People-flow counting in complex environments by combining depth and color information

Chenqiang Gao[1] · Jun Liu[1] · Qi Feng[1] · Jing Lv[1]

Received: 8 October 2015 / Revised: 4 January 2016 / Accepted: 4 February 2016 /
Published online: 20 February 2016

Abstract People-flow counting is one of the key techniques of intelligence video surveillance systems and the information of people-flow obtained from this technique is an very important evidence for many applications, such as business analysis, staff planning, security, etc. Traditionally, the color image information based methods encounter kinds of challenges, such as shadows, illumination changing, cloth color, etc., while the depth information based methods suffer from lack of texture. In this paper, we propose an effective approach of people-flow counting by combining color and depth information. First, we adopt a background subtraction technique to fast obtain the moving regions on depth images. Second, the water filling algorithm is used to effectively detect head candidates on the moving regions. Then we use the SVM to recognize the real heads from the candidates. Finally, we adopt a weighted K Nearest Neighbor based multi-target tracking method to track each confirmed head and count the people through the surveillance region. Four datasets constructed from two surveillance scenes are used to evaluate the proposed method. Experimental results show that our method outperform the state-of-the-art methods. Our method can work stably on condition of kinds of interruptions and can not only obtain high precisions, but also high recalls on four datasets.

Keywords People-flow counting · Head detection · SVM · Multi-target tracking

1 Introduction

People-flow counting is one of the key techniques of intelligence video surveillance systems and this technique has many practical applications, such as business analysis, staff planning,

✉ Chenqiang Gao
gaocq@cqupt.edu.cn

[1] Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing, China

security, etc. Different from the people counting task which counts people within a given area, this task needs to count people moving along some directions. Thus, the sequential image information has to be utilized, while the former task can be done just based on the single image information.

A number of methods have been proposed to address the people-flow counting problem [2, 3, 11–13, 30]. As the video surveillance system is very common nowadays, these methods can be convenient and immediate to extensively apply in practice if they become mature. However, since the these methods just rely on the color video information, they inevitably encounter kinds of challenges, including shadow, illumination changing, cloth color, etc. These challenges usually make many of them not work stably in practical applications.

Since the depth image from depth sensors, such as Microsoft Kinects, can effectively avoid many of challenges mentioned previously, it is becoming hot to use depth information to solve traditional computer vision problems. Specifically, up to now, some depth information based people-flow counting methods have been presented [8–10, 19, 20, 24]. In practices, methods based on color information, depth information or both of them are usually suffered from the occlusion. To address this problem, a good strategy is to install the sensor in the top-down view, namely vertical installation. On this condition, as heads of persons and the other body parts have different distances to depth sensor, the head regions can be more easily detected using depth information than color information. However, since the depth image lacks the texture information, it is hard to discriminate real heads from other things with similar shape, even they have obviously different texture. For example, one person walks with a basketball in hand. This case can be easily recognized as that a adult walks with a kid if just using depth images. Actually, this situation is very common in some practical scenes, such as supermarkets, shopping malls, etc. These confusions usually lead to high false alarms. Thus, how to effectively recognize non-head objects is very important to improve the precision of people-flow counting. In the past decade, machine learning techniques have widely applied to solve kinds of practical application problems [16, 17, 27–29, 32]. This offer us an effective way to address the problem of non-head objects rejection.

In this paper, we propose a effective people-flow counting method in crowded environment with kinds of similar interruptions by combining depth and color image information from the Microsoft Kinect installed in the top-down view. We first fast and effectively detect the head candidates just using depth image information. This head candidate detection can work stably without influence of illumination conditions, shadows, cloth colors, etc. In order to effectively reject the false head detections with similar shapes to heads, we adopt the machine learning technique to recognize the real heads from the candidates. Experimental results demonstrate that our method outperform the state-of-the-art approaches and can work well even on condition of kinds of interruptions.

The rest of paper is organized as follows. The related works are reviewed in Section 2. In Section 3, we first describe briefly the proposed framework and then explain the key modules of the proposed method in details, including the moving object detection, hole filling, head candidate detection, false head rejection and multi-target tracking based on weighted KNN. The experiments and results are provided in Section 4. We give conclusions and future works in Section 5.

2 Related work

Generally speaking, most of methods for people-flow counting have two basic modules: individual detection and individual tracking. The former module is to detect persons from a scene. To do this, most methods are through head or head-shoulder detection because these two parts are most easily observed even in heavy crowded environments. As long as the people is detected, the off-the-shelf or new tracking methods can be used to complete the task of people-flow counting. These two basic modules can be done on color images, depth images or both of them (RGB-D). We will review the related work through three aspects: color images based methods, depth images based methods, and RGB-D images based methods.

Color images based methods Yam et al. [26] adopted a background modeling method to detect moving object and then used a simple feature matching method with the center information of the detected object to address the track problem. Lin et al. [13] located and tracked the head-shoulders of pedestrian via the integrated bottom-up/top-down processes. Raheja et al. [18] proposed a robust real time people-flow counting method by background modeling. The Chromatic color model and a multi-class feature based tracking algorithm were proposed to handle shadow and occlusion problems, respectively. Cai et al. [2] proposed a head detection method based on the boosted cascade of statistically effective multi-scale block local binary pattern features. The detected head is then tracked by a module matching method using harr-like feature.

Color images based methods are suitable for some applications with large, wide surveillance areas if the people are not very crowded and its illumination condition are kept under control.

Depth images based methods Hernandez et al. [9] proposed a people-flow counting with re-identification method. Zhang et al. [31] designed a people-flow counting system which captured the depth stream from Microsoft Kinect. They proposed a novel algorithm called water filling to detect people. This algorithm takes the depth image as a topology of geodesy, and finds local minimum regions in input depth image information based on the principle of water flowing downward. In this way, the regions detected are correspond to people heads. Then adding a nearest neighborhood multi-target tracking module to track each pedestrian. Zhu et al. [33] took a head and shoulder profile of a human as the input feature, and then the Adaboost algorithm was used to detect human objects. Finally, a Kalman based tracker was adopted to track the detected human objects and filter false detection.

