

REAL-TIME ACCURATE CROWD COUNTING BASED ON RGB-D INFORMATION

Huiyuan Fu, Huadong Ma, Hongtian Xiao

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing University of Posts and Telecommunications, Beijing 100876, China
fu_hui_yuan@yahoo.com.cn, mhd@bupt.edu.cn, buptxht@gmail.com

ABSTRACT

Real-time accurate crowd counting is one of important tasks in intelligent visual surveillance systems. Most previous works can only count passing people robustly without heavy occlusions which are very common in the practical surveillance scenes. To solve this difficult problem, we propose a new method for crowd counting for RGB-D (RGB plus depth) data using a commodity depth camera. In our method, we first detect each head-shoulder of the passing or still person in the surveillance region with fast template matching based on depth information including pedestrian filling with convex hull segmentation. Then, we track and count each detected head-shoulder based on RGB information bidirectionally. By using this approach, we have built a practical system for robust and fast crowd counting. Extensive experimental results show that our method achieves significant improvement comparing to states-of-the-art approach, and the built system is not only robust to heavy occlusions, but also can be deployed in the real time crowd counting application scenes.

Index Terms— Surveillance, crowd counting, people counting, RGB-D data, real time system

1. INTRODUCTION

Reliable and real-time crowd counting is a critical task in intelligent visual surveillance. It is the foundation for so many typical vision based applications. For example, knowing the crowd density in public places can help managers identify unsafe situations and regulate traffic appropriately. As another example, the public museum can control the number of people entering according to the real-time people flow information. Although it is so important, robust crowd counting in complex environments is still far from being completely solved.

Recently, the computer vision based methods are widely used to crowd counting or people counting [1, 2, 3, 4, 5].

The research reported in this paper was supported in part by the National 973 Project of China under Grant No. 2011CB302701 the Natural Science Foundation of China under Grant No. 60833009, No.60903072, and No. 61003280; the National Science Fund for Distinguished Young Scholars under Grant No. 60925010; and the Program for Changjiang Scholars and Innovative Research Team in University under Grant No.IRT1049.



Fig. 1. Typical scenes with heavy occlusions we addressed for crowd counting in this paper. Current methods are not very robust to these complex scenes.

Crowd counting can be also seen as people counting in the complex scenes with heavy occlusions (see Fig. 1). The authors of [1] proposed a method for estimating the crowd, but they can not accurately count the number of current flow. The authors of [2, 3, 4, 5] can count people well in the simple environments. However, they will inevitably be affected by the unpredictable complicated environments with heavy occlusions such as in Fig. 1. In this paper, we propose a new scheme to overcome this difficult but very important problem for crowd counting. We implement our approach mainly based on depth information from a cheap depth camera (such as Kinect). To our knowledge, it is rare to use depth data for counting people by co-processing the captured real-time stream of passing people both in color and depth format. In our presented approach, we first remove the background for better crowd counting. Then we use a normalized correlation based template matching method to help for finding the pedestrian locations. Third, we propose a new scheme for choosing the convex hull segmentation point for getting each independent pedestrian. Finally, we can detect the head-shoulder for crowd counting bidirectionally in real time. Moreover, our method is not only for detecting and counting the moving person, but also for still person. The experiment results show the effectiveness of our approach.

The remainder of this paper is organized as follows: In Section 2, the proposed method is presented in detail. Experimental results and analysis are described in Section 3. Finally, Section 4 concludes the work.

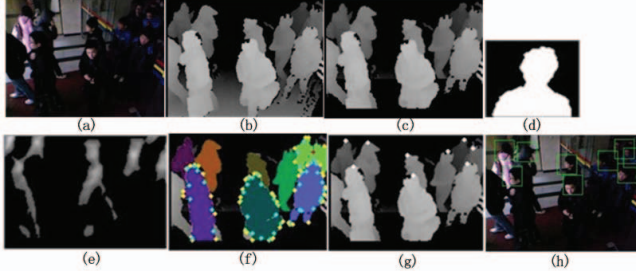


Fig. 2. Illumination of our algorithm. (a) origin color image. (b) origin depth image. (c) background removal result. (d) model for matching. (e) matching result. (f) pedestrian filling and convex hull detection. (g) top head point detection. (h) final head-shoulder detection result.

2. OUR ALGORITHM FOR CROWD COUNTING

In this section, we introduce our algorithm in detail in the following four fields: background removal, template matching, pedestrian filling with convex hull segmentation and head-shoulder based detection for final crowd counting. The summary of our algorithm is listed at the end of the section.

