An Evaluation of two People Counting Systems with Zenithal Camera

Suellen Silva de Almeida
susilvaalmeida@gmail.com

Victor Hugo Cunha de Melo
victorhcmelo@gmail.com

David Menotti
menottid@gmail.com

Universidade Federal de Ouro Preto
Campus Universitário, Morro do Cruzeiro
Ouro Preto - Minas Gerais, Brazil

Abstract

This paper presents two methods for people counting. The first one is divided into people segmentation, tracking and counting, developed for a system using a zhenital camera. The initial step consists of block-wise background subtraction, followed by k-means clustering to allow segmentation of single persons in the scene. The number of people in the scene is estimated as the maximal number of clusters with acceptable inter-cluster separation. Tracking of segmented people is addressed as a problem of dynamic cluster assignment between two consecutive frames and it is solved in a greedy fashion. Further details of this greedy solution can be found in this paper. Moreover, this paper presents another method to count people based on analysis of multiple lines. The first part of this algorithm is to detect the movement of people, and regions through which they pass are extracted. From this, the count is performed by virtual lines. Finally, it examines the results for each line. Further details on the two strategies can be found in this article.

Keywords– *People counting; people segmentation; background subtraction; people tracking.*

1. Introduction

Detection, tracking and people counting [1] is very useful for many commercial applications such as monitoring of public spaces, soccer stadiums, or bus stations. It has implications for security, and allows to collect information about systems which can be used to identify patterns in traffic by hours, optimize scheduling work, monitor the effectiveness of events, and other applications.

Beyond image sensors, mechanical and other forms of technology sensors are used to count people [19]. Systems using mechanical counters, such as turnstiles, count only one person at a time and may obstruct the passage, causing congestion if there are many pedestrians. Due to its design, it is subject to subcounting. Systems using infrared beam or heat sensors do not obstruct the passage, but don't present accuracy to identify people in crowded groups. For those reasons, cameras were selected as instrument of detection.

The background segmentation is the first step in several computer vision applications. It is usually obtained in systems of human detection by computing the difference pixel-by-pixel between the current frame and the image of the background, followed by an automatic threshold [19] [9] [18]. If the accuracy of this approach is not granted [13] [5], a strategy by means of blocks is preferable because it produces a more stable segmentation in the presence of light and shadow changes.

Rossi and Bozzoli [17] uses features in shades of gray, sensitive to high frequency changes in the scene to detect moving objects. They use template matching to track the extracted features. Another approach is based on the shape detection or recognition. Theses approaches aim to detecting people by searching for heads, legs or silhouettes [20] [15] [12]. There is a solution specially suitable for crowded situations based on feature points clustering [16] [7] in order to identify each moving entity thanks to their independent motion.

Huang and Chow [13] utilizes more elaborated features to describe the blur in the foreground. Instead of tracking the individuals, they simply count the number of people in the area of interest. Velipasalar *et al.* [19] proposes the use of size from blots detected to target individuals and the procedure of mean-shift as a way to handle blots merged. In [4], Beleznai *et al.* also employ the mean-shift procedure to develop a generally people tracking system .

Some authors [14] [11] use the oblique camera positioning. This allows the detection of more features, but present

problems with respect to occlusions and about the privacy of individuals. The zhenital camera positioning, in turn, consists of a camera placed overhead people that effectively removes the problem of occlusions between objects. Moreover, the overhead positioning offers further advantages [19] [6] as:

- objects appears with size relatively constant;
- provides a better view of people in the scenario;
- more privacy, because it does not recognize the person's face;
- eliminates the need for calibration;
- simple and easy to maintain.

However, the segmentation result often contains mixed blots, belonging to people very close.

In [8] is proposed a method for counting people entering or leaving a bus based on video processing, which camera positioning is also zhenital. Each frame caught is divided into several blocks, and each block is classified according to its movement vector . If the number of blocks with similar motion vectors is greater than a threshold, these blocks belong to the same moving object. As a result, the number of moving objects is the number of passengers entering or exiting the bus. The problem of agitation in the camera and light variation in bus is overcome in this method, showing 92% accuracy in tests.

