People Counting in Low Density Video Sequences

J.D. Valle Jr., L.E.S. Oliveira, A.L. Koerich, and A.S. Britto Jr.

Postgraduate Program in Computer Science (PPGIa)
Pontifical Catholic University of Parana (PUCPR)
R. Imaculada Conceição, 1155 Prado Velho
80215-901, Curitiba, PR, Brazil
{jaimejr,soares,alekoe,alceu}@ppgia.pucpr.br
www.ppgia.pucpr.br

Abstract. This paper presents a novel approach for automatic people counting in videos captured through a conventional closed-circuit television (CCTV) using computer vision techniques. The proposed approach consists of detecting and tracking moving objects in video scenes to further counting them when they enters into a virtual counting zone defined in the scene. One of the main problems of using conventional CCTV cameras is that they are usually not placed into a convenient position for counting and this may cause a lot of occlusions between persons when they are walking very close or in groups. To tackle this problem two strategies are investigated. The first one is based on two thresholds which are related to the average width and to the average area of a blob top zone, which represents a person head. By matching the width and the head region area of a current blob against these thresholds it is possible to estimate if the blob encloses one, two or three persons. The second strategy is based on a zoning scheme and extracts low level features from the top region of the blob, which is also related to a person head. Such feature vectors are used together with an instance-based classifier to estimate the number of persons enclosed by the blob. Experimental results on videos from two different databases have shown that the proposed approach is able to count the number of persons that pass through a counting zone with accuracy higher than 85%.

Keywords: People Counting, Computer Vision, Tracking.

1 Introduction

The automatic counting of people in video scenes is a relative new subject of research in computer vision. The growing interest on such a subject is due to its large applicability. Among several possibilities, by counting people in an environment is possible to control the functionalities of a building such as optimizing the setup of an heating, ventilation, and air conditioning system (HVAC), measure the impact of a marketing campaign, estimate pedestrian flows or the number of visitor in tourist attractions, and make possible a number of surveillance applications. However, nowadays people counting is done in a manual fashion or by

D. Mery and L. Rueda (Eds.): PSIVT 2007, LNCS 4872, pp. 737–748, 2007.
© Springer-Verlag Berlin Heidelberg 2007

means of electronic equipments based on infrared sensors or through very simple video-based analysis. These solutions usually fail when trying to count individuals when they are very close to each other or in groups due to the occurrence of partial or total occlusion or to the difficult to distinguish a clear frontier between one or more persons. In other words, such systems do not have any embedded intelligence which is able to handle close persons or groups.

Different approaches have been proposed to deal with this problem. The most straightforward strategy has been avoiding occlusions. Here, the main idea is to gather videos from cameras positioned at the top of the counting zones, which are known as top-view cameras. Kim et al. [1] and Snidaro et al. [2] avoided the total occlusion problem by placing the top-view cameras to count the passing people in a corridor. The camera position (top-view) allows the authors to estimate the number of people considering a simple strategy which matches the area of the moving object with a predefined value of the maximum area occupied by a single object. However, the main drawback of such a strategy is that the top-view cameras are usually not available in most of the current surveillance systems what requires the installation of dedicated cameras for counting purposes.

Some authors have based their approach on general purpose CCTV cameras which usually are placed in an oblique position to cover a large area of an environment. To deal with such oblique cameras some authors have employed classification techniques to take a decision about how many individuals are into the counting zone or enclosed by a blob, instead of carrying out a blind segmentation of groups into individuals. Nakajima et al. [3] use Support Vector Machines (SVM) to deal with this problem, while Gavrila [4] uses a tree-based classifier to represent possible shapes of pedestrians. A similar strategy was proposed by Giebel et al. [5] which uses dynamic point distribution models. Lin et al. [6] describe the region of interest based on Haar Wavelet features and SVM classification. A segmentation-based strategy was proposed by Haritaoglu et al. [7]. The authors have attempted to count the number of people in groups by identifying their heads. To detect the heads, such an approach uses convex hull-corner vertices on silhouette boundary combined with the vertical projection histogram of the binary silhouette.

In this paper we propose a novel people counting approach which is inspired in the Haritaoglu et al. [7]. The proposed approach is also based on the head area detection but it does not attempt to segment groups into individuals. Instead of that, features are extracted from the head region of the blobs and matched against simple templates that models individuals and groups of persons. The match is carried out through an instance-based learning algorithm. These models can be easily adapted for different application environments thanks to the simplicity of learning algorithm and the low dimensionality of the feature vectors.

