Robust Object Tracking with Bidirectional Corner Matching and Trajectory Smoothness Algorithm

Guoshan Wu[1], Yi Xu[2], Xiaokang Yang, Qing Yan, Ke Gu

Institute of Image Communication and Information Processing, Shanghai JiaoTong University
Shanghai Key Lab.of Digital Media Processing and Transmissions, Shanghai, China

[1] yuyushi@sjtu.edu.cn
[2] xuyi@sjtu.edu.cn

***Abstract*—This paper proposes a novel method for robust object tracking. The method consists of three different components: a short term tracker, an object detector, and an online object model. For the short term tracker, we use an advanced Lucas Kanade tracker with bidirectional corner matching to capture object frame by frame. Meanwhile, statistical filtering and matching algorithm combined with haar-like feature random fern play as a detector to extract all possible object candidates in the current frame. Making use of trajectory information, the online object model decides the best target match among the candidates. And the model also trains the random fern feature adaptively online to better guide consecutive tracking. We demonstrate our method is robust to track an object in a long term and under large variations of view angle and lighting conditions. Moreover, our method is efficient to re-detect the object and keep tracking even after it's out of view or recover from heavy occlusion. To achieve state-of-the-art performance, it is highlighted that our method can be extended to multiple objects tracking application. Finally, comparisons with other state-of-the-art trackers are presented to show the robustness of our tracker.**

I. INTRODUCTION

Visual tracking is one of the most popular research topic in computer vision and machine learning/understanding. It has many applications in reality, such as real time surveillance, human-computer interaction, cognitive video retrieval and robot intelligence. However, long-term robust visual tracking faces tough challenges in view angle changing, lighting condition variance and heavy occlusions. For example, re-tracking of an occluded object is a big problem when the object reappears in the view.

Lucas-Kanade optical flow method performs frame-by-frame point tracking with an assumption that the target object continually exists in the consecutive frames [2]. The tracker may be drift away if tracking points are not distinct. Then some authors use corner points [3, 4] to conduct feature-based object tracking. Corner points provide the most representative object features, but they would be vulnerable to noises, and might contain many outliers. Recently, Kalal *et.al.* [1] proposed a forward-backward error detection algorithm to improve the accuracy of tracking points. Inspired by their works, we propose a short term tracker with bidirectional corner matching, which combines the Lucas Kanade corner point tracker with forward-backward error detection.

It should be noted that the Lucas Kanade based short term tracker assumes all tracking points are visible in the consecutive frames. As a result, it cannot recover from object occlusion and capture the object once it reappears after occlusion. Kalal *et.al.* [5, 6] proposed a TLD framework which integrates an online trained detector into tracking procedure , which greatly improved tracking efficiency under object appearance changes through online adaptation and multiple appearance template matching. However, this work may be unstable since it hasn't considered trajectory consistency. In addition, multiple object tracking is not discussed in this work.

In this paper, we add statistical features and trajectory smoothness algorithm to increase tracking stability. Computational complexity is also reduced to meet real time applications. Our framework also can be applied to track multiple objects and it achieves state-of-the art performance.

The rest of the paper is organized as follows: Section II presents our short term tracker with bidirectional corner matching; Statistical feature and trajectory based detection strategy is discussed in Section III; Section IV provides comparison analysis and experimental results. Section V gives conclusion remarks.

II. SHORT TERM TRACKER WITH BIDIRECTIONAL CORNER MATCHING

This section proposes the algorithm of bidirectional corner matching, which introduces forward-backward error detection strategy into corner feature tracking results. Firstly, we extract Harris corner points to represent object at frame t. Such points provide rich context information and are more convincible to be used in Lucas Kanade tracking (L-K Tracking).Then forward and backward L-K tracking are conducted on these points. Finally, the false tracking results would be filtered out by forward-backward error detection. As shown in Fig. 1, the forward tracking result of point ① is located as point ③. Meanwhile, point ③ gets corresponding point ① as backward tracking result. Point pairs ① and ③ would be kept as high-confidence tracking result. In contrast, point ② and ④ could not form such a one-to-one match and would be filtered out as an outlier in tracking procedure. Thus, our short term tracker could get better performance under noises and false matches, as compared in Fig.2 with L-K tracker.

