Contents

Introduction

The video is made from multiple image sequences called frames. Human eyes perceive that these frames are moving because these frames are running under a high frequency. In image processing, frames can be processed individually. But in a video, frames will always have a relationship with the frames that follow. In video object tracking, it will need to detect a moving object or an object with certain properties in each of the frame. Therefore, to track an object, it will need to follow steps of segmentation and recognition steps in image processing (Watve). Object tracking has a wide variety of application, for example, video compression, video surveillance, vision-based control, human-computer interfaces and medical imaging. Object tracking also enable computers to monitor the movement of objects through spatial and temporal changes during video sequence (Guo, 2001). When using object tracking in videos, there are a few methods to be implemented. The first thing would be a feature based method, that is getting all the properties of the video that can be anchored as characteristics such as points, line segment from image sequences and tracking stages so that it can be used to ensure matching procedure instantly without having to process these information again every time. Differential methods are also used to differentiate the image sequences. Other than that, correlation is used to measure the inter-image displacements. Any issues that might happen depends on all the methods and their abilities to function from the domain (Watve).

Delegation of task

In this assignment, we work as a team and we are separated the assignment into 2 part. Firstly, Chan Choon Hee and Yeow Jian Shyang were in charge of doing first frame method, thresholding, dilation, finding difference object and draw it with a rectangle. While Hoo Chun Hoong and Low Khai Wynn were in charge of finding centroid, direction of the object going, counting object and condition to close the video.

Related work with proper citation

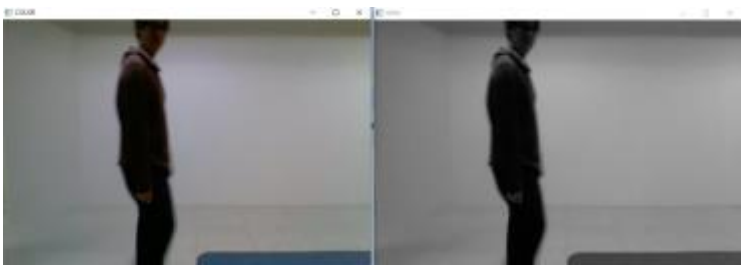
This assignment uses object detection in videos. It can detect any movement in the video and highlight the object. To identify moving objects in a video sequence, require a few fundamental steps and certain critical tasks must be executed to extract information. Videos are made up of continuous sequences of frames and the first frame method is used to set the first frame as background of the video so that any movement in the video can be detected. The method will always use the first frame of the video as background and it will be used in background subtraction method (Matin Kalcounis, Thomash Parrish, Sebastion Pauli). The background subtraction method is also known as the frame difference method is one of the easiest method to identify moving objects in videos. The background subtraction method contains many types of algorithms (Kamath, 2005). It checks whether the pixels in the frame has any changes by subtracting the frame with a background frame. By doing this, the absolute difference between the two frames can be found. This makes the background the most important component in this method as when the background isn't static, calculations made by comparing the frames will not be accurate (Singh, 2011). After extracting the information from the two different frames, thresholding is used to display the differences. The thresholding operation converts the information extracted into a binary image with values 0 (black) and 255 (white) allowing users to differentiate between objects. If required, the threshold value can be manually configured, otherwise, an algorithm will compute a value automatically which is known as automatic thresholding (Singh, 2011). By using the contour function, the edges of the objects can be determined and it will draw the edges continuously. After that, the moment function is used to determine the centroid of the object to ensure the object is in the boundary. Lastly, a rectangle will be drawn for each object in respect to the centroid by utilizing the contour value obtained from before (Jaya P, Geethu Balakrishnan, 2014).

Comparison with existing algorithm

For pyimagesearch, the algorithm to detect moving objects in a video is provided online. Pyimagesearch uses the built in background subtraction which is absdiff(), it is similar when compared to our solution. Pyimagesearch is able to find the differences between the background frame and the frame at the moment just by utilizing absdiff() while we deduct the two frames manually and then only obtain the differences by using abs. We used the same way of getting the mask area but the syntax and intensity for the mask is different. For pyimagesearch, there are no centroid points and direction indicators for the object but for our method, the direction of the objects and total objects and be calculated.

