12th International Conference on Advanced Computational Intelligence (ICACI)
Dali, China, August 14-16, 2020

A Real-Time Pedestrian Counting System Based on RGB-D

1st Yang Yao
1st *Science and Technology on Communication Information Security Control Laboratory*
2nd *State Key Laboratory for Novel Software Technology*
Nanjing University
Nanjing, China
yaoyang@smail.nju.edu.cn

2nd Xu Zhang, 3rd Yu Liang, 5th Furao Shen
2nd *State Key Laboratory for Novel Software Technology*
Nanjing University
Nanjing, China
zhangxu037@smail.nju.edu.cn,
liangyu9256@163.com,
frshen@nju.edu.cn

4th Xin Zhang
3rd *School of Marine Science* and Technology
Northwestern Polytechnical University
XiAn, China
zhangxin@nwpu.edu.cn

6th Jian Zhao
4th *School of Electronic Science and Engineering*
Nanjing University
Nanjing, China
jianzhao@nju.edu.cn

***Abstract*—Pedestrian counting is an important task in visual surveillance. Existing computer vision based pedestrian counting methods usually require high image quality and simple background, hence their applications are limited. In this paper, we introduce an innovative RGB-D based system for real-time pedestrian counting, which is easy to install and robust to work under various conditions. The system uses a RGB-Time Of Flight (TOF) camera to capture depth and RGB images simultaneously, then detects and tracks pedestrians based on fusion of both images. We propose a Deep Convex Convolution Filtering (DCCF) algorithm for pedestrian detection in depth images in order to overcome the problem of parameter sensitiveness in traditional methods. Experimental results highlight the effectiveness and efficiency of our designed system. Our proposed system has already been put to work in public places of multiple cities successfully.**

***Keywords*—Pedestrian counting, TOF, RGB-D, real time**

I. Introduction

Pedestrian counting is an important task in visual surveillance that provides valuable information for management of public spaces like transportation and supermarket. Its objective is to count the number of pedestrians passing through a specific location from different directions (in/out). Traditional methods [1] require pedestrians to pass through the monitoring areas of specific sensors, such as revolving doors, lasers, infrared, pressure, etc. Those sensors usually limit the speed or direction of passing pedestrians to achieve accurate results, hence cause inconvenience for pedestrians. Moreover, it usually needs to change the environment to install the sensors.

In recent years, pedestrian counting systems using computer vision techniques [2] have been developed. Nevertheless, vision based methods tend to fail in varying conditions, such as low image quality, complicated background, poor lighting etc. Deep learning could achieve remarkable results in various visual tasks such as pedestrian detection [3], crowd behavior analysis [4] and other video monitoring applications. However, deep learning algorithms must rely on expensive computing devices and large data sets. In addition, due to the resource limitation, it is also very difficult for deep learning to achieve real-time performance. In fact, this is the main reason for limiting the application of deep learning in practice.

In this paper, we introduce an innovative **RGB-D** based **P**edestrian **C**ounting **S**ystem (RGBD-PCS) that has the following advantages:

1) The necessary hardwares of the system is inexpensive and easy to install, and the system does not need access to networks. Hence it could be implemented in challenging scenarios like buses, where bulky computing devices or stable network connections could not be provided.
2) The algorithms in the system are computationally efficient. In the case of scarce computing resources on embedded devices, the system can still achieve **24fps** processing speed, which could meet the requirement of real-time.
3) The proposed system is robust enough to work in various environments with different backgrounds and lighting conditions, even in the completely **dark environment**.
4) The proposed system could achieve satisfying pedestrian counting accuracy, even under some **hard cases**.

The rest parts of this paper is organized as following: related works are introduced in section II, RGBD-PCS is described

This work is supported in part by the National Natural Science Foundation of China under Grant Nos. (61876076), Jiangsu NSF grant (BK20171344).

in section III, section IV presents experimental results, section V concludes the paper.