Besides this strategy to cope with groups of persons, this paper presents a complete approach for automatic people counting in videos captured through a CCTV camera using computer vision techniques. The proposed method consists of detecting and tracking foreground objects in video scenes to further make the counting. As we have discussed earlier, the main problem resides in to have

a correct counting when the individuals are close to each other or in groups. In these situations the individuals are usually occluding each other. For each tracked object that reaches a counting zone, two strategies were evaluated: the first one is based on two thresholds that represent the average width and average area of a blob that enclosed a single individual. By comparing the width and area of a blob against these thresholds, one can decide if it is representing one, two or three persons; in the second one, a feature set based on a zoning scheme and shape descriptors, is computed from the head region of the blob while it is inside of the counting zone and a classifier is used to decide if the blob encloses one, two or three persons.

This paper is organized as follows: Section 2 presents an overview of the proposed approach as well as the main details of the detection, segmentation and tracking algorithms. Section 3 presents the strategies for counting people in the video scenes. The experimental results on video clips from two different databases are presented in Section 4. Conclusions and perspectives of future work are stated in the last section.

2 System Overview

Figure 1 presents an overview of the proposed method. Through a video camera placed in an oblique position in the environment, video is captured and the frames are preprocessed to reduce noise caused mainly by lightning variations. After detecting and segmenting the motion objects, which represent regions of interest, their blobs are defined and tracked at the subsequent frames. When a blob enters a counting zone its shape is described based on a set of features. Two strategies were evaluated for executing the counting process. The first one considers the use of threshold values computed on the blob width and the head region area to decide how many individuals are enclosed by the blob while the second focuses on the top region of the blob, called the head region and extracts a set of features that describe the shape of the objects enclosed by head region of the blob. The resulting feature vector is the input to a k Nearest-Neighbor classifier (k-NN) which decides if the blob encloses one, two or three individuals.

2.1 Detection and Segmentation of Foreground Objects

In the last years, several approaches to detect motion in video have been proposed in the literature [8]. However, the main limitation of such techniques refers to the presence of noise mainly due to the variations in the scene illumination, shadows, or spurious generated by video compression algorithms. Background subtraction technique is one of such segmentation approaches that is very sensitive to illumination. However it is a straightforward approach and it has a low computational complexity since each new frame is subtracted from a fixed background model image followed by a thresholding algorithm to obtain a binary image which segments the foreground from the background. A median filter of size 3x3 is applied to eliminate the remaining noise. The resulting image is

