Optical Engineering 49(3), 037203 (March 2010)

Clustering method for counting passengers getting in a bus with single camera

Tao Yang
Yanning Zhang
Dapei Shao
Ying Li
Northwestern Polytechnical University
School of Computer Science
ShaanXi Key Laboratory of Speech and Image Information Processing
P.O. Box 886
Xi'an, 710129 China
yangtaonwpu@163.com

Abstract. Automatic counting of passengers is very important for both business and security applications. We present a single-camera-based vision system that is able to count passengers in a highly crowded situation at the entrance of a traffic bus. The unique characteristics of the proposed system include, First, a novel feature-point-tracking- and online clustering-based passenger counting framework, which performs much better than those of background-modeling-and foreground-blob-tracking-based methods. Second, a simple and highly accurate clustering algorithm is developed that projects the high-dimensional feature point trajectories into a 2-D feature space by their appearance and disappearance times and counts the number of people through online clustering. Finally, all test video sequences in the experiment are captured from a real traffic bus in Shanghai, China. The results show that the system can process two 320×240 video sequences at a frame rate of 25 fps simultaneously, and can count passengers reliably in various difficult scenarios with complex interaction and occlusion among people. The method achieves high accuracy rates up to 96.5%. [DOI: 10.1117/1.3374439]

Subject terms: people counting; feature tracking; online clustering.

Paper 090623R received Aug. 15, 2009; revised manuscript received Feb. 10, 2010; accepted for publication Feb. 23, 2010; published online Mar. 30, 2010.

1 Introduction

Developing a fully automatic, efficient, and robust people-counting system in surveillance scene is a subject of great scientific and commercial interest. For instance, traffic buses and railway stations can use this information to identify hourly traffic patterns, shopping malls and supermarkets can use it to analyze customer consumption habits, and security people can use it to monitor some key places and detect suspicious events.

Due to its vast number of applications, many people-counting approaches and systems have been developed. Earlier attempts include the contact-type counters and IR beams. However, those approaches count people only one at a time and are not suitable for the high-density traffic flow at the entrance of a bus (see Fig. 1).

Recently, vision-based people counting has become a very active research area in the computer vision community.[1–13] The main challenges come from complex interaction and occlusion among people. According to the number of sensors, the existing passenger counting approaches can be roughly classified into three categories: methods based on multiple camera, methods based on stereo camera, and single-camera-based approaches.

Some approaches[2,5] adopt an overhead stereo camera to take the advantages of object height information and segment people in a disparity map. Although the stereo-based method achieves good accuracy in a scarce scenario, its performance in largely dependent on the accuracy of disparity, and will be degraded under illumination changes. In addition, the cost of a stereo camera and the requirement of precise calibration eliminate its flexibility.

Multiple cameras are used in Refs. 9 and 10 and these approaches achieve satisfactory results in monitoring a large area. However, realistic traffic scenarios contain not only loose groups of people but rather crowds of individuals such as those shown in Fig. 1. To the best of our knowledge, no multiple-camera tracking algorithm can handle such severe occlusion effectively. Moreover, bus space is limited and a multiple-camera system cannot be installed conveniently.

Because of the advantages in terms of low cost, easy installation and low complexity, single-camera-based passenger counting techniques have received a lot of recent attention.[1–3,6–8,11,12] To cope with realistic high-density scenarios, two principally different strategies have been followed in the literature. The first approach[1,4] attempts to set up a relationship model between the density of the crowd and some region-based statistical image features, so as to avoid segmenting a large crowd into individuals. The performance of this approach is mainly dependent on robust feature extraction and training. The other approach aims at segmenting and tracking individuals under heavy occlusion. The most common methods of this class use background modeling and foreground blob tracking,[14,15] where counting is performed on interactions of foreground trajectory and user-defined tripwires or virtual gates. Typically, the moving foreground is obtained by a threshold. However, finding an appropriate threshold is difficult in a real dynamic scene, and it is also hard to integrated some task-specific information into this bottom-up process. Instead of

0091-3286/2010/$25.00 © 2010 SPIE

Fig. 1 Example of a dense crowd on a traffic bus. Our goal is to count the number of passengers in video sequences such as this.

background subtraction, recent efforts[3,6–8] employ clustering techniques, and processing results show that it is a powerful potential way to solve a passenger counting problem. The essential step of this approach is to design an effective and efficient feature extraction and online clustering algorithm, to balance the computer cost and accuracy of the overall system.

In this paper, we present a novel passenger-counting algorithm and a system for real-time traffic flow estimation in a traffic bus. The system detects the motion trajectory within the region-of-interest by a KLT (Kanade-Lucas-Tomasi) tracker rather than a background model, an integrates task-driven information for trajectory validation. Counting is performed by applying a novel appearance and disappearance time clustering of the feature point trajectory.

The rest of this paper is organized as follows. Section 2 introduces the framework of our system. Section 3 describes the passenger-counting system with KLT-based feature tracking, trajectory validation, and an online trajectory clustering method. Section 4 presents experimental results and discussions. The paper is concluded in Sec. 5.

2 System Overview

The framework of the proposed system is displayed in Fig. 2. The system mainly contains two modules: (1) a KLT-

