

Trajectory Rectification and Path Modeling for Video Surveillance*

Imran N. Junejo and Hassan Foroosh

School of Electrical Engineering and Computer Science, University of Central Florida

Abstract

Path modeling for video surveillance is an active area of research. 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 (i) a novel linear solution to auto-calibrate any camera observing pedestrians and (ii) to use these calibrated cameras to detect unusual object behavior. During the unsupervised training phase, after auto-calibrating a camera and metric rectifying the input trajectories, the input sequences are registered to the satellite imagery and prototype path models are constructed. This allows us to estimate metric information directly from the video sequences. 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. We test the proposed method on synthetic as well as on real-world pedestrian sequences.

1. Introduction

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. Moreover, as common pathways are detected by clustering the trajectories, we can efficiently assign detected trajectory its associated path model, thereby only storing the path label and the object labels instead of the whole trajectory set, resulting in a significant compression for storing surveillance data.

The task of path surveillance through a single camera into two steps. The first step is the removal of projective distortion from the object trajectories once the camera is calibrated. Original work on camera calibration using vanishing points started from the seminal paper by Caprile and

Torre[4]. Liebowitz et al.[10] developed a method to compute the camera intrinsics by using the Cholesky Decomposition [6]. Lv et al.[11] were the first to propose calibration by recovering the horizon line and the vanishing points from observed walking humans. However, their formulation does not handle robustness issues. Recently Krahnstover and Mendonça[9] proposed a Bayesian approach for auto-calibration by observing pedestrians. Foot-to-head homology is decomposed to extract the vanishing point and the horizon line for calibration. However, their method requires prior knowledge about unknown calibration parameters and prior knowledge about the location of people; their algorithm is also non-linear.

Path model is created for once the object trajectories are metric rectified. Grimson et al. [5] 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. Recently [14] uses the 3D structure tensor for representing global patterns of local motion. Makris and Ellis [12] develop a spatial model to represent the routes in an image. Once a trajectory of a moving object is obtained, it is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function. One limitation of this approach is that only spatial information is used for trajectory clustering and behavior recognition.

In this paper, we present a novel linear method to metric rectify object trajectories by observing pedestrians in a scene(Section 2). A novel application of Normalized-cuts is used to cluster the rectified trajectories in to distinct paths (Section 3). Once the trajectories are clustered, we extract meaningful features to build our model and check the conformity of a test trajectory to our built path model (Section 4). We rigourously test the proposed method on real and synthetic data and the results are encouraging (Section 5). We also demonstrate an application of the proposed method for registration to the satellite imagery (Section 6).

*The support of the National Science Foundation(NSF) through the Grant # IIS-0644280 is gratefully acknowledged.

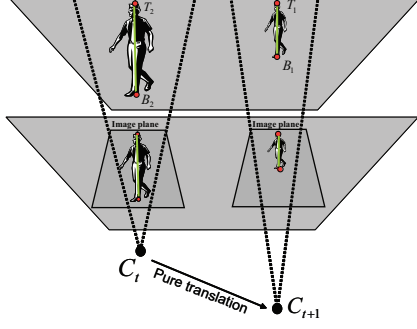


Figure 1. **Observing pedestrians:** Each of the two instances of a pedestrian can be assumed to be a stationary camera. Then, the two cameras define an epipolar geometry between them. See text for more details.

2. Training Phase

Our goal in the training phase is to *first*: calibrate the camera so that the extracted object trajectories are metric rectified. *Second*, to cluster the input trajectories and build a model based on our features (Section 3) which are then used to test the incoming trajectories.

2.1. Induced fundamental matrices

The fundamental matrix satisfies the condition that for any pair of corresponding points $x \longleftrightarrow x'$ (in two images):

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (1)$$

where the point \mathbf{x}'^T is mapped to a line $\mathbf{l} = \mathbf{F} \mathbf{x}$ in the other image such that $\mathbf{x}'^T \mathbf{l} = 0$ [6]. An important concept is the **epipole** - the image in one view of the camera center of the other view (cf. Fig 2a). It is also the vanishing point of the baseline (i.e. the line joining the two camera centers) direction. The epipole \mathbf{e} is given as the right null-vector of \mathbf{F} : $\mathbf{F} \mathbf{e} = \mathbf{0}$. Similarly, \mathbf{e}' is the left null-vector of \mathbf{F} .

