Path Analysis in Multiple-Target Video Sequences

Brais Cancela, Marcos Ortega, Alba Fernández, and Manuel G. Penedo

Varpa Group, Department of Computer Science
University of A Coruña, Spain
{brais.cancela,mortega,alba.fernandez,mgpenedo}@udc.es

Abstract. Path analysis becomes a powerful tool when dealing with behavior analysis, i. e., detecting abnormal movements. In a multiple target scenario it is complicated to obtain each object path because of collision events, such as grouping and splitting targets, and occlusions, both total or partial. In this work, a method to obtain the similarity between different trajectories is presented, based in register techniques. In addition, an hierarchical architecture is used to obtain the corresponding paths of the objects in a scene, to cope with collision events. Experimental results show promising results in path analysis, enabling it to establish thresholds to abnormal path detection.

1 Introduction

In the security field, path analysis is a powerful tool for detection of abnormal behavior in a given scenario. An strange movement of an object could result as an abnormal behavior, which should be detected by a surveillance system. Typically, this kind of scenarios comprise series of individuals (or objects in general) crossing the scene simultaneously. In order to obtain a good approach for the path associated with each moving object, a robust target tracking must be implemented able of dealing with multiple objects. This involves target detection, classification and collision processing (grouping, splitting, leaving and entering the scene or partial occlusions). This is necessary because the system cannot make a mistake identifying each object. An error identifying a moving object could result in a wrong associated path, making the comparison useless.

In this work we present a methodology for path analysis under multiple-target condition, establishing an adequate framework for higher-level applications that require behavior analysis, such as video surveillance. This paper is organized as follows: section 2 discusses the state-of-the-art in this field and related work; section 3 describes our proposed multiple-target tracking method, whereas section 4 describes how to model each object path and the method used to determine the similarity between two of them; section 5 shows the method results and section 6 offers conclusions and future work.

2 Related Work

There are in the literature many different approaches for path analysis. One of the earliest methods uses a Self-Organizing Feature Map [1]. They encode every path with a

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part II, LNCS 6979, pp. 50–59, 2011.

fixed length, which is introduced into the SOM to classify the trajectory between normal and abnormal. It cannot cope with multiple targets, and needs to train the system with a priori knowledge. In [2], Hidden Markov Models are used to define each path. Orientation is also introduced. It track multiple objects around a scene, but it does not deal with collision events. A log-likelihood function is used to classify each event. It is also used by Rao et al. [3], who use a probabilistic model to define model trajectories, but it only detects one object in each scene. Each feature vectors of a trajectory is assumed to have Markovian dependence rather than being independent. Calderara et al. [4] use a mixture of Von Mises distributions created from an unsupervised training set. No more than one one is detected under this technique.

Our approach to this topic is based in the register techniques to determine path similarities. [5, 6]. Each path is shown as a binary image, which is compared against others using restricted transformations to obtain the best fit. In addition, a multiple-target tracking based in an hierarchical architecture is performed in order to deal with the occlusion problem. None of the methods explained before can handle with this issue.

Our goal is to obtain an architecture which can enable us to to establish thresholds to abnormal path detection. This system must be robust against multiple-tracking main problems: grouping, splitting and total or partial occlusion events. Our methodology works as follows: first, a multiple-target tracking method is developed to distinguish every object in the scene. Second, a method to detect each object path is presented, based in the tracking method properties. Finally, each path is classified into normal or abnormal using a restricted register technique.

3 Multiple-Target Tracking

In order to cope with the problems of multiple-target tracking mentioned before, our system includes two different ways to detect an object: first, a low-level tracking is used to detect each object into the scene; second, a high-level representation must be implemented so that it could identify every moving object under different collision problems. Therefore, our method comprises a hierarchical architecture [7]. However, different changes are introduced to the original hierarchical model in order to further improve the accuracy.

