USING CALIBRATED CAMERA FOR EUCLIDEAN PATH MODELING

Imran N. Junejo

University of Central Florida,
Orlando, FL 32816, U.S.A.

Hassan Foroosh

University of Central Florida,
Orlando, FL 32816, U.S.A.

ABSTRACT

In this paper, we address the issue of Euclidean path modeling in a single camera for activity monitoring in a multi-camera video surveillance system. The paper proposes to use calibrated cameras to detect unusual object behavior. During the unsupervised training phase, after metric rectifying the input trajectories, the input sequences are registered to the satellite imagery and prototype path models are constructed. During the testing phase, using our simple yet efficient similarity measures, we seek a relation between the input trajectories derived from a sequence and the prototype path models. Real-world pedestrian sequences are used to demonstrate the practicality of the proposed method.

Index Terms— Machine Vision, Camera Calibration, Euclidean Path Modeling, Image Registration.

1. INTRODUCTION

There is a growing interest in analyzing objects such as cars and pedestrians using computer vision systems. This is mainly due to the fact the computer vision research has advanced to systems that can accurately detect, recognize and track object as they move through a scene. Based on this, it is possible for other systems to make higher level inferences. In path modeling and surveillance, our purpose is to build a system that, once given an acceptable set of trajectories of objects in a scene, is able to build a path model. We aim to learn the routes or paths most commonly taken by objects as they traverse through a scene. Once we have a model for the scene, the method should be able to classify incoming trajectories as conforming to our model or not.

Once the objects of interest (e.g. pedestrians) are successfully detected and tracked, using Javed et al. [1], the first step is the removal of projective distortion from the object trajectories. Thus upgrading them to Euclidean trajectories. This is only possible once the camera is calibrated and we do this by adopting the approach presented by [**?**] for its simplicity and ease of use.

Once the object trajectories are metric rectified, a path model for the scene is constructed. Grimson et al. [2] records object parameters like the position, direction of motion, velocity size and aspect ratio of each connected region which are then used to classify the objects. Boyd et al. [3] demonstrate the use of network tomography for statistical tracking of activities in a video sequence. Johnson et al. [4] use a neural network to model the trajectory distribution for event recognition and prediction. Recently [5] uses the 3D structure tensor for representing global patterns of local motion. The most related work is that of Makris and Ellis [6] where they develop a spatial model to represent the routes in an image and use a simple distance measure to check the validity of a test trajectory. One limitation of this approach is that *only* spatial information is used for trajectory clustering and behavior recognition. The system cannot distinguish between a person walking and a person lingering around, or between a running and a walking person. There exist no stopping criteria for merging of routes.

During our *training phase*, assuming that the camera has already been calibrated, object trajectories are rectified, clustered and meaningful features are extracted to build a path model, described in Section 2. During the *test phase*, this path model is used to test the validity of the an incoming trajectory (cf Section 3). We present results on real data in Section 4 before conclusion 5.

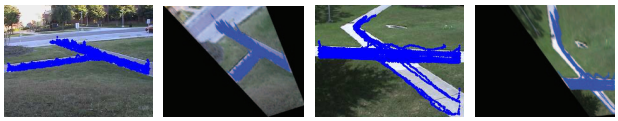
2. TRAINING PHASE

2.1. Trajectory and Image Rectification

Once the camera is calibrated, the object trajectories obtained in the training phase can be metric rectified, as a part of our training phase. As argued above, metric rectified trajectory data presents a truer picture of the original data. From *projective geometry*, the line at infinity $\mathbf{l}_\infty$ intersects the image of the absolute conic $\boldsymbol{\omega}$ at two complex conjugate ideal points $\mathbf{I}$ and $\mathbf{J}$, called the *circular points* [7]. The conic dual to the circular points is given as $\mathbf{C}^*_\infty = \mathbf{I}\mathbf{J}^\mathbf{T} + \mathbf{J}\mathbf{I}^\mathbf{T}$, where $\mathbf{C}^*_\infty$ is a degenerate conic consisting of two circular points. Under a point transformation, $\mathbf{C}^*_\infty$, invariant under similarity transformation, transforms as:

