HIGHER EDUCATIONAL INSTITUTE OF ENGINEERING OF LISBON

Computer Science and Multimedia Engineering department

Digital Image Processing

2nd Laboratory Project – Motion Detection for Object Tracking

Yeldos Adetbekov

Lisbon, January 2018

ABSTRACT

This project was an attempt at developing a motion detection using modern computer vision technology. The algorithm uses adaptive methods to segment image to identify objects. Incidentally, objects mean moving people and moving cars. Main objectives are to develop a computer vision algorithm able to detect, classify and track objects that appear in a video sequence typical from surveillance environment and segment foreground from background considering moving objects. In this project work, operations on segmentation of objects are presented step wisely. The main thesis of all work is thresholding and consecutive background subtraction algorithms which are included in standard OpenCV library's functionality.

1 INTRODUCTION

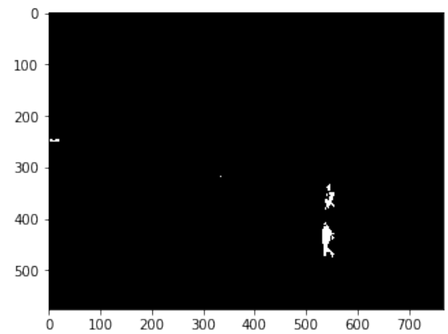
In this laboratory project it was developed an algorithm to classify moving objects and display their identifiers, where we consider camera movement. This algorithm integrated with an automatic surveillance system and capable to detect active regions (areas of movements) in a sequence of images. The objects of interest are people and cars. There are 3 active region classifications, which are PEOPLE(P), CARS(C) and OTHERS. The detected regions marked with a colored bounding box according to the results of the classification procedure, where PEOPLE marked as green, CAR – blue, OTHER – red and associated with an identifier (number), corresponding to the detected object.

To achive main objectives, it is obligatory to evaluate with a test set of videos, where algorithm should return stable outputs. This report details the development and implementation of an algorithm that can be used for this purpose. The report first analyses the problem and establishes the requirements for the algorithm. The results of testing the algorithm are discussed before the report concludes with a summary.

2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

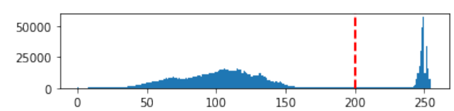
2.1 Background subtraction

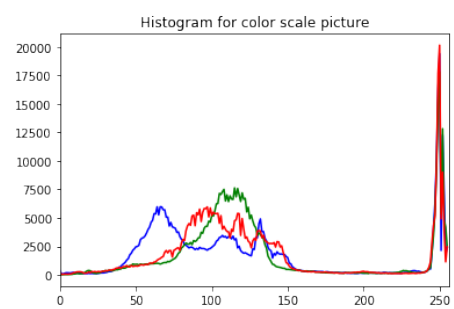
There are so many different methods to subtract background from foreground, but all of them has own predestination. For example in our case the image is not stable and it has a multiple image sequences. So that we should make some operations on active regions detection. I used standard openCV's algorithm which called cv2.createBackgroundSubtractorMOG2(), which updates background by itself. In the outcome of this algorithm we have image with three intensity. Zero(black) is background, 0.5(gray) is shadows and One(white) is foreground. But there some vulnerability in this method. The problem appears if the object stays on 1 point without any action. The object is going to be merged with background. But in our case delay of objects are not so critical.



2.2 Thresholding image

Because after background subtraction operation we have only 3 level depth image (intensity), and we know that the interested object is pretty white, so that we can take as threshold as 200-255. And we need to keep in mind that adaptive binary OTSU algorithm is not valid to the sequence of image, because in every time calculates new threshold and classification always jumps.





2.3 Morphological Transformation

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion.

I have to say that order what we apply erosion and dilation operations is important. And there are two different method as closing and opening. It will be very useful in our case.

First, used an opening mathematical transformation to minimize objects countours, due to we have connected (coinsided) elements. Here we have to apply opening with structuring element with (2, 2) matrix and make this operation once, which is can be setted in the iteration attribute. After opening opperation we have little objects, which are separated from each other. Then, used closing and dilate() mathematical transformation to make it more larger.

