LETTER

Online HOG Method in Pedestrian Tracking

Chang LIU[†a)], ***Student Member***, **Guijin WANG**[†b)], ***Member***, **Fan JIANG**[†c)], ***Student Member***, ***and*** **Xinggang LIN**[†d)], ***Nonmember***

SUMMARY Object detection and tracking is one of the most important research topics in pattern recognition and the basis of many computer vision systems. Many accomplishments in this field have been achieved recently. Some specific objects, such as human face and vehicles, can already be detected in various applications. However, tracking objects with large variances in color, texture and local shape (such as pedestrians) is still a challenging topic in this field. To solve this problem, a pedestrian tracking scheme is proposed in this paper, including online training for pedestrian-detector. Simulation and analysis of the results shows that, the proposal method could deal with illumination change, pose change and occlusion problem and any combination thereof.

key words: *pedestrian tracking, HOG detector, pose change, online training*

1. Introduction

In recent years, pedestrian detection and tracking has been an active research area. Tracking is essentially the problem of finding the object states (position, scale and other parameters characterizing the object) from the observed image sequence. The task is challenging because of:

- Background clutter which makes it difficult to distinguish the object from the background.
- Illumination changes make the target appearance complicated.
- Complicated pedestrian appearance changes, especially when the pedestrian's pose change.
- Part of the target disappears because of occlusion between pedestrian or occlusion by background.

Many proposed tracking algorithms have tried to solve the above problems by building a generative model to describe the target's visual appearance. Selecting discriminative features is always an important problem in object tracking. Color, edge (or gradient), texture feature have been widely used in tracking algorithms. For object tracking, kernel-based tracking [2] and particle filter [3] are two effective and popular algorithms. Many researchers have improved upon the original algorithm since it was first introduced. Kernel-based tracking through scale space [4] has been added to overcome the shortcomings during scale change. Multi-kernel [5] and asymmetric kernel [6] have also improved the algorithm performance. This type of algorithm has the ability of finding the most similar possible position or scale in the neighbor region. However, it requires the use of a discriminating model to supervise the tracking.

Predefined features integrating on-line feature selection [1] is also an attractive development direction. The usage of on-line features and on-line training is useful for dealing with drastic pose changes. Y. Li [7] followed the ideas of using an independent detector to guide the search of an existing tracker when target motion becomes unpredictable. Grabner [8] proposes a novel on-line AdaBoost feature selection method. A semi-supervised learning algorithm using a co-training framework is proposed [9] to solve the puzzle of self-training.

There are still two main weaknesses that exist in the on-line tracking systems: 1) The combination of illumination changes, pose changes, and occlusion would result in tracking failure. 2) The performance of the online system is not always robust. An improperly updated system will hurt the performance of the tracking system.

To solve these problems, we present a particle filter tracking system with online training. We opted not to use background subtraction in order to deal with a moving camera. The feature we used is HOG (Histogram of Oriented Gradient) [10]. As we intend to solve the problem of serious pose change, we developed an online training algorithm for HOG detector. Therefore, our work focuses on two aspects: offline pedestrian-detector and online training for pedestrian-detector.

The rest of the paper is organized as follows: Section 2 gives a brief introduction of using Gentle Boosting [11] to train a pedestrian detector. The proposed online training method is presented in Sect. 4. Implementation details and experiment results are presented in Sect. 3. The conclusion and future works are in Sect. 5.

2. Pedestrian-Detector

This section describes a pedestrian-detector's training method and the corresponding feature. First, we resize the image patch for training to a uniform size of 64×128 pixel. We used Gentle-Boosting [11] method with HOG feature [10] to build a pedestrian detector. To make the on-line training more robust, we select one cell bin from each HOG

Manuscript received November 5, 2009.

Manuscript revised February 2, 2010.

[†]The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

a) E-mail: chang-liu06@mails.tsinghua.edu.cn

b) E-mail: wangguijin@tsinghua.edu.cn

c) E-mail: jiang_f99@mails.tsinghua.edu.cn

d) E-mail: xglin@tsinghua.edu.cn

DOI: 10.1587/transinf.E93.D.1321

Table 1 Overview of Gentle-Boosting algorithm.

