Signal Processing 124 (2016) 27–35

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Real-time people counting for indoor scenes

Jun Luo [a,b,*], Jinqiao Wang [b], Huazhong Xu [a], Hanqing Lu [b]

[a] *School of Automation, Wuhan University of Technology, Wuhan, 430070, China*
[b] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China*

ARTICLE INFO

Article history:
Received 20 June 2015
Received in revised form
10 October 2015
Accepted 30 October 2015
Available online 2 December 2015

Keywords:
People counting
Head-shoulder matching
K-means clustering

ABSTRACT

People counting in indoor environment is a challenging task due to the coexistence of moving crowds with stationary crowds, recurrent occlusions and complex background information. The performance of existing crowd counting methods drops significantly for indoor scene since the stationary people are missed due to moving foreground segmentation and the counting results are often disturbed by occlusions. To address the above problems, in this paper we propose a counting approach for indoor scenes, which can count not only moving crowds but also stationary crowds efficiently. Firstly, a foreground extraction assisted by detection is introduced for crowd segmentation and noise removal with a feedback update scheme. Then we build a multi-view head-shoulder model for people matching in the foreground and estimate the number of people with an improved K-mean clustering approach. Finally, to reduce the disturbance of occlusions, we present a temporal filter with frame-difference to further refine the counting results. To evaluate the performance of the proposed approach, a new indoor counting dataset including about 570,000 frames was collected from four different scenarios. Experiments and comparisons show the superiority of the proposed approach.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Counting people from videos draws a lot of attentions because of its wide range of applications, such as building security, room resources adjustment, market research, intelligent building, etc., as shown in Fig. 1. Most existing approaches pay more attention to outdoor scenes or part of indoor scenes like passageway, and the motion information in these scenes can be utilized to reduce error. In recent years, the state-of-the-art methods based on supervised learning or semi-supervised learning can be classified into three categories: counting by detection [1–5], counting by clustering [6,7] and counting by regression [8–10], they have shown promising results for people counting in outdoor scenes. However, due to the coexistence of moving crowds with stationary crowds, recurrent occlusions and complex background information, the performance of existing crowd counting methods drops significantly for indoor scene since the stationary people are missed due to moving foreground segmentation and the counting results are often disturbed by occlusions.

Different from outdoor scene, indoor scene has its own characteristics, which make the counting task more challenging: (1) There exists stationary or slightly moving people in most indoor scenes and they will be classified as background by traditional foreground segmentation. Foreground segmentation is an indispensable step for most existing crowd counting approaches [3–5] to reduce background noise, removing foreground segmentation will lead to more computation burden and increase the risk of false alarms for complex background. (2) Frequent occlusion is another key

[*] Corresponding author.
E-mail addresses: junjing2218@gmail.com (J. Luo), jqwang@nlpr.ia.ac.cn (J. Wang), wutxhz@163.com (H. Xu), luhq@nlpr.ia.ac.cn (H. Lu).

http://dx.doi.org/10.1016/j.sigpro.2015.10.036
0165-1684/© 2015 Elsevier B.V. All rights reserved.

obstacle that holds back accurate crowd counting, which happens more often in indoor scenes than outdoor ones. For example, when people get together, we will extract some large blobs with several people inside, which lack object level information and are hard to be segmented. Some state-of-the-art methods [8–10] adopt regression-based techniques to learn a mapping between low-level features and the number of people, so as to circumvent explicit object segmentation and detection in these blobs. But these techniques generally involve a time-consuming frame-wise labelling process (even head-position annotations [3]) to train a regression model. What's more, the trained regression model is not easily adapted to a new scene. (3) The number of people often remains stable even with internal movement in indoor space when no one enters or exits. This kind of dynamic stability provides an important cue for accurate counting.

As analysis above, we propose a head-shoulder detection based crowd counting framework for indoor scenes. Firstly, for accurate foreground segmentation and background noise removal, we propose an update by detection method to conduct human-blob segmentation. Secondly, we introduce a multi-view head-shoulder model, which is generic, with no need for re-training, and useful to reduce the impact of occlusions. Thirdly, to reduce the disturbance of occlusions, we present a temporal filter with frame-difference to further refine the counting results by the state of dynamic stability in indoor scenes.

2. Related work

Various approaches for crowd counting have been proposed, which broadly fall into three categories [11]: counting by detection, counting by clustering and counting by regression. For counting by detection, some whole pedestrians detection based methods [1,2] are not effective because features of whole pedestrian are not obvious in densely crowded scenes. This problem has been addressed by some approaches based on part-based detectors [12], especially head-shoulder detectors [3–5]. Moreover, the counting accuracy can be further improved by post-processing methods, such as [4,13,14,12]. Zhao and Nevatia [4] treated problem of segmenting individual humans in crowds as a model-based Bayesian segmentation problem and presented an efficient Markov Chain Monte Carlo (MCMC) method to get the solution. Wang et al. [13] built a spatio-temporal group context model to model the spatio-temporal relationships between groups, formulating the problem of pedestrian counting as a joint posteriori maximum one. Zhang and Chen [14] used group tracking to compensate weakness of multiple human segmentation, which can handle complete occlusion.

