

The Application of Image Processing to Solve Occlusion Issue in Object Tracking

Yun Zhe Cheong¹ and Wei Jen Chew^{1*}

¹ School of Engineering, Taylor's University, 47500 Subang Jaya, Selangor, Malaysia.

Abstract. Object tracking is a computer vision field that involves identifying and tracking either a single or multiple objects in an environment. This is extremely useful to help observe the movements of the target object like people in the street or cars on the road. However, a common issue with tracking an object in an environment with many moving objects is occlusion. Occlusion can cause the system to lose track of the object being tracked or after overlapping, the wrong object will be tracked instead. In this paper, a system that is able to correctly track occluded objects is proposed. This system includes algorithms such as foreground object segmentation, colour tracking, object specification and occlusion handling. An input video is input to the system and every single frame of the video is analysed. The foreground objects are segmented with object segmentation algorithm and tracked with the colour tracking algorithm. An ID is assigned to each tracked object. Results obtained shows that the proposed system is able to continuously track an object and maintain the correct identity even after is has been occluded by another object.

1 Introduction

Multiple object tracking (MOT) is a process that commonly uses an image processing system to track objects in a busy environment [1]. Development of multiple object tracking has begun since 1988 [2]. Since then, many different tracking techniques have been proposed to improve and enhance the current object tracking techniques [3]. Typically, multiple object tracking recognizes and tracks each unique object with a specific ID [4]. However, there are several issues that affect the MOT process. One of the issue is the occlusion of objects [5].

Occlusion often occurs when two or more objects come too close and seemingly merge or combine with each other. Image processing system with object tracking often wrongly track the occluded objects [6]. Sometimes, after occlusion, the system will wrongly identify the initially tracked object as a new object [7].

Figure 1 depicts the occlusion issues in 3 images. As observed from the 3 images in Figure 1, the normal colour tracking system only track the exposed region and neglected the occluded region. As observed in the second image of Figure 1, the blue object was wrongly

* Corresponding author: weijen.chew@taylors.edu.my

tracked as 2 separate objects. The main objective in this paper is to solve the occlusion issue that occurs in object tracking with the application of image processing.

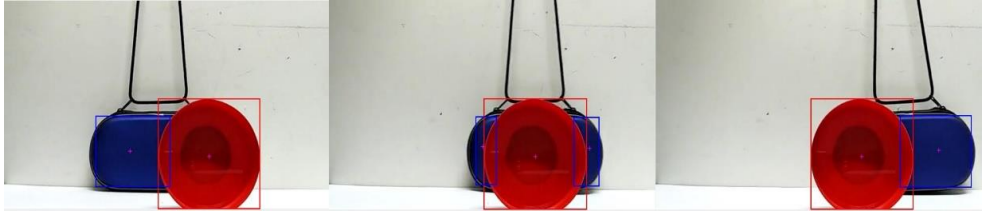


Fig. 1. Occlusion issues in multiple object tracking.

2 Research Methodology

Generally, the research methodology is separated into four different phases. These phases are the modelling phase, simulation phase, measuring phase and data collection phase, as shown in Figure 2.