The depth images based methods has great superiority over handling the challenges of shadow, illumination change, etc., compared to color images based methods. The depth information is very suitable for fast detecting object candidates.

RGB-D images based methods RGB-D images based methods can be considered as a kind of hybrid methods of depth and color information. How to reasonably to utilize the advantages of color information and depth information is crucial for designing a successful method. Dan et al. [5] proposed a people counting system based on fusing the depth and vision data. This method first recovered the depth image by combining the edge of object on depth and color image, then extracted the human object using a human model and tracked the detected object applying the bidirectional matching algorithm. Wateosot et al. [25] presented a top-view based people counting using mixture of depth and color information. This method use a background subtraction method to detect moving objects using depth images, and then a particle filter was used to track objects using the depth-color feature of the head. Fu et al. [7] used the color and depth information to count the people-flow. This method first detected the head of person on depth information and then counted each detected head on color information.

Since the depth information and color information are complementary, the hybrid of these two kinds of information is promising for handling the people-flow counting problem in practice.

3 The proposed method

3.1 The proposed framework

The framework of our method is presented in Fig. 1. It can be divided into three main modules: head candidate detection, off-line head classifier training and tracking for people-flow counting. In the head candidate detection module, we first use the background subtraction method to obtain the moving object regions and then a water filling algorithm [31] is adopted to detect head candidates just on depth image sequences. Based on the head candidate detection results, the corresponding color sub-images can be achieved on color images for following false detection rejection. In the off-line head classifier training module, we first construct a training data from color images, and then train a SVM classifier which is used to reject the false head candidates. In the tracking for people-flow counting module, a weighted *K* Nearest Neighbor [22] tracking method is used to track each confirmed head object. Finally, the tracking information is used to count people-flow in different directions. In the following sections, we will explain each key module in details.

3.2 Moving object region detection

Each person in the people-flow counting task is moving in some direction. Thus, the off-the-shelf moving object detection technique can be used to determine the person regions, and the following processing can be just based on these regions. This will reduce greatly the processing time of the whole algorithm, as well as false alarms.

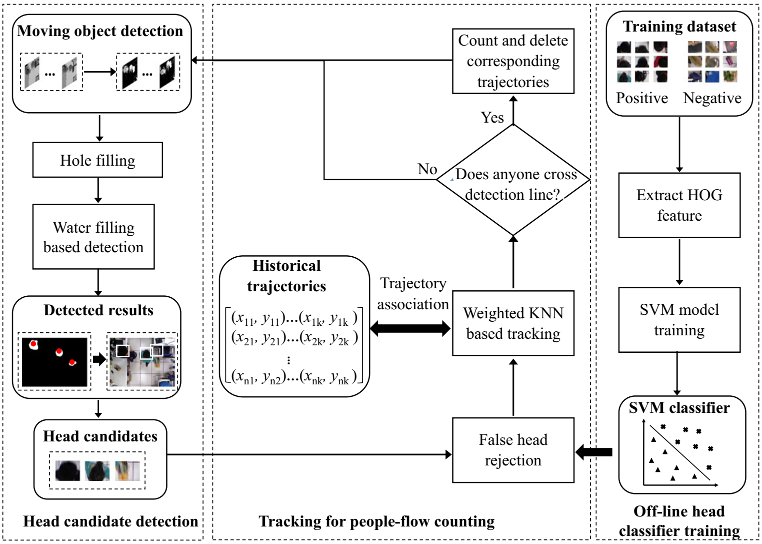

Moving object detection
Hole filling
Water filling based detection
Detected results
Head candidates
Head candidate detection
Count and delete corresponding trajectories
Yes
No
Does anyone cross detection line?
Historical trajectories
$(x_{11}, y_{11})...(x_{1k}, y_{1k})$
$(x_{21}, y_{21})...(x_{2k}, y_{2k})$
$(x_{n1}, y_{n2})...(x_{nk}, y_{nk})$
Trajectory association
Weighted KNN based tracking
False head rejection
Tracking for people-flow counting
Training dataset
Positive
Negative
Extract HOG feature
SVM model training
SVM classifier
Off-line head classifier training

Fig. 1 The framework of the proposed method

In this paper, we adopt the background subtraction [6, 15, 21] method based on the Gaussian Mixture Model (GMM) [32, 34] to detect moving regions. For each pixel location, the pixel value from the background is modeled by a mixture of multiple Gaussian distributions and the probability $P(X_t)$ of observing a given pixel value X_t can be estimated as follows:

$$P(X_t) = \sum_{i=1}^{M} w_{i,t}\eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{1}$$

where $w_{i,t}$ is the weight of the i^{th} Gaussian function in the mixture model at time t, M is the number of Gaussian distributions (Set to 3 in this paper), and η is a Gaussian probability density function.

The given pixel value is labeled as background if $|X_t - \mu_{i,t}| \leq d\sigma$, otherwise, it is labeled as foreground. In this paper, we detect the moving object regions on depth images, as shown in Fig. 2b.