Clustering based crowd counting consists of identifying and tracking visual features over time. Feature trajectories that exhibit coherent motion are clustered, and the number of cluster centers is regarded as an estimate of the number of moving objects. For example, Rabaud et al. [6] relied on KLT tracker and agglomerative clustering, Brostow et al. [7] used an unsupervised bayesian to decide the number of moving objects.

The third category estimates the crowd density or crowd count with a regression function [8,9] and various features of the foreground pixels, including total area [15–17], edge count [16,18,17], or texture [19]. The regression function is also various. For example, Chan and Vasconcelos [10] segmented the scene into different regions with different motions, and extracted various features from each segment. Then a Gaussian process regression was used for estimating the pedestrian count for each segment. Kong et al. [20] applied neural networks to the histograms of foreground segments and edge orientations.

Some detection or clustering based methods cannot work well in most indoor environment because they rely on the movement of crowd, but most indoor scenes often have some stationary crowd with few movements. Khemlani et al. [21]

Classroom Station Bank Mall

Fig. 1. Examples of indoor scenes.

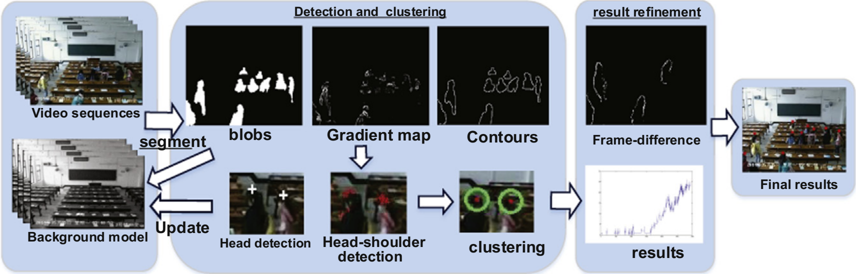


Fig. 2. Overall framework of people counting in indoor scenes.

counted people by exploiting the spatio-temporal coherence among small body part movements. Another solution [22] is to utilize semi-supervised learning and transfer learning to reduce the amount of manual annotation and make the model more applicable. However, the accuracy rate of people counting decreased with annotating only a handful of frames, and counting results were inevitably influenced by background noise. A preliminary version of our work was presented in [23]. This paper is an extension of [23], where we add a feedback scheme in the counting framework and more experimental comparisons.

3. Indoor people counting

In this section, we introduce the proposed indoor people counting framework. As illustrated in Fig. 2, our approach mainly includes three modules: blob segmentation, head-shoulder detection and temporal refinement. Given an input video sequence, we first segment the video into several blobs of interest. The corresponding gradient map in original frame and contours of blobs are extracted simultaneously. Next, a trained multi-view head-shoulder model is used for head-shoulder detection. Head detection and Head-shoulder detections can help find the candidates of people, vice versa, detection results can help update the background model. Finally, the characteristic of dynamic stability in indoor scenes is applied to estimate occlusions and refine counting results.

3.1. Background modeling and blob segmentation

Traditional background subtraction approaches aim to divide video frames into stationary pixels and moving pixels, which are generally applied by pedestrian counting methods to segment crowds. These methods build and update the background model with stationary pixels as a reference of moving pixels. Foregrounds are segmented based on difference between background model and current frames. Inspired by traditional approaches, in blob segmentation stage, we treat the frame as a combination of pixels on and off human bodies. If the background model is updated from pixels off human's bodies, pixels on bodies (including moving bodies and stationary ones) will always be segmented as foreground. Hence, our idea is to build a new updating procedure for people-related foreground segmentation. In this way, we can remove the background noise, meanwhile keep the crowd complete.

3.1.1. Background modeling

A initial version of blob segmentation is introduced in [23]. Due to the imprecise updating way, the blob segmentation in [23] is unstable and easy to be influenced by dynamic change in background. Inspired by Hofmann et al. [24], we propose an improved blob segmentation method by adding a feedback scheme to update the background model. The background model, named B, is formed by a series of pixel models, each of which contains a set of N recent background samples and a local updating rate T_x:

$$B(x) = \{b_1(x), b_2(x), \ldots, b_N(x), T(x)\} \tag{1}$$

3.1.2. Blob segmentation

These samples $b_n(x)$ are matched with their observation on the current input frame, depicted as $I_t(x)$ at time t respectively, to classify corresponding pixels as foreground 1 or background 0.

$$S_t(x) = \begin{cases} 1 & \text{if } \{\text{dist}(I_t(x), b_n(x)) < R, \ \forall n\} < \min \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where S_t is the segmentation result, $dist(I_t(x), b_n(x))$ measures the distance between a given background sample and corresponding current observation. R is the distance threshold and min is the minimum number of matches required for a background classification.

