

## Real-time Target Human Tracking using Camshift and LucasKanade Optical Flow Algorithm

Van-Truong Nguyen\*, Anh-Tu Nguyen, Viet-Thang Nguyen, Huy-Anh Bui, Xuan-Thuan Nguyen

Faculty of Mechanical Engineering, Hanoi University of Industry, Hanoi, SO298, Vietnam

### ARTICLE INFO

Article history:

Received: 31 January, 2021

Accepted: 24 March, 2021

Online: 10 April, 2021

Keywords:

Human tracking

Camshift algorithm

Lucas-Kanade Optical Flow

Kalman Filter

### ABSTRACT

*In this paper, a novel is proposed for real-time tracking human targets in cases of high influence from complexity environment with a normal camera. Firstly, based on Oriented FAST and Rotated BRIEF features, the Lucas-Kanade Optical Flow algorithm is used to track reliable keypoints. This method represents a valuable performance to decline the effect of the illumination or displacement of human targets. Secondly, the area of the human target in the frame is determined more precise by using the Camshift algorithm. Compared to the existing approaches, the proposed method has some merits to some extents including rapid calculations in implementation, high accuracy in case of similar objects detection, the ability to deploy easily on mobile devices. Finally, the effectiveness of the proposed tracking algorithm is demonstrated via experimental results.*

### 1. Introduction

These days, with the continuous development of high precision, cost-effective computer imaging hardware equipment and corresponding auxiliary detection algorithm, real-time target human tracking technology [1]-[8] has become a research hotspot in various fields such as interactive video games, military, robotic, etc. Human tracking is the process of locating moving targets by performing tracking algorithms and analyzing that trajectory in many applications over sequential video frames [9]-[12]. The models of the target could be a single human or multiple humans based on different applications. However, human tracking is a difficult task with many challenges such as the system needs to be fast enough to carry out real-time applications or operate in different environments with high accuracy. The impact of external factors affects the accuracy of the algorithm such as moving obstacles, illuminations, or similar targets, etc. Hence, researches of higher accuracy and better computer vision detection algorithm are still developed.

With the development of technology, various methodologies have been proposed to solve real-time human tracking. In the literature review, the Kalman Filter algorithm [13] could only be used in a linear system in case of mimic real scenarios. In [14] - [16], Laser Range Finder to track human is proposed to determine the human' leg and analyze the human tracking motion. However, the measurement of this method is limited in case of long distance between human' leg and sensor. The multi-camera systems [17] -

[21] are also designed to track human, but those approaches only process in a small area which installs cameras. The hand-held cameras system is equipped for the sports player in [22] to synthesize a stroboscopic image of a moving target. Nevertheless, its computational speed and accuracy could not be optimal in case of the elaborate environments. In [23]-[25], Particle filter-based vehicle tracking via HOG features is very robust when fast-moving human target but this algorithm is hard to integrate with a mobile system. In [26] and [27], the Camshift algorithm is well-known as popular tracking human method. The advantages of this methodology are low-cost computation, easy to manipulate, etc. The Camshift algorithm follows the track of the targets based on histogram back projection. The stability of this method is impacted by color, illumination, and noise. Hence, this method suffers from series of errors in case of similar color background.

In light of the remarkable importance and advantages previously mentioned, a real-time tracking human system is proposed for normal cameras under the circumstances of high influence from complicated environments. Lucas-Kanade Optical Flow Algorithm is a well know method for tracking humans for a decade [28]-[30]. The method is designed based on the combination between Camshift algorithm and Lucas-Kanade Optical Flow Algorithm (LK-OFA) [31], [32] with Oriented FAST and Rotated BRIEF features [33]. Compared to the existing works, the proposed approach has several contributions as follows:

(1) The reliable keypoints which are less affected by illumination or displacement of the human targets are tracked based on the LK-OFA features. It is noted that optical follow algorithm is very good at tracking points in the next frame from

\*Corresponding Author: Van-Truong Nguyen, Hanoi, Vietnam, [nguyenvantruong@hau.edu.vn](mailto:nguyenvantruong@hau.edu.vn)

[www.astesj.com](http://www.astesj.com)

<https://dx.doi.org/10.25046/aj0602103>

the locations of them in the previous frame. The Camshift algorithm is carried out to determine the area of the human target with similar appearances in the frames more precisely.

(2) Since the proposed method reduces number of loops that need to converge the center of the search window and centroid of human target, it is easy to implement on mobile devices.

(3) The designed system generates a small amount of data to be processed; therefore, this could be simply done in real-time, and a powerful computer is not required.

The rest of the paper is organized as follows. The next section analyses the detailed structure of the proposed approach. Section 3 summarizes the experiment platform and its result, followed by the brief conclusions and the outlook in section 4.