$$\mathbf{C}^{*'}_\infty = (\mathbf{H_P H_A})\mathbf{C}^*_\infty(\mathbf{H_P H_A})^\mathbf{T} = \begin{bmatrix} \mathbf{K}\mathbf{K}^\mathbf{T} & \mathbf{K}\mathbf{K}^\mathbf{T}\mathbf{v} \\ \mathbf{v}^\mathbf{T}\mathbf{K}\mathbf{K}^\mathbf{T} & \mathbf{v}^\mathbf{T}\mathbf{K}\mathbf{K}^\mathbf{T}\mathbf{v} \end{bmatrix} \quad (1)$$

where $\mathbf{H_P}$ and $\mathbf{H_A}$ are respectively the projective and affine components of the projective transformation. It is clear that



(a) (b) (c) (d)

Fig. 1. **Rectified Trajectories**:(b) represents reconstructed trajectories for **Seq** #2 - shown in (a), while (d) represents **Seq** #3, shown in (c), rectified.

the affine ($\mathbf{K}$) and the projective($\mathbf{v}$) components are determined directly from the image of $\mathbf{C}^*_\infty$. Once $\mathbf{C}^{*'}_\infty$ is identified, a suitable rectifying homography is obtained by using the SVD decomposition: $\mathbf{C}^{*'}_\infty = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{U}^\mathbf{T}$ where $\mathbf{U}$ is the rectifying projectivity. Fig. 1 depicts some results obtained by rectifying the obtained training trajectories from two of our three test sequences.

2.1.1. Registration to Aerial Imagery

Registration to the satellite imagery gives a global view of the scene under observation. Since we do not have the knowledge of Euclidean world coordinates of (at least) four points and we want to make the process automatic, the estimated affine and the perspective transform can be combined together to efficiently metric rectify the video sequence such that the only unknown transformation is the similarity transformation [8].

The results obtained by rectifying the test sequences are shown in Fig. 2. Due to space limitations, only rectified results from sequence **Seq # 3** are shown. The scene is rectified by using the line at infinity which is obtained as: $\mathbf{l}_\infty = \boldsymbol{\omega}\mathbf{v}_z$ [7]. The obtained circular points are used to construct the conic $\mathbf{C}^{*'}_\infty$ in order to obtain the rectifying projectivity, as described in Section 2. The rectified image is shown in Fig. 2(b), and the registered image is shown in Fig. 2(c).

Registration of multiple cameras to the satellite image is shown in Fig. 3. Three cameras were placed at three different locations along the path shown in the figure. Behavior of objects in the regions covered by the three cameras can be modeled by the proposed method and gives, in essence, the global behavior of the objects.

This registration is automated and provides satisfactory results. Thus for any test sequence, the obtained path model (to be described shortly) can be mapped to the corresponding satellite image in order to obtain a global view - representing the behavior of pedestrians in that particular area.

2.2. Model Building

Another important step during the training phase is to identifying the different paths traversed by pedestrians in a scene. This section elaborates on how the extracted trajectories are used to create a path model.

A Typical Scene: A typical scene consists of a single camera mounted on a wall or on a tripod looking at a certain location. For any object i tracked through n frames, the 2-**D** image coordinates for the trajectory obtained can be given as $\mathbf{T}_i = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

(a) (b) (c)

Fig. 2. Image rectification and registration results for **Seq # 3**, **Seq # 2** and **Seq # 1**. See text for more details.

Fig. 3. Multiple cameras registered to the corresponding satellite image: The input images have a few new structures compared to the old satellite image.

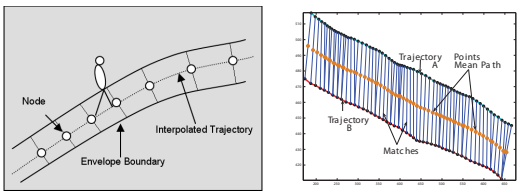
Note that the trajectories will be of varying lengths, depending on the location and velocity of the person. Since we are dealing with physical pathways where the position of an object is very important, we track the feet of the objects for more precise measurement. The trajectories obtained through the tracker are sometimes very noisy; therefore, trajectory smoothing is performed.