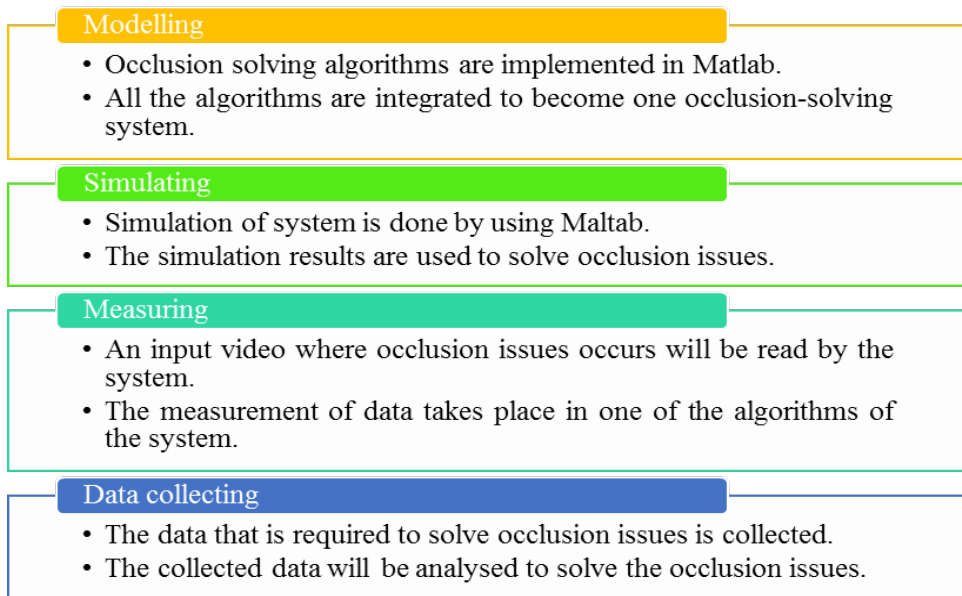


Fig. 2. Project overview chart.

Firstly, an input video will be read by the system. Next, the system extract and reads every frame from the input video with the frame extracting function. After all the frames from the video has been read, each frame will then be processed by foreground object segmentation algorithm. Mean colour values of the foreground objects will be detected with the colour detection algorithm and then the foreground objects will be tracked based on their own mean colours values with a given ID. In order to identify tracked foreground objects clearly, the coordinate, area, width, and height of each foreground object will be identified using the object specifying algorithm. Finally, the occlusion solving algorithm will be implemented to solve any occlusion issues the video had.

2.1 Overall Configuration of the system

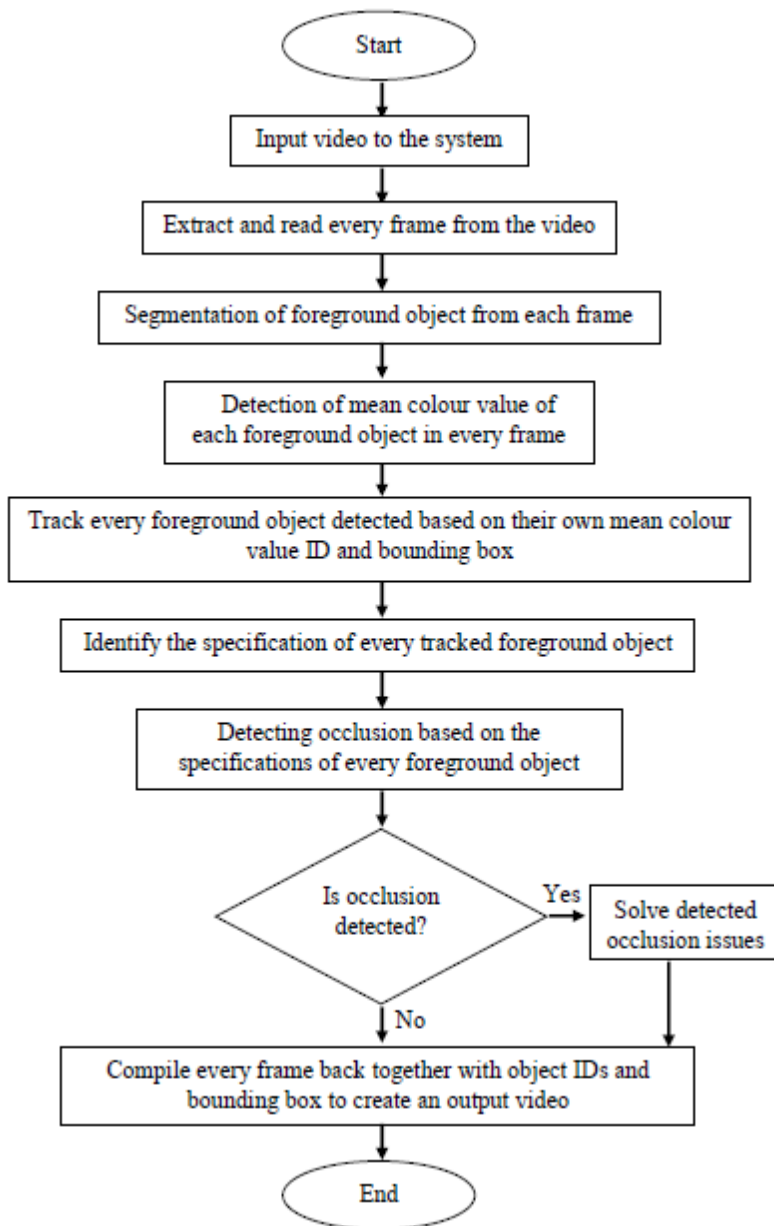


Fig. 3. Flowchart of the proposed system.