This paper presents an evaluation of a method for people segmentation, tracking and counting [1]. It describe in more details the method and the greedy algorithm for tracking people. In addition, we present another approach [3] for the proposed problem, but did not achieve any results.

The article is organized as follows. Section 2 presents a detailed explanation of the first approach. In Section 4 is the analysis of experimental results. Section 3 presents a study of the second method. Conclusions are presented in Section 5 and future work in Section 6.

2. System Architecture

The method for counting people is divided into: video capture, background subtraction, segmentation, tracking and people counting (Figure 1). Operations in video frames are made in blocks of pixels, which reduces the amount of computations and obtains the same effect of operations made pixel by pixel. The standard block size is 8x8.

2.1. Background Subtraction

The first part of the method is the subtraction of the background. This operation is essential for detection of persons which will be done later by comparison of blocks of frame

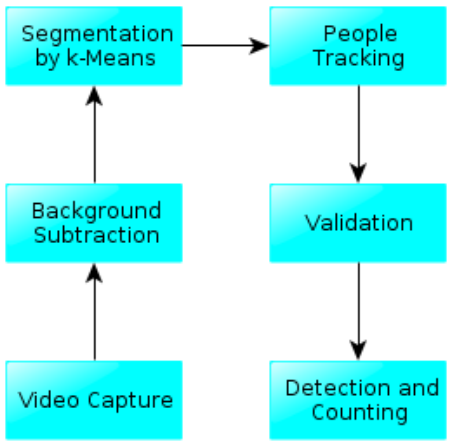


Figura 1. System's Flowchart

with the blocks of the current frame belonging to the background. Images (frames) belonging to the video's background are obtained by the following filter

$$F^{t+1} = (1 - \alpha) \cdot F^t + \alpha \cdot I^t \tag{1}$$

where F and I represents, respectively, background frames and the original video frames, t is the number of frame, and α is a learning rate that can range between 0.01 to 0.1. This rate should be adjusted according to the situation, but for this work it was arbitrary set to 0.01. The filter is applied to all frames and its channels.

The algorithm uses multiplicative factors $\beta^t_{m,n,p}$, determined using *maximum likelihood estimation* (MLE). MLE is a statistical method used to adjust the data to a model and provide estimations to the model's parameters. Indices (m, n) refer to the coordinates of the blocks and p the image channels (RGB - red, green and blue).

$$\beta^t_{m,n,p} = \frac{\sum I^t_{m,n,p} \cdot F^t_{m,n,p}}{\sum (F^t_{m,n,p})^2} \tag{2}$$

The people detection in a frame is achieved by the difference between the maximum and minimum multiplicative factors. They are calculated by the highest and lowest β between the image channels and the difference between them is stored in $\delta\beta^t$, for each frame, *i.e.*,

$$\delta\beta^t = \max_p \beta^t_{m,n,p} - \min_p \beta^t_{m,n,p} \tag{3}$$

The multiplicative factors from the background blocks has values close to 1. If $\delta\beta^t$ is not small or if some multiplicative factor is very different from 1, the block belongs to the object, *i.e.*,

$$P^t = \begin{cases} 1, if\ \delta\beta^t > T_1 \vee |\beta^t_{m,n,p}| > T_2 \\ 0, otherwise \end{cases} \quad (4)$$

P is the image with people and T_1, T_2 are limits between $[0.1, 0.2]$ and $[0.3, 0.6]$, respectively. These parameters should also be adjusted through experiments for each specific situation.

2.2. People Segmentation

At this point, there is an image for each P frame where people appears. The next step of the algorithm is segmentation these people. Segmentation is a difficult problem in image analysis due to several characteristics that represent a person. As people appear in these videos from above, this problem is reduced. So, people are seen as geometric shapes (Figure 2), which can be extracted by traditional techniques of *clustering* like *k-means*.