Fig.2 (a-d) shows the tracking points' accuracy (yellow dots) and bounding box replacement at time $t+1$ between our short term tracker and other short term trackers.

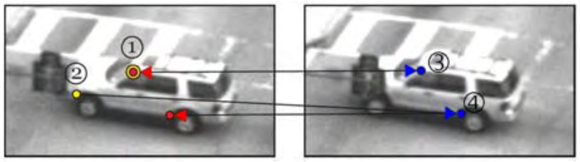

Fig.1 Backward forward error detection strategy（Blue arrows means forward tracking and the red ones means backward）. This figure comes from [1]

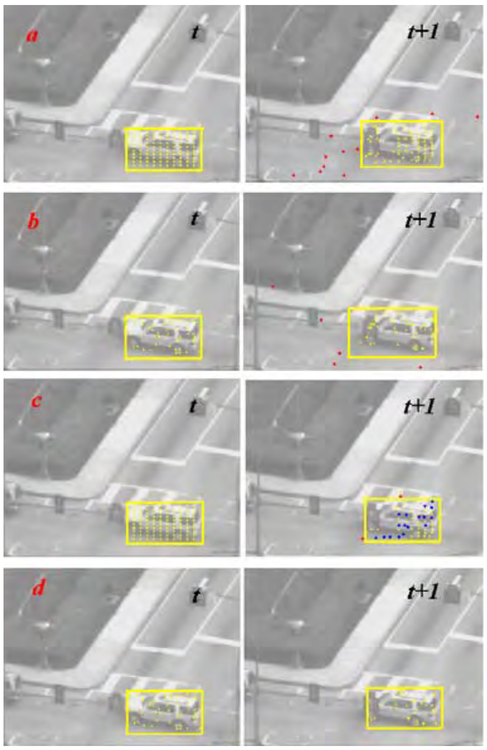

Fig. 2. Sample frames' tracking by (a) original Lucas Kanade tracker (b) tracking by corner points (c) tracking by forward backward error detection (d) tracking by bidirectional corner matching

From the tracking points results, we can see that: Fig. 2 (a) L-K tracker with average optical flow has the most outliers (denoted as red dots); Fig. 2 (b) tracking with corner points makes initial points more representative of object context, but noise points are much easier to influent the result by far outliers; Fig. 2 (c) Forward backward error detection avoids many outliers but has some points on the non-silhouette part(denoted as blue dots), which may induce drift away effect; Fig. 2 (d) the proposed method by us absorbs the advantages of all the methods mentioned above, making the initial points more representative as well as filtering out the bad effect coming from noise points. We can also demonstrate our short time tracker's accuracy by achieving the most preciseness in bounding box.

Corner points make our tracking points more texture aware; meanwhile bidirectional matching ensures outliers' elimination. For these reasons, our short term tracker is able to track the object more accurate.

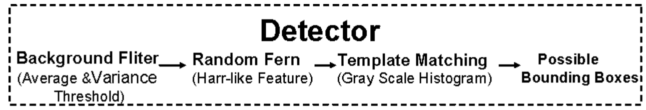

Fig. 3. Framework of detector with statistical filter and matching

III. STATISTICAL COGNITIVE AND TRAJECTORY SMOOTHNESS STRATEGY

The short term tracker is not strong enough to handle re-tracking problem after object disappearance. We adopt TLD's detector framework into our short term tracker but with different components, to decrease computational complexity. When an object reappeared after full occlusion, we can immediately capture it through statistical template filtering and matching. Since these appearance templates are relatively invariant to different poses and viewpoints, it would be expected to capture the reappearing object through fast template matching. With regard to unstable problem, we add trajectory cognitive algorithm in track-detect clustering, greatly reducing jitter errors under low confidence cases. Our detector structure goes like Fig. 3 shows.