First, a blob detection is performed to detect every moving object in the scene. A background subtraction method based in Mixture of Gaussians [8] (MoG) is used. Each background pixel is modeled as a bunch of five gaussians. Pixels which values do not fit under any background distribution are marked as foreground until there is a Gaussian distribution that includes them with enough evidence. Background model is updated with every new frame using a training parameter, which is the learning rate. To prevent the algorithm from stopped foreground pixels, which could be considered as background because of the updating system, the training parameter is set to a very low value. A requirement is that an only-background sequence is needed to train the algorithm.

After the background subtraction method is applied, the image is divided into foreground and background pixels. To fill the blobs and to avoid small regions holes or noise due to camera movement or video compression, opening and closing morphological operations and a minimum-area filter are applied when detecting blob regions.

Once detected, an ellipse representation for each blob is used, because it is really useful for dealing with collision problems, since it is a simple geometry which fits better with the object than polygons. Therefore, the j-observed blob at time t is given by $z_j^t = (x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t)$, where x_j^t, y_j^t represent the ellipse centroid, h_j^t, w_j^t are the major and minor axes, and θ_j^t the ellipse orientation. Fig. 1 shows an example of this methodology.

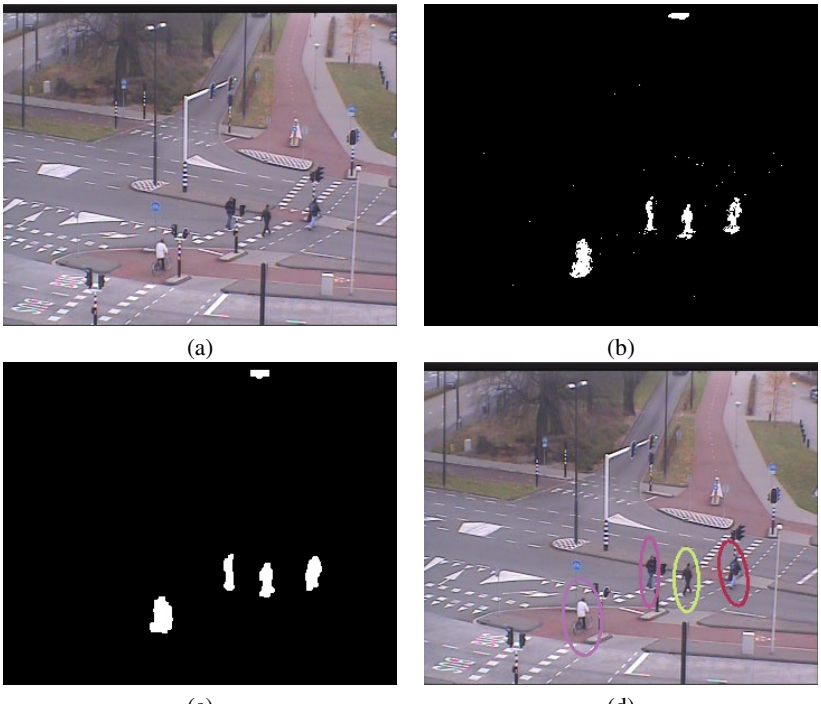

Fig. 1. (a) Frame. (b) MoG foreground detection. (c) Morphological operators. (d) Blob detection after applying minimum-area filter.

To track each moving object into successive frames, our approach stores the position of the ellipse in a window of size M. Subsequently, a median filter is used in order to smooth the values and a set of adaptive filters (Adalines) predict the velocity of each ellipse parameter. Adding this velocity to the previous position, a new predicted position is computed.

To match each ellipse with the appropriate tracking object, we look at the centroid point of the tracking object. Since we assumed that every object in the frame moves slowly enough compared to the frame rate, if the ellipse centroid is included into the ellipse of the new predicted position, and it is the only one, the tracking object is matched with the new ellipse and the low level tracking is confirmed. If two or more ellipse centroids are within the predicted position ellipse, then a splitting event is created. On

the other hand, if two or more predicted position centroids are within a new ellipse, a grouping event is created.

