Real-Time Directional Object Tracking

Deogratias N. LUKAMBA[1], Sadiq MACAULAY O.[1], Abdul Salam Rasmi ASRAF ALI[1], David FOFI[2]

I. Introduction

Now-a-days, the world contains immense quantities of Visual information and there exists several methods to analyze, understand and use those information for advanced applications. One such Method discussed in this project is **Visual Tracking**. Tracking finds its application in vast areas of Computer Vision to extract Motion Information in Video Sequences.

Real-time Visual Tracking (also known as Object Tracking) is extensively used and improved in many domains such as in Robotics, Control and Intelligent systems and it is currently becoming core of innovation. It is taking over tasks in many Industrial Needs to reduce Human efforts and provides efficiency.

In this project, We were intended to implement Object Tracking in real-time by applying a popular technique, **Background Subtraction**. Later, We developed an User-friendly Stand-alone Application which performs Region based Directional Object Tracking (DOT). Our implementation is in Python using MobileNet-SSD of OpenCV-dnn Module (for Multiple Object Tracking).

II. Related work

A. Background Subtraction

Background Subtraction is mostly used in computer vision-based application as a pre-processing step. In this technique, a Static Scene in an Image(or frame) is considered as Background. Any Objects present in a New Image(or frame) which is not found in the Background is considered as Foreground. Later we can subtract those Images to acquire the Objects of Interest (say the Moving Objects, *see figure 1*). A mathematical expression of Background Subtraction is given below [1].

$$R_t(x,y) = f_t(x,y) - B(x,y) \tag{1}$$

$$D_t(x,y) = \begin{cases} 0 \quad \textit{in motion} \quad R_t(x,y) > T \\ \\ 1 \quad \textit{no motion} \quad R_t(x,y) \leq T \end{cases} \tag{2}$$

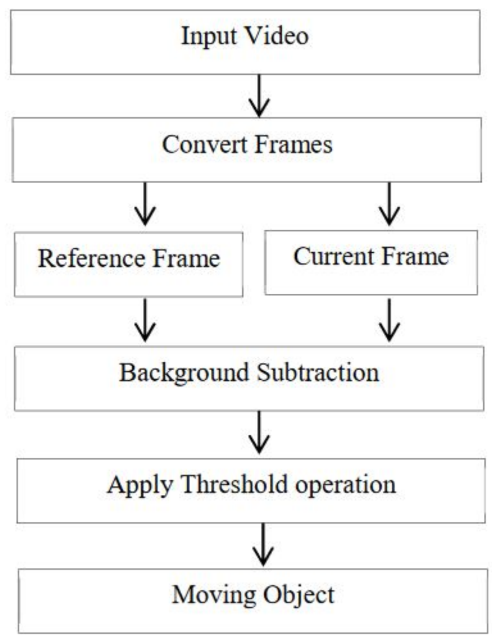


Fig. 1. Background Subtraction Flowchart

Background Subtraction is sensitive to noise as it deals with Binary Images. It becomes more complicated if we consider the object's shadow. The simple motion in shadow will also be considered as foreground. In order to overcome this defect, many techniques have been developed by Researchers

[1]Masters in Computer Vision, University of Burgundy, 71200 - Le Creusot, France
[2]Professor - University of Burgundy, VIBOT ERL CNRS 6000/ ImViA EA 7535, Le Creusot, France

such as Mixture of Gaussian (MOG), Adaptive MOG [2] and Bayesian Estimation [3], etc.

In short, the Mixture of Gaussian takes the first frame of a Video Sequence as a reference and subtracts with all other frames. The Adaptive Mixture of Gaussian takes the previous frame as reference and subtracts with the next frame. In Bayesian Estimation, we consider N no. of frames to build a background model which is considered as reference frame and then subtract it with a new frame.

B. Morphology-Based Single Object Tracking

Building upon the knowledge of the simple and basic methods of background subtraction algorithm for object detection, we have attempted to implement a method for Single Object tracking in videos considering the morphology of a selected group of pixel in the initial frame and matching it with all the video frames.

The goal behind this implementation is intended for use in a sparsely occupied scene with low activity level such as a video taken from the webcam of a computer. An user-defined region is drawn with a bounding box around the area occupied by the object to be tracked on a frame. The morphology of object within the defined region can then be tracked in the the subsequent frames of the video. Thanks to the implementations of some robust object tracking algorithm in openCV like the Kernelized Correlation filter (KCF) algorithm which we have used for our computation to keep track of the selected object throughout the video sequence.

A typical application of this kind tracking in a video sequence is used in conjunction with object detection which we have employed in our implementation of Region based DOT described in the following section.