2.1. Background Removal

In order to remove the effect of floor for pedestrian segmentation and speed up the template matching in the following phase, we need to remove the background first in our algorithm. We traverse the depth data of pixel in each row to get the maximum value H and minimum value L . The final depth value of each pixel can be computed as follows:

$$depth_{new} = \begin{cases} depth_{old} & \text{if } depth_{old} > L + a(H - L) \\ 0 & \text{else} \end{cases} \quad (1)$$

where a represents for the scale factor in a specific scene. Experientially, we set a to 105 in our experiment. In this way, we can remove most of background effectively (especially for the floor, as seen in Fig. 2(c)).

2.2. Normalized Correlation Based Template Matching

In our algorithm, we use the correlation based template matching method to locate the best fit point. The details of the choice for template model (see Fig. 2(d)) will be stated in the experiment section (see Fig. 5). Motivated by [6], we employ the normalized scheme in our template matching for reducing the effect of the light changes between the template and image. The method is as follows:

$$R(x, y) = \frac{\sum_{x', y'} [(x', y') \cdot I(x + x', y + y')]^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (2)$$

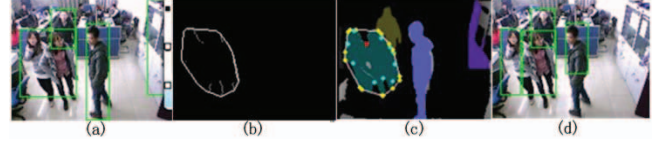


Fig. 3. Segmentation point chosen condition for convex hull segmentation. (a) original segmentation result from filling image. (b) minimum convex polygon detection result. (c) segmentation point detection result. (d) final result.

where I means the input image, T means the template and R means the result. $T(x', y')$ and $I(x + x', y + y')$ are represented as follows:

$$T(x', y') = T(x', y') - \frac{1}{(w, h) \cdot \sum_{x'', y''} T(x'', y'')} \quad (3)$$

$$I(x + x', y + y') = I(x + x', y + y') - \frac{1}{(w, h) \cdot \sum_{x'', y''} I(x + x'', y + y'')} \quad (4)$$

where w, h means the weight and height of the input image. The matching result can be seen in Fig. 2(e).

2.3. Pedestrian Filling with Convex Hull Segmentation

When we obtain the best fit point in each pedestrian, we fill the pedestrian in depth image. Our scheme is filling the adjacent point when its depth is less than a beforehand threshold α until the condition is not fit.

When the occlusion is heavy, the closed people will be recognized as only one person (see Fig. 3(a)). To overcome this problem, we bring the convex hull segmentation method in our algorithm. As seen in Fig. 3(b), we can easily obtain the minimum convex polygon. Then, as seen in Fig. 3(c), we find out the start point and end point (the yellow labeled) and the depth point (the blue labeled). To get the convex hull segmentation point, we propose a new scheme as follows:

$$condition : \begin{cases} |start.x - \min x| < \beta \text{ or } |end.x - \min x| < \beta \\ |start.x - end.x| > \eta \\ strat.y < depth.p.y \text{ or } end.y < depth.p.y \\ |strat.y - end.y / strat.x - end.x| < \varepsilon \end{cases} \quad (5)$$

where $depth.p$ are the candidate points for segmentation. $\min x$ means the minimum value for x of the enclosing rectangle. β, η , and ε are thresholds. In this way, we can obtain the final red point (see Fig. 3(c)) for segmentation, and the final segmentation result can be seen in Fig. 3(d).

2.4. Head-Shoulder Detection, Tracking and Counting

As seen in Fig. 2(f), we have fill each pedestrian. We can find out the top head point easily, such as in Fig. 2(g) (the white point). Then, we can detect the head-shoulder as in Fig. 2(h) by setting a square with 20×20 scale (our experiment

Algorithm 1 The proposed algorithm for crowd counting

Initialization: $In = 0, Out = 0$

-Remove the background according to Eq. (1).

-Locate the best fit point using the template matching method between Eq. (2) and Eq. (4).

-Fill each pedestrian in colour and select the fit convex hull segmentation point according to Eq. (5).

-Detect, track and count the crowded people bidirectionally based on above steps.