3.1.3. Feedback loop

In traditional background modeling, all the pixels have the same updating rate. The blob segmentation is unstable and easy to be influenced by dynamic change of the scenes such as light. Therefore we introduce a pixel-level feedback scheme to adaptive update the background model for accurately extracting the human blobs. The background model B is updated based on the "time subsampling factor" $T(x)$ [25]. A randomly selected sample in B has a $1/T(x)$ probability to be replaced by current observation $I_t(x)$. At the same time, one sample of the neighbors of $B(x)$ is also replaced by $I_t(x)$ with probability $1/T(x)$. Instead of imprecise updating way in [23], we build a feedback loop based on updating rate map T in this update process:

$$T(x) = \begin{cases} T(x) + \dfrac{1}{\overline{d}_{\min}(x)} & \text{if } S'(x) = 1 \\ T(x) - \dfrac{1}{\overline{d}_{\min}(x)} & \text{if } S'(x) = 0 \end{cases} \tag{3}$$

where $\overline{d}_{\min}$ is the average value of minimal decision distances. It is a measure of the background dynamics:

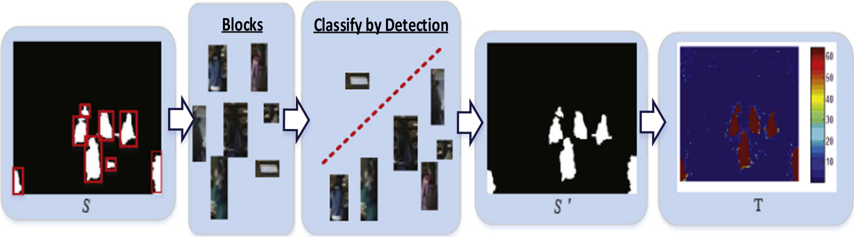


Fig. 3. Foreground refinement from $S(x)$ to $S'(x)$. Some blobs without people inside are removed through head-shoulder detection, the "time subsampling factor" map $T(x)$ is calculated according to $S'(x)$.

$\overline{d}_{\min}(x) = 1/N\sum_k D_k(x)$. $D_k(x)$ is the historical array of minimal decision distances:

$$D_k(x) \leftarrow d_{min}(x), \quad d_{min}(x) = \min_k \operatorname{dist}(I_t(x), b_n(x)) \tag{4}$$

As shown in Fig. 3, $S'(x)$ is refined from $S(x)$. Given a segmentation result $S(x)$, blobs are divided into several blocks based on connected component analysis. Head-shoulder detection classifies the image patches corresponding to these blocks in original frame as blocks with or without people inside. Blocks without people inside will be removed from $S'(x)$. Using the feedback loop (3), blobs without people inside will get a smaller $T(x)$, meaning that these pixels will be more likely to be updated into $B(x)$. Similarly, it is hard to update blobs with people inside into the background model. Because we always update the no-human pixel into the background model, both moving and stationary people can be extracted through this process.

3.2. Multi-view head-shoulder detection

Inspired by the multi-view learning for object [26–29], we present to build a multi-view head-shoulder model for head-shoulder parts are rarely occluded in a crowd. The shape of blobs can provide us with priori knowledge about the number of people in these blobs, which can benefit crowd counting greatly. As illustrated in Fig. 4(b) and (c), we detect head candidates and head-shoulder candidates from the boundary of foreground and gradient maps respectively. Counting results are estimated by the distribution of these candidates.

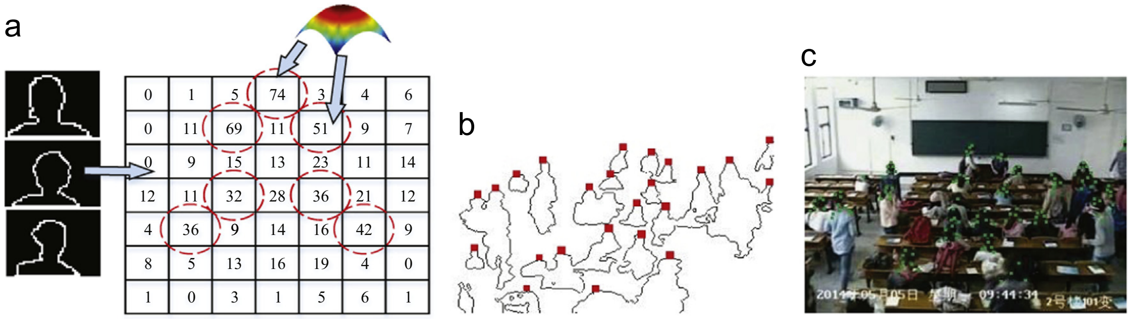
Head candidates are detected based on two conditions: it is the local vertical peak of the boundary, and there are enough foreground pixels under it [4]. Head-shoulder candidates are extracted using a trained multi-view head-shoulder model. As shown in Fig. 4(a), the training process of this model is a collection of the probabilities of the key points' appearances in the sampling images. Different from the training process in [30], our method can avoid the error brought by key points selection. Firstly, we sample a lot of head-shoulder image patches from surveillance video dataset. These training samples are classified as two representative categories: back view and side view. After manual refinement for the extracted edges, we normalize all the samples to a fixed size and project them into a grid-model which is partitioned into several smaller blocks. Note that the top point in this model is selected as the reference point and all the head points in samples are aligned to this reference point in the projecting process. Finally, we compared the number of edge points fell into each block, selecting some blocks with more edge points as positions of key points. A Parzen Windows method is used to assign each pixel a weight in each block. Assuming the window function obeys a Gaussian distribution, the weight is computed as follows:

