

Measurement of Displacement and Velocity of a Moving Object from Real Time Video

Pritam Das

Department of Information Technology
St. Thomas' College of Engineering and Technology
Kolkata-23, West Bengal, India

Ranjit Ghoshal

Department of Information Technology
St. Thomas' College of Engineering and Technology
Kolkata-23, West Bengal, India

Dipak Kumar Kole

Department of Computer Science & Engineering
St. Thomas' College of Engineering and Technology
Kolkata-23, West Bengal, India

Rabindranath Ghosh

Department of Electronics & Communication
Engineering
St. Thomas' College of Engineering and Technology
Kolkata-23, West Bengal, India

ABSTRACT

This article focuses on an efficient algorithm for measuring object displacement and velocity from real time video. The proposed technique for object identification and tracking is based on background subtraction with optimized threshold binarization. Mapping techniques have been developed to relate image with real world. The algorithm is also capable of working with a bad lighting conditions using histogram equalization approach. Further, the real scenarios like presence of noise, shadow, and multiple moving object environments have been taken under consideration for developing the algorithm.

General Terms

Algorithms, Computer Vision, Image Processing, Object Tracking.

Keywords

Video Capture, Histogram Equalization, CLAHE, Binarization, Background Estimation, Background Subtraction, Mapping, Camera Coverage, Camera Span, Displacement and Velocity, Memory Management, Time Complexity.

1. INTRODUCTION

Detection and tracking of object have always been the most interesting part of image processing. Complex algorithms are developed from time to time to detect objects using various approaches. Surveys have been carried out to find the available methods for Object Tracking [1]. Feature based object tracking [2] also gained significance in various fields which isolate each object based on its featured appearance in image. A number of techniques have been proposed in different researches for providing object-centric information such as orientation, area or shape [3] [4] which helped in document analysis like Devnagari and Bengali font extraction proposed by Bhattacharya et al [5] and Text extraction techniques comparison by Graham Leedham et al [6]. Those analytical approaches have also been proved beneficiary for real life applications like medical diagnostic systems [7]. These static imaging approaches create the base for object tracking for real time video analysis from live camera. More efficient techniques are proposed to overcome the challenges in real time tracking systems [8], where extracted information from video are used for further data analysis. Johan Sommerfeld proposed the techniques for tracking objects from single camera [9]. Recent researches have proposed methods for real life application with real time camera

but they are also limited in their domain and application environment. Some examples are Anomaly Detection in Surveillance Video using Color Modeling [10] and Accident Prevention Using Eye Blinking and Head Movement [11]. Moving a step further we have tried to expand the domain where object properties can be mapped directly from image to real world scenario. From detected object in the image we tried to relate the object properties i.e. displacement and velocity with some supplied information and constraints. The proposed approach not only tracks an object but also deals with a velocity calculation method which can be extended to any other domain to support other systems.

Rest of this article is organized as follows: Section 2 describes some technical aspects of video capturing, color to gray image conversion techniques, histogram equalization approach and background estimation technique. The proposed method is discussed in Section 3 along with Section 4. Section 3 illustrates the detection technique and Section 4 discusses the displacement and velocity calculation. Finally, Section 5 and Section 6 show the experimental results and conclusion respectively.

2. PREPROCESSING

2.1 Video Capture

Video capture is the basic input part of our algorithm. Digital video camera devices (not still cameras) are used for this purpose. Algorithm is tested through laptop inbuilt webcam and wired webcam situated at any position. MATLAB platform is used for accessing the device and taking video frames, using the device drivers and adaptors installed in the windows operating system. Input video is taken as RGB format video frames of default size supported by the device. Images are dumped into the memory buffer for further processing. Memory management is done through clearing the used frames so that available buffer does not crash; otherwise the algorithm will be terminated abnormally in between.

