Journal of Industrial Information Technology and Application Vol. 2 No. 3

ISSN(Online) :2586-0852

Real-time People Counting System Using Simplified Motion Detection

Sung In Cho [1,*], Suk-Ju Kang [2]

[1] Department of Electronic Engineering, Daegu University, South Korea.
[2] Department of Engineering, Sogang University, Seoul, South Korea.

Abstract. This paper proposes a real-time system for estimating the count of pedestrians. The proposed method consists of two main procedures: estimation of binary motion and motion flow rate on a line of interest (LOI). By modeling a background, the motion generated by a human movement can be simply extracted. Then, the motion flow which is the number of motion pixels crossing the LOI is calculated. Finally, the number of people crossing the LOI is generated by normalizing the motion flow. Simulations show that the proposed method successfully estimates the number people crossing LOI with simple operations.

Keywords; component; formatting; style; styling; insert (key words)

1. Introduction

The goal of people counting is to estimate the number of people in a specific area. The technology for people counting [1]-[3] is widely used in various applications including surveillance, urban planning, and store management. Specifically, if the number of people entering and exiting store is available, the store-specific management is possible. In addition, it is possible to analyze the commercial power of a specific area if the results of people counting for each store in that area are available. In this paper, we propose a real-time people counting system which is based on a motion flow rate estimation

[*] Corresponding author: [1] 524033109@qq.com [2] yuzhou0229@gmail.com [3] wcf@ttu.edu.tw
Received: 2017.10.2; Accepted: 2017.12.6; Published: 2018.9.30

2. Proposed Method

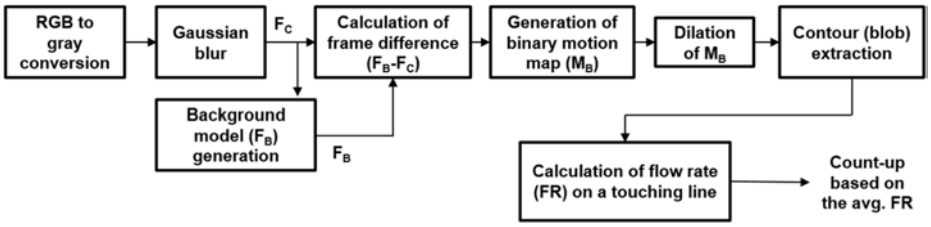

Fig. 1 An overall procedure of the proposed system

Fig. 1 shows the overall procedure of the proposed system. Firstly, RGB to gray conversion and Gaussian blur is performed on a given RGB image. The purpose of a Gaussian blur is to suppress of image noise and to use the high level features. Then, the background model (F_B) is generated by averaging the previous F_B and the current resultant image (F_C) of a Gaussian blur. After F_B generation, a motion is extracted from the absolute difference between F_B and F_C by assuming that the position of camera is fixed. From the extracted motion, a binary motion (M_B) shown in Fig. 2 (b) is calculated by thresholding the calculated frame difference. For preventing the motion splitting for each person, the dilation on M_B is conducted. The contour of each MB is extracted. Black bounding box in Fig. 2 (c) shows the extracted contour. For estimating the number of people crossing a certain region, we set the line of interest (LOI). LOI denoted by the highlighted blue line in Fig.2 (c) can be customized depending on the geometric characteristics of the target region and the camera position. The number of M_B pixels (motion flow) crossing this LOI is calculated. Finally, the flow rate (FR) is calculated by the spatio-temporal normalization of the motion flow as follows:

$$FR = \frac{\#\,of\ M_B}{N_{nor}} \tag{1}$$

where N_{nor} is the normalizing factor. N_{nor} is determined by the distance between the target area and the camera. In addition to this, the average moving velocity can be considered to determine N_{nor}. Instantaneous and accumulated FRs are demonstrated in Fig. 2 (c).

