XIV KRAJOWA KONFERENCJA ELEKTRONIKI
Darłówko Wschodnie, 08 - 12.06.2015

mgr inż. Agata Chmielewska
dr inż. Tomasz Marciniak
prof. dr hab. inż. Adam Dąbrowski

Wydział Informatyki
Politechnika Poznańska
ul. Piotrowo 3a, 60-965 Poznań

Separation of moving objects for automatic people counting

Key words: CCTV, intelligent video analysis, BLOB analysis, moving objects detection and tracking, object counting

ABSTRACT

In this paper, we adress a problem of automatic counting of people treated as specific moving objects visible in video recordings obtained from CCTV (closed circuit television) systems in urban areas (e.g. the pedestrian walkways monitoring). Our goal was to develop an effective method to assess a number of separate moving objects recognized in a single BLOB (binary large object) in order to count people, who are close to each other. This problem was solved with the use of the Bayes classifier and the results of the people counting system are shown. The people counting mechanism is based on a virtual division of the input image into parts. The system was implemented with the use of the Matlab environment. The tests were performed with our own databases of long-term video recordings.

1. INTRODUCTION

Nowadays in modern cities numerous CCTV (closed circuit television) systems are commonly installed. Video cameras can be found both in streets and on/in the buildings. Thanks to that, a fully automatic analysis of movements and behavior of objects such as vehicles and individual people or even crowd can be realized in the monitored urban areas.

This paper presents construction and implementation of the algorithm for detecting and counting people using the video surveillance systems. First, the methods used so far in experimental and even in commercial systems, which can be used for people counting, are recapitulated. Next, the model for detecting people, who passed under a camera positioned vertically down, is discussed. The prepared algorithm takes the direction of persons movement into account. This article presents also important features of the prepared people counting model and the appropriate choice of the control parameters, which are necessary for proper operation in the continuous mode.

Previous authors' studies concerned a problem of real-time people counting by monitoring an entrance to a building with a digital video camera positioned vertically down, using the indoor scenes [1, 2]. In paper [1] the video processing operations necessary for the bidirectional people counting were described using the modified people tracking model realized in the Matlab/Simulink environment [3]. In paper [2] a design of the autonomous embedded arrangement realized with the Raspberry Pi microcomputer [4] was presented. The developed device performs all calculations directly on the input data and transmits the video signal together with the respective metadata to the display.

2. RELATED WORK AND COMMERCIAL SYSTEMS

An intereseting system for counting people attending large public events was described in [15]. There are also some commercially available systems for people counting, which can mainly be used to control movements of people in public places (such as offices, shopping centers, etc.) to provide statistical data about the sale success or the rush hour crowd density. The CCTV systems are increasingly being equipped with some people counting facilities e.g. by the following companies: CountWise, Honeywell, VideoTurnstile and VisualTools [11]. These systems offer continuous bidirectional people counting and displaying the current number of people in a building. Available DVRs (digital video recorders) with a people counting function provide a quite high accuracy of counting. Some authors still try to improve effectiveness of such object counting systems by the analysis of the HSI (hue saturation intensity) histograms calculated from color images [7, 8]. An improvement of the people counting quality is also possible with the use of the tracking-by-detection models in a particle filtering framework [9]. Effective people counting systems can also be realized with the time-of-flight camera, which uses the depth information to detect the moving objects [10].

However, as we had to design an autonomous compact system with all discussed functionalities integrated in it, we needed to design our own people counting algorithm, although we decided to use a quite standard approch for the automatic people counting, i.e. the BLOB (binary large object) tracking realized with the Kalman filtering [5, 6]. The system output is a set of temporal and spatial coordinates of each recognized pedestrian.

3. PROPOSED SOLUTION FOR PEOPLE COUNTING

In the presented approach a standard camera without the depth data and the authors' modified Motion-Based Multiple Object Tracking model [3] were used in order to count people in outdoor / indoor scenes with a camera positioned vertically down. The counted people must move from the left to the right or in the opposite direction (in relation to the sides in the video frame). A virtual line placed on the ground should be pointed. Crossing this line will increment (or decrement) the counter.