A. *Statistical Feature Filter and Matching Algorithm*

As explained in [5], TLD detector uses sliding windows to captured candidate patches that have high similarity to target object. Such similarity is measured by statistical feature and harr-like feature in our algorithm. Statistical features like mean, variance and histogram contain rich information about target object. Moreover, they're also invariant to multiple appearance changes. To prove it, we take pedestrian as an example. We do experiment on pedestrian dataset from MIT [11], and get the distributions of gray scale mean and variance, which are shown in Fig. 4. Having sample variables $\{x_i\}$, the standard normal distribution approach is the maximum log-likelihood method(Equation (1)). Using Gaussian distribution approach, we formula the distribution of mean and value as the probability density Equation (2), in which μ is the expectation and σ the standard deviation. With the knowledge that about 95% of the values lie within two standard deviations [12], we take 2σ as watershed to obtain the specific mean and variance range. Using this knowledge, our detector adopts mean and variance threshold to filter out background. Such method helps us to save time to focus detector on more possible area. It also shows that mean and variance are not practical to distinguish pedestrians from each other since the same class of object share the same range of distribution of mean and variance.

$$\begin{aligned}\ln L(\mu,\sigma^2) &= \sum_{i=1}^{n} \ln f(x_i;\mu,\sigma^2) \\ &= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\end{aligned} \tag{1}$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{2}$$

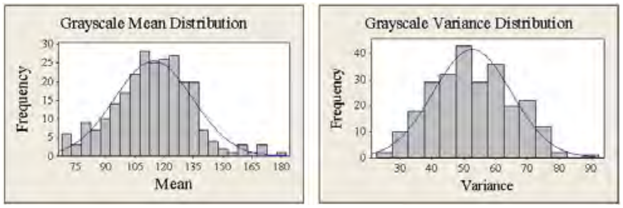

Fig.4. Distributions of grayscale mean and variance value on MIT pedestrian dataset with Guassian approach

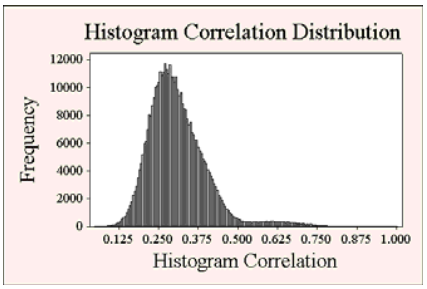

Fig.5. Distribution of Histogram Correlation Distribution value on MIT pedestrian dataset

Fig. 5 is the distribution of histogram correlation between each other on the same dataset. The correlation value almost all locate in low value around $[0.125, 0.5]$, showing the big difference between pedestrian to pedestrian. It proves the fact that histogram correlation is capable to detect different pedestrians. Experiments of statistical features on cars, faces and other class of object support our method as well.

Based on the knowledge above, we introduce the detector framework (Fig.3). Mean and variance threshold play as filter of background against object class, then local harr-like random fern to find possible bounding box, and finally histogram correlation as template matching to further select high confidence applicants.

Fig.6. If previous frame's trajectory is believable (left image with confidence 0.72) as well as the current frame's short term tracking module's confidence is above threshold(orange bounding box in right image with confidence 0.67), we set a neighbour range(yellow circle) for detector to find best object location (blue bounding box with confidence 0.78).

Fig.7. If previous frame's trajectory is not good (left image with the confidence 0.23), the detector performs global search by sliding window to find best object location (blue bounding box with the confidence 0.75).

