TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 268– 274

Implementation of Feature Extraction and Improved Tracking Algorithm for Real Time Tracking of Multiple Moving Objects

Hyochang Ahn[1], Han-Jin Cho[1], June-Hwan Lee[1*]

[1]Department of Energy IT, Far East University, 27601, Republic of Korea
Corresponding Author : June-Hwan Lee, Email: rainbow@kdu.ac.kr[*1]

Article Info
Volume 81
Page Number: 268 - 274
Publication Issue:
November-December 2019

Article History
Article Received: 3 Jan 2019
Revised: 25 March 2019
Accepted: 28 July 2019
Publication: 22 November 2019

Abstract

Recently, a real-time monitoring system needs to have the function of detecting and tracking the existence of an automatically moving object. These computer vision systems can be used in many fields because they can replace human roles. We propose a method to track objects by merging SURF, CAMShift, and Optical Flow. Through the SURF method, a search window is created for the region of interest for the object to be tracked, and the object is tracked through a combination of the CAMShift method and the optical flow. We also propose using Optical Flow to predict the position and velocity of a fast moving object. Experimental results show that the proposed method shows good results in terms of both accuracy and speed.

Keywords: *CAMShift, Optical Flow, SURF, Object Detection, Object Tracking*

I. Introduction

Moving objects, motion tracking and human detection are widely used in video conferencing systems and real-time surveillance systems[1,2]. Among them, the real-time monitoring system is mainly applied to a computer vision system which can substitute human role because it has the function of automatically detecting the presence of objects in an environment where many moving objects do not appear[2,3]. This research has been used especially in security surveillance field, weather observation system, intelligent traffic control system, military field and many other fields. Currently, various studies are being conducted for high accuracy and fast processing for object recognition and tracking[3]. As information technology develops, there is an increasing need for security monitoring for theft prevention and information leakage. Accordingly, tracking and detection of moving objects is becoming an important technology in security related surveillance field[1,3].

In order to track moving objects in each frame, frame analysis, real-time monitoring analysis, and prediction of instantaneous changes of moving objects are required for successive video images[4,5]. By analyzing the frame of the image, it detects the object to be tracked and calculates the position change of the object, extracts information such as the moving

November-December 2019
ISSN: 0193-4120 Page No. 268– 274

direction and speed of the moving object, and continuously tracks the moving object based on the extracted information[5]. However, in the tracking of objects, there are various changes according to the movement, change, location and illumination of the object, so there is a problem that the object is determined and the ROI is reassigned to adapt to the surrounding environment.

In this paper, we propose a method to track target objects using CAMShift and Optical Flow. The CAMShift algorithm is vulnerable to illumination, brightness and speed because it is based on a color space model. In order to overcome this difference of illumination condition, the search window of CAMShift is set to the SURF algorithm. You can also use the optical flow to predict the position and velocity of a fast moving object, so you can quickly adjust the search window in CAMShift.

The rest of this paper is organized as follows. In Section 2, we briefly introduce a method for detecting objects in an image. In Section 3, we explain how to use CAMShift and Optical Flow to track objects. Section 4 presents the experiments and results for the performance evaluation of the proposed method. Finally, Chapter 5 presents conclusions and future work.

II. Material and Methods

In this paper, we propose a fusion method of SURF, CAMShift, and Optical Flow for object tracking system with improved performance. Even if the object moves quickly, a method of recognizing the object without external influences is needed by quickly obtaining the detection speed early. However, in the case of the SURF algorithm, there are disadvantages that it cannot utilize many useful features given in the color space by using only information in the gray space.

In addition, since the computation complexity is high for a specific object detection process, detection of the detection in the real-time image is limited, and there is a difficulty in detecting an object in which the object moves fast. In order to improve this, in this paper, we use the area of interest obtained by the SURF algorithm only for the designation of the search window, and the real-time object tracking improves the overall operation speed by using the CAMShift algorithm.

In addition, when the objects are tracked using CAMShift, fusion of object movement information obtained by Optical Flow to the generated search window can improve the tracking accuracy compared to the search windows generated only by the existing CAMShift

SURF(Speed Up Robust Features)

It is not easy to construct a descriptor for discriminating all the pixels in an image or to find corresponding points by comparing descriptors of all pixels[6,7]. Therefore, there is a need for an algorithm that finds feature points that are robust to environmental changes and generates descriptors that can distinguish them from other feature points. In the proposed extended SURF algorithm, feature points are extracted using a high-speed Hessian detector. A high-speed Hessian detector can be generated using the integral image and approximated Hessian detector[6,7].