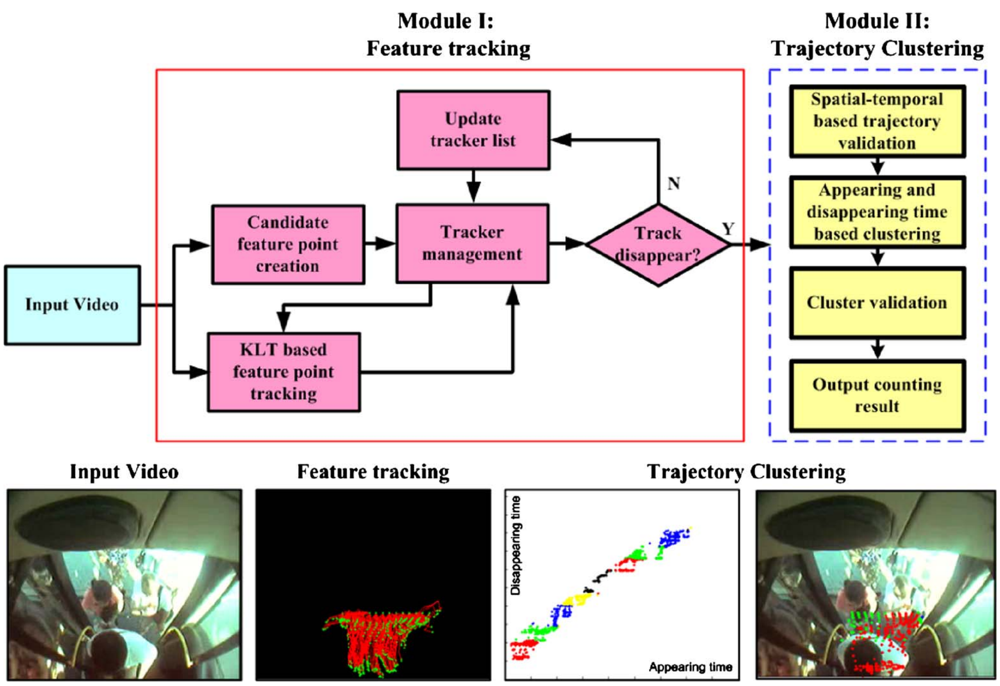

Fig. 2 Flowchart of our single-camera passenger-counting system.

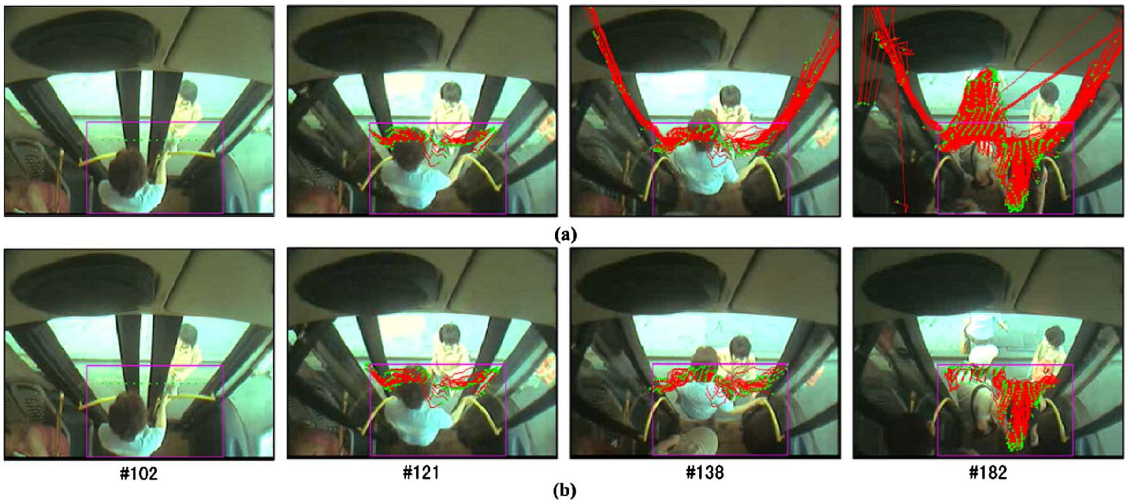

Fig. 3 (a) Initial trajectories and (b) trajectories after validation.

tracker-based feature tracking and management module developed to create the candidate feature point on a tripwire, compute object motion characteristics inside the defined region of interest, and maintain or remove feature point from active tracking list; and (2) a spatial-temporal clustering module designed to delete an invalid trajectory, project feature trajectory into a low-dimensional feature space, and automatically assign each trajectory to a certain object by online clustering. The individual components of the system are described in the following sections.

3 Feature Tracking and Online Clustering

3.1 *Feature Tracking and Candidate Creation*

Instead of implementing background subtraction and blob tracking, our approach is based on clustering a rich set of extended tracked features. In this paper, a pyramidal implementation of the classical KLT tracker[16–18] is used for its simplicity and robustness. The driving principle behind the KLT tracker is to determine the motion parameters of local windows W from an image I to image J. The optical flow is computed at the lowest level of the pyramid and then propagated to the higher levels, thus the lower level provides an initialization for tracking at the higher resolution. The number of pyramid levels is three and the patch size used is 16×16 pixels in our experiment. These parameters provide a good trade-off between the accuracy of the motion estimation and the robustness to large motion.

Because the computer cost linearly increases with the number of tracked feature points. To improve the efficiency of feature tracking, we manually select a virtual tripwire on the input image, and create candidate feature points on it only at every input frame. We verify candidate feature points by their displacement for consecutive m frames, and initialize new trackers only for moving feature points. Let $\tau^i(t)=[\tau_x^i(t),\tau_y^i(t)]^T$ refer to the track of feature i at time t. Then the motion state $S_m(i)$ of a candidate feature i can be computed as

$$S_m(i)=\begin{cases}1 & \text{if} \sum_{t=1}^{m}|\Delta\tau_x^i(t)|>T_d \text{ or } \sum_{t=1}^{m}|\Delta\tau_y^i(t)|>T_d \\ 0 & \text{otherwise},\end{cases} \quad (1)$$

where $\Delta\tau^i(t)=\tau^i(t+1)-\tau^i(t)$ is the displacement threshold. The main purpose of Eq. (1) is to compute the motion state of the candidate feature points, and feature points with large displacement in m consecutive frames will be initialized as new tracks for further processing. Thus, either computing the distance of x and y dimensions or Euclidean distances of points can be used in this case.

In some cases, random motion of background can be "technically" solved by constraining the area where features are sought in a small, user-defined, and carefully selected area. However, our system does not require the background to be completely static for following two reasons. First, in our system, we create candidate feature points only on a single virtual tripwire in the input image; due to the view angle of the camera, the virtual tripwire contains lots of randomly moving objects (e.g., door switching, cars, or other people passing by), and it is difficult to select a clean area in this case. Second, because we have a trajectory validation step, the random trajectories of feature points on the predefined tripwire are removed by analyzing their spatial and temporal characteristics, and it does not influence the final people-counting results.