The flowchart of proposed system is shown in Figure 3. Generally, the system starts off by extracting all the frames of the input video and then read it frame by frame to segment the foreground objects. After the segmentation of foreground objects, the system will further analyse the mean colour value of each object and track it with an ID and a bounding box. The system identifies the tracked object's specifications such as height, width and area to detect the occurrence of an occlusion. The system will then proceed to handle the occlusion

issue if it occurs. Finally, after handling the occlusion issue, the system will compile every post-processed frame into one output video.

2.2 Object segmentation and colour tracking algorithms

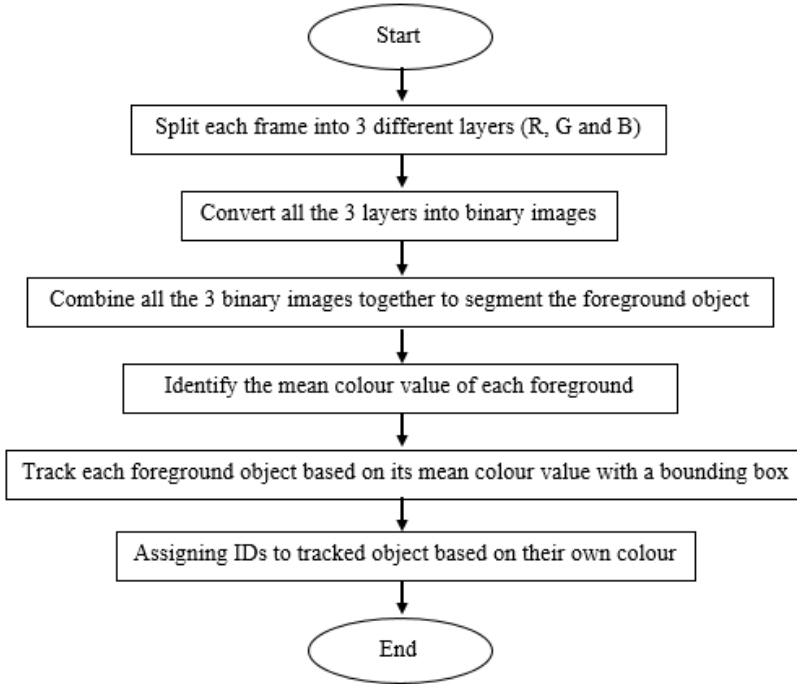


Fig. 4. Flowchart of object segmentation and colour tracking algorithms.

Figure 4 shows the flowchart of object segmentation and colour tracking algorithms. The segmentation of foreground objects is done by combining 3 R, G and B layers. After the object segmentation process, the mean colour value of each segmented object is identified. After the mean colour of each foreground object is identified, the algorithm proceeds by tracking the object with a bounding box and assigning an ID to the object in the frame.

2.3 Object specifying algorithm

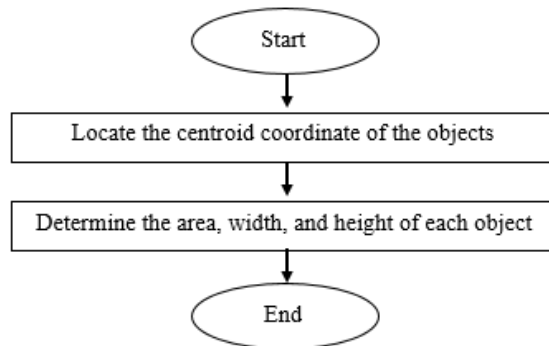


Fig. 5. Flowchart of object specifying algorithm.

Figure 5 shows the flowchart of the object specifying algorithm. Technically, this algorithm detects every tracked object specification in every frame. First, the algorithm locates the centroid coordinate of every tracked object. Next, it determines and saves the area, width and height of every tracked object.