III. Object Detection and Directional Tracking

In this section, We described the process and implementation of the typical application of object tracking in video sequence following an ideal application for detection and tracking and counting of objects (say moving cars or pedestrians) crossing from one region to another with a surveillance camera *(see figure 2)*. The work-flow of our implementation can be subdivided into *1. Object Detection and Labelling, 2. Object Tracking, 3. Region Separation and Object Counting*.

A. Object Detection and Labeling

Following the stream of our work presentation starting from various sorts of prepossessing of our input video frames such as image enhancement for noise reduction, to the primary implementation of background subtraction for object detection in the video sequence, and then description of the objects detected from subsequent frames of the video which we have employed in the previous implementations. The same idea is employed here, however, thanks to the use of the development of the **blobFomImage Algorithm** in OpenCV-dnn Module which uses pre-trained data for object detection. Object detected from the frames of video is specifically selected (person, car, motorbikes, etc.) and labeled using the centeroid of the selected objects in the image frame.

B. Object Tracking

In object tracking, there is the common problem of occlusion, and sudden disappearing of objects that was present in the previous frame and then suddenly reappears again. The reappearing object will get new label. Since the goal of our work is to keep track of an object throughout the video sequence, we have employed the Centeroid Tracking Algorithm [5] in order to minimize the effect of occlusion or lost object in the video sequence. This algorithm is based on the computation of the euclidean distances between existing centeroids on the current image frame and comparing them with a newly detected object in consequent image frames *(see figure 3)*. The primary assumption of the centroid tracking algorithm is that a given object will potentially move in between subsequent frames, but the distance between the centroids for frames $F_t and F_{t+1}$ will be smaller than all other distances between objects. Therefore, by associating centroids with minimum distances between subsequent frames we keep track of the object without assigning a new label to it [5].