2.2 RGB to Gray Conversion

Here we are interested only in object detection, so full information of the image frames like color information, color format specification etc, are not required. They will eat up the memory as well as make the time complexity worse for any computation part. So when we start processing, the frames are converted into grayscale images. This is done by the formula $G = 0.299 * R + 0.587 * G + 0.114 * B$ applicable for pixel by pixel conversion from RGB to grayscale.

2.3 Histogram Equalization

We have considered the condition of bad camera quality or lighting conditions. After converting the frame to grayscale image if it is found to be blurred or not up to that quality where objects can be separated, this histogram equalization technique is applied. It enhances the contrast of the grayscale image by transforming the values using contrast-limited adaptive histogram equalization (CLAHE). Histogram is the statistical representation of image data according to the frequency of appearance of illumination/grayscale pixel values. CLAHE operates on small regions in the image, called tiles, rather than the entire image. Each tile's contrast is enhanced analyzing the local histograms and modifying them. The neighboring tiles are then combined using bilinear interpolation to eliminate artificially induced boundaries. For better performance instead of applying CLAHE on the grayscale image it is applied separately on the R, G and B component of the image before conversion. However it does not differ much in the processing if the enhancement is applied on gray image but processing in color image helps in better representation of the video where marked object will be shown.

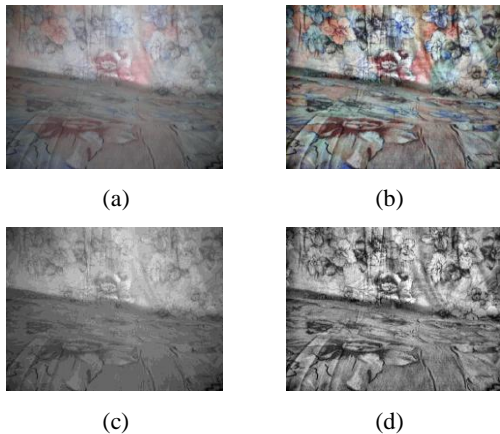


Figure 1: (a) RGB frame, (b) Histogram Equalized RGB frame, (c) and (d) Grayscale conversion of (a) and (b) respectively

2.4 Background Estimation

We use the base platform as background subtraction for detection of a moving object. Background subtraction is a computational vision process of extracting foreground objects in a particular scene. A foreground object can be described as an object of attention which helps in reducing the amount of data to be processed as well as provide important information to the task under consideration. Often, the foreground object can be thought of as a coherently moving object in a scene. We must emphasize the word coherent here because if a person is walking in front of moving leaves, the person forms the foreground object while leaves though having motion associated with them are considered background due to its repetitive behavior. In some cases, distance of the moving object also forms a basis for it to be considered a background, e.g. if in a scene one person is close to the camera while there is a person far away in background, in this case the nearby person is considered as foreground while the person far away is ignored due to its small size and the lack of information that it provides. Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of

video frame that differs from the background model. For simplicity we restricted our testing environment to a fixed static background and also we have taken the first captured video frame as background considering that no object is present in the first frame. We can use some other techniques if it is needed for any situation. It can be an average of previous n frames using various techniques like Gaussian calculation of pixels, Eigen space calculation, Temporal Median filter etc where small movements of camera or jitter can be handled. Those can also be useful in the cases where image capturing camera is shifted to various locations frequently.