The proposed modified and highly efficient model for people counting consists of the following steps:

- acquisition of video sequences with the resolution of 720×576 pixels at 25 fps (frame per second)
- moving objects detection based on GMM (Gaussian mixture model); this model was chosen in order to minimize the impact of lighting changes in outdoor scenes (in this model there is no reference frame)
- object detection with the BLOB analysis; the minimum BlobArea parameter defines the threshold in the binarized image below of which a specified number of white pixels in the BLOB are not considered as a human object and are not taken into account for the further analysis; the minimum BlobArea parameter was optimized in order to not take too small objects into account, e.g., birds, dogs, or the noise resulting from changes of the light conditions; the outputs of this algorithm are: BoundingBoxOutputPort (which determines the x and y image coordinates of the upper-left corner of the object and the width and height of this object), AreaOutputPort (which determines the number of white pixels in the BLOB but it obviously is not the product of the width and height), CentroidOutputPort (which

determines the *x* and *y* image coordinates of object center)

- object tracking with the use of the Kalman filter; this algorithm deletes objects from tracking if they are invisible for too many consecutive frames (5 frames) and creates new tracks for objects after their 0.2 sec presence in the observed scene; the motion model was improved for varying velocity and thanks to that the predicted positions of walking people are more exact
- dividing the image into two parts (left and right) with the use of the virtual line
- determining position of the object – an attribute is assigned for every moving object visible in the scene, determining whether the object is located in the left or in the right hand side of the frame according to the virtual line
- designation of directions of moving objects
- counter increment – an adequate counter is incremented, when the object has just exceeded the virtual line placed vertically along the video frame (typically in the center of the image); in the previous authors' models [1, 2] changing of the object BLOB shape and location of the object center near the virtual line may led to several increments of the corresponding counter; this disadvantage has now been resolved in the present model by appropriate reaction delay
- BLOB classification – in order to correctly count the objects, the BLOBs should be classified relating to the number of objects they contain, this is done with the naive Bayes classifier; if the classifier predicts *n* persons in a single BLOB, the actual counter value is incremented by *n*
- displaying the results – the analyzed images are displayed together with additional metadata marking the detected moving objects by surrounding the BLOBs with frames.

The optimized parameters of the motion-based multiple object tracking algorithm for the bidirectional people counting are listed in Table 1.

Table 1. Parameters optimization of the Motion-Based Multiple Object Tracking algorithm from the Matlab environment for bidirectional people counting

Algorithm module	**Algorithm parameter**	**Default**	**Optimized**
object: visionForegroundDetector	NumTrainingFrames	40	5
	MinimumBackgroundRatio	0.7	0.87
object: visionBlobAnalysis	MinimumBlobArea	400	1000
	MaximumBlobArea	-	100000
	Connectivity	-	8
	ExcludeBorderBlobs	false	false
function: detectObject()	Imopen('rectangle', *u*)	$[u, u] = [3, 3]$	$[u, u] = [3, 3]$
	Imclose('rectangle', *v*)	$[v, v] = [15, 15]$	$[v, v] = [9, 9]$
function: deleteLostTracks()	InvisibleForTooLong	10	5
	AgeThreshold	8	5
function: createNewTracks()	ConfigureKalman Filter: ConstantVelocity	[200, 50], [100, 25], 100	[50, 25], [30, 15], 40

4. EXPERIMENTS

The length of the video sequences used for tests is 5 hours. In order to check performance of the algorithm during bad weather conditions, a one-hour long video sequence was recorded during strong rain conditions. The main test video sequence contains 144 people passing under the camera from the left to the right hand side of the image and 131 people passing in the opposite direction.

The bidirectional people counting algorithm was optimized in order to precisely detect and track proper moving objects (Table 1). Incrementation of the counters were optimized, as described in the previous section, in order to avoid multiple counting of the same objects.

The algorithm efficiency statistics can be calculated with the use of: true positives (*TP*), false positives (*FP*), true negatives (*TN*) and false negatives (*FN*) [12]. A true positive value (equivalent with the hit) is the number of the correctly counted people in both directions. A false positive value (equivalent with the false alarm) is the number of false detections of moving objects. A false alarm can e.g. be a large BLOB that contains two or more persons possibly together with animals (dogs) and others objects, which should not be detected. A true negative value (equivalent with the reject) is the number of objects that should be correctly rejected. Rejections are for example animals (birds or dogs), big raindrops or bushes waving in the wind. A false negative value (equivalent with the miss) is the number of people, which are not detected and counted.