2.4 Occlusion handling algorithm

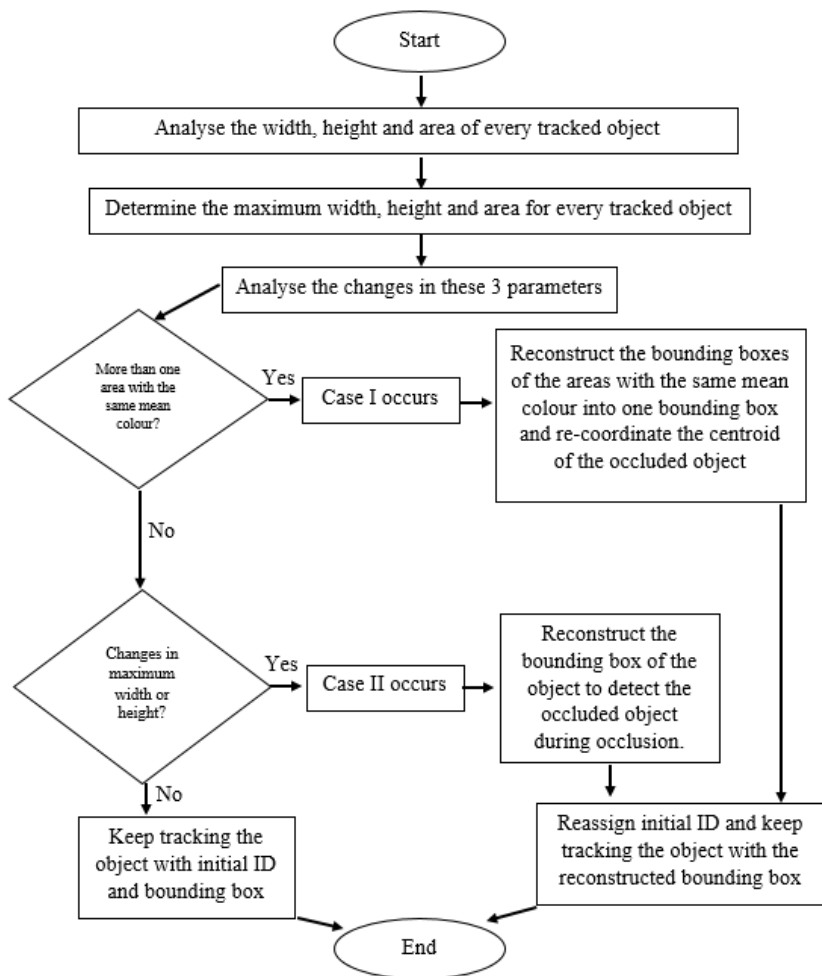


Fig. 6. Flowchart of occlusion handling algorithm.

Figure 6 shows the flowchart of the occlusion handling algorithm. In this algorithm, the saved data from the previous algorithm will be analysed to detect whether occlusion issues occur in a frame. Before deciding whether occlusion do occur, the maximum width, height and area of every tracked object is obtained from the previous algorithm. Case I occurs when 2 or more areas of with the same mean colour is detected. Technically, an object with a mean colour only has one area. If the number of areas of the same colour is 2 or more, then the object is being occluded in the middle while exposing its edges at the left and right sides. Therefore, it is considered that the object is moving horizontally from left to right and

being blocked in the middle by another object. The Case I occlusion issue will be solved by replacing the 2 bounding boxes of two different areas with a reconstructed bounding box to track the middle-occluded object from behind. Case II occurs when the change in width or height of an object is detected. For example, considering the same motion of moving horizontally from left to right. When the width of an object changes significantly along with the area of the object, the object might be occluded either on its left side or right side. Case II occlusion is solved by reconstructing the smaller bounding box that only tracks the exposed part of an object, with a new bounding box that tracks the whole object including its occluded part

2.5 Limitation of the proposed system

The proposed system's scope is limited to solving occlusion issues in recorded video with multiple single-coloured objects and a white background. Also, the shapes of all the objects must remain unchanged throughout the video.

3 Simulation Results

For this paper, three different videos with different occlusion issues were tested with the proposed system. The results were recorded and discussed below.