B. Trajectory Smoothness Algorithm

Having bounding box applicants, we do clustering of candidate patches and decide the best position. A direct search would be a big computational burden for videos with large image sizes. In our algorithm, we first use trajectory cognitive to check trajectory confidence judgment. Global search is implemented only when the trajectory confidence is un-satisfied.

The tracker has got a bounding box as $\{BB^T, Con^T\}$ and the detector has obtained a series of the candidate bounding boxes $\{BB_i^D, Con_i^D, OP_i^D\}$, where BB means the bounding box, Con means the feature similarity confidence, and OP is the overlapping percentage with the tracking module. Subscript T represents the result of short term tracker, and D represents our detector. If the tracking module's result and previous final bounding box's confidence is higher than a threshold, we can use the trajectory consistency theory to find the most precise bounding box in a neighborhood range of trajectory. If the target disappear and appear again in the video, the short term tracking may drift away or fail to recover. At that time, we believe the previous trajectory and current frame's tracking result are unlikely to be trusted, and then the detector uses the global search for possible locations. Our strategy goes like Fig. 6 and 7 represent.

Fig. 6 shows the situation when trajectory is believable. The detector only do narrow search in the neighbor range of short time tracker; Fig. 7 is the worse situation that the previous trajectory is not effective, and then we deal with it with global search and clustering.

IV. EXPERIMENTS AND RESULT

We test our method on the challenging datasets which first used in [7]. These datasets includes severe light condition changes (David), camera jittering view (pedestrian 1), disappear and reappear problem (pedestrian 2, 3 and car). We've also conducted experiments in our real time captured videos. Those experiments all demonstrate the superiority of our method in the accuracy of long term tracking and the extension capability to the multi-pedestrian tracking in real world surveillance.

A. Accuracy of long term tracking

For the entire object tracking algorithm, Fig.7 shows some of the snapshots of our tracking performance on some challenging datasets. Our tracker is implemented on C/C++ platform, and the parameters are initiated and fixed for all experiments.

Table I shows the comparison between our tracker and other state-of-art approaches in terms of the number of correctly tracked frames. These other approaches are DTF [8], ET [9], MIL [10] and F-B [1]. For all 6 datasets, our tracker performs best in 4 of them, which include David, Jumping, Pedestrian 2 and Car. For the other two datasets, our tracker ranks the second (Pedestrian 1) and the third (Pedestrian 3).

These results demonstrate the performance of our tracking framework.

TABLE I

COMPARISON WITH STATE-OF-THE-ART APPROACHES IN TERMS OF THE NUMBER OF CORRECTLY TRACKED FRAMES.

Sequence	**Frames**	**DTF**	**ET**	**MIL**	**F-B**	**Our**
David	761	n/a	94	135	761	761
Jumping	313	313	44	313	170	313
Pedestrian1	140	6	22	101	140	134
Pedestrian2	338	8	118	37	97	335
Pedestrian3	184	5	53	49	52	49
Car	945	n/a	10	45	510	853
Best	-	1	2	1	2	4

B. Extension to multi-pedestrian tracking in real time surveillance video

Our tracker can track single object in complicated background, which shows its capability to track multiple pedestrians at the same time. Therefore, using same structure, we integrated similar independent tracker in our multi-tracking algorithm. And we achieve excellent performance as we can show in Fig.8.

In the upper images of Fig. 8, we simultaneously track 3 pedestrians from the high angle shot video. During the period, 3 pedestrian have been out of view for about 100 frames, and then reappear. Our tracker won't locate the wrong patches when they are out of view, and can redetect them precisely when they show out again. Even one or two pedestrians are out of view; the respective tracker will not mix with each other and jump to track another pedestrian.

The lower images of Fig. 8 are captured by surveillance camera in our university. Once defined the initial locations of three pedestrians, our tracker starts to learn the representative feature subspaces independently and track the targets stably and robustly in a long term.

