

Research Article

Real-Time Pedestrian Tracking and Counting with TLD

Jiawei Shi,¹ Xianmei Wang ,¹ and Huer Xiao²

¹School of Computer and Communication Engineering, University of Science and Technology Beijing, 100083, China

²Center International Economic and Technological Cooperation Ministry and Information Technology, Beijing 100846, China

Correspondence should be addressed to Xianmei Wang; xmwang@ustb.edu.cn

Received 15 April 2018; Revised 8 August 2018; Accepted 26 August 2018; Published 29 October 2018

Academic Editor: Fulvio Simonelli

Copyright © 2018 Jiawei Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper describes a solution to solve the issue of automatic multipedestrian tracking and counting. First, background modeling algorithm is applied to actively obtain multipedestrian candidates, followed by a confirmation step with classification. Then each pedestrian patch is handled by real-time TLD (Tracking-Learning-Detection) to get a new predication position according to similarity measure. Further TLD results are compared with classification list to determine a new, disappeared, or existing pedestrian. Finally single line counting with buffer zone is employed to count pedestrians. Experiments results on the public database, PETS, demonstrate the validity of our solution.

1. Introduction

Pedestrian tracking and counting is an extremely significant research in the field of computer vision. It plays a crucial role in many applications including the intelligent monitoring and traffic safety. Nevertheless, some challenges such as great variation of the pedestrian posture, background clutter, partial occlusions, and illumination changing complicate the issue.

Current state-of-the-art tracking algorithms can be roughly divided into two categories: generative and discriminative pattern. Generative methods [1, 2], generally, describe the characteristics performance via the generated model and then minimize the reconstruction error by searching the candidate targets. Compared with generative methods, discriminative approaches [3–5] distinguish target and background by trained classifiers to find a decision boundary between object and background. By adequately using both of the target and background information, this approach achieves higher tracking accuracy. In recent years, discriminative approaches have been obtaining vigorous developments.

TLD is a popular discriminative method [5]. It integrates an online detector to obtain a good ability of target redetect, which is important to realize continuous tracking in the case of target disappearance in tracking process. In addition,

TLD can solve long-term accurate tracking by online learning.

TLD has been used in different tracking tasks. Zhang et al. [6] use TLD to realize dynamic gesture tracking. To initialize TLD algorithm, a specific gesture is manually marked by a bounding box in the first frame. Chen et al. [7] improved TLD to track a human in unconstrained environments, and the object needs to be initialized manually as well. Crane tracking and monitoring with stable TLD also show strong robustness and accuracy [8]. However all of these studies are under single target tracking conditions.

Pedestrian flow statistics is used to track pedestrians and count their number in the video [9], such as the surveillance of crossroads. Compared with above studies, many people may appear at the same time, leading to the request of multiple object tracking. Moreover, any pedestrian may randomly appear. As a result, the targets cannot be extracted manually and automatic searching for tracking objects is also a key for multipedestrian tracking and counting.

We propose a new method called improved TLD (ITLD) by introducing background subtraction to automatically obtain multiobject and then use the updating mechanism of tracking list to realize multiobject management. Figure 1 shows the framework of our system, integrating with counting module.

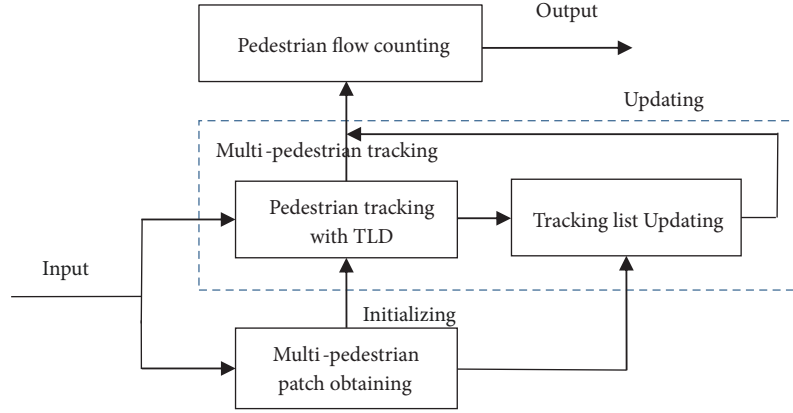


FIGURE 1: The framework of our proposed system.