3.3 Hole filling

Microsoft Kinect sensors used in this paper adopt the structural light technique to measure the distance between the sensor and an object. In fact, the sensor may fail to measure the

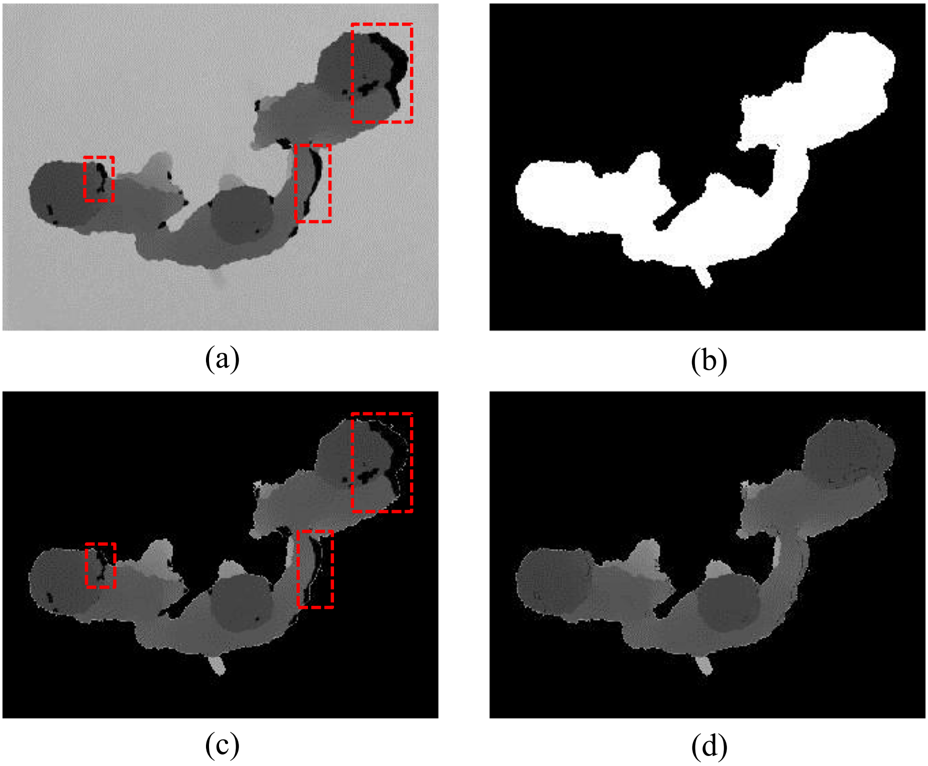


Fig. 2 Moving object detection and hole filling. **a** Raw depth image, **b** moving object regions after GMM, **c** foreground image after AND-operation of (a) and (b), **d** foreground image after repairing

depth information even within its valid measure range. For example, a glass door or a black-appearance object would both result in failure. For this situation, the corresponding pixel values will be set to be zero and thus there are some "holes" in the depth image, as the regions labelled with dotted box shown in Fig. 2a. These image areas with missing depth information have negative impact on the accuracy of head detection. Thus, it is necessary to repair the missing value in the acquired depth image. Besides, some pixels with small values still need to be repaired, since these pixels also lead to the shapes of "holes", which would be detected as heads in the following head detection based on the water filling algorithm. Therefore, all pixels with values less than a threshold θ are repaired, where the threshold θ is experimentally determined as 30 in this paper.

We use a simple strategy to repair these pixels, just by substituting the average of depth values within the neighborhood area for them, as used in [31]. After that, applying a median filter of a size of 3 × 3 to the repaired depth image would further improve its quality. To reduce the computation time, we just repair the "holes" on moving regions, as shown in Fig. 2c. The final repaired results can be seen in Fig. 2d.

3.4 Head candidate detection

Given the depth image $f(x, y)$, where f is the depth information of the pixel at (x, y), heads will correspond to the local minimum regions because our used Kinect sensor is installed in the top-down view. Thus, detecting heads equals to finding local minimum regions in f.

In this paper, we adopt the water filling algorithm [31] to effectively find the local minimum regions. This algorithm is inspired by the process of water filling. Namely, when raining, the water always flows to the hollows from humps. After raining, we can determine whether there are hollows on a land according to the amount of water. Actually, the form of one depth image $f(x, y)$ can be seen as a land, in which heads tend to be hollows. The details of the water filling algorithm can be found in [31].

3.5 False head rejection

The water filling algorithm detects the head candidate regions by searching the regions of local minimum pixel values (hollow regions) in depth images. Actually, many non-head objects have similar regions in depth images, and thus are easily detected as head candidates. As can be seen in Fig. 3a, a person is holding a kettle and walking through an aisle. For this situation, the kettle tends to be detected as a head if just relying on the depth information, as shown in Fig. 3c. In fact, these interruptions can be discriminated from heads on the color image, since they usually have obviously different characteristics, e.g, color, texture, etc.

In our scheme, we adopt the SVM to recognize real heads from the head candidates based on color information. A SVM constructs a hyperplane based on the the principle of structural risk minimization [23], which can be used for classification. Given a training dataset with l samples $(x_i, y_i), i = 1, 2, ..., l$, where $x_i \in R^n$ and $y_i \in \{1, -1\}^l$, the learning process of SVM can be expressed as the optimization problem as follows:

$$\begin{cases} \min\limits_{\omega,b,\varepsilon} \dfrac{1}{2}\omega^T\omega + C \cdot \sum\limits_{i=1}^{l} \varepsilon_i \\ s.t. \quad y_i(\omega^T\phi(x_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0, \end{cases} \tag{2}$$

where the offset of the hyperplane is determined by $\dfrac{b}{\omega}$, the $\phi(x_i)$ is a nonlinear function which maps the input space into a higher dimensional space and C is the regularization

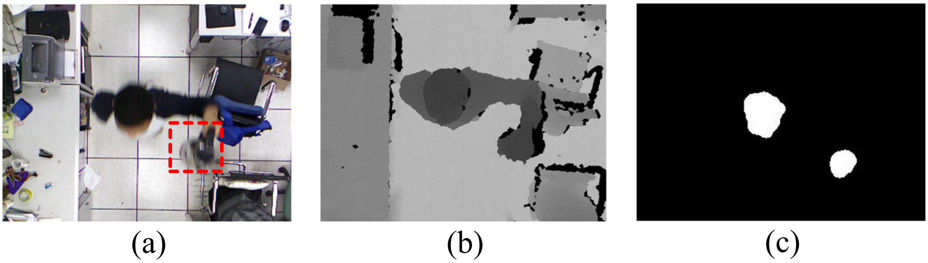


Fig. 3 An interruption would be identified as a person by the head detection algorithm. **a** Color image, **b** depth image, **c** detected result

parameter. In our method, the radial basis function kernel $K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$ is used, and the parameters C and γ are 12.5 and 2.25×10^{-3}, respectively, obtained by cross-validation.