## 2. Proposed Methodology

In this section, a short overview of the human tracking algorithm is illustrated in the flowchart as Fig.1. To begin with, for

choosing the tracking target person, a region of interest (ROI) is created manually. After that, HSV histogram of ROI is computed to prepare for CAMSHIFT later. At the following stage, the system finds ORB features in both ROI and initial frame. In the subsequent step, 50 best keypoints are chosen from best matched features based on Bruce-Force matcher. Those keypoints are traced by using Optical flow with Lucas-Kanade algorithm. At the next stage, the failed tracked points are totally removed by utilizing sum square of difference (SSD) and Backward tracking method. If the number of the keypoints is smaller than three, the process comes back to the finding ORB features step. In contrast, the process continues with good keypoints left, the center of them is determined using first and zeros moment. This center is used to create the initial search window of Camshift. By applying Camshift algorithm, the area of the human target can be found. Finally, the accuracy of tracking human's center is improved by Kalman filter.

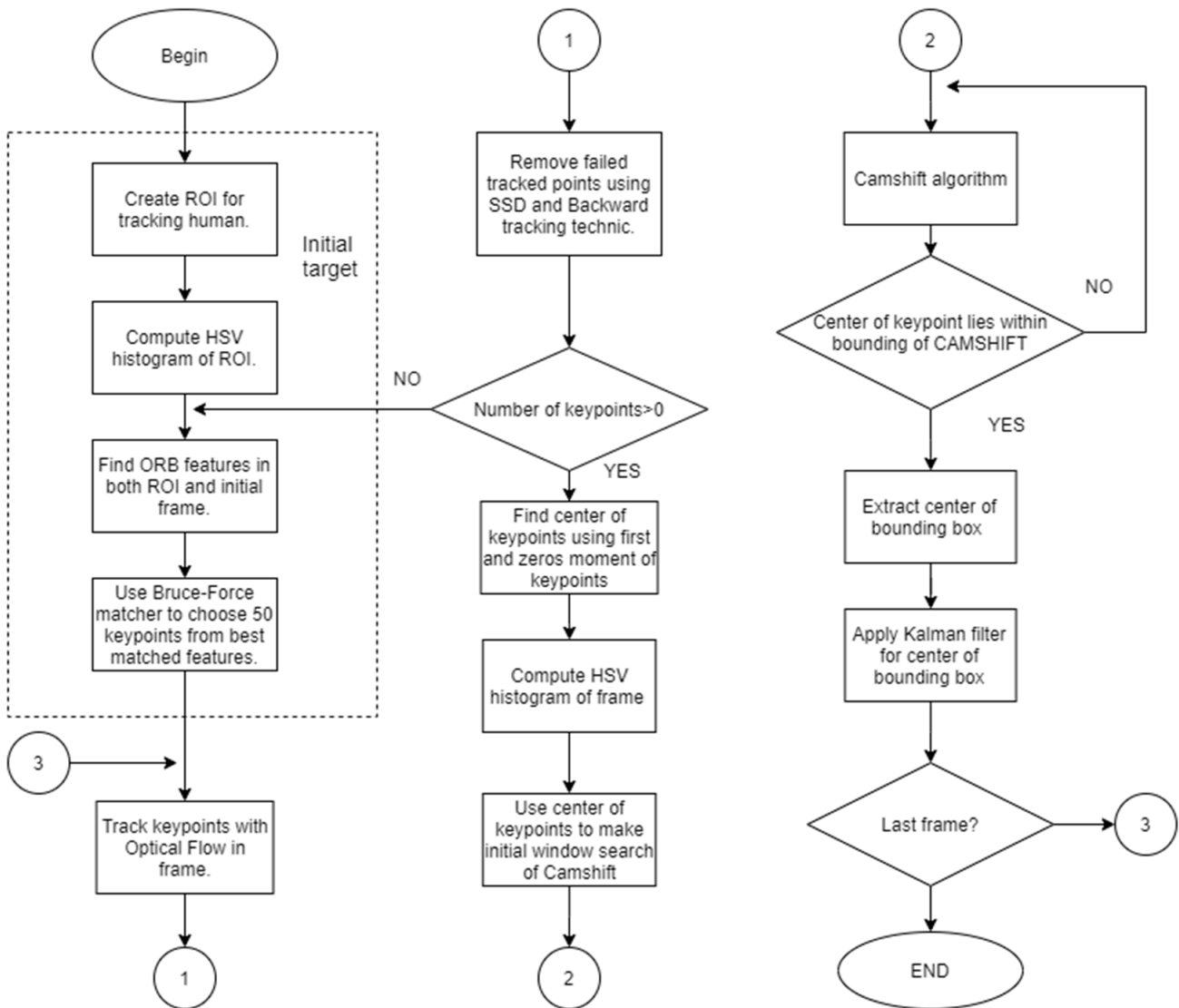


Figure 1: Flow diagram of system

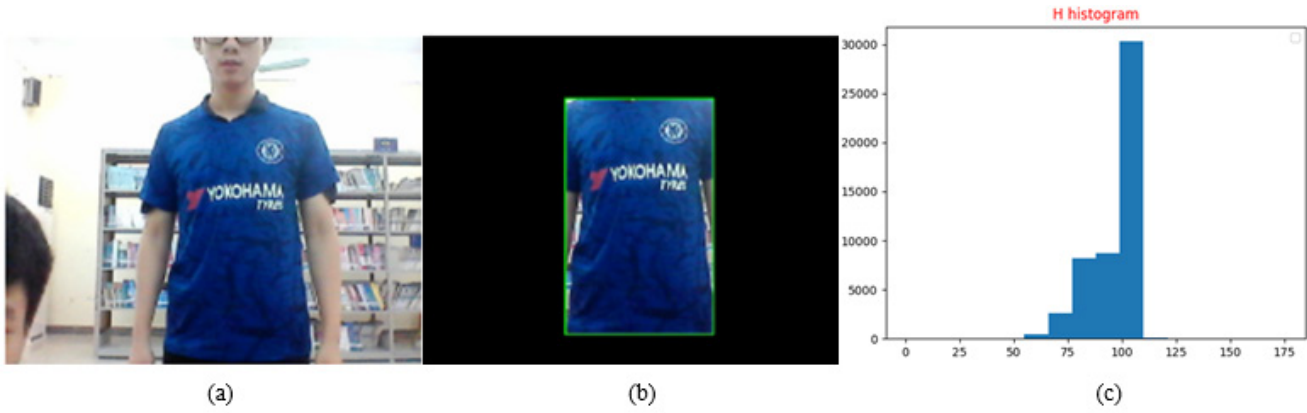


Figure 2: Create ROI and calculate H histogram of ROI

### 2.1. Create ROI and compute H histogram

Before tracking human, an area (ROI) on the target person is established (Figure 2 (b)). This ROI is used as a template of the target. Then, the HSV histogram of ROI is computed, but only the H channel is used with 16 bins to get the best result (Figure 2 (c)).

### 2.2. ORB feature and matching

For the purpose of tracking and creating the initial window for Camshift, this paper selected ORB feature detector (Oriented FAST and Rotated BRIEF) to find the best points. ORB algorithm modifies both FAST [34] keypoint detector and BRIEF [35] descriptor to reach the best result. ORB algorithm detects FAST points in the image, then applies Harris corner measure to discover top  $N$  points among them. After that, the algorithm searches