Automatic passenger counting system for bus based on RGB-D video

Feng Li, Fuwei Yang, Haiwei Liang and Wenming Yang[†]
Shenzhen Key Lab. of Information Sci&Tech/Shenzhen Engineering Lab. of IS&DCP
Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua University, China
[†]corresponding author, E-mail: yangelwm@163.com

Counting the number of passengers getting in/out of a bus in each station is a challenging task in smart bus field. To solve the problem, this paper presents a method of automatic people counting for a bus based on a RGB-D video, a zenithal camera in the bus door capturing the passengers flow is set, and the camera can get depth Image and RGB image simultaneously, we proposed a method combining RGB image and depth image to detect the head of the passengers, and then a novel tracking strategy is proposed. The test results show that this method can accurately extract the head area in the video and can accurately count the number of people getting in/out of a bus at a fast speed. The test data in different time periods show that the method has strong robustness and stability.

Keywords: depth image; head detection; tracking; counting;

1. Introduction

With the development of artificial intelligence and Internet, the public have an increasing demand for smart bus, as a part of smart city. In a smart bus system, real-time bus passenger flow information is very important: on the one hand, passengers can optimization their route for going out with the information; Bus company, on the other hand, can be flexible at bus scheduling. Besides, with large amounts of traffic data, bus company can optimize their bus lines arrangement. Counting the number of people getting in/out of each bus in every station is important, however, there is no mature method to get the data. Nowadays, video image processing technology is used for solving the problem with the help of a video installed in the ceiling of the bus door. However, the video data is strongly influenced by light and weather due to the motion of the vehicle, therefore, it’s still a challenge for accurate counting.

There are mainly three counting methods: 1) infrared-based method, 2) pressure-sensing-based methods, 3) method based on computer vision.

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

Passenger flow counting method based on infrared is with low accuracy in case of occlusion and crowding, and the pressure-sensing-based system needs complex installation and the equipment is easy to be damaged. Passenger counting system based on computer vision is now a hot research. It is divided into two categories: 1) people counting based on monocular camera [7]. 2) people counting based on stereo vision [8]. Methods based on monocular vision use two-dimensional data acquisition for target detection, tracking and counting. Some scholars use two monocular cameras to calculate the depth information [1,6,9], and now, with the depth of the popularity of depth camera, we can obtain depth information directly through the depth camera, thus, methods using depth camera directly are proposed in recent years.

With the development of the depth camera, it is possible to obtain low-cost depth image. Using depth camera make it possible for us to make use of shape information of pedestrian without much calculation. Besides, it's relatively insensitive to the illumination and weather. Therefore, this paper selects the depth image obtained by the depth camera to design our algorithm. However, the depth image obtained directly by depth camera exists much noise, and the range of measurement is also limited. Thus, we use RGB image to make up the imperfection of depth image.

Thus, the basic scheme of the paper is: On the basis of depth image, combine with RGB image, using image segmentation and pattern recognition method, to do target detection. In this paper, target detection, as with most of the scholars, means head extraction, because the camera is set zenithal, looking down at the passengers, whether crowded case or not, the head is stable and detectable. Finally, we track and count with the result of detection. In the part of tracking, our feature-based tracking, combined with mean-shift, have a good effect.

The flow chart of the method can be summarized as figure 1.1:

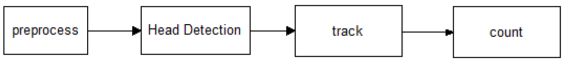

Fig. 1.1 System flow chart

In this paper, the selected camera is Microsoft Kinect, which is a cheap depth camera, and can obtain depth image and RGB image simultaneously, most importantly, the coordinate of the two images are aligned(see in figure 1.2). Depth image and true color images obtained have a resolution of 240 * 320. The data of the paper is collected in rush hour in a busy bus line in ShenZhen, China, the frame rate of 10 frames per second.

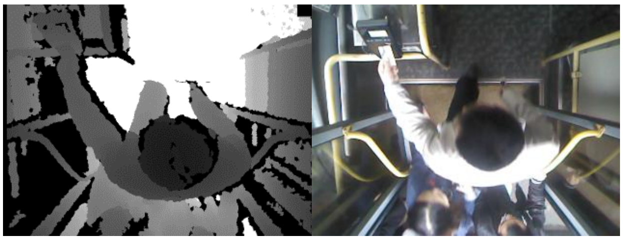

Fig. 1.2 Depth image (left) and RGB image (right) captured by kinect, the coordinate of the two images are aligned

The rest of the paper is organized as follows: Section 2 introduces our way for head detection, combining depth image and RGB image. Section 3 introduces our novel way for tacking and counting. our feature-based tracking, combined with mean-shift, have a good effect. Section 4 presents the results of detection and counting. The final section 5 concludes the paper and discusses the future work.