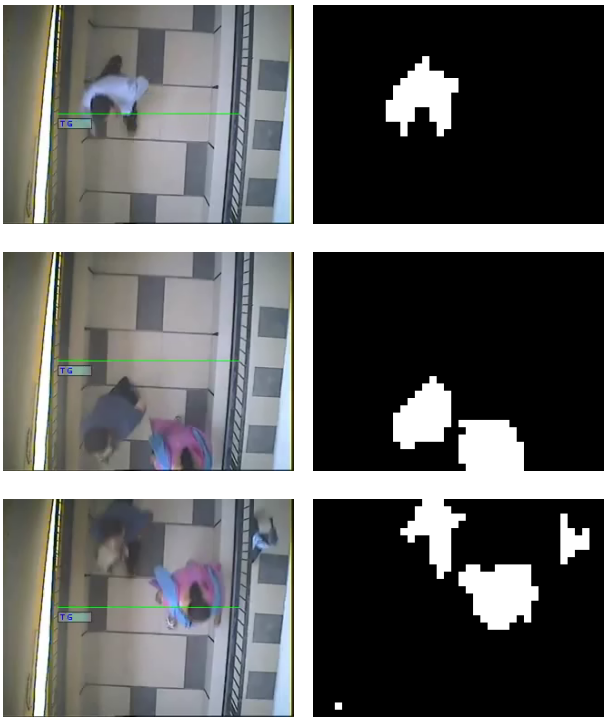
In *k-means* [10] exist k centroids, one for each group (cluster). Each individual is associated with the nearest centroid and the centroids are recalculated based on the individuals classified. However, the value of k is not known *a priori*. Finding out its value, the number of people in the scene is obtained. The value of k is estimated as the maximum number of clusters in which the distance within the clusters is greater than a minimum distance D_{min}. This constant corresponds to the average size of a person on the scene, and must be established through experiments. In an image with k clusters, whose centroids are $C_i, i = 1, 2, ..., k$, the minimum distance within the cluster is defined as

$$d^k_{min} = \min_{1 \le i < j \le k} ||C_i - C_j|| \quad (5)$$

If only one cluster, formally define $d^1_{min} = \infty$. The current number of clusters $k*$ is then estimated as the maximum number of clusters which have the minimum distance within the cluster d^k_{min} higher that D_{min}, *i.e.*,

$$k* = \max\{k | d^k_{min} \ge D_{min} \wedge d^{k+1}_{min} < D_{min}\} \quad (6)$$

In *k-means*, the initialization of the centroids is very important because it can improve the convergence of the algorithm. So whenever possible, the centroids of *k-means* algorithm was initialized as the centroids found in the previous iteration. Thus the centroid is always initialized with a position likely to be the best for the clusters.



(a) (b)

Figura 2. Results showing the steps of background subtraction and people segmentation. (a) original frame. (b) Subtracted background and segmented people.

2.3. Tracking People

At this point of the algorithm, are known people in each frame of the video. The next part is to track these people, *i.e.*, find out if the same person is in multiple frames to count them. This step was implemented in a *greedy* fashion, analyzing two consecutive frames by time. The algorithm finds out the clusters corresponding to two consecutive frames that has the shortest distance. The objective is to obtain the smallest squared Euclidean distance between clusters. So these clusters with minimum distance are marked as the same person in a binary matrix, where lines represent the clusters and the columns represent the frames. Thus, if the cluster i from the frame t corresponds to cluster i of frame $t + 1$, the matrix in position (i, t) has a value. By end of all iterations, this matrix has the value 1 in intervals in which the same person is in several frames.

$$M_{i,t} = \begin{cases} 1, \text{if } c^t_i = c^{t+1}_i \\ 0, \text{otherwise} \end{cases} \quad (7)$$

where $M_{i,t}$ represents the binary matrix of cluster i in frame t. C_i^t is equivalent to cluster i of frame t.