for the intensity weighted centroid of the patch with the located corner at the center in order to calculate the orientation of points. The orientation value is computed by using direction of the vector from this corner point to centroid. After finishing the implementation of FAST, ORB algorithm modifies BRIEF descriptors or rBRIEF to create a descriptor for each keypoint.

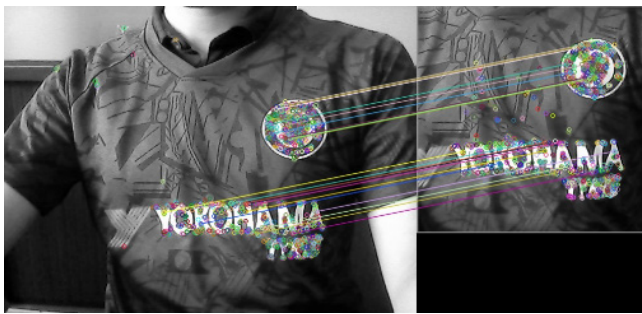


Figure 3: Matching keypoints between ROI and initial Frame

In our system, ORB algorithm is implemented by extracting keypoints from both ROI and frame. The keypoints are searched in Gray image extracting from the original image and the archetype of ROI. Bruce-Force matcher is used to calculate the distance value between each pair of keypoints in both Gray images. By sorted pair of keypoints based on distance value, some keypoints best matched will be kept to use later as shown in Figure 3. To be more specific, the algorithm could operate with higher accuracy in case of more keypoints selection. However, if the number of keypoints incline too big, the calculation process would be taken

long time to finish, especially, in case of mobile robots. In this paper, based on trial and errors method, having 50 keypoints enabled us to ensure the balance between the calculation time and the accuracy of our algorithm effectively.

### 2.3. Track keypoints and remove failure points

The positions of the previous keypoints are tracked by using Optical Flow with Lucas-Kanade Method. Here, a 3x3 patch around the points with an assumption that all the neighboring pixels have similar motion is applied to estimate the new locations of keypoints in the frame based on their previous positions by using equation (1).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i}^2 & \sum_i f_{y_i} \sum_i f_{x_i} \\ \sum_i f_{x_i} \sum_i f_{y_i} & \sum_i f_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i f_{x_i} \cdot f_{t_i} \\ \sum_i f_{y_i} \cdot f_{t_i} \end{bmatrix} \quad (1)$$

where  $u = \frac{dx}{dt}$ ,  $v = \frac{dy}{dt}$  and  $f_x, f_y$  are gradient of keypoints and surrounding points along  $x, y$  axis.