2. Head Detection

As we can see from figure 2.1, the depth image obtained by Kinect, there are some defects in depth image, thus, we need preprocessing. After preprocessing, the moving target is extracted from the whole image, and then the head area is extracted accurately by the way of machine learning according to the shape and color of the head. Target detection flowchart is shown in Figure 2.

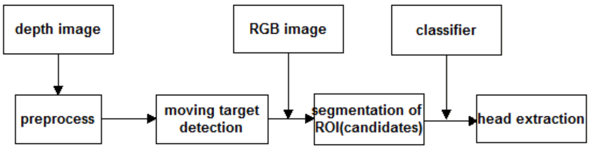

Fig. 2.1 The flow diagram of detection

2.1. *Preprocess*

As we can see from the depth image, in the original image, high pixel value means the place is very low, high pixel vale correspond to low places. However, due to the image-forming principle of Kinect, there are a lot of “zero points”. these zero points’ s pixel value are zero, but it doesn’t mean that this place is very high. Thus, zero points have negative effects on shape recognizing. A median filter is used to wipe out some small blocks of zero points. However, the rest of the zero points, we can’t wipe them out, it is necessary to record their coordinates for subsequent processing. For the convenience of processing and viewing, for each pixel, we have

$$p_{new} = \begin{cases} 255 - p_{old}, & p_{old} \neq 0 \\ p_{old}, & p_{old} = 0 \end{cases} (1)$$

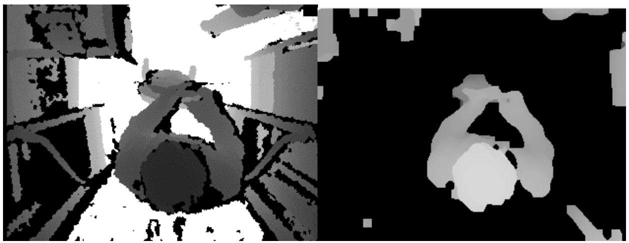

Fig. 2.2Beforepreprocessing(left), after preprocessing(right)

2.2. *Foreground extraction*

There are mainly three ways for voving foreground extraction: 1) optical flow;2) frame-to-frame difference;3) background modeling method. The optical flow method is complex and unfavorable for real-time processing. The frame-to-frame difference method is more dependent on the speed of passengers getting on and off. Both have their limitations, and we select background modeling method because the installed scene is fixed.

In this paper, a Hybrid Gaussian model with high accuracy and robustness is used to model the background. The interference can be eliminated by morphological processing. Finally, most of the non-foreground area can be eliminated by area filtering. To do area filtering, connected component detection is down, after which the area of each connected component is calculated, if the area is smaller than a thresh, the pixels of the connected component are all set to zero. In this way, the extracted foreground contains mainly the passengers themselves, with the majority of the heads, arms and shoulders. The effects of the step can be seen in figure 2.3.

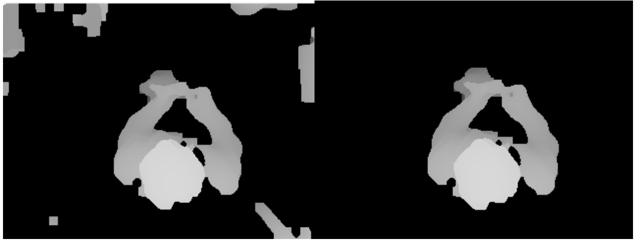

Fig.2.3 Before area filtering (left)and after area filtering (right)

2.3. *Candidate region segmentation*

After capturing the foreground image, locate the potential head region in the foreground image. In the depth map, we can make use of the shape of head: the head is obviously a middle high and low side shape. Using this feature, we can locate the area of the head as follows :

i) Divide the image into several patches equally (see in figure 2.4).