1	Start with weights
	$\omega_i = \frac{1}{N^+}, i = 1, 2 \ldots, N^+,$
	$\omega_i = \frac{1}{N^-}, i = N^+ + 1, N^+ + 2 \ldots, N^+ + N^-,$
2	for $m = 1 \ldots M$
	(a) for $k = 1 \ldots M$
	Fit the regression function $f_{m,k}(x) = a_k x + b_k$ by weighted least-squares of y_i to x_i with weight w_i and feature k.
	End.
	(b) Set a best regression function $f_{m,k}(x) = a_k x + b_k$ as $f_m(x)$
	(c) Update $F(x) = F(x) + f_m(x)$
	(d) Update $\omega_i = \omega_i * exp(-y_i f_m(x_i))$
	(e) Renormalize $\omega_i = \omega_i / (\sum \omega_i)$
	End

cell to use as a weak classifier; since we assume that there is a dominating direction in every cell, a single cell bin is sufficiently representative of a whole cell. The cells are sized from 8×8 to 64×128; the height-width proportion can be 1:1, 1:2 or 2:1. We use each cell bin as a weak classifier. So we can get 53928 weak classifiers from 5992 cells in every patch. Meanwhile, we use 3542 manually cut pedestrian patches as positive samples, while 4250 other patches as negative samples.

Using Gentle-Boosting for classification provides a scheme for generalize models from high dimensional feature space. Boosting classifiers of the form $F(x) = \sum f_k(x_k)$ can be learned from the training data $(x_i, y_i), x_i \in R, y_i \in \{+1, -1\}, i \in 1, 2 \ldots, N$

In our experiment, we set M as 40 and K as a smaller number than the total feature number of 53928 in Table 1. Because of the large amount of weak classifiers to consider, we opted to use a classifier sampling method to reduce computation time at different stage of Gentle-Boosting, the difficulty of selecting a new classifier is quite different. In the earlier stage, the selecting is easy. In the later stages, selecting a new classifier is difficult. So we use a different percentage in classifier sampling.

In tracking, the general location of the pedestrian is usually known; therefore, the system can tolerate a pedestrian detector with high false alarm provided the detector also has high accuracy. In our system, the pedestrian detector uses 40 weak classifiers. The false alarm is 5.0% while precise detection is over 95.1%.

3. Online Training for Pedestrian-Detector

This section describes an online pedestrian-detector's training method and the corresponding training time and strategy. We utilize an online Gentle-Boosting to solve the problem of posture variation and partial occlusion. As shown in Fig. 1, the online Gentle-Boosting include two steps. In the first step, discard one of the weak classifier from the offline boosting result; in the second step, train new weak classifier from the feature pool and add it into weak classifier sequence.

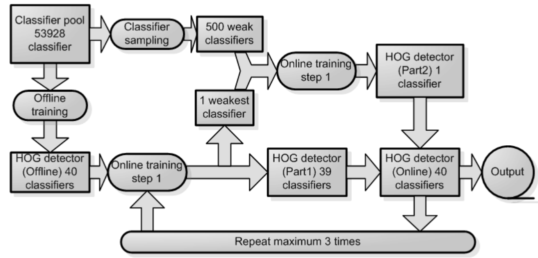


Fig. 1 Online training framework.

Fig. 2 Online training samples.

3.1 Online Samples

In every frame, we get 10 positive samples (as shown in Fig. 2) and 20 negative samples. All the online samples are selected as the same size as the tracking result of the current frame. The centers of online positive samples are randomly located in a small circle whose center is the center of the tracking result of the current frame. The radius of the circle is manually set to 5 pixels. The centers of online negative samples are located around the tracking result. The center of it is in a rectangle area. The patches are non-overlapping with the tracking target.

3.2 Parameter Update Time

We checked the pedestrian detector every 2 frames. If the pedestrian detector can correctly identify a certain percentage of the online samples, we do not update the parameters of online pedestrian detector. If the pedestrian detector degrades, the boosting process is activated. If the performance of newly trained detector is still not good, the online training process will continue to further improve the detector. After the detector performance returns to a satisfying level, online training module will become dormant again. Through the use of such a strategy, the method solves the dilemma between fast pose change and almost no pose change for a long time.