Algorithm Explanation

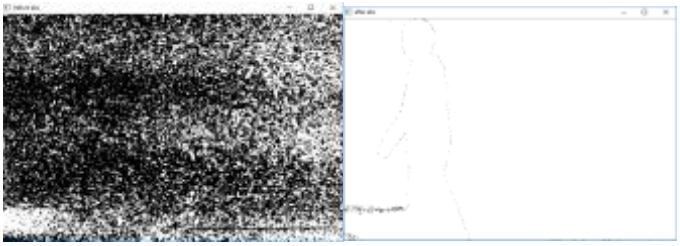
By importing numpy and cv2, it will allow me to use the function contained in the library. To do that, the code needed will be imported where numpy as np and import cv2. After doing that, it will allow the computer to read the video by using cv2.VideoCapture(). This function allows computer to read the video and store it into a variable. By so, the computer will be able to access it easier just by calling the variable. After that, video.read() is used to get the video properties such as the frame and returning the next video frame. Each of these information will be stored into a variable. Two new variables will be created in the variable explorer where the first will be grabbed and the second will be frame. By using a while loop, and the condition to start the while loop is using grabbed variable and check whether is equal to true. Cv2.resize is used to change the video windows size, because if the default video of resolution 1920x1080 is used, it will require more processing time. The frames captured should be converted into grayscale as well as thresholding to be done on it. Example of before and after converting the frame to gray.



After converting the frame into grayscale, the rows and columns of the grayscale image are collected by calling the variable named gray and the function .shape. This is done to obtain the size of the images. After that, an empty array is created with the size of the grayscale by using the numpy library and also calling the function .zeros() to create an empty array with the same row and column as the grayscale image. A check statement is then used to check whether the firstFrame variable contains anything, if firstFrame does not have anything, then gray is stored into firstFrame. This allows the firstFrame to be compared with other frames and differentiate any changes. In the image, it shows that the firstFrame variable contains the first frame and it will used to compare with other frames.

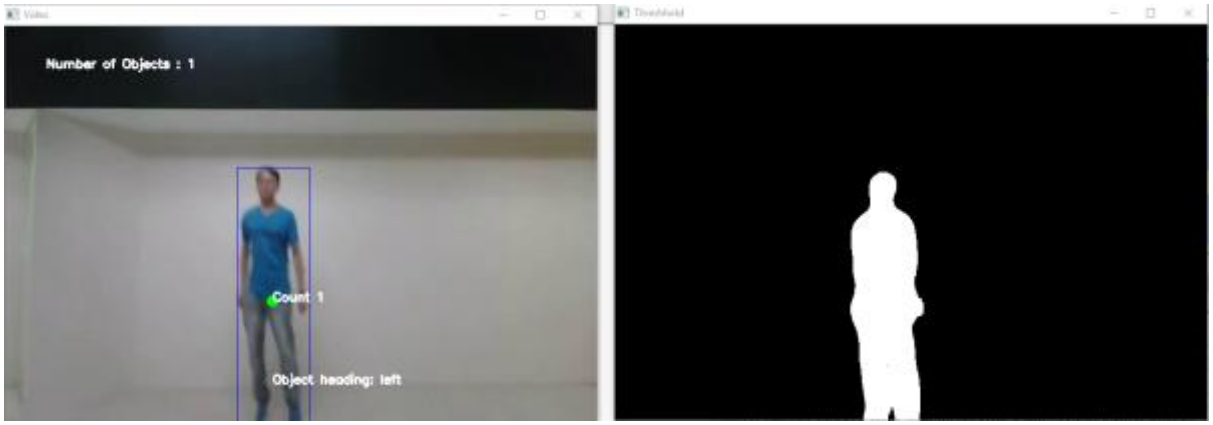
Because firstFrame is created as a uint8 datatype, so it will need to be converted into float64. To do that, the numpy library and astype function are used. It will also be used to get the grayscale image.

firstFrame will be stored into a variable and grayscale will be stored into a variable. To find the differences between two frames, it will need to use both variable in a subtraction calculation. The result will give us the remaining moving object. Due to the float number value, abs() function is used to convert the value into a number. The images below show the before and after abs function effects.



After that is done, it will use thresholding matter which is mask[diff>35] = 255. When the diff each value is bigger than 35, it will change the value to 255 which is white colour. By using the findContour function, the contour of the binary image is determined. Contour is a curve joining all the points that have the same colour or intensity. This will help us to determine where the object is located in the binary image. By doing that, we can obtain the contour and the hierarchy. Due to the nature of this application, it does not need to know the hierarchy, it will not store into any variable but for contour, it will store into an array and it will be called as cnts. By using foreach function in python which is a loop function to process each contour stored into cnts. In the for loop, it will have to calculate the area of the contour. By using the build in function in cv2, which is contourArea(), it can be computed by using the Green formula. After that, the size of the contour Area is checked whether to be smaller than 3500, if it is not smaller than 3500, execute boundingRect(). It will calculate the upright bounding rectangle of point set then draw the rectangle with the x,y,w,h provided from boundingRect(). And also, using moments() function provided by cv2 to gain access of a dictionary of moment values calculated. By using the keywords provided by the moments function, it can be used to calculate the centroid. The centroid formula is as shown, $C_x = \frac{M_{10}}{M_{00}}$ and $C_y = \frac{M_{01}}{M_{00}}$. By accessing the dictionary, it needs to be written like this, cX = int(M["m10"] / M["m00"]) and cY = int(M["m01"] / M["m00"]). After calculating the centroid, it can be used to determine many values. In this case, the centroid is used to determine whether the object is going to the left, right or stationary. To get that method, it will need to use a check statement with the previous centroid but in this algorithm, it will work on only one object. But to prevent any error from appearing while running the algorithm, it will be checked whether the frame contains two objects or one object. If the frame contains two object, it will disable the direction detection. To show how many objects are currently in the frame, it will use a normal counter which is count = count + 1 to label each object. It also be used to calculate total object inside the video in each frame. After all these processes, have been done, it will need to display all the information. Using circle() function to draw a circle with cX and cY point so that it will represent the