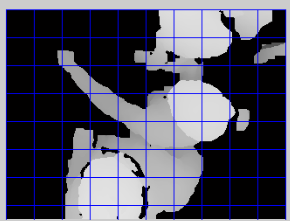

Fig.2.4Awhole image is divided into several patches equally

ii) calculate the average height of each patch, we have

$$Aver_{m,n} = \frac{\sum_{i,j} Img_{m,n}(i,j)*sign_{m.n}(i,j)}{\sum_{i,j} sign_{m,n}(i,j)} \quad (2)$$

where

$$sign(i,j) = \begin{cases} 1, if(Img_{m,n}(i,j) > 0) \\ 0, if(Img_{m,n}(i,j) = 0) \end{cases} (3)$$

$Aver_{m,n}$ denotes the average height of patch in m row, n column, $Img_{m,n}(i,j)$denotes the pixel of i row and j column in the patch above. It is important to set $Aver_{m,n}$=0 if$\sum_{i,j} sign_{m,n}(i,j)$=0.

iii) if the height of a patch is higher than all the patches surrounding it, then the patch is possibly a part of head, and we selected it as candidate patch. Examples are given in figure 2.5.

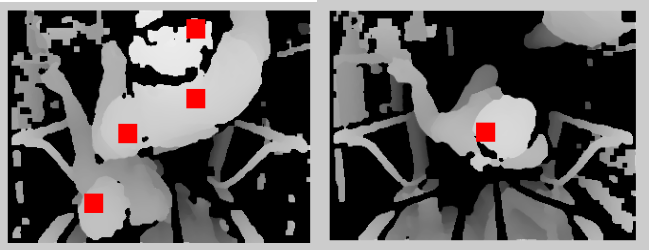

Fig 2.5Obtained candidate patches, some of the candidate patches are not locate in head regions, further measures should be taken to distinguish them.

The candidate obtained above is not perfect, some of the candidate patches are not locate in head region, in order to distinguish head patch from non-head patch, it's necessary for us to obtain the head region precisely. Then, we can extract features of the region and distinguish them by means of machine learning and pattern recognition. ROI segmentation is necessary before feature extraction. Obviously, there are gradient between head region and non-head region such as shoulder, we can use region growing method or super pixel segmentation method for ROI segmentation. This paper select SLIC super pixel segmentation for it's good effect as well as fast speed [2].

After SLIC super pixel segmentation, the result is exhibited in figure 2.6.

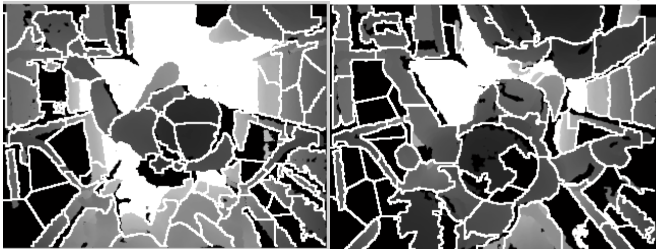

Fig 2.6Theresult of SLIC super pixel segmentation

A head region is divided into several small super pixel blocks, what we can do is to merge these blocks into one single block.

After super pixel segmentation, each pixel is labeled, and from the candidate patches, we have candidate labels.

For each candidate label, do

i) Calculate the average pixel value of the super pixel block corresponding to the label with the same way of equation (2) and (3). We denote the average pixel value as S_c..

ii) For each label adjacent to the candidate label, calculate the average pixel value, for block k, denote the average pixel value as S_k

iii) If $T_l \leq S_c - S_k \leq T_h$, change all the labels in block k into the candidate label.

Repeat step (i) until there's no new blocks merging to the candidate label. The result of merging can be seen in figure 2.7.

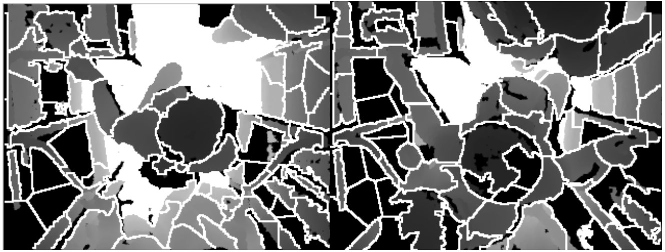

Fig. 2.7The result of merging blocks. Compared with fig 2.3.4, most of the blocks belonging to one object (head or shoulder or others) are merged, except for blocks whose average pixel value are zero.