2. Multipedestrian Patch Obtaining

There are several ways to realize multiple objects obtaining. Cao et al. [10] use Haar features and AdaBoost algorithm to automatically detect human faces. Zhou et al. [11] proposed PE-TLD, after the process of ViBe and variance filter, hog features, and SVM classifier were used for automatic targets recognition. Besides, S. Sharma et al. [12] let users manually select the desired tracking objects in the initialization process.

In our system a coarse to fine strategy is employed to obtain multitarget. Firstly, a dynamic average background model followed by Ostu algorithm is used to extract candidate pedestrian patches. Next, we utilize pedestrian detection combining Haar-AdaBoost [13, 14] and Hog-SVM [15] to further exclude those nonpedestrian patches.

2.1. Extraction of Candidate Pedestrian Patch. Our dynamic average background modeling is based on Gauss statistics, considering the distinction of three color channel variances. In order to reduce modeling error, we make the different statistics in RGB space for all pixels. After the construction of background model, a simple subtract operation between current frame and latest background is employed to get a rough foreground part. In this step Ostu algorithm is used to obtain segmentation threshold. Then morphological processing is used to further extract those small regions and fill the gaps among foreground part. Finally, minimum circumscribed rectangle is used for getting possible pedestrian targets. The detailed process is shown in [16].

Given a pixel, suppose n is the serial number of frame in video stream, (x, y) is its position, $v_r^{(n)}(x, y)$, $v_g^{(n)}(x, y)$, and $v_b^{(n)}(x, y)$ are the values of its three color components; then this pixel can be represented by $p^{(n)}(x, y) = (v_r^{(n)}(x, y), v_g^{(n)}(x, y), v_b^{(n)}(x, y))$. The construct process of background model $M_c(x, y)$ can be expressed by (1)-(3).

$$M_c(x, y) = \frac{\sum_{n=0}^{N-1} v_c^{(n)}(x, y) F^{(n)}(x, y)}{\sum_{n=0}^{N-1} F^{(n)}(x, y)}, \quad c \in \{r, g, b\} \quad (1)$$

where $F^{(n)}(x, y)$ denotes the binary mask, which is calculated by

$$F_c^{(n)}(x, y) = \begin{cases} 1 & \text{if } |v_c^{(n)}(i, j) - \mu_c^0(i, j)| \leq \sigma_c^0(i, j) \quad \forall c \in \{r, g, b\} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$n = 0, 1 \dots N - 1$$

$$F^{(n)}(x, y) = \sqrt{F_c^{(n)}(x, y)}$$

where $\mu_c^0(x, y)$ and $\sigma_c^0(x, y)$ represent the component mean and standard deviation of pixel $p^{(n)}(x, y)$ from frame $n-N+1$ to current frame n . In our work, N is set as 300.

The recurrences of background model updating are shown in the following:

$$M_c^{(k)}(x, y) = \frac{M_c^{(k-1)}(x, y) \left(\sum_{n=0}^{N-1} F^{(n+k-1)}(x, y) \right) + v_c^k(x, y) F^{(k)}(x, y) - v_c^{k-N}(x, y) F^{(k-N)}(x, y)}{\left(\sum_{n=0}^{N-1} F^{(n+k-1)}(x, y) \right) + F^{(k)}(x, y) - F^{(k-N)}(x, y)} \quad (3)$$

$$c \in \{r, g, b\}, \quad k \geq N$$

2.2. Pedestrian Confirmation. Haar and HOG are two excellent features to describe pedestrians. In each frame, to obtain higher performance, we integrate HOG feature with

SVM classifier and Haar feature with AdaBoost classifier (shown in Figure 2) to determine if an image patch includes a pedestrian or not. An image patch will be

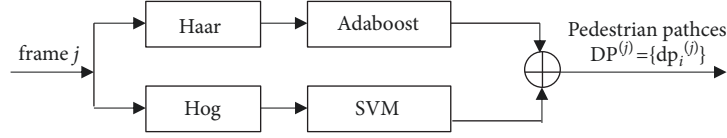


FIGURE 2: Classifier combination for pedestrian confirmation.

thought to include a pedestrian, if either classifiers output yes.

