An Improved TLD Visual Tracking and Detection Algorithm Based on Depth Image Information

Jialiang MAO[1, a], Xiaorong SHEN[2, b]

[1] School of Automation Science & Electrical Engineering of Beihang University, Beijing, 100191, China

[2] School of Automation Science & Electrical Engineering of Beihang University, Beijing, 100191, China

[a]E-mail: maojialiang@outlook.com, [b]E-mail: Sarah_shen@buaa.edu.cn

Keywords: Visual Tracking; Tracking-Learning-Detection; Depth Information; Blob detection

Abstract. Tracking-Learning-Detection(TLD) algorithm has the advantages of high speed, high accuracy of tracking rate and self-detection mechanism. However the sensitivity to illumination variation and background clutter leads to drift even miss for TLD algorithm under the complex environment. An improved TLD visual tracking algorithm based on depth information is proposed, which is improved from three aspects. The first is that a foreground extraction algorithm based on denoised depth data is adopted to extract the region of interest. Second, a sophisticated detection of blob approach is used to narrow down the searching area. And the third is that the non-maximal suppression strategies are applied to optimize the result. The experimental results show that the time complexity of proposed is decreased and the robustness is upgrade under challenging circumstances.

Introduction

Recently, tracking-by-detection approaches have shown to provide excellent tracking performance. These methods work by taking target localization as a classification problem. The decision boundary is obtained by learning a discriminative classifier online using image patches from both the target and the background [1]. Babenko et al. [2]pose the tracking problem within the multiple instance learning (MIL) framework and develop an online algorithm, improving the flexibility of finding a definite bounding box by labeling and passing overlapping samples of the target to the learning model. Santner et al. [3]employ a PROST approach that combines an optical-flow-based tracker, a template model and an online learning mechanism to deal with appearance changes. Bolme et al. [4] exploit MOSSE (minimizing the output sum of squared error) method to search an adaptive correlation filter. Hare et al. [5] put binary classification problem as a prediction of structured output, using kernelized structured SVM to directly estimate location of the target.

In 2012，Kalal et al. [6] propose an algorithm called TLD (Tracking-Learning-Detection） which based on Lucas-Kanade [7] optical flow tracking algorithm to train the detector and updates the learner if the discovered patch is resemble to the original one. TLD reinitializes the detector only when LK based tracker failed, which leads the robustness of long-term tracking and more accurate results. However, the problem of Kalal's work is that the bounding box of target may drift or even not appear under the circumstance of long-term tracking or high speed movement of the

target.

This paper exploits color video sequences and the corresponding depth information obtained from Kinect sensor to improve the tracking robustness and reduce the processing time of TLD algorithm. Improvements mainly include three aspects: First, using the denoised and filtered depth information to extraction foreground. Second, the blob detection algorithm is adopted to refine the foreground objects. Third, non-maximum suppression strategies and resizing bounding box method are applied to modify the bounding box of target.

The Fundamental Principles of TLD

TLD algorithm is mainly composed by three modules: the tracker, the detector and the learner. As shown in Fig.1: Tracker and detector operate independently, when the tracker finds the target with high confidence level , then the object of interest is regard as the ultimate goal. When tracker fails, the result from detector which has the highest confidence level is chosen as the ultimate one. Certain criteria must to be met to process the learning procedure, then the model of tracker and part of detection stage are trained online.

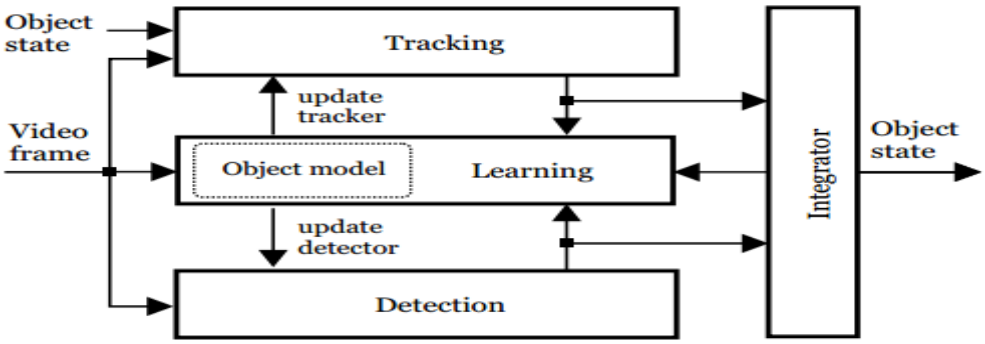


Fig.1 Block diagram of the TLD framework

The tracker is designed for tracking sequential movements between frames. It only works when the object of interest appears in frames. TLD supposes that a good tracking algorithm must have forward-backward consistency, means that the tracking trajectory should be the same no matter whether tracking is according to the time flow or reversely. Based on this hypothesis, TLD algorithm proposes a tracker called Media Flow [8]. With forward-backward errors rule and the errors estimated with normalized cross correlation (NCC) [9], outliers are filtered out and the rest of all points are used for estimate the bounding box motion and size.