As an object or a pedestrian of height h traverses the ground plane, each location on this plane corresponds to exactly one location on the head plane. As shown in Fig. 1, the head of the pedestrian is labeled as \mathbf{T}_i , while the feet as \mathbf{B}_i , where $i = 1, 2, \dots, n$; n being the number of frames in which the pedestrian is visible. Without loss of generality, for a simple case of two frames, this head-to-feet correspondence can be mapped by a fundamental matrix.

A special case of fundamental matrix is *induced* by the pedestrian movements in our scene. *The key idea is:* instead of considering translation of the pedestrians (any two instances can be considered as being translating), one may equivalently consider the situation in which the camera undergoes translation, and the world is stationary. This is as depicted in Fig. 1. This *re-formulation* of the problem allows us to introduce the concept of fundamental matrix into our problem. Each instance of a pedestrian (head and feet location) can be treated as one single image. Therefore, in

our case, when the motion of the camera is pure translational, the fundamental matrix has the form:

$$\mathbf{F}_h = [\mathbf{e}']_{\times} \mathbf{K} \mathbf{R} \mathbf{K}^{-1} = [\mathbf{e}']_{\times} \quad (2)$$

where $\mathbf{R} = \mathbf{I}$, $[\mathbf{e}']_{\times}$ is the skew-symmetric matrix representation of the epipole and \mathbf{F}_h is defined as $\mathbf{T}_i^T \mathbf{F}_h \mathbf{T}_j = 0$, where $i \neq j$. Note that \mathbf{F}_h now has only 2 d.o.f., instead of 7 [6], which correspond to the position of the epipole. Therefore, only two point correspondences, $\mathbf{T}_i \longleftrightarrow \mathbf{T}_j, \mathbf{B}_i \longleftrightarrow \mathbf{B}_j$ where $i \neq j$, should be sufficient to compute \mathbf{F}_h . The two epipoles \mathbf{e} and \mathbf{e}' are also collinear.

\mathbf{F}_h can be considered as mapping points in a direction parallel to the ground plane or *horizontally*. We can also introduce another fundamental matrix for the *vertical* direction such that $\mathbf{T}_i^T \mathbf{F}_v \mathbf{B}_i = 0$. Thus now we are looking at the correspondences $\mathbf{T}_i \longleftrightarrow \mathbf{B}_i$. This is shown in Fig. 2b. The special epipolar geometry arising for a pure translating camera is depicted in Fig. 2a. As this figure shows, the intersection of the baseline with the image plane is at infinity. That is, the epipole lies at infinity or the epipole becomes a vanishing point.

Fig. 2b depicts the unique geometry induced by pedestrians. For any two instances of a pedestrian, the 2 d.o.f. \mathbf{F}_h can be estimated by solving the following two linear equations:

$$\mathbf{T}_1^T \mathbf{F}_h \mathbf{T}_2 = 0 \quad (3)$$

$$\mathbf{B}_1^T \mathbf{F}_h \mathbf{B}_2 = 0 \quad (4)$$

Similarly, \mathbf{F}_v can be estimated by solving:

$$\mathbf{T}_1^T \mathbf{F}_v \mathbf{B}_1 = 0 \quad (5)$$

$$\mathbf{T}_2^T \mathbf{F}_v \mathbf{B}_2 = 0 \quad (6)$$

Once the fundamental matrix is determined, the epipole is computed as the null-vector of the fundamental matrix, as described above.