3.1 Object segmentation and colour tracking

In the object segmentation and colour tracking algorithm, the foreground objects are segmented out from the background. Firstly, the RGB frames in a single frame is split into 3 different layers. These 3 different layers are R layer, G layer and B layer, each layer carries its own colour value. R layer carries red value, whereas G and B carry green value and blue value respectively. Next, every layer is converted to binary images. By combining all the 3 RGB binary layers together with certain binary thresholding value, the foreground objects can be segmented from the white background. Since white background has lower mean colour value, it does not consider as a foreground object. The combination of the 3 RGB binary layers are shown in II of Figure 7.

The average mean colour of an object, V_a , is determined by calculating the average value of R, G, B (R_a, G_a, B_a) in the pixels inside the bounding box. After obtaining the average mean colour, the range of mean colour is set as $[R_a \pm 8\%, G_a \pm 8\%, B_a \pm 8\%]$ and any pixel within the range of mean colour is treated as part of a foreground object's pixel. Once a foreground object with a mean colour is tracked, its pixel-area can be obtained as shown in IV and V of Figure 7. Finally, ID is given to every tracked foreground object according to its mean colour. For example, in Figure 7, ID:B is given to a blue object and ID:R is given to a red object.

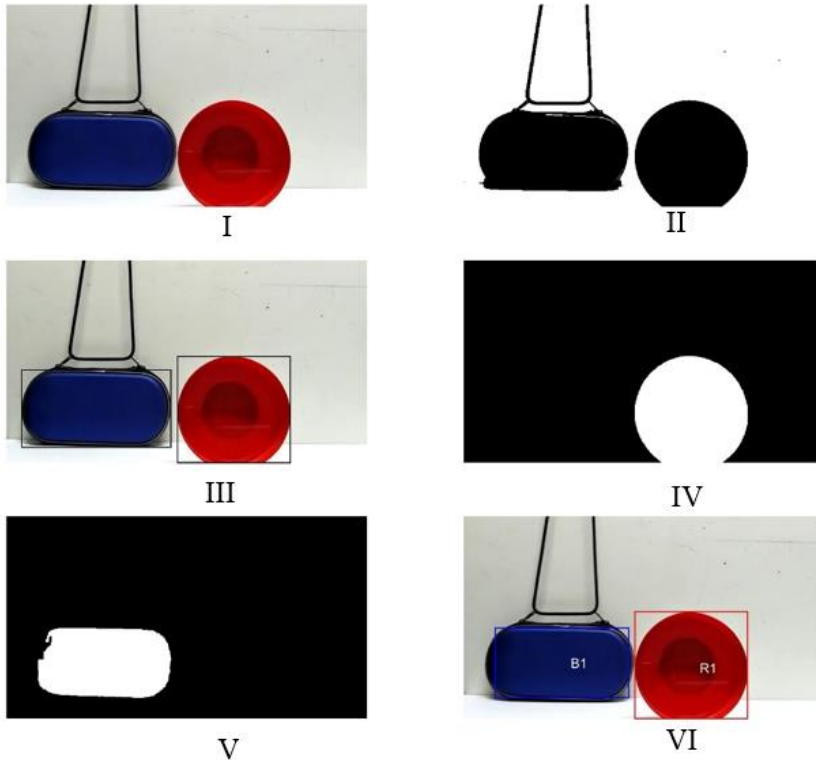


Fig. 7. Object segmentation and colour detection results.

Figure 7 shows the results of object segmentation and colour tracking algorithm. The process of the algorithm is explained below:

- I) The original extracted image of a frame.
- II) The foreground objects are segmented by processing R, G, and B layers.
- III) The mean colour value inside the box is determined.
- IV) Red colour object is detected, the white region is the area of the red object.
- V) Same as the previous step, the blue object is detected and the area of blue object is represented in the white region.
- VI) Both blue and red objects are tracked with different bounding boxes and IDs are assigned to both objects.

The criteria for constructing bounding boxes will be explained in the next section.

3.2 Object specification

Object specification algorithm is considered as a short algorithm as it only determines the specification of every tracked object. Technically, this algorithm determines the maximum areas of all foreground objects at the beginning of every input video. After the maximum area of an object is determined, the algorithm will proceed to locate the coordinates of centroid of the object. Next, the properties of the bounding box will also be determined with this algorithm.

Construction of bounding box requires four different properties. These properties are right (r_b), top (t_b), width (w_b) and height (h_b). r_b and t_b refers to the right and top starting points of the bounding box respectively. Whereas w_b and h_b determine the width and height of the bounding box.