For most of the case, after merging, the whole head region is extracted (such as left figure of fig 2.7), however, there are still some bad depth image like right figure of fig 2.7, in which exists too much zero points, leading to an incomplete extraction. The zero points offer us no information, thus, RGB image is used to make up the zero-points-block. The RGB image corresponding to fig 2.7 is showed in fig 2.8-A:

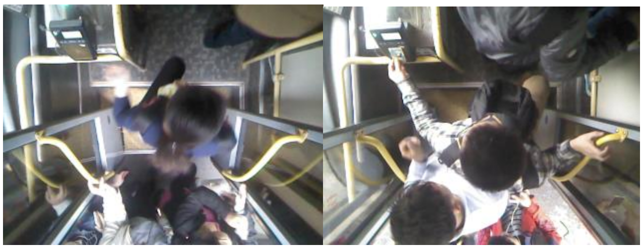

Fig. 2.8-A The RGB image of fig 2.3.5

Assume that head region has the same color, we make sure whether a zero-point-block belongs to potential head region as follows:

For each candidate block, we have a block in RGB image. For each pixel (R_1, G_1, B_1) in the block, we define its color as$(\left\lceil\frac{R_1}{10}\right\rceil, \left\lceil\frac{R_1}{10}\right\rceil, \left\lceil\frac{R_1}{10}\right\rceil)$, in which $\lceil R_1/10 \rceil$ means round up to an integer. We have ⌈256/10⌉*3=78 colors, do histogram statistic for color in the RGB block, find the most frequency colorC_{f1}: (R_{f1}, G_{f1}, B_{f1}) . For each zero-points-block next to the candidate block, do the same thing and find the most frequency color isC_{f2}: (R_{f2}, G_{f2}, B_{f2}),, calculate the Euclidean distance of C_{f1} and C_{f2}:

$$d = \sqrt{(R_{f1} - R_{f2})^2 + (B_{f1} - B_{f2})^2 + (G_{f1} - G_{f2})^2} \quad (4)$$

If d ≤ Th, then we regard the zero-points-block as part of potential head region.

The final effect of region segmentation is shown in fig 2.8-B

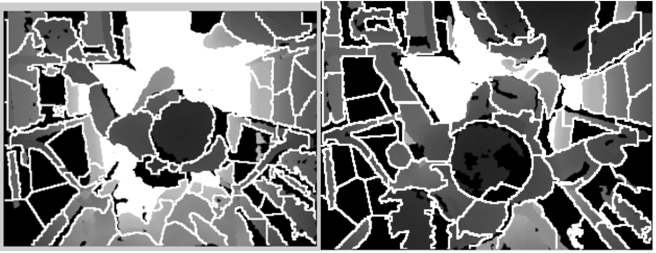

Fig.2.8-B The final result of segmentation, the zero area is correctly merged.

2.4. *Classification*

After candidate region segmentation, there are several candidates, and only part of them are heads. Example given in figure 2.9

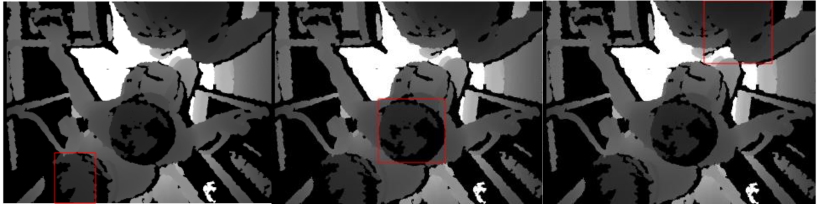

Fig. 2.9Thethree candidates obtained in chapter 2.3, the first two are heads, and the last one is interference that we must remove

To judge whether the candidate is real head or not, a SVM is trained to recognize it. We cut 1000 positive samples and 2000 negaive samples, which are shown in fig. 2.10, by program, and then extract feature and train.

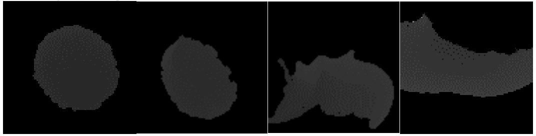

Fig.2.10 Theleft two figures are positive samples and the left two are negative

The feature we extracted is composed as follows:

i) circle degree

$$\boldsymbol{d} = \frac{\boldsymbol{4\pi S}}{\boldsymbol{C^2}}(5)$$

in which S denotes the area of region, and C denotes the perimeter of the region. The heads of human beings are close to a circle, which means its circle degree is close to 1.

ii) eccentricity of ellipse who has the same standard two order central moment with the candidate region. A large eccentricity means the region is likely to have the shape of circle.

iii) variance of the region (zero points are not counted), the variance is divide by 10 for normalization.

iv) divide the region into 3*3=9 parts, calculate the difference of the average pixel value of each part and the average value of the whole region, here we get 9 numbers, do two norm normalization.