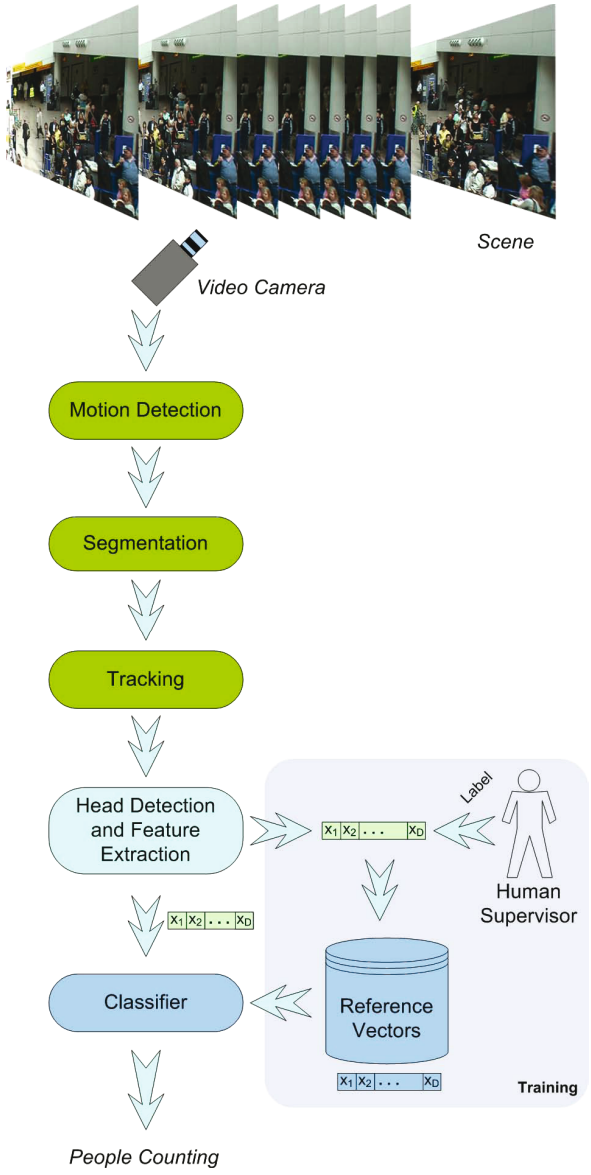

Fig. 1. An overview of the proposed approach to count people in video scenes

a set of pixels from the object in motion, possibly with some non-connected pixels. Morphological operations such as dilation and erosion are employed to assure that these pixels make up a single object. In such a way, what was before a set of pixels is now a single object called blob which has all its pixels connected.

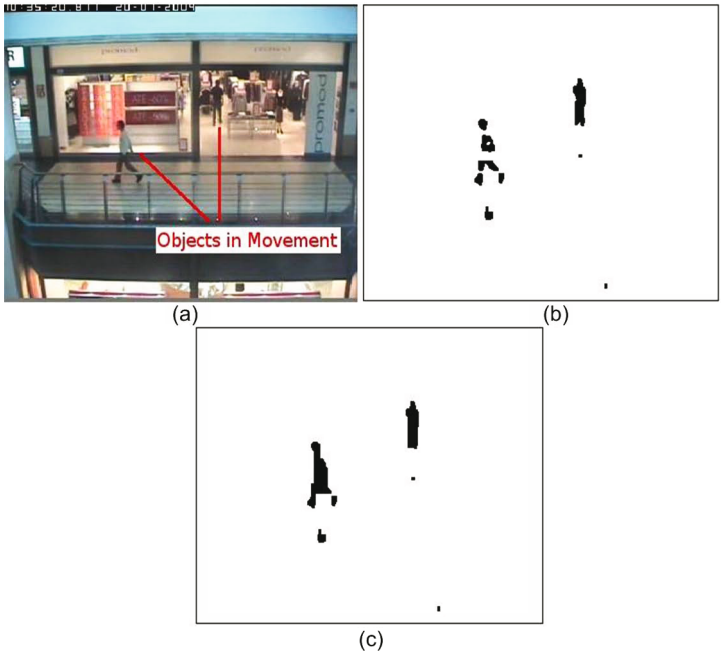

Fig. 2. Motion detection and segmentation in a video clip from CAVIAR Database: (a) original video frame with objects in motion, (b) motion segmentation by background subtraction, (c) resulting *blobs* after filtering, background subtraction and morphological operations

Figure 2 shows in a simplified manner the detection and segmentation of foreground objects in the scene. Once a blob is identified, it must be tracked while it is presented in the camera field of view.

2.2 Tracking Objects

The tracking consists of evaluating the trajectory of the object in motion while it remains in the scene. To eliminate objects that are not interesting under the point of view of the tracking, it is applied a size filter which discards blobs that are not consistent with the expected width and height of the objects of interest. The idea of using filters to eliminate undesirable regions was proposed by Lei e Xu [9]. Filtering takes into account the velocity and direction of the motion applied to a cost function. The tracking of the objects in the scene and the prediction of its position in the scene is done by an approach proposed by Latecki and Miezianko [10] with some modifications. Suppose an object O^i in the frame F^n, where O^i denotes a tracking object. In the next frame F^{n+1}, given j regions of motion, R^j, we have to know which R^j represents the object O^i from the preceding frame. The following cost function is used:

$$Cost = (w_P * d_P) + (w_S * d_S) + (w_D * d_D) + d_T \tag{1}$$

where w_P,w_S,e w_D are weights that sum to one and d_P is the Euclidean distance in pixels between the object centers, d_S is the cost of the size difference, d_D is the cost of the direction difference between the object position estimated by the Lucas-Kanade algorithm [11] in the current frame and the difference between the center of the region of motion and the center of the object, and d_T is the time to live (TTL) of the object. These parameters are better described as follows.

$$d_P = |R_c^j - O_c^i| \tag{2}$$

where R_c^j is the center of the region of motion and O_c^i is the last known center of the object. The value of d_P should not be higher than a threshold of proximity measured in pixels. This proximity threshold varies according to the objects are being tracked, mainly due to the speed of such objects in the scene.