The integral image is the sum of all the pixel values in the rectangular area from the origin to the position of each pixel[7,8]. Therefore, once the integral image is generated, the sum of all the pixel values in the specified rectangle can be obtained through four operations of any size rectangular area. Formula (1) (x, y) represents the process of calculating the integral image value $II(x, y)$ at a position where $I(i, j)$denotes

November-December 2019
ISSN: 0193-4120 Page No. 268– 274

a pixel value of position (i, j) in the original image.

$$II(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) (1)$$

And, the Hessian detector is a feature extraction algorithm based on the Hessian determinant as defined in equation (2) and shows good performance in velocity and accuracy.

$$H(x, y, \sigma) = \begin{bmatrix} LI_{xx}(x, y, , \sigma) & LI_{xy}(x, y, , \sigma) \\ LI_{xy}(x, y, , \sigma) & LI_{yy}(x, y, , \sigma) \end{bmatrix} (2)$$

In formula (2), $LI_{xx}(x, y, \sigma)$ is (x, y) of a Gaussian with the input image and σ the variance of the position x direction second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$. The remaining $LI_{xy}(x, y, \sigma)$ and $LI_{yy}(x, y, \sigma)$ denote the derivative in the xy direction and the second-differentiated Gaussian filter and convolution in the y direction. The approximated Hessian detector is a method that uses an approximated Hessian determinant using a square filter instead of using the Hessian determinant. In order to obtain the feature which is invariant to the scale, the feature is extracted by changing the size of the square filter without using the scaled image. As a result, by using the integral image obtained before calculating the convolution of the quadrature filter, the Hessian determinant is quickly constructed and the feature point is found irrespective of the size of the rectangular area.

CAMShift(Continuously Adaptive Mean Shift)

The CAMShift algorithm is an algorithm that improves the average moving algorithm[9,10]. Unlike the average moving algorithm in which the size of the search window is fixed, the CAMShift algorithm uses the technique of adjusting the size of the search window by itself. Using the distribution of the hue values of the detected object region, the position to be changed is predicted and the object is traced by searching the detected center of the number. Generally, since the change of illumination in the image is the largest in luminance, using a relatively hue value is less sensitive to illumination. The general CAMShift algorithm operates in the following order[9,10].

Table 1: The general CAMShift algorithm

1) Set the region of interest of the probability distribution image over the entire image.
2) Select the initial position of the search window of the average movement algorithm. The selected location is the target distribution for tracking.
3) Calculate the color probability distribution in the search window of the average movement algorithm.
4) Repeat the average moving algorithm to find the center of probability image. The zero moment and the center position are stored.
5) In the following frame, the center of the search window is changed to the center found in 4), and the size of the search window is set as a function of the moment. Repeat steps 3) to 5).

Any method can be used if the probability distribution image only has a value indicating the probability of the pixels included in the target. However, in general, the probability distribution image is obtained by histogram back projection, and the back projection of the target histogram of a successive frame generates a probability image by the following method.

The n image pixel positions using the histogram with m of bin are $\{x_i\}_{i=1,......,n}$. The histogram is a function that maps the position x_i^* of the pixel $\{\hat{q}\}_{u=1,......,m}$ to the bin of the histogram $c: R^2 \rightarrow \{1 \dots \dots m\}$, the histogram is calculated as shown in the following equation (3).

November-December 2019
ISSN: 0193-4120 Page No. 268– 274

$$\widehat{q_u} = \sum_{i=1}^{n} \delta[c(x_i^*) - u] \qquad (3)$$

In addition, all histogram bin values are scaled from $[0, max(q)]$to the new range $[0, 255]$ in the pixel range of each 2 dimension probability distribution image by the following equation (4).

$$\left\{\widehat{p_u} = min\left(\frac{255}{max\,(\hat{q})}\widehat{q_u}, 255\right)\right\}_{u=1......m} \qquad (4)$$