3. PROPOSED METHOD FOR OBJECT IDENTIFICATION AND TRACKING

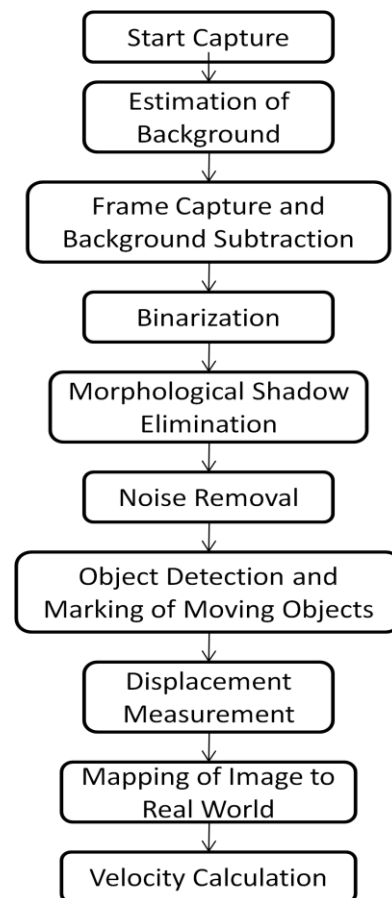


Figure 2: Process Overview of the Proposed Method

3.1 Background Subtraction using Current Frame Approach

After getting our background model frame the algorithm starts the detection process with each frame. As described for the background frame, from next frame onward each incoming RGB frames are converted into grayscale image by the same process. Then this image is taken as current frame image and background is subtracted from this frame image. This subtraction is done simply by image matrix subtraction where the gray scale value of the background frame is subtracted from the current frame value pixel by pixel. Except the moving part the other portions of the scene is static with respect to background. So the subtraction results in zero value in those

areas, which is interpreted as black in the subtracted image. Only the object parts (moving) result in numeric value or gray scale values. Object grayscale value can be less than that of the background which produces negative value in subtracted image. So for subtraction absolute difference is taken, as gray scale value cannot be negative. Finally, the result image contains black part as “nothing” and grayscale parts as “objects”.

3.2 Black and White Conversion using Minimum Threshold Approach

Binarization of image (video frame) is the most important issue in this process. Our focus is limited to a binarization algorithm which can give an output where desired objects can be pointed out as pure white parts in image. Otsu’s global thresholding method and binarization algorithm is tested to be effective in any real scenic image. This method calculates a threshold grayscale value for the whole image, based on which foreground objects are separated from the background. But when we apply it on the difference image generated by background subtraction it is bounded to separate the desired objects partially, as the difference image only contains the moving part. If we apply Otsu’s algorithm on that part also the object gets divided into parts as background and foreground. So we need to set the threshold empirically only to reduce the noise, and not to divide our object. It has been seen that setting the threshold to 0.0784 results into effective output. Theoretically this value is 0.0039 for an ideal condition where no noise is there, but it is quite impossible to achieve this for any practical application.

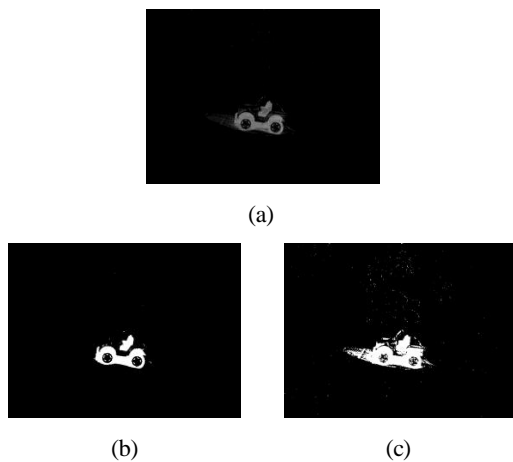


Figure 3: (a) Background subtracted grayscale image with object, (b) Binarization with Otsu, (c) Binarization with minimum threshold

3.3 Shadow Elimination using Mathematical Morphological Operation

In any practical scenario presence of shadow is obvious. As the shadows are always connected with corresponding objects it is also binarized and becomes the part of the object. However the shadow part is not as solid in nature with respect to object. So, we use morphological erosion operation to convert the shadow part into disconnected object as noise. For the erosion operation a disk shape structuring element is used with a minimum radius one, because we do not want to erode the desired object. Only the porous shadow part is desired to be converted to noise, which will be removed later at the time of noise removal.

3.4 Noise Removal using Pixel Connectivity

After binarization the image contains object(s), but there can be some unwanted area of image where noise is present as pseudo object(s). Shadows are also converted to noise in the previous step. We need to remove those parts before further processing. For this, we use opening operation using pixel connectivity to remove unwanted objects. Sufficiently small connected components containing less than 500 pixels are removed as noise.