$$W(x) = \frac{n}{N}\varphi\left(\frac{x - x_c}{h_N}\right) \tag{5}$$

where n represents the number of edge points in each selected block, N is the number of edge points in key blocks, x is the coordinate of arbitrary point in a block, x_c is coordinate of the center pixel in the block, h_N is the variance of the Gaussian distribution, φ is the window function, i.e. the Gaussian distribution in our case. Except for fuzzy probabilities of the key points' appearances, the model also stores the average unit normals v_i from samples. Unit normals are calculated according to coordinate value of neighbor edge points [4]. So in our model, the appearance probability expresses relative locations of key points, and corresponding unit normal expresses the shape of the model. What's more, the block-wise way makes this model more flexible to slight scale change and shape change. We extract head-shoulder candidates by matching the trained model with the gradient map of foreground area based on sliding windows. The matching score is defined as

$$S(x, y) = \sum_{i=1}^{k} W_i * v_i * O_i \tag{6}$$

where O_i stands for gradient vector in gradient map, k is the number of key points. When the matching score is larger than a threshold value, these matched key points are selected as head-shoulder candidates. After head-shoulder detection, we choose K-means clustering with max and min distance to estimate the number of people. The clustering centers are filtered by checking if there are enough head-shoulder candidates inside. The number of remained clustering centers is the final output of people count. Uncertainty of initial center points limits the convergence speed and effect of K-means. therefore, we select the head candidates as the initial center point to accelerate the clustering process.



0	1	5	74	3	4	6
0	11	69	11	51	9	7
0	9	15	13	23	11	14
12	11	32	28	36	21	12
4	36	9	14	16	42	9
8	5	13	16	19	4	0
1	0	3	1	5	6	1

Fig. 4. (a) A toy example of Head-shoulder model. (b) Head candidates (marked in red points). (c) Head-shoulder candidates (marked in green points).(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Given a head candidate sequence $A=\{\alpha_j\}_{j=1}^{M}$ and a head-shoulder candidate sequence $B=\{\beta_j\}_{j=1}^{N}$, the task is to find clustering center sequence $C=\{\gamma_j\}_{j=1}^{K}$ with the initial clustering center A. In order to reduce the impact of occlusions, we defined a maximum cluster radius R_{max} and a minimum cluster radius R_{min}. The detailed clustering process is given in Algorithm 1. For these points in B without head candidates, a standard K-means is used to clustering.

Algorithm 1. Dynamic programming for K-means clustering.

```
Input: $A=\{\alpha_j\}_{j=1}^{M}$, $B=\{\beta_j\}_{j=1}^{N}$.
Output: cluster center sequence $C$.
for $i=1$; $i \le N$; $i++$ do
    for $j=1$; $j \le M$; $j++$ do
        if $distance(\alpha_j, \beta_j) < R_{max}$ then
            delete $\beta_j$;
            $R_{max} = distance(\alpha_j, \beta_j)$;
        else
            $R_{max} = R_{max}$;
        end if
        if $distance(\alpha_j, \beta_j) > R_{min}$ then
            $C \leftarrow \alpha_j$;
        else
            delete $\beta_j$;
        end if
    end for
end for
return $C$;
```

3.3. Temporal refinement

Except for entrances or exits, most indoor scenes exist dynamic stability state during most periods. Although we cannot get an accurate count due to occlusion at some points, by making use of dynamic stability in indoor scenes, we can infer the accurate count gradually through a couple of frames. When the number of people remains stable, occlusion is the main cause for fluctuations on the counting number. We regard this as two representative states: one is that the counting number increases as the occlusion disappears. The other is that the counting number decreases when the occlusion happens. In any case, there are some moving pixels, which can be used to estimate whether the change of counting number is caused by occlusions. We mainly consider two kinds of occlusions: moving people occluded by stationary people and stationary people occluded by moving people.

