Real Time Implementation of Surf Based Target Tracking Algorithm

V. Kamatchi Sundari[1] M. Manikandan[2]

[1]Faculty of Electronics Engineering, Sathyabama University, Chennai, India
[2]Department of Electronics Engineering, Anna University, Chennai, India

Abstract

SURF based Object Tracking has been a well exploited one in the field of research. SURF is an effective key point localization algorithm for successful object detection and tracking. Our paper spotlights on the subject of development and implementation of target tracking in live video on Beagle Bone Black, an open source Linux based embedded platform with Open CV. Incorporating embedded platform with image processing algorithm leads to many real world applications. Two targets are considered in this work. Our experimental results illustrate that our proposed system implements a real time target tracking with sturdiness against variations in target elevation, rotation and also the distance between target and camera.

Keywords: Beagle Bone Black; Embedded Platform; Open CV; SURF; Target Tracking.

I. INTRODUCTION

In recent years the necessity of detection followed by tracking of moving object(s) is increasing a lot and becomes more popular in the field of computer vision. The scene taken in one angle with one scale differs with another angle with same scale. Also looks differently in the same angle with another scale. So finding out the similarities between two images of the same object or same scene becomes an exigent job. The key point behind target tracking is image registration. Image registration algorithm [1] fails in situations like target scale change, target rotation and variation in target elevation angle. Earlier various feature point detector algorithms have been proposed [2-9] for descriptor calculations. SURF descriptors are the one used to produce good results in target tracking and also to overcome all the above mentioned issues SURF is used in our work with a slight change. Instead of three scales and three octaves [10], 4 octaves and 4 scales in each octave are used in our work to get better performance. Brief description of SURF algorithm is given in section 2. Experimental setup is discussed in section 3. Experimental results are given in section 4 and finally conclusive remarks are addressed at the tip of this paper.

II. SURF (Speeded Up Robust Features)

SURF (Speeded Up Robust Features) modified version of SIFT [11]. SURF is faster and robust than SIFT.SURF employs integral images and scale space construction to produce keypoints and descriptors in an effective manner. SURF mainly uses two stages namely feature point detection and feature point description [6]. In SURF evaluation of integral images enable the fast computation procedure using a box filter as is independent of the size of the filter. Eigen values of the Hessian matrix are used to detect the keypoints. In this way SURF constructs its scale space by keeping the image size the same and varies the filter size only which leads to invariance to scale change, angle of elevation and angle of rotation.

A. Steps involved in SURF Feature Matching Algorithm

- Get the Input Video through webcam.
- Interest Point calculation

  Find interest points in the image using integral images with which computation of intensities for any rectangle within the image using box filter.
- Feature Point Detection

It can be done with the help of Hessian Matrix.

Give point is X i.e X=(x,y), the Hessian matrix[12] in X at scale σ is well defined as

$$H(p,\sigma) = \begin{bmatrix} L_{xx(p,\sigma)} & L_{xy(p,\sigma)} \\ L_{xy\ (p,\sigma)} & L_{yy(p,\sigma)} \end{bmatrix} \quad (1)$$

where H is Hessian matrix , L is the image established via convolving the image of input with the second order derivative of guassian on point x. Find eigen values (λ1 &λ2) of Hessian matrix. Corner can easily been identified with large eigen values (both), a step edge can be identified with one large and one small eigen value and two small eigen values give details about low contrast region. At this stage thresholding with a value 10 is applied to detect major feature points and to discard inefficient keypoints.

- Find Major Interest Points in scale space

Normal 3x3 non-maximal suppression is carried out within the same blob response map and also a non-maximal suppression with the blob response map over the image and below the image in each scale space for each octave should be done. Herbert bay used only three scales and three octaves whereas 4 octaves and 4 scales in each octave are used in our work for better performance.

- Find Feature Direction

Haar Transform is used to assess the primary direction of the feature.

Computation of rotation can be calculated by looking at pixels in a circle of 6*σ radius and thereby choosing the direction of maximum total weight.