In addition, the center of color probability distribution of the search window obtained as a result of (3) of the operation sequence of the CAMShift algorithm described above, and the size and angle of the target can be obtained using the moment. Assuming that the intensity of the probability image at the (x, y)position of the search window is $I(x, y)$, each moment can be obtained as shown in Equation (5) below.

$$M_{00} = \sum_x \sum_y I(x, y)$$

$$M_{10} = \sum_x \sum_y xI(x, y),\ M_{01} = xyyI(x,y) \qquad (5)$$

$$M_{20} = \sum_x \sum_y x^2(x, y), \qquad M_{02} = xyy2(x,y)$$

The center of probability distribution (x_c, y_c) is shown in equation (6).

$$x_c = \frac{M_{10}}{M_{00}}, \quad x_c = \frac{M_{01}}{M_{00}} \qquad (6)$$

In order to simplify the formula for obtaining the magnitude and angle of the probability distribution, the three variables a, b, c are defined by the following equation (7).

$$a = \frac{M_{20}}{M_{00}} - x_c^2$$

$$b = 2\left[\frac{M_{11}}{M_{00}} - x_c y_c\right] \qquad (7)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2$$

The size and angle of the horizontal(h) and vertical(v) of the search window by the three variables a, b, c are defined by the following equation (8).

$$h = \sqrt{\frac{(a+b)+\sqrt{b^2+(a-c)^2}}{2}}$$

$$\theta = \frac{1}{2}tan^{-1}\left(\frac{b}{a-c}\right)^2 \qquad (8)$$

Optical Flow

Optical Flow is an algorithm that tracks how an object moves in an image. It detects a feature point with prominent attributes such as corners and textures in the image, and then calculates the movement amount through matching between the previous frame and the next frame to be[11]. The Lucas-Kanade method is the most widely used, with a rare Optical Flow method with low computational complexity[12].

In general, Optical Flows are divided into dense Optical Flows and rare optical flows[11,13]. There is a Lucas-Kanade method of dense Optical Flow, which computes the amount of movement in a way that sets the pixel window in the current frame and finds the closest match to the window set in the next frame. The calculation amount is smaller than the calculation of the entire pixel and the movement cannot be calculated when the motion is larger than the set window due to the use of the local window. In order to solve the disadvantages of the Lucas-Kanade method, a pyramid Lucas-Kanade method which can detect motion larger than a set window by using an input image as an image pyramid has been studied.

Generally, the Lucas-Kanade Optical Flow method assumes the following three assumptions.

1) Brightness constancy

2) Temporal persistence or small movements.

3) Spatial coherence

Among them, the first brightness constancy assumes that the pixel brightness value of the object being tracked is also unchanged with time. Temporal persistence or small movements assume that the motion of the object in the image is not so fast and that the time change is faster than the movement of the object. The third spatial coherence assumes that spatially close points are likely to belong to the same object and have the same movement. By the first hypothesis, the image intensity function $I(x(t), t)$ is defined as the following equation (10).

$$I(x(t), t) \approx I(x(t + dx), t + dt) \qquad (9)$$

Also, by the second assumption, as equation (10)

$$I_x u + I_y v + I_t = 0 \qquad (10)$$

Where $u$is the velocity in the x axis direction, v is the velocity in the y axis direction, and I_t is the first-order partial differential of $I(x(t), t)$. If you use an $n \times$ nwindow, n^2 equation (11) are obtained as

$$\begin{bmatrix} I_x(p_1) I_y(p_1) \\ \dots\dots \\ I_x(p_{n^2}) I_y(p_{n^2}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ \dots\dots \\ I_t(p_{n^2}) \end{bmatrix} \qquad (11)$$

Using the least squares method, we can obtain the values of u and v by solving the above equation (11).

III. Results and Discussion

The environment used for the performance evaluation of the proposed method is Python on Windows 10. In addition, PET2000 dataset is detected by using OpenCV library and tracking algorithm is implemented[14]. Figure 1 and 2 show the running of the proposed method in the experimental environment. In order to measure the performance improvement of our proposed method, we compared three algorithms: SURF tracking, CAMShift tracking, and tracking by the improvement algorithm.