As Fig. 2a shows, the epipole \mathbf{e}_h for \mathbf{F}_h lies on the plane at infinity i.e. it is a vanishing point. Similarly \mathbf{e}_v is also a vanishing point. These two vanishing points represent mutually orthogonal (horizontal and vertical) directions in world. Therefore, these points are used to enforce orthogonality constraint [6] on the IAC $\boldsymbol{\omega}$:

$$\mathbf{e}_v^T \boldsymbol{\omega} \mathbf{e}_h = 0 \quad (7)$$

Eq. (7) is a linear equation with an unknown parameter w_{11} of $\boldsymbol{\omega}$. Once w_{11} is determined, Cholesky decomposition is applied to $\boldsymbol{\omega}$ to obtain the camera calibration matrix \mathbf{K} .

Determining head/foot locations: We use [7] to extract the foreground objects from the video sequence. The head and feet locations are easily estimated by calculating the center of mass and the second order moment of the lower and the upper portion of the bounding box of the foreground region [9].

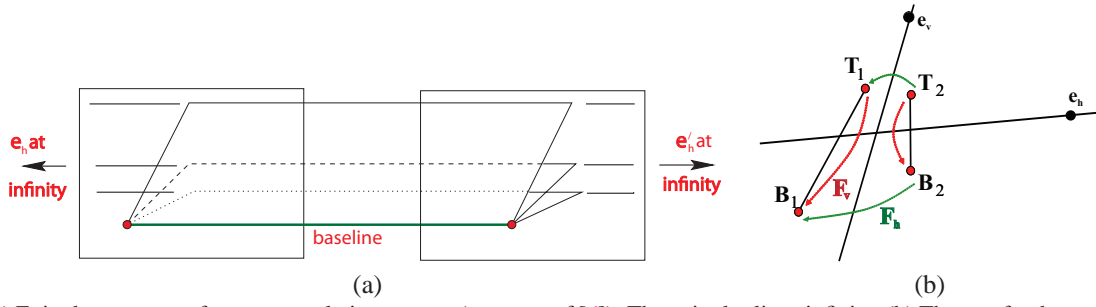


Figure 2. (a) Epipolar geometry for pure translating camera (courtesy of [6]). The epipoles lie at infinity. (b) The two fundamental matrices, F_v and F_h , induced by pedestrians.

2.2. Robust auto-calibration

Eq. (7) provides only one constraint on ω . Unless we have more information, we can only solve for one unknown in $\omega = \text{diag}(\omega_{11}, \omega_{11}, 1)$. Fortunately, this equation is linear and therefore can be simplified to the form: $a_i w_{11} + b_i = 0$, where the subscript i indicates the frame number. Thus from each image pair we obtain one equation with one unknown. Equations obtained from a sequence are used to construct an over-determined system of equations:

$$\underbrace{\begin{bmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix}}_{\mathbf{Q}} \begin{bmatrix} w_{11} \\ 1 \end{bmatrix} = 0 \quad (8)$$

We want to use robust statistics to recover the best w_{11} such that \mathbf{K} is closest to the actual calibration matrix. Therefore, to deal with the outliers in the data, Total Least Squares (TLS) method is adopted to solve the system of Eqs (8). Given an over-determined system of equations, TLS problem is to find the smallest perturbation to the data and the observation matrix to make the system of equations compatible. A suitable function also needs to be selected that is less forgiving to outliers, one such example is the *truncated quadratic* [2], commonly used in computer vision. The errors are weighted up to a fixed threshold, but beyond that, errors receive constant penalty. Thus the influence of outliers goes to zero beyond the threshold.

We use the truncated Rayleigh quotient to remove outlier influence. The quotients are estimated as:

$$\rho(w_{11}) = \sum \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} < \xi \quad (9)$$

where $\mathbf{x} = \begin{bmatrix} w_{11} \\ 1 \end{bmatrix}$, $A = [a_i^j \ b_i^j]^T [a_i^j \ b_i^j]$ and ξ is the threshold. The Rayleigh quotients are estimated from the observation points and the residual errors are estimated. The threshold ξ is set to the median of all the residual errors. Observation points obtained from Eq. 8 having residual errors greater than ξ are removed as outliers. After outlier removal, the *outlier-free* remaining observation points \mathbf{Q} are used to construct the over-determined system of Eqs.