Result Output: In, Out

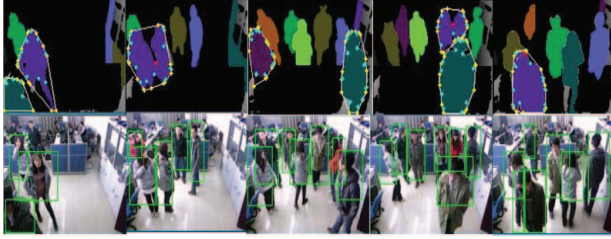


Fig. 4. Results of pedestrian filling with convex hull segmentation according to our proposed condition in Eq. (5).

shows this scale is fit for most scenes). If we can detect each pedestrian right in the crowded scene, the following tracking and counting is an easy task. Because of the space limit, we don't introduce in detail here. In the experiment, we will give our final results.

3. EXPERIMENTAL RESULTS

In this paper, we propose a new way for crowd counting using the RGB-D information. In order to demonstrate that our presented scheme is effective, we test our method in extensive different fields in the following place.

First, we evaluate the performance of convex hull segmentation and template choice for the final result in our proposed algorithm. The threshold value α is set to 4 in our experiment. Meanwhile, we set β , η , and ε in our experiment to 3, 15 and 0.8, respectively. As seen in Fig. 4, we can find that the convex hull segmentation results are right all time.

Second, we show the choice of our matching model in detail. We adopt the statistical strategy in our method. We synthetically consider the shape and scale of the model. We have tested the models with 10 different shapes and 5 different scales (In Fig. 5, we show the typical models). In our experiment, we find that the scale will slightly affect the performance. In Fig. 5, we show 4 models with different scales: 10×10 , 14×14 , 20×20 , and 30×30 . We find that 20×20 is the best (with red rectangle). Then, we compare the models with different shapes (see Fig. 5) in our algorithm. The visual illustration (top of Fig. 5) and running time table (bottom of Fig. 5) both demonstrate that they are very similar for the performance of the final crowd counting in our experiment.

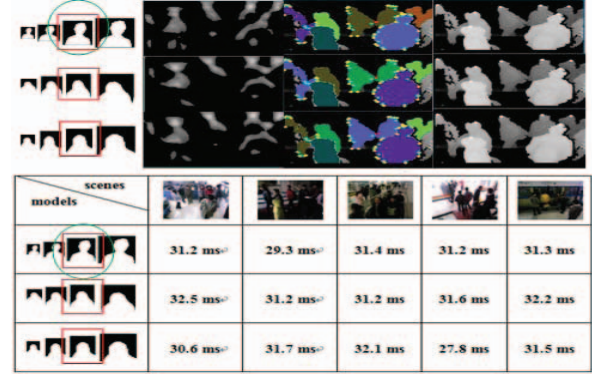


Fig. 5. Statistical comparison on different models with visual results and mean processing speed for crowd counting. We show three typical models here. Considering the results are similar and processing speed reduction with multiple models meantime, we only use the first model in our experiment.

So we choose the model with green ring in our experiment.

Then, we demonstrate our presented method by comparing it to well-known states-of-the-art algorithm [7, 8]. Part based model method [7] is one of the best algorithms for pedestrian detection. Skeleton tracking method [8] which based on [9] is effective in the general environments. Considering the space, we only compare the approaches in typical office scene and building corridor scene. Note that we focus on the comparison of counting accuracy and counting speed directly between the different approaches. More colloquially, we don't care the specific technologies used in these approaches. As seen in Fig. 6, our algorithm is more robust than [7] and [8] for crowd detection. In the following, we compare them both on accuracy and speed of crowd counting in these two scenes respectively. As seen in Fig. 7, our approach and skeleton tracking based method [8] has the close result when the crowd is not very heavy. They are both better than skeleton tracking method [8]. But when the crowd is heavy, our approach is more effective. Then, we evaluate the processing speed for these three algorithms. As seen in Fig. 8, we find that our method and the method of [8] both run fast than the method of [7].

Finally, we estimate our algorithm in different complex scenes in our daily life. We select five typical scenes: Office, Subway Entrance, Escalator Export, Building Export and Building Corridor. These scenes are common in our life, and their need for intelligent surveillance is urgent. Based on our presented method, we have developed a practical system facing the crowd scenes such as these five scenes. The sample results can be seen in Fig. 9. Table 1 shows the accuracy and speed of our algorithm in these five different scenes. In this paper, we focus on monitoring the close range but crowded regions (such as indoor scenes, escalator export, etc). So although the sensing range of depth camera is restrained, it is enough for our surveillance purpose.