Take Fig. 5 as an example, there are two stationary people and four moving people in this frame. An occlusion is occurring between one stationary people and one moving people. We use the frame-difference method to get moving pixels and or operations between two contiguous frame-difference results to enhance them. According to the connective relations, moving pixels are divided into different sequences $\{(x_i, y_i)\}_{i=1}^{n}$. Through matching clustering centers $C=\{\gamma_j\}_{j=1}^{K}$ between neighbouring frames by the nearest neighbor method, when the counting number decreases, we can know historical location of the disappeared clustering center γ_j. For the case of stationary people occluded by moving people, the historical position of disappeared clustering center will be surrounded by one moving pixels sequence. The judgement rule is described as follows:

$$O(\gamma_j)=\begin{cases} 1 & \text{if } \begin{cases} \min(x_i) \le x_{\gamma_j} \le \max(x_i) \\ \min(y_i) \le y_{\gamma_j} \le \max(y_i) \end{cases} \\ 0 & \text{else} \end{cases} \tag{7}$$

where x_i and y_i are the coordinate of moving pixels in one sequence. x_{γ_j} and y_{γ_j} is the historical coordinate of the disappeared clustering center. If $O(\gamma_j)$ equals to 1, we consider that γ_j is occluded. The counting result will ignore the change brought by this disappeared clustering center. For the case of moving people occluded by stationary people, occlusions are easier to be found because the disappeared clustering center was moving before. Furthermore, if the number of moving pixels is less than a threshold in one frame, we think the state in this indoor space is steady and keep the counting result unchanged.

4. Experimental results

4.1. Experiment setup

Since there is no available standard database of people counting for indoor scenes, we collect a dataset with HD cameras mounted at different indoor scenes. This dataset contains four types of indoor scenes, over 570,000 frames (about 380 minutes), including classroom, meeting room, office and mess hall. These are typical representatives for indoor scenes, which reflect the characteristic events for indoor scenes like entering, existing, gathering, discussion,

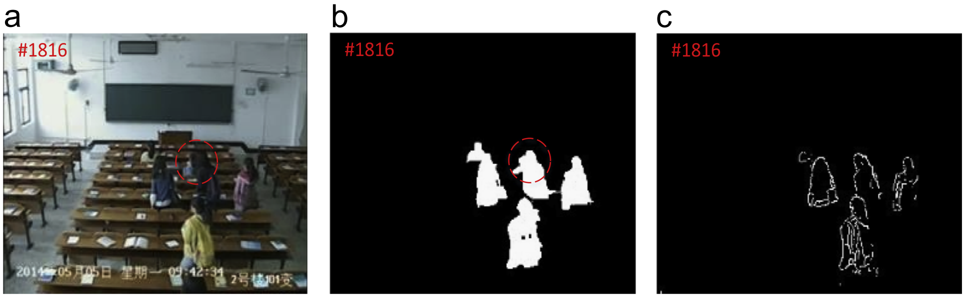


Fig. 5. Temporal refinement for people counting: (a) occlusion is occurring; (b) some information of head-shoulder disappeared and (c) moving pixels.

etc. The following criterion is used to evaluated the performance over all the indoor dataset:

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(num_g(i) - num_t(i)\right)^2} \tag{8}$$

where n is the total number of frames in the test video. $num_g(i)$ is the ground truth of the *ith* frame, and $num_t(i)$ is the test result of the *ith* frame.

4.2. Experimental results with temporal refinement

We select two groups of video segments from two indoor scenes to verify the effectiveness of temporal refinement, which are named as video1 (classroom) and video2 (mess hall) respectively. Fig. 6 shows the comparison results on video1 of our approach with and without temporal refinement (TR). It is obvious that the counting result without TR fluctuates more remarkably, especially

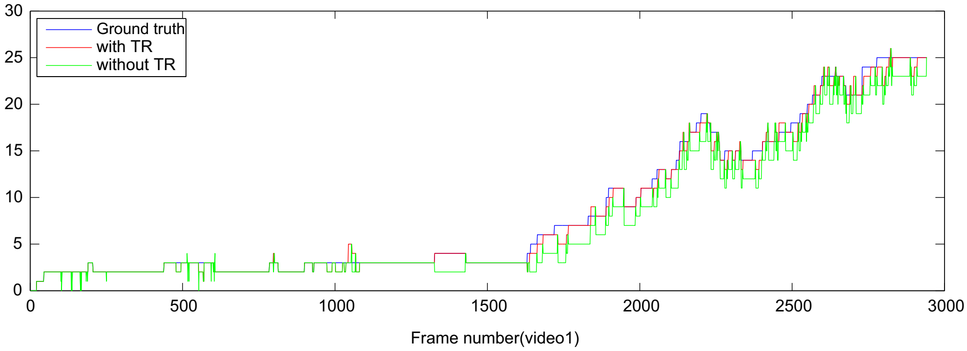


Fig. 6. Comparison results with and without temporal refinement (TR).

