Counting People in Crowded Scenes by Video Analyzing

Zebin Cai[1], Zhu Liang Yu[1], Hao Liu [2]and Ke Zhang[3]
[1]College of Automation Science and Engineering, South China University, Guanghzou, 510641 Email: zlyu@scut.edu.cn
[2]Beijing Transportation Information Center, Beijing
[3]Beijing Transportation Operation Coordination Center, Beijing

***Abstract*—People counting has many important applications in practice. The two key techniques in video based people counting system are people detection and people tracking. Most of the current people counting methods use body detection or motion detection to detect the people, which can produce some good results in sparse situations but fail in crowded scenes. In the crowded scenes, we know that the head is the most probable target that can be full visual. In this paper, we propose a people counting method in crowded scenes by detection the head information from the video taken from a camera installed straight down on the ceiling. A head classifier based on boosted cascade of Statistically Effective Multi-scale Block Local Binary Pattern (SEMB-LBP) features is proposed for people detection. The detected head is then tracked by a model matching method using harr feature. Combining the head detection and tracking together, a people counting strategy is presented to count the number of the people in the video frames. Experiment results show that the proposed method work well and robust in crowded scenes.**

***Index Terms*—People counting, human detection, tracking, crowded scenes.**

I. INTRODUCTION

People counting system has a wide application in people flow statistics of subway, airport's ticket selling, markets' traffic restrictions, security monitoring, etc. Traditional methods based on manual counting or infrared transceiver counting have a lot of drawbacks. For example, manual counting can not work around the clock or cope with the crowded scenes. The infrared transceiver counting system not only can not distinguish whether the target is a pedestrian or not, but also can not work correctly when the people walk side by side. Nowadays the people counting based on video becomes a hot research topic for its advantages like simple hardware, non-contact, non-invasive and wealth of information.

Video based people counting includes two aspect technologies: target detection and target tracking. Target detection is to find the interested target's position and size in the video frame, while target tracking is to predict the target's position along the video sequences with the information from the detection. Detection and tracking can work together to count the interested targets in the video. For people counting systems, since the people is interested target, the detection task is very challengeable because of the people's irregularity. The difficulties of the people detection lie in follow factors: (1) people's wearing has no uniform patterns; (2) people is not a rigid object and their poses change all the time; (3) the partial or full occlusions make the motion detection inapplicable and the shadow is easy to cause false detection.

The popular methods on people detection can be divided into three classes: (1) background modeling [1]–[3], which need to learn the background model over time, then subtract the background model from the video frame to get the foreground motion target. But in the crowded scenes, the occlusions will make the foreground's motion targets connected together, as a result the individual can not be distinguished; (2) body part detection [3], [4]. These methods divide the body into many parts: head, upper body, arms and legs. These body parts are detected separately, then reconstruction analysis is used to judge weather the detected target is a pedestrian or not. This methods can work in some special situations, but fail in the crowded situation for most body parts is invisible in most of time; (3) body shape detection [2], [5], [6]. They approximate the body parts into some special shapes. In [5], it approximates the upper body, arms and legs by rectangulars with different sizes, while the method in [2] uses the ellipses to approximate the upper body and the lower body. These shapes are used to match the probable body parts in the frame, the relations among these shapes are used to decide whether the target is a pedestrian or not. All the above methods can not work in the crowded scenes because of the occlusion.

For the target tracking, there are many choices, such as Kalman filter [7], Meanshift [8], Camshift [9], optical flow [10] and particle filter [11], and so on. Kalman filter is a theoretically optimal tracker under the assumption of linear model and Gaussian noises. For Kalman filter, the predicted state vector of the target need to be corrected after the measurement is obtained. However, in video based people counting system, the target's position measurement may be very difficult to be obtained in every frame. The Meanshift algorithm is a non-parametric method, it is an iterative kernel-based deterministic procedure which converges to a local maximum of the measurement function under certain assumptions about the kernel behaviors [12]. CamShift algorithm [9] is based on an adaptation of Meanshift that, with a given probability density image, it finds the mean (mode) of the distribution by iterating in the direction of maximum increase in probability density. Camshift fails to track multi-hued targets or targets where allow the target to be distinguished from the background

978-1-4799-4315-9/14/$31.00

and other targets. Particle filter is a theoretically optimal and effective tracker. It needs mass particles to approximate the posterior probability function of the tracked target. The drawback of particle filter is its huge computational cost.

