt-1)/cnt,avg);
        cnt++;
        current_frame++;
    }//background
    frame = avg.clone();
    cvtColor(frame, frame, CV_BGR2GRAY);

    Mat background, image, gray, result, foregroundMask, foregroundImg;
    VideoCapture capture("background.mp4");
    background = frame;
    Mat element = getStructuringElement(MORPH_ELLIPSE, Size(2, 2));
    while (true) {
        if (capture.grab() == 0) break;
        capture.retrieve(image);
        cvtColor(image, gray, CV_BGR2GRAY);
        absdiff(background, gray, foregroundMask);
        threshold(foregroundMask, foregroundMask, 20, 255, CV_THRESH_BINARY);
        //erode(foregroundMask, foregroundMask, element);
        //dilate(foregroundMask, foregroundMask, element);
        foregroundMask.copyTo(foregroundImg);
        gray.copyTo(foregroundImg, foregroundMask);

        vector<vector<Point>> contours;
        vector<Vec4i>hierarchy;
        findContours(foregroundMask, contours, hierarchy, CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_SIMPLE);

        vector<Rect> boundRect(contours.size());
        for (int i = 0; i < contours.size(); i++)
            boundRect[i] = boundingRect(Mat(contours[i]));
        //draw rectangles on the contours
        int count_object = 0;
        for (int i = 0; i < contours.size(); i++){
            if(abs(boundRect[i].tl().x-boundRect[i].br().x)*abs(boundRect[i].tl().y-boundRect[i].br().y)>400){
                rectangle(image, boundRect[i].tl(), boundRect[i].br(), Scalar(255, 255, 255), 2, 8, 0);
                count_object++;
            }
        }
        cvtColor(image, image, CV_BGR2GRAY);
        putText(image, format("number of moving object: %d",count_object ), Point(50, 80),
FONT_HERSHEY_SIMPLEX, 1, Scalar(255, 255, 0), 4);
        imshow("image", image);
        imshow("Result(x,y)", foregroundMask);
        imshow("background image", background);
        waitKey(33);
    }

    waitKey(0);
}
```

I References

https://docs.opencv.org/master/