

Video-based Automatic Transit Vehicle Ingress/Egress Counting using Trajectory Clustering

Guangchun Cheng¹, Yan Huang¹, Arash Mirzaei², Bill P. Buckles¹ and Hua Yang²

Abstract—In this paper we present an automatic vehicle ingress/egress counting method by clustering dense trajectories extracted from monitoring videos. Dense trajectories are extracted based on dense optical flow when passengers cross the door of the vehicle, and then clustered into different groups according to their descriptors with each legitimate group as a passenger. The contribution of the proposed method is twofold. First, we put forward an online passenger counting framework which is based on feature-points tracking and can be easily deployed to different scenarios. The method works even in low illumination conditions as demonstrated in experiments. Second, vehicle running information was combined to improve the accuracy of passenger counting. The transit vehicle settings are unconstrained and complex due to variations from illumination, movement and uncontrolled passenger behaviors. We tackle this by incorporating different modalities besides videos such as the status of the vehicle (e.g., in motion or not). The experimental results on multiple real bus videos show that the proposed system can count passengers with average accuracy of 94.9% at an average frame rate of 38 fps.

I. INTRODUCTION

Accurate ridership estimation is an essential step in intelligent transportation network evaluation and planning. Developing an automatic passenger counting (APC) system has both scientific and commercial interest. It requires an efficient solution to passenger detection and counting, and robust treatment to varying bus running states and lighting conditions. The application of APC can save time and labor for transportation agencies. In this work, we introduce an automated solution to automatic passenger counting problem by tabulating the number of vehicle ingress and egress.

There exist many people counting approaches and systems using different techniques and data sources. Attempts include contact-type counters and non-contact infrared beams and radio frequency sensors. While they can achieve high accuracy when applied properly, such equipment induces additional deployment and cost. Machine vision technologies provide an ideal potential for the passenger counting since video cameras have been widely mounted on buses and other transit vehicles. They use the same data sources (i.e. ordinary camera videos) as a human rider checker does. Therefore, the solution we present in this paper explores video analytics based on dense passenger trajectories, and

it achieves sufficient accuracy for passenger counting (i.e. 95%).

Vision-based counting and flow estimation have attracted interest in computer vision and intelligent systems communities, such as Refs. [1], [2], [3] for general people counting and Refs. [4], [5], [6], [7], [8] for passenger counting on public transportation. Existing approaches for passenger counting can be divided into the following three categories: body-part-recognition-based, passenger-segmentation-based and feature-point-tracking-based. Body-part-recognition-based approaches detect (and probably track) the body parts of a passenger such as heads and shoulders [3], [5], and then count based on the detection results. A challenge with body-part-recognition-based approaches lies in the fact that most bus surveillance videos are recorded in low illumination with low resolution, which makes it difficult to extract distinctive features of the passenger body parts. A similar challenge exists for segmentation-based approaches.

The proposed method in this paper can be categorized as among the feature-point-tracking-based approaches. It tackles the problems under low lighting conditions where traditional body-part recognition fails. In addition, the proposed method is easily applied to scenarios where the cameras are not mounted overhead, a condition under which most existing papers reported their results. The proposed method provides a complete solution to detecting type (i.e. boarding or alighting) and counting occurrences. We evaluate our approach using real bus surveillance videos captured by Fort Worth Transportation Authority (FWTA) in the USA.

A. Related Work

Trajectories have been utilized for action detection and recognition for a long time as reviewed in [9], and are still explored for video analysis [10], [11], [12]. Feature-point-based trajectories have been used more widely than object-based trajectories in the context of passenger counting [4], [6], [7], [8]. In [4] the authors divide image frames into blocks and select moving pixels based on eigenvalues of structure tensors. The direction of the movement is determined by the usage of two virtual base-lines. [6] takes a clustering-based approach. The authors cluster the obtained trajectories into trajectory groups, each of which then corresponds to one passenger. In addition, it uses only one virtual tripwire. However, the paper only tackles “getting in” without consideration on the discriminating between boarding and alighting. A real-time dense stereo-matching procedure is proposed in [8] for estimation of passenger flows. It computes a disparity map by a stereo-matching for each pair of images, and gets