After updating, some of those keypoints in the frame might be failed. In order to remove these failure points, this article combines two methods at the same time: SSD and the Backward tracking technic. Firstly, the SSD in RGB color-space for consecutive frames is expressed as:

$$SSD_R = \sum_{m,n} \begin{pmatrix} R_{t-1}(k_x^{t-1} + m, k_y^{t-1} + n) \\ -R_t(k_x^t + m, k_y^t + n) \end{pmatrix}^2 \quad (2)$$

$$SSD_G = \sum_{m,n} \begin{pmatrix} G_{t-1}(k_x^{t-1} + m, k_y^{t-1} + n) \\ -G_t(k_x^t + m, k_y^t + n) \end{pmatrix}^2 \quad (3)$$

$$SSD_B = \sum_{m,n} \begin{pmatrix} B_{t-1}(k_x^{t-1} + m, k_y^{t-1} + n) \\ -B_t(k_x^t + m, k_y^t + n) \end{pmatrix}^2 \quad (4)$$

where  $k^t$  is the position of keypoint,  $m$  and  $n$  are width and height of the compared window. The idea of  $SSD$  is to compute the total error between a patch around keypoint in the current frame

with the location of them in the previous frame. Figure 4 and Figure 5 illustrate errors between the location of keypoints in consecutive frames corresponding to two cases: small error and big error. After reckoning the  $SSD$  values in three channels, the total error is expressed as follows:

$$Error = \alpha SSD_R + \beta SSD_G + \gamma SSD_B \tag{5}$$

where  $\alpha, \beta, \gamma$  are three positive coefficients and the sum of  $\alpha, \beta, \gamma$  is equal to one:

$$\alpha, \beta, \gamma > 0, \alpha + \beta + \gamma = 1 \tag{6}$$

In this paper, the coefficients are chosen as follows:  $\alpha = 0.3, \beta = 0.4, \gamma = 0.3$ . The error value of all keypoints are illustrated in Figure 6.

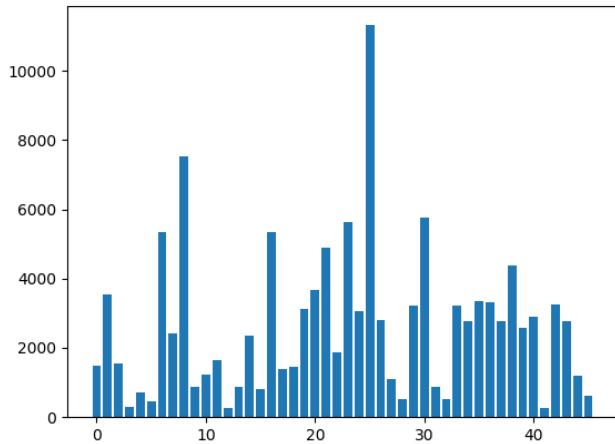


Figure 4: Small error between the location of key points in consecutive frames



Figure 5: Big error between the location of key points in consecutive frames

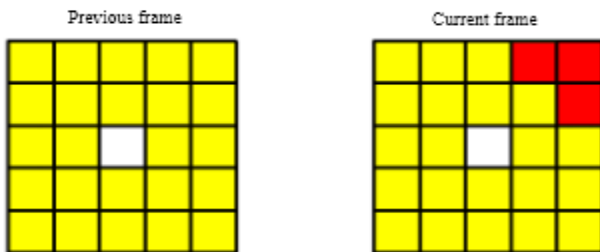


Figure 6: Error value of all key points

After applying  $SSD$  for the first filter, the Backward tracking technic examines all keypoints again and removes the failed points. As shown in Figure 7, the keypoints in the current frame are tracked back to the previous frame by applying Optical Flow.

Let denote  $r^{t-1}$  is the estimated position of  $k^t$  in the previous frame. The distance  $d$  between  $r^{t-1}$  and  $k^{t-1}$  is calculated by using

the Euclid distance method. All keypoints whose distance value is smaller than a threshold hold are kept.

$$d = \sqrt{(k_x^{t-1} - r_x^{t-1})^2 + (k_y^{t-1} - r_y^{t-1})^2} \tag{7}$$

By utilizing two mentioned method above, all failed keypoints are removed. If all keypoints are failed, our method has a viable improvement on current methods. To be more detailed, in this case, a new set of keypoints would be created inside Camshift bounding box of the target.

#### 2.4. Initial search Window for Camshift algorithm

By applying first and zeros moment for all remain keypoints, the mean position of them is indicated as:

$$M_{00} = \sum_x \sum_y I(x, y) \tag{8}$$

$$M_{10} = \sum_x \sum_y xI(x, y) \tag{9}$$

$$M_{01} = \sum_x \sum_y yI(x, y) \tag{10}$$

$$X_c = \frac{M_{10}}{M_{00}}, Y_c = \frac{M_{01}}{M_{00}} \tag{11}$$

with  $I$  is the intensity of keypoints. Since we project all keypoints into a bitmask, so  $I = 1$ . Location of initial search window Camshift is created from the center points.