- Computation of Feature Point Descriptor

Construct square descriptor window with a size of 20*σ centered on all interest point and orientation based on the derived rotation and then divide the descriptor window into 4 x 4 sub-regions and each sub-region is of 5*σ square. Compute Haar wavelets of size 2*σ for regularly spaced points in each sub-region.

Compute 4 values such as Sum of dx, Sum of dy, Sum of abs(dx) and Sum of abs(dy) for all the 16 sub-regions.

- Feature Matching

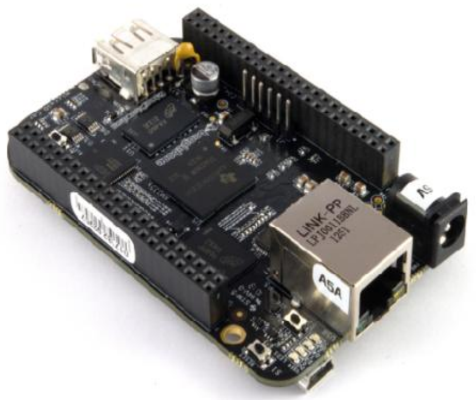
Feature matching can be done with the help of nearest neighbor on the Euclidean scale.

If two or more feature points having the similar feature values then excluded those to avoid incorrect matching of keypoints.

III. EXPERIMENTAL SETUP

Figure1shows the implementation setup used in our work and figure 2 shows the Beagle board used and its special features.

Fig.1. Implementation Setup



Special Features

Processor Speed 1Ghz

2GB NAND Flash Memory

512MB DDR3 RAM

microHDMI(audio and video)

Fig. 2.Beagle Board and its Features

Two objects (robot) are used in our experiment each connected with Arduino board to enable motion in them. Dimensions of the robot used are 20cm x 15cm x10cm (length, breath and height).Various stickers are pasted on the objects to get more perfect interest points during operation. Matlab is used to acquire real time video through webcam. Reference object is taken in the first frame of the test video.

SURF algorithm for tracking is written in Open CV and all operations are performed in Beagle Board and finally results with tracked objects are displayed in the monitor through Matlab.

IV. RESULTS AND DISCUSSION

Tests are carried out to evaluate the performance of our work. In this regard, capture the video with objects in different elevations, angle of rotation and by varying the distance between the object and camera. From our results it is found that tracking speed is improved a lot through Beagle Board when compared to Matlab without sacrificing the performance. In all our experiments changes in the position of the object is done in every 10 seconds. Computation of percentage of correctly matching (POM) can be found by the ratio of total number of correctly matched points in 10 seconds to that of the total number of frames in 10 seconds.

A. Change in mutual distance between camera and object

As the distance between the object and camera increases, the size of the object starts to decrease. In that condition only bigger keypoints can easily been identified and smaller keypoints begins to disappear gradually.

Table 1. Evaluation at various heights

Distance in cms	Percentage of Matching	No of frames processed per sec
5	93	3.8044
10	87	3.9699
15	89	3.9827
20	90	4.0712
25	85	4.1134
30	81	4.1550
35	72	4.2230
40	78	4.2426
45	72	4.2440
50	62	4.310
55	61	4.2932
60	57	4.3476
65	52	4.3524
70	32	4.3555
75	15	4.3714
80	10	4.4091

It is found that the performance is good when then scale is small around 2-3 scale. As the scale increase beyond 4 the repeatability decrease below 50%.

B. Change in angle of elevation

It is found that the object recognition and tracking is good when the elevated angle is below 45 degrees. After 45 degrees elevation, the elevated object varies largely compared to the reference image and so the performance is not fair for above 45 degrees of elevated angle.