To have the objects in the scene identified we propose an appearance model. This is needed because the low-level tracking has problems in cases of grouping and occlusion, since even although it can detect when a grouping or a splitting event occurs, it cannot detect which of the new blobs corresponds with a particular target stored into the group. As we mentioned before, a color representation of each target is computed. We select five different fixed color histograms, all of them into L*a*b space color. The histograms selected are the following:

$$\omega_1 * L + \omega_2 * a + \omega_3 * b, (\omega_1, \omega_2, \omega_3) \in \{(1,0,0); (0,1,0); (0,0,1); (0,1,1); (0,1,-1)\},$$

so that the histograms stored in each tracking object correspond with the appearance average of all objects along the sequence identified as the tracking object. This is needed because of light changes or orientation and position changes in the tracking object. With L*a*b color space we can isolate illumination into one component, which is useful to make this method invariant to fast illumination changes.

As we mentioned before, low-level tracking can detect grouping and splitting events, which are evaluated by the Event Management module. It defines six different states: single target, target grouping, grouped, splitting, split and occluded. When a grouping or splitting event occurs, if it is confirmed in the next frame, its state is changed to grouped or split, respectively. When, after t frames, typically the frame rate, a target does not appear, its state is changed to occluded. When it occurs, low-level tracking is not useful because the position is unpredictable after a long time.

When a low-level tracking is trained (detection after six consecutive frames), the Tracking Object Matching module is activated. After it is activated, every time the low-level tracker is confirmed, a feature comparison is computed to guarantee the low-level tracker confirmation is correct. If this module confirms the match, its state is updated with the new position and color histograms are updated. If no observation is associated to a particular target, its state is set using the previously predicted state.

When a target is marked as *occluded*, it means that low-level tracker is unused, so we only make an appearance comparison to locate the object. These trackers have lower priority than the others, meaning that they can only be compared when the rest of trackers failed.

If there is no matching between the low-level comparison in a long-duration occlusion, the tracker is marked as *occluded*. This means that we only make an appearance comparison to locate the object. However, these trackers have lower priority than the others, meaning that they can only be compared when the rest of trackers failed.

To make the appearance computation, we normalize and quantify each histogram into 64 bins to improve process velocity, which is high enough to prevent wrong matchings [7]. Thus, the probability of each feature is calculated as:

$$p_k^i = C^i \sum_{a=1}^{M} \delta(b(x_a) - k), \tag{1}$$

where C_i is a normalization constant which ensures $\sum_{k=1}^{64} p_k^i = 1$, δ is the Kronecker delta, $\{x_a; a = 1 : M\}$ represent the pixel locations, M is the number of target pixels, and $b(x_a)$ is a function that associates pixels to their corresponding bins.

Then, the similarity between two histograms is computed using the *Hellinger distance*, $d_H = \sum_{k=1}^{64} \sqrt{p_k q_k}$, where q_k are the bins stored in the tracked object. Similarity criterion acceptance uses both mean and standard deviation of the Hellinger Distance in all of the different matches, and they are updated every new match occurs. When at least 60% of the comparisons pass the test (three out of five histograms) we accept the match. If there is no match with any of the tracking objects, a new one is instantiated and trained. It does not mean that the new tracker could have never appeared before, so, once this is trained, it is compared against other *occluded* trackers, because it could be one of the previously defined. If this is the case, the tracker is merged with the previous one. Fig. 2 shows an example of a case in which it produces both grouping and splitting events. The method can keep correctly the identification of all the moving objects.

Finally, a tracker is deleted if it is lost before it is trained or if the number of times being present is much lower than the number of frames since it appeared for the first time.

(a) (b) (c)

Fig. 2. (a) Tracking before grouping. (b) Tracking during grouping. (c) Tracking after splitting event. The algorithm detects every object and can match all the objects in a correct way.

4 Path Analysis

As mentioned earlier, one of the most common uses for object tracking is the analysis of its behavior in the scene. This can be human actions, object interactions, etc. In this work we focus on path analysis to model object behavior. This can lead, for example, to the detection of suspicious trajectories in controlled environments. In our case, the detection of every object movement could define the most usual tracks around the scene.

Our model for path analysis comprises two step: a) Object path modeling and b) Path comparison methodology.