By applying the preceding KLT tracker and candidate feature point selection in each frame, a number of feature points with their associated trajectories in the sequence are obtained for further processing.

3.2 *Trajectory Validation*

Usually the passengers require a certain period of time to get on the bus; during this time the passenger's interaction, door switching, and other complex conditions will influence the tracking accuracy. As a result, the set of feature point trajectories we are dealing with are unusable for directly clustering [see Fig. 3(a), red curves]. Thus the first

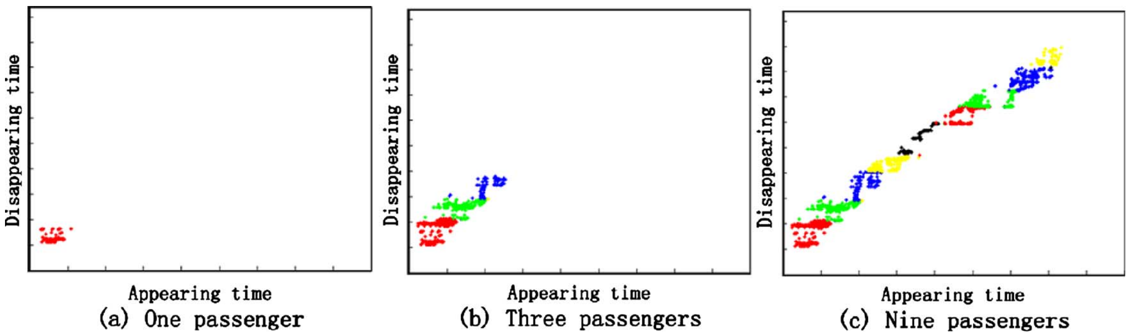

Fig. 4 Trajectory projection and clustering results in a low-dimensional feature space.

challenge before clustering is to find a way to validate the created feature point trajectories. This section demonstrates how spatial and temporal cues can be used to prune many invalid trajectories.

First, let us assume the region of interest (ROI) is known: a user-defined box at the entrance of bus (see Fig. 3, red rectangle). Since under normal circumstances, a passenger's location at the entrance of the traffic bus is within a certain range, the trajectories outside the ROI should be removed. By definition, if a trajectory τ^i satisfies one of the following conditions, it will be deleted from valid tracks.

$$\exists t, \tau_x^i(t) \notin \text{ROI} \quad \text{or} \quad \exists\, t, \tau_y^i(t) \notin \text{ROI}. \tag{2}$$

Similarly, since the passenger must pass through the entire ROI to enter the bus, the distance between the starting point $\tau^i(t_s)$ and the ending point $\tau^i(t_e)$ of a trajectory should be larger than the height of the ROI:

$$\|\tau^i(t_e) - \tau^i(t_s)\|_2 > L_{\text{ROI}}, \tag{3}$$

where t_s and t_e represent the appearance and disappearance times of τ^i, and $\|\cdot\|_2$ refers to the L_2 form.

Usually the passengers require a certain period of time to get on the bus, and we believe this information is helpful for trajectory validation. To get the statistical characteristics of the time duration, we first use the KLT tracker to track interest points on the body of the passengers in many surveillance videos, and then manually label many valid trajectories as samples. Finally, through analyzing the lifetime of the selected valid feature point trajectories, we find that the time interval between the appearance and disappearance times of passenger is within the scope of certain period, and its probability density basically obeys the Gaussian distribution. Thus in our system, the Gaussian distribution is selected to model the trajectory lifetime, and the maximum likelihood estimation of the mean μ_t and standard deviation σ_t are learned by computing the lifetime of trajectories passing through the ROI:

$$\mu_t = \frac{1}{N}\sum_{i=1}^{N} \tau_f^j \quad \sigma_t^2 = \frac{1}{N}\sum_{i=1}^{N} (\tau_f^j - \mu_t)(\tau_f^j - \mu_t)^T, \tag{4}$$

where τ_f^j is the lifetime of the trajectory. If one trajectory τ^j does not comply to the condition $\|\tau_f^j - \mu_t\| > 2.5\sigma_t$, then it does not belong to a valid object.

Figure 3 shows the trajectory validation results of our approach. In this case, many random trajectories are created in the process of opening the door [Fig. 3(a), #138], at the same time, some passengers are getting off [Fig. 3(a), #182]. By analyzing the spatial and temporal clues of the trajectories, our method correctly handles the mentioned problems, and acquires clean trajectories for further clustering.

3.3 *Temporal-Features-Based Trajectory Clustering*

The objective of trajectory clustering is to assign tracked feature points into groups, so that features from the same passenger are more similar to each other than features from different persons. After the process of trajectory validation, we develop a temporal-features-based online clustering algorithm to segment individuals from crowd.

Here, the similarity measure is the key to the design of an accurate clustering algorithm. An intuitive idea is the use of a trajectory's spatial distance; however, because of complex interactions among passengers in the crowded scenario, the position of trajectories are very close to each other and it is not conducive to the accuracy and robustness of clustering.

Differently, through statistical analysis of a large amount of trajectory data, we find that the appearance and disappearance times of valid feature points from the same person show significant clustering features (see Fig. 4). Thus, instead of computing the spatial distance between trajectories, we project the high-dimensional feature point trajectories into a 2-D feature space by their appearance and disappearance times, and similar to the training process of trajectory lifetime (see Sec. 3.2), a zero mean Gaussian distribution model is automatically learned for each dimension. The following is the clustering algorithm, where the $\sigma_{t_s}^k$ and $\sigma_{t_e}^k$ denotes the standard deviation of appearance and disappearance times, respectively. Figure 4 shows the trajectory projection and online clustering result for nine passengers.