We train the head classifier based on the HOG feature [4] and our training data is constructed from head candidates, where each sample is resized to the size of 60×50. According to the information of head detections in depth images, we can obtain the corresponding sub-images on color images. Then, we manually select the real head sub-images as the positive samples, and the rest as the negative samples. In this way, the diversity of the negative samples is usually not adequate for training a good head classifier. To address this problem, we augment the negative samples by iteratively mining the hard negative samples [14] in background images with interruptions, while keeping the positive samples fixed.

3.6 Multi-target tracking based on weighted KNN

In order to count the people walking along some directions, we need to track the detected heads across frames. In this paper, we adopt a simple and effective track method based on weighted KNN [22]. Here, we just use the location information of head regions to design the tracker, due to two aspects. On one hand, for the crowded situation, the head regions of pedestrians are close to each other. Besides, these head regions are usually similar to each other, including the shapes, color and texture. Thus, the tracking methods relying on shapes, color or texture [1] would not work stably. On the other hand, the pedestrians usually walk through a surveillance region with a normal velocity. Thus the locations of the head regions of the same pedestrian basically keep good continuity in sequential frames.

Given n detection results $(x_1, y_1), \cdots, (x_i, y_i), \cdots, (x_n, y_n)$ in the current frame and N historical trajectories, where the (x_i, y_i) is the center of the head region and the i^{th} trajectory $T_i^{n_i} = \{(x_{i,1}, y_{i,1}), \cdots, (x_{i,j}, y_{i,j}), \cdots, (x_{i,n_i}, y_{i,n_i})\}$, with n_i points. Our task is to associate the n detection results to N trajectories and to create new trajectories for the points out of association. Concretely, for each detection result (x_i, y_i), we need to find the corresponding trajectory label or create a new trajectory. To do this, we search the K nearest trajectory points from all ones $(x_{i,j}, y_{i,j})$, where $i = 1, \cdots, N$, and $j = 1, \cdots, n_i$, based on the Euclidean distance. Here, assuming that the distances of between the point (x_i, y_i) and K nearest trajectory points are $d_1, \cdots, d_i, \cdots, d_K$, we can calculate the corresponding weights based on the Gaussian function:

$$f(d) = exp(-\mu d^2), \tag{3}$$

where, d is the Euclidean distance, $\mu = 0.5$ in our experiment. Assuming that the K nearest points come from m trajectories, we can calculate the summation of weights for points from the same trajectory. Thus we can obtain m summations of weights, denoted as $S_1, \cdots, S_m$, and the corresponding trajectory labels are $L(x) = \{l_1, \cdots, l_x, \cdots, l_m\}$, respectively. Finally, the corresponding trajectory label l can be found by searching the maximum of m summations of weights, as follows:

$$i_{max} = \mathop{\arg\max}_{i \in (1,\cdots,m)} S_i. \tag{4}$$

Thus, the trajectory label is $L(i_{max})$. If the maximum of all summations is less than a threshold θ_S ($\theta_S = 3.45 \times 10^{-4}$ in our experiment), this detected head can be considered as a new one and we need to create a new trajectory for it. It is worth noting that there would be a situation that two or more detection results correspond to the same trajectory. For this situation, our strategy is that the trajectory label is associated to the one with the maximum of summations among these detection results. Then the others redo the previous search process within the trajectories excluding the currently searched trajectory until this situation disappears.

4 Experiments

In this section, we design several groups of experiments to validate the effectiveness of our method. First, we introduce the construction of our experimental datasets, including the experimental environment. Then, we evaluate the performance of our method on the head detection and the people-flow counting, respectively, with comparing to the state-of-the-art methods. Our method is implemented in C++ and all of the experiments run on a PC with Inter Celeron E3400 with 4.0 G memory.

4.1 Datasets and baseline methods

As shown in Fig. 4, the used datasets in this paper are collected from two surveillance scenes, where one is entrance of an elevator, while the other is the passageway inside a laboratory. The former has a clean background, but there are obvious shadows and illumination changing when pedestrians walk through the surveillance regions. The latter has relatively

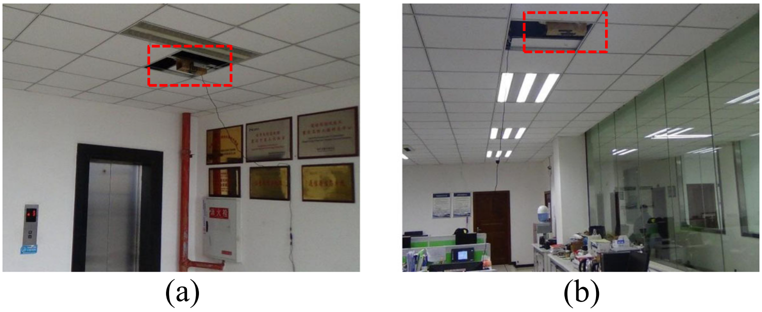


Fig. 4 Two surveillance scenes used for experiments. **a** The entrance of a elevator, and **b** the passageway inside a laboratory

Table 1 The details of four datatsts

Data	Samples	Crowded	Interruption	Background
Dataset 1	2532 images 67 people 1608 heads	Moderately crowded	Without interruptions	Clear background
Dataset 2	3968 images 127 people 2921 heads	Heavily crowded	Without interruptions	Clear background
Dataset 3	2145 images 56 people 1344 heads	Moderately crowded	With interruptions	Complex background
Dataset 4	3245 images 93 people 2469 heads	Heavily crowded	With interruptions	Complex background

stable illumination but has complex background cutter. In both scenes Kinect sensors are installed in top-down view, and the captured depth and color images are both of 640 × 480 sizes.