In this paper, we propose a people counting system which can work in crowded scenes. We study the applications in which a camera can be installed straight down on the ceiling. The benefit of such configuration is that it can reduce the variation of target features. Meanwhile, a region of interest (ROI) is selected in video frame as the detection and tracking area, which can eliminate some uninterested areas' interference. For target detection, in [13], a face classifier with AdaBoost learning based on the Statistically Effective Multi-scale Block Local Binary Pattern (SEMB-LBP) feature vector is proposed. In [14], the same method is applied to train a face classifier based on harr feature. SEMB-LBP is a statistical analysis result of the LBP feature of image patches. It demonstrates more robust performance in practice. Moreover, experimental results show that the training and detection speed based on LBP feature is faster than that base on harr feature, hence, in this paper, we select the LBP feature to train a head classifier. Considering target tracking, the tracking process only proceeds in ROI, which can always be set as a small area, so a short time tracking process is required in practice and we don't need a complex tracker like particle filter. Here, we borrow the idea from Compressive Tracker (CT) [15] and propose a modified model matching tracker to meet the requirements of short time tracking and low computational cost. Finally, a counting strategy is proposed to count the people by combining the detection and tracking results. Experiment results show that the proposed people counting system is effective and robust under the practical test video sequences.

II. The Proposed People Counting System

Pedestrians are complex targets in real applications as shown in Fig.1a. It is a challenge task to detect pedestrians from video sequence as we discussed in Section I. A robust feature for detection become the key problem for such system. To get a robust feature, it is important to find that in which case a specific feature is robust. Our research finds that the head of a pedestrian in crowded scenes is the most probably visible part. Feature extracted from head is a good choice for people counting applications. Although the head is a rigid object, it demonstrates very different characteristics from a different viewpoints as illustrated in Fig.1b. It is difficult to find a robust feature pattern in such case. In order to overcome the above propblem, we adopt a configuration that the camera is installed straight down on the ceiling, so that the head have a stable shape feature shown in Fig.1c, which is a suitable robust feature for people detection. The hardware configuration is shown in Fig.2a and Fig.2b shows the field of view from the camera.

A. Overview of Proposed Method

Based on the hardware configuration described above, the algorithm flow of people counting is shown in Fig.3. The detail of processing steps are described as follows.

First of all, the system is initialized before processing and a

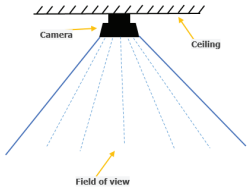
(a) Pedestrians in different poses

(b) Heads from different point of views

(c) Positive image samples

(d) Negative image samples

Fig. 1: Some kinds of pedestrian features



(a) Configuration of camera installation

(b) Field of view

Fig. 2: Configuration of hardware setting

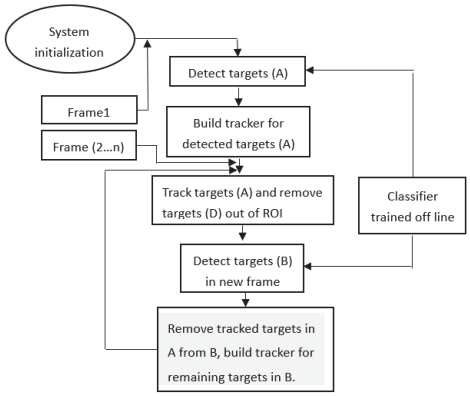


Fig. 3: Processing flow of the proposed method

classifier for target detection is trained off-line with provided samples of top view of human head as well as some samples of non-interested objects. The counter of people $count$ is set as $count = 0$ and the set of tracked target $A = \phi$. The ROI is set as the purple bounding box in Fig.2b. The selection of ROI is of great importance in practice. It should be emphasized that the selection of ROI needs to fullfill some constraints. For example, the ROI needs to cover the walkway, so that people go through the walkway will not be missed. It is better to select the width of ROI to be be 1.5-2.0 times larger than the size of pedestrians' head. This can reduce the detection area and the tracking distance, so that we can not only reduce the computational cost, but also avid tracking lost within a long distance.