$$d_S = \frac{|R_r^j - O_r^i|}{R_r^j + O_r^i} \tag{3}$$

where R_r^j and O_r^i denote the size of the region of motion and the size of the object respectively.

$$d_D = |arctan(O_{lkc}^i) - arctan(R_c^j - O_c^i)| \tag{4}$$

where the value of the angle lies between zero and 2π. O_{lkc}^i is the object position estimated by the Lucas-Kanade algorithm.

$$d_T = \frac{(TTL_{MAX} - O_{TTL}^i)}{TTL_{MAX}} \tag{5}$$

where TTL_{MAX} is the maximum persistence in frames and O_{TTL}^i is the object persistence . If the object is found in the current frame, the value of O_{TTL}^i is set to TTL_{MAX}, otherwise it is decreased by one until O_{TTL}^i becomes equal zero, where the object must be eliminated from the tracking.

Each object from the preceding frame must be absorbed by the region of motion in the current frame that leads to the lowest cost. The values of the object and bounding box centers assume the values of the regions of motion. If there is a region of motion that was not assigned to any object, then a new object is created with the values of such a region. If there is an object that was not assigned to any region of motion, such an object may be occluded and the *Lucas-Kanade* algorithm will fail to predict the corresponding motion. In this case, the motion of such an object is predicted as:

$$O_s^i = S * O_s^i + (1 - S) * (R_c^j - O_c^i) \tag{6}$$

where S is a fixed value of the speed. The region of motion R_c^j, should be the closest region to the object, respecting the proximity threshold. Then, the new position of the object and its bounding box is computed by Equation 7 and 8.

$$O_c^i = O_c^i + O_s^i \tag{7}$$

$$O_r^i = O_r^i + O_s^i \tag{8}$$

3 People Counting

The counting algorithm starts to analyze the segmented objects (blobs) when they enter in a counting zone (Figure 3). Two strategies were proposed to tackle the problem of estimating the number of persons enclosed by the blobs. The first one employs two thresholds that are learned from the width and area of the blobs, that is, the average width of blobs enclosing one person as well as the average area of the head region of blobs. A person is added to the counting when the analyzed blob does not have a width greater than the width threshold. Otherwise, additional information based on the blob head region area is used to estimate the number of persons enclosed by the blob. To such an aim the average head area region from a blob enclosing single persons is estimated through the analysis of objects in motion in the video frames and it is further used as a reference value. The head region is computed by splitting the blob height into four zones. Thus, the head region is considered as the first one piece at the top of the blob, as shown in Figure 3. The area of the head region is obtained by counting the number of foreground pixels. The process consists of, given a blob into the counting zone, extracting its head region area and divide it by the one person head region reference area. The outcome value, denoted as v, is used to decide the number of persons enclosed by the blob as shown in Equation 9.

$$count = \begin{cases} count + 2, \text{ if } v < 2 \\ count + 3, \text{ if } v \geq 2 \end{cases} \tag{9}$$

where *count* is the variable which stores the number of persons.

In the second strategy, a zoning scheme divides the head region into ten vertical zones of equal size. From each vertical zone is computed the number of foreground pixels divided by the subregion total area. The features extracted from the zoning scheme plus the width of the head region are used to make up an 11-dimensional feature vector. The feature vectors generated from the objects in motion are further matched against reference feature vectors stored in a database by using a non-parametric classifier.

The second strategy has two steps: training and classification. At the training step, a database with reference feature vectors is build from the analysis of blobs in motion in the frames of videos. Each reference feature vector receives a label to indicate if it encloses one, two, or three persons. The labels are assigned by human operators at the calibration/training step. At the end, the database is composed by Z reference feature vectors representing all possible classes (one, two or three persons).

The classification consists of, given a blob in motion, a set $\mathcal{V}$ of feature vectors is extracted. The number of vectors in the $\mathcal{V}$ set varies according to the period of

Fig. 3. Key regions to the counting system

time the blob takes to pass through the counting zone, which characterizes a size variant temporal window. The classification process is composed by two stages: first, each $V_i \in \mathcal{V}$ is classified using an instance-based approach, more specifically the k nearest neighbor algorithm (k-NN) [12]; next, the majority voting rule is applied to the feature vectors in $\mathcal{V}$ to come up to a final decision.