The detector is aimed at reinitialize the tracker that fails to recover losing target. In the first frame, a target bounding box is selected manually, then the detector applies sliding-window approach [10], which classifies all predefined sub-windows based on a cascade classifier. The cascade classifier is composed of three small classifiers: patch variance, ensemble classifier and nearest neighbor classifier [6]. As shown in Fig. 2:

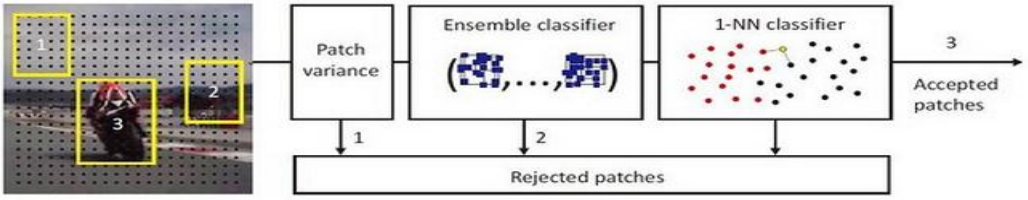


Fig.2 Block diagram of the detector

The Learner [11] is proposed to combine the outputs of tracker and detector into an ultimate bounding box. When the object of interest is localized, 10 positive samples are chosen from sub-windows that are closest to the ultimate bound box, then warped these samples to 200 synthetic positive patches at first frame, after that, 100 warped synthetic positive patches are generated and passed to the target model of tracker as well as detector's. Meanwhile, negative samples that far from ultimate bounding box are selected (overlap < 0.2) to update the target model of detector.

An improved TLD visual tracking algorithm based on depth information

TLD has the advantages of high speed, high accuracy of tracking rate and self-detection mechanism. But in practice, for a target in high-speed movement or under a long-term tracking, the bounding box might be drift away or lose their target, see in Fig. 10(a), 11(a). This problem is mainly due to the following two reasons:

1） For fast moving objects, background and illumination change dramatically between frames, which violates the spatial continuity estimation of optical flow, so trackers based on optical flow drift much more quickly for tracking speedy targets, in other words, at slower frame rates;

2） When do long-term tracking, the problematic is that detector that applies sliding-window approach to select window with highest confidence level as the final bounding box might ignores local optimum values, which might lead to poor result of re-initialize and weaken the robustness of whole algorithm.

Based on the foregoing analysis, keys for the purpose of improving performance of TLD are to satisfy the optical flow assumptions and restraint drifting from re-initialization. In this paper, with the depth information of RGB images from Kinect sensor, the modifications mainly carry on from three aspects of TLD:

1）Depth information denoising : Two different noise filtering methods are introduced, and finally both methods are combined to improve the accuracy of depth information;

2）Foreground extraction: A blob detection approach is applied to foreground extraction, with a more refined and smaller region of interest (ROI), less processing time is needed for detection stage;

3）Bounding box modification: Non-maximal suppression strategies and a modification method based on depth information are proposed, which can generate a higher confidence level of bounding box.

Depth Information Denoising

In general, the depth accuracy of newly launched Kinect V2 sensor is extremely close up to millimeter in its view distance [12], recording 1080p video at 30fps. In practice, the depth image made by depth arrays from Kinect can be seen in Fig. 5(a), noises in the video sequence manifest themselves as black spots gathering in the edge of objects and continuously popping in and out of the picture, mainly caused by depth values absent and random noise error of Kinect.

Given inner band threshold as 3 and outer band threshold as 7, the pixel filtering method [13] is proposed to handle the noise. As shown in Fig.3, pixel filtering method scan the entire pixels for searching zero values to determine the candidates for filtering which are black spots displayed in the video sequences. Second, record all the value of pixels next to a certain candidate. Only two bands around the candidates are cared and the amounts of non-zero values in each band are calculated. Next, compare these values to a predefined threshold for each band to determine whether the candidate should be filtered. If the threshold for each band is not satisfied, then the statistical mode of all the non-zero values will be applied to the candidate, otherwise leave it alone.

