ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Estimating Speed, Velocity, Acceleration and Angle Using Image Addition Method

Sawan Singh

Third Year Student, Dept. of ECE, UIET, CSJM University, Kanpur, India

ABSTRACT: This paper leads us to a new and easy way to estimate the velocity, speed, acceleration and angle at which the object deviates by processing the image in MATLAB. The image addition is used to find the final image which is then further processed in order to the basic quantities like speed, velocities e.t.c. This method provides us the ability to work with very small objects even small as the pixel of the image. Image addition does not require the background images to process the data it only requires the range of RGB values of the object in order to convert the image into binary image.

KEYWORDS: Object detection; Speed; Velocity; Acceleration; Angle of deviation; Digital image processing; MATLAB.

I. INTRODUCTION

In today's world of research and experiment it is very necessary to calculate the speed velocity acceleration of many particles to many points of digits. The objects may be of very small in size. Calculating these parameter in real time is a very challenging task. Therefore the first thing we need to setup a system to calculate these parameter is the detection of the moving object therefor the need is to develop more efficient and accurate detection systems.[1] A typical way to detect the object is the method of background substation. In this method we need the stationary (background) image than this image is been subtracted by the images taken in real time to detect the object.In the new method which is image addition we do not require the background image we just need the RGB range of the object. Moving object detection is always an integral part of many automated system such as visual surveillance, tracking system and also to avoid collision.

II. RELATED WORK

In [2] authors have used the image subtraction method to detect the image and then applying the threshold values they estimate the velocity. In this method the background is obtained then the incoming frame is obtained. With the background model a moving object can be detected this algorithm is called image subtraction. Any small change in the background can make this algorithm to fail. In the authors have added a step in the algorithm which is GMM. A Gaussian mixture model(GMM) is used. GMM was proposed for the background subtraction in Friedman and Russell [3] and the efficient and update equation are given in Stauffer and Grimson [4]. This method uses a Gaussian probability density function to evaluate the pixel intensity value. It finds the difference of the current pixel intensity value and cumulative average of the previous values.

III. PROPOSED ALGORITHM

A. *Design Considerations:*

- A rough range of RGB values of the object is known.
- The rate at which the camera is taking images is known (frame rate).
- A rough idea of image size is known.
- Image is of uniform color.
- Background is of uniform color (not of the color of object).

DOI: 10.15680/IJIRCCE.2015. 0311070

B. *Description of the Proposed Algorithm:*

Aim of the proposed algorithm is to draw a point at the centroid of the object and then the next image is taken and we again draw a point to the centroid of the object then we add these two images. The addition of these two images give us a image having two points (dots). Then we can easily calculate the velocity speed and other parameters. The proposed algorithmis divided into 4 parts :-

Step 1: Pre image processing:

Every image is passed from the preprocessing. In this the range of RGB values of the object is given to a function which make each pixel 1(white) falling in the range of object's RGB values and make other pixels 0(black). After this the images is converted into a binary image that is having only '1' and '0' at the pixels. Some other operations are also performed in order to get a sharp image. This image is stored in a matrix and the next image is than processed.

Step 2: Labelling images and getting Centroid:

After pre-processing of the images we have 2 binary images. After this we take the first image which is '1'(white) at the object and '0'(black) at all other places. When we use label function in MATLAB which is *BWlabel* this labels the object area as "1". Now centroid of this labelled area can be easily calculated by the *regionprops* function. The centroid is saved in a matrix of 1x2 as the centroid contain 2 values one the x coordinate and other is the y coordinate. The above procedure is also applied to the second image and the centroid is saved in different matrix. The image matrixes may be deleted now as we extracted the centroid information.

Step 3: Image addition:

Till this stage we have 2 different centroid for the object moving. After this we create a very small circle to the a point (x1,y1)(coordinates of 1st image) and another circle at point (x2,y2)(coordinates of 2nd image). After this we add both the images containing small circles at the different centroid positions. Then we get a final image with 2 small circles (dots) which represent the path of the object.

Step 4: Calculations [5]:

i) *Speed :-* $\frac{\sqrt{(x2-x1)^2+(y2-y1)^2}}{Frame\ \ rate}$

ii) Angle :- $\tan^{-1}\frac{(y2-y1)}{(x2-x1)}$

iii) *Velocity:-* SPEED X ANGLE = $\frac{\sqrt{(x2-x1)^2+(y2-y1)^2}}{Frame\ \ rate}$X $\tan^{-1}\frac{(y2-y1)}{(x2-x1)}$

iv) *Acceleration:-* $\frac{\mathrm{v2-v1}}{\mathrm{Frame\ \ rate}}$

IV. PSEUDO CODE

Step 1: Save the first photo.
Step 2: Check the range of RGB values of object
Step 3: Now use the following to convert and get a binary image

```
If
     ((image(i,j,1)>RED1 && image(i,j,1)<RED2)&& (image(i,j,2)>GREEN1 && image(i,j,2)<GREEN2) &&
                    (image(i,j,3)>BLUE1 && image(i,j,3)<BLUE2))
binaryimage(i,j)=1;
else
        binaryimage(i,j)=0;
```