According to the pedestrian density and whether there are interruptions moving with pedestrians together, we construct four datassets, denoted as *dataset 1*, *dataset 2*, *dataset 3*, and *dataset 4*, respectively. All details of four datasets are listed in Table 1. It is worth noting that there are two kinds of groudtruths of people and heads for each dataset (Please see the column of *samples*). The number of people means how many pedestrians walk through the surveillance region, while the number of heads means how many heads there are in all images, where it is possible that several head samples come from a same pedestrian. The *people* samples are used to evaluate the performance of people-flow counting, while the *head* samples are used to train and test the head detection module in this paper, where the details of training and testing samples are listed in Table 2. Four representative images from four datasets, respectively, are shown in Fig. 5. Basically, our used datasets has a certain diversity and can represent the typically practical situations.

In this paper, two state-of-the-art methods are used as the baseline methods for comparisons, including the water filling based method [31], color gradient module based method [7].

Table 2 The number of training and testing samples from each dataset

	Train sample		Test sample	
	Positive	Negative	Positive	Negative
Dataset 1	350	359	1358	137
Dataset 2	1000	1117	1921	247
Dataset 3	750	1068	594	738
Dataset 4	1200	1769	1269	1311

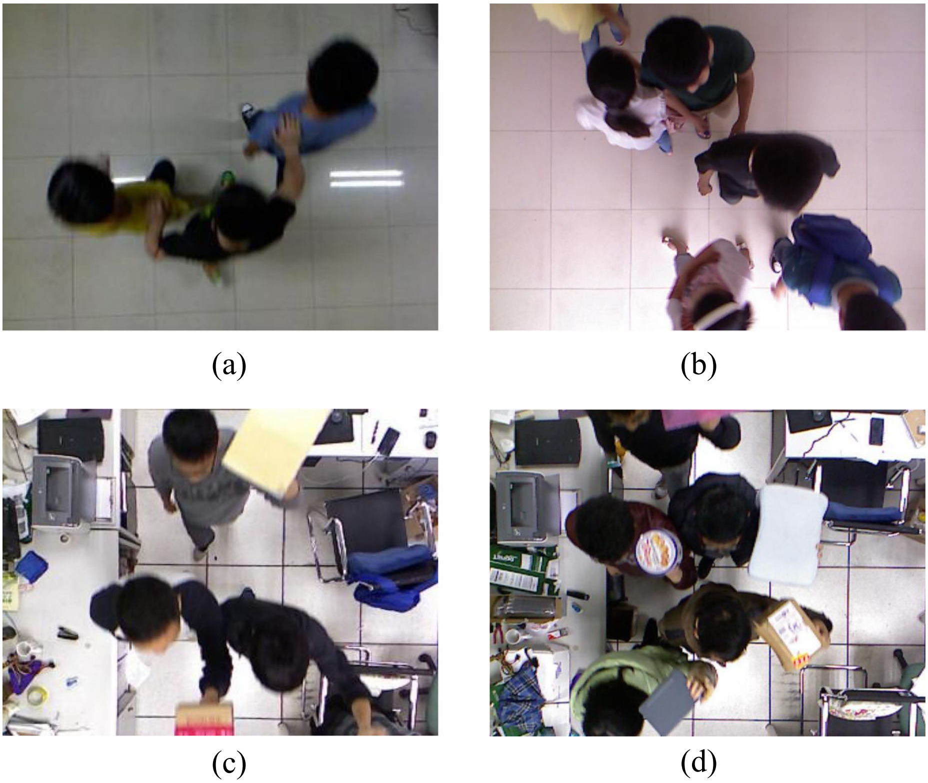


Fig. 5 Four representative images from four datasets. **a** One frame from *dataset 1* (moderately crowded and without interruptions), **b** one frame from *dataset 2* (heavily crowded and without interruptions), **c** one frame from *dataset 3* (moderately crowded and with interruptions), and **d** one frame from *dataset 4* (heavily crowded and with interruptions)

Water filling (WF) based method [31] This method first used the water filling algorithm to detect the head regions based on depth information, and then tracked and counted the detected head regions by a simple nearest neighborhood module. In our comparison experiment, the amount of water dropped one time (the parameter r in water filling algorithm) is set to 28.

Color gradient module (CGM) based method [7] This method proposed a scene adaptive head detection scheme to detect the head regions based on depth information, then tracked and counted the detected objects on color images. As used in [7], the parameters P and D are 255 and 170, respectively, in our comparison experiment.

4.2 Evaluation on head detection

The head detection is one of the key modules of our method. This module determines the final performance of people-flow counting to some extent. Specifically, within this module, the effectiveness of rejecting the interruptions in scenes is very important to improve the precision of people-flow counting.