To model these trajectories, we use the properties defined in the Multiple-Target Tracking algorithm. As we mentioned before, this method needs to be able to distinguish each object among the others. Although the low-level tracker is unused when the moving object is lost, it will be used to obtain the path associated to each one. As each object is located within the scene by an ellipse, we only need to store its ellipse centroid

in every frame to store its path on the scene. So the j-observed object path at time t is represented as

$$\mathcal{P}_j^t = \{(x_j^\tau, y_j^\tau) \in \mathbb{R}^2 \ \mid \ \tau = 0..t\} \tag{2}$$

Fig. 3 shows an example of different paths found in a scene. When a group of several objects is formed in a scene, a solution for their particular paths definition must be obtained. The intuitive notion of assigning the group ellipse centroid to all of the objects turns to be a bad idea when the ellipse is much greater than the individual ones as seen in Fig. 2-b. This is because the real location of the object is significantly displaced respect to the ellipse centroid. We propose a simple solution to overcome this problem in most of the situations. No positions are stored for particular objects while in a group. When the group is split, and some object is located as an individual again, the original path of the object is linked to the current one by linear interpolation. This solution is adequate assuming group path is coherent respect to the objects, which is true in the vast majority of cases.

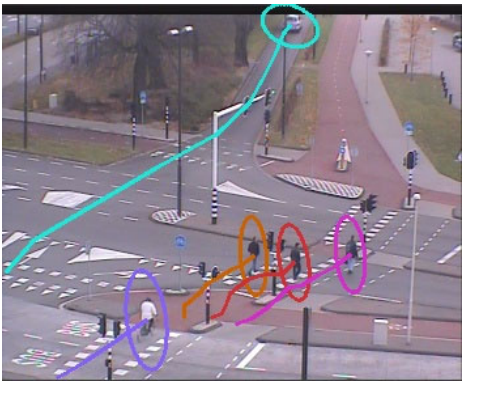

Fig. 3. Using the Multiple-Tracking framework, the path associated to each moving object involves the ellipse centroid, which represents the position over each time step.

Once the paths has been modeled, they can be compared and, therefore, a similarity measure must be defined to detect abnormal paths. In [9], an abnormal activity (abnormal trajectory in our case) is defined as a change in the system model, which could be slow or drastic, and whose parameters are unknown. In motion objects, there are a wide range of different situations that could be perceived as abnormal behavior, such as sudden changes in the speed or the direction. It could also be regarded as an abnormal situation when an object is stopped in its normal path [10].

In our case, our aim is to detect abnormal object movements with respect to the others, i. e., objects which path highly differs from the usual ones. Our first approach does not take into account neither orientation nor speed, but could be easily introduced in the model as this information is already known by the training system. Once the object has left the scene, its path is compared to determine if it may be regarded as an

abnormal activity. Once we have all the points which determine the path of an object, a binary image is made, setting the path in white, as shown in Fig. 4.

To compare the path image against the others, we use a register technique, a method used in other fields such as biometrics, where templates are compared to each other to determine similarities. [5, 6]. The idea is to compare two templates, one used as reference and other that can be modified using geometric transformations in order to maximize the similarity by aligning the templates. In our case, the possible transformations is very limited and, therefore, we use a set of affine transformations allowing for small translations, scales and rotations. In order to measure the similarity between two aligned images, we use the normalized cross-correlation:

$$\mathcal{R}(x,y) = \frac{\sum_{x',y'}(T(x',y') \cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}, \tag{3}$$

where T is the reference template and I the one which can be modified.

5 Experimental Results

In our experiments we have used the CANDELA Intersection Scenarios [11] in order to test the methodology. Overall, we use over two minutes video recording at 25 frames per second. These videos take place outdoors. Partial occlusions, grouping and splitting events are frequent in these sequences making it a very suitable and challenging scenario to test our method.

To perform the abnormal activity detection, a few restrictions are inserted to guarantee a low rate of bad matches. First, we only evaluate $\mathcal{R}(0,0)$. This is because the nature of the path. For instance, a person who walks in the top of the image is not the same than other person walking in the bottom. This implies that we have to restrict the movement of the image against the template as much as it possible.