II. Related Works

The computer vision based methods are widely used in people counting because of its easiness of installation and accuracy. In recent years, deep learning has achieved remarkable results in various visual tasks such as pedestrian detection, crowd behavior analysis and other video monitoring applications. About the algorithm of pedestrian counting, Sermanet et al. [5] pointed out that the features obtained from deep learning model is more differentiated than manual features. Zhang et al. [6] proposed a simple and effective multi-column convolution neural network (MCNN) architecture to map images to population density maps. There is also a series of methods based on RNN [7], Fast-RCNN [8] and other models. For obtaining higher accuracy, modern deep networks tend to be deeper and wider which need more parameters and computation. The large scale of deep learning make it difficult to deploy state-of-art CNN models on the resource-constrained platforms. On the other hand, it also cannot satisfy the real-time performance due to the slow processing speed. According to the demand of real system, general deep learning is not suitable in practice currently.

Except deep learning models, there are also many other traditional methods applied on RGB video. The authors of [9] have proposed a blob-based system to estimate the number of people in urban environments. However, the blob-based methods can only detect moving objects [10], [11]. For avoiding the mutual occlusion between each pedestrian in multi-pedestrian counting, Rossi et al. [11] and Sexton et al. [10] mounted the camera vertically with respect to the floor plane (the way our system adopted), and pedestrian is observed from just overhead. They use Histograms of Oriented Gradients (HOG) features [11] or combination of HOG and Local Binary Pattern (LBP) [10] features to distinguish the head from the background respectively. Li [12] applied the HOG features [11] to detect head-shoulder of pedestrians. The merit of the head-shoulder detection is its effectiveness of reducing the partial occlusion. However, the accuracy of these methods [10]–[12] does not reach the ideal state because of the insufficient differentiation of manual features. In addition, the RGB image will fail in the case of poor lighting conditions. So only using RGB images cannot meet the demand of the system to adapt to the different environments.

Another kind of methods is to perform pedestrian counting by means of depth images based on Time of Flight (ToF) camera [13]–[15] or structure light camera [16]. Almost all these works are based on overhead cameras, with the objective of reducing the occlusion effects. In [13], they use ToF camera (Canesta EP205 with a resolution of 64 × 64 pixels) in low illumination conditions. However, it only works well if people enter the scene one by one. After the advent of Microsoft Kinect that provides depth image through structured light, researchers have proposed new approaches that use this acquisition device. However, the raw depth image obtained from camera is full of noise, which makes it difficult to analyze. On the other hand, depth cameras cannot work under strong illumination conditions because of the imaging principle of depth cameras [13]. Due to both reasons above, we need a more effective and robust algorithm to process depth image in advance.

III. System Description: RGBD-PCS

A. Overview

The objective of our proposed system, namely RGBD-PCS, is counting the number of pedestrians that pass through the specific monitoring area. The system consists of four components: image capture, detecting, tracking and counting, as illustrated in Fig. 1. We will simply introduce these components in overview.

Firstly, we build up the image capture module. RGBD-PCS is equipped with a RGB-Time Of Flight (TOF) camera with overlooking perspective to capture depth and RGB images of the monitoring area simultaneously. This camera connects with a embedded processor such as ARM which is responsible for deploying the model and interacting with users.

Secondly, we detect heads of pedestrians based on RGB-D images. We propose an effective and efficient head detection approach combining RGB channel and depth channel. We leverage the HOG feature [11] and SVM classifier [11] to detect pedestrians in RGB images and propose a Deep Convex Convolution Filtering (DCCF) algorithm to obtain *Depth Filter Images*, from which pedestrians are detected. Our system can switch among different modes automatically to deal with various and complicated scenarios.

Next, we track the detected pedestrians until they leave the monitoring area. We design a novel strategy combined KCF tracking model and detection results given by last step, in order to achieve real-time performance.

Finally, according to trajectories we have obtained in tracking step, we can count the number of pedestrians with respect to their moving directions.

B. Monitoring area and image capture

As illustrated in Fig.2(a), the monitoring area is defined as the region between line1 and line3 (the green region) according to the requirements of application. We mount the RGB-TOF camera (the yellow region which is labeled ”C” in Fig.2(b)) on the top of the monitoring area with a overlooking perspective for detecting and tracking passing people. Such camera configuration facilitates pedestrian counting because the heads of every person can be directly observed from the vintage point without occlusion. The TOF camera continuously emits light pulses to pedestrians, and then uses the sensor to receive the light reflected from pedestrians. The distances between the camera and pedestrians are obtained by computing the flight time of light pulse, and then the depth image is rendered accordingly.