1. For the first disappeared valid feature point trajectory, generate a cluster in the appearance- and disappearance-time-based low-dimensional feature space, and take the point (t_e^1, t_s^1) as the cluster center.
2. For each new disappeared valid trajectory τ^i at time t, compute its distance to every existing cluster center

in the feature space. If one of the following conditions satisfied, a new cluster will be generated.

$$|t_e^i - c_{t_s}^k| > 2.5\sigma_{t_s}^k \quad \text{or} \quad |t_s^i - c_{t_e}^k| > 2.5\sigma_{t_e}^k. \tag{5}$$

Otherwise, the disappeared trajectory will be assigned to the nearest cluster, and the new center of this cluster is updated as

$$c_{t_s}^k(t) = \frac{N_k - 1}{N_k} c_{t_s}^k(k-1) + \frac{1}{N_k} t_s^i$$
$$c_{t_e}^k(t) = \frac{N_k - 1}{N_k} c_{t_e}^k(t-1) + \frac{1}{N_k} t_e^i, \tag{6}$$

where N_k is the total feature number of cluster k.

3. Check the number of feature points of every existing cluster, and output the passenger counting result NC as

$$\mathrm{NC} = \sum_{k=1}^{K} h(c^k) \quad h(c^k) = \begin{cases} 1 & \text{if } N_k > T_N \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where K is the total number of existing clusters, and T_N is the minimal feature point number for a valid cluster. To choose a suitable T_N, we must estimate the person's average image size. After installing the camera on a certain bus, given the intrinsic and extrinsic parameters of the camera, we can estimate the average image size of a person. In our experiment, T_N is set to 60 according to the camera setup.

4 Experimental Results and Discussion

To evaluate the performance of the presented passenger-counting system, extensive practical tests were undertaken in realistic complex traffic scenarios with vastly different conditions. The setups and the resulting counting statistics are described in the next four subsections.

4.1 *System Setup and Performance Evaluation Dataset*

We implemented our algorithm in C++; the developed passenger counting system uses a single computer (3.0-GHz dual core) to process two cameras simultaneously with a resolution of 320×240 pixels, and achieves a frame rate of 25 fps. The system was set up at the entrance of several traffic buses in the city of Shanghai.

Images were recorded with one camera at a resolution of 320×240 pixels with a 25 Hz frame rate. We have captured 33 video sequences at different times during a workday, so as to set up a performance evaluation data set. The video contain both loose groups of people and crowds of individuals. In addition, some other challenges such as frequent illumination changes, moving background, door switching, and people getting on and off at the same time are included. The ground truth of the passengers was determined manually, by looking at the consecutive frames to determine how many passengers got on the bus.

4.2 *Typical Traffic Bus Scene Disturbance Handling*

Usually the door switching detection is a prerequisite for accurate passenger counting. This is because during the process of the bus moving, many environmental disturbances will influence the performance of the vision-based counting system.

Although we can acquire the door switch signal from the bus's access control system, it will increase the complexity of the overall system. To minimize the interactions with other systems, the idea for the design of a passenger-counting approach should have the ability to handle such problems.

In the proposed system, we created only the candidate feature point on the predefined tripwire, and use only the valid trajectories that pass through the entire ROI for further clustering. As a result random feature trajectories caused by the preceding disturbances are automatically removed, and it is not necessary to detect the door switching in our system.

Figure 5 gives example sequences of the clustering results under three typical scenarios: frequent illumination changes (Fig. 5, first row), dynamic moving background (Fig. 5, second row), and the bus door opening (Fig. 5, third row). The proposed system correctly handled these complex conditions, and no false alarm cluster was created (Fig. 5, first column).

4.3 *Passenger Counting Under Different Scenarios*

In this section, we give the passenger-counting results under two typical scenarios. First, the proposed system was tested for normal traffic flow, i.e., people getting on the bus without too many interactions around the entrance. Figure 6 shows the passenger-counting result under a scarce scenario. In this video, two persons get on the bus one after another, and each individual has a number of tacked feature points (Fig. 6, #57, #71, and #101, red and green dots). The first column shows the corresponding online feature-clustering results for each person. As we can see, through projecting the high-dimensional feature trajectories into the low-dimensional feature space, feature points for each individual in the new feature space show clear clustering characteristics. Thus, not only do we reduce the feature dimensions but we also improve the classification accuracy.

Figure 7 contains the passenger-counting results under a challenging crowd scenario. Nine passengers get on the bus in this test video. On one hand, due to the camera view angle, the occlusions become so severe that it is difficult for many tracking algorithms to accurately segment and track the entire body of each individual. On the other hand, because the background is invisible in the whole sequence (Fig. 7, #37, through #441), no background modeling methods can train an effective background and segment a foreground blob precisely in this case. In contrast, through feature tracking and online clustering, the proposed system successfully handled this challenging case, and correctly clustered the input feature trajectories into nine classes (Fig. 7, bottom left).