Second, several paths which are close together in a parallel way could be matched as the same track. To solve this situation, also improving the performance, we reduce the size of the image by a correction factor. This helps us to obtain more accurate results. To determine the correction factor value, we choose an exponential manner with powers of two. The value we use is the closest to the mean height of all the objects detected. Finally, restricted affine transformations are used. This includes rotations between -10 and 10 degrees and translations, both in x and y, between -1 and 1. A threshold is used to define if two path are the same or are different.

In Fig. 4 we can see an example of this methodology. Two different cars drive in the same road while one person is walking near them. Path images of both cars are close, while the other remains quite different. Using the restrictions explained before, we obtain a similarity value of 0.707776, while the similarity between the cars and the person is lower than 0.13, allowing to establish a confidence band to select a similarity threshold to consider two paths the same one in our domain.

Over the videos used to test this system, thirty one different paths were stored and compared. Fig. 5-a shows a bar plot containing cross-correlation values in two different classes, those which are the same path under our criteria and those which are different.

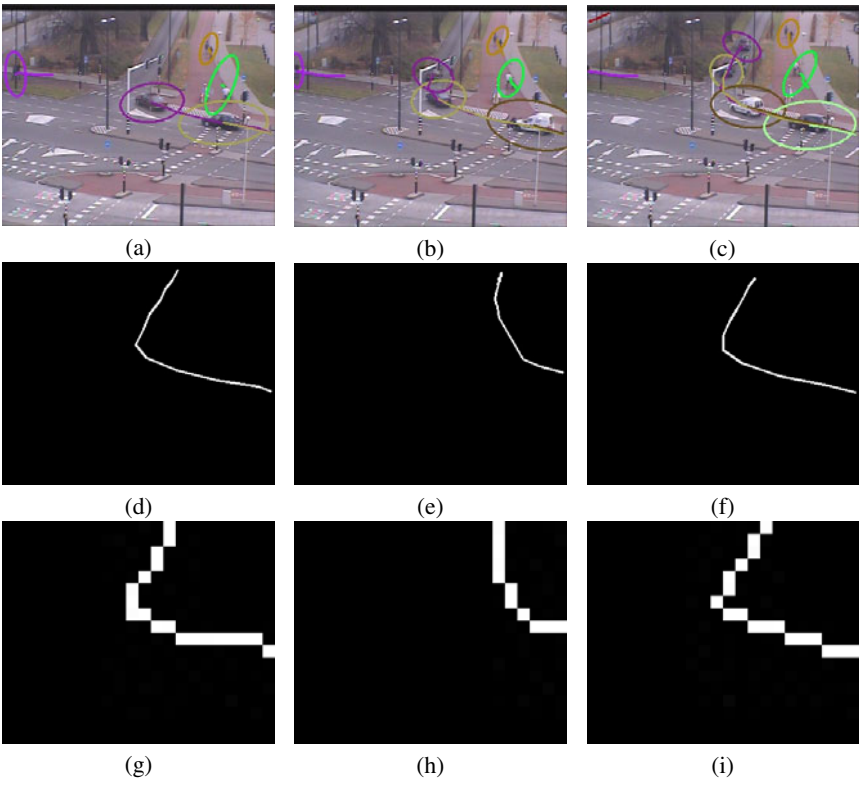

Fig. 4. Ellipse paths after a sequence. (a, b, c) frame sequence. (d) purple ellipse path template. (e) green ellipse path template. (f) ocher ellipse path template. (g) purple ellipse path template reduced by a correction factor. (h) green ellipse path template reduced by a correction factor. (i) ocher ellipse path template reduced by a correction factor.