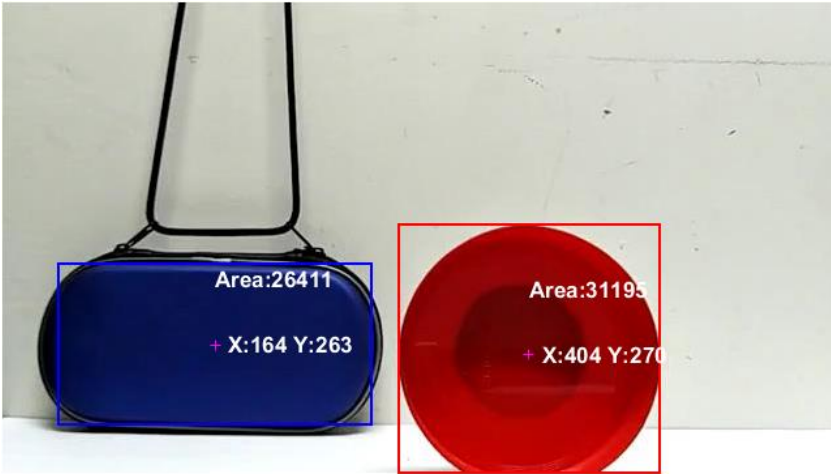


Fig. 8. Object specification result.

Figure 8 shows that each segmented object is tracked with a bounding box with its centroid coordinates identified. The centroid coordinates of the objects are important to justify which case of occlusion issue occurs.

3.3 Occlusion handling

The occlusion handling algorithm handles occlusion issues that occurs throughout the input video. In this algorithm, the bounding box of an occluded object is reshaped to accurately track occluded object with its true size. The bounding box is reshaped according to how the object is occluded. Following the project scope, there are two potential occlusion cases, Case I and Case II. Case I only occurs when an object is being occluded at the middle with its left and right sides exposed as shown in Figure 9. Case II occurs when an object is being partially occluded by another object on one side, either left or right. The different cases of occlusion issues are solved with the occlusion handling algorithm.

Case I

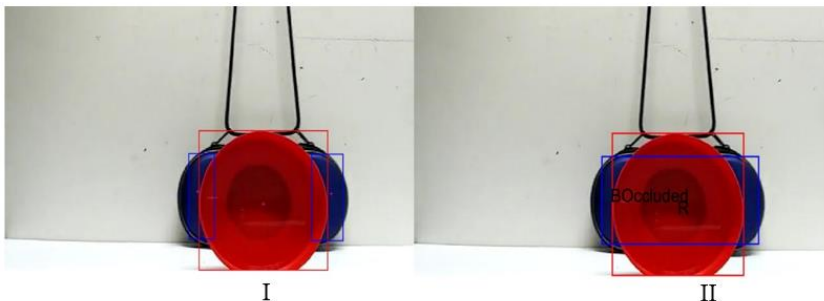


Fig. 9. Object is being occluded in the middle.

When the blue object is occluded by the red object in the middle, two different bounding boxes wrongly track a single object as two different objects with the same colour as shown in I (left) of Figure 9. To rectify this occlusion issue, a new bounding box is constructed to replace the old ones as shown in II (right) of Figure 9. In this case, the new bounding box is constructed with maximum width data obtained from the previous frame.

Since the new bounding box is tracking an occluded object, the initial ID assigned for the object will be retained along with word “occluded” to indicate the occurrence of occlusion.

Case II

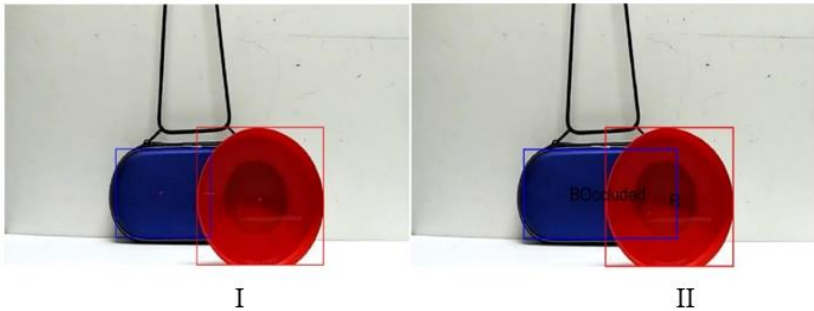


Fig. 10. Object is being occluded on the right side.