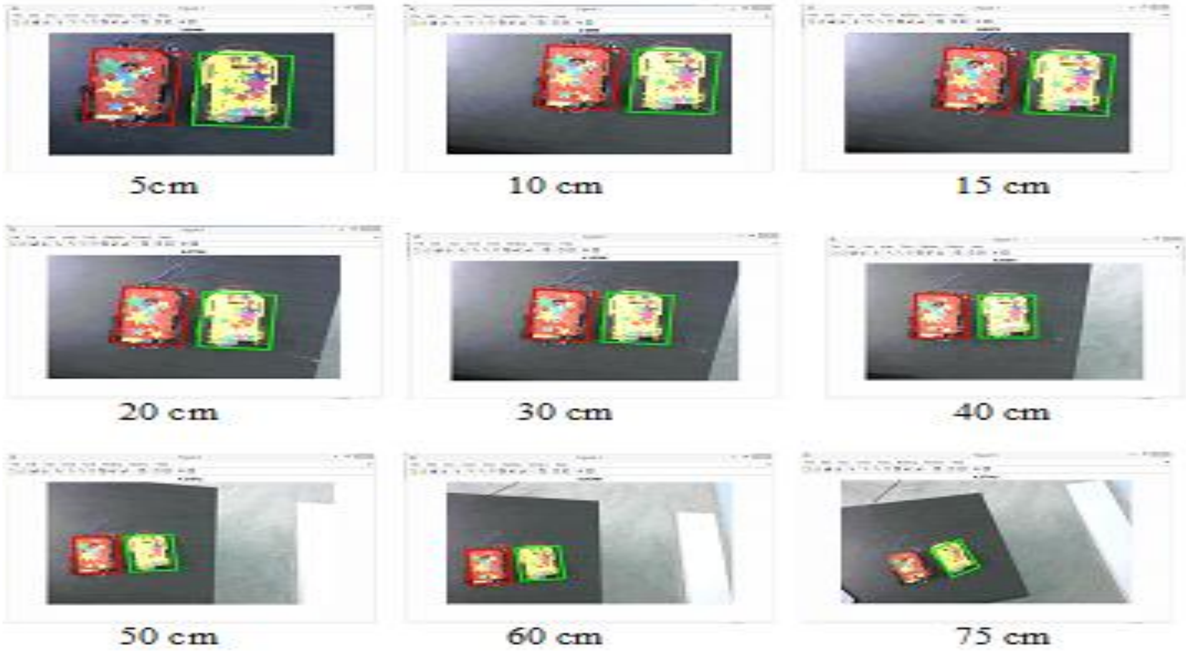


Fig. 3. Snap shots of tracking results at various height

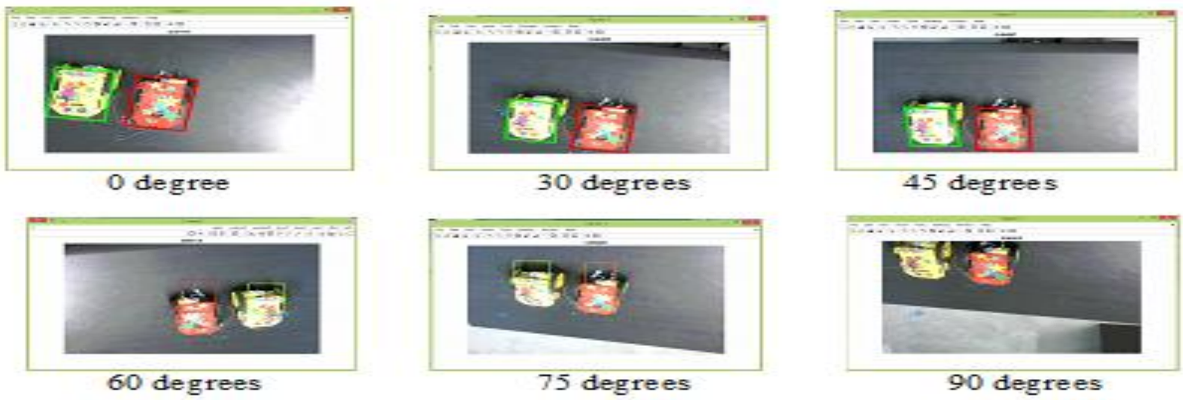


Fig. 4. Snap shots of tracking results at various elevation angle

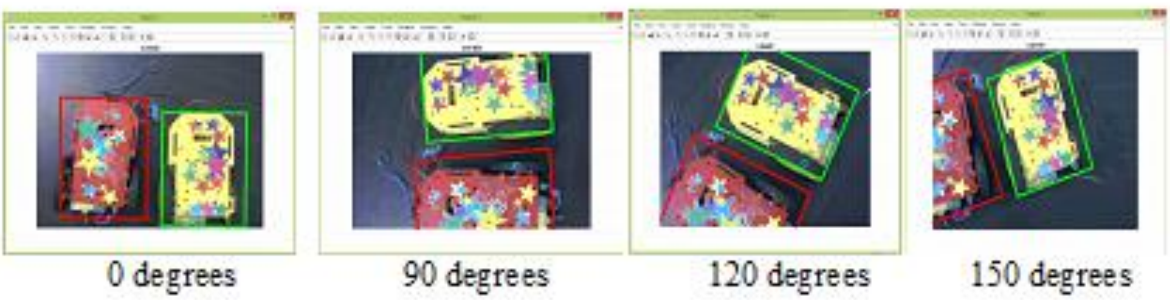


Fig.5. Snap shots of tracking results at various angle of rotation

Table 2. Evaluation at various angle of elevations

Elevation Angle in degrees	Percentage of Matching	No of frames processed per sec
0	95	5.4367
15	91	5.4195
30	78	5.4945
45	22	5.4988
60	16	5.7100
75	10	5.8580
90	0	5.8681

A. Change in angle of rotation

To check the clearly known fact that SURF algorithm is rotation invariant, here camera is kept at a height of 10cm above the objects.

Table 3. Evaluation at various angle of rotations

Angle of Rotation in degrees	Percentage of Matching	No of frames processed per sec
0	93	3.338
30	91	3.447
60	90	3.526
90	95	3.574
120	89	3.599
150	92	3.621
180	85	3.621

From the results it is found that performance is not affected vastly due to change in rotation.

V. CONCLUSION

In this paper a real time tracking system using SURF and Open CV is implemented in Beagle board and tested. It uses SURF to identify feature points and updated adaptively for tracking process. The system is tested with different cases and is found that our work produces promising results. Further, the performance of our system can be enhanced with the help of high resolution camera.

REFERENCES

[1]. Barbara Zitova, Jan Flusser (2003), “Image Registration methods: A Survey”.

[2]. G. David Lowe (1997), “Object recognition from local scale-invariant features”, Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.