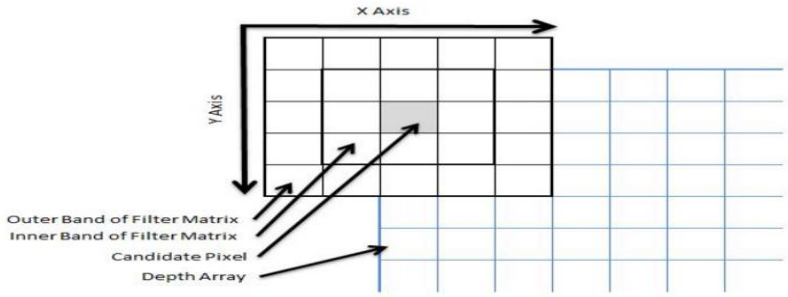


Fig.3 Pixel Filtering Method

Furthermore, weighted moving average method [13] is adopted to moderate the flickering issue after pixel filtering. Save most recent N number of depth frames in a queue, then weighted the highest score to the newest depth frame and the least to the oldest. For the K-th depth frame $I_k(i, j)$, the proposed weighted average mean is defined as follow:

$$I_k(i, j) = \frac{\sum_{i=k-n+1}^{k} w_i * I_i(i, j)}{n} \tag{1}$$

Where $I_k(i, j)$ is the depth value of pixel at point (i,j) in the K-th depth frame , w_i is the weighted value of the i-th depth frame and N is the number of depth frames we presetted. Set N to 4 and a random video sequence samled by Kinect are denoised. Fig.4 depicts the comparison before and after the denoising process:

Fig. 4 denoised result comparison (the left is Original depth image, the right is denoised result with pixel filtering and weighted moving average method)

From Fig.4, the ceiling and the top right corner have a large number of randomly distributed black spots in the original image, which is "flicker effect" in depth video sequence. In addition, the body and the rear potted plant display noticeable black edge on their contour. Combined with the mentioned approaches, the black spots in the ceiling and the black edge on the contour of objects are greatly removed.

Foreground Extraction Based on Depth information

Foreground extraction based on depth information is divided into two steps: First, exploiting the initial bounding box to do coarse background segmentation; second, using blob detection to extract ROI.

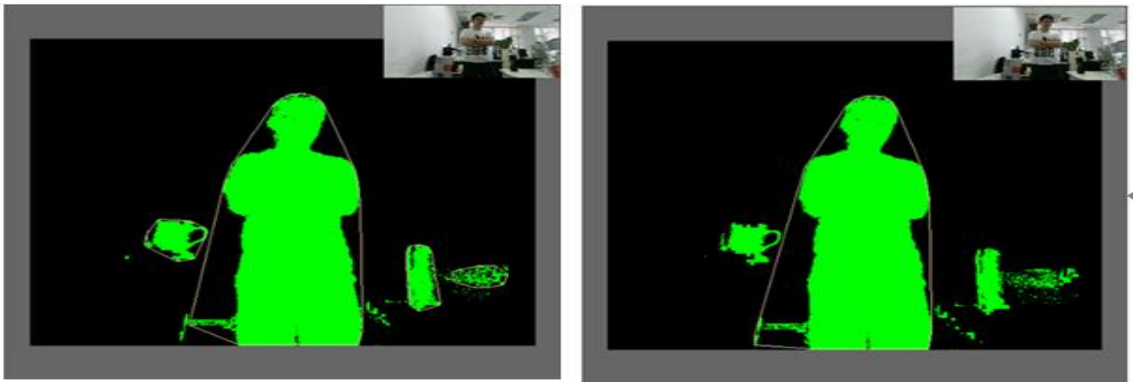
Coarse background segmentation method utilizes the property of object depth continuity to extend or divide objects of interest. Given that initial bounding box must be selected manually, mean Shift approach [14] is adopted to find out the point of maximum density in the current bounding box, D_i refers to the corresponding depth value of the point, then a threshold range to [D_i-700 , D_i+700] is set to extract the interest area. Actually, it might contain a number of irrelevant objects after coarse background segmentation depending on shooting environment, the unrelated blobs is needed to be excluded.

With blob detection method [15], foreground pixel is counted as a starting point for an outer contour, then search clockwise to mark the neighboring pixels in turn, and completed the entire outer contour marking till back to the start point.