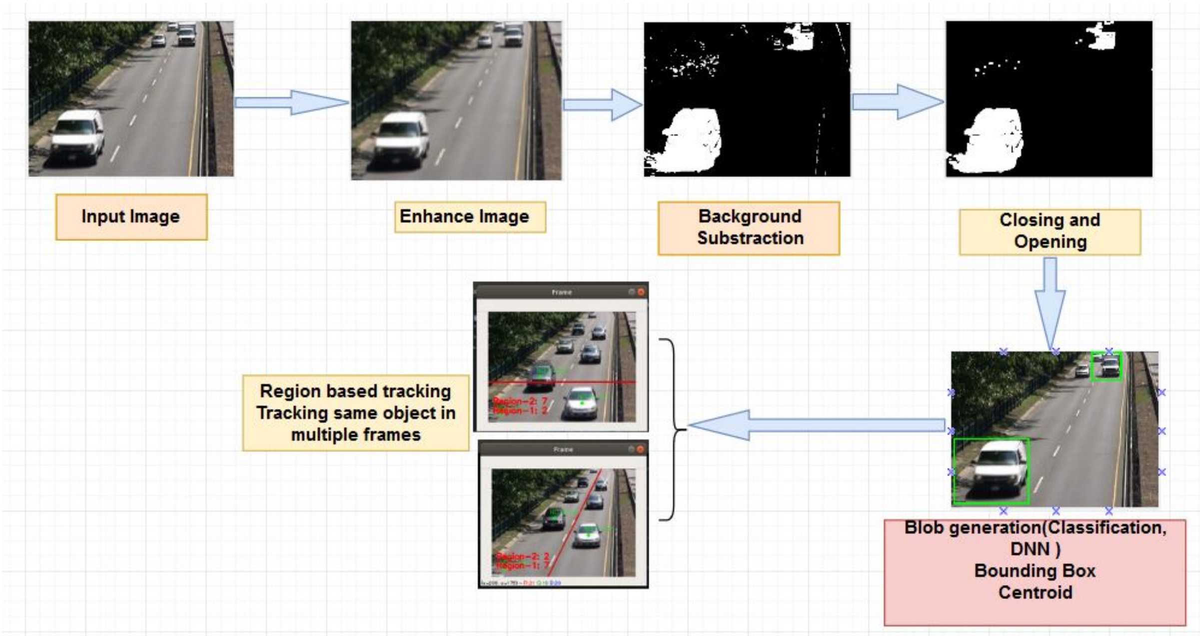


Fig. 2. Methodology

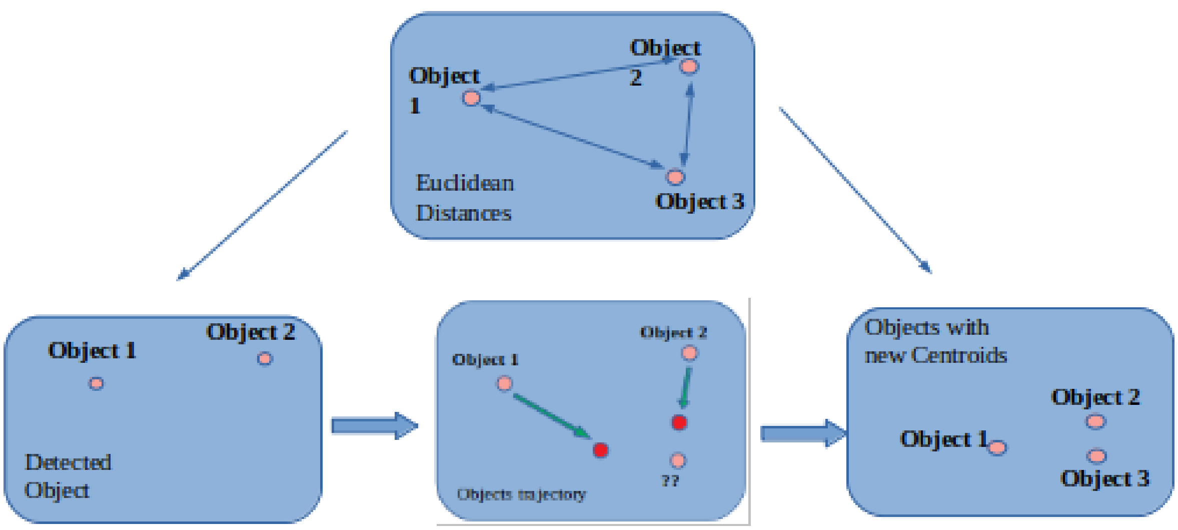


Fig. 3. Centeroid Detection Algorithm

C. Region Separation and Object Counting

With a successful detection and directional tracking of object,in the video sequence, each unique Object-ID is recorded and stored. Further we have implemented a **Region Separation Algorithm** that separates the each Video Frame into two regions with the help of a line which acts as a Boundary. This Implementation is purely based on the equation of line, $y = mx + c$. The Coordinates of any two points on the frame acquired from User-Selection serves as an input to the Region Separation Algorithm for computing the Boundary line. This enables the Separation of Region based on User's needs. Each time a target object crosses the line boundaries, the object is said to be moving ***in*** **or** ***out*** depending on the users definition of the regions and the direction of the moving objects.

IV. Results and Discussions

We achieved better Results for Real-time Tracking. The Region Separation works well but in some cases the Boundary Line skews away from the selected Points hence, further modification is needed to make the Algorithm more efficient. The Results are shown in the figures 4, 5 & 6. This Application is well suited for Real-time Applications as the Boundary line to distinguish the regions can be changed by the User in Real Time.

V. Conclusion and Future Scope

In this project, We presented an efficient Real-time Region-based Directional Object Tracking using Background subtraction technique. The proposed method reduces the occlusion problem thanks to the use of **Centroid Tracking Algorithm**. Also **Region Separation and Object Counting Algorithms** were introduced. Some minor difficulties were observed whilst defining the line equation that distinguished our regions and this can be improved in the Future Work. This method gives better results than the Previous methods of Object Tracking.

References

[1] Massimo Piccardi, Background subtraction techniques: a review, 2004 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2004.

[2] Andrew B. Godbehere, Akihiro Matsukawa, Ken Goldberg, Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation, 2012 American Control Conference (ACC), IEEE, 2012.

[3] Zoran Zivkovic, Ferdinand van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern recognition letters, Elsevier, 2006.

[4] OpenCV Documentation, Background Subtraction, `https://docs.opencv.org/3.3.0/db/d5c/tutorial_py_bg_subtraction.html`.

[5] Adrian Rosebrock, Object Tracking, Tutorials, `https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/`