[3]. K. Mikolajczyk and C. Schmid (2002), “An affine invariant interest point detector”,, Proc. European Conference on Computer Vision,

[4]. pp. 128-142.

[5]. J. Matas, O. Chum, M. Urban and T. Padjla(2002), “ Robust wide baseline stereo from maximally stable external regions”, Proc. British Machine Vision Conference, Vol. 1, pp. 384-393.

[6]. Mikolajczyk .K, Zisserman.A, and Schmid .C (2003), “Shape Recognition with Edge-Based Features,” Proc. British Machine Vision Conf., pp. 779-788.

[7]. H. Bay, T. Tuytelaars and L. Van Gool(2006), “ SURF: Speeded Up Robust Features”, Proc. European Conference on Computer Vision, Vol. 110, pp. 407-417.

[8]. F. Schaffalitzky and A. Zisserman (2002), “Multi view image matching for unordered image sets”, Proc. European Conference on Computer Vision, Vol. 1,pp. 414-431.

[9]. David .P and DeMenthon .D (2005) “Object Recognition in High Clutter Images Using Line Features,” Proc. Int’l Conf. Computer Vision, pp. 1581-1588.

[10]. Zhang. J M. Marszalek .M, Lazebnik .S, and Schmid .C (2007) “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study,” International Journal of Computer Vision, vol. 73, no. 2, pp. 213-238.

[11]. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool(2008), “SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), pp. 346-359.

[12]. D. Lowe(2004), “ Distinctive Image features from scale invariant keypoints”, International journal of Computer Vision, Vol. 60, pp. 91-110.

[13]. R.Gonzalaz, R Woods (2002), “Digital Image Processing”, New Jersey: Prentice –Hall.

[14]. www.beagleboard.org/BLACK

[15]. G. Bradski & A. Kaehler(2008), “ Learning Open CV”, OReilly Publications.