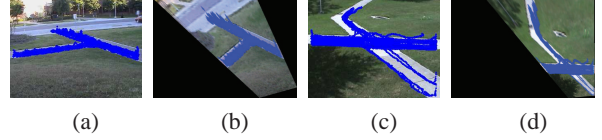


Figure 3. **Rectified Trajectories:**(b) represents reconstructed trajectories for **Seq #2** - shown in (a), while (d) represents **Seq #3**, shown in (c), rectified.

(8). The system is then solved using the Singular Value Decomposition (SVD). The correct solution is the eigenvector corresponding to the smallest eigenvalue.

In summary, in order to minimize the influence of noise on our observation matrix \mathbf{Q} , we apply the Rayleigh quotient to *filter* out the noisy data points. Once the outliers are removed, the Total Least Squares method is applied to the remaining observation points to estimate the unknown parameter w_{11} of the IAC.

2.3. Trajectory and Image Rectification

Metric rectified trajectory data presents a truer picture of the original data. Therefore, once the camera is calibrated, the object trajectories are metric rectified. The line at infinity l_∞ intersects ω at two complex conjugate ideal points \mathbf{I} and \mathbf{J} , called the *circular points* [6]. The conic dual to these circular points is a degenerate conic and is invariant under similarity transformation. Once this conic $\mathbf{C}_\infty^{*'}$ is identified, a suitable rectifying homography is obtained by using the

SVD decomposition: $\mathbf{C}_\infty^{*'} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}^T$ where \mathbf{U}

is the rectifying projectivity. Fig. 3 depicts some results obtained by rectifying the obtained training trajectories from two of our three test sequences.

From here on, all references to 2-D trajectories imply *rectified 2-D* trajectories. For simplicity and better visualization, the results are still shown on un-rectified image plane in subsequent sections.

3. Model Building

A typical video sequence 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 im-

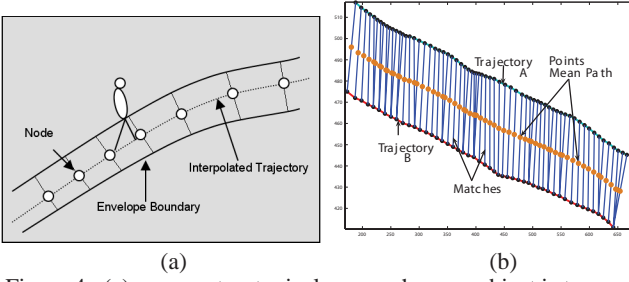


Figure 4. (a) represents a typical scene where an object is traversing an existing path. An average trajectory and an envelope boundary are calculated for each set of clustered trajectories. (b) An example of an average trajectory obtained by applying DTW on two sample trajectories. Blue lines connect corresponding matched points between the two trajectories.

age coordinates for the trajectory obtained can be given as $\mathbf{T}_i = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. Depending on the velocity of a person and the location in the image plane, the trajectories will be of varying lengths. Instead of tracking the centroid of an object, tracking feet gives more accurate results for our method.

3.1. Trajectory Clustering

Perceptually, humans tend to group trajectories based on their spatial proximity. Since we are trying to build a path model, it is essential that we perform clustering using the spatial characteristics of the trajectories. One such measure is the Hausdorff distance. For two trajectories \mathbf{T}_i and \mathbf{T}_j , the Hausdorff distance, $D(\mathbf{T}_i, \mathbf{T}_j)$, is defined as

$$D(\mathbf{T}_i, \mathbf{T}_j) = \max\{d(\mathbf{T}_i, \mathbf{T}_j), d(\mathbf{T}_j, \mathbf{T}_i)\}$$

$$\text{where } d(\mathbf{T}_i, \mathbf{T}_j) = \max_{a \in \mathbf{T}_i} \min_{b \in \mathbf{T}_j} \|a - b\|.$$

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 [13] are used recursively to partition the graph. Normalized-cuts avoid bias for partitioning out small sets of points and it is also very easy to compute. Fig. 8a-d shows the results obtained by clustering one of our data set.