The area of each blob is calculated and the blobs whose areas are beyond the dynamic threshold range are filtered out. The dynamic threshold T_k is defined as

$$T_k = 1.2^a S_{k-1} \tag{2}$$

Where parameter a is adjustable, S_{k-1} is the area of bounding box in k-1-th frame. Set a =-2, 2, the foreground extraction result is shown in Fig.5.



(a) Blob detection (b) blob detection with filtering methods

Fig. 5 Blob detection and filtering based on depth information (all pixels which depth values range between [D_i-700 , D_i+700] are shown)

Non-maximal Suppression

Processed by detector cascade classifiers，sub-windows with highest confidence is selected as the final bounding box of detector. However, according to Blaschko [16], it is problematic to consider only sub-window with the highest confidence, which can lead to other local maxima being ignored. For long-term tracking, the final bounding box may be wrongly targeted due to accumulated drift. Instead, it is applicable to adopt non-maximal suppression strategies that take all relevant local maxima into account.

As method presented in [17] that compared highest confidence bounding box with several local maximal, in Fig.6 ， B1 denotes the area of one bounding box and B2 represents the other one, I is the intersection between two boxes. The formula [18] is applied：

$$\text{overlap} = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{I}{B_1 + B_2 - I} \tag{3}$$

Hierarchical clustering algorithm [19] is adopted to determine the sub-windows left by cascade classifier whether they belong to the same cluster, and then integrate sub-windows in the same cluster into a single target box. With Eq.3, the overlap area of two bounding boxes is calculated.

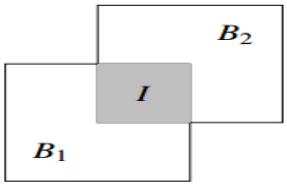


Fig. 6 Overlapping Between Two Bounding Boxes

Resizing bounding box with depth information

The resizing of bounding box approach is based on the pinhole imaging principle. An object

area on the imaging plane is proportional to the squared Euclidean distance between the object to the camera lens plane：

$$\frac{S_{\mathrm{k}}(M)}{S_{k-1}(M)}=\frac{{D_k}^2(M)}{{D_{k-1}}^2(M)} \tag{4}$$

In Eq. (4), $S_{\mathrm{k}}(M)$ and $S_{\mathrm{k-1}}(M)$ are represent respectively the areas of bounding box in k-th frame and K-1-th frame，D_{k} and $D_{\mathrm{k-1}}$ corresponding to the depth value of k-th and k-1-th frame. After equivalent transformation, Eq. (4) can be written as:

$$S_{\mathrm{k}}(M)=S_{k-1}(M)\frac{{D_k}^2(M)}{{D_{k-1}}^2(M)} \tag{5}$$

In practice, the depth value of centroid location of a certain bounding box is close to its neighboring sub-windows', therefore the size of bounding box in k-th frame can be approximate obtained according to Eq.(5), which is considered to be the ultimate output of the detector.

Experiments

Two video sequences of different scenarios are captured by Kinect, which are named as "Ceramic cat" and "Wrist". The former contains background clutter, scale variation; the latter includes fast moving, in-plane and out-of-plane rotation. The experimental platforms configuration are Intel Core i7-2600 (3.4 GHz), 8GB memory and Kinect for Windows V2, the development environments are Visual Studio 2013, OpenCV 2.4.10 library. The default video resolution of Kinect is 1920*1080, two sequences consist of 483 individual frames and 380 individual frames respectively.

Tracking moving target under background clutter scenario

The tracking of ceramic cat is shown in Fig.7， as we can see background switch quickly from green plants to man's head, and finally move to the transparent window. The yellow circle and red rectangle represent respectively the output location of original TLD and proposed method. Fig.7(a)~(b), from left to right, top to bottom corresponding to the video sequence frame at 53,78,199,213,275, 433.

(a) Results of TLD algorithm

（b）Results of algorithm proposed in this paper

Fig.7 Target with Background Clutter

The results show that although TLD can handle with scale variation, a few details are lost, which can lead to drifting or even tracking on the wrong target after a long time tracking. Besides, object of interest is not recognized when target background switch to others. Since the proposed method based on depth information, which sweeps out irrelevant backgrounds, makes the algorithm not sensitive to the background clutter and track smoothly.

Tracking moving target with high speed and rotations

In Fig.8, the tracking target is right wrist of a man，the movements is fast combined with in-plane and out-of-plane rotation. In Fig.8(a)~(b), from left to right, top to bottom corresponding to the video sequence frame at 39,84,133,206,240,255.