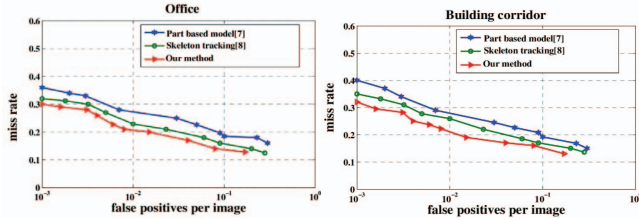


Fig. 6. Comparison on crowd detection precision with two state-of-art algorithms in office scene (left) and building corridor scene (right).

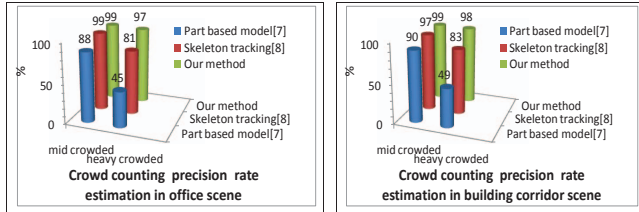


Fig. 7. Comparison on crowd counting precision with two state-of-art algorithms in office scene (left) and building corridor scene (right).

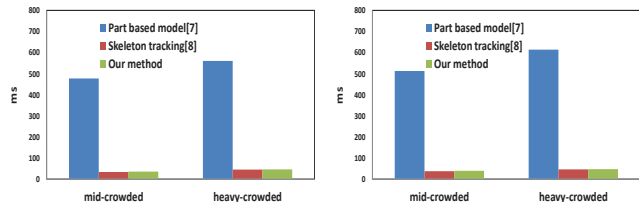


Fig. 8. Comparison on crowd counting speed with two state-of-art algorithms in office scene (left) and building corridor scene (right).

4. CONCLUSIONS

In this paper, we propose a new scheme using the RGB-D information for crowd counting. We evaluate our algorithm comparing to the states-of-the-art approach, and the experimental results are convincing. Based on this approach, we have built a practical system for crowd counting. The obtained results demonstrate the effectiveness of our approach.

5. REFERENCES

- [1] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos, "Estimating pedestrian counts in groups," *Computer Vision and Image Understanding*, vol. 110, pp. 43–59, 2008.
- [2] H. D. Ma, C. B. Zeng, and C. X. Ling, "A reliable people counting system via multiple cameras," *ACM Transactions on Intelligent Systems and Technology*, vol. 3, pp. 1–22, 2012.

Table 1. Test for accuracy and speed of our algorithm for crowd counting

		# IN	# OUT		# SPEED
Video 1 (office)	Truth	410	370	Head-shoulder detection	26.2 FPS
	Result	398	359	Total algorithm	21.5 FPS
	Accuracy	97.1%	97.0%		
Video 2 (subway entrance)	Truth	360	469	Head-shoulder detection	24.3 FPS
	Result	351	458	Total algorithm	20.2 FPS
	Accuracy	97.5%	97.7%		
Video 3 (escalaor export)	Truth	950	1034	Head-shoulder detection	23.5 FPS
	Result	932	989	Total algorithm	19.8 FPS
	Accuracy	98.1%	95.6%		
Video 4 (building export)	Truth	1342	875	Head-shoulder detection	25.5 FPS
	Result	1306	861	Total algorithm	21.2 FPS
	Accuracy	97.3%	98.4%		
Video 5 (building corridor)	Truth	856	834	Head-shoulder detection	23.7 FPS
	Result	835	819	Total algorithm	20.9 FPS
	Accuracy	97.5%	98.2%		



Fig. 9. Sample results for crowd counting in different scenes. From left to right: building corridor, building export, escalator export, office and subway entrance.

- [3] B. Antic, D. Letic, D. Culibrk, and V. Crnojevic, "K-means based segmentation for real-time zenithal people counting," *IEEE International Conference on Image Processing*, pp. 2565–2568, 2009.
- [4] Y. Cong, H. F. Gong, S. C. Zhu, and Y. D. Tang, "Flow mosaicking: real-time pedestrian counting without scene-specific learning," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1093–1100, 2009.
- [5] L. Snidaro, C. Micheloni, and C. Chiavedale, "Video security for ambient intelligence," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 1, pp. 133–144, 2005.
- [6] G. Bradski and A. Kaebler, "Learning opencv-computer vision with the opencv library," *O'REILLY press*, 2009.
- [7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2010.
- [8] Microsoft, "Demo of skeleton tracking based method for people tracking," <http://www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx>, 2011.
- [9] J. Shotton, A. Fitzgibbon, and etc M. Cook, "Real-time human pose recognition in parts from single depth images," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1297–1304, 2011.