Performance Analysis of Metric Threshold in SURF for Object Tracking

Ari Putra Nugraha*, Suryo Adhi Wibowo*, and Nur Andini*
*School of Electrical Engineering, Telkom University
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, 40257 INDONESIA
E-mail:ariputranugraha@student.telkomuniversity.ac.id,
{suryoadhiwibowo, nurandini}@telkomuniversity.ac.id

***Abstract*—Object tracking has a lot of progress every year and gives something new, so the trackers and method itself is getting better. Many researchers are engaged in one of the fields of computer vision to provide good benefits for human life in the field of Internet of Things. Feature extraction is needed in object tracking processes. One of the scale and rotation-invariant local feature extraction methods, namely Speeded-Up Robust Feature (SURF). In implementing it on object tracking, SURF will extract features from two frames and match them. Then, the Random Sample Consensus (RANSAC) as a rejection correspondence using an inlier on the features obtained and then performs the estimation transformation which is applied to the input match the reference image. In this paper, we analyze one of the parameters of SURF, namely Metric Threshold as a parameter that determines the strongest feature threshold. From the evaluation results, it was found that the default parameters gave non-optimal results.**

I. Introduction

Feature extraction is an important step or method in object tracking and is used in almost all machine learning algorithms. In relation to object tracking, optimized feature extraction is the key to producing good model construction. In comparing two images for object tracking, a connection between the features of the two images is needed and it is very important to choose a good feature point to increase robustness [1]. SURF is one of the local feature detectors and descriptors that can be used in tracking objects. The output of SURF itself is a scale and rotation invariant keypoint that has a high tolerance to noise and light [2]. However, SURF which is implemented on object tracking often fails due to the lack of suitable features. In addition, the metric threshold is one of the parameters in SURF that affects the number of points that appear. The greater the value of the parameters, the smaller the tolerance of features and the fewer points. If the the matched points between two images are four or less points, The probability is high that the tracking will fail.

In this paper, we analyze the metric threshold parameters that affects in tracking the object by considering the wrong feature points.

II. System Model

A. Speeded Up Robust Feature

SURF uses Fast Hessian Detector to find interest points and calculate its Hessian Matrix[2]. SURF in obtaining a scale using Hessian determinant in each $k = (x, y)$ for image I. Matrix of Hessian $H(k, \sigma)$ with point k and scale σ, is defines as follows:

$$H(k,\sigma) = \begin{pmatrix} L_{xx}(k,\sigma) & L_{xy}(k,\sigma) \\ L_{yx}(k,\sigma) & L_{yy}(k,\sigma) \end{pmatrix}, \qquad (1)$$

where $L_{xx}(k,\sigma)$, $L_{xy}(k,\sigma)$, $L_{yx}(k,\sigma)$, and $L_{yy}(k,\sigma)$ is respectively second-order derivate of Gaussian with image $I(x, y)$.

Herbert Bay et al. explained that SURF fixes reproducible orientation according to information from circular regions around points of interest. Then when features are extracted, SURF builds a square region centered around interest points and oriented along orientation selected [3].

B. Random Sample Consensus

RANSAC is an algorithm for matching features that match incorrectly or outliers and only takes interest points or inliers in their calculations. Then, polygons drawn in inlier space as a result of the approximate location of the object [4].

III. Experimental Result

We evaluate the object tracker based on surf using three datasets from the OTB-50 [5], namely dragonBaby, Liquor, and Coupon. To find out how many frames are not detected, we select several metric threshold parameters to be tested, they are 100, 500, 1000 (default), 2000, and 2500 as we can see in Table I, II, III.

This system is tested on Intel Core i7 laptop with 8GB RAM and nVidia GeForce GT 755M as Discrete Graphic. Of course, computation time will be different than using another laptop. In addition, there is a difference in computing time between the metric threshold parameters tested.

TABLE I: Evaluation of how many frames the object in it cannot be detected for sequence *dragonBaby* that consists of 113 frames.

Metric Threshold	Undetected Frame(s)	Computation Time (sec)
100	7	9.52
500	7	8.04
1000	10	6.96
2000	34	4.52
2500	80	1.72

TABLE II: Evaluation of how many frames the object in it cannot be detected for sequence *Liquor* that consists of 1741 frames.

Metric Threshold	Undetected Frame(s)	Computation Time (sec)
100	34	224.06
500	91	192.8
1000	297	144.72
2000	128	112.9
2500	379	101.28

TABLE III: Evaluation of how many frames the object in it cannot be detected for sequence *Coupon* that consists of 140 frames.