2.4 Finding contours

Next step after all filtering, masking, noise reduction and mathematical transformation, we should to classify active regions. For that purposes in the standard OpenCV functionality, there is cv2.findContours() function. This function scans our cleaned binary image pixel by pixel from the right-bottom corner to left-top corner. When algorithm get 1, it labels contour by number identifier, which takes digits iteratively. Here we don't care about inner shape of object, so that we can take external contour, which called cv2.RETR_EXTERNAL.

2.5 Validation

Next stage is validating objects to their own classes. We remember that there are three main features, which classify objects as cars and people. They are width, height and area. Now for validation, we are interested in the area of an object. We can classify people and cars if only if they have an area more than 250 pixels, other ones related to 'other' class.

Contours that has area more than 2000 pixels and width is more than height could be classified as CAR and contours that has area less than 6000 pixels and height is more than width could be classified as PEOPLE.

2.6 Virtual boundary

Initially, without any bounding checking operations, if we consider people class appearance in the frame, contour can be uniquely classified without any bug, because of normal shape of PEOPLE class. But if we consider appearance of the car, the contour or the object is not appeared fully in the frame. This object is going to be classified as PEOPLE instead of CAR untill the height is more than width.

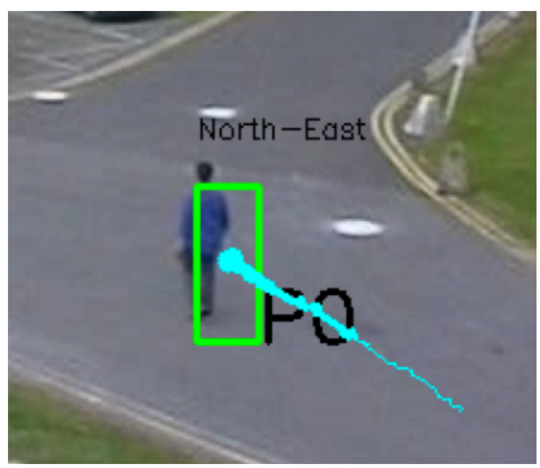
For precisely classification objects, it has created a function check_boundary which eliminates any not holistic objects, in other words not consider, if they have coincided edges with left, right and bottom corners.

2.7 Classification and vector tracking

For the best tracking It has created two classes as Location of object, Pull of Location-s, which are common for PEOPLE and CARS classes. In Location we store main information starts from position finishes with trajectory. Using OOP, we make every validated and classified object encapsulated. Every object has own status of aliveness and has functions that checks activity of object.

Pull is necessary for staying at the same identification which is given to the active region. Because we consider only the sequence of independent images, there is no any concern of moving object. To classify current active region as previous active region, the distance between current object and previous object should be nearby located. So that we consider relation of all objects with current object. The nearest active region by coordinate should be near less than 30 pixels for CARS and 15 pixels for PEOPLE. If there are no any nearest objects in previous frames, create new Location object in the Pull.

All results such as appearance, disappearance and new directions are logging into log file with identifiers and time.



3 CONCLUSION

In summary, we have performed both an experimental and theoretical study of the background subtraction, motion detection, thresholding, and classifing. The experimental results have been successfully interpreted by listed operations. Main problems (same id assignment, active region direction, bounding objects) is done as estimated. To improve the result, machine learning might be needed. Since the distances and the size cannot be accurately determined.

We got some much needed practical experience with image processing and were able to learn about multiple techniques and strategies.

A HEADINGS IN APPENDICES

A.1 Introduction

A.2 Experimental and Computational Details

A.2.1 Background Subtraction

A.2.2 Thresholding image

A.2.3 Morphological Transformations

A.2.4 Finding contours

A.2.5 Validation

A.2.6 Classification and vector tracking

A.3 Conclusions

A.4 References

ACKNOWLEDGMENTS

This work was supported by assistant professor Pedro Mendes Jorge.

REFERENCES

[1] L. Shapiro, G. Stockman, "Computer Vision", 2001, Prentice Hall, ISBN – 0-13-030796-3;
[2] R. Gonzalez, R. Woods, "Digital Image Processing", Pearson International Edition, 3rd edition, 2008, ISBN: 0-13- 505267-X;
[3] Mubarak Shah, Fundamentals of Computer Vision, December, 1997, on-line publication;
[4] Joseph Howse, OpenCV Computer Vision with Python, PACKT Publishing, 2013;
[5] D. Baggio et al, Mastering OpenCV with Practical Computer Vision Projects, PACKT Publishing, 2012;