Step 4: Calculate the centroid of the object by labeling it first then by using *regionprops* function in MATLAB.
Step 5: Do other operations for making a sharp binary image so it contain only one object.(removing noise)
Step 6: Now save the binary image.
Step 7: Now take the second image and do all the process form step 1 to 5.
Step 8: Add the two binary images.
Step 9: Do calculation.
Step 10: Goto step 1
Step 11: END

V. SIMULATION RESULTS

The figure (a) is the first image taken by the camera. This is the initial position of the object. The pre-processing starts as soon as we get aur first image.Now doing pre image processing to the figure(a) which consist of reading all the RGB values of each pixels and comparing it with the range of RGB values of the object. If the RGB value lies in the range we make that pixel logic ‘1’ that is white and other pixel logic’0’ that is black. So we obtain the figure (b)

Figure (a)

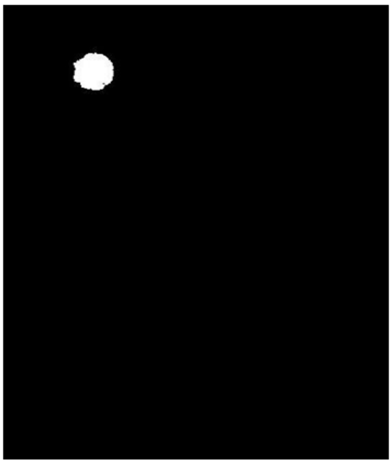

Figure (b)

Now when we done with the processing of the 1st image and we obtain the figure (b) as a result than we label this figure and we calculate the centroid and store it in a variable. Now we have to click another image which will give us the next position of the object. Figure(c) is the new position of the object. We again do all the processing we have done for the previous image and we obtain figure (d) as the result.

Figure (c)

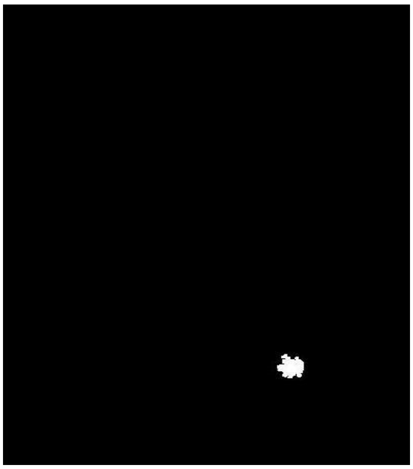

Figure (d)

Till now we have processed 2 images and saved the centroid of object at its 2 different positions. By the centroid values we can easily calculate different parameters very easily. The centroid can be further converted into very small circle in order to measure the very small distance as if the distance is small than the size of the object on addition the image will contain only one image which has both the objects overlapped. Therefore replacing these objects into a small circle would make our way easy.

We will add both the images in order to visualise the path of the object. The addition gives us the figure (e) which shows both the position of the object and now the further images can be added and labelled in order to get the full path map of the object.

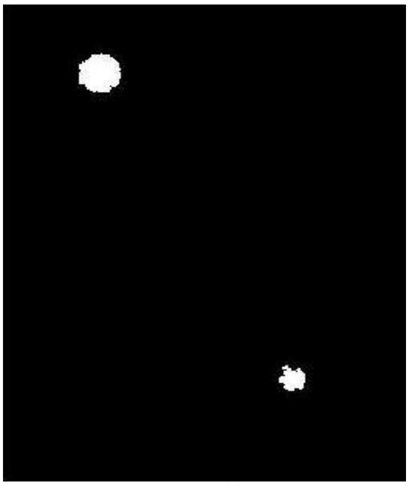

Figure (e)

VI. CONCLUSION AND FUTURE WORK

The simulation results showed that the proposed algorithm performs better with comparison to the basic image subtraction method used. This method is need to be more explore in order to design complicated algorithms. Image addition is the future of object tracking. This method provide us a very simple approach in order to get the very sophisticated parameters which are very important to be calculated in order to do many operations and calculations.

DOI: 10.15680/IJIRCCE.2015. 0311070

REFERENCES

1. A. Yilmaz, O. Javed and M. Shah,"Object Tracking: A Survey" inACM Journal of Computing Surveys," Vol. 38, No. 4, 2006.
2. PritiP.Kuralkar, Prof. V.T.Gaikwad," Human Object Trackingusing Background Subtraction and Shadow Removal Techniques," inInternational Journal of Advanced Research in Computer Science andSoftware Engineering, Vol. 2, No. 3, 2012.
3. Friedman N., Russell, S, "Image segmentation in videosequences: a probabilistic approach", In: Proc. 13th Conf. on Uncertainty inArtificial Intelligence, 1997.
4. C.Stauffer, E.Grimson, "Adaptive background mixture models forreal-time tracking", IEEE International Conference on Compute Vision andPattern Recognition (CVPR), 2:246-252, 1999.
5. L. KoteswaraRao, K. Sivanagi Reddy, K. PradeepVinaik,"Implementation of Object Tracking and Velocity Determination",inInternational Journal of Information Technology and KnowledgeManagement, Vol. 5, No. 1, 2012.

BIOGRAPHY

Sawan Singh is a 3rd year student of B.Tech, ECE department at UIET, CSJM university, Kanpur. Sawan has completed many successful project based on human and machine integration. His areas of interest are signal processing, human machine interface, gesture control e.t.c.