¹G. Cheng, Y. Huang and B. P. Buckles are with Department of Computer Science and Engineering, University of North Texas, Denton, TX 76201, USA. G. Cheng: guangchuncheng@my.unt.edu, Y. Huang: huangyan@unt.edu, B. P. Buckles: bbuckles@cse.unt.edu

²A. Mirzaei and H. Yang are with Department of Transportation of North Central Texas Council Of Governments, Arlington, TX 76005, USA. A. Mirzaei: amirzaei@nctcog.org, H. Yang: hyang@nctcog.org

the trajectories of passengers' heads by segmentation and tracking. A passenger is counted when his/her head enters or leaves the stereo field of view. This approach requires multiple cameras to construct the stereoscope. In this paper, we provide a solution which uses traditional bus cameras and achieves efficient passenger boarding/alighting counting.

The remainder of this paper is organized as follows. We introduce our the approach in Section II, including trajectory generation and passenger counting by an online clustering algorithm. Section III presents experimental results on real-world datasets. We conclude the paper in Section IV.

II. THE PROPOSED PASSENGER COUNTING METHOD

A. System Overview

The flowchart of the proposed system is illustrated in Fig. 1. The system consists of two major modules. The first module collects trajectories by tracking feature points, and performs validation to eliminate unnecessary tracking and noisy trajectories. Dense optical flow is computed to track candidate points on a predefined tripwire. The second module performs online clustering for any legitimate trajectories based on their spatial and temporal description. To distinguish between boarding and alighting, we learn a weighted histogram of motion orientations (wHMO) for boarding and alighting respectively. Finally, passengers are counted according to the clustering results.

In the system we also explore non-visual information to improve the counting performance, both accuracy and efficiency. In this paper we exploit the use of vehicle running status. In absence of such information, our system can still count passengers with high accuracy, and we show performance improvement when incorporating such information.

B. Generating Trajectories of Feature Points

To obtain trajectories for passengers, there are basically three different ways: (1) detecting passengers and tracking them, (2) detecting keypoints of passengers and tracking them, and (3) tracking passengers only when they pass our region of interest. Considering the bus environment, we choose the last one as our trajectory generation strategy. We have also performed experiments with human body and face detection, and they both yielded unsatisfactory results for our datasets. Instead of tracking the whole passenger body, we track feature points that move with the passenger based on dense optical flow. We resort to OpenCV [13] to calculate optical flow fields with a window size of 31×31 and a maximum pyramid level of 3.

A region of interest is set to trigger our tracking process for the passengers. Unlike previous work specifying the region by two tripwires, our approach sets up one virtual tripwire as a set of locations passengers pass by. Because our approach is based on feature points tracking, the specific tripwire location is unimportant as long as it guarantees that passengers pass by it. In the experiments, we set the tripwire along the bottom of the bus door, but any lines nearly parallel to it would also be good choices as they are perpendicular to the direction passengers enter/exit the bus. Let $L = \{p_1, p_2, \dots, p_M\}$ be

M locations on the virtual tripwire, and $\Gamma = \{\tau_1, \tau_2, \dots, \tau_N\}$ be N trajectories that are actively tracked. For each frame from the video, we track the pixels at locations L if the bus is not in motion (see Section II-C), and add them to tracking list Γ . For the pixels that have already been in Γ , update Γ by tracking the last feature point of each trajectory τ_i based on optical flow.

In bus surveillance videos, there exist many cases when the tracking fails to obtain trajectories of passengers. We perform validation to the obtained trajectories Γ , including

- discontinuing tracking of stationary trajectories whose x and y variances are small and whose length is insufficient, and discarding them,
- discarding feature points which are lost in tracking and their corresponding trajectories, and
- discarding trajectories which are temporally too long, e.g. 120 frames, which should be determined based on the frame rate and specific application.