centroid point in each object. The putText() function is inserted to show the total object in the frame. In the image below, the drawing function mentioned is shown.

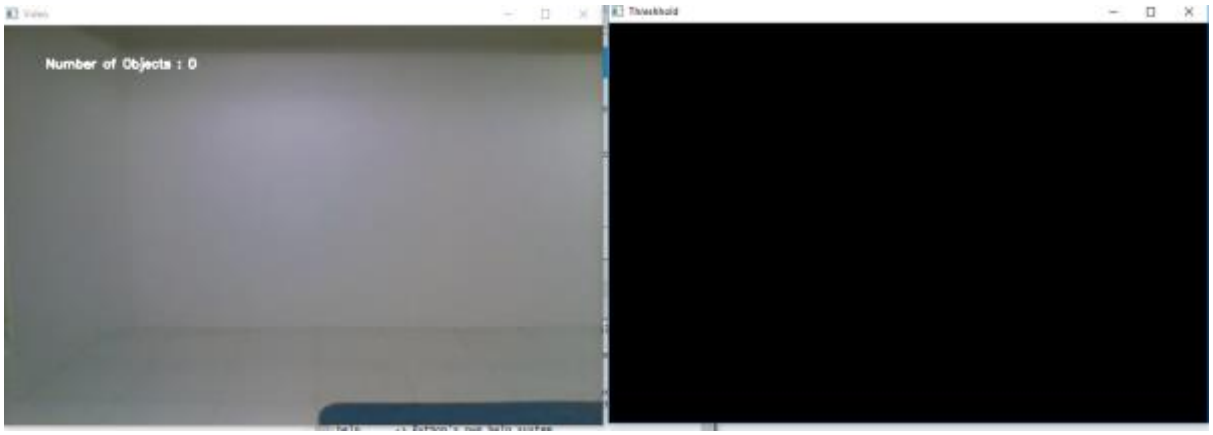


Enhancement of the algorithm

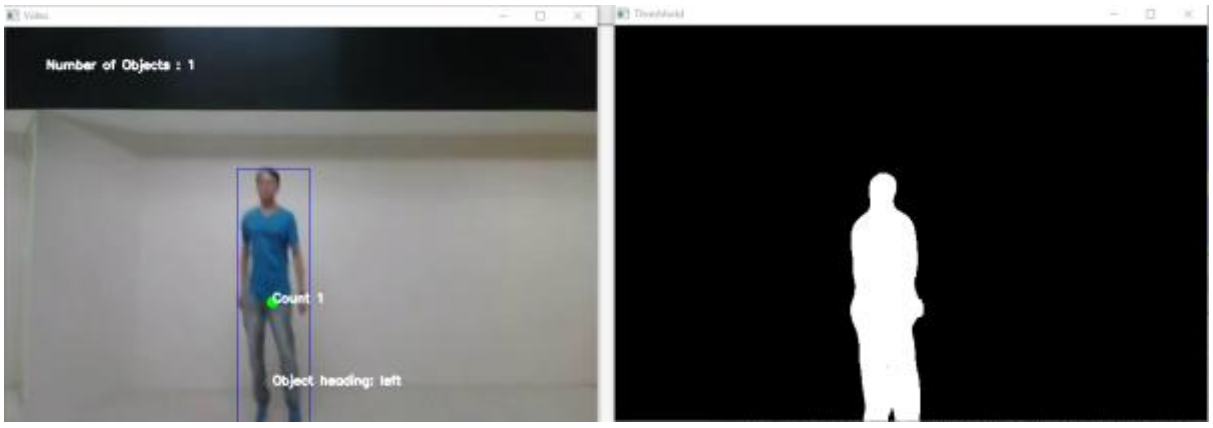
In this algorithm, the blur() effect provided by cv2 can be included. However in this video, it does not need any blurring effect to achieve the desired result. But for other videos, it might produce a better result. Non-maximum suppression can also be included as well. Non-maximum suppression is used along with edge detection algorithms. It is used to detect whether is there any multiple bounding boxes in an image. But it is useful for video object tracking because in video sequences, sometimes the frame will cause multiple false bounding box. By using this, false bounding box can be removed and make the bounding box accurate. By using colour features, objects could be traced by assigning a unique identifier. But this feature is only able to be used for objects with different colour intensity, so if there are multiple object with same colour then it will cause errors and settings will have to be adjusted. To use colour features, the colour information of an object RGB colour space will have to be described.