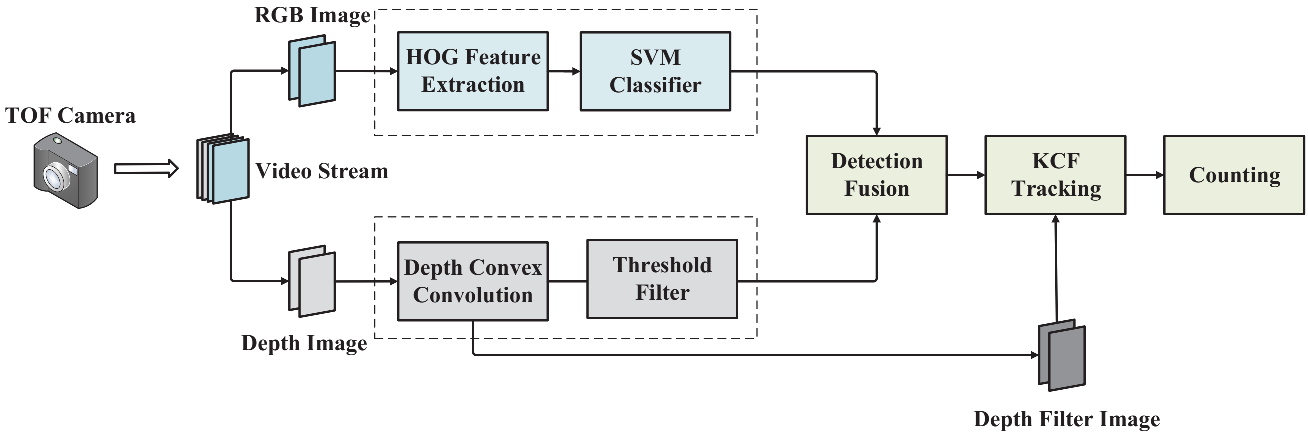


Fig. 1. The flow chart of RGBD-PCS

C. Pedestrian detection

In order to increase the accuracy of pedestrian detection and the robustness of the system, we first detect pedestrians in RGB and depth images respectively and obtain a set of proposal bounding box from each image, then we use the intersection of the two sets as final results.

1) Detection by RGB: The combination of HOG feature and SVM classifier is widely used in object detection tasks. HOG [11] is an excellent descriptor for capturing the edge direction or the distribution of local intensity gradients of objects. It has been applied successfully to detect the holistic body of humans. Meanwhile, compared with other features, HOG can be computed from a local grid cell of an image. Thus it is robust to both geometric and optical deformation of the image. HOG also has many other merits such as coarse airspace sampling, fine direction sampling, and strong local optical normalization. Therefore human subtle movements will not affect the detection based on HOG.

On the other hand, low computational cost is a necessary property for real time systems, therefore we need an accurate but effective binary classifier to identify pedestrians. Taking the advantage of HOG and the simplicity of SVM into consideration, we choose HOG and SVM as the RGB detector of our system. The detection procedure is quite straightforward. We first slide square windows with different sizes along the vertical and horizontal axes of the RGB image, generating a group of candidate windows. Then the HOG features of the windows are computed and used as the input of the trained SVM classifier. If the SVM predicts that a candidate window contains a pedestrian, then the window is added to the set of proposal windows. The set of proposal windows is given as the result of detection.

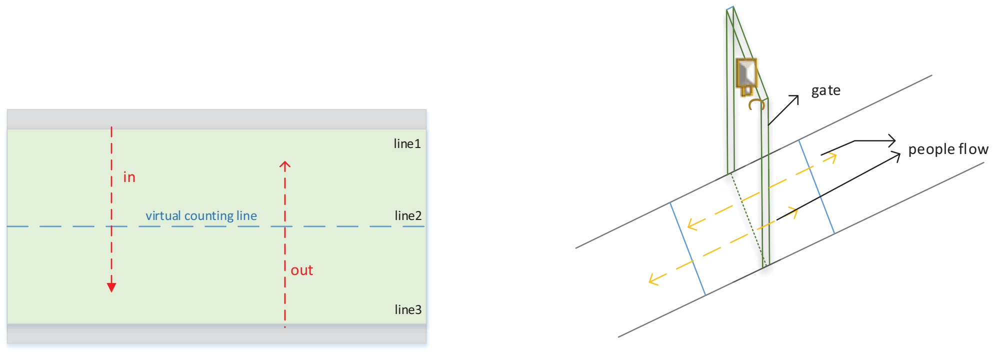
For training the SVM classifier, we collect about 10,000 overlooking images of human heads. Each image contains one or more people's heads, and we manually label the head area use bounding box like format of COCO dataset [17]. We also random sample the background and some other objects appearing in the image as the negative samples.