3.2. Envelope & Mean Path Construction

Once all the trajectories are clustered into different paths, we create a spatial envelope for each single clustered path. An *envelope* can be defined as the spatial extent of a path (cf. Fig. 4(a)). Applying Dynamic time Warping (DTW)

algorithm ([8]), 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 $q(i, j) = \frac{e^{-\frac{(-\kappa(i, j))}{\sigma_\kappa}}}{2} + e^{-\frac{(-i\bar{j})}{\sigma_e}}$, $\frac{(-\kappa(i, j))}{\sigma_\kappa}$ 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 the 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).

4. 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 $\mathbf{T}_i(x_i, y_i, t_i)$, $i = 0, 1, \dots, N-1$, is calculated as:

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

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,

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

where \mathbf{v}'_i is velocity from the test trajectory, \mathbf{m}'_p is the mean, φ a distance threshold, and Σ 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.

In summary, we initially detect non-conforming trajectories on the basis of spatial dissimilarity. In case the given trajectory is spatially similar to one of the path models, the similarity in the velocity feature of the trajectories in that path and the given trajectory is computed. If the motion features are also similar then a final check on spatio-temporal curvature is made. The trajectory is deemed to be anomalous if it fails to satisfy any one of the spatial, velocity or spatio-temporal curvature constraints.

5. Results

The proposed system has been tested on three 320×240 pixels resolution sequences containing a variety of motion trajectories:

Seq #1: This is a short sequence of 3730 frames with 15 different trajectories forming two unique paths. The clustered trajectories are shown in Fig. 7. Trajectories obtained for the training sequence are depicted in Fig. 7(a)(b)(c), representing different behavior of the pedestrians.

Seq #2: A real sequence of 9284 frames with 27 different trajectories forming 3 different paths after clustering. The length of the trajectories varies from 250 points to almost 800 points. The trajectories clustered into paths are shown in Fig. 8.

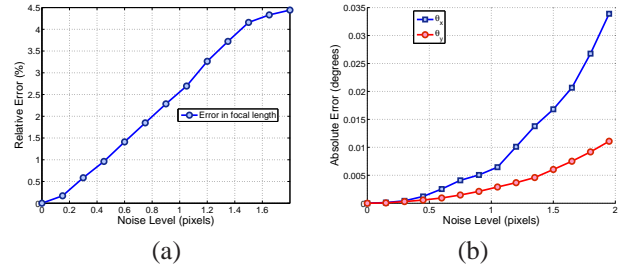


Figure 5. Performance of auto-calibration method VS. Noise level in pixels: (a) error in focal length. (b) error in the estimated angles.



Figure 6. (a) - (i) from left-to-right: The figure depicts instances of the data sets used for testing the proposed auto-calibration method.

Seq #3: The training sequence 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. 9(a)-(e).

5.1. Evaluating the Auto-Calibration

Synthetic data: Eleven vertical lines of same height but random locations were generated to represent pedestrians in our synthetic data. The ends of the lines indicate the head or the foot locations. We gradually add a Gaussian noise with $\mu = 0$ and $\sigma \leq 2$ pixels to the data-points and perform 1000 independent trials for each noise level, the results are shown in Figure 5. The relative error in f increases almost linearly with respect to the noise level. For a maximum noise of 2 pixels, we found that the error was under 5%. The absolute error in the rotation angles increases linearly and is well under 0.5 degrees.

Real Data: As reported by [15], the mean of the estimated focal length is taken as the ground truth and the standard deviation as a measure of uncertainty in the results. Due to space limitations, we only show results for the obtained focal lengths.

For **Seq #1**, the estimated results for f are given in Table 1(left column). The estimated focal length is $f = 948.74$ with a low standard deviation of $\sigma = 8.7$. **Seq #2** is another sequence used for testing, a couple of instances are shown in Fig. 6f-g. The estimated focal lengths are very close to each other, as shown in Table 1(right column - top). Similarly, results for **Seq #3** are shown in Table 1(right column - bottom).

The error in the results can be attributed to many factors. One of the main reason is that only a few frames are used per sequence. The standard deviation for f in all our experiments is found to be less than reported by [9].