After the tracking and noisy trajectory removal, a number of trajectories are collected for each and every observed passenger. It is worth noting that all the trajectories start with the locations on the virtual tripwire no matter if it is boarding or alighting, and thus all the trajectories of the same passenger should have near-equal starting time. This observation will be used for clustering of the trajectories as in [6]. Figure 1 shows the trajectories of a passenger passing by the virtual tripwire.

C. Detection of Bus Motion Status for Tracking

For trajectory-based counting, accurate trajectories are crucial for the success in getting the correct number of passengers. During the operation, door opening and closing, a passenger's activities inside the bus and trajectories caused by light changes among others can lower the tracking accuracy. We found from observation of bus videos that most of such influences can be removed if we take into consideration the running status of the bus, which has not been explored in previous work to our best knowledge. We present a simple yet effective way for detecting the running status of the bus to assist feature point tracking. We confine the running status to whether the bus is in motion (i.e. "in motion" or "not in motion") in this paper.

We detect the bus motion status by modeling the status of the bus door, either open or closed. For frames when the door is closed, we compute a histogram of oriented gradients (HOG) [14] for the region where the door is located. Let h_t be the HOG histogram at time t , and g_t be the HOG histogram based on the information until time t . We update the histogram according to (1).

$$g_t = \begin{cases} \alpha g_{t-1} + (1 - \alpha) h_t & \text{if } \text{sim}(h_t, g_{t-1}) \geq \theta_D \\ g_{t-1} & \text{if } \text{sim}(h_t, g_{t-1}) < \theta_D \end{cases} \quad (1)$$

where $\text{sim}(h_t, g_{t-1})$ is a measurement of similarity between h_t and g_{t-1} , and θ_D is a threshold (0.9 in experiments). If $\text{sim}(h_t, g_{t-1}) < \theta_D$, we assume that the door is open

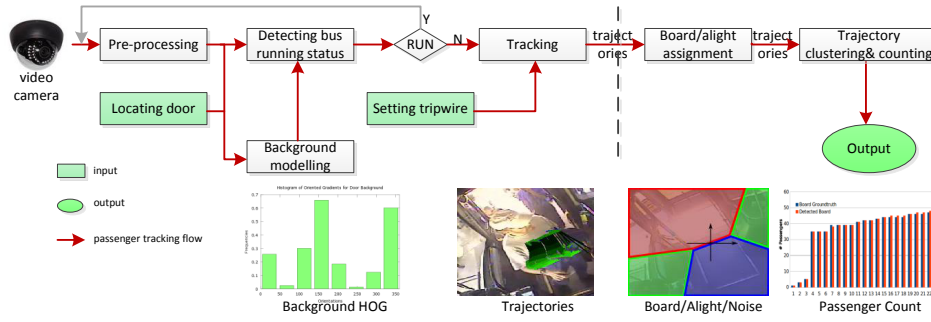


Fig. 1. Flowchart of our bus passenger counting system.

and the bus is not in motion; otherwise the door is closed. When the door is closed, we stop adding and tracking any new feature points. This avoids many trajectories of passengers lingering by the door as well as the door in the process of opening/closing. Because the light shadow usually changes while the bus is moving, the technique can also avoid tracking of the shadow of the bus and outside objects such as trees.

D. Trajectory Clustering and Counting

Passenger counting is fulfilled by clustering the trajectories into groups so that each group corresponds to a different passenger. Similar to [6], we use a temporal-features-based online clustering approach. However, we only use the appearance times of passengers based on the observation that passengers can have different disappearance times because of occlusion and speed of motion. To develop a complete bus passenger counting system, we distinguish between boarding and alighting which are not discussed in [6].