The 12-dimension-feature is extracted and a SVM is trained, through the SVM, we can extract head targets precisely. The effect of detection is shown in figure 2.11

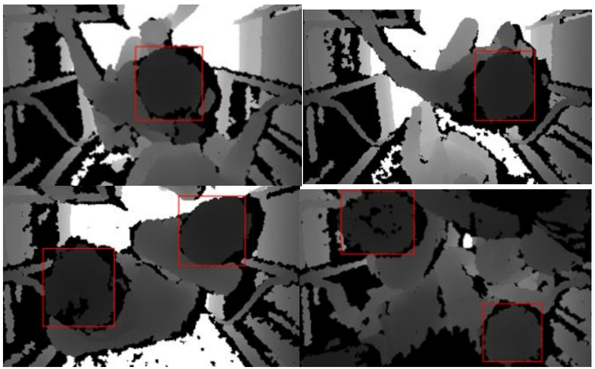

Fig. 2.11The result of detection

3. Tracking and Counting

The most popular tracking methods are kalman filter, meanshift, camshift and so on, these methods have good robustness, however, they need too much calculation to fit for multi-target tracking. According to the feature of depth image and that the system must be real time, a feature-based tracking strategy combined with meanshift is proposed in this paper.

3.1. *multi-target tracking*

The most important thing in multi-target tracking is the match of each target. In chapter 2, we have obtained head targets in each frame. This paper selected a feature-based method to match the head targets. The features are:
i) circle degree d. It’s the same as chapter 2.4.

ii) eccentricity of ellipse who has the same standard two order central moment with the target region. We denote it with e;

iii) variance of the region, the same as chapter 2.4. denote it with v;

iv) histogram statistic of pixel values: there are 8 intervals, or 8 dimensions, for each pixel, which interval it belongs is listed as follows:

$$\begin{cases} n = 1; & p - m < 15 \\ n = 2; & -15 \le p - m < -10 \\ n = 3; & -10 \le p - m < -5 \\ n = 4; & -5 \le p - m < 0 \\ n = 5; & 0 \le p - m < 5 \\ n = 6; & 5 \le p - m < 10 \\ n = 7; & 10 \le p - m < 15 \\ n = 8; & p - m \ge 15 \end{cases} \quad (6)$$

In the equation(6), p denotes the pixel value, and m denotes the mean value of the head region. Counting the number of pixels belonging to each interval, we have $\boldsymbol{X_i}$ (i=1,2…8), $\boldsymbol{X_i}$ denotes the number of pixel belonging to interval i. Normalize with the equation:

$$\boldsymbol{X_i^n} = \frac{x_i}{\sqrt{\sum_{k=1}^{8} x_k{}^2}} (7)$$

In which $\boldsymbol{X_i^n}$ denotes the normalized value of dimensioni.

v) the coordinate of the region center [x, y].

There are 13 demensions in all, that are [d, e, v, $\boldsymbol{X_k^n}$ $(\boldsymbol{k = 1, 2 \dots 8}),$ m*x, m*y], m denotes the weight of position variation. Calculate the feature euclidean distance of targets in adjacent frame, the pair with minimum distanceare matched if they satisfy

$$\begin{cases} d_f \leq Th1 \\ d_d \leq Th2 \end{cases} (8)$$

Where$d_f$denotes the distance of the feature of the two targets and $d_d$denotes the distance of region center of the two targets.

as well as the distance of their region center smaller than a threshold, are matched.

3.2. *tracking by detection and detection by tracking*

Chapter 3.1 shows a strategyof tracking by detection, however, detection is not perfect, it has false-alarm and false-dismissal. The paper proposed a simple superviser to detect false-alarm and false-desmissal [3,5]. with the fact that people can't vanish into thin air or appear out of nowhere, we set a region in the center of the image, from where people can't walk out or walk in, if a target in frame $f_1$appear in the region, but in the frame$f_1 + 1$, there's no target to match with, we believe that there's false-dismissal in frame$f_1 + 1$. Thus, mean-shift tracking is used on the target in frame , and we regard the tracking result as the detection result of frame$f_1 + 1$. For targets pop up in the region, we regard it as flase-alarm, and do no tracking for it [4].

3.3. *Counting*

We set two virtual lines for counting, which can be seen in figure 3.1.

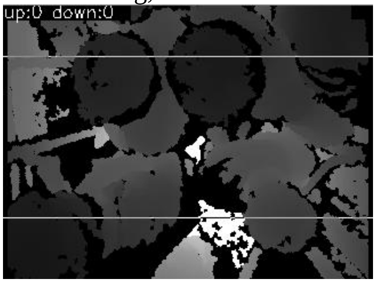

Fig.3.1Two virtual lines for counting

Only if people pass the two lines successively, then we regard it as getting in/out of the bus.