2.4. Counting People

The last step is to count people. This part is done by analysis of the binary matrix constructed in the previous step. As each row of this matrix represents a cluster, it is necessary analyze each line separately. By walking through these lines, if there is a change from 0 to 1, it is because a person was detected and the counter increased.

Figure 2 shows the main steps of the algorithm. The first column shows images of the documents. The images from the second column illustrates the subtraction of the background through proposed blocks (blocks of size 8 x 8 pixels) followed by people segmentation.

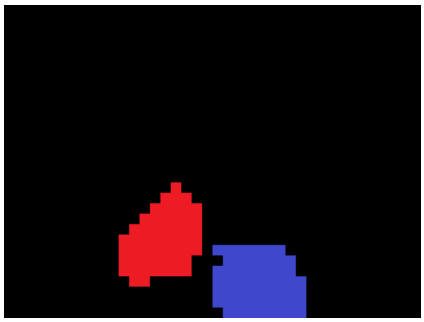


Figura 3. After segmentation by *k-means*

Figure 3 presents the results of people segmentation through k-means, where the number of clusters is automatically set using the minimum distance inter-cluster. In this case, the method correctly segmented the objects by finding the value of $k = 2$, *i.e.*, two people.

3. A Second Approach

This section presents another solution to the problem of people counting [3]. The main idea of this solution is to define an area of interest in the images where the movement of people is analyzed. Virtual lines are established in orthogonal direction to the motion.

The Figure 4 shows the parameters involved with the area of interest or counting zone, where motion will be analysed. The distance between lines must the greater than half of a person's width.

The algorithm is divided into three different steps. First motion is detected and regions where people go by are extracted. After that, the counting is performed by virtual lines. Finally, it examines the results for each line.

3.1. Extracting the Motion

To extract the motion information from background, the method uses the difference between consecutive frames of

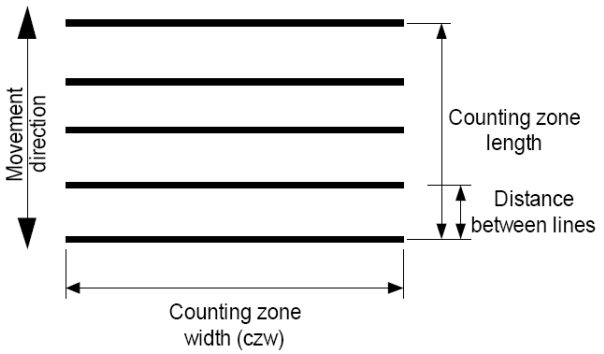


Figura 4. Couting zone

images. To avoid false positives due to noise in images, this difference is limited to a value. If the difference is greater than this value, the image contains people in movement, *i.e.*

$$D^t = \begin{cases} 1, if\ |I^t - I^{t-1}| > threshold \\ 0, otherwise \end{cases} \quad (8)$$

1where D and I respectively represents the images with people in motion and the original frames; t indicates the frame number, $threshold$ is a value determined through experiments.

3.2. Counting People

The second part of the algorithm consists of counting performed independently for each line which belongs to the area of interest from images. Each line is represented by a function l, where x axis and y corresponding to the location within the line and the cumulative number of pixels in the foreground.

$$l_x^t = l_x^{t-1} + D^t \ \ \forall x \wedge 0 \leq x < czw \quad (9)$$

$$l_x^0 = 0 \ \ \forall x \wedge 0 \leq x < czw \quad (10)$$

where x is a point on the line; czw is the width in the area of interest, which is equal to the width of the lines.

When a person crosses the line, the pixels are accumulated as described by the line's function. The people crossing the video are detected by analyzing the function of each line, and detecting ranges of values different from zero. A counter is incremented for each row when this interval is big enough to represent a person. This size is the number of occupied points on the line in interval.