In this step, a patch set $DP^{(j)} = \{dp_i^{(j)}(x, y, w, h), i = 1, 2, \dots, N_d\}$ is used to record the pedestrian patches in frame j , where x, y, w , and h are the position and size of $dp_i^{(j)}$.

3. Multipedestrian Tracking with TLD

The output $DP^{(j)}$ of pedestrian classification servers as the input of TLD algorithm, which can solve the automatic selection of multiobject. Since pedestrian presence or disappearance from the camera is random, the management of multiobject list, including inserting, deleting, or maintaining, becomes important.

3.1. Single Pedestrian Tracking with TLD. Single pedestrian tracking with TLD include three components, namely, tracking, learning, and detection [5]. Data preparation is the first step to implement TLD framework. Given a target $dp_i^{(j)}$, use overlap area A_o under different scales to choose positive and negative samples. Within each scale, we can get a collection of patches from top to bottom, left to right. Then calculate the overlap ratio $r = A_o / (dp_i^{(j)}.w * dp_i^{(j)}.h + A_{patch} - A_o)$ between a patch and $dp_i^{(j)}$. Add the best 10 patches with maximum overlap ration (larger than 0.6) into the positive sample set. For those patches with overlap ration smaller than 0.1, we put them into the negative samples set. Before adding, each patch is resized into normalized size.

Thus i^{th} object model in frame j can be expressed as $M_i^{(j)} = \{M_i^{(j)+}, M_i^{(j)-}\} = \{p_{i1}^{(j)+}, p_{i2}^{(j)+}, \dots, p_{im}^{(j)+}, p_{i1}^{(j)-}, p_{i2}^{(j)-}, \dots, p_{in}^{(j)-}\}$. $p_{i1}^{(j)+}$ is the first positive patch added to the set of $M_i^{(j)}$. And $p_{im}^{(j)+}$ is the positive sample added last so far. After learning initialization, we can use the TLD framework shown in [5]. In model update process, relative similarity $S(a,b)$ is used to measure the similarity between objects a and b . The similarity between the image patch P_{new} and set $M_i^{(j)}$ is calculated as (4)-(7).

$$S^r(P_{new}, M_i^{(j)}) = \frac{S^+(P_{new}, M_i^{(j)+})}{S^+(P_{new}, M_i^{(j)+}) + S^-(P_{new}, M_i^{(j)-})} \quad (4)$$

$$S^+(P_{new}, M_i^{(j)+}) = \max_{p_{ik}^{(j)+} \in M_i^{(j)+}} S(P_{new}, p_{ik}^{(j)+}) \quad (5)$$

$$S^-(P_{new}, M_i^{(j)-}) = \max_{p_{ik}^{(j)-} \in M_i^{(j)-}} S(P_{new}, p_{ik}^{(j)-}) \quad (6)$$

where $S^+(P_{new}, M_i^{(j)+})$ and $S^-(P_{new}, M_i^{(j)-})$ are the similarity between P_{new} and positive training set $M_i^{(j)+}$ and negative training set $M_i^{(j)-}$.

Expression (7) is used to decide whether a new patch P_{new} is a positive one.

$$\begin{aligned} & \text{if } S^r(P_{new}, M_i^{(j)}) > \theta_i^{(j)} \\ & \text{then } P_{new} \in M_i^{(j)+} \\ & \text{else } P_{new} \in M_i^{(j)-} \end{aligned} \quad (7)$$

The output of integrator is measured by conservative similarity shown in the following.

$$S^c(P_{new}, M_i^{(j)}) = \frac{S_{50\%}^+(P_{new}, M_i^{(j)})}{S_{50\%}^+(P_{new}, M_i^{(j)}) + S^-(P_{new}, M_i^{(j)})} \quad (8)$$

where $S_{50\%}^+(P_{new}, M_i^{(j)})$ is the similarity of the first 50% of the positive patches.