For the k-NN algorithm, the Euclidean distance among each feature vector in $\mathcal{V}$ and the Z reference feature vectors stored in the database is computed. The Euclidean distance between a D-dimensional reference feature vector V_z and a testing feature vector V_i is defined in Equation 10.

$$d(V_z, V_i) = \sqrt{\sum_{d=1}^{D} (V_{z\,d} - V_{i\,d})^2} \tag{10}$$

The k-closest reference feature vectors will label each feature vector in $\mathcal{V}$ with their labels. After the classification of all feature vectors in $\mathcal{V}$, a final decision is given by the vote of each member of the set $\mathcal{V}$, and the classification "one", "two" or "three" is assigned to the object in motion according to the majority vote rule. For example, if there are in seven feature vectors in $\mathcal{V}$ classified by the k-NN as "one person" and two classified as "two persons", the final decision is to assign the label "one-person" to the blob in motion.

4 Experimental Results

The proposed approach was evaluated in two different databases. The first database consists of videos clips where people walk alone and in groups through the field of view of a CCTV camera positioned in a oblique view. These videos were gathered at a university entrance hall through a security camera installed at

second floor of the building and without any control of illumination and background. The second database is made up of some videos available in the CAVIAR database [1]. One of the goals of this experiment is to evaluate the adaptability of the proposed approach to different scenarios. CAVIAR video clips were captured with a wide angle camera lens in front of a Mall store. Both databases have half-resolution PAL standard (320 x 240 pixels, 25 frames per second) video which are compressed using MPEG2. About one minute of video from each database was used for calibration and training purposes, that is, to generate the reference feature vectors and adjust some parameters. The remaining video were used for testing, that is, for evaluating the counting algorithms. The number of samples for each class (one, two or three persons enclosed by a blob) is shown in Table 1. Table 2 shows the total number of persons who passed through the virtual counting zone in each environment. These numbers were manually obtained through human interaction with the videos.

Table 1. Number of samples in each class at CAVIAR Mall and University Hall databases

Class (Number of Persons)	CAVIAR Mall Database	University Hall Database
one	49	45
two	14	20
three	4	11

Table 2. Total number of persons that cross the counting zones in the videos clips used in the tests

Database	Total Number of Persons
CAVIAR Mall	92
University Hall	128

The weights and thresholds described in Section 2.2 were empirically defined. This is called the calibration procedure. The d_P proximity threshold was set to 40, TTL_{MAX} to 45, S to 0.9 and the values of the weights w_P, w_S, w_D to 0.4, 0.1 and 0.5 respectively. About one minute of video was used for calibration. The same values were uses for both databases.

First, we have evaluated the total amount provided by the automatic counting, where the manual process is compared with the automatic results of each approach. Table 3 presents such a comparison. In order to demonstrate the necessity of an approach which treats the people proximity, the outcome of tracking algorithm is also presented, because it does not account for any treatment to this problem.

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

Table 3. Automatic counting outcome achieved by different strategies on both databases

Database	Tracking	Threshold Based	Head Region Analysis 11-nn	5-nn	1-nn
CAVIAR	74 (80.43%)	81 (88.04%)	92 (100%)	93 (98.92%)	91 (98.91%)
University	94 (73.74%)	155 (82.58%)	142 (90.14%)	149 (85.91%)	157 (81.53%)

Next, we have evaluated the behavior of each approach when dealing with the proximity of persons in the scene. The main goal of these experiments is to evaluate whether the approaches are able to estimate the correct amount of persons in groups or not. Tables 4 and 5 show the confusion matrices for the threshold-based approach for the CAVIAR and University databases respectively, while Tables 6 and 7 present the confusion matrices for the head region analysis approach for the CAVIAR and University databases respectively.

Table 4. Confusion matrix for the threshold-based approach on the CAVIAR videos

Class	one	two	three	Correct Counting Rate
one	**44**	5	0	89.79%
two	8	**6**	0	42.85%
three	0	2	**2**	50%
			Average	77.61%

Table 5. Confusion matrix for the threshold-based approach on the University Hall videos

Class	one	two	three	Correct Counting Rate
one	**33**	7	5	73.33%
two	5	**9**	6	45%
three	0	5	**6**	54.54%
			Average	63.15%

Both approaches achieved encouraging results when just the final counting is observed (Table 3). However, when the way to obtain these results is detailed, it can be seen that the head region analysis has got reliable results in comparison with the threshold-based approach.