After performing target detection in the first input frame, all the detected targets are inserted into the set A and the system sets the $count = n_A$ where n_A is the number of elements in A. For each element in A, a tracker is initialized. With the new input frame, the system first track the movement of each target in A, determine which target is out of ROI. Denoting all the targets out of ROI as a set D, the set A is updated as $A \leftarrow A - D$. For the new input frame, the target detection process is then applied and all the detected targets are denoted as set B. Obviously, some of the elements in B are already tracked by the system, so that a matching method must be applied to find which target is already tracked. The tracked target is removed from B, and the remaining elements in B are the new targets to be tracked. Then, all the elements in B are inserted into A and corresponding trackers are initialized for each new found targets. The counter is also updated as $count \leftarrow count + n_B$. We use a simple yet efficient matching method in this paper. It is described briefly as follows. If the distance between an element in A and an element in B is less than p pixels, then these two targets are considered to be the same one, and the position of the target in A is updated as the one in B.

For the algorithm flow presented above, it is clear that there are two important technical problems must be solved, the target detector, and the target tracker. The technical details are introduced as follows.

B. Target Detector

In this paper, a two class classifier is used to detect the probable targets in the frame. The classifier is trained with mass of positive and negative samples. The positive samples are obtained from interested targets and the negative samples are extracted from the possible interruptable background. The feature used in classifier is the SEMB-LBP [13]. It is a statistics of the MB-LBP feature [13] extracted from image patches.

In the application of SEMB-LBP in face recognition [13], the intra-personal and extra-personal variations are utilized in classification. The SEMB-LBP variation is defined as

$$D(H_s^1(i), H_s^2(i)) = |H_s^1(i) - H_s^2(i)| \qquad i = 0, \cdots, N \quad (1)$$

Fig. 4: Illustration of search window in target detection

where $H_s^j(i)$ denotes the ith element of SEMB-LBP feature vector H_s of target j [13], and s is the scale of MB-LBP. The intra-personal variation is the difference of two image's SEMB-LBP feature from the same person, while extra-personal variation is the difference of two image's SEMB-LBP feature from different person. The intra-personal and extra-personal variations are regarded as positive and negative samples respectively in the classifier training.

Different to the method used in [13], in this paper, we use the SEMB-LBP directly instead of its variations because in face recognition, the positive samples and negative samples are all human faces while in the proposed head detector, the positive samples are the top view of head in Fig.1c and negative samples are extracted from images in Fig.1d which are patches from background. The intra-personal and extra-personal variations have no big distinction. The SEMB-LBP feature vectors extracted from the positive images in Fig.1c and negative images in Fig.1d are directly used as positive feature and negative features in classifier training. The training method is same as [13]. We select the Gentle AdaBoost [16], which is a modified form of the Real AdaBoost procedure [17] and is a more conservative algorithm that has similar performance to Real AdaBoost.

During the detection process, a search window with different size is used as shown in Fig.4. The yellow box represents the ROI and the purple dashed box stands for the search window. Since the camera is fix and the distance between camera and head does not change significantly, the size of head in ROI only has slight change. We know that the head size of different people are different. Hence, during the detection, the size of the search window can be empirically set as 0.7 to 1.3 times of the average size of human head. The search window moves around all the possible place in the ROI and the image patch in the search window is the candidate sample. The classifier will judge whether the candidate sample is a head or not.

C. Target Tracker

For the target tracker, we use a modified version of the compressive tracking (CT) tracker [15]. The original CT tracker is able to do long-term tracking of a target. As discussed in Section I, in the studied application, the tracker only track target in a short-range distance. Hence, the CT tracker can be modified to be a more simple one. We propose to use a model

matching method to reduce the computational cost while meet the system requirements.

In this paper, we use ℓ_t to represent the targets location in frame t and $\ell(x)$ to denote the center location of a image patch x. For a image patch centered at location $\ell(x)$ and of size (w, h), a set of pixels are randomly selected from the patch with uniform probability. These selected pixels are also called feature points in the following context. There are m feature points and the generalized Harr-like feature ξ_i of these feature points are used to construct a feature vector $F = \{\xi_1, \xi_2, \cdots, \xi_m\}$, which represents the appearance model of the target in frame t.

Assuming the tracking target's position ℓ_t in frame t is known and we want search the target's position in frame $t+1$. We limit the search range of image patch as $x^s = \{x : ||\ell(x) - \ell_t^*|| \leq s\}$ in frame $t+1$, and $\ell(x)$ is the same size as the target size. It means that the image patch with a radius of s from the target's previous position is regarded candidate sample. The observation of target's location in frame $t+1$ is found by

$$\ell_{t+1} = \ell(\arg\min_{x \in x^s}(|F_t - F(x)|) \tag{2}$$

where F_t is the appearance model of target in frame t, and $F(x)$ stands for the feature vector of image patch x. After the target is found, its appearance model is updated using the most recent one. The tracking algorithm is summarized in Algorithm 1

Algorithm 1 tracking algorithm

Input: Given initial location ℓ_0 and appearance model $F_0 = \{\xi_1, \xi_2, \cdots, \xi_m\}$.