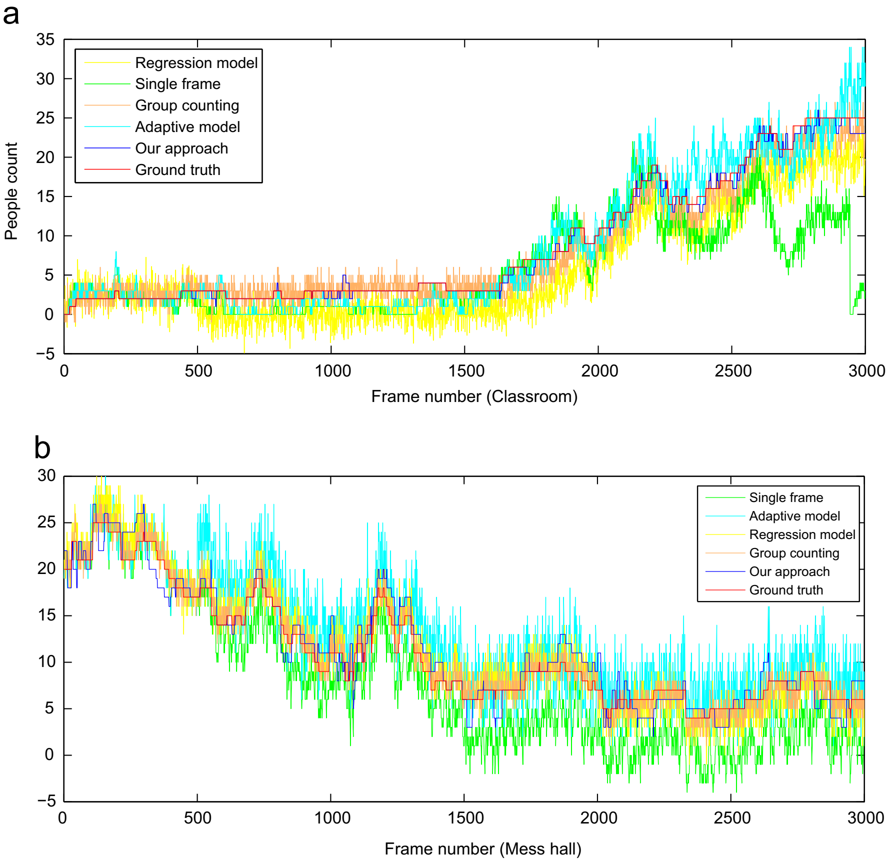


Fig. 7. Counting results on *video1* and *video2*.

when the number of people in the room increases. We can see that in Fig. 6, since the occlusion often causes the decrease on counting results, the counting results with temporal refinement are a little bigger than the results without temporal refinement. Furthermore, the counting results are more stable and smooth.

4.3. Comparison with state-of-the-art approaches

We compare our approach with "single frame" [4], "Adaptive model" [31], "Regression model" [31] and "Group counting" [13]. As shown in Fig. 7, "Regression model" [31] and "single frame" [4] shows poor performance. This can be

Table 1
The comparison results over all the indoor scenes dataset.

Video name	RMSE				
	Our approach	[4]	[31]	[10]	[13]
Classroom	1.0225	4.9142	2.1847	1.9864	1.5274
Office	1.3879	4.9618	2.7618	2.1215	2.2214
Meeting room	1.2613	4.9252	2.0137	2.1861	2.3116
Mess hall	1.4537	5.1372	3.4392	2.9101	2.8653

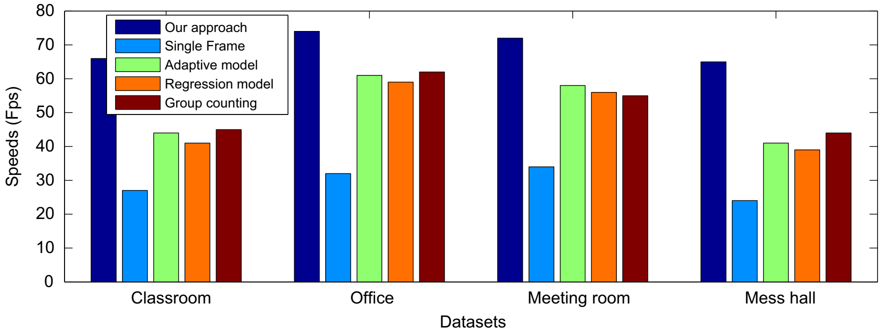


Fig. 8. Average speed comparison on whole indoor scenes dataset.

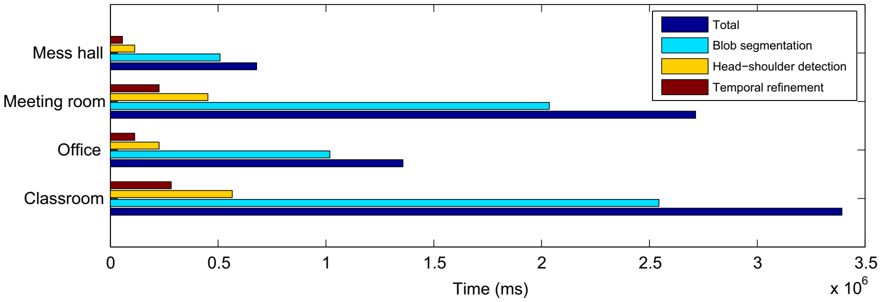


Fig. 9. Time cost distribution on whole indoor scenes dataset.