We propose a two-phase classification/clustering strategy based on spatiotemporal features of trajectories. There are two motion directions for the passengers: boarding and alighting. The first phase is classification of trajectories into either boarding or alighting using magnitude-weighted orientations of trajectories. We learn from samples the histogram of trajectory orientations. The second phase is online clustering which employs the temporal features of each trajectory to segment individual passengers. The objective is to count passengers of both boarding and alighting, respectively.

Weighted histogram of motion orientations: To determine whether a passenger is boarding or alighting, we compute a motion orientation histogram of all segments of a trajectory (referred to as wHMO hereafter). Since the segments can have different lengths, we weight the orientation of each segment by its length. After we obtain histograms for all trajectories during training, we train a classifier to get one representation histogram for boarding, alighting and invalid trajectories respectively. The invalid trajectories are those caused by motion of body parts or erroneous tracking. Let $\tau_j = \{(x_j^1, y_j^1), \dots, (x_j^n, y_j^n)\}$ be a trajectory. The procedure we used to construct the histogram of trajectory orientations is described next.

a). Calculate the orientation θ_j^i of each trajectory segment

$\{(x_j^i, y_j^i), (x_j^{i+1}, y_j^{i+1})\}$ as $\theta_j^i = \arctan\left(\frac{y_j^{i+1} - y_j^i}{x_j^{i+1} - x_j^i}\right)$, and each segment's length l_j^i , which is the Euclidean distance between two end points.

b). Quantize the orientations, and construct a histogram with N_θ bins out of $\{\theta_j^1, \dots, \theta_j^{n-1}\}$. When θ_j^i votes for k -th bin, it votes with its length l_j^i .

c). For all the histograms of the same type of passing (i.e. boarding, alighting or invalid), obtain the average histogram as its representation. Thus we have $\overline{h_B}$ for boarding, $\overline{h_A}$ for alighting and $\overline{h_I}$ for invalid trajectories as our passenger passing models.

Figure 2 shows histograms of motion orientations from three different types of trajectories. It is obvious that the histograms can be easily distinguished from one another. The wHMO's give an accurate representation to classify trajectories into boarding, alighting or invalid, as illustrated in 2(d) and 2(h). 2(d) is the result of dividing the camera coverage into regions according to 2(h). If a trajectory travels towards the red region, the passenger is likely alighting; if a trajectory goes towards the blue region, the passenger is likely boarding. Trajectories are invalid if they go to the green regions. There may exist some wrongly-classified trajectories because of the uncontrolled movement of passengers, and this problem is solved by the online clustering in next section.

During testing we perform the same histogram construction in steps a) and b). Suppose the histogram is h_{test} for a given test trajectory. We classify this trajectory by calculating the distance between h_{test} and each of the three passing models. So the passing model for h_{test} is found as

$$m^* = \arg \min_{m \in \{B, A, I\}} d(h_{test}, \overline{h_m}) \quad (2)$$

where $d(h_{test}, \overline{h_m})$ measures dissimilarity between h_{test} and $\overline{h_m}$. After classification, all the trajectories are divided into three categories: boarding, alighting and others, and we perform temporal-feature-based clustering for trajectories of boarding and alighting.

Online clustering for counting: The same passenger usually generates features with similar temporal and/or spatial properties. Since they start at a single tripwire, the trajectories of the same passenger should have very close starting time. Figure 3 shows some examples of the starting times