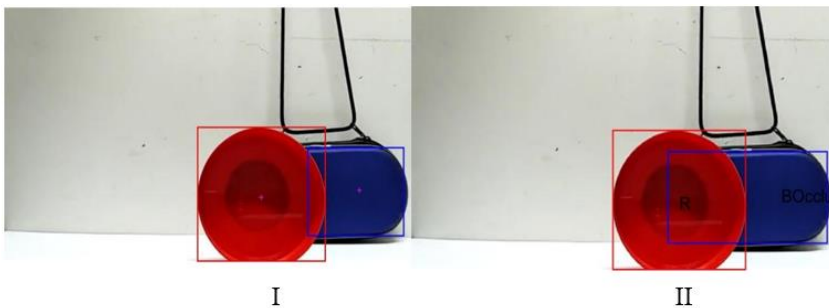


Fig. 11. Object is being occluded on the left side.

When either side of the blue object is occluded as shown in I of both Figure 10 and Figure 11, the bounding box only tracked the exposed portion of the blue object. To rectify this occlusion issue, the bounding box is reconstructed to track the blue object even when it is occluded. In Figure 10, the right side of the blue object is occluded. The system analyses the centroid coordinate of both the blue and red object to select the appropriate expansion of the bounding box. In this case, the x-coordinate of the centroid of the blue object is less than the red object and the exposed area of the blue object is smaller than the maximum determined area of blue object. Thus, the bounding box is expanded to the right to track the occluded right portion of the blue object as shown in II of Figure 10. The same principle is used to solve the occlusion issue in Figure 11, where the x-coordinate of the centroid of blue object is greater than the red object. Figure 12 and 13 shows other examples of occlusion handling by the proposed system



Fig. 12. Object is being occluded on the left side.

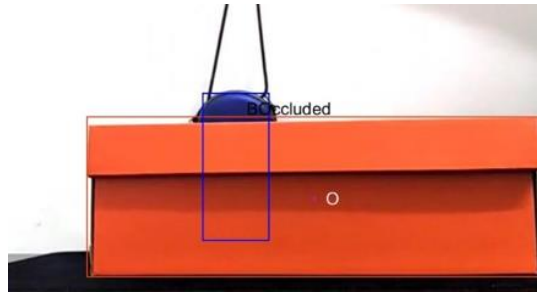


Fig. 13. Object being occluded at the bottom.

4 Conclusion

In conclusion, the occlusion issue that occurs in the input video is properly handled by using the system proposed in this paper. This proposed method was able to handle 2 objects overlapping each other and was able to identify the occluded object as a single object even though it was blocked in the middle. For future work, further improvements need to be done for handling occlusion in real-life situations with complex-coloured objects such as pedestrian or vehicles.

References

1. Y. Cai, H. Wang, X. Chen, and L. Chen, “Multilevel framework to handle object occlusions for real-time tracking,” pp. 1–8, 2016.
2. Pylyshyn, Z. (2007). Multiple object tracking. [online] Scholarpedia. Available at: http://www.scholarpedia.org/article/Multiple_object_tracking. [Accessed 26 Oct. 2017]
3. P. Ding and Y. Song, “Robust Object Tracking using Color and Depth Images with a Depth based Occlusion Handling and Recovery,” pp. 930–935, 2015.
4. I. Engineering and S. Province, “Variable Scale and Anti-occlusion Object Tracking Method with Multiple Feature Integration,” pp. 426–433, 2016.
5. W. Luo, J. Xing, X. Zhang, X. Zhao, and T.-K. Kim, “Multiple Object Tracking: A Literature Review,” arXiv, 1-39. Retrieved from <http://arxiv.org/abs/1409.7618>, vol. V, no. 212, 2014.
6. M. Intelligence, Q. Zhao, W. Tian, Q. Zhang, and J. Wei, “Robust object tracking method dealing with occlusion,” 2016.
7. Z. M. et al. Shehzad, Muhammad Imran, Yasir A. Shah, “K -means based multiple objects tracking with long-term occlusion handling,” 2016.