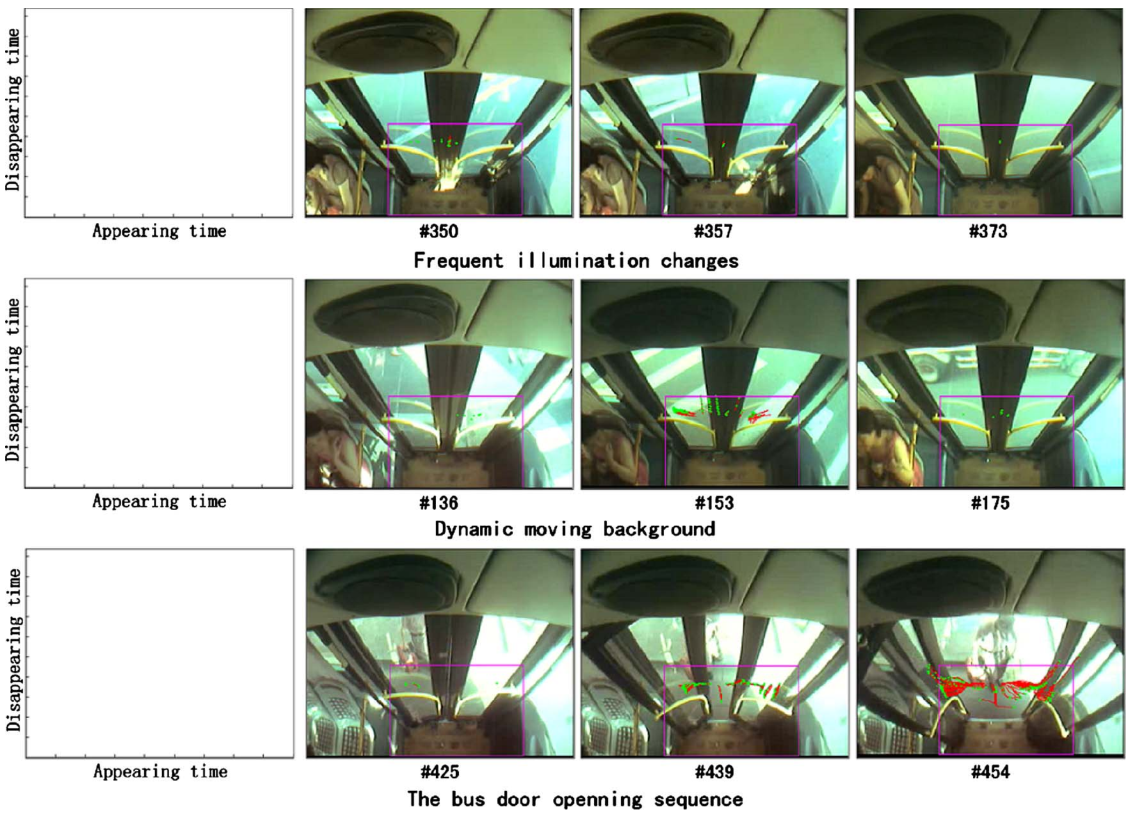

Fig. 5 Passenger-counting results under bus scene disturbances.

4.4 *Comparison with the Background-Modeling-based Approach*

Although vision-based people counters have recently been employed in many situations, there are still situations that consistently present challenges to them, and the entrance of a bus is one of them. To segment passengers accurately, some existing counters for buses, such as those produced by Traf-sys[19] and Acorel,[20] use expensive thermal or IR sensors to count people, and achieve 95% accuracy, as they have reported. Differently, our system uses only a very reasonable CCD camera for passenger counting on buses, and achieves similar performance with surveillance videos cap-

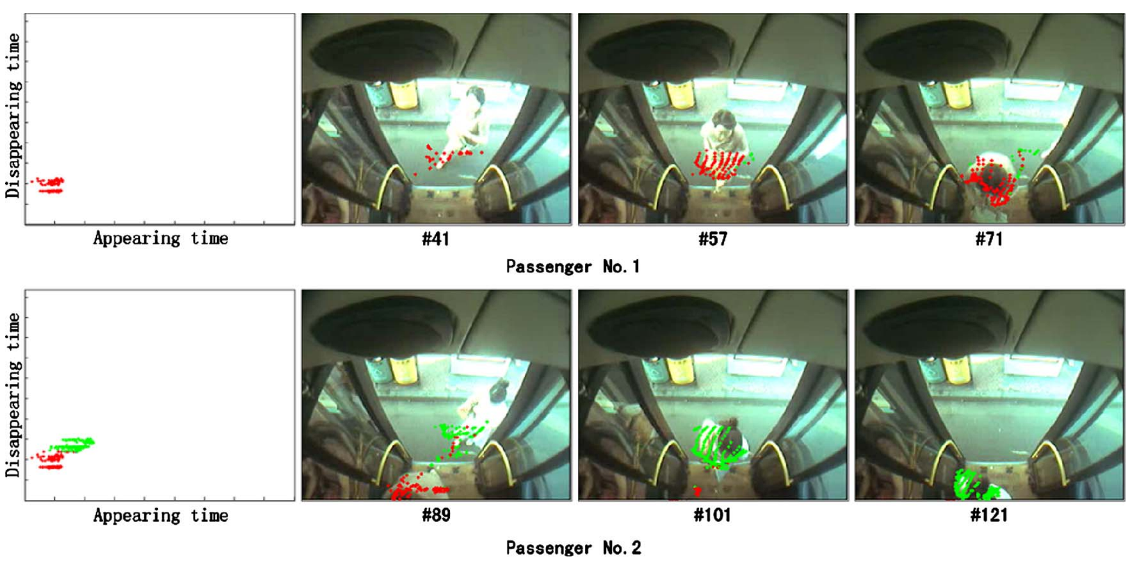

Fig. 6 Online clustering and passenger-counting results under a scarce scenario. Each row displays the clustering result in the first column, and superimposes the classified features on the corresponding passengers.