2.3. Trajectory Clustering

Once we have rectified trajectories from our training set, the next task to cluster the trajectories into different paths. Clustering has to be based on some kind for similarity criteria. Perceptually, humans tend to group trajectories based on their spatial proximity. Since we are trying to create a path model, it is essential that we perform clustering using the spatial characteristics of the trajectories. Thus, we choose the Hausdorff distance as our similarity measure. For two trajectories $\mathbf{T}_i$ and $\mathbf{T}_j$, the Hausdorff distance, $\mathbf{D}(\mathbf{T}_i, \mathbf{T}_j)$, is defined as $\mathbf{D}(\mathbf{T}_i, \mathbf{T}_j) = \max\{d(\mathbf{T}_i, \mathbf{T}_j), d(\mathbf{T}_j, \mathbf{T}_i))\}$, where $d(\mathbf{T}_i, \mathbf{T}_j) = \max_{a \in \mathbf{T}_i} \min_{b \in \mathbf{T}_j} \|a - b\|$.

One advantage of using Hausdorff distance is that it allows us to compare two trajectories of different lengths. In order to cluster trajectories into different paths, we formulate a complete graph. Each node of the graph represents a trajectory. The weight of each edge is determined by the Hausdorff distance between the two trajectories. Spatially proximal trajectories will have small weights because of lesser Hausdorff distance, and vice versa. The constructed complete graph needs to be partitioned; each partition having one or more trajectories corresponds to a unique path. To perform such a partition accurately and automatically, Normalized-cuts [9] are used recursively to partition the graph.

The novel usage of Normalized-cuts for trajectory clustering has certain advantages over other graph cut techniques. First, it avoids bias for partitioning out small sets of points. And second, the problem is reduced to finding the eigenvec-



(a) (b)

Fig. 4. (a) represents a typical scene where an object is traversing an existing path. (b) An example of an average trajectory obtained by applying DTW on two sample trajectories. Blue lines connect corresponding matched points between the two trajectories.

tors of the system, which is very easy to compute. This technique makes it possible to perform recursive cuts by using special properties of the eigenvectors. Fig. 5e-h shows the results obtained by clustering one of our data set.

2.4. Envelope & Mean Path Construction

At this stage, trajectories have been clustered into different paths by applying Normalized-cuts. Each path is represented by trajectories that make up that particular path. Now we create for all trajectories in each clustered path an envelope and a mean path representation. An *envelope* can be defined as the spatial extent of a path (cf. Fig. 4(a)). Applying Dynamic time Warping (DTW) algorithm [10], where column represent trajectory **A** and the row represent trajectory **B**, pair-wise correspondences between the two trajectories is determined. Using DTW, distance at each instance is given by:

$$S(i,j) = \min\{S(i-1,j-1), S(i-1,j), S(i,j-1)\} + q(i,j)$$

where the distance measure is $q(i,j) = \frac{e^{\frac{(-\kappa(i,j))}{\sigma_\kappa}} + e^{\frac{(-\overline{ij})}{\sigma_e}}}{2}$, $\overline{ij}$ represents the Euclidean distance, σ_κ represent standard deviation in spatio-temporal curvature, and σ_e represent a suitable standard deviation parameter for the trajectory (in pixels). This distance measure finds correspondences between trajectories based on spatial as well as spatio-temporal curvature similarity. By pair-wise application of the above mentioned algorithm on all trajectories of each path, (i) an envelope is created to represent the spatial extent of the path, and (ii) a mean trajectory (using DTW) to represent all trajectories in the path. For two trajectories, the mid-point of the line joining the matched corresponding points is taken as the mean path (cf. Figure 4b).