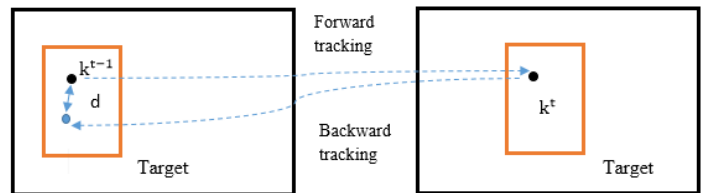


Figure 7: Backward tracking technic

#### 2.5. Human tracking with Camshift

Camshift algorithm is an effective method to find color feature of target in frame. The principle of Camshift algorithm bases on Meanshift algorithm. However, it was modified to update the size of the tracking window in the next frame and find the orientation of the target. For more specific, Camshift computes  $H$  histogram of both ROI and Frame, and then exchanges pixel' value in the frame with the probability value of its color appearance. Then, it creates the color probability distribution image that performs the probability of appearance of each pixel within the range. Camshift algorithm uses a search window and loops it until the center of the search window converges with the centroid of points having high probability. By using an initial search window created with tracking keypoints, it reduces the number of iterations to find the area of the target. The size of tracking window is updated by zeros moment as follows:

$$S = 2 \times \sqrt{\frac{M_{00}}{256}} \tag{12}$$

Then, the orientation of the target is obtained by using the second central moments:

$$\theta = \frac{1}{2} \arctan \left( \frac{2 \left( \frac{M_{11}}{M_{00}} \right) - X_c Y_c}{\left( \frac{M_{20}}{M_{00}} - X_c^2 \right) - \left( \frac{M_{02}}{M_{00}} - Y_c^2 \right)} \right) \quad (13)$$

The aspect ratio is expressed as:

$$Ratio = \frac{M_{20}}{X_c^2} \frac{Y_c^2}{M_{02}} \quad (14)$$

The width and the height of the window in the frame are estimated as follows:

$$w = 2.M_{00}.Ratio \quad (15)$$

$$h = 2.M_{00}.Ratio \quad (16)$$

### 2.6. Kalman filter for the position of the human target

The center of human target finding by Camshift algorithm has some vibration and needs to be removed to improve the smoothness of the human tracking process. Kalman filter is used in this paper to solve the vibration of center position. Kalman filter method estimates the location of the center based on value prediction and the new location of the center. It contains two main stages: prediction and correction.

Let denote  $X_t = [x_t, y_t, v_{xt}, v_{yt}]$  as a state variable and  $Z_t = [x, y]^T$  as measurement variable. Here,  $x_t, y_t$  are the positions of the center along  $x$ -axis and  $y$ -axis in the image.  $v_{xt}$  and  $v_{yt}$  are displaced in the direction of the horizontal position and vertical position. The equations of the motions on the  $x$ -axis and  $y$ -axis without the acceleration are calculated as follows:

$$\hat{x}_t = x_{t-1} + v_{xt} \cdot \Delta t \quad (17)$$

$$\hat{y}_t = y_{t-1} + v_{yt} \cdot \Delta t \quad (18)$$

The following equation is inferred.

$$\hat{X}_t = F.X_{t-1} \quad (19)$$

With  $\hat{X}_t$  is the value prediction of  $X_t$ ,  $F$  is translation matrix.

The predictor covariance equation is expressed as follows:

$$\hat{P}_t = F.P_{t-1}.F^T + Q \quad (20)$$

In the equation,  $\hat{P}_t$  is the value prediction of covariance  $\hat{P}_t$ ,  $Q$  is the interference factor.

In correction step, there are three equations named as: the Kalman gain equation, the State update equation, and the Covariance update equation:

$$K_t = \hat{P}_t.H^T.(H.\hat{P}_t.H^T + R)^{-1} \quad (21)$$

$$X_t = \hat{X}_t + K_t.(Z_t - H\hat{X}_t) \quad (22)$$

$$P_t = \hat{P}_t + K_t.H.\hat{P}_t \quad (23)$$

In these above equations,  $R$  is measurement noise covariance matrix,  $H$  is measurement matrix. The value of  $R, H, F$  and  $Q$  is implemented as follows:

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

### 2.7. Summary methodology

Our algorithm is carried out with the following step:

*Step 1:* Create ROI for human target and analysis the HSV Histogram of ROI.

*Step 2:* Compute ORB features in ROI.

*Step 3:* Compute ORB features in the first frame and match them with ORB features in ROI using BruceForce matching. Take 50 best matching points to track later.

*Step 4:* Tracking those keypoints in the previous step by using the Lucas-Kanade Optical Flow algorithm.

*Step 5:* Remove failed keypoints by using SSD and Backward tracking technic.

*Step 6:* Find the center  $k$  of remain keypoints in the frame with zeros and the first moment.

*Step 7:* Use the center  $k$  to create the initial window for Camshift algorithm.

*Step 8:* Find the bounding box and update centroid of the tracking human by applying Camshift algorithm.

*Step 9:* Use the Kalman filter to improve the smoothness of the human tracking process.

### 3. Experiment results

During the experimental process, the configuration of the computer is as follows: 4 GB ram, Intel(R) Celeron (R) CPU N3350 @ 1.1 GHz, webcam with 0.3 Mpx (480x640) and 2 Mpx Webcam. The algorithm is carried out in Python with the support of OpenCV library and Numpy Library.

For the purpose of showing the development in performance, this paper offers comparisons between the proposed method and the original Camshift [36] The comparisons are checked in different scenes as shown in Figure 8 and Figure 9. To be more specific, the blue bounding box is created by using Camshift and the green one is created by using the proposed method.