2) Detection by depth: As camera installation we described in section III-B, the TOF camera can overlook person's heads and shoulders from top view. Therefore, the depth image captured from TOF camera reflects the distance information of these parts to the lens. A naive method to detect pedestrians by depth image is setting a height threshold to filter out objects below the human head. However, such a threshold is sensitive to the height of pedestrians and camera installation. Pedestrians are of different heights and therefore we cannot set a fixed threshold for filtering. On the other hand, if the installation height of TOF camera changes, this threshold parameters need to be recalibrated. For robustness of our system, we propose a Deep Convex Convolution Filtering (DCCF) algorithm to process the raw depth image.

From the view of the camera, the head of a pedestrian is closer to the lens than the shoulder and other surroundings. Hence the pixel values of head region in the depth image will be relatively smaller than its surroundings. This region can be viewed as a convex region. Therefore we design a convolution kernel to extract this kind of convex feature.

Our approach is dividing the convolution kernel into two parts, i.e. the central part and the surrounding part. The convolution kernel $\boldsymbol{k}$ is shown in Fig.3. The central part is filled with negative values (red part) and the surrounding part is filled with positive values (yellow part), where the the size of central part is set as the size of a human head. From Fig.3, it is easy to see that this convolution operation is equivalent to weighted sum of surrounding pixel values minus weighted sum of the central pixel value.

Importantly, all values inside the convolution kernel follow two-dimensional Gaussian distribution along the center to surroundings, as shown in Eq. (1). Where $val(\boldsymbol{x})$ is a random variable in the kernel, $\boldsymbol{\Sigma}$ is the co-variance matrix and $\boldsymbol{\mu}$ is the mean vector.



(a) The monitoring area and virtual counting line (b) TOF camera installation

Fig. 2. System architecture

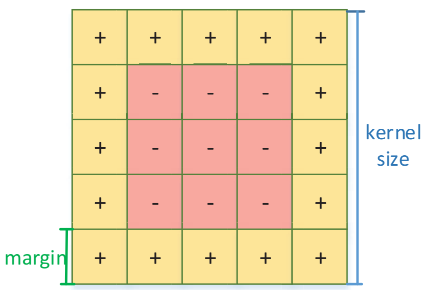


Fig. 3. kernel design

$$val(\boldsymbol{x}) \sim \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)}{(2\pi)\sqrt{|\boldsymbol{\Sigma}|}} \tag{1}$$

We set $\boldsymbol{\mu}$ as a zero vector and $\boldsymbol{\Sigma}$ is a diagonal two-dimensional matrix, like shown in Eq. (2), which means that we build up an isotropic and symmetrical convolution kernel without co-variance between x and y axis.

$$\mu = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix} \tag{2}$$

We only need to adjust one parameter, i.e. σ , to make the pixel value on the filter image be close to 0 after the convolution operation if there is a flat region on raw depth image. On the other hand, if there is a head region, the pixel value of head area is smaller than the surroundings. After the convolution operation, the value of this region in the filter image will become larger.

In this way, if we apply this convolution operation on whole depth image as Eq. (3), we can easily distinguish between the head and background in the depth filter image, as shown in Fig. 4(c), which is inverted the colors for clearness.

$$DFI(x,y) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} RDI(x+i-a_i, y+j-a_j)k(i,j) \tag{3}$$

In this equation, we denote the Raw Depth Image as RDI, where $RDI(x,y)$ is the depth value of the pixel whose coordinate is (x,y). Depth Filter Image is DFI. The size of kernel is $M \times N$, (a_i, a_j) is the relative position of the processing pixel (x,y) in the kernel.

D. Pedestrian Tracking

After the pedestrians are detected, their moving trajectories can be obtained by means of tracking algorithm.

We adopt Kernel Correlation Filter (KCF) [18] as base tracking algorithm in the proposed system. Because it is a linear algorithm with pretty low computational complexity, but could approximate the accuracy of non-linear kernel methods like Dual Correlation Filter (DCF) [18].