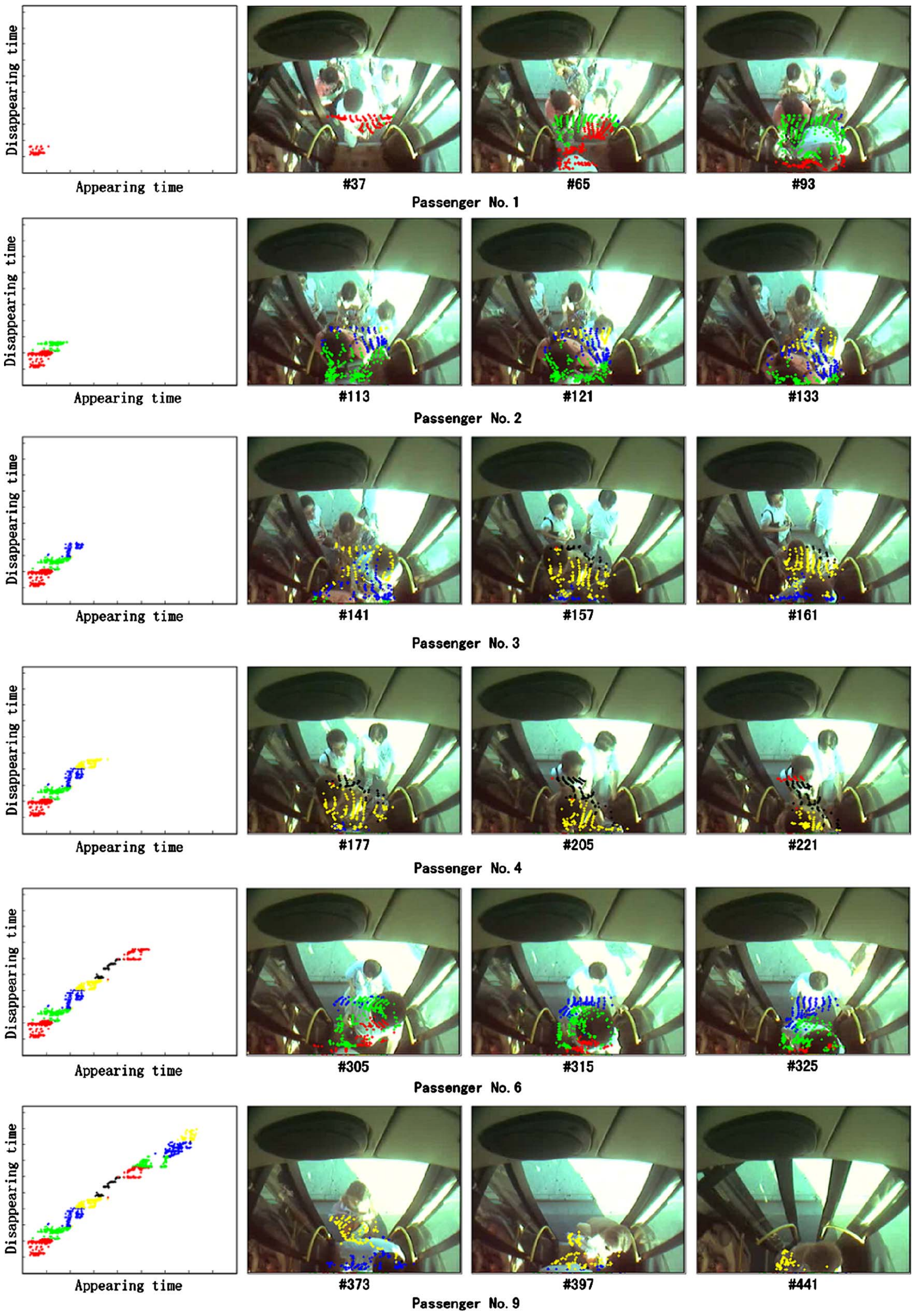

Fig. 7 Passenger-counting result in a crowded scenario. Each row displays the clustering result, and superimposes the classified features on the corresponding passengers.

tured from Shanghai City. Because the sensor is different, it is hard for us to compare our algorithm with those in detail. Thus, to demonstrate the advantages of our approach, we compare it with other state of the art people-counting methods using a single CCD camera.

The state of the art methods[21–23] for people counting passing through a gate based on image processing usually includes the following two steps: (1) background modeling and moving foreground blob segmentation and (2) bounding-box detection and tracking. Sometimes merge/split detection are also added to handle possible occlusion among passengers. Although those methods work well in some scenarios, our experimental results (see Fig. 8) demonstrate that they cannot handle a people-counting problem in a traffic bus such as ours due to the following reasons.

First, the state of the art approaches are based on an assumption that the camera is not moving and the background is static. However, in our system, the camera is mounted on a moving bus with a rapidly changing background. Moreover, even when the bus stops at the station, the camera is often shaking when the passengers are getting on the bus. As a result, the background subtraction results of the state of the art methods contains many false alarms (see Fig. 8) in the traffic bus scenario.

Second, the performances of the background-subtraction-based methods are sensitive to the number of people in the surveillance scene, and it is difficult for them to segment and track individuals correctly under heavy occlusion.

In contrast, our method changes the people-counting problem into a feature-point-trajectory-clustering problem. This method not only does not require background modeling with a static camera, but also can effectively deal with people counting under heavy occlusion in a crowded scenario. Figure 8 displays the comparison results of our approach and results of standard GMM (Gaussian mixture model) background-modeling approach. Additional demonstrations can be found in the following address http://www.saiip-vision.org/tyang/peoplecounting.htm.

4.5 *Performance Evaluation Results on the Whole Data Set*

Figure 9 shows the overall evaluation of counting results for 32 video sequences captured from the real traffic bus in Shanghai City on a workday. Figure 9 (top Figure) displays a comparison result for the number of passengers estimated by our approach (Fig. 9, top figure, red bins) and the manually determined ground truth (Fig. 9, top figure, blue bins). In most cases, the system achieves very high accuracy. However, in video No. 10, the manually determined count number is 16, while our approach counted only 12. We checked the video sequence, and found that the error was caused by several women carrying their children, and getting on bus together precisely at the same time. In this low-probability case, our system has classified them into one object.

The accumulated sum reveals that our approach is highly accurate (Fig. 9, bottom left), the relative error rate for the total thirty three video sequences is only 3.4% (Fig. 9, bottom right), and the final correct counting rate achieved by the system is 96.5%.