3.3 Online Boosting

We also control the number of weak classifier to be updated. According to the percentage of online samples that are correctly identified, the number of classifiers to be updated can vary from one to three. If three weak classifiers' replacement still can not correctly identify the online samples, we will not introduce more weak classifiers in because too many updates at the same time will seriously affect the system robustness.

For every weak classifier replacement, we have two

Table 2 Overview of online Gentle-Boosting algorithm.

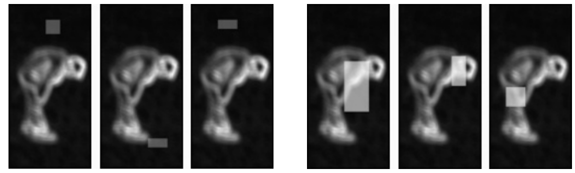
1 Start with weights
$\omega_i = \frac{1}{N^+}, i = 1, 2 \ldots, N^+,$
$\omega_i = \frac{1}{N^-}, i = N^+ + 1, N^+ + 2 \ldots, N^+ + N^-,$
2 for $m = 1, 2, 3$
(a) for $k = 1 \ldots 40$
Set the weak classifier function
$f'_{m,k}(x) = -f_{m,k}(x) = (-a_k)x + (-b_k)$
End.
(b) Set a best regression function $f'_{m,k}(x)$'s k as k^*
(c) Update $F(x) = F(x) - f_{m,k^*}(x)$
(d) Update $\omega_i = \omega_i * exp(+y_i f_{m,k^*}(x_i))$
(e) Renormalize $\omega_i = \omega_i / (\sum \omega_i)$
(f) for $k = 1 \ldots K$
Fit the regression function $f_{m,k}(x) = a_k x + b_k$ by weighted least-squares of y_i to x_i with weight w_i and feature k.
End.
(g) Set a best regression function $f_{m,k}(x) = a_k x + b_k$ as $f_m(x)$
(h) Update $F(x) = F(x) + f_m(x)$
(i) Update $\omega_i = \omega_i * exp(-y_i f_m(x_i))$
(j) Renormalize $\omega_i = \omega_i / (\sum \omega_i)$
End

steps. In the first step, we discard the weakest classifier out of 40. To find the classifier to discard, we define 40 reverse weak classifiers that use same feature but have opposite slope and opposite y-intercept of the linear regression functions. Choosing the best performing reverse weak classifier is the same selecting the weakest performing classifier. In the second step, we randomly sample 500 weak classifiers from all 53928 possible choices, plus the one discarded in the first step to form a pool of 501 weak classifiers. The replacement classifier is selected from this pool. As result, we again obtain a new detector with 40 weak classifiers. A detailed description is shown in Table 2. This discard-replacement step can be repeated at most three times in order to achieve the training objective (i.e. sufficient percentage of properly identified on-line samples).

4. Implementation and Experiment Results

4.1 Data and Experiment Setup

We have used two data sets to test our method. These include both indoor and outdoor contexts, varying illumination and resolution conditions, and pedestrian of various poses and occlusion circumstance. The first database is taken from the CAVIAR video dataset [12]. The second database is taken by us and contains a lot of pose change. All experiments run in about 2.5 fps on a standard 2.33 GHz PC with 2 GB RAM and a video resolution of 640×480. The computation time is mainly limited by the number of particles and the online training frequency and not by the video resolution.



(a) (b)

Fig. 3 Online (a) discarded and (b) selected weak classifiers.

Table 3 Experiment results.

Database	Correct rate with only offline detector	Correct rate with online learning
CAVIAR	77.2%	80.2%
Our database	64.3%	72.4%

4.2 Results

We manually assign the initial rectangles of pedestrians. The height-to-width ratios of these rectangles remain the same in subsequent tracking. We manually calibrated the position of pedestrians in the video sequence. And use the overlap rate to evaluate the accuracy of tracking.

$$overlap\ rate = \frac{area(tracking\ rect \cap Calibration\ rect)}{area(Calibration\ rect)}$$

If the overlap rate is over 0.8 in 90% of the frames, we think this pedestrian is being tracked correctly. Under this criterion, we evaluate our algorithm in the two data sets we used. Comparison example frames from the result videos is given in Fig. 4. The results in summarized in Table 3.