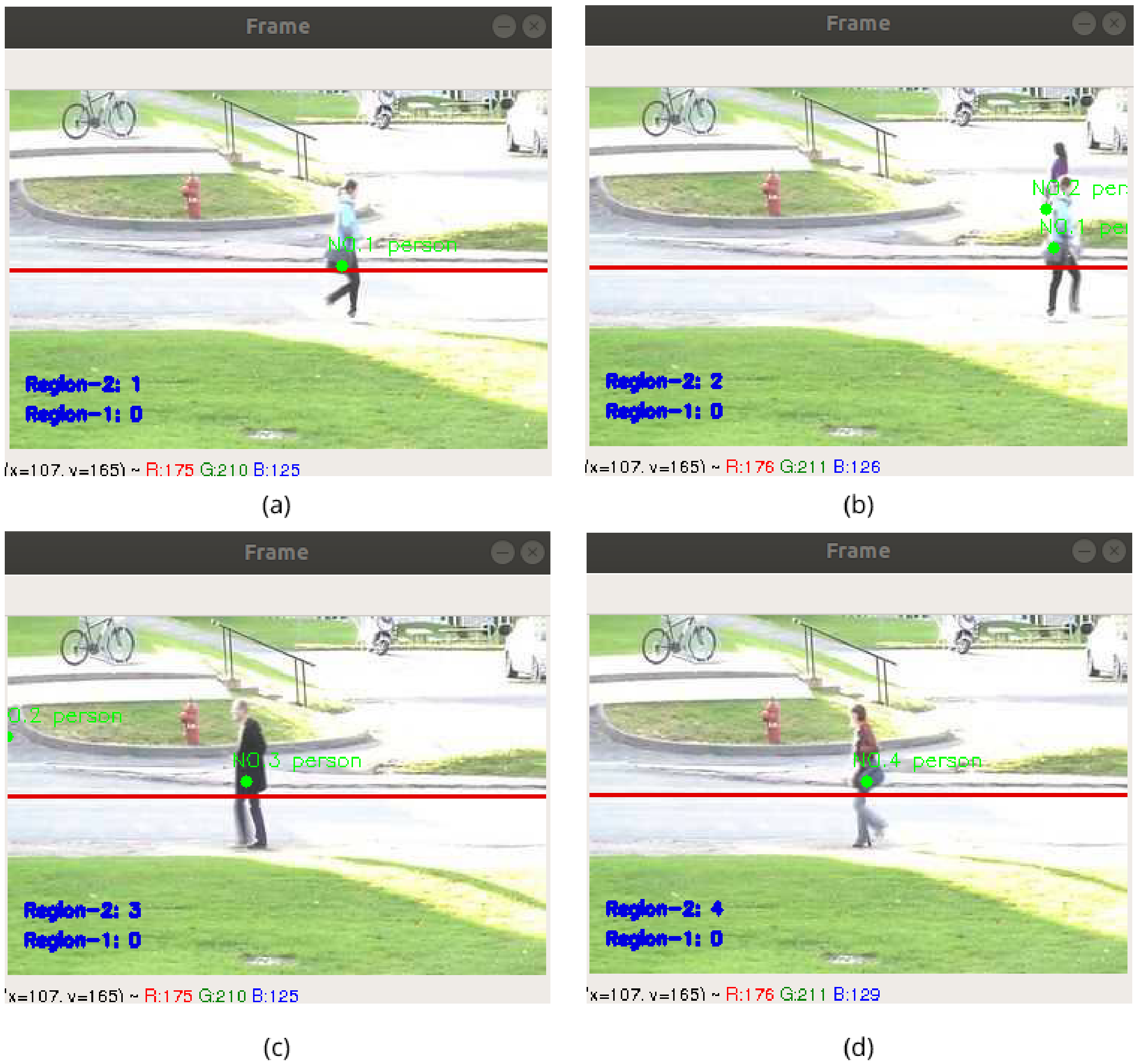
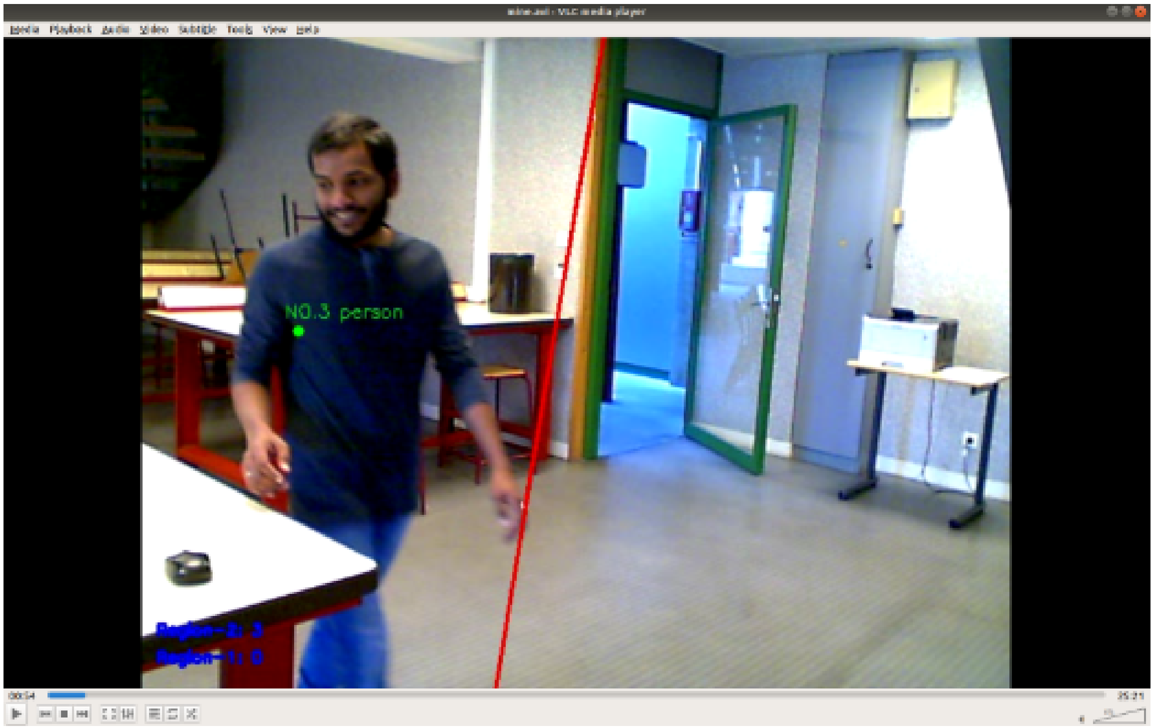
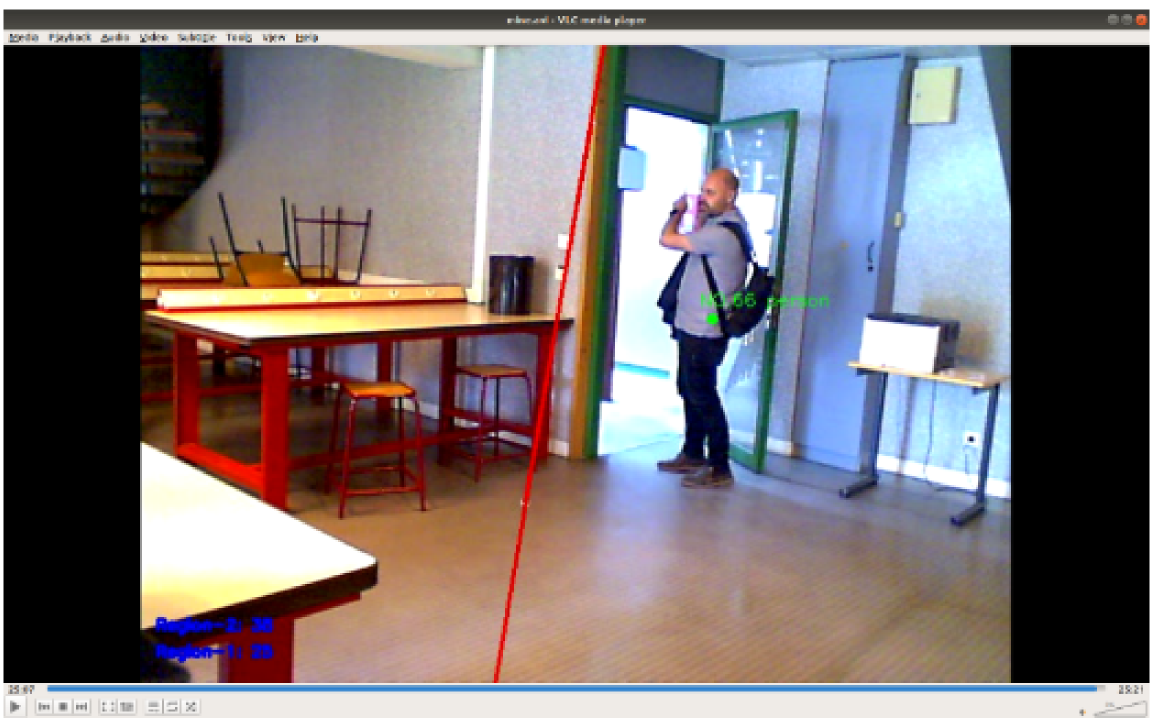


Fig. 4. Output of Region based Directional Object Tracking (With Default Boundary Line). We can see that the Centroids of all the Persons are above the Boundary and so they are classified as in the Region-2.

Fig. 5. Output of Region based Directional Object Tracking (With User-defined Boundary Line).



(a)



(b)

Fig. 6. Video Capturing (Output can be Saved for further Investigations)