The results of the tests are shown in Table 2. The average sensitivity value (TPR – true positive rate), together with the video sequence containing rain, was 0.92. In the same case, the precision (PPV – positive predictive value) of the algorithm was 0. 9. The fall-out (FPR – false positive rate) was equal to 0.20.

The effectiveness of the algorithm was calculated as the ratio of the number of people computed by the algorithm to the actual number of people, which passed under the camera. Generally, the algorithm occured to be quite exact but it rather underestimates the actual numbers. Only in one video sequence (under the strong rain) the obtained counting results overestimated the real values. The total average calculated number to actual number of people ratio (including the strong rain case) achieved ca. 1.02. These results are also summarized in Table 2.

With the optimized parameters the proposed model does not detect small objects in the analyzed video sequence like raindrops. Only a large, steady stream of water can very seldom be false detected as a moving object of interest. Due to placement of the camera vertically down the rain typically does not strongly pass from one side of the image to another side. Thus even a quite strong rain does not affect the correct counter increments.

Table 2. Effectiveness of the bidirectional people counting algorithm

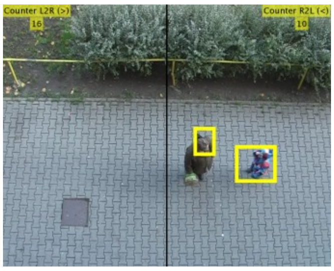
Time or rain	**1 PM**	**2 PM**	**3 PM**	**4 PM**	**RAIN**	**Average with rain**
TP	51	47	47	47	35	227
FP	3	4	3	2	12	24
TN	12	12	9	10	76	119
FN	1	1	11	8	0	21
TPR = TP / (TP+FN)	0.98	0.97	0.81	0.85	1.00	0.92
FPR = FP / (FP + TN)	0.20	0.25	0.25	0.16	0.13	0.20
PPV = TP / (TP + FP)	0.94	0.92	0.94	0.95	0.74	0.90
NPV = TN / (TN + FN)	0.92	0.92	0.45	0.55	1.00	0.77
Actual number of people	53	52	59	56	39	259
Calculated by the algorithm number of people	53	55	49	50	48	255
Calculated to actual number of people ratio	1.00	1.05	0.83	0.89	1.33	1.02

During the preparation of data to create the classifier the weights were assigned to the BLOBs. The weights for detected moving objects define how well the algorithm operates. A five-point scale (from 1 to 5) was established with 1 a very low and 5 a very high quality of the objects detection (Fig. 1). During the analysis 2582 objects (BLOBs) were examined and rated by author. The algorithm for the object detection worked very well – only 6.46 % of the moving objects were assigned as low and very low detection quality. The 22.39 % of the moving objects were assigned as the medium detection quality. The rest of the results (71.15 %) were assigned as objects with high and very high detection quality.

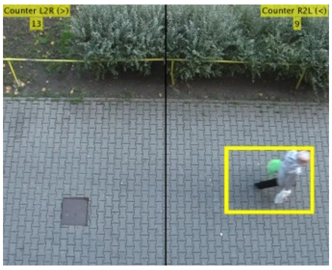
The BLOB dimensions: width, height, and area (the number of pixels in the BLOB) do not rise linearly with the number of objects in the BLOB. Results of the analysis of the BLOB height, width, and area distributions according to the assigned BLOB class are shown in Fig. 2. A probabilistic nBayes classifier was used for the purpose of prediction the number of people in the BLOB. It is easy to implement and computationally not very time-consuming.

When creating a classifier, the authors tested different configurations of features, which can assist in prediction the number of people in the BLOB. These are the following sets of features (listed together with correlations between attributes):

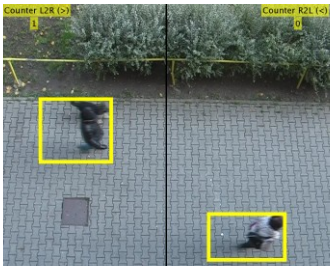
- width and area – correlation is equal to 0.80;
- height and area – correlation is equal to 0.84;
- width and height – correlation is equal to 0.55;
- area, width, and height.