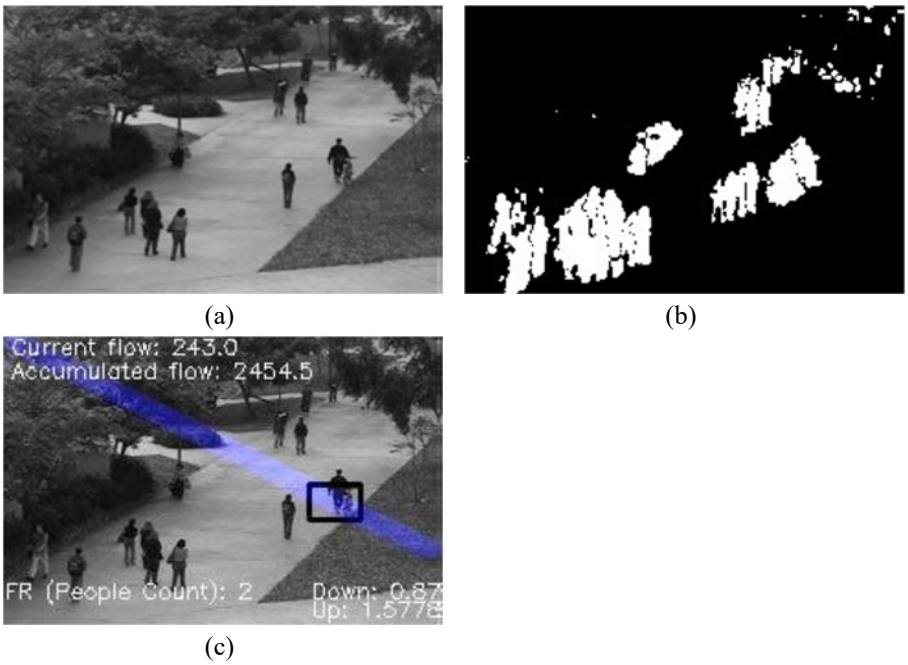

Fig. 2 (a) Input image (b) Binary motion, M_B (c) Result of blob extraction

3. Simulation Results

For the simulations, we used UCSD people counting dataset [4] as the test image sequence. Fig. 2 (a) shows the example of UCSD dataset. UCSD dataset contains 41 kinds of sequence sets and each set consists of 200 frames. The resolution of UCSD dataset is 238×158 pixels. The proposed method was implemented by using Raspberry pi 3 B. For the development of a software, Python 2 with openCV 3.2 was used. The normalizing factor used in equation (1) and the frame interval were set to 800 and 8, respectively. Table 1 shows the accuracy of people counting by the proposed system. The absolute error in table 1 denotes the absolute difference between the ground-truth and calculated numbers of people crossing LOI at the end of each image sequence.

TABLE I. ACCURACY OF PEOPLE COUNTING BY THE PROPOSED SYSTEM

Test Sequence	vidf1_33_000	vidf1_33_027	vidf2_33_011
Absolute Error	0	1	1
Recall	1	0.875	0.8
Precision	1	1	1

As shown in Table 1, the proposed system provides very accurate results of people counting. To verify the possibility of the real-time people counting, we used the Raspberry pi camera module v2 with 640×480 pixels resolution and 16 frame per second (fps). Under this environments, the propose system successfully estimated the number of people crossing LOI

4. Conclusions

This paper proposed a real-time people counting system that utilizes motion and its flow rate estimation. In the proposed method, the motion information is calculated by the difference between the current input frame and the background model. The motion flow which is the number of the motion pixels crossing line of interest (LOI) is calculated. Final people count can be extracted by the motion flow rate which is calculated by the motion flow normalization. In the simulation results, the proposed system showed the high recall and precision values. In addition, we verified that the proposed system can operate in real-time using Raspberry pi 3 B.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; Ministry of Science, ICT & Future Planning) (No. 2017R1C1B5075091) and the Ministry of Trade, Industry and Energy(MOTIE) and the Korea Institute for the Advancement of Technology(KIAT) (N0002436, 2017).

References

[1] A. B. Chan and N. Vasconcelos, “Counting people with low-level features and Bayesian regression,” IEEE Trans. Image Process., vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[2] D. Kong, D. Gray, and H. Tao, “A viewpoint invariant approach for crowd counting,” in Proc. 18th Int. Conf. Pattern Recognit. (ICPR), 2006, pp. 1187–1190.

[3] Ma, Z., & Chan, A. B.. “Counting people crossing a line using integer programming and local features,” IEEE Trans. Circuits Syst. Video Technol., vol. 26 no. 10, pp. 1955-1969, Oct. 2016

[4] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, “Privacy preserving crowd monitoring: Counting people without people models or tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2008, pp. 1 –7.