KCF is applied on the depth filter image instead of the RGB image or the raw depth image. Because there is a significant segmentation between the human head areas and the surroundings in depth filter image. Comparing the RGB image (Fig.4(a)) and the depth image (Fig.4(b)) with depth filter image (Fig.4(c)), we can see that the depth filter image is much more concise, therefore it is more suitable for pedestrian tracking.

Generally speaking, because tracking algorithms are much faster than detection algorithms, the speed of the system could be increased by only performing detection once every $n(n > 1)$ frames, and performing tracking on the rest $n-1$ frames. However, the tracking result will slowly deviate from the actual target without revision from detection, especially when n is large. Therefore we choose to perform detection on each frame due to the high efficiency of our proposed detection algorithm. It is to say that the detection and tracking are performed simultaneously.

In each frame the tracking algorithm generates a set of bounding boxes each of which contains a pedestrian, while the detection algorithm also reports such a set of bounding boxes. The system needs to combine the two sets and distinguish newcomers from the pedestrians that have already been tracked. At the same time, we also should remove the targets which have left the monitoring area. We show our adopted strategy in Algorithm 1.

Algorithm 1 Tracking and Detection Combination

Input:

Current Input Frame Image at time t, X^t ;

Tracking Bounding Box Set in time $t-1$, $S^{t-1} = KCF.update\left(X^{t-1}\right)$;

Tracker Set $S_{tracker}$;

Threshold, T;

Output:

Current Tracking Bounding Box Set, S^t;

Current Tracker Set $S_{tracker}$;

1: Detection Bounding Box Set, $S_d = Detctor\left(X^t\right)$;
2: **for** $box_t^j \in S^{t-1}$ **do**
3:   **if** The center of box_t^j is out of image **then**
4:     $S_{tracker} = S_{tracker} \setminus S_{tracker}[j]$
5:   **end if**
6: **end for**
7: **for** $box_d \in S^d$ **do**
8:   **if** $\frac{Area(box_d \bigcap box_t^j)}{Area(box_d)} \leq T \ \forall box_t^j \in S_t$ **then**
9:     $new\_tracker = KCF.Init\left(box_d\right)$
10:     $S_{tracker} = S_{tracker} \cap new\_tracker$
11:   **end if**
12: **end for**
13: $S^t = KCF.update\left(X^t\right)$
14: **return** $S^t, S_{tracker}$

The basic idea of Algorithm 1 is that we compute the overlap ratio between both sets of bounding boxes to find newcomers. Denote the bounding box reported by the detection algorithm as box_d and the bounding box generated by the tracking algorithm as box_t. The pedestrian bounding box can be recognized as a newcomer if the overlapping ratio is smaller than certain given threshold and a new tracker is assigned to it.

E. Pedestrian counting

When a tracked pedestrian leaves the monitoring area, we identify the moving direction of the pedestrian and renew the output of the system accordingly. Specifically, when the central point of the corresponding bounding box exceeds line1 or line3 in Fig. 2(a), the start point p_{start} and the end point p_{end} of its trajectory are fetched from the tracking algorithm, and then its tracking information is deleted (Line. 4 in Algorithm 1). Given a predefined constant l, the direction of the pedestrian is set as 1 if $p_{end} - p_{start} > l$ or -1 if $p_{end} - p_{start} < -l$. Then the pedestrian count of the corresponding direction is increased by 1 and output by the system.

IV. Experimental Results

At present, RGBD-PCS has been applied in public places of multiple cities successfully which has a strong robustness and high accuracy. Our system is equipped with powerful algorithms which can be able to meet all actual requirements. Firstly, this system can achieve **24fps** (actually 27 or 28) processing speed in embedded platforms such as ARM. Secondly, it can obtain a high accuracy, which are introduced in Table III and Table IV. Finally, it can adaptive to meet changes in the environment, which is introduced in Case Study.

Through collection and analysis of real world data, we conduct comparative experiments to verify effectiveness and efficiency of our proposed method. The datasets we used in experiments is captured by TOF camera in real world. The experimental setup and results are described in the following subsections.

A. Experimental Protocol and Dataset

The experiment consists of three parts. The first is detection experiment in static images. The second is pedestrian counting experiment in video streams. The last part is a study of challenging cases.