Table 3 Comparisons of different methods on head detection

		Precision	Recall
	CGM [7]	88.64	92.16
Dataset 1	WF [31]	98.38	97.49
	Ours	98.56	98.31
	CGM [7]	86.24	89.59
Dataset 2	WF [31]	95.22	96.16
	Ours	97.85	98.11
	CGM [7]	69.51	92.04
Dataset 3	WF [31]	83.63	95.26
	Ours	95.96	97.66
	CGM [7]	66.07	88.67
Dataset 4	WF [31]	79.89	95.03
	Ours	94.37	96.78

The comparisons of several methods are listed in Table 3. We can observe that our method outperform other two baseline methods over both precision and recall. From the

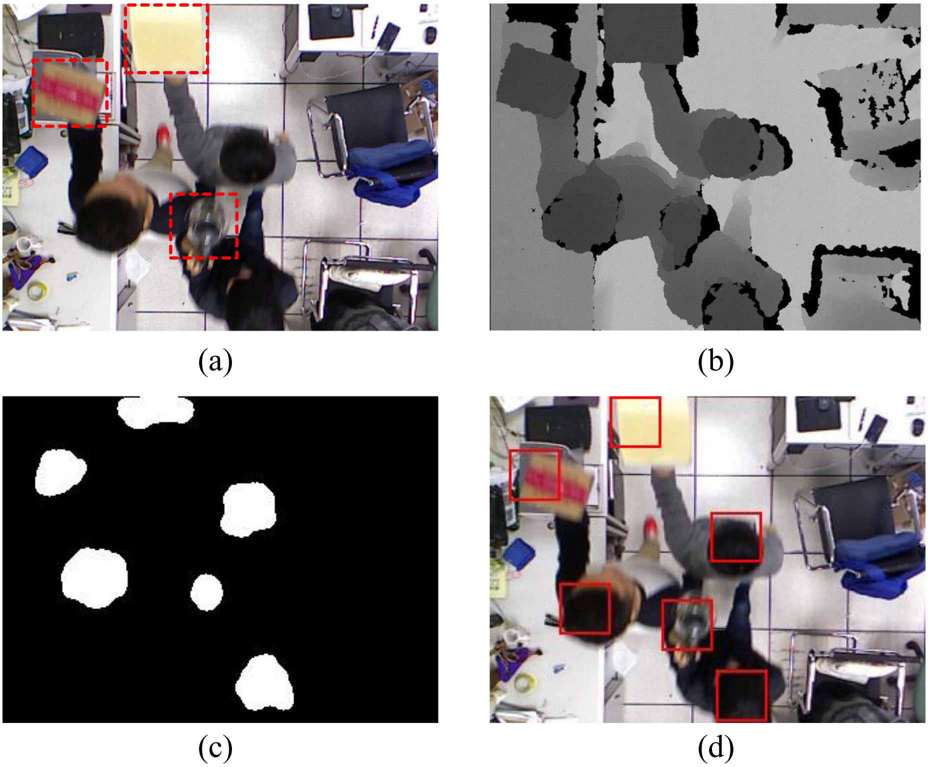


Fig. 6 The detection results without the head classification. **a** One frame of the color image channel, **b** the corresponding depth image, **c** the detected head regions, and **d** the corresponding head detection results in the color image

rows of *Dataset 1* and *Dataset 2*, we can observe that both our method and WF method have good performance for the cases without interruptions, noting that our head detection module is based on the water filling method. This means that the water filling algorithm can works well under the condition of vertical installation of Kinect sensors. However, from the rows of *Dataset 3* and *Dataset 4*, we can see that for the cases of the interruptions, the performance of both of two baseline methods degrade greatly. This is since these interruptions are very similar to heads on depth images and thus are easily detected as heads. This case can be observed in Fig. 6, from which we can see that three interruptions moving with pedestrians are all detected as the heads (please see the Fig. 6d). In contrast, our method can handle this case very well. The precisions of our method on *Dataset 3* and *Dataset 4* reach 95.96% and 94.37%, respectively. There are big improvements over the baseline methods. This is since our head classification module can effectively reject these interruptions based on the color information, which can be easily observed by comparing Figs. 6d to 7d.

4.3 Evaluation on people-flow counting

The performance comparisons of different methods are listed in Table 4. We can observe that our method is the best among three methods on all four test datasets. For the cases of

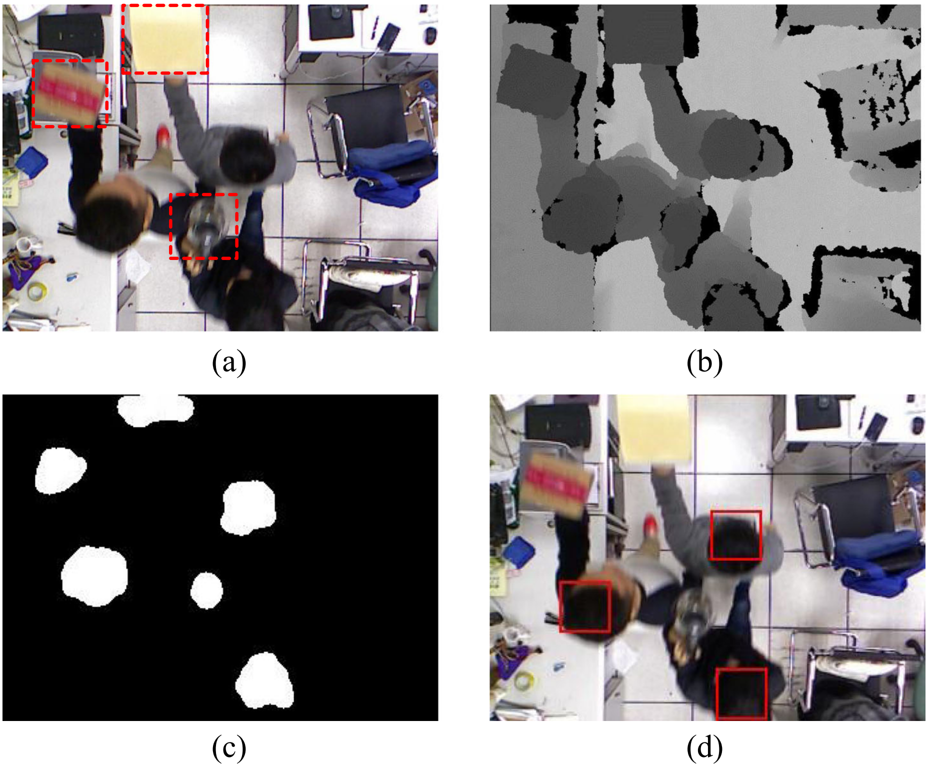


Fig. 7 The detection results with head classification. **a** One frame of the color image channel, **b** the corresponding depth image, **c** the detected head regions, and **d** the corresponding head detection results in color image