As people across the video in different directions, it is necessary to detect the direction of each person. We need to

compute the optical flow by the method of *Lucas-Kanade*. The motion vector is estimated by calculating the average optical flow in the region of the range, and taking into account only the pixels where motion is detected. Finally, the dot product between the normal direction of motion is computed to determine whether the person is entering or exiting.

If two or more people crosses a line at the same time and the distance between two people is very small, the corresponding intervals detected overlap. The number of people is then determined from the overall size of the range and size of the person. Another case is when two people go through one line after another, without separation between them. Therefore, the function is not equal to zero after the first person went through a row because the second person held the same interval. To overcome this problem, time is needed for deciding when someone should be counted. This time depends on the speed of movement. If time is very small, a person walking slowly can be counted as multiple people. Therefore, time is calculated independently for each interval, based on the magnitude of the optical flow, when the camera and the size of the person.

(a) (b)

Figura 5. Results showing the steps of multiple lines. (a) original frame. (b) frame segmented and with lines.

Figure 5 shows the first two steps of the algorithm. The first column shows the original images from the video. The images from the second column illustrates the motion detection of people through multiple lines. At first, there is no line because no one passed through the region of interest. The lines began to appear when people cross this region. The line length is equal to the width of the person.

One problem with this approach is the need to reset the lines. As you can see the third image, the row size is greater than the size of people. That's because other people had before in the region where the lines are out of people.

3.3. Analysis of Multiple Lines

The last step of the algorithm consists in the overall analysis score by combining the results for each line. The counter for each region of interest is equal to the counter supported by the maximum number of rows. Thus, the zone score is obtained by combining the multiple independent lines. When motion is detected, the counter is reset each line to remove any accumulated errors on some lines.

Any results for this method were not achieved yet because of lack of information about the last step. However, as presented in the original article, the algorithm had 95% accuracy. The author used the same metrics we use to evaluate the results obtained with the first approach 2.

4. Experimental Results and Analysis

To evaluate the performance of the proposed methods, we used two generated video cameras placed at the zhenital position. The first video was shot on a bus terminal (b) (320 x 240 pixels, 145 frames) and the second provided by the authors of the article [1], was filmed in an office (a) (640 x 480 pixels, 524 frames). Both sequences have at most three people in the scene at the same time. The D_{min} varied greatly between the two videos. This was due to the distance from the camera floor that is different between the two, beyond the dimensions which are also different.

The correctness of the method was evaluated by calculating the *precision* and *recall* [2], commonly used metrics in Pattern Recognition Algorithms . By using *precision* and *recall*, the set of possible labels for a given instance is divided into two subgroups, one which is considered relevant for the objectives of the metric. *Recall* is then calculated as a fraction to correct instances of all instances that really belong to the relevant subset. *Precision* is the fraction of correct instances among those who belong to the algorithm considers the relevant subset. The *precision* can be seen as a measure of accuracy or fidelity, while *recall* is a measure of completeness.

The terms *true positive* (TP), *true negative* (TN), *false positive* (FP) and *false negatives* (FN) are used to compare the classification of an item (according to an algorithm) with the real classification of this item.

Thus, *precision* and *recall* were defined as

$$precisao = \frac{TP}{TP + FP} \tag{11}$$

Tabela 1. Comparison of results of the presented method with respect to the real number of people

	Office		**Terminal**	
	real	method	real	method
people	6	7	6	5
TP	6	7	6	5
FP + FN	0+0	1+0	0+0	0+1
precision	1.00	0.87	1.00	1.00
recall	1.00	1.00	1.00	0.83
F-score	1.00	0.93	1.00	0.90

Tabela 2. Results from [1]

	Original Method
people	20
TP	20
FP+FN	0+1
precision	1.00
recall	0.95
F-score	0.97

$$recall = \frac{TP}{TP + FN} \tag{12}$$

Another measure used was the *F-score*, which combines the *precision* with the *recall*. This metric can be interpreted as an average weighted *precision* and *recall*, where a score of *F-score* reaches its best value and the worst result in a 0.