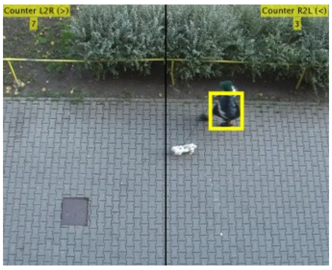
Weight for
left object:1 (very low)
Number of objects in the BLOB: 1



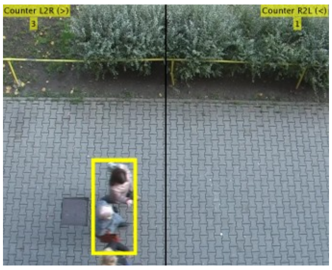
Weight: 2 (low)
Number of objects
in the BLOB: 1



Weight for both objects:
3 (medium)
Number of objects in the BLOB: 1



Weight: 4 (high)
Number of objects in the BLOB: 1



Weight: 5 (very high)
Number of objects in the BLOB: 2

Fig. 1. Examples of weights assessment to detected moving objects in the bidirectional people counting system in order to number objects in the BLOB

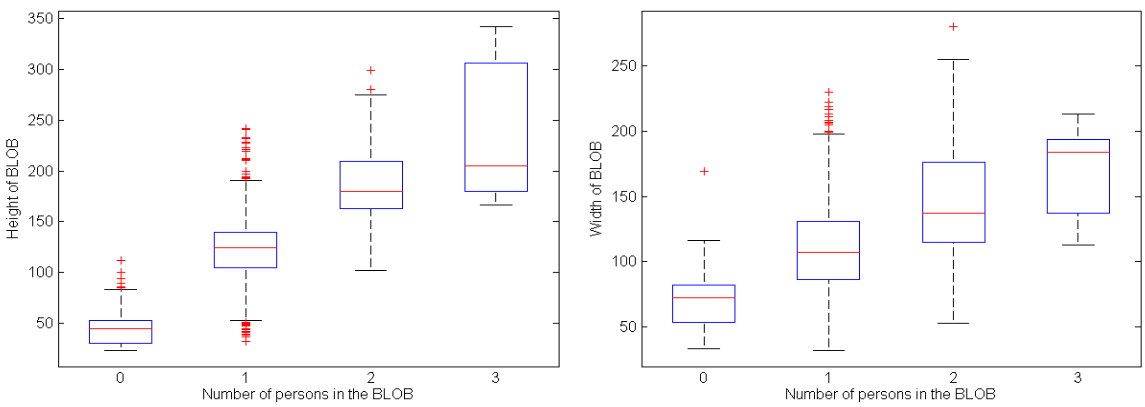


Fig. 2. Height and width of the BLOB according to the number of people in the BLOB

The best result with the BLOB width and height was obtained for 75 % of the tests and with the BLOB height and area for the rest, i.e., 25 % of the tests. Thus the BLOB width and height features were used to build the classifier.

5. CONCLUSIONS

The solution presented in this paper strongly improves effectiveness of the automatic counting of moving pedestrians present in public places. The properly selected algorithm parameters affect the correct detection and tracking of people in outdoor scenes even during the rain. Over 70 % of the detected BLOBs were assigned very good quality.

In future, the authors intend to expand the system with classifiers, which will be appropriate for cameras placed at various heights. It is also planned to realize the presented algorithm in an embedded system equipped with a DSP (*digital signal processor*) using the experience gained in the designs reported in [13, 14].

ACKNOWLEDGMENT

The work supported partly by DS/2015 and by the Scholarship for Ph.D. students specializing in major strategic development for Wielkopolska, Poland, submeasure 8.2.2 “Human Capital Operational Program”, co-financed by the European Union under European Social Fund.