TABLE I
HOG-SVM Detection on RGB image with different Confidence

SVM Confidence	Reality	True	Error	Miss	Precision	Recall
1.0	115	129	60	0	0.683	1.000
1.5	115	110	32	5	0.775	0.957
2.0	115	107	25	8	0.811	0.930
2.5	115	102	11	13	0.903	0.887

We define a indicator function (Correct Detecting Index: CDI) to indicate whether our proposed method hits the target object or not.

$$CDI = \frac{Area(D) \bigcap Area(G)}{Area(G)} \tag{4}$$

$Area(D)$: area of detection bounding box, $Area(G)$: area of ground truth bounding box. We hold the view that detecting bounding box hits the target when $CDI > 0.5$. *Precision*, *Recall* and *F1-score* are used as evaluation metrics.

All of video datasets are recorded in the real world by our RGB-TOF camera. About 100 people pass the gate in each video dataset. We report the detailed results which allow readers to get the results of different image types, scenarios and camera installation heights (dataset 'outdoor1/2': collect videos outdoors using the camera with 2.0m installation height; dataset 'indoor1/2': collect videos indoors using the camera with 2.0m installation height; dataset 'high': capture videos with the 2.5m camera installation height;).

B. Detection Analysis

Detection is the first step in entire pipeline. The accuracy of detection is directly related to the final counting result. In this section, we validate the availability of detection model on

(a) RGB image (b) depth image (c) depth filter image

Fig. 4. Images with different type

TABLE II
DETECTING RESULT BASED ON COMBINED METHOD AND TWO SEPARATED METHODS

Methods	Height	Pedestrians	True	Error
Fusion Method	2.0m	119	104	1
	2.5m	115	112	6
Threshold in Raw Depth	2.0m	119	102	8
	2.5m	115	103	16
DCCF	2.0m	119	103	0
	2.5m	115	109	4
HOG-SVM	2.0m	119	107	8
	2.5m	115	111	23
Methods	**Miss**	**Precision**	**Recall**	***F1-score***
Fusion Method	15	0.990	0.874	0.929
	3	0.949	0.974	**0.961**
Threshold in Raw Depth	17	0.857	0.927	0.890
	12	0.895	0.865	0.879
DCCF	16	1.000	0.866	0.928
	6	0.965	0.948	0.956
HOG-SVM	12	0.930	0.899	0.915
	4	0.828	0.828	0.892

TABLE III
COMPREHENSIVE COUNTING RESULTS

Scenario	Direction	Precision		
		R+D	R	D
outdoor1	in	1.000	0.875	1.000
	out	0.967	0.903	0.917
outdoor2	in	0.952	0.952	1.000
	out	1.000	1.000	0.955
indoor1	in	0.980	0.980	1.000
	out	1.000	0.893	1.000
indoor2	in	1.000	0.962	1.000
	out	0.960	0.800	1.000
Scenario	**Direction**	**Recall**		
		R+D	R	D
outdoor1	in	1.000	0.933	1.000
	out	0.967	0.933	1.000
outdoor2	in	1.000	1.000	1.000
	out	1.000	0.905	1.000
indoor1	in	1.000	1.000	1.000
	out	1.000	1.000	1.000
indoor2	in	0.981	0.911	0.981
	out	1.000	1.000	1.000
Scenario	**Direction**	***F1-score***		
		R+D	R	D
outdoor1	in	**1.000**	0.903	1.000
	out	**1.000**	0.903	1.000
outdoor2	in	0.976	0.976	1.000
	out	**1.000**	0.950	0.977
indoor1	in	0.990	0.990	1.000
	out	**1.000**	0.943	1.000
indoor2	in	**0.990**	0.936	0.990
	out	0.980	0.889	1.000

static images instead of video streams to rule out the impact of subsequent processes.

Firstly, we conduct explore experiment about HOG-SVM detection method on static RGB image. Table. I manifests that the detection based on RGB image is sensitive to the parameter (SVM confidence). It means that we should balance the precision and recall in RGB detection.

Next, we compare experimental results of our proposed fusion detection method with single HOG-SVM method, single DCCF method and naive threshold method in raw depth image. We also compare results on different camera installation height (2.0m and 2.5m) to explore the height influence on detection.