An error analysis has shown the two main sources of errors in the proposed people counting method. First is related to the background/foreground segmentation process, mainly due to the clothes colors that have been confused with the background. Second is related to the tracking process, especially in the presence

Table 6. Confusion matrix for the head region analysis approach on the CAVIAR videos

Class	one	two	three	Correct Counting Rate
one	**49**	0	0	100%
two	3	**9**	2	64.28%
three	0	2	**2**	50%
			Average	89.55%

Table 7. Confusion matrix for the head region analysis approach on the University Hall videos

Class	one	two	three	Correct Counting Rate
one	**43**	2	0	95.55%
two	5	**14**	1	70%
three	0	7	**4**	36.36%
			Average	80.26%

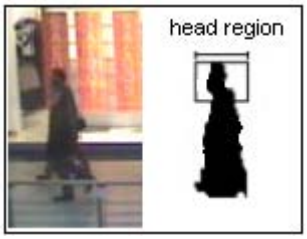

Fig. 4. Error caused by occlusion: both strategies have count one person instead of two

of occlusions. Figure 4 shows an example in which both strategies used for people counting have failed since one of the two individuals in the scene is almost totally occluded by the other.

5 Conclusion

In this paper we have presented an approach for counting people that pass through a virtual counting zone and which are gathered by a general purpose CCTV camera.

One of the proposed strategies is able to count with a relative accuracy the number of persons even when they are very close or when they are in groups. Such a strategy is able to classify the number of persons enclosed into a blob. The use of this simple approach has lead to satisfactory results and the classification of the number of people in a group was carried out in real-time due to the simplicity

of the classification technique and the low dimensionality of the feature vectors used to discriminate the number of persons.

In spite of the preliminary results are very encouraging, since we have achieved correct counting rates up to 85% on video clips captured in a non-controlled environment, further improvements are required, specially in the first steps of the proposed approach which includes more sophisticated and reliable segmentation and tracking algorithms.

Compared with other works that deal with counting persons through a non-dedicated camera, the proposed approach is more simple but it has achieved similar correct counting rate. However, at this moment a direct comparison is not possible due to different experimental setups and databases used in the tests. Furthermore, a large scale test on a number of different environments is also necessary.

References

1. Kim, J.-W., Choi, K.-S., Choi, B.-D., Ko, S.-J.: Real-time Vision-based People Counting System for the Security Door. In: Proc. of 2002 International Technical Conference On Circuits Systems Computers and Communications, Phuket (July 2002)
2. Snidaro, L., Micheloni, C., Chiavedale, C.: Video security for ambient intelligence. IEEE Transactions on Systems, Man and Cybernetics PART A 35(1), 133–144 (2005)
3. Nakajima, C., Pontil, M., Heisele, B., Poggio, T.: People recognition in image sequences by supervised learning. In: MIT AI Memo (2000)
4. Gavrila, D.: Pedestrian detection from a moving vehicle. In: Proc. 6th European Conf. Computer Vision, Dublin, Ireland, vol. 2, pp. 37–49 (2000)
5. Giebel, J., Gavrila, D.M., Schnorr, C.: A bayesian framework for multi-cue 3d object tracking. In: Proc. 8th European Conf. Computer Vision, pp. 241–252. Prague, Czech Republic (2004)
6. Lin, S-F., Chen, J-Y., Chao, H-X.: Estimation of Number of People in Crowded Scenes Using Perspective Transformation. IEEE Trans. Systems, Man, and Cybernetics Part A 31(6), 645–654 (2001)
7. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-Time Surveillance of People and Their Activities. IEEE Transactions on Pattern Analysis and Machine Intelligence, 809–830 (2000)
8. Hu, W., Tan, T., Wang, L., Maybank, S.J.: A Survey on Visual Surveillance of Object Motion and Behaviors. IEEE Trans. Systems, Man, Cybernetics, Part C, 334–352 (2004)
9. Lei, B., Xu, L.Q.: From Pixels to Objects and Trajectories: A generic real-time outdoor video surveillance system. In: IEE Intl Symp Imaging for Crime Detection and Prevention, pp. 117–122 (2005)
10. Latecki, L.J., Miezianko, R.: Object Tracking with Dynamic Template Update and Occlusion Detection. In: 18th Intl Conf on Pattern Recognition, pp. 556–560 (2006)
11. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: 7th Intl Joint Conf Artificial Intelligence, pp. 674–679 (1981)
12. Aha, D.W., Kibler, D., Albert, M.K.: Instance-Based Learning Algorithms. Machine Learning 6, 37–66 (1991)