For each $dp_i^{(j-1)}$, we use single TLD framework to get its new position if it is visible in frame j . If n objects are visible in current frame, we can get a new list and record it as $TLD^{(j)} = \{tb_i(x, y, w, h), i = 1, 2, \dots, N\}$. Here x, y, w , and h are the position and size of the bounding box given by TLD.

3.2. List Updating for Multiobject Tracking. Because of the random of pedestrian's appearance and disappearance, the dynamical maintenance of pedestrian list becomes a key in multipedestrian tracking and counting. For example, how to decide a target is a new or an existing one, especially for the case of occlusion or out of vision and then back again. Suppose every target is independent; we design a mechanism for the tracking list updating.

We record the information such as position and size and lifespan from the first appearance to current frame j as the trajectory of a tracking object. Tracking list T is the set of all tracking object. For current trajectory T , we note its i^{th} tracking target as $T_i = \{tb, tbold, len, vlen, ivlen, rd, ld\}$. Here, tb is the position and size of current output by TLD in frame j . $tbold$ is the last record of position and size before current. And len , $vlen$, and $ivlen$ are, respectively, the length of total frames (from first appearance to current frame), visible frames, and consecutive invisible frames. rd and ld are the labels to describe if a pedestrian passes the left or right border of buffer zone (introduced in Section 4). By comparing current $DP^{(j)}$ and the existing object trajectory list T , a new tracking list can be updated.

Our tracking list updating can be divided into two steps: correlation matching and list adjustment. For each trajectory, correlation matching step is to find the most similar patch in the detection set $DP^{(j)}$. The similarity $s(a, b)$ between patches a and b is measured by the Euclidean distance between their

centroids. If the similarity $s(T_i.tb, dp_h^{(j)})$ satisfies (9), patch $dp_h^{(j)}$ is accepted as the most resemble appearance of T_i .

$$s(T_i.tb, dp_h^{(j)}) = \min_{dp_i^{(j)} \in DP^{(j)}} \{s(T_i.tb, dp_i^{(j)})\} \quad (9)$$

$$\text{and } s(T_i.tb, dp_h^{(j)}) < T_0$$

List adjustment process is to update T . For those matched trajectories we update their records with new parameters. For those unmatched patches in $DP^{(j)}$, we regard them as new pedestrian targets. Then we add them into T as new trajectory elements. As the trajectories disappeared or unmatched for requirement threshold, we incline to view them as vanishing from the video and then remove them from the set T . The detail updating flow of tracking list is shown in Algorithm 1.

4. Pedestrian Flow Counting

Tracking trajectory is the source of pedestrian counting. So far, single or double line counting is the popular way to

count pedestrians (moving left or right). Considering the acquirement of real-time, stability, and accuracy, we choose single line counting with buffer zone shown in Figure 3 to count pedestrians.

To avoid the direction statistical error caused by the wandering pedestrian near the counting line, buffer zone is introduced. An image frame only has one buffer which is a small range centered on the counting line (the center line of the video). And only when a target crosses the whole zone, does the direction counting continue.