REFERENCES

[1] T. Marciniak, A. Dabrowski, A. Chmielewska, M. Nowakowski, “*Real-Time Bi-Directional People Counting Using Video Blob Analysis*”, IEEE NTAV/SPA 2012, IEEE New Trends in Audio and Video / Signal Processing Algorithms, Architectures, Arrangements and Applications, pp. 161–166, 2012

[2] T. Marciniak, A. Chmielewska, A. Dabrowski, A. Malina, “*People counting vision system based on ARM processor programmed using Simulink environment*,” Electronics – constructions, technologies, applications, No. 6, 2014, pp. 55–59, 2014

[3] Computer Vision System Toolbox; Matlab http://www.mathworks.com/help/vision/index.html; 2015

[4] Raspberry Pi microcomputer, www.raspberrypi.org, available in 2015

[5] O. Masoud, N. P. Papanikolopoulos, “*A Novel Method for Tracking and Counting Pedestrians in Real-Time Using a Single Camera,*” IEEE Trans. Vehicular Technology, vol. 50, No. 5, pp. 1267–1278, 2001

[6] W. Ye, Z. Zhong, “*Robust People Counting in Crowded Environment*,” Proceedings of IEEE International Conference on Robotics and Biometrics, China, pp. 1133–1137, 2007

[7] C. Thou-Ho, C. Tsong-Yi, C. Zhi-Xian, “*An Intelligent People-Flow Counting Method for Passing Through a Gate,*” 2006 IEEE Conf. on Robotics, Automation and Mechatronics, pp. 1–6, 2006

[8] C. Thou-Ho et. al., “*People Counting System for Getting In/Out of a Bus Based on Video Processing,*” Eight International Conf. on Intelligent Systems Design and Applications, pp. 565–569, 2008

[9] M. D. Breitenstein et. al., “*Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera,*” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, No. 9 pp. 1820–1833, 2011

[10] C. Stahlschmidt, A. Gavriilidis, A. Kummert, “*Density Measurements from a Top-View Position using a Time-of-Flight Camera*,” International Workshop on Multidimensional Systems, pp. 193–198, 2013

[11] Retail Surveillance, http://retail-surveillance.com/index.htm, access: 02/12/2015

[12] T. Fawcett, “*An introduction to ROC analysis*,” Pattern Recognition Letters 27, pp. 861–874, 2006

[13] A. Chmielewska, A. Dąbrowski, A. Namerła, P. Pawłowski, R. Weychan, M. Stankiewicz, “*Comparison of NI LabVIEW and NI Vision Builder AI environments in fast prototyping of video processing algorithms for CCTV using smart camera*,” Electronics – constructions, technologies, applications, No. 5 pp. 72–76, 2011

[14] A. Chmielewska, R. Weychan, T. Marciniak, A. Dąbrowski, M. Hartwich, M. Owczarczak, “*Fast Prototyping for Video Monitoring Systems with the Use of DSP Module*”, International Journal of Electronics and Telecommunications, Volume 59, Issue 4: 375-381, 2013

[15] K. Kopaczewski, M. Szczodak, A. Czyżewski, H. Krawczyk, "A method for counting people attending large public events", Multimedia Tools and Applications, 2013 (online). DOI: 10.1007/s11042-013-1628-0

Separacja obiektów ruchomych w celu automatycznego zliczania osób

Słowa kluczowe: CCTV, inteligentna analiza wideo, analiza BLOB, detekcja i śledzenie obiektów ruchomych, zliczanie osób

STRESZCZENIE

W artykule omówiono zagadnienie automatycznego zliczania obiektów (osób) na podstawie nagrań wideo pochodzących z monitoringu miejskiego. Zaproponowany algorytm zlicza osoby, np. podczas obserwacji chodnika dla pieszych. W celu prawidłowego zliczania osób, naszym zadaniem było zwłaszcza opracowanie efektywnej metody szacowania liczby osób wykrywanych w pojedynczym BLOBie (ang. binary large object) z powodu ich bliskości i tej samej prędkości poruszania się. Problem ten został rozwiązany przy użyciu klasyfikatora Bayes'a, a osiągnięta efektywność systemu została przedstawiona w artykule. System został zrealizowany w środowisku Matlab a testy przeprowadzono z wykorzystaniem własnej bazy nagrań.

Rys. 1. Przykłady przypisania odpowiednich wag obiektom ruchomym w systemie dwukierunkwego zliczania osób w celu dokładnej predykcji liczby osób w danym BLOBie

Rys. 2. (Od lewej:) wysokość i szerokość pojedynczego BLOBu zależnie od liczby osób w danym BLOBie

Tabela 1. Optymalizacja parametrów modelu „motion-based multiple object tracking", który został wykorzystany do opracowania algorytmu dwukierunkowego zliczania osób

Tabela 2. Efektywność systemu służącego do dwukierunkowego zliczania osób