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{13}$$

For the first method, a statistical evaluation for the proposed segmentation is shown in Table 2. The values presented in this table shows that the method is efficient by having a rate of correct segmentation approximately 90% for the office. In the end, this rate was lower than 83%. You can see that for the video Terminal, the precision's result was better, which means that, for this case, the algorithm was successful in relation to accuracy. However, the video of Office had the best results in *recall*, showing that the algorithm lost accuracy, but earn in relation to completeness. Analyzing the *F-score*, which can be interpreted as an average of two other metrics, for both videos the algorithm achieved an average of 0.91 precision. This result would be satisfactory because the best result is 1, but people counting must be performed accurately. Then, the result of *F-score* needs improvement.

Analyzing this algorithm, these errors were caused by two reasons. The first is the setting of parameters, such as D_{min} that is essential to the algorithm, and it is not trivial to determine. The other reason is the noise in images. It is necessary to filter the images to improve results.

Comparing the results with the results of the article (Table 2), we observe that the algorithm had a better performance than our experiments showed. The main reasons have been cited and one more feature can be analyzed. In [1], people are counted only when they exceed a line, it would eliminate the noise in our method and possibly improve performance.

In the second method we don't achieve results due to lack of information on your last step. However, you can analyze it due to the results presented in the original paper.

Table 3 presents the results achieved by the author through the proposed algorithm. These results were better than the results we obtained with the first algorithm, but are worse than the authors presented with the first approach. The number of *fn+fp* are greater than ours, but it probably is because they used larger videos than us. Their *precision* were more than 0.97% in all tests while we obtain at least 0.87% and maximum of 100%. T1, T2 and T3 are different cenarios tested.

Tabela 3. Tests results

	real in+out	*system in+out*	*tp*	*fn+fp*	*p*	*r*	*F*
T1	101+95	97+90	182	14+5	0.97	0.93	0.95
T2	232+241	225+233	445	28+13	0.97	0.94	0.96
T3	127+128	116+117	231	24+2	0.99	0.91	0.95

Table 4 shows a comparison between the best results obtained with the algorithms under study. You can see that in relation to the accuracy, Method 1 showed the best result, which means that this algorithm is more accurate in relation to reality. Analyzing the two other metrics used, we can complete Method 1 showed better results than the second method. But we still need to get the results of our implementation for this second algorithm to compare really.

5. Conclusions

This paper presents an evaluation of two methods for counting people. The first one is divided into segmentation people, counting and tracking using a camera system zhenital. The algorithm performs the removal of the background then the segmentation people through the *k*-means. The greedy solution to the problem of association of clus-

Tabela 4. Comparison between the best results of these two Methods

	Method 1	Method 2
precisão	1.00	0.99
recall	0.95	0.91
F-score	0.97	0.95

ters between two consecutive frames is exploited for tracking people. Video sequences were used to evaluate the results. The result is not so accurate as the original article because of the problem to determine the application's parameters and the noise is not removed from frames.

The second method is based on analysis of multiple lines and is divided into three parts. First the movement of people is detected, and the regions where people pass through are extracted. After the count is performed by virtual lines. Finally, it examines the results for each line. For this method we don't achieve results due to lack of information about the last part. Comparing the results presented in the both papers, one can conclude that the first is more accurate.

6. Future Works

As future work, we intend to implement other papers to comparison of effectiveness between methods. In addition, for the first presented system, we want to replace the segmentation via *k-means algorithm* by one *labeling algorithm* which can improve performance (in the MATLAB, implemented by the function `bwlabel`). Setup parameters need to be improved and the noise removal filter should be applied to improve accuracy. In this filter, all three channels of the image are multiplied by a constant within a block (Formula 14).

$$I^t_{m,n,p} = \beta^t_{m,n,p} \cdot F^t_{m,n,p} + W^t_{m,n,p} \qquad (14)$$

where W means the Additive White Gaussian Noise (AWGN) and $\beta_{m,n,1} \approx \beta_{m,n,2} \approx \beta_{m,n,3}$.