From the Table II (bold font indicates best), we can see that fusion detection method achieves the best performance in *F1-score*. Our proposed DCCF also gets a remarkable result as well. On the other hand, HOG-SVM which is only conducted on the RGB image cannot be satisfying. Naive threshold method on raw depth image tend to detect many non-head objects and miss some true targets so its *F1-score* is relative lower. In addition, raising the height of RGB-TOF camera could improve detection accuracy because the camera gets a wider field of view, which is conducive to distinguishing between the background and detection targets.

TABLE IV
COUNTING RESULTS AGGREGATED FROM TABLE III

Scenario	Precision		
	R+D	R	D
high	1.000	0.978	0.978
outdoor1	0.983	0.889	0.958
outdoor2	0.976	0.976	0.976
Scenario	**Recall**		
	R+D	R	D
high	1.000	0.935	1.000
outdoor1	0.983	0.933	1.000
outdoor2	1.000	0.952	1.000
Scenario	***F1-score***		
	R+D	R	D
high	**1.000**	0.956	0.989
outdoor1	**0.983**	0.911	0.978
outdoor2	**0.988**	0.963	**0.988**

C. Counting Analysis

In this section, we evaluate the effectiveness of RGBD-PCS in video streams datasets. Table III provides a view of the results aggregated over the type of scenario and used images.

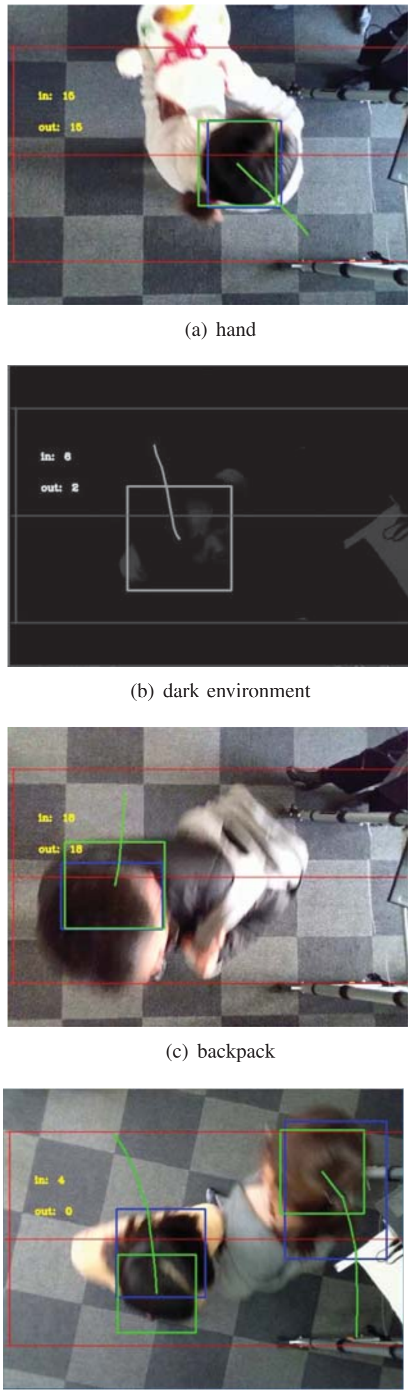
From Table III we can notice the following phenomenons: First, it reveals the high validity of DCCF. We can get a satisfying result only based on depth image (D). Second, the result only based on RGB image (R) is not desirable. Third, the fusion algorithm (R+D) can also get a good performance similar with D, which shows the effectiveness of the fusion algorithm. In a word, the performance of R+D is much better than RGB and similar with D. Moreover, we have to take into account the failure situation of the depth image under strong light, and in this situation, usable information from RGB image is the unique choice. Thus, we cannot remove the RGB image in our algorithm.

We take the camera installation height into account in Table IV. It is the result of outdoor datasets aggregated from Table III. In the same outdoor environment, we raised the height of RGB-TOF camera from 2.0m to 2.5m to observe the influence of camera height on counting. The highest value of *F1-score* (bold number) illustrates the validity of increasing camera installation height. The reason is that the effective detection distance of TOF camera is limited (larger than 40cm at least), thus the depth information would be distorted when the targets are too close to the TOF lens. Therefore a simple but effective solution is raising camera installation height so that we can get an expanding detecting scope.

Finally, we can make a conclusion that the fusion algorithm (R+D) can achieve a higher and more robust results comparing to each single method (R and D), and increasing camera installation height is very useful.