3.5 Object Detection

Binarization followed by noise removal makes the image (transformed video frames) so particular that the objects can be marked directly using connectivity of the pixels. After the noise removal is done only objects are left in the frame as white part and the rest part is converted to black. Thereafter we can detect all groups of connected pixels in the frame and label them as objects. We always consider that multiple objects are detected in this process. So for all objects found we calculate the area and thereafter the biggest object is selected as our desired object. Then according to the label of the detected object boundary the centroid is calculated. With this information we mark it in original video frame sequence.

4. DISPLACEMENT AND VELOCITY MEASUREMENT

On the detected object obtained by the methods discussed above we apply movement and velocity calculation technique.

4.1 Displacement Measurement

With the detection of the object we calculate and record centroid position of its area. From the next frame onwards we track and measure the movement of centroid in terms of co-ordinate geometry as pixels positions in an image can be treated as a 2D graph. The new centroid position is recorded. This process continues until the object disappears from image frame. After that we get the total displacement of the object. But this displacement is only in the image domain in pixel unit. For real movement we go for the following technique.

4.2 Image Mapping Technique

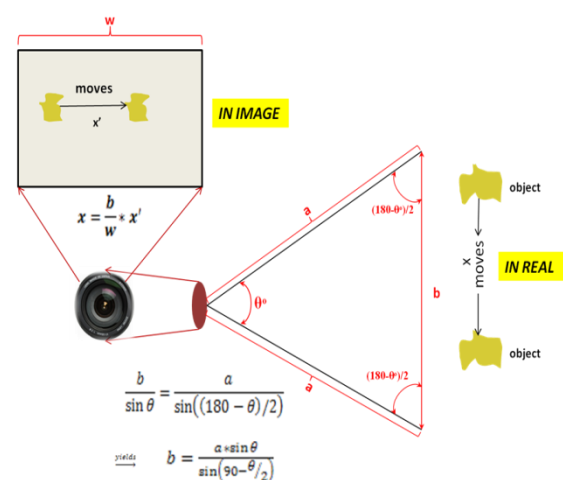


Figure 4: Image Mapping Technique

4.2.1 Calculating Camera Coverage at a Particular Distance

To calculate real movement with a camera-captured image at first we need to know the camera properties. Suppose we have

a camera which has a span of θ degree. Then we can calculate the coverage area of the camera at a certain distance. We consider the distance to be a and the coverage area to be b . According to the properties of triangle, from the above diagram we can write the following equation:

$$\frac{b}{\sin \theta} = \frac{a}{\sin((180 - \theta)/2)}$$

$$\xrightarrow{\text{yields}} b = \frac{a * \sin \theta}{\sin(90 - \theta/2)}$$

So, to get b we just need to know camera span θ and the distance from camera i.e. a .

4.2.2 Measuring Movement of an Object

After knowing b we can now measure movement of an object at distance a just from captured images (video sequence) by the camera. We know that the width w pixel of the sequential images corresponds to the coverage area at any distance. So when we know the distance of an object as a then we can say that movement of 1 pixel of the object in image is equal to b/w in real. If we see that in video object has moved x' pixel, then the real displacement x of object (at a distance) will be:

$$x = \frac{b}{w} * x'$$

4.3 Velocity Measurement

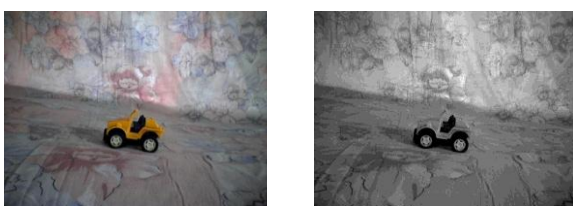
A timer is started as soon as an object is found in the video sequence. The timer expires with the disappearance of the object. From this timer we get to know the time span for which the object was in movement. We also now know the measurement of real displacement of the object. So, the velocity can be calculated by the simple formula

$$\text{Velocity} = \frac{\text{Displacement}}{\text{Time Taken}}$$

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed algorithm is applied to an experimental environment of a fixed background, in front of which a yellow toy car was set to move along the perpendicular to camera focus. We have used laptop inbuilt webcam with 640*480 resolution for this experiment. Captured images are of RGB format on which all the processing is done. The implementation is represented in the Figure 5.