1: **for** $t = 1$ to K **do**

2: Search the location of patch using $\ell_t = \ell(\arg\min_{X \in X^s} w^T * (F_{t-1} - F(x))$ in the range $x^s = \{x : ||\ell(x) - \ell_t|| \leq s\}$;

3: Update the target's appearance model $F_t = F(x)$, where x is the image patch centered at tracked target location;

4: **end for**

III. Numerical Results

In order to train the classifier, we need mass of samples. A lot of photos taken in subway, airport, market, square and so on are cropped to get the positive images like Fig.1c and negative images like Fig.1d. There are 6383 positive samples and 6439 negative samples are used in classifier training. All the positive samples are scaled to be of size 10×10 pixels. The OpenCV provided utility, *opencv_traincascade.exe* [18], is used to train the classifier. The training parameters are given in TABLE.I.

Two real video sequences are used in performance evaluation. The length of video sequence 1 is 42 seconds and there are 45 pedestrains. Video sequence 2 contains 17 pedestrians. The detection result is shown in TABLE.II. It is clear that system has accuracy rate as high as 86% in average. The proposed system is implemented in C++ with OpenCV library. It can process about 16 frames/second for a frame of size $1280x720$ on $Intel(R)core\ 3.2GHz\ CPU$ with $8Gb\ RAM$. Some of the sample tracking results are illustrated in Fig.5.

TABLE II: Tracking Result

Vedio	Total Heads	Tracked	False Positives	Missed
one	45	38(84%)	2	9
second	17	15(88.2%)	2	4

IV. Conclusions

An effective and robust method for automatic people counting based on video processing and analysis is proposed in this paper. The proposed method utilize a head classifier based on boosted cascade of SEMB-LBP features is used to detect person in video sequence and a modified CT tracker to track the pedestrians. Experiment results show that proposed method has high accuracy in people counting on real video sequences. Moreover, the proposed method has a simple implementation structure and low computational load. All of these facts support that the proposed method is a suitable choice for practical applications.

Acknowledgements

This work was supported in part by the Beijing Science and Technology Program under Grant Z131106002813012, the National Natural Science Foundation of China under grants 61105121 and 61175114, the Natural Science Foundation of Guangdong under grants S2012020010945, the Fundamental Research Funds for the Central Universities, SCUT under grant 2013ZZ0040, the High Level Talent Project of Guangdong Province 2013KJCX0009.

References

[1] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *International journal of computer vision*, vol. 75, no. 2, pp. 247–266, 2007.

[2] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 30, no. 7, pp. 1198–1211, 2008.

[3] B. Wu, R. Nevatia, and Y. Li, "Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, 2008, pp. 1–8.

[4] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, 2008, pp. 1–8.

[5] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 29, no. 1, pp. 65–81, 2007.

[6] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 2, 2004, pp. 406–413.

[7] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using Matlab*. John Wiley & Sons, Inc., 2001.

[8] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 17, no. 8, pp. 790–799, 1995.

[9] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technol. J*, vol. 2.

[10] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *Proc. 7th Int. Joint Conf. Arti. Intell*, vol. 81, 1981, pp. 121–130.

TABLE I: Training parameters of classifier

Parameters	values	Description
-numPos	5000	Number of positive samples
-numNeg	3000	Number of negative samples
-numStages	20	Number of cascade stages to be trained
-featureType	LBP	Type of features
-w	10	Sample width
-h	10	Sample height
-bt	GAB	Gentle AdaBoost
-minHitRate	0.995	Minimal desired hit rate for each stage of the classifier
-maxFalseAlarmRate	0.5	Maximal desired false alarm rate for each stage of the classifier

Fig. 5: Sample tracking results

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

[12] R. T. Collins, "Mean-shift blob tracking through scale space," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 2, 2003, pp. II–234.

[13] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Proc. Int. Conf. Biometrics*, 2007, vol. 4642, pp. 828–837.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. IEEE, 2001, pp. I–511.

[15] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *European Conf. on Computer Vision*, 2012, vol. 7574, pp. 864–877.

[16] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

[17] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[18] "http://docs.opencv.org/."