D. Case Study

Obviously, RGBD-PCS can operate normally under usual environment, but the following special cases should also be taken into account.



(a) hand

(b) dark environment

(c) backpack

(d) multiple pedestrians

Fig. 5. Some hard cases in pedestrians counting. The blue rectangles are the detected bounding boxes and the green rectangles are the tracking boxes. Green lines are the tracking trajectories

1) Backpack, Luggage, Hat and so on: Some objects may also have a similar relative raised region like the head. We may mistakenly identify these objects as head if we only use depth image for detection. Thus we need to combine the information of RGB image at this time, and filter out undetected regions in RGB image to achieve better performance. Our system could solve this problem most of the time. Some cases are shown in the Fig.5(a) and Fig.5(c).

2) Darkness: In dark environment, it is obvious that RGB image will lose most information, but depth image can keep valid. Such scenarios require our proposed system to work under weak lighting conditions. Our system can automatically switch to dark mode (running only based on depth image) according to the brightness in the RGB image. Like cases in Fig.5(b), the green bounding box and trajectory show that our method can work in dark environment very well.

3) Multiple pedestrians counting: Most of the time, there are many people appearing in the monitoring area at the same time. We should detect all pedestrians in the frames and track them until they leave. Fig. 5(d) illustrates that two people walk through the monitoring area from different directions. Our system can identify both of them and track their routes successfully.

V. Conclusion

This paper proposes an accurate pedestrian counting method and design a complete system based on RGB-TOF camera. We propose a combination method using RGB and depth image. In process of depth image, we propose DCCF and use it to detect the target objects. The validity of proposed method is verified by experiments in multiple scenarios. In fact, the entire system has been used in the production environment and it works well. Our research step of studying is to train a more powerful classifier, develop a more efficient fusion algorithm and make more accurate analysis of sophisticated target trajectories.

References

[1] K. Hashimoto, M. Yoshinomoto, S. Matsueda, K. Morinaka, and N. Yoshiike, "Development of people-counting system with human-information sensor using multi-element pyroelectric infrared array detector," *Sensors Actuators A Physical*, vol. 58, no. 2, pp. 165–171, 1997.

[2] Y. Cong, H. Gong, S. C. Zhu, and Y. Tang, "Flow mosaicking: Real-time pedestrian counting without scene-specific learning," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1093–1100, IEEE, 2009.

[3] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[4] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 935–942, 2009.

[5] P. Sermanet and Y. Lecun, "Traffic sign recognition with multi-scale convolutional networks," in *International Joint Conference on Neural Networks*, pp. 2809–2813, 2011.

[6] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Computer Vision and Pattern Recognition*, pp. 589–597, 2016.

[7] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September*, pp. 1045–1048, 2010.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *International Conference on Neural Information Processing Systems*, pp. 91–99, 2015.

[9] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos, "Estimating pedestrian counts in groups," *Computer Vision Image Understanding*, vol. 110, no. 1, pp. 43–59, 2015.

[10] X. Wang, "An hog-lbp human detector with partial occlusion handling," *Proc.ieee Int.conf.on Computer Vision Kyoto Japan Sept*, vol. 30, no. 2, pp. 32–39, 2009.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

[12] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *International Conference on Pattern Recognition*, pp. 1–4, 2009.

[13] A. Bevilacqua, L. D. Stefano, and P. Azzari, "People tracking using a time-of-flight depth sensor," in *IEEE International Conference on Video and Signal Based Surveillance*, p. 89, 2006.

[14] C. A. Luna, C. Losada-Gutierrez, D. Fuentes-Jimenez, A. Fernandez-Rincon, M. Mazo, and J. Macias-Guarasa, "Robust people detection using depth information from an overhead time-of-flight camera," *Expert Systems with Applications*, vol. 71, pp. 240–256, 2017.

[15] Y. H. Chou, I. K. Lim, J. S. Shim, S. I. Chung, and S. M. Kwon, "People counter using tof camera and counting method thereof," Nov. 14 2017. US Patent 9,818,026.

[16] L. Del Pizzo, P. Foggia, A. Greco, G. Percannella, and M. Vento, "Counting people by rgb or depth overhead cameras," *Pattern Recognition Letters*, vol. 81, pp. 41–50, 2016.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[18] J. F. Henriques, C. Rui, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.