In the depicted figures, from 5(a)-5(f), the processing steps are shown where binarized object is finally found in 5(f) as white part. From this image we track the object and mark it in the original video sequence 5(g).



(a) (b)

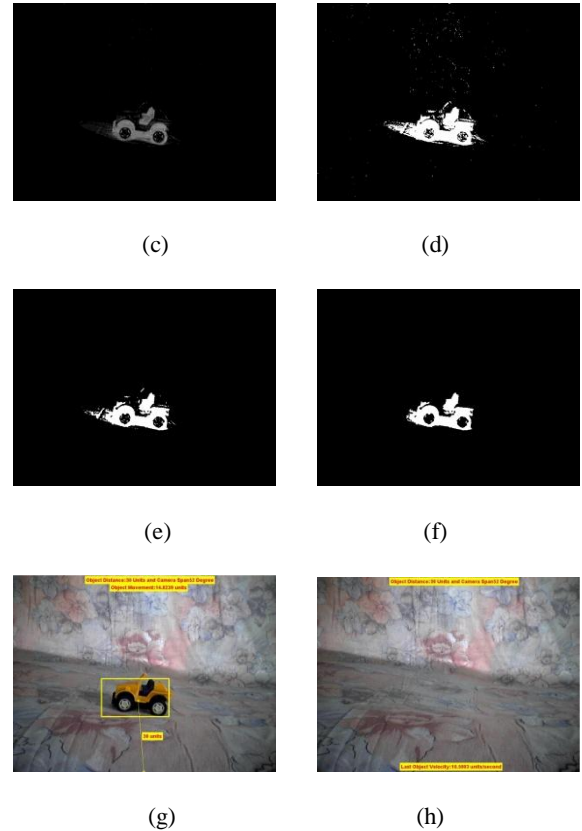


Figure 5: (a) RGB frame with object, (b) Converted grayscale frame, (c) Background subtracted grayscale image with object, (d) Binarized image using minimum threshold, (e) Shadow eliminated image, (f) Binarized image after noise reduction, (g) Tracked object in original video (h) Showing tracking result

In this experiment the object distance and the camera span were 30 cm and 52 degree respectively. Manually intercepted velocity was 10.914 cm/sec and the velocity measured by the algorithm was 10.5 cm/sec. So the error found is 0.379 %. The experiment was designed to show the result in the video itself. Figure 5(h) shows the video snapshot where result is displayed after the object is disappeared from the scene.

5.1 Memory Management

For a real-time continuous tracking process, sequences of captured frames need to be stored in memory. Programmatically there is a restriction of memory buffer in terms of limit for storage and also we need not have all the previous frames to be retained as no recording is needed for future use. We just need to track an object and give outputs for that moment only. So, for the process all the previous frames, except those which are needed, are deleted in a cyclic order for clearing the memory.

5.2 Time Complexity

As all the tracking process is of real-time we have to design all these things keeping in mind the time complexity of running a program. Faster implementation of programming logic is used among a few after comparing and testing them all, under same conditions. However sometimes time complexity may be dependent upon the performance of the computing system. This has not been considered by our algorithm.

6. CONCLUSION

Detection of velocity or speed for any moving object from a single camera is a challenging problem for the real time video system. All possible real scenarios of noise and blurring have been experimented successfully with our algorithm. The proposed algorithm can be extended to vehicle speed detection system where from a fixed camera speed analysis can be done for all passing vehicles. The algorithm is also compatible with conventional grayscale imaging video cameras as our base computation is on grayscale image.