Table 4 Performance comparisons of different methods on people-flow counting

	Method	Data type	Precision	Recall
	CGM [7]	Depth+color	82.21	91.02
Dataset 1	WF [31]	Depth	93.76	92.17
	Ours	Depth+color	98.16	97.97
	CGM [7]	Depth+color	78.34	88.73
Dataset 2	WF [31]	Depth	87.38	90.19
	Ours	Depth+color	96.87	96.91
	CGM [7]	Depth+color	66.42	90.15
Dataset 3	WF [31]	Depth	81.65	91.57
	Ours	Depth+color	94.95	96.91
	CGM [7]	Depth+color	65.84	87.63
Dataset 4	WF [31]	Depth	77.69	89.83
	Ours	Depth+color	93.28	96.52

interruptions, the improvements of the precisions and recalls of our method are more than 13 percent and 5 percent, respectively, compared to the baseline methods (please see the rows of *Dataset 3* and *Dataset 4*). Meanwhile, compared *WF* to *CGM*, we can see that the former can obtain relatively better results, although the it just relies on the depth information. This means that an effective head detection module is crucial for performance of people-flow counting. Additionally, an effective interruption rejection module is also very important. This can be easily observed by comparing our method with *WF*. Although our method just uses a simple track method, both the precision and recall of our method are better than *WF* on four datasets.

In term of influence factors of pedestrian density and interruptions, from Table 4, we can observe that the influence of interruptions is more obvious than the pedestrian density. Commonly, the crowded pedestrians (a high pedestrian density) will lead to occlusion. As our sensor is installed in the top-down view, this problem is well relieved. However, the crowded situation will make the tracking more difficult. Thus, the performance will degrade

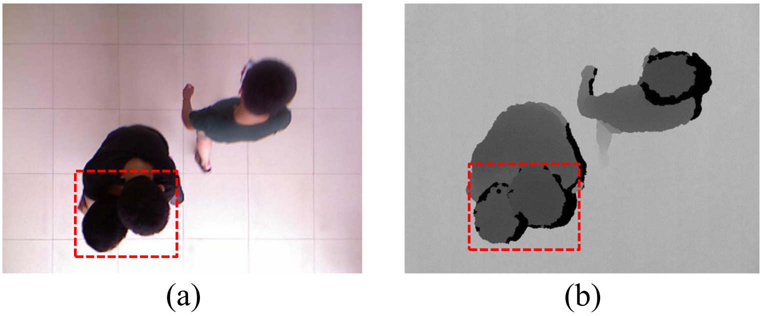


Fig. 8 A failure case of our method: a person carry another one. **a** The color image, and **b** the corresponding depth image

for all testing methods. This can be seen by comparing the rows of *Dataset 1* with *Dataset 2*, and comparing the rows of *Dataset 3* with *Dataset 4*, respectively.

5 Conclusion

In this paper, we propose an effective approach of people-flow counting by combining color and depth information. We adopt the water filling algorithm to effectively detect head candidates on the moving regions of depth images. Then we use the SVM to recognize the real heads from the candidates. Finally, we adopt a weighted KNN based multi-target tracking method to track each confirmed head and the tracking information is further used to count the people-flow. Experimental results show that our method outperform the state-of-the-art methods.

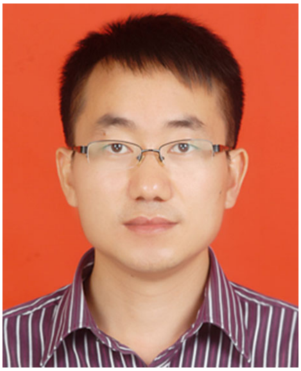
For some specific cases, our method would fail to count the pedestrians. For example, a person carries another one, as shown in Fig. 8. For this case, the heads of two person have overlap, which leads to that two heads are detected as one head by the water filling algorithm. In the future, we will use kinds of clues to improve the precision of people-flow counting. These clues include shape, area, etc.

Acknowledgments This work is supported by the National Natural Science Foundation of China (No. 61571071, 61102131), the Natural Science Foundation of Chongqing Science and Technology Commission (No. cstc2014jcyjA40048), Wenfeng innovation and start-up project of Chongqing University of Posts and Telecommunications (No. WF201404).

References

1. Benfold B, Reid I (2011) Stable multi-target tracking in real-time surveillance video. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3457–3464
2. Cai Z, Yu ZL, Liu H, Zhang K (2014) Counting people in crowded scenes by video analyzing. In: 2014 IEEE 9th conference on industrial electronics and applications (ICIEA). IEEE, pp 1841–1845
3. Chen TH, Chen TY, Chen ZX (2006) An intelligent people-flow counting method for passing through a gate. In: 2006 IEEE conference on robotics, automation and mechatronics. IEEE, pp 1–6
4. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, CVPR 2005, vol 1. IEEE, pp 886–893
5. Dan BK, Kim YS, Jung JY, Ko SJ et al. (2012) Robust people counting system based on sensor fusion. In: IEEE transactions on consumer electronics, vol 58, pp 1013–1021
6. Evangelio RH, Patzold M, Keller I, Sikora T (2014) Adaptively splitted gmm with feedback improvement for the task of background subtraction. In: IEEE transactions on information forensics and security, vol 9, pp 863–874
7. Fu H, Ma H, Xiao H (2014) Scene-adaptive accurate and fast vertical crowd counting via joint using depth and color information. Multimedia Tools and Applications 73(1):273–289
8. Galčík F, Gargalík R. (2013) Real-time depth map based people counting. In: Advanced concepts for intelligent vision systems. Springer, pp 330–341
9. Hernandez D, Castrillon M, Lorenzo J (2011) People counting with re-identification using depth cameras. IET Conference Proceedings pp 16–16(1)
10. Hsieh CT, Wang HC, Wu YK, Chang LC, Kuo TK (2012) A kinect-based people-flow counting system. In: 2012 international symposium on intelligent signal processing and communications systems (ISPACS). IEEE, pp 146–150