Table 1: Average Frame per second value

Criteria	Laptop webcam (0.3Mpx)	Webcam (2Mpx)
Set FPS	30	30
Real FPS	20-25	14-17
Processing time (s)	0.04-0.05	0.06-0.07



Table 2: Experiment results

Environmental conditions	Cases	Number of tests frames	Results of Original Camshift		Results of the proposed method	
			Number of false frames	Accuracy (%)	Number of false frames	Accuracy (%)
Good lighting room condition (500 lux)	One person	323	8	97.5	5	98.45
	Group of people wearing the same shirts	412	298	27.66	15	96.35
	Group of people wearing the different shirts	452	15	96.68	12	97.34
Poor lighting room condition (250 lux)	One person	255	24	90.58	22	91.37
	Group of people wearing the same shirts	542	420	22.5	49	90.95
	Group of people wearing the different shirts	526	47	91.06	38	92.77
Outdoor Condition (700 lux)	One person	552	53	90.39	45	91.84
	Group of people wearing the same shirts	587	401	31.68	55	90.06
	Group of people wearing the different shirts	479	48	89.97	44	90.81

Firstly, in the case of different color shirts as shown in Figure 8, it could be seen that there are almost none of the differences between two approaches. In the second case, we evaluate the performance of the proposed algorithm with undesired humans having the same shirt in random backgrounds. From Figure 9, it is noticeable that the original Camshift recognizes and tracks the wrong target. In contrast, the analyzing and human tracking of the proposed method is very clear and precise.

Table 1 provides the data about the Average Frame per second value of the proposed algorithm on both 0.3 Mpx webcam and 2 Mpx Webcam. On average, it could be seen that the system is fast enough to be utilized in real-time with FPS value of up to 25. Meanwhile, the effect of Kalman filter is evaluated over 400 frames and the results are shown in Figure 10. Furthermore, the proposed method indicates a significant decline in the chattering of the bounding box centre especially in x-axis when human target moves.



Figure 8: Tracking human in case of different color shirts

Table 2 compares the experimental results between the Original Camshift and the proposed approach among different

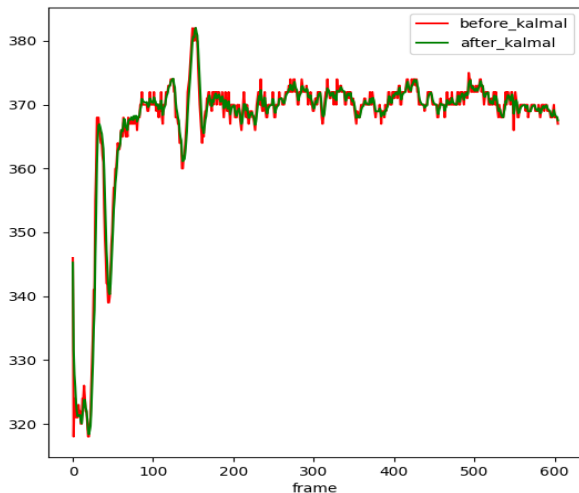
types of the environmental conditions. In each scene, we test 60 times for 3 cases: One person, Group of people wearing the same shirts, Group of people wearing the different shirts. Out of three environmental conditions, it is clear that the false frames in poor lighting room condition of the Original Camshift is the highest, at about 420. By construct, the figure for the proposed method is only 49. Over different cases, the precision rate of the proposed method is quite higher compared to the Original Camshift. For instance, the accuracy of Original Camshift in good lighting room condition is just over 27.66%, while the proportion of the proposed method is nearly 96.35%. In addition, for tracking human wearing the same shirts, the tracking rates of the Original Camshift are noticeably low, at about 27% in average. On the other hand, the percentages of the developed method always maintain over 90%. Furthermore, the results illustrates that the proposed system could work effectively even in the indoor environments or the outdoor environments with the accuracy up to 98%.

#### 4. Conclusions

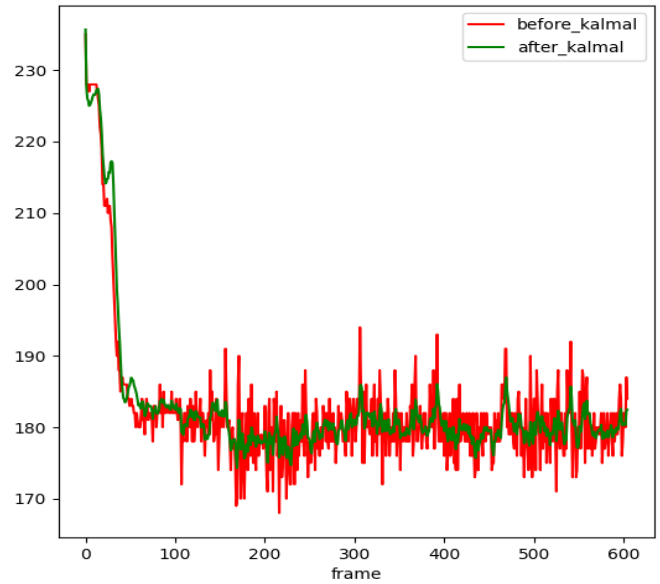
In this article, the new method of human tracking is analyzed in depth. The proposed method is developed based on Camshift algorithm and the Lucas-Kanade Optical Flow Algorithm with Oriented FAST and Rotated BRIEF features. The experiment results indicate that the proposed system could be well-adjusted in real-time applications. By comparing with the existing human tracking algorithms, it could be noted that the proposed algorithm reaches to the higher accuracy. Furthermore, its computing time is relatively faster. Hence, the adaptability of the proposed method is better in practical applications. In conclusion, the present results provide practical reference values about human tracking algorithm for the development of equipment with high anti-interference performance, the design of test plans, and the establishment of international standards in the future.