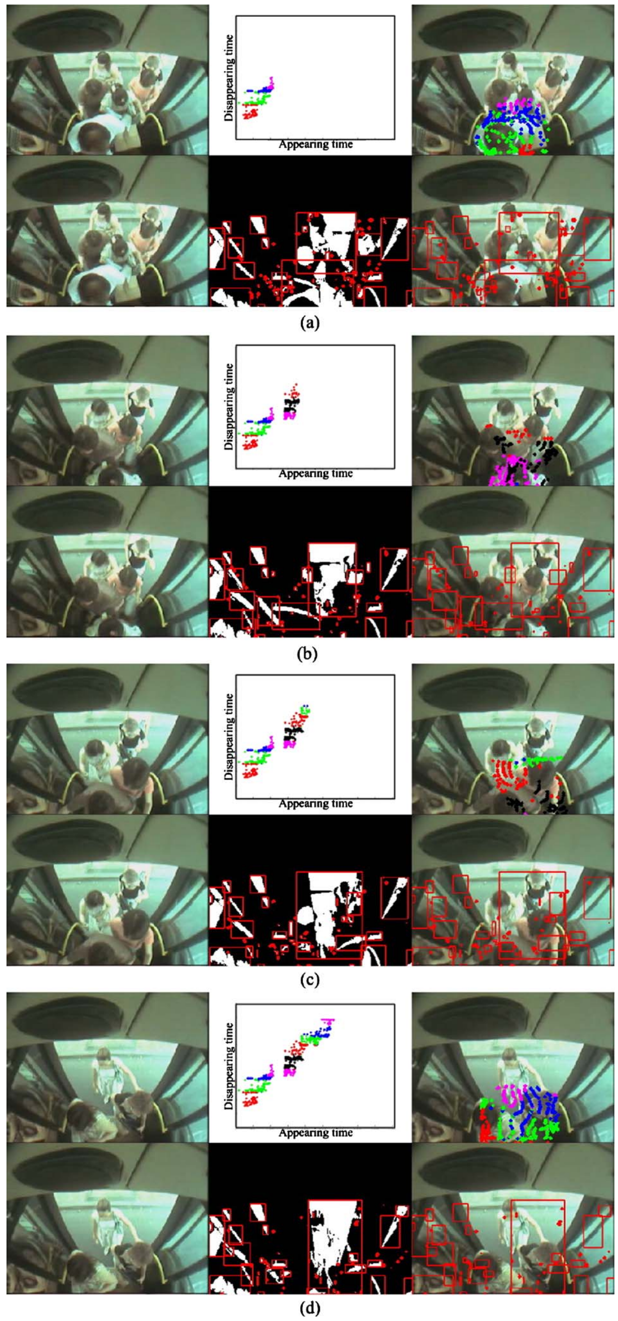

Fig. 8 Comparison of results with the background-modeling approach. The first row of each group shows the input image, the clustering result, and the feature point tracking result of our approach. The second row of each group displays the input image, the foreground segmentation, and the moving object detection result of the GMM model.

5 Summary and Conclusions

We presented a single-camera passenger-counting system. The proposed system uses a KLT feature tracker to estimate feature point trajectories in the ROI, and applies a novel online spatial-temporal clustering algorithm to segment people in crowded situations. The counting system runs at 25 fps on two 320×240 image sequences simultaneously without any code optimization on a single computer, and achieves accuracy rates up to 96.5% on 32 challenge video

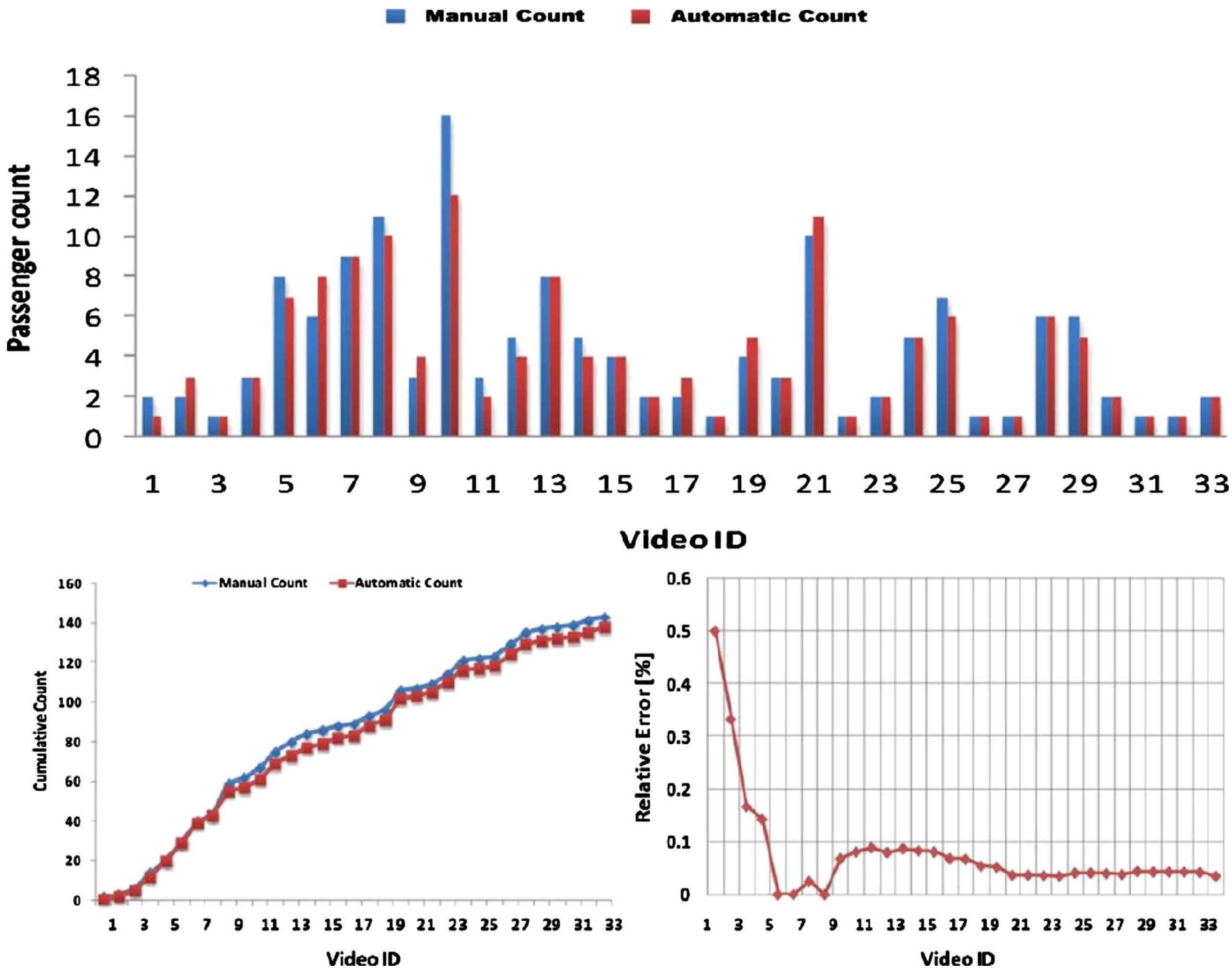

Fig. 9 Passenger-counting result for the traffic bus in Shanghai city. Top, number of passengers estimated by our approach (red bins) compared to ground truth (blue bins); bottom left, cumulative count result, bottom right, relative error of our approach. (Color online only.)

sequences captured from real traffic buses in Shanghai City. To the best of our knowledge, the existing single-CCD-camera-based passenger-counting system cannot handle such challenging conditions in a traffic bus in real time as does ours. Future work will concentrate on developing an embedded passenger-counting system for a traffic bus, and will extend our work to other people-counting applications.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 60903126 and No. 60872145), the China Postdoctoral Science Foundation (No. 20090451397), and the National High Technology Research and Development Program (863) of China (No. 2009AA01Z315).