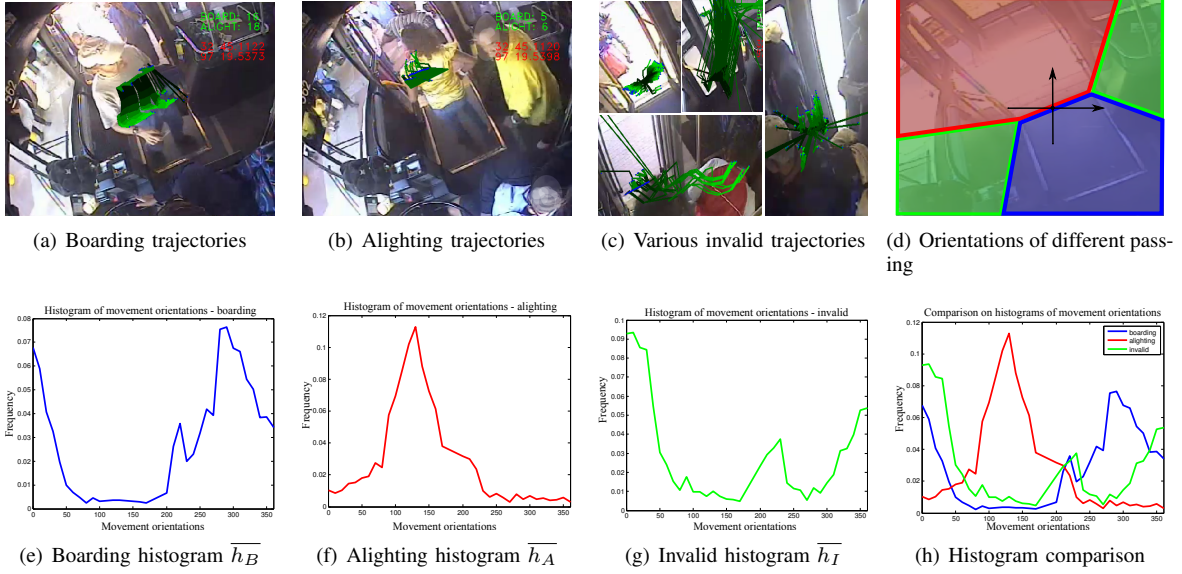


Fig. 2. Histograms of motion orientations and different types of trajectories. According to the histograms in 2(h), we can divide the camera coverage at the door into three regions shown in 2(d). The result coincides with commonsense regarding boarding and alighting.

of different passengers' trajectories. It can be seen that the starting times of different passengers are well separated. Most of the passenger trajectories in our dataset follow the same observation. Therefore, we base our online clustering algorithm on this observation by mapping multi-dimensional trajectories to 1-dimensional starting times. The temporal closeness among trajectories is used to measure the similarity between trajectories.

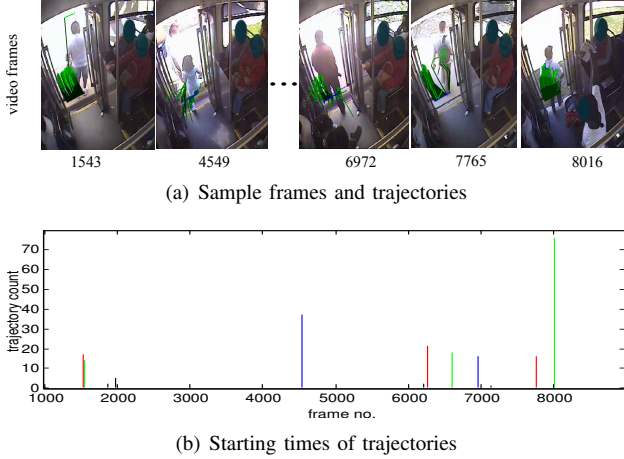


Fig. 3. Illustration of correspondence between separable starting times of different passengers.

Below we show the procedure of the online passengers counting. The same procedure is used for both boarding and alighting counting. We have two models, M_B and M_A , for boarding and alighting, respectively. $M_B = \{T_i | T_i = \{t_j | t_j \text{ is start time of a boarding passenger } j\}, 1 \leq j \leq N_B\}$ represents a set of N_B passengers, and $M_A = \{T_i | T_i = \{t_j | t_j \text{ is start time of an alighting passenger } j\}, 1 \leq j \leq N_B\}$. At the beginning before counting, $M_B = M_A = \emptyset$.

Each time a set of legitimate trajectories Γ is generated, the following algorithm updates M_B and M_A .