3. SCENE MODELING - TEST PHASE

A path model is developed that distinguishes between trajectories that are (a) Spatially unlike, (b) Spatially proximal but of different speeds, or (c) Spatially proximal but crooked. Once the path models are learned as described above, we extract more features from the trajectories in each path in order to verify the conformity of a candidate test trajectory.

Spatial Proximity: To verify spatial similarity, membership of the test trajectory is verified to the developed path model. All points on the candidate trajectory are compared to the envelope of the path model. The result of this process is a binary vector with 1 when a trajectory points is inside the envelope and 0 (zero) when the point is outside the envelope. This information is used to make a final decision for a candidate trajectory along with the spatio-temporal curvature measure. If all candidate trajectory points are outside the envelope, then this is an outright rejection.

Motion Characteristics: The second step is essential to discriminate between trajectories of varying motion characteristics. The trajectory whose velocity is similar to the velocity characteristics of an existing route is considered similar. Velocity for a trajectory $\mathbf{T}_i(x_i, y_i, t_i)$, $i = 0, 1, , N-1$, is calculated as:

$$\mathbf{v}'_i = (\frac{x_{i+1} - x_i}{t_{i+1} - t_i} \frac{y_{i+1} - y_i}{t_{i+1} - t_i}), i = 0, 1, \ldots, N-1$$

Mean and the standard deviation of the motion characteristics of the training trajectories are computed. A Gaussian distribution is fitted to model the velocities of the trajectories in the path model. The Mahalanobis distance measure is used to decide if the test trajectory is anomalous,

$$\tau = \sqrt{(\mathbf{v}'_i - \mathbf{m}'_p)^T (\sum)^{-1} (\mathbf{v}'_i - \mathbf{m}'_p)} < \varphi$$

Where $\mathbf{v}'_i$ is velocity from the test trajectory, $\mathbf{m}'_p$ is the mean, φ a distance threshold, and $\sum$ is the covariance matrix of our path velocity distribution.

Spatio-Temporal Curvature Similarity: The third step allows us to capture the discontinuity in the velocity, acceleration and position of our trajectory. Thus we are able to discriminate between a person walking in a straight line and a person walking in an errant path. The velocity $\mathbf{v}'_i$ and acceleration $\mathbf{v}''_i$, first derivative of the velocity, is used to calculate the curvature of the trajectory. Curvature is defined as,

$$\kappa = \frac{\sqrt{y''(t)^2 + x''(t)^2 + (x'(t)y''(t) - x''(t)y(t))^2}}{(\sqrt{x'(t)^2 + y'(t)^2 + 1})^3}$$

where x' and y' are the x and y components of the velocity. Mean and standard deviation of κ's are determined to fit a Gaussian distribution for spatio-temporal characteristic. We compare the curvature of the test trajectory with our distribution using the Mahalanobis distance, bounded by a threshold. By using this measure we are able to detect irregular motion. For example, a drunkard walking in a zigzag path, or a person slowing down and making a u-turn.

4. RESULTS

The proposed system has been tested on three 320×240 pixels resolution sequences containing a variety of motion trajectories: **Seq** #1 is a short sequence of 3730 frames with 15 different trajectories forming two unique paths, the clustered trajectories are shown in Fig. 5(b)-(d), **Seq** #2 is a sequence

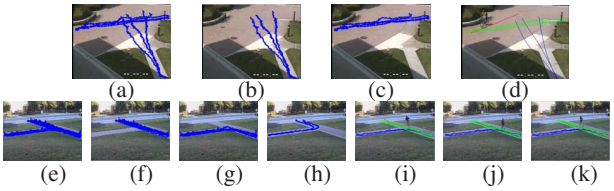


Fig. 5. (b)(c) show two clustered path for **Seq** #1 while (a) shows all the trajectories in the training phase. (d) demonstrates a test case. For **Seq** #2, all the trajectories in the training set and the subsequent clusters are shown in (e)-(h). The test cases are shown in (i),(j) and (k).