Figure 9: The test results in case of similar color shirt



(a) Y-axis



(b) X-axis

Figure 10: The position center of bounding box before and after using Kalman filter: (a) Y-axis, (b) X-axis

### Acknowledgement

This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant No. “107.01-2019.311”, and in Hanoi University of Industry, Hanoi, Vietnam, Project no. 04-2020-RD/HD-ĐHCN.

### References

- [1] R. Henschel, T. Marcard, B. Rosenhahn, "Accurate Long-Term Multiple People Tracking Using Video and Body-Worn IMUs," *IET Image Processing*, **29**(11), 8476-8489, 2020, doi: 10.1109/TIP.2020.3013801.
- [2] S. Damocharasamy, "Approach to model human appearance based on sparse representation for human tracking in surveillance," *IET Image Processing*, **14**(11), 2383-2394, 2020, doi: 10.1049/iet-ipr.2018.5961.
- [3] F. Angelini, Z. Fu, Y. Long, L. Shao, S.M. Naqvi, "2D Pose-based Real-time Human Action Recognition with Occlusion-handling," *IEEE Transactions on Multimedia*, **22**(6), 1433- 1446, 2019, doi: 10.1109/TMM.2019.2944745.
- [4] X. Yang, et al, "Human Motion Serialization Recognition With Through-the-Wall Radar," *IEEE Access*, **8**, 186879-186889, 2020, doi: 10.1109/ACCESS.2020.3029247.
- [5] P. Vaishnav, A. Santra, "Continuous Human Activity Classification With Unscented Kalman Filter Tracking Using FMCW Radar," *IEEE Sensors Letters*, **4**(5), 1-4, 2020, doi: 10.1109/LSENS.2020.2991367.
- [6] K.M. Abughalieh, S.G. Alawneh, "Predicting Pedestrian Intention to Cross the Road," *IEEE Access*, **8**, 72558 - 72569, 2020, doi: 10.1109/ACCESS.2020.2987777.
- [7] G. Ren, X. Lu ,Y. Li, "A Cross-Camera Multi-Face Tracking System Based on Double Triplet Networks," *IEEE Access*, **9**, 43759-43774, 2021, doi: 10.1109/ACCESS.2021.3061572.
- [9] Q.Q. Chen, Y.J. Zhang, "Sequential segment networks for action recognition." *IEEE Signal Processing Letters*, **24**(5), 712-716, 2017, doi: 10.1109/LSP.2017.2689921.
- [11] B. Chung, C. Yim, "Bi-sequential video error concealment method using adaptive homographybased registration," *IEEE Transactions on Circuits and Systems for Video Technology*, **30**(6), 1535 - 1549, 2019, doi: 10.1109/TCSVT.2019.2909564.