To calculate the similarity s_{kp} between τ_j^k and each passenger trajectory cluster T_p in the algorithm above, we first define similarity between two trajectories using their starting times. Suppose the starting time with trajectory τ_i is t_i , then the trajectory similarity s_{ij} between trajectories τ_i and τ_j is defined as $s_{ij} = e^{-|t_i - t_j|}$. If T_p contains s trajectories, we then have s similarities from τ_j^k to each of them. Denote them as $\{s_{kp}^i : 1 \leq i \leq s\}$. In order to reduce the influence of noisy trajectories, we use the median value of $\{s_{kp}^i : 1 \leq i \leq s\}$ as the similarity between the individual trajectory τ_j^k and a cluster of trajectories T_p of passenger p , which is s_{kp} .

For each frame t_j , we use the trajectories to update the boarding model M_B and the alighting model M_A . The processing is online in the sense that we dynamically add trajectories as new members to M_B or M_A when they are temporally faraway from all the members of M_B or M_A . The final passenger counts for boarding and alighting are the number of clusters which contain enough trajectories:

$$\#boarding = |\{T_i | T_i \in M_B, |T_i| \geq \theta_B\}|$$

$$\#alighting = |\{T_i | T_i \in M_A, |T_i| \geq \theta_A\}|$$

III. EVALUATION

In this section we describe the evaluation of the proposed approach using real bus surveillance videos. The performance is measured by counting accuracy and stop-by-stop counting statistics, overall counting accuracy, and speed for real-time processing.

A. Dataset and System Setup

The dataset used for evaluation consists of videos provided by North Central Texas Council of Governments

Algorithm 1 Algorithm for online trajectory clustering

Require: M_B : model for boarding, M_A : model for alighting, and Γ : a set of N trajectories;

- 1: Obtain all the times $\{t_j : 1 \leq j \leq N\}$ for the trajectories in Γ ;
- 2: **for each** τ_i in Γ **do**
- 3: Determine its passing model m_i using wHMO-based classification

$$m_i = \arg \min_{m \in \{B, A, I\}} d(h_{\tau_i}, \overline{h_m});$$
- 4: **end for**
- 5: **for each** time t_j **do**
- 6: Determine dominant passing model $m^+ \in \{B, A, I\}$;
- 7: Retrieve trajectories with starting time t_j and passing model m^+ ; let them be Γ_j ;
- 8: **if** $m^+ = I$ **then**
- 9: Remove Γ_j ; continue;
- 10: **end if**
- 11: **for each** trajectory $\tau_j^k \in \Gamma_j$ **do**
- 12: Calculate similarity s_{kp} between τ_j^k to each existing passenger cluster $T_p \in M_{m^+}$;
- 13: **if** $\max_{p \in [1, N_{m^+}]} \{s_{kp}\} \leq \theta_s$ **then**
- 14: Add $\{\tau_j^k\}$ to M_{m^+} as a new passenger candidate;
- 15: **else**
- 16: Add τ_j^k to the passenger T_{p^*} with maximum similarity s_{kp^*} ;
- 17: $N_{m^+} \leftarrow N_{m^+} + 1$;
- 18: **end if**
- 19: **end for**
- 20: **end for**
- 21: **return** M_B, M_A ;

(NCTCOG). The videos were recorded on buses operated by the Fort Worth Transportation Authority (FWTA). For each bus, there are five cameras mounted for a necessary coverage of the inside and the surroundings of the bus. For evaluation of the proposed approach, we make use of the videos from two of the cameras which cover boarding areas around the front and the rear doors. There are two hour long videos, with totally about 152,960 frames. The recording resolution is 580×450 . The ground truth of passenger count at each stop was obtained by manually checking the actual numbers of boardings and alightings.

In the following experiments, we manually selected the virtual tripwires. The histograms of motion orientations are automatically learned and updated along with the passenger counting. Two parameters have impact on the sensitivity of the counting results. The parameters specify the minimum number of trajectories in a cluster to be a valid passenger passing, θ_B for boarding and θ_A for alighting. Based on the dataset, we set both of them to 50. They could also be chosen by training by thresholding the clusters to give the best estimation with the ground truth.