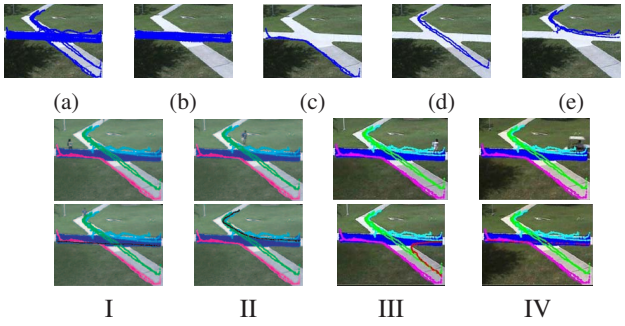


Fig. 6. Clustered trajectories from the training sequence of **Seq** #3: (a) shows all the trajectories used in the training set. (b)-(e) are the 4 paths clustered from the input data. Column I and II demonstrate normal behavior, while column III and IV demonstrate two examples of unacceptable behaviors. See text for more details.

of 9284 frames with 27 different trajectories forming 3 different paths after clustering, as shown in Fig. 5(e)-(h), **Seq** #3 contains over 20 minutes of data forming over 100 trajectories of people walking around in the scene. The trajectories are clustered into 4 path models, as shown in Fig. 6(a)-(e).

One test case from **Seq** #1 is shown in Fig. 5(d). The training sequence only contained people walking in the scene. But the bicyclist shown in (d) has motion characteristics different (having higher speed) than the training cases, hence detected as abnormal behavior (displayed in red). Three test cases from **Seq** #2 are depicted in Fig. 5(i)-(k). A person walking in a zig-zag fashion (Fig. 5(i)), and a person running (Fig. 5(j)) are flagged for an activity that is considered as an unusual behavior. Fig. 5(k) demonstrates a case where a person walks at a normal pase in conforming behavior. Some of the test cases from **Seq** #3 are shown in Fig.6. Two cases in the first two columns contain people walking at normal pace - following the path model constructed in the training phase, hence flagged with a black trajectory i.e. acceptable behavior. Third column is flagged unacceptable as the person moves left, which is not contained in the model. Similarly, 4^{th} column contains a golf cart driven across the scene.

5. CONCLUSION

This paper proposes a unified method for path modeling, detection and surveillance. The trajectory data is metric rectified to represent a truer picture of the data. Metric rectified observed scene is registered to aerial view to extract metric information from the video sequence, for example, the actual speed of an object. Normalized-cuts are then used to cluster metric rectified input training trajectories into various paths. We extract spatial, velocity and spatio-temporal curvature based features from the clustered paths and use it for unusual behavior detection. The proposed path modeling method has been extensively tested on a number of sequences and have demonstrated satisfactory results. Recognizing more complex events by attaching meanings to the trajectories is also one of our future goals.

References

[1] Omar Javed and Mubarak Shah, "Tracking and object classification for automated surveillance," in *the seventh European Conference on Computer Vision*, 2002.

[2] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[3] Jeffrey E. Boyd, Jean Meloche, and Y. Vardi, "Statistical tracking in video traffic surveillanc," in *International Conference on Computer Vision (ICCV)*, 1999.

[4] Neil Johnson and David Hogg, "Learning the distribution of object trajectories for event recognition," in *Proc. of British Machine Vision Conference (BMVC)*, 1995.

[5] John Wright and Robert Pless, "Analysis of persistent motion patterns using the 3d structure tensor," in *Proceedings of the IEEE Workshop on Motion and Video Computing*, 2005, pp. 14–19.

[6] Dimitrios Makris and Tim Ellis, "Path detection in video surveillance," *Image and Vision Computing Journal (IVC)*, vol. 20, no. 12, pp. 895–903, 2002.

[7] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] Yaser Sheikh and Mubarak Shah, "Object tracking across multiple independently moving airborne cameras," in *IEEE International Conference on Computer Vision, 2005.*, 2006.

[9] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, 2000.

[10] Eamonn Keogh, "Exact indexing of dynamic time warping," in *28th International Conference on Very Large Data Bases. Hong Kong*, 2002, pp. 406–417.