Let the margins of the buffer range be L_L and L_R . The way to count right or left moving pedestrians is shown in the following.

$$\begin{aligned} &\text{if } T_i.rd = 1 \& T_i.ld = 1 \\ &\text{then } Num_R = Num_R + 1 \\ &\text{if } T_i.rd = -1 \& T_i.ld = -1 \\ &\text{then } Num_L = Num_L + 1 \end{aligned} \quad (14)$$

where

$$\begin{aligned} T_i.rd &= \begin{cases} 1 & T_i.tb.x + T_i.tb.w > L_R \& T_i.tb.x < L_R \& T_i.tb.x > T_i.tbold.x \\ -1 & T_i.tb.x + T_i.tb.w > L_R \& T_i.tb.x < L_R \& T_i.tb.x < T_i.tbold.x \\ 0 & T_i.tb.x + T_i.tb.w < L_L \parallel T_i.tb.x > L_R \end{cases} \\ T_i.ld &= \begin{cases} 1 & T_i.tb.x + T_i.tb.w > L_L \& T_i.tb.x < L_L \& T_i.tb.x > T_i.tbold.x \\ -1 & T_i.tb.x + T_i.tb.w > L_L \& T_i.tb.x < L_L \& T_i.tb.x < T_i.tbold.x \\ 0 & T_i.tb.x + T_i.tb.w < L_L \parallel T_i.tb.x > L_R \end{cases} \end{aligned} \quad (15)$$

Once target T_i has been counted, moving label rd and ld will be reset to zero immediately.

5. Experiments and Analysis

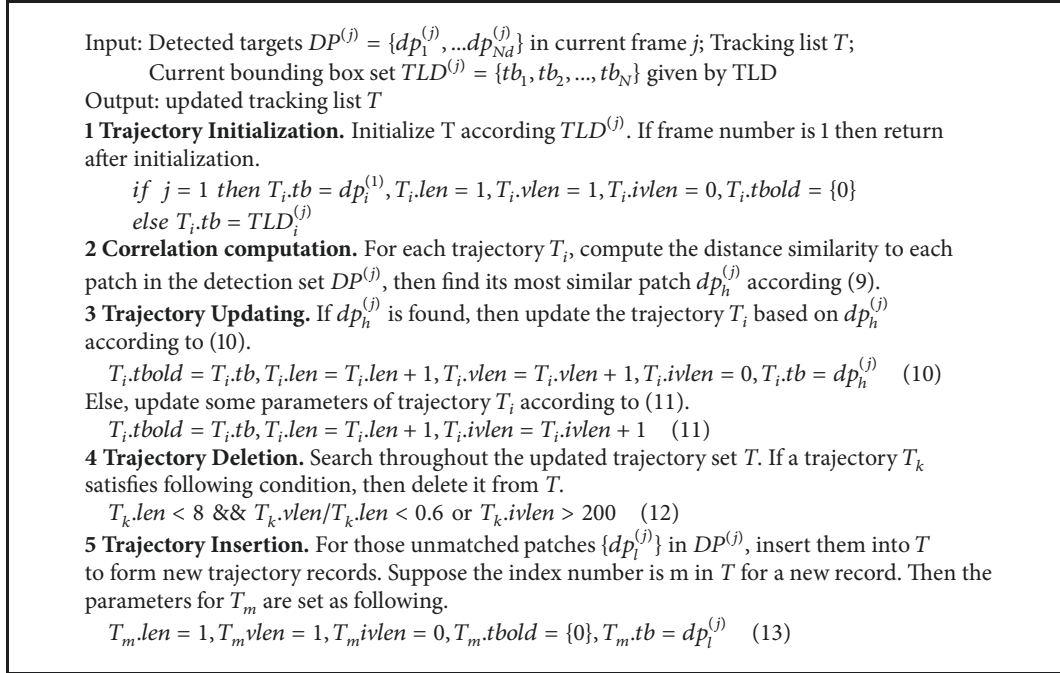
Two experiments are designed to test the performance of our multiobject tracking and pedestrian counting on the database of PETS [21]. PETS is a public dataset which consists of multisensor sequences containing different crowd activities. It has been widely used to test the performance of new or existing systems of pedestrian detection and tracking within a real-world environment. We select ten clips for pedestrian counting from the PETS. The longest clip has 1189 frames and the shortest is 276 frames.