(a) Results of TLD algorithm

（b）Results of algorithm proposed in this paper

Fig.8 Target with High Speed and Rotations

TLD often misses target and performs worse on this challenging scenario. The modified method can reduce the detection time owing to adopt the foreground extraction strategy at the beginning of detector. Meanwhile, the non-maximal suppression strategies and modification of bounding box to detection stage improves the drifting issue in long-term tracking has been improved (see Fig.8(b) and table 4.1).

Define recall as follows:

$$recall = \frac{TP}{TP + FN} \tag{6}$$

Precision is defined as

$$precision = \frac{TP}{TP + FP} \tag{7}$$

Where true Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) are statistically calculated.

Compare performance between improved TLD algorithm before and after and results are shown in Table 4.1.

Table 4.1 Performance Comparison

	TLD	Ours
China Cat	**0.59**/0.53	0.53/**1.00**
Wrist	0.45/0.97	**0.91**/**1.00**

Conclusion

In view of the poor performance of TLD in background clutter scenario and drifting in long-term tracking, an improved TLD tracking method based on depth information is proposed. Since the depth information is immune to color and light variation, makes it as an excellent point to enhance the TLD tracking algorithm. In the early stage of the detector cascade classifier, a foreground extraction method is introduced, meanwhile, non-maximal suppression strategies and modification of bounding box based on depth information are also brought into the last stage of detector, which not only reducing computation time, but also further improve the robustness of tracking.

However, the paper still does not take orientations of target into accounts, the proposed approach is also subject to the range limitation of the Kinect depth sensor. The next step of the research is to establish affine transformation model to calculate the orientation of object.

Acknowledgement

This work is financed by the NSF of China (No.61203186).

References

[1] Danelljan, M., H ¨ager, G., Khan, F.S., Felsberg, M. *Accurate scale estimation for robust visual tracking*. In: Proceedings of the British Machine Vision Conference BMVC (2014)

[2] B. Babenko, Ming-Hsuan Yang, and S. Belongie. *Visual tracking with online multiple instance learning*. *CVPR Workshops*, pages 983-990. IEEE, June (2009)

[3] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. *PROST: Parallel robust online simple tracking*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 723-730. IEEE, June (2010).

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Yui M. Lui. *Visual object tracking using adaptive correlation filters*. In CVPR, (2010)

[5] S. Hare, A. Saffari, and P. H. S. Torr. *Struck: Structured output tracking with kernels*. In IEEE International Conference on Computer Vision, pages 263-270. IEEE, Nov. (2011)

[6] Z. Kalal, K. Mikolajczyk, and J. Matas. *Tracking-learning-detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(7):1409 -1422, july (2012)

[7] B. D. Lucas and T. Kanade. *An iterative image registration technique with an application to stereo vision*. In *Proceedin*gs of the International Joint Conference on Artificial Intelligence, pages 674-679,(1981)

[8] Z. Kalal, K. Mikolajczyk, and J. Matas. *Forward-Backward Error: Automatic Detection of Tracking Failures*. In International Conference on Pattern Recognition, pages 23-26,(2010).

[9] J.P. Lewis, &ldquo,*Fast Normalized Cross-Correlation*,&rdquo, Vision Interface, (1995).

[10] P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*.In IEEE Computer Society Conference on Computer Vision and Pattern Recognition.(2001)

[11] Z. Kalal, J. Matas, and K. Mikolajczyk. *P-N learning: Bootstrapping binary classifiers by structural constraints*. In 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 49-56. IEEE, June (2010)

[12] Information on https://en.wikipedia.org/wiki/Kinect.

[13] Information on http://www.codeproject.com/Articles/317974/KinectDepthSmoothing

[14] Fukunaga, Keinosuke; Larry D. Hostetler (January 1975). *The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition.* IEEE Transactions on Information Theory (IEEE) 21 (1): 32-40.

[15] Chang F, Chen C J, Lu C J. *A linear-time component-labeling algorithm using contour tracing technique*[J], Computer Vision and Image Understanding, (2004),93(2): 206-220

[16] M. B. Blaschko. *Branch and Bound Strategies for Non-maximal Suppression in Object Detection*, volume 6819 of Lecture Notes in Computer Science, chapter 28, pages 385-398. (2011).

[17] Lindeberg, Tony *Edge detection and ridge detection with automatic scale selection,* International Journal of Computer Vision, 30, 2, pp 117-154, (1998)

[18] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. *The pascal visual object classes (VOC) challenge*. International Journal of Computer Vision, 88(2):303-338, June (2010).

[19] F. Murtagh. *A survey of recent advances in hierarchical clustering algorithms*. *The Computer Journal*, 26(4):354-359, Nov. (1983).