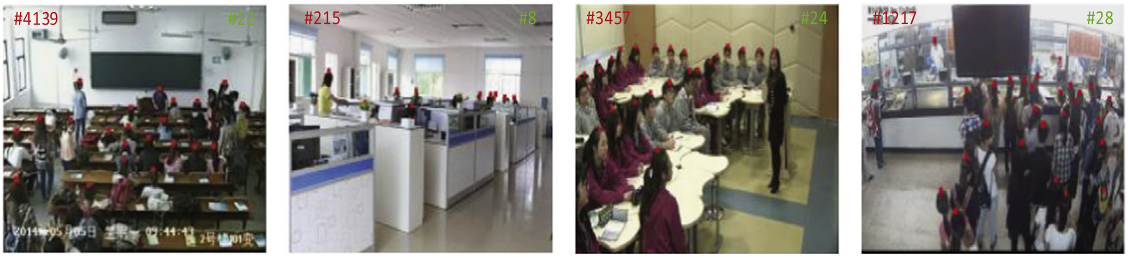


Fig. 10. Counting results on indoor scenes dataset, the number of people is marked in green on the top right corner of images, and the clustering result is marked on the head of human in red points.

explained by the limitation of traditional foreground segmentation in indoor environment that some blobs with stationary people inside cannot be extracted by foreground segmentation. So in most periods the counting results of “single frame” [4] are below the groundtruth, especially when most of the people are not moving. As shown in Fig. 7 (a), compared to our approach, “Adaptive model” [31] and “Group counting” [13] are more sensitive to noises caused by occlusions. The result of our approach is more smooth. Although sometimes there are some errors in our results, it can get closer to the groundtruth slowly after a number of frames. If the state in the classroom remains stable for a long time, the refinement on counting results will be constant. Fig. 7(b) shows the comparison results on *video2*. Fig. 10 shows several frames with counting results for different scenes. The RMSE comparison with “single frame” [4], “Adaptive model” [31], “Regression model” [31] and “Group counting” [13] is given in Table 1. which shows that our approach achieve the state-of-the-art performance on the indoor scenes.

4.4. Speed

Without any optimization for the c/c++ implementation, our approach runs an average of over 70 frames per second on classroom dataset (the resolution is 704 × 576) on a Intel i7 CPU at 3.4 GHz with no architecture-specific instruction. Because of changing density in scenes, the counting speed is unstable. For comparison, we compute the average speeds of our approach, [4,31,10] and [13] on whole classroom datasets (about 380 min), which is illustrated in Fig. 8. In addition, as shown in Fig. 9, the time cost of the whole counting process is mostly determined by the blob segmentation, since the head-shoulder detection and temporal refinement in our approach is efficient. The speed of whole process can be further accelerated by optimizing blob segmentation method.

5. Discussion and future work

In this paper, we proposed a fast and accurate counting people method for indoor scenes. Through applying a feedback update-by-detection scheme in foreground segmentation, our method approach a balance between background noise removing and stationary people segmentation. Comparative experiments show that traditional approach improved by our segmentation can reach a better performance in indoor scenes. The head-shoulder detection is the key problem in our approach, because it is linked closely with blob segmentation and clustering. A generic and flexible head-shoulder model is trained from a large number of samples. Head detection is treated as a prior to accurate the counting result and accelerate the counting process. To deal with occlusions, our method focuses on filtering the counting result in a number of frames based on the spatial temporal information between frames. Compared with [4,31,10] and [13], our approach achieves a better performance in the indoor environment.

In future work, two potential improvements will be considered. First, the head-shoulder model has a poor performance on detecting people with strange clothes and hats. Hence more robust features should be combined into head-shoulder model. Second, clustering in a circular region doesn't consider distribution characteristic of head-shoulder points, which leads to false positives or false negatives. The clustering process could be more robust by considering spatial distribution.

References

[1] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, In: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003 IEEE, 2003, pp. 734–741.

[2] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005. vol. 1, IEEE, 2005, pp. 878–885.

[3] M. Li, Z. Zhang, K. Huang, T. Tan, Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection, In: Proceedings of the 19th International Conference on Pattern Recognition, 2008, ICPR 2008, IEEE, 2008, pp. 1–4.

[4] T. Zhao, R. Nevatia, Bayesian human segmentation in crowded situations, In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003 vol. 2, IEEE, 2003, pp. II–459.