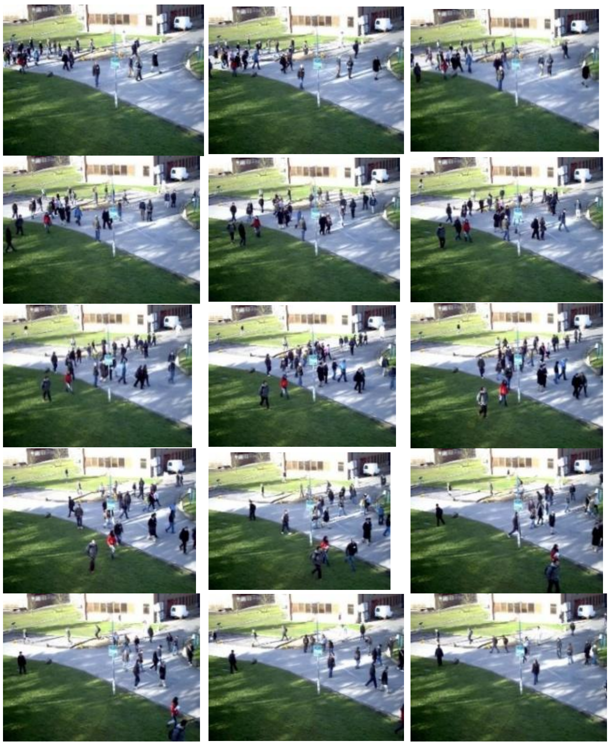

Figure 1: Video Data for Experiments

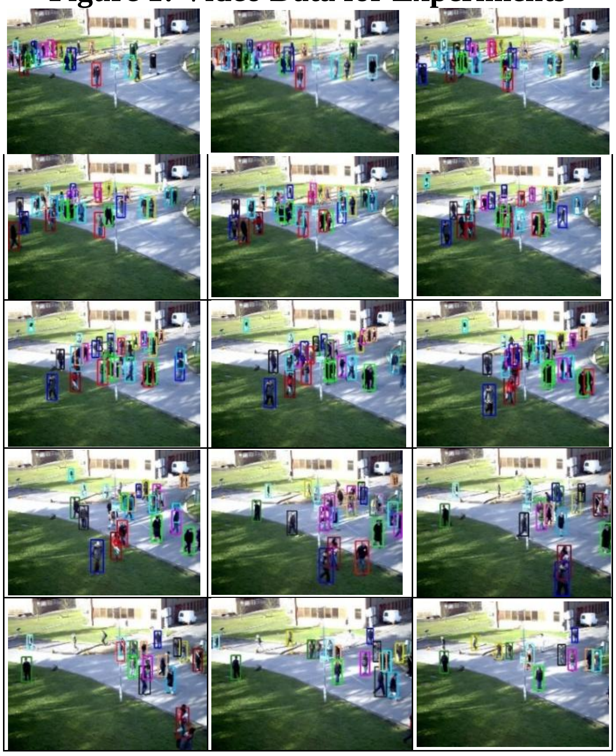

Figure 2: Results for proposed algorithm in Video Data

Table 2:The comparison of objects tracking results

Method	TF (Frame)	TS (Frame)	TF (Frame)	PT (sec)
SURF	200	185	15	13.7
CAMShift	200	135	65	8.1
Proposed Method	200	178	22	10.2

Table 2 shows experimental comparison data for each algorithm. The experimental images were 200 of total frames(TF) of data set provided by PET2000. As shown in the experimental results, the tracking success(TS) rate of the proposed algorithm is 178 frames, CAMShift is 135 frames, and SURF is 185 frames. The tracking failure(TF) rate was 22 frames for the proposed algorithm, 15 frames for the SURF, and 65 frames for the CAMShift. Although the proposed algorithm has a somewhat lower success rate than SURF, it shows a fairly good result compared with CAMShift.

The processing time(PT) of the proposed algorithm is 10.2 seconds, CAMShift is 8.1 seconds, and SURF is 13.7 seconds. The proposed algorithm is 3.5 seconds faster than SURF, but it is 2.1 seconds slower than CAMShift.

Therefore, the proposed method shows good results in terms of both accuracy and speed, so it is possible to track real-time multiple objects

IV. Conclusion

In this paper, we propose a method to track target objects by merging SURF, CAMShift, and Optical Flow. Through the SURF method, a search window is created for the region of interest for the object to be tracked, and the object is tracked through a combination of the CAMShift method and the optical flow. Since the CAMShift algorithm is based on a color space model, the search window of CAMShift is set through the SURF algorithm to solve problems that are vulnerable to light, brightness, and speed. Also, by using Optical Flow to predict the location and velocity of a fast moving object, the search window can be adjusted quickly in CAMShift.