5.1. The Evaluation of Multitarget Tracking. MOTA (multiple object tracking accuracy) and MOTP (multiple object tracking precision) [22] are adopted to evaluate tracking process. MOTP evaluates the alignment of tracks with the ground truth. It measures the precision with which objects are located using the intersection of the estimated region with the ground truth region. And MOTA combines all missed targets, false positives, and identity mismatches and is normalized with

the total number of targets (100% corresponds to no errors) [17]. Table 1 presents the comparison results of our method and some of the state-of-the-art approaches for multiobject tracking.

From Table 1, we can observe that our tracking approach can achieve a competitive result, especially the metric value of MOTP. For [17, 18], MOTP and MOTA cannot achieve good values simultaneously. When one measure increases, the other metric decreases obviously. Moreover, the MOTA of Berclaz approach is the highest, because it uses probabilistic occupancy map (POM) to create background and detection data. Compared with our basic foreground segmentation process without considering any prior real data, POM model uses some empty background images in PETS.

As [19], it can obtain a good balance between MOTA and MOTP, but its metric value is a little lower than that of our method. The reason may be attributed to the performance of TLD algorithm. TLD combines learning, detection, and tracking together for a tracking task with parameters online updating. To achieve high detection accuracy, P-N learning paradigm is used to exploit the temporal and spatial structure in data, leading to the mutual compensation of missed



ALGORITHM 1: The flow for multiobject tracking.

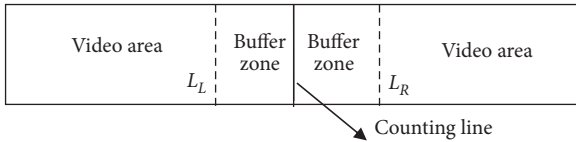


FIGURE 3: Single line counting method with buffer zone.

TABLE 1: The comparison results of multitarget tracking.

Algorithm	MOTP	MOTA
Andriyenko [17]	69.0%	63.7%
Berclaz [18]	63.0%	77.0%
Milan [19]	67.2%	67.0%
Jin [20]	72.4%	72.1%
Our method	69.9%	71.4%

detection and false alarms and the promotion of tracking accuracy and precision.

We can also see that [20] can get a bit higher performance than our method. To use the holography information among multiple view, [20] integrates the crowd simulation into traditional single camera method. However, the two metric values in [20] are computed by manually assigning the size and position of the initial patch for each tracking. In contrast, our method extracts foreground and initializes each tracker automatically. This fully automatic step may decrease the performance of our tracking system to some extent. In order to improve the performance, in the initialization stage, we can use more proper detection method and better background modeling method in our future work.

5.2. Statistics of Pedestrian Flow. Table 2 displays the results of our statistics of pedestrian flow. L and R mean the number of pedestrians walking toward left or right. In our 10 clips, the illumination of 1 and 2 is inadequate, clips 7 and 8 have many pedestrians, and 9 and 10 have serious occlusion problem. The other clips are in normal condition.

From Table 2, it is easy to be observed that our system obtains higher accuracy during normal condition (especially 3 and 5). For those clips with crowded pedestrians or frequent occlusion (clips 7, 8, 9, and 10), the accuracy is about 81% to 84%. For those clips (clips 1 and 2) with inadequate illumination but less objects and occlusion, the accuracy is over 86%. And for the all clips, our average statistical accuracy is 87.4%.

6. Conclusions

To realize multiobject tracking, our system combines TLD tracking algorithm and dynamic average background modeling. The former can track objects with long term and robustness in real-time. The latter with confirmation module can automatically localize candidate multiobject, which are further tested by pedestrian detection. By comparing the TLD output and pedestrian detection results, we can manage pedestrian records easily, such as updating parameters, inserting new objects, and deleting those disappeared ones for long time. Counting with buffer zone can decrease the influence of the wandering of pedestrians around counting line. The accuracy and stability of our system have been proved by several experiments and analysis. Future work will focus on how to improve the tracking accuracy for high-density crowd and better robustness.

TABLE 2: The statistics result of pedestrian flow counting.