[5] T. Zhao, R. Nevatia, F. Lv, Segmentation and tracking of multiple humans in complex situations, In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, CVPR 2001, vol. 2, IEEE, 2001, pp. II–194.

[6] V. Rabaud, S. Belongie, Counting crowded moving objects, In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006 vol. 1, IEEE, 2006, pp. 705–711.

[7] G.J. Brostow, R. Cipolla, Unsupervised bayesian detection of independent motion in crowds, In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, vol. 1, IEEE, 2006, pp. 594–601.

[8] K. Chen, C.C. Loy, S. Gong, T. Xiang, Feature mining for localised crowd counting, In: BMVC, vol. 1, 2012, p. 3.

[9] V. Lempitsky, A. Zisserman, Learning to count objects in images, In: Advances in Neural Information Processing Systems, 2010, pp. 1324–1332.

[10] A.B. Chan, N. Vasconcelos, Counting people with low-level features and bayesian regression, Image Process IEEE Trans. Image Process. IEEE Trans 21 (4) (2012) 2160–2177.

[11] C.C. Loy, K. Chen, S. Gong, T. Xiang, Crowd counting and profiling: Methodology and evaluation, In: Modeling, Simulation and Visual Analysis of Crowds, Springer, 2013, pp. 347–382.

[12] W. Ge, R.T. Collins, Marked point processes for crowd counting, In: IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009. IEEE, 2009, pp. 2913–2920.

[13] J. Wang, W. Fu, J. Liu, H. Lu, Spatio-temporal group context for pedestrian counting.

[14] E. Zhang, F. Chen, A fast and robust people counting method in video surveillance, In: International Conference on Computational Intelligence and Security, 2007 IEEE, 2007, pp. 339–343.

[15] N. Paragios, V. Ramesh, A mrf-based approach for real-time subway monitoring, In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, vol. 1, IEEE, 2001, pp. I–1034.

[16] S.-Y. Cho, T.W. Chow, C.-T. Leung, A neural-based crowd estimation by hybrid global learning algorithm, IEEE Trans. Syst. Man Cybern Part B: Cybern. 29 (4) (1999) 535–541.

[17] A.C. Davies, J.H. Yin, S.A. Velastin, Crowd monitoring using image processing, Electron Commun Eng J 7 (1) (1995) 37–47.

[18] C.S. Regazzoni, A. Tesei, Distributed data fusion for real-time crowding estimation, Signal Process 53 (1) (1996) 47–63.

[19] A. Marana, L.d.F. Costa, R. Lotufo, S. Velastin, On the efficacy of texture analysis for crowd monitoring, In: Proceedings. SIBGRAPI'98. International Symposium on Computer Graphics, Image Processing, and Vision, 1998, IEEE, 1998, pp. 354–361.

[20] D. Kong, D. Gray, H. Tao, Counting pedestrians in crowds using viewpoint invariant training, In: BMVC, Citeseer, 2005.

[21] A. Khemlani, K. Duncan, S. Sarkar, People counter: counting of mostly static people in indoor conditions, In: Video Analytics for Business Intelligence, Springer, 2012, pp. 133–159.

[22] C.C. Loy, S. Gong, T. Xiang, From semi-supervised to transfer counting of crowds, In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2256–2263.

[23] J. Luo, J. Wang, H. Xu, H. Lu, A real-time people counting approach in indoor environment, In: MultiMedia Modeling, Springer, 2015, pp. 214–223.

[24] M. Hofmann, P. Tiefenbacher, G. Rigoll, Background segmentation with feedback: the pixel-based adaptive segmenter, In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, IEEE, 2012, pp. 38–43.

[25] O. Barnich, M. Van Droogenbroeck, Vibe: a universal background subtraction algorithm for video sequences, IEEE Trans. Image Process. 20 (6) (2011) 1709–1724.

[26] Y. Luo, D. Tao, B. Geng, C. Xu, S.J. Maybank, Manifold regularized multitask learning for semi-supervised multilabel image classification, IEEE Trans. Image Process. 22 (2) (2013) 523–536.

[27] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, Y. Wen, Multiview vector-valued manifold regularization for multilabel image classification, IEEE Trans. Neural Netw. Learn. Syst. 24 (5) (2013) 709–722.

[28] Y. Luo, T. Liu, D. Tao, C. Xu, Decomposition-based transfer distance metric learning for image classification, IEEE Trans. Image Process. 23 (9) (2014) 3789–3801.

[29] Y. Luo, T. Liu, D. Tao, C. Xu, Multi-view matrix completion for multilabel image classification.

[30] C. Sun, Q. Zou, W. Fu, J. Wang, Multiple hypotheses based spatial-temporal association for stable pedestrian counting, In: Advances in Multimedia Information Processing–PCM 2013, Springer, 2013, pp. 803–810.

[31] J. Liu, J. Wang, H. Lu, Adaptive model for robust pedestrian counting, In: Advances in Multimedia Modeling, Springer, 2011, pp. 481–491.