V. CONCLUSION

We have presented a novel framework for object tracking. The basic framework is based on the TLD's concept, which includes a short term tracker, an object detector and an online model. We have improved the efficiency of the short term Lucas Kanade tracker by bidirectional corner matching; Meanwhile, we have introduced trajectory cognitive algorithm to replace the complicated classification strategy in online modeling; we have also extend our method to multiple pedestrians tracking in real world. Extensive experiments and comparisons with other state-of-art methods have validated the robustness of our tracker. This work would be extended in future to tackle activity understanding task.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China 60902073, 973 project 2010CB731401 and STCSM project 10511500802 and 12DZ2272600.

REFERENCES

[1] Z. Kanlal, K. Mikolajczyk, J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," in *ICPR*, Aug. 2010, pp.2756-2759.

[2] B. Lucas and T. Kanade,"An iterative image registration technique with an application to stereo vision," *IJCAI,* 1981, *pp*. 674–679,

[3] C. Harris and M. Stephens (1988). "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[4] J. Shi and C. Tomasi, "Good Features to Track," *9th IEEE Conference on Computer Vision and Pattern Recognition*, 21-23 Jun 1994, pp.593-600.

[5] Z. Kanlal, K. Mikolajczyk, J. Matas, "Online learning of robust object detectors during unstable tracking," *3rd On-line Learning for Computer Vision Workshop*,2009, pp.1417-1424.

[6] Z. Kanlal, K. Mikolajczyk, J. Matas, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," *CVPR,* June 2010, pp.49-56.

[7] Q. Yu, T. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," *ECCV*, 2008, Vol. 5303/2008, pp. 678-691.

[8] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *PAMI*, vol.27, no.10, pp.1631-1643, Oct. 2005.

[9] S. Avidan, "Ensemble tracking," *PAMI*, vol.29, no.2, pp. 261–271, 2007.

[10] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," *CVPR*, June 2009, pp. 983 - 990.

[11] *MIT pedestrian dataset*. Available: http://cbcl.mit.edu/software-datasets/PedestrianData.html

[12] M. S. Braverman, "Characteristic properties of normal and stable distributions," *Theor. Probab. Appl. 30(1985)*, pp. 465–474.

Fig.7. Snapshot of single object tracking from dataset David, jumping, Pedestrian 1, Pedestrian 2 and car (From up to down).

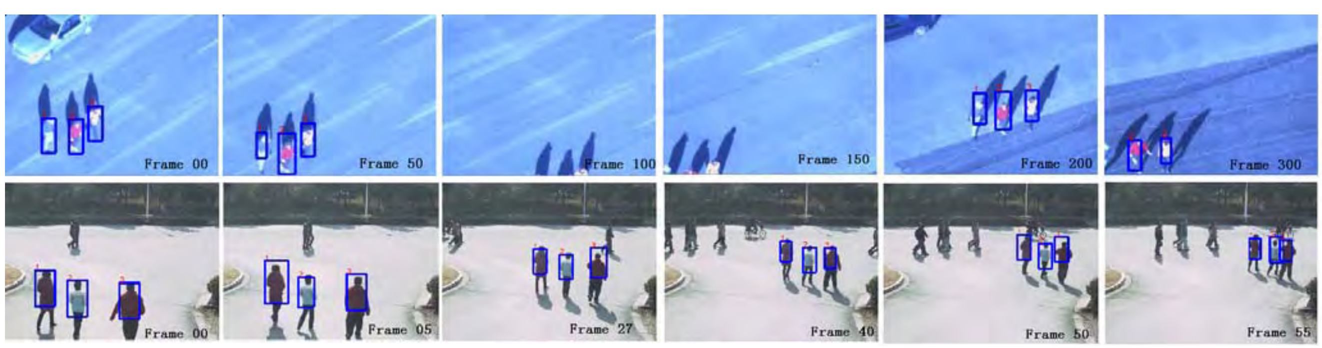

Fig.8. Multiple pedestrian tracking on dataset Pedestrian 2 and our real world video.