B. Evaluation Results and Discussion

To evaluate the approach, experiments were performed on a one-hour video of the front door and a one-hour video of the rear door. The prototype achieves real-time computation speed and low error rates for boarding and alighting counts. Processing speeds of 38 fps on average are achieved, more or fewer depending on the number of trajectories to be processed. In the following, a report is given of the cumulative error rate, $E(s)$, of boarding and alighting detection, for which the metric is defined in equation (3). The ground truth is obtained manually for each video in dataset separately. The count tabulations of boarding and alighting are the cumulative number of passengers following each bus stop. The results for each stop (i.e., non-cumulative) are also described below.

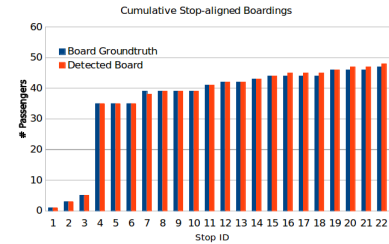
$$E(s) = \frac{|detection(s) - groundtruth(s)|}{groundtruth(s)} \times 100\%; \quad (3)$$

where s is a bus stop.

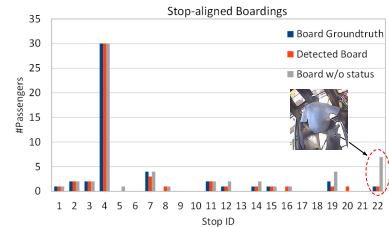
The analytic methods for the front and rear doors are exactly the same, though having different configurations for the virtual tripwires and coordinates of the door. Also, parameters for trajectory clustering have been fine-tuned separately for each door video. The parameter tuning is necessary because the two cameras have different mounting positions and viewing perspectives.

C. Results of Front Door Videos

The front door embeds more complex situations than the rear door, including mixed boarding-alighting, passenger inquiries, and the driver performing vehicle checks. For the current tracking-based approach, not all of the anomalous motion can be removed, but the prototype exhibits sufficient accuracy to serve decision makers.



(a) Cumulative boarding



(b) Each stop boarding

Fig. 4. Results on front door boarding. (a) is the comparison of cumulative passengers over the whole video, and (b) is a stop-by-stop comparison.

The boarding detection results for the front door camera are shown in Fig. 4. Fig. 4(a) shows the cumulative number

of boardings beginning with the first recorded stop and the one at right is the stop-by-stop results. The average error rate is 0.93%, and its standard deviation is 0.012. Fig. 4(b) is the comparison of detection and groundtruth on a stop-by-stop basis. Fig 5 is the results of alighting at the front door. The error rate is 9.22% with standard deviation 0.036. In Fig. 4(b) and Fig. 5(b), we also show the results without testing the vehicle running state. More counting errors occur due to passengers' unconstrained behaviors while the vehicle is in motion (i.e., doors are closed). One example is shown at stop-22 in the figures. The overall error rates without bus state testing are 7.66% and 10.35% for boarding and alighting respectively.

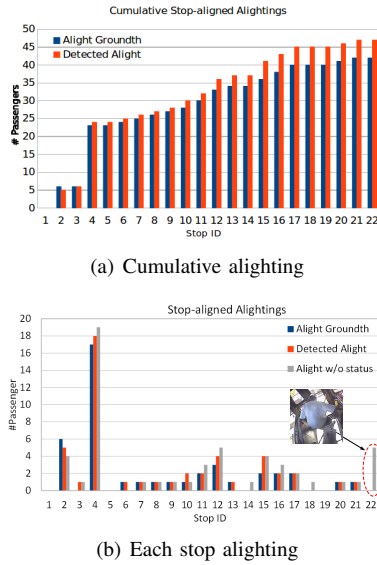


Fig. 5. Results on front door alighting. (a) is the comparison of cumulative passengers over the whole video, and (b) is a stop-by-stop comparison.