Regarding the second approach, we need to finish the implementation and analyze the results. If they are similar to results presented in this paper, we intend to implement improvements in the algorithm. We hope to improve the algorithm for people counting, where accuracy is as close as possible to 100%. By comparison of these papers and implemented improvements of these, we hope to reach our goal.

Finally, we will study other tracking methods, such as Particle Filter and Ant Colony, to evaluate its accuracy and use larger videos in tests.

Referências

[1] B. Antic, D. Letic, D. Culibrk, and V. Crnojevic. K-means based segmentation for real-time zenithal people counting. In *Image Processing (International Conference on Image Processing (ICIP)), 16th IEEE*, pages 2565 –2568, nov. 2009.

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[3] J. Barandiaran, B. Murguia, and F. Boto. Real-time people counting using multiple lines. *Image Analysis for Multimedia Interactive Services, International Workshop on*, 0:159–162, 2008.

[4] C. Beleznai, B. Frühstück, and H. Bischof. Human tracking by fast mean shift mode seeking, 2006.

[5] J. Bescos, J. Menendez, and N. Garcia. Dct based segmentation applied to a scalable zenithal people counter. In *Image Processing, 2003. International Conference on Image Processing (ICIP) 2003. Proceedings. 2003 International Conference on*, volume 3, pages III – 1005–8, sept. 2003.

[6] M. Bozzoli and L. Cinque. A statistical method for people counting in crowded environments. *Image Analysis and Processing, International Conference on*, 0:506–511, 2007.

[7] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 594–601, Washington, DC, USA, 2006. IEEE Computer Society.

[8] T.-Y. C. Chao-Ho (Thou-Ho) Chen, Yin-Chan Chang and D.-J. Wang. People counting system for getting in/out of a bus based on video processing. *Intelligent Systems Design and Applications (ISDA)*, pages 565–569, 2008.

[9] S.-Y. Chien, Y.-W. Huang, B.-Y. Hsieh, S.-Y. Ma, and L.-G. Chen. Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques. *Multimedia, IEEE Transactions on*, 6(5):732 – 748, oct. 2004.

[10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2000.

[11] H. Elik, A. Hanjalic, and E. Hendriks. Towards a robust solution to people counting. In *International Conference on Image Processing (ICIP))*. IEEE Computer Society, 2006.

[12] M. Han, W. Xu, H. Tao, and Y. Gong. An algorithm for multiple object trajectory tracking. In *CVPR (1)'04*, pages 864–871, 2004.

[13] D. Huang and T. W. S. Chow. A people-counting system using a hybrid rbf neural network. *Neural Processing Letters*, 18:97–113, October 2003.

[14] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos. Estimating pedestrian counts in groups. *Comput. Vis. Image Underst.*, 110(1):43–59, 2008.

[15] X. Liu, P. H. Tu, J. Rittscher, A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, pages 306–311. IEEE, 2005.

[16] V. Rabaud and S. Belongie. Counting crowded moving objects. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 705–711, Washington, DC, USA, 2006. IEEE Computer Society.

[17] M. Rossi and A. Bozzoli. Tracking and counting moving people. In *Image Processing, 1994. Proceedings. International Conference on Image Processing (ICIP)-94., IEEE International Conference*, volume 3, pages 212 –216 vol.3, Nov. 1994.

[18] L. Snidaro, C. Micheloni, and C. Chiavedale. Video security for ambient intelligence. *Systems, Man and Cyberne-*

tics, Part A: Systems and Humans, IEEE Transactions on, 35(1):133 – 144, jan. 2005.

[19] S. Velipasalar, Y.-L. Tian, and A. Hampapur. Automatic counting of interacting people by using a single uncalibrated camera. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1265 –1268, july 2006.

[20] X. Zhang and G. Sexton. A new method for pedestrian counting. *Fifth Int. Conf. on Image Processing and its Applications*, pages 208–212, Jul 1995.