Video sequence	Authentic pedestrian number	Our Statistics result	Statistical errors	Accuracy rate
1	L:12 R:6	L:10 R:6	L:-2 R:0	88.88%
2	L:15 R:7	L:13 R:6	L:-2 R:-1	86.36%
3	L:6 R:6	L:6 R:6	L:0 R:0	100%
4	L:8 R:15	L:10 R:14	L:+2 R:-1	86.96%
5	L:17 R:14	L:16 R:13	L:-1 R:-1	93.55%
6	L:9 R:11	L:8 R:12	L:-1 R:+1	90%
7	L:42 R:33	L:34 R:28	L:-8 R:-5	81.58%
8	L:39 R:25	L:32 R:20	L:-7 R:-5	81.25%
9	L:18 R:26	L:14 R:23	L:-4 R:-3	84.09%
10	L:21 R:12	L:17 R:10	L:-4 R:-2	81.82%

Data Availability

All the data used in our experiments are from the public database PETS and can be download from <http://www.cvg.reading.ac.uk/PETS2009/a.html>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is partially sponsored by the National Key Development Plan of Fundamental Research 2017YFB1302203.

References

- [1] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125-141, 2008.
- [2] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 213-228, 2015.
- [3] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096-2109, 2016.
- [4] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1090-1097, IEEE, Columbus, Ohio, USA, June 2014.
- [5] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, 2012.
- [6] Y. Zhang, Y. Yao, Y. Luo, and T. Zhang, "A dynamic gesture tracking algorithm based on the improved tracking-learning-detection," *Robot*, vol. 37, no. 6, pp. 754-759, 2015.
- [7] L. Chen, S. Zhu, X. Li, and J. Liu, "Target tracking via improved TLD algorithm," in *Proceedings of the 27th Chinese Control and Decision Conference, CCDC 2015*, pp. 5942-5947, China, May 2015.
- [8] H. Sun and T. Yu, "Crane tracking and monitoring system based on TLD algorithm," in *Proceedings of the 2016 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1-5, Taipei, Taiwan, May 2016.
- [9] C. Yan-yan, C. Ning, Z. Yu-yang, W. Ke-han, and Z. Wei-wei, "Pedestrian detection and tracking for counting applications in metro station," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 712041, 11 pages, 2014.
- [10] W. Cao, G. Xu, B. Lei, P. Yin, and F. Dong, "A multiple face detection and tracking system based on TLD," in *Proceedings of the 5th International Conference on Internet Multimedia Computing and Service, ICIMCS 2013*, pp. 386-389, China, August 2013.
- [11] C. Zhou, Q. Dong, W. Ma, G. Long, and T. Li, "PE-TLD: parallel extended tracking-learning-detection for multi-target tracking," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 665-677, 2015.
- [12] S. Sharma, A. Khachane, and D. Motwani, "Real time multi-object tracking using TLD framework," in *Proceedings of the 2016 International Conference on Inventive Computation Technologies, ICICT 2016*, pp. 1-6, India, August 2016.
- [13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 511-518, December 2001.
- [14] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. 900-903, September 2002.
- [15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886-893, June 2005.
- [16] X. Wang, Z. Zhang, X. Lou, and J. Wang, "Real-time pedestrian detection by background subtraction," *Journal of Information and Computational Science*, vol. 12, no. 16, pp. 6005-6013, 2015.
- [17] A. Andriyenko, S. Roth, and K. Schindler, "An analytical formulation of global occlusion reasoning for multi-target tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1839-1846, November 2011.
- [18] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806-1819, 2011.

- [19] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.
- [20] Z. Jin and B. Bhanu, "Analysis-by-synthesis: Pedestrian tracking with crowd simulation models in a multi-camera video network," *Computer Vision and Image Understanding*, vol. 134, pp. 48–63, 2015.
- [21] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proceedings of the 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter)*, pp. 1–6, Snowbird, UT, USA, December 2009.
- [22] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *Eurasip Journal on Image and Video Processing*, vol. 2008, no. 1, 2008.



Hindawi

Submit your manuscripts at
www.hindawi.com