As shown previously in Fig. 3, each weak classifier is determined by local information; this allows us to deal with pose change problem in an elegant way. On the one hand, online samples during tracking give us the information of the current pedestrian. On the other hand, online learning module put this information into some weak classifiers which can be used in the following tracking process. Due to the characteristic of local description, the online learning method is also good for occluded situations. Figure 5 shows the tracking results when occlusion by background or other pedestrian occurs.

5. Conclusion

In this paper we propose a pedestrian tracking scheme with a pedestrian detector and its online updating method. Our algorithm employs a pedestrian detector in tracking framework, and the algorithm can continually improve the pedestrian detector by online boosting method. This makes the tracking more robust, especially in the condition of drastic pose change. Experimental results show that most examples on pose change and occlusion are resolved through online training method. Future work is aimed at adding relative position information into online training method to more robustly handle partial occlusion situations.

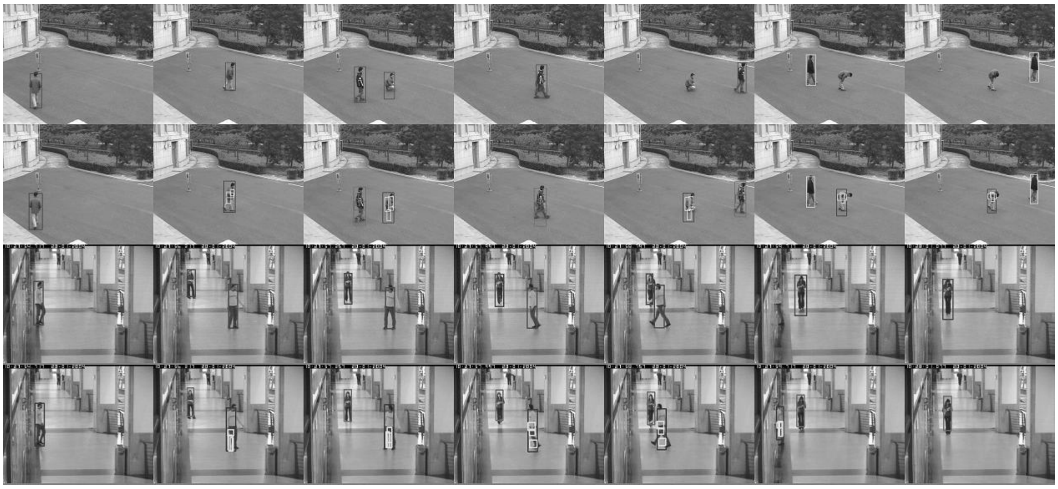


Fig. 4 Performance comparison: with only offline detector for (a) and (c); with online learning for (b) and (d).

Fig. 5 Tracking result in occluded circumstance.

Acknowledgement

We acknowledge with thanks that the research was supported by Sony China Research Lab.

References

[1] J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.2, pp.1037–1042, 2005.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. Pattern. Anal. Mach. Intell., vol.25, no.5, pp.564–577, 2003.

[3] M. Isard and A. Blake, "CONDENSATION-Conditional density propagation for visual tracking," Int. J. Comput. Vis., vol.29, no.1, pp.5–28, 1998.

[4] R. Collins, "Mean-shift blob tracking through scale space," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.2, pp.II-234–240, 2003.

[5] G. Hager, M. Dewan, and C. Stewart, "Multiple kernel tracking with SSD," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.790–797, 2004.

[6] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.1–6, 2007.

[7] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans," IEEE Trans. Pattern Anal. Mach. Intell., vol.30, no.10, pp.1728–1740, 2008.

[8] H. Grabner and H. Bischof, "On-line boosting and vision," Proc. IEEE Int. Conf. Comp. Vision Pattern Recognition, vol.1, pp.17–22, 2006.

[9] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," Proc. IEEE Int. Conf. Computer Vision, vol.1, pp.1–8, 2007.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proc. IEEE Int. Conf. Comp. Vision Pattern Recognition, vol.1, pp.886–893, 2005.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," The Annals of Statistics, vol.28, no.2, pp.337–374, 2000.

[12] CAVIAR: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/