11. Hu Y, Zhou P, Zhou H (2011) A new fast and robust method based on head detection for people-flow counting system. International Journal of Information Engineering 1(1):33–43
12. Li B, Zhang J, Zhang Z, Xu Y (2014) A people counting method based on head detection and tracking. In: 2014 international conference on smart computing (SMARTCOMP). IEEE, pp 136–141
13. Lin Y, Liu N (2012) Integrating bottom-up and top-down processes for accurate pedestrian counting. In: 2012 21st international conference on pattern recognition (ICPR). IEEE, pp 2508–2511
14. Liu Y, Nie L, Han L, Zhang L, Rosenblum DS (2015) Action2activity: recognizing complex activities from sensor data. pp 1617–1623
15. Mukherjee S, Das K (2013) An adaptive gmm approach to background subtraction for application in real time surveillance. arXiv:1307.5800
16. Nie L, Wang M, Zha ZJ, Chua TS (2012) Oracle in image search: a content-based approach to performance prediction. ACM Trans Inf Syst (TOIS) 30(2):13
17. Nie L, Yan S, Wang M, Hong R, Chua TS (2012) Harvesting visual concepts for image search with complex queries. In: Proceedings of the 20th ACM international conference on multimedia. ACM, pp 59–68
18. Raheja J, Kalita S, Dutta PJ, Lovendra S (2012) A robust real time people tracking and counting incorporating shadow detection and removal. Int J Comput Appl 46(4):51–58
19. Rauter M (2013) Reliable human detection and tracking in top-view depth images. In: 2013 IEEE conference on computer vision and pattern recognition workshops (CVPRW). IEEE, pp 529–534
20. Stahlschmidt C, Gavriilidis A, Kummert A (2013) Density measurements from a top-view position using a time-of-flight camera. In: Proceedings of the 8th international workshop on Multidimensional systems (nDS), 2013. VDE, pp 1–6
21. Suresh S, Deepak P, Chitra K (2014) An efficient low cost background subtraction method to extract foreground object during human tracking. In: 2014 international conference on circuit, power and computing technologies (ICCPCT). IEEE, pp 1432–1436
22. Tan S (2005) Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Syst Appl 28(4):667–671
23. Tong S, Koller D (2002) Support vector machine active learning with applications to text classification. J Mach Learn Res 2:45–66
24. Van Oosterhout T, Bakkes S, Kröse BJ (2011) Head detection in stereo data for people counting and segmentation. In: VISAPP, pp 620–625
25. Wateosot C, Suvonvorn N (2013) Top-view based people counting using mixture of depth and color information. In: The second asian conference on information systems, ACIS
26. Yam KY, Siu WC, Law NF, Chan CK (2011) Effective bi-directional people flow counting for real time surveillance system. In: Proceedings, ICCE, vol 11, pp 863–864
27. Yan Y, Ricci E, Subramanian R, Lanz O, Sebe N (2013) No matter where you are: flexible graph-guided multi-task learning for multi-view head pose classification under target motion. In: 2013 IEEE international conference on computer vision (ICCV). IEEE, pp 1177–1184
28. Yan Y, Ricci E, Subramanian R, Liu G, Lanz O, Sebe N (2015) A multi-task learning framework for head pose estimation under target motion. IEEE Trans Pattern Anal Mach Intell PP(99):1–1. doi:10.1109/TPAMI.2015.2477843
29. Yan Y, Shen H, Liu G, Ma Z, Gao C, Sebe N (2014) Glocal tells you more: coupling glocal structural for feature selection with sparsity for image and video classification. Comput Vis Image Underst 124:99–109
30. Zeng C, Ma H (2010) Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In: 2010 20th international conference on pattern recognition (ICPR). IEEE, pp 2069–2072
31. Zhang X, Yan J, Feng S, Lei Z, Yi D, Li SZ (2012) Water filling: unsupervised people counting via vertical kinect sensor 2012 IEEE 9th international conference on advanced video and signal-based surveillance (AVSS). IEEE, pp 215–220
32. Zhao YL, Nie L, Wang X, Chua TS (2014) Personalized recommendations of locally interesting venues to tourists via cross-region community matching. ACM Trans Intell Syst Technol (TIST) 5(3):50
33. Zhu L, Wong KH (2013) Human tracking and counting using the kinect range sensor based on adaboost and kalman filter. In: Advances in visual computing. Springer, pp 582–591
34. Zivkovic Z (2004) Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition, ICPR 2004, vol 2. IEEE, pp 28–31



Chenqiang Gao is a professor in Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include image processing, computer vision, machine learning.

Jun Liu is currently working towards the M.S degree in information and telecommunication engineering from Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include people-flow counting and multi-target tracking.

Qi Feng is currently studying for the M.S degree in information and telecommunication engineering from Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include digital image processing, computer vision and machine learning.

Jing Lv is currently studying for the M.S degree in information and telecommunication engineering from Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include computer vision and action recognition.