Metric Threshold	Undetected Frame(s)	Computation Time (sec)
100	10	4.80
500	12	4.16
1000	22	3.03
2000	25	3.50
2500	26	3.37

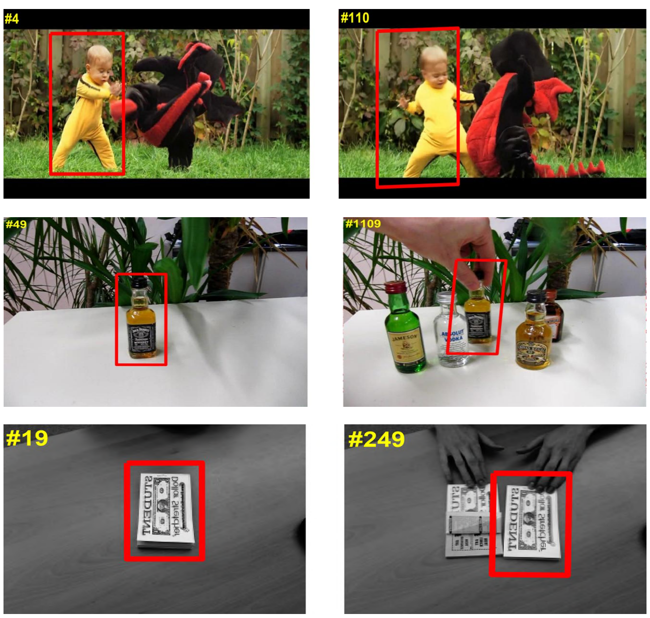

Fig. 1: Tracking result for tested sequence *dragonBaby*(Top), *Liquor*(Middle), and *Coupon*(Bottom)

Keep in mind that the output of this tracker is a polygon as it is shown in Fig 1, not a bounding box. Therefore, the shape of the output can be a square, parallelogram or rectangle. With this system, tracker can detect rotating or scale-changing objects. Tracker that applies Convolutional Network will have the better accuracy. Futhermore, it can be applied as in Collaborative Learning based on Convolutional Features and Correlation Filter that used collaborative learning and confidence score to predict the target [6]. But the computation time will be slower due to much processes in their system.

In terms of object tracking, other tracker that is based on Complementary Learners used integral image to calculate the color-histogram [7]. But, SURF can handle more problems, such as occlusion. This is already tested in sequence Coupon and the result is shown in Fig 1. But, when working at local-feature, level of invariance is need to be solved and this depends on the estimated geometric deformation. Appearance changes in viewing conditions will affect to how many features are extracted. SURF itself focuses more in scale and image rotations.

IV. Conclusion

In this paper, we evaluate an object tracker based on SURF feature matching which is scale-invariant and rotate-invariant. SURF looks for keypoints on each image and then matches the keypoint between 2 images. The process runs until the end of the frame, but often in the process there is no matching keypoint so that an error can occurs and in the frame concerned, the object will not be detected. The number of undetectable frames can also be affected by the metric threshold parameter because the smaller the parameter value, the greater the tolerance to the strongest feature which causes more keypoint output.

Experimental results on the test sequences indicate that the output of the system is a polygon line, but can be converted into a bounding box. This method is more applicable due to its role as feature extractor and descriptor. Some trackers could use it to achieve robustness in scale and rotation, so it is expendable for invariant region. Second, SURF is out-standing in terms of speed. Improvements in SURF make it better in extracting the features.

References

[1] Y. Sakai, T. Oda, M. Ikeda, and L. Barolli, “An object tracking system based on sift and surf feature extraction methods,” in *2015 18th International Conference on Network-Based Information Systems*, Sept 2015, pp. 561–565.

[2] D. Zhou and D. Hu, “A robust object tracking algorithm based on surf,” in *2013 International Conference on Wireless Communications and Signal Processing*, Oct 2013, pp. 1–5.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[4] S. Kole, C. Agarwal, T. Gupta, and S. Singh, “Surf and ransac: A conglomerative approach to object recognition,” *International Journal of Computer Applications*, vol. 109, no. 4, 2015.

[5] Y. Wu, J. Lim, and M. Yang, “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, Sept 2015.

[6] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, “Collaborative learning based on convolutional features and correlation filter for visual tracking,” *International Journal of Control, Automation and Systems*, vol. 16, no. 1, pp. 335–349, 2018.

[7] ——, “Visual tracking based on complementary learners with distractor handling,” *Mathematical Problems in Engineering*, vol. 2017, 2017.