In our algorithm we have assumed that the distance of the object is known, because the distance is needed to establish the mapping relation. In this case we do not have any idea about the object, but in any case if we have the knowledge of the shape or length of the object we can establish the relation from this information itself; we need not to know the distance. We have measured displacement and velocity of an object with our algorithm, but it also creates the provision and future aspect as predicting movement trajectories from the current statistics of object. Taking the camera captured image as reference we can create a virtual 3D environment and plot the predicted future position of the object in that dimension. For that we also need to analyze the movement trajectory if it is not a straight line. Curve equations can be used as reference models in this approach and also momentary acceleration of the object is more important here rather than a constant velocity calculation. However it would be a much complex approach with respect to the time constraint for real time approach.

In future, use of machine learning tools can be used to improve the performance of the proposed algorithm as well as to develop further. This is useful for training the system for better detection and prediction with the knowledge of previous experience. But, it also has its own limitations to be applied in real time systems and they should be taken care of properly.

7. ACKNOWLEDGMENTS

This work has been fully supported through “Computer Vision Based Robotic Interception” Project funded by the Department of Science and Technology, Govt. of West Bengal, Bikash Bhavan, Bidhannagar, Kolkata 91, India.

8. REFERENCES

- [1] Alper Yilmaz, Ohio State University, Omar Javed, ObjectVideo, Inc., and Mubarak Shah, University of Central Florida, Object Tracking: A Survey, ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication date: December 2006.
- [2] Duy-Nguyen Ta, Georgia Institute of Technology, Wei-Chao Chen, Natasha Gelfand and Kari Pulli, Nokia Research Center, Palo Alto, SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors, CVPR 2009:2937-2944.
- [3] XuLiang, Image Binarization using Otsu Method, NLPR-PAL Group, CASIA Conference, pp 345-349, 2009.
- [4] N. Otsu. A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, vol.9, no.1, pp.377-393, 1979.
- [5] U. Bhattacharya, S. K. Parui and S. Mondal, Devanagari and Bangla Text Extraction from Natural Scene Images, 10th International Conference on Document Analysis and Recognition, 2009
- [6] Graham Leedham, Chen Yan, Kalyan Takru, Joie Hadi Nata Tan and Li Mian, Comparison of Some Thresholding Algorithms for Text/Background Segmentation in Difficult Document Images, Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), 2003.
- [7] Dipak Kumar Kole and Amiya Halder, Automatic Brain Tumor Detection and Isolation of Tumor Cells from MRI Images, International Journal of Computer Applications, Vol. 39, No.16,pp.26-30, February 2012.
- [8] Damien Lefloch, Real-Time People Counting system using Video Camera, Department of Computer Science and Media Technology, Gjøvik University College, Norway, 2007.
- [9] Johan Sommerfeld, Image processing and object tracking from single camera, KTH Electrical Engineering, Stockholm, Sweden 2006-12-13.
- [10] M Gangadharappa, Pooja Goel and Rajiv Kapoor, Anomaly Detection in Surveillance Video using Color Modeling, International Journal of Computer Applications, Vol. 45, No.14,pp.1-6, May 2012.
- [11] Abhi R Varma, Seema V Arote, Chetna Bharti and Kuldeep Singh, Accident Prevention Using Eye Blinking and Head Movement, IJCA Proceedings on Emerging Trends in Computer Science and Information Technology -2012, April 2012.
- [12] Ahmad Sedkyadly, M B Abdelhalim and Amrbadr, Analyzing and Measuring Human Joints Movements using a Computer Vision System, International Journal of Computer Applications, Vol. 45, No.20,pp.21-29, May 2012.
- [13] Amiya Halder, Soumajit Pramanik and Arindam Kar, Dynamic Image Segmentation using Fuzzy C- means based Genetic Algorithm, International Journal of Computer Applications, Vol. 28, No.6,pp.15-20, August 2011.
- [14] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Prentice-Hall, Inc., second edition, 2001.