4. Results

The data was obtained in a busy bus line in Shenzhen, China, in was recorded in rush hour, we selected one day's data and check the accuracy of the system, the chart is shown in figure 4.1

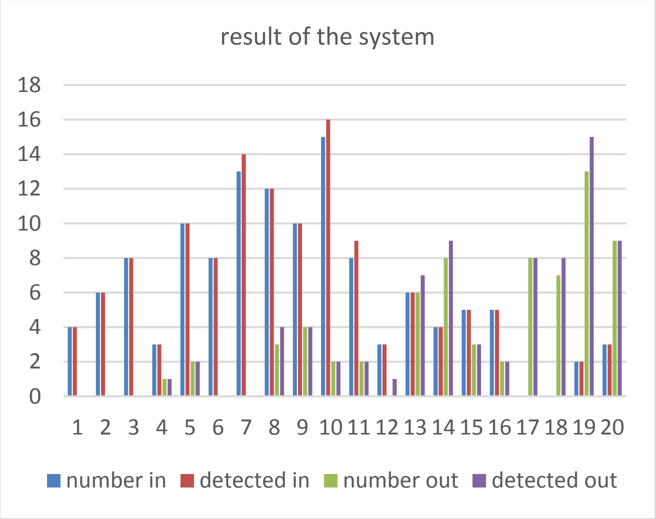

Fig.4.1 The result of the system

We have the accuracy of 95.4%, in fact, if it’s not crowd, we have an even higher accurancy.

We have the speed of 15 fps in the condition of i5-4590 and 8GB RAM, and our video is 10 frame per second, satisfy the demond of real-time.

5. Conclusion and Future Works

In this paper, we use Microsoft Kinect cameras simultaneously obtain depth image and RGB image to design a people-counting-algorithm for getting in/out of a bus, a complete set of algorithms including detection, tracking and counting are designed. Wherein the detection part of this super we use super resolution technique for dividing the depth map, and we combined RBG image for head region extraction, and our tracking method combined feature-based-tracking and mean-shift tracking. Experimental show that the proposed algorithm is stable and accurate, and meets the real-time requirements.

Although our algorithm tests have achieved good results, there is still a long way to go if we want to come it into actual use, we need to optimize the algorithm; In addition, in the case of very crowded, due to increased shaking, the accuracy rate declines, which are the future work we need to do.

Acknowledgement

This work was supported by the National Key Research and Development Program of China with No.2016YFB0101001 in part, the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen under Grant

No.JCYJ20150831192224146 and National Science Foundation of China under Grant No.61471216.

References

1. Terada K Yoshida D, Oe S, and Yamaguchi J. A method of counting the passing people by using the stereo images. IEEE International Conference on Image Processing.1999(2):338-342.
2. Achanta,Radhakrishna, et al. Slicsuperpixels. No. EPFL REPORT 149300. 2010
3. MykhayloAndriluka Stefan Roth Bernt Schiele. People-Tracking-by-Detection and People-Detection-by-Tracking. CVPR.2008:1-8
4. Comaniciu D, Meer P. Mean shift analysis and applications. Proceedings of the Seventh IEEE International Conference Computer Vision,1999
5. ZdenekKalal, KrystianMikolajczyk, and Jiri Matas. Tracking-Learning-Detection. ieeeTranscations on Pattern Analysis and Machine Intelligence, vol. 34, No.7, July 2012
6. Yahiaoui, Meurie ,Khoudour , and Cabestaing , A people counting system based on dense and close stereovision.International Conference on Image and Signal Processin. 2008(5099):59-66.
7. Chao-Ho Chen, Yin-Chan Chang, Tsong-Yi Chen, Da-Jinn Wang. People Counting System for Getting in/Out of a Bus Based on Video Processing. 2008 Eighth International Conference on Intelligent Systems Design and Applications
8. Nicola Bernini, Luca Bombini, Michele Buzzoni, PietroCerri, Paolo Grisleri. An embedded system for counting passengers in public transportation vehicles. Mechatronic and Embedded Systems and Applications (MESA), 2014
9. Jau-WoeiPerng, Ting-Yen Wang, Ya-Wen Hsu, Bing-Fei Wu. The design and implementation of a vision-based people counting system in buses. System Science and Engineering (ICSSE), 2016