Seq #1	(f)	Seq #2	(f)
Fig. 6a	$f = 955.31$	Fig. 6f	$f = 976.09$
Fig. 6b	$f = 938.87$	Fig. 6g	$f = 980.24$
Fig. 6c	$f = 952.05$	Seq #3	(f)
		Fig. 6f	$f = 840.68$
		Fig. 6g	$f = 837.84$

Table 1. The recovered focal length for (starting from the left column, going clock wise direction) Seq #1, Seq #2 and Seq #3.

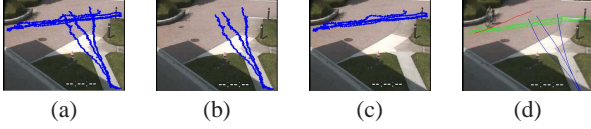


Figure 7. (b)(c) show three clustered path for Seq #1 while (a) shows all the trajectories in the training phase. (d) demonstrates a test case where a bicyclist is detected as having unusual behavior.

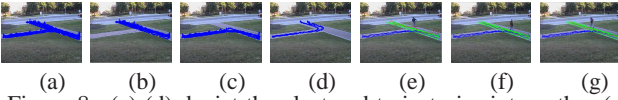


Figure 8. (a)-(d) depict the clustered trajectories into paths. (e)-(g) show instances of a drunkard walking, a person running, and a person walking, respectively. Red trajectories denote unusual behavior while the black trajectories are the casual behavior.

5.2. Evaluating path modeling

Result for Seq #1 is shown in Fig. 7(d). The training sequence only contained people walking in the scene. But the bicyclist shown in (d) has motion characteristics different (containing faster movement) than the training cases, hence detected as abnormal behavior (displayed in red).

Three test cases from Seq #2 are depicted in Fig. 8(e)-(g). A person walking in a zig-zag fashion (Fig. 8(e)), and a person running (Fig. 8(f)) are flagged for an activity that is considered as an unusual behavior. Fig. 8(g) demonstrates a case where a person walks at a normal pace in conforming behavior.

Some of the test cases from Seq #3 are shown in Fig. 10. 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, 4th column contains a golf cart driven across the scene.

The system gives satisfactory results for our experiments. Although some existing methods do incorporate model update, we believe this is what leads to a *model drift*. That is, after a number of updates the model can become general enough to accommodate any behavior considering it as acceptable behavior. But certainly, the applicability of our proposed system lies in the spheres where there is



Figure 9. Results 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.

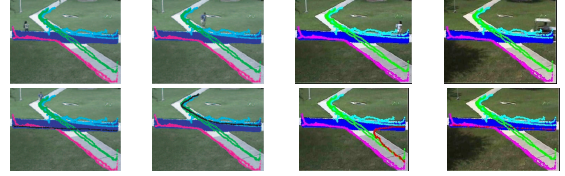


Figure 10. Results for Seq #3. Column 1 and 2 demonstrate normal behavior, while column 3 and 4 demonstrate two examples of unacceptable behaviors. See text for more details.

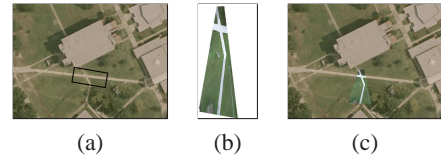


Figure 11. Image rectification and registration results for Seq #3.

a defined behavior, differentiable from certain other unacceptable behavior for, lets say, security reasons.

6. Registration to Aerial Imagery

Registration to the satellite imagery gives a global view of the scene under observation. Once the observed scene is metric rectified, the only unknown transformation is the similarity transformation. This rectified image can then be automatically registered to the aerial image [1]. The results obtained by rectifying the test sequences are shown in Fig. 11. Due to space limitation, we show results only on Seq #3. A frame from the test sequence Seq #3 is rectified by using the line at infinity which is obtained as: $l_\infty = \omega v_z$. The obtained circular points are used to construct the conic C_∞^* in order to obtain the rectifying projectivity, as described in Section 2.3. The rectified image is shown in Fig. 11(b), and the registered image is shown in Fig. 11(c).