- [12] Z. Yu, L. Han, Q. An, H. Chen, H. Yin and Z. Yu, "Co-Tracking: Target Tracking via Collaborative Sensing of Stationary Cameras and Mobile Phones," *IEEE Access*, **8**, 92591-92602, 2020, doi: 10.1109/ACCESS.2020.2979933.
- [13] B. Deori and D. M. Thounaojam, "A survey on moving object tracking in video," *International Journal on Information Theory (IJIT)*, **3**(3), 31-46, 2014, doi: 10.5121/ijit.2014.3304.
- [14] E.J. Jung, et al, "Development of a laser-rangefinder-based human tracking and control algorithm for a marathoner service robot," *IEEE/ASME transactions on mechatronics*, **19**(6), 1963-1976, 2014, doi: 10.1109/TMECH.2013.2294180.
- [15] K. Dai, Y. Wang, Q. Ji, H. Du, C. Yin, "Multiple Vehicle Tracking Based on Labeled Multiple Bernoulli Filter Using Pre-Clustered Laser Range Finder Data," *IEEE Transactions on Vehicular Technology*, **68**(11), 10382-10393, 2019, doi: 10.1109/TVT.2019.2938253.
- [16] D. Li, et al, "A multitype features method for leg detection in 2-D laser range data," *IEEE Sensors Journal*, **18**(4), 1675-1684, 2017, doi: 10.1109/JSEN.2017.2784900.
- [17] M. J. Dominguez-Morales, et al, "A. Bio-Inspired Stereo Vision Calibration for Dynamic Vision Sensors," *IEEE Access*, **7**, 138415-138425, 2019, doi: 10.1109/ACCESS.2019.2943160.
- [18] B. Bozorgtabar and R. Goecke, "MSMCT: Multi-State Multi-Camera Tracker," *IEEE Transactions on Circuits and Systems for Video Technology*, **28**(12), 3361-3376, 2018, doi: 10.1109/TCSVT.2017.2755038.
- [19] Y. Han, R. Tse and M. Campbell, "Pedestrian Motion Model Using Non-Parametric Trajectory Clustering and Discrete Transition Points," *IEEE Robotics and Automation Letters*, **4**(3), 2614-2621, 2019, doi: 10.1109/LRA.2019.2898464.
- [20] V. Carletti, A. Greco, A. Saggese, M. Vento, "Multi-object tracking by flying cameras based on a forward-backward interaction," *IEEE Access*, **6**, 43905-43919, 2018, doi: 10.1109/ACCESS.2018.2864672.
- [21] F.L. Zhang, et al, "Coherent video generation for multiple hand-held cameras with dynamic foreground," *Computational Visual Media*, **6**(3), 291-306, 2020, doi: <https://doi.org/10.1007/s41095-020-0187-3>.
- [22] K. Hasegawa, H. Saito, "Synthesis of a stroboscopic image from a hand-held camera sequence for a sports analysis," *Computational Visual Media*, **2**(3), 277-289, 2016, doi: <https://doi.org/10.1007/s41095-016-0053-5>.
- [23] X. Kong, et al, "Particle filter-based vehicle tracking via HOG features after image stabilisation in intelligent drive system," *IET Intelligent Transport Systems*, **13**(6), 942-949, 2018, doi: <https://doi.org/10.1049/iet-its.2018.5334>.
- [24] S. Papaioannou, A. Markham, N. Trigoni, "Tracking people in highly dynamic industrial environments," *IEEE Transactions on mobile computing*, **16**(8), 2351-2365, 2016, doi: 10.1109/TMC.2016.2613523.
- [25] Y. Zhai, et al, "Occlusion-aware correlation particle filter target tracking based on RGBD data," *IEEE Access*, **6**, 50752-50764, 2018, doi: 10.1109/ACCESS.2018.2869766.
- [26] Y. Zhang, "Detection and Tracking of Human Motion Targets in Video Images Based on Camshift Algorithms," *IEEE Sensors Journal*, **20**(20), 11887-11893, 2020, doi: 10.1109/JSEN.2019.2956051.
- [27] F. Li, R. Zhang, F. You, "Fast pedestrian detection and dynamic tracking for intelligent vehicles within V2V cooperative environment," *IET Image Processing*, **11**(10), 833-840, 2017, doi: 10.1049/iet-ipr.2016.0931.
- [28] Q. Tran, S. Su and V.T. Nguyen, "Pyramidal Lucas—Kanade-Based Noncontact Breath Motion Detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **50**(7), 2659-2670, 2020, doi: 10.1109/TSMC.2018.2825458.
- [29] B. Du, S. Cai and C. Wu, "Object Tracking in Satellite Videos Based on a Multiframe Optical Flow Tracker," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **12**(8), 3043-3055, 2019, doi: 10.1109/JSTARS.2019.2917703.
- [30] Q. Tran, S. Su, C. Chuang, V.T. Nguyen and N. Nguyen, "Real-time non-contact breath detection from video using adaboost and Lucas-Kanade algorithm," in 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSAS-SCIS), Otsu, Japan, 1-4, 2017.
- [31] T. Song, B. Chen, F. M. Zhao, Z. Huang, M. J. Huang, "Research on image feature matching algorithm based on feature optical flow and corner feature," *The Journal of Engineering*, **2020**(13), 529-534, 2020, doi: <https://doi.org/10.1049/joe.2019.1156>.
- [32] J. de Jesus Rubio, Z. Zamudio, J. A. M. Campana, M. A. M. Armendariz, "Experimental vision regulation of a quadrotor," *The Journal of Engineering*, **13**(8), 2514-2523, 2015, doi: 10.1109/TLA.2015.7331906.
- [33] J. Li, Z. Li, Y. Feng, Y. Liu, G. Shi, "Development of a human-robot hybrid intelligent system based on brain teleoperation and deep learning SLAM," *IEEE Transactions on Automation Science and Engineering*, **16**(4), 1664-1674, 2019, doi: 10.1109/TASE.2019.2911667.
- [34] C. Xin, Z. Jingmei, Z. Xiangmo, W. Hongfei, C. Hui " Vehicle ego-localization based on the fusion of optical flow and feature points matching," *IEEE Access*, **7**, 167310-167319, 2019, doi: 10.1109/ACCESS.2019.2954341.
- [35] S. K. Lam, G. Jiang, M. Wu, B. Cao, "Aretime efficient streaming architecture for FAST and BRIEF detector," *IEEE Transactions on Circuits and Systems II: Express Briefs*, **66**(2), 282-286, 2019, doi: 10.1109/TCSII.2018.2846683.
- [36] N. Q. Nguyen, S. F. Su, Q. V. Tran, V. T. Nguyen, J. T. Jeng, "Real time human tracking using improved CAM-shift," in *IEEE International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSAS-SCIS)*, 1-5, 2017.