References

1. B. Antoni, Z.-S. Chan, L. John, and V. Nuno, "Privacy preserving crowd monitoring: counting people without people models or tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–7, Alaska (2008).
2. T. Yahiaoui, C. Meurie, L. Khoudour, and F. Cabestaing, "A people counting system based on dense and close stereovision,", *Image Signal Process* **5099**, 59–66 (2008).
3. V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 705–711 (2008).
4. D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting," in *IEEE Proc. Int. Conf. on Pattern Recognition*, pp. 1187–1190 (2006).
5. W. Vandermark and D.-M. Gavrila, "Real-time dense stereo for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.* **7**(1), 38–50 (2006).
6. C. Beleznai, P. Sommer, and H. Bischof, "Scale-adaptive clustering for object detection and counting," in *Proc. IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 9–16, Rio de Janeiro, Brazil (2007).
7. Y.-J. Jeon and P. Rybski, "Analysis of a spatio-temporal clustering algorithm for counting people in a meeting," Carnegie Mellon University Technical Report CMU-RI-TR-06-04, CMU (2006).
8. G.-J. Brostow and R. Cipolla, "Unsupervised Bayesian detection of independent motion in crowds," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 594–601 (2006).
9. D.-B. Yang, H.-H. González-Banos, and L.-J. Guibas, "Counting people in crowds with a real-time network of simple image sensors," in *Proc. IEEE Int. Conf. on Computer Vision*, pp. 122–129 (2003).
10. F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multi-camera people tracking with a probabilistic occupancy map," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 267–282 (2007).
11. C.-H. Chen, Y.-C. Chang, T.-Y. Chen, and D.-J. Wang, "People counting system for getting in/out of a bus based on video processing," in *Proc. 8th Int. Conf. on Intelligent Systems Design and Applications*,

pp. 565–569, IEEE, Taiwan (2008).
12. H.-B. Yu, Z.-W. He, and J.-L. Liu, "A vision-based method to estimate passenger flow in bus," in *Proc. Int. Symp. on Intelligent Signal Processing and Communication Systems*, pp. 654–657, IEEE, Xiamen, China (2007).
13. O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer, "Pedestrian detection and tracking for counting applications in crowded situations," in *Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, pp. 70–75, Sydney, Australia (2006).
14. M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(4), 657–662 (2006).
15. T. Yang, Z. Stan, JingLi, Q. Pan, and J. Li, "Real-time object tracking with occlusion handling in dynamic scene," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 970–975, San Diego, CA (2005).
16. S. Birchfield, "KLT: an implementation of the Kanade-Lucas-Tomasi feature tracker," http://www.ces.clemson.edu/~stb/klt/.
17. J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593–600 (1994).
18. C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University Technical Report CMU-CS-91-132 (Apr. 1991).
19. Traf-SYS People counters, http://www.trafsys.com/people-counters.aspx.
20. Acore http://www.acorel.com/en/ ACOREL%20Onboard%20Counter.pdf.
21. J. W. Kim, K. S. Choi, B. D. Choi, J. Y. Lee, and S. J. Ko, "Real-time system for counting the number of passing people using a single camera," *Pattern Recogn.* **2781**, 466–473 (2003).
22. S. Velipasalar, Y. L. Tian, and A. Hampapur, "Automatic counting of interacting people by using a single uncalibrated camera," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, pp. 1265–1268 (2006).
23. P. Kilambi, E. Ribnick, A. J. Joshi, O. T. Masoud, and N. P. Papanikolopoulos, "Estimating pedestrian counts in groups," *Comput. Vis. Image Underst.* **110**, 43–59 (2008).

Tao Yang received his PhD degree from the School of Automation, Northwestern Polytechnical University, in 2008. He is currently with the Digital Video Processing Group, Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, School of Computer Science, Northwestern Polytechnical University. Prior to that, he was an intern at FX Palo Alto Laboratory (FXPAL), Palo Alto, California, from August 2006 to January 2007, a visiting student of National Laboratory of Pattern Recognition (NLPR), Beijing, China, from September 2004 to March 2005, and a visiting student with the Visual Computing Group, Microsoft Research Asia (MSRA), from April 2004 to June 2004. Dr. Yang's research interests include intelligent video surveillance, motion video registration, and moving camera moving object detection and tracking. He has published over 30 papers in international journals, conference proceedings, and Chinese key journals, including *International Journal of Image and Graphics*, the IEEE Conference on Computer Vision and Pattern Recognition, *ACM Multimedia*, the International Conference on Pattern Recognition, and the IEEE International Conference on Multimedia & Expo.

Yanning Zhang received her BS degree from Dalian University of Science and Engineering in 1988 and her MS and PhD degrees from Northwestern Polytechnical University in 1993 and 1996, respectively. She is currently a professor with the School of Computer Science, Northwestern Polytechnical University. She is also the organization chair of the Ninth Asian Conference on Computer Vision (ACCV2009). Her research work focuses on signal and image processing, computer vision, and pattern recognition. She has published over 200 papers in international journals, conference proceedings, and Chinese key journals.

Dapei Shao received his BS degree in 2008 from the School of Computer Science, Northwestern Polytechnical University, where he is currently pursuing his MS degree with the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing. His research interests are visual object detection and tracking.

Ying Li received her PhD degree from the State Key Laboratory of Radar Signal Processing Circuits and Systems, Xi'an University, in 2002. She is currently a professor with the School of Computer Science, Northwestern Polytechnical University. Her research interests include image processing, pattern recognition, and video surveillance. She has published over 50 papers in international journals, conference proceedings, and Chinese key journals.