Registration of multiple cameras to the satellite image is shown in Fig. 12. Three cameras were placed at three different locations along the path shown in the figure. The proposed method is automated and provides satisfactory results. Thus for any test sequences, the obtained path model can be mapped to the corresponding satellite image in order to obtain a global view - representing the behavior of pedestrians in that particular area.

Retrieving metric information: Generally, the satellite image contains the world-to-image scale information, as

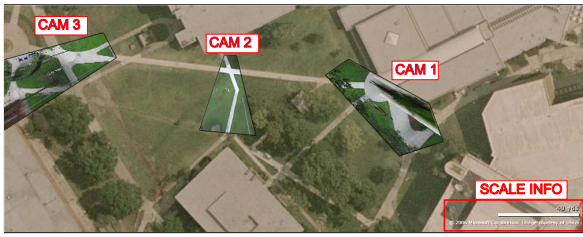


Figure 12. Multiple cameras registered to the corresponding satellite image: The input images have a few new structures compared to the old satellite image.

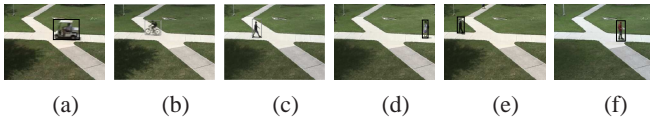


Figure 13. Six test cases used to retrieve metric information. See text for more.

shown in Fig. 12, where 140 pixels correspond to 40 yards. Five cases are shown in Fig. 13. Fig. 13(a) shows a golf cart that takes only two seconds to move across the scene - the true speed obtained from the registered image is found to be 20.369 km/hr. The velocity of the bicycle, as shown in Fig. 13(b), is found to be 12.22 km/hr, whereas for three cases of pedestrians (i.e. Fig. 13(c)-(e)) the velocity is determined to be 4.58 km/hr, 3.66 km/hr, and 4.22 km/hr, respectively, which is very close to the average human walking speed. A case of a person riding a skateboard is shown in Fig. 13(f) and the retrieved velocity is 9 km/hr.

7. Conclusion

This paper proposes a unified method for path modeling, detection and surveillance. We propose a novel linear method for auto-calibrating any camera that may be involved. After calibration, 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. Calibration method and the path modeling method has been extensively tested on a number of sequences and have demonstrated satisfactory results. We plan on using multiple cameras to build path models in a large scale environment. Recognizing more complex events by attaching meanings to the trajectories is also one of our future goals.

References

[1] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In

Proceedings of the Second European Conference on Computer Vision (ECCV), 1992. 6

- [2] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Journal of Computer Vision and Image Understanding*, 63(1):75–104, January 1996. 3
- [3] J. E. Boyd, J. Meloche, and Y. Vardi. Statistical tracking in video traffic surveillance. In *International Conference on Computer Vision (ICCV)*, 1999. 1
- [4] B. Caprile and V. Torre. Using vanishing points for camera calibration. *Int. J. Comput. Vision*, 4(2):127–140, 1990. 1
- [5] W. 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. 1
- [6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 1, 2, 3
- [7] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *the seventh European Conference on Computer Vision*, 2002. 2
- [8] E. Keogh. Exact indexing of dynamic time warping. In *28th International Conference on Very Large Data Bases. Hong Kong*, pages 406–417, 2002. 4
- [9] N. Krahnstoever and P. R. S. Mendonca. Bayesian autocalibration for surveillance. In *Tenth IEEE International Conference on Computer Vision*, 2005. 1, 2, 5
- [10] D. Liebowitz and A. Zisserman. Combining scene and auto-calibration constraints. In *Proc. IEEE ICCV*, pages 293–300, 1999. 1
- [11] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *IEEE International Conference of Pattern Recognition*, 2002. 1
- [12] D. Makris and T. Ellis. Path detection in video surveillance. *Image and Vision Computing Journal (IVC)*, 20(12):895–903, 2002. 1
- [13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000. 4
- [14] J. Wright and R. Pless. Analysis of persistent motion patterns using the 3d structure tensor. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 14–19, 2005. 1
- [15] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. 5