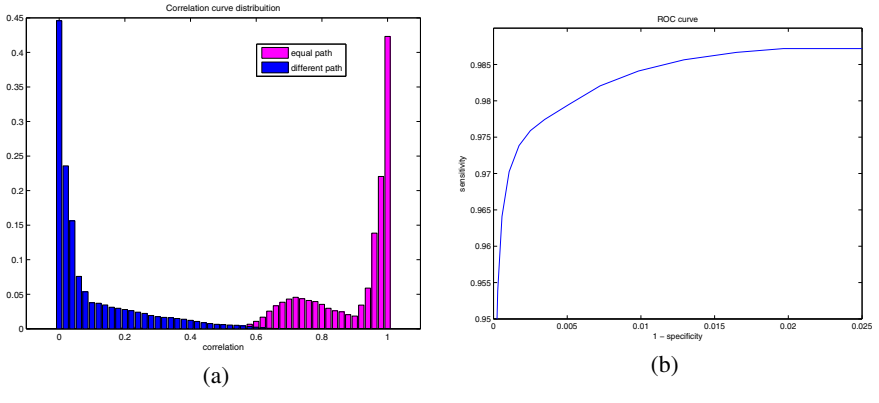

Fig. 5. (a) Correlation curve distribution. (b) ROC curve. Equal path and different path values are highly separated.

These distributions are highly separated, so we can obtain very good results. In Fig. 5-b we can see the ROC curve, which is close to the Heaviside step function. In fact, choosing the threshold value 0.56 we obtain the results showed in Table 1, which demonstrate the validity of the methodology for path analysis.

Table 1. Sensitivity and specificity of the method using the threshold value 0.56. Results show the accuracy of the methodology.

	Value
Sensitivity	98.75%
Specificity	98%

6 Conclusions

In this paper a new approach to the path detection for multiple object tracking is presented using an hierarchical architecture. Low-level tracker properties are used in order to obtain the path followed by every moving object detected in the scene. An abnormal path detection is used, based in register techniques with restrictions, obtaining promising results.

A multiple-tracking object is developed to operate under collision events, like grouping, splitting or total and partial occlusions. A register-based technique is used to obtain the similarity between trajectories, which allow us to establish thresholds to abnormal paths detection, obtaining over a 98.5% of success.

In a future research a dynamic methodology would be interesting to enable path comparisons even if the moving object have not left the scene. A methodology to detect different paths because of the different direction of the movement will be developed.

Acknowledgments. This paper has been partly funded by the Consellería de Industria. Xunta de Galicia through grant contracts 10/CSA918054PR and 10TIC009CT.

References

1. Owens, J., Hunter, A.: Application of the self-organising map to trajectory classification. In: Proceedings of Third IEEE International Workshop on Visual Surveillance (2000)
2. Jiang, F., Wu, Y., Katsaggelos, A.K.: Abnormal event detection from surveillance video by dynamic hierarchical clustering. In: IEEE International Conference on Image Processing, ICIP 2007, vol. 16 (2007)
3. Rao, S., Sastry, P.S.: Abnormal activity detection in video sequences using learnt probability densities. In: TENCON 2003, Conference on Convergent Technologies for Asia-Pacific Region, vol. 1, pp. 369–372 (2003)
4. Calderara, S., Cucchiara, R., Prati, A.: Detection of abnormal behaviors using a mixture of von mises distributions. In: IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007, pp. 141–146 (2007)

5. Mariño, C., Penedo, M.G., Penas, M., Carreira, M.J., González, F.: Personal authentication using digital retinal images. Pattern Analysis and Applications 9, 21–33 (2006)
6. Mariño, C., Ortego, M., Barreira, N., Penodo, M.G., Carreira, M.J., González, F.: Algorithm for registration of full scanning laser ophtalmoscope video sequences. Computer Methods and Programs in Biomedicine 102(1), 1–16 (2011)
7. Rowe, D., Reid, I., Gonzàlez, J., Villanueva, J.J.: Unconstrained multiple-people tracking. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 505–514. Springer, Heidelberg (2006)
8. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252 (1999)
9. Vaswani, N., Chowdhury, A.R., Chellappa, R.: "shape activity": A continuous state hmm for moving/deforming shapes with application to abnormal activity detection. IEEE Transactions on Image Processing 14(10), 1603–1616 (2005)
10. Grimson, W., Lee, L., Romano, R., Stauffer, C.: Using adaptive tracking to classify and monitor activities in a site. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Santa Barbara, CA, pp. 22–31 (1998)
11. CANDELA: content analysis and networked delivery architectures, `http://www.multitel.be/~va/candela/`