Experimental results show that the success rate of the proposed algorithm is somewhat lower than that of the SURF algorithm, but it is significantly better than CAMShift algorithm. Also, the performance of the proposed algorithm is about 3.5 seconds faster than that of SURF, but it is about 2.1 seconds slower than CAMShift. Therefore, the proposed method shows good results in terms of both accuracy and speed, so it is possible to track real-time multiple objects.

Future research needs to study the surveillance system that can quickly detect multiple objects and predict motion using the proposed method.

References

[1] M. Valera and S. A. Velastin. "Intelligent distributed surveillance systems: a review." IEE Proceedings-Vision, Image and Signal Processing 152(2), 192-204, 2005. https://doi.org/10.1049/ip-vis:20041147

[2] R. T.Collins, A. J.Lipton, T.Kanade, H.Fujiyoshi, D.Duggins, Y.Tsin, and L.Wixson, "A system for video surveillance and monitoring" Robotics Inst., CMU-RI-TR-00-12, VSAM final report, 1-68,2000.

[3] H. Ahn and Y. Lee, "Performance analysis of object recognition and tracking for the use of surveillance system." Journal of Ambient Intelligence and Humanized Computing 7(5), 673-679, 2016. https://doi.org/10.1007/s12652-015-0325-4

November-December 2019
ISSN: 0193-4120 Page No. 268– 274

[4] J. G.Allen, R. YD Xu, and J. S. Jin,"Object tracking using camshift algorithm and multiple quantized feature spaces." Proceedings of the Pan-Sydney area workshop on Visual information processing. Australian Computer Society, Inc., 3-7, 2004.

[5] R.Stolkin , I. Florescu, and G. Kamberov. "An Adaptive Background Model for Camshift with a Moving Camera, 6th Int." Conf. on Advances in Pattern Recognition,261-265, 2007.

[6] H.Bay, A.Ess, T.Tuytelaars, and L.Van Gool, "Speeded-up robust features (SURF)" Computer vision and image understanding, 110(3), 346-359, 2008, https://doi.org/10.1016/j.cviu.2007.09.014

[7] D. N.Ta, W. C.Chen, N.Gelfand, and K.Pulli, "Surftrac: Efficient tracking and continuous object recognition using local feature descriptors." In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2937-2944, 2009, June, https://doi.org/10.1109/CVPR.2009.5206831.

[8] T. M. Barroso, and P. F. Whelan. "Enhancing SURF Feature Matching Using Colour Histograms." 2011 Irish Machine Vision and Image Processing Conference. IEEE, 111-112, 2011, https://doi.org/10.1109/IMVIP.2011.31.

[9] G. Kim, and D. Kang. "Modified CAMshift Algorithm Based on HSV Color Model for Tracking Objects." International Journal of Software Engineering and Its Applications, 9(7), 193-200.2015, https://doi.org/10.14257/ijseia.2015.9.7.20

[10] X. Liu, H. Chu, and P. Li. "Research of the improved camshift tracking algorithm." 2007 International Conference on Mechatronics and Automation. IEEE, 968-972, 2007.https://doi.org/10.14257/10.1109/ICMA.2007.4303678

[11] T. Senst, V.Eiselein, and T.Sikora. "Robust local optical flow for feature tracking." IEEE Transactions on Circuits and Systems for Video Technology 22(9), 1377-1387, 2012, https://doi.org/10.1109/TCSVT.2012.2202070

[12] B. D. Lucas and T.Kanade. "An iterative image registration technique with an application to stereo vision.", Proceedings DARPA Image Understanding Workshop, 121-130, 1981, April.

[13] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. "Performance of optical flow techniques.' International journal of computer vision, 12(1), 43-77, 1994, https://doi.org/10.1007/BF01420984

[14] A. Ellis, A.Shahrokni, and J. M. Ferryman. "Pets2009 and winter-pets 2009 results: A combined evaluation." 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. IEEE, 1-8, 2009, https://doi.org/10.1109/PETS-WINTER.2009.5399728