D. Results of Rear Door Camera

For the one-hour video, no boardings occurred at the rear door and the system detected none. Thus, we only report the results for alightings in Figure 6. The average cumulative error rate is 4.98%, and the standard deviation of the error rate is 0.084.

Overall, the proposed approach achieves 94.9% of accuracy on average for different types of passengers at both front and rear doors.

IV. CONCLUSIONS

We described an effective solution to automatic passenger counting using videos with perspectives on each bus portal. Based on feature points, the proposed solution achieves state-of-the-art performance under different conditions, including dramatical light changing and low illumination. On average, the computational speed is 38fps, which meets the real-time processing requirement. Experiments on trial videos show promising results for the counting of passengers boarding and alighting, i.e 94.9% counting accuracy on average. The spatial information of trajectories will be exploited for

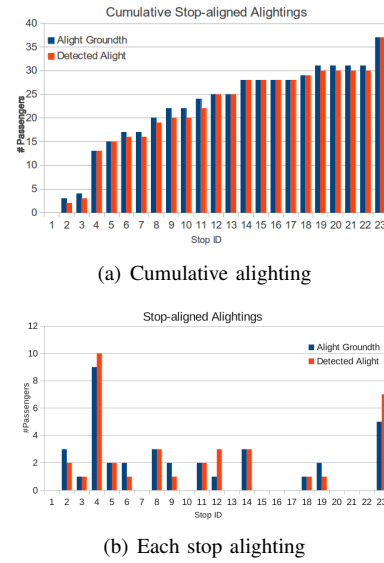


Fig. 6. Results on rear door alighting. (a) is the comparison of cumulative passengers over the whole video, and (b) is a stop-by-stop comparison.

clustering in future, which will allow detecting multiple passengers boarding or alighting simultaneously.

REFERENCES

- [1] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento, "Counting moving people in videos by salient points detection," in *International Conference on Pattern Recognition*, 2010, pp. 1743–1746.
- [2] K. Kopaczewski, M. Szczodrak, A. Czyzewski, and H. Krawczyk, "A method for counting people attending large public events," *Multimedia Tools and Applications*, vol. 65, no. 3, pp. 1–10, Aug 2013.
- [3] H. Xu, P. Lv, and L. Meng, "A people counting system based on head-shoulder detection and tracking in surveillance video," in *International Conference on Computer Design and Application*, 2010, pp. 394–398.
- [4] C.-H. Chen, Y.-C. Chang, C. Tsong-Yi, and D.-J. Wang, "People counting system for getting in/out of a bus based on video processing," in *International Conference on Intelligent Systems Design and Application*, 2008, pp. 565–569.
- [5] G. K.-H. Pang and C.-K. Ng, "Automated people counting using template matching and head search," in *IEEE Conference on Mechatronics and Machine Vision in Practice*, 2002, pp. 323–331.
- [6] T. Yang, Y. Zhang, D. Shao, and Y. Li, "Clustering method for counting passengers getting in a bus with single camera," *Optical Engineering*, vol. 49, no. 3, pp. 037 201.1–037 201.10, 2010.
- [7] H. Yan, "A bus passenger flow estimation method based on feature points trajectory clustering," in *IEEE International Conference on Intelligent Computing and Intelligent Systems*, Oct 2010, pp. 426–430.
- [8] T. Yahiaoui, L. Khoudour, and C. Meurie, "Real-time passenger counting in buses using dense stereovision," *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 031 202.1–031 202.11, 2010.
- [9] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.
- [10] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajecton: Action recognition through the motion analysis of tracked features," in *ICCV Workshops on Video-Oriented Object and Event Classification*, 2009.
- [11] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, 2013.
- [12] G. Cheng, Y. Wan, W. Santiterakul, S. Tang, and B. Buckles, "Action recognition with temporal relationships," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 671–675.
- [13] OpenCV. (2013, Sep.). [Online]. Available: <http://opencv.org>
- [14] N. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.