Result

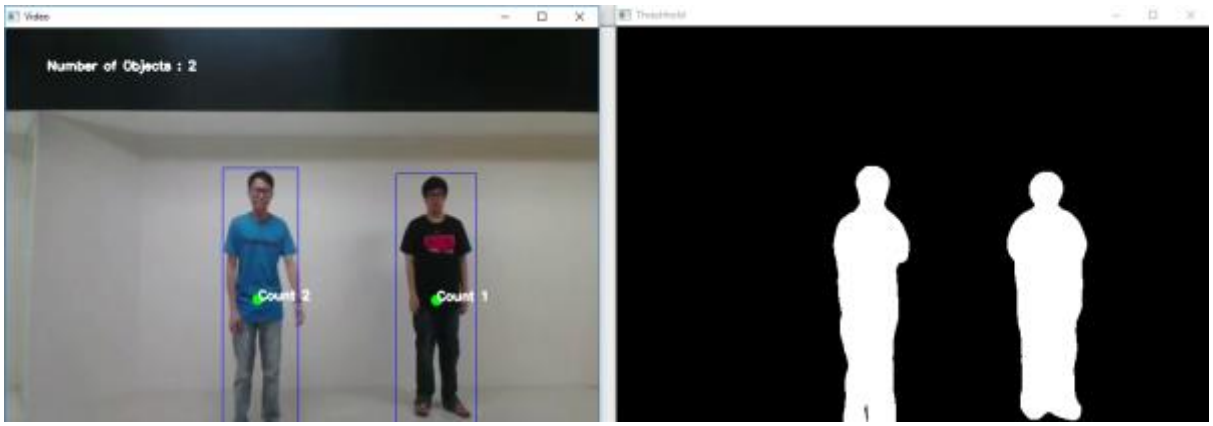
Background frame with threshold frame



First motion detected with threshold frame.



Two object motion detected with threshold frame.



Discussion

For the background frame and threshold frame shown in our result, the first screenshot occurs when the algorithm takes the first frame value into a first frame variable so that it can detect the object movement later. The second screenshot shows an object that is jumping or running around with direction. When an object is jumping, or walking to the left or right, it will display that object is heading left or right. It also has the stationary option for object. This option is only available when one object is in the frame. In the third screenshot, it shows that two object is in the frame. The direction tracking will be disabled and the normal functions will remain as usual.

The algorithm for object tracking in this project not only applicable in this video. It is also applicable in some other situation and can be tested using webcam. To test using webcam, simply change the name of the video to 0.

Conclusion

A system has been created using background subtraction and thresholding functions to determine the differences between two frames. The system was not able to detect multiple objects with the object direction function on and it does not have an object identifier or object descriptor ID. Therefore, it won't be able to understand which object existed before. It is also able to detect objects thrown into the video but it must be set in slow motion so it can be detected easier. When an object is too small, the system will not able to draw a rectangle to show that object but it can still detect the movement of the object in the thresholding figure. That part of the system can be improved and made more efficient. The improvements that can be made are listed in the enhancement section.

References

Guo, Z. (November, 2001). *Medianet*. Retrieved from Object Detection and Tracking in Video: http://medianet.kent.edu/surveys/IAD01F-objdetection/index.html

Jaya P , Geethu Balakrishnan. (2014). Contour Based Object Tracking. *International Journal of Computer Science and information Technologies*, 4128-4130.

Kamath, C. (8 August, 2005). *Computation*. Retrieved from Background Subtraction for detection of moving objects: https://computation.llnl.gov/casc/sapphire/background/background.html

Matin Kalcounis, Thomash Parrish ,Sebastian Pauli. (n.d.). *Application of Object Tracking in Video Recordings to the Observation of Mice in the Wild.* Retrieved from uncg: http://www.uncg.edu/mat/faculty/pauli/mouse/Applications-of-Object-Tracking-in-Video.pdf

S, F. (2003). *HIPR2*. Retrieved from Thresholding: http://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm

singh, R. (12 August, 2011). *slideshare.* Retrieved from Background subtraction: http://www.slideshare.net/ravi5raj_88/background-subtraction

Watve, A. K. (n.d.). *School of Information Technology.* Retrieved from Object Tracking in video scenes: http://www.sit.iitkgp.ernet.in/research/aut04seminar1/12r.pdf