Linear cyclic pursuit based prediction of personal space violation in surveillance video

Neha Bhargava*, Subhasis Chaudhuri*, and Guna Seetharaman‡
*Indian Institute of Technology Bombay, India
Email: neha@ee.iitb.ac.in, sc@ee.iitb.ac.in
‡Information Directorate, Air Force Research Laboratory, Rome, NY 13441
Email: gunasekaran.seetharaman@us.af.mil

***Abstract*—Analysis of human interaction in a social gathering is of high interest in security and surveillance applications. It is also of psychological interest to study the interaction to get a better understanding of the participant behavior. This paper is an attempt to explore and analyze interactions among the individuals from a single calibrated camera. We are particularly interested in trajectory prediction. These predicted trajectories of individuals are then used in predicting personal space violation. Each individual, represented by a feature point in a 2.5D coordinate system, is tracked using Lucas-Kanade tracking algorithm. We use the linear cyclic pursuit framework to model this point motion. This model is used for short-term prediction of individual trajectory. We demonstrate these ideas on different types of datasets.**

Keywords- human interaction, linear cyclic pursuit, short-term prediction

I. Introduction

Security and safety are among the major concerns in crowded public places such as airports, shopping malls or in any social gathering. Analyzing human activity at such crowded places is interesting and challenging at the same time. Crowd can be defined as a group of interacting individuals. This interaction is guided by an underlying set of rules that depends highly on the place under observation and the undergoing event. For example, in a football match players interact according to their game rule and their strategy. There might be an obvious pattern in the crowd motion or it might look random in the extreme case. For example, in a marathon the participants follow a well defined pattern whereas in shopping malls it may be difficult to find any such obvious pattern.

Various attempts have been made in the past to model and analyze the crowd motion. Zhan et al. [1] provide a survey of various crowd analysis methods employed in computer vision applications. They also discuss the crowd motion from the perspectives of sociology and psychology. The crowd analysis approaches can be categorized into two groups at the top level: holistic[1] and reductionist. Holistic approaches consider crowd as a single entity [1]–[4]. In [3], authors modeled the crowd movement as the fluid flow and use fluid dynamic concepts for abnormal flow detection. Andrade et al. [4] use optical flow to analyze crowd motion. They use unsupervised learning to detect abnormal crowd behavior. On the other hand, we refer reductionist approaches as those approaches which consider crowd as a group of individuals [5]–[7] and not to be very dense. The crowd behavior is studied by analyzing individuals and their interactions. So it is required to detect and track individuals. Kalman filtering has been a popular approach for tracking trajectories. Choi et al. [5] exploit the spatial distribution of the pedestrians with their pose and motion to identify the collective activity. They used extended Kalman filtering for tracking pedestrians. Robert et al. [6] use a Kalman filter for tracking pedestrians and then recognize human activities based on position and velocity of pedestrian. Ali and Terada [7] proposed a robust framework for multi human tracking based on Kalman filtering.

Andrade et al. in [8] model an individual's motion in a crowd based on the social force model [9]. This model assumes that pedestrian motion is influenced by the other pedestrians and the environment. They model the motion by forces of attraction and repulsion acting on the pedestrian. Srikrishnan et al. [10] adopted the linear cyclic pursuit (LCP) [11] framework for modeling crowd motion. They used a sparse set of individuals from the crowd to model the motion in image plane. This model is used for predicting short term trajectories of individual agents. They have also shown the equivalence of LCP and crowd motion model based on social force model. Therefore we will use LCP for motion modeling in this paper.

Our contribution in this direction is two fold. Firstly instead of tracking individuals in a image plane, we track them in 2.5D coordinate system, i.e. on a plane in world coordinate system. This helps in estimating true distances between individuals. Secondly we also predict the trajectories under LCP framework in order to analyze the individual behavior in the near future. This helps in predicting personal space violation which in turn can assist in security applications.

The rest of the paper is organized as follows: Section II motivates and states the problem. Next section discusses the proposed method. Section IV describes the experimental setup and results, followed by conclusions in section V.

II. Motivation and Problem Definition

We are interested in exploiting the spatio-temporal distribution of the individuals in a gathering. By spatio-temporal distribution we refer to the positions of the individuals which change with the time. Consider a social gathering where people interact with each other. A socially important person is present in the gathering whose security is of concern. Is it possible to predict her/his personal space violation? If yes, then it can help in taking timely security action. It requires modeling the motion and then prediction of the individuals' trajectories.

III. Methodology

We represent each agent (individual) by a feature point which are manually initialized. To get the true trajectories, we need 3D coordinates of these points. Since it is an ill-posed problem with a single camera, we estimate the 2.5D coordinates. With the assumption that all the feature points are on the same plane (which is parallel to the ground plane) and camera parameters known, 2.5D coordinates for the feature points can be estimated. The camera parameters can be estimated by camera calibration. Once we get the 2.5D coordinates, we track these feature points in a 2.5D coordinate system using the Lucas-Kanade tracking algorithm [15]. Once these trajectories of the feature points are known, we use the LCP framework to model the crowd motion. Finally we use this model for prediction of trajectories. Since agents are intelligent they can move on their own which leads to time varying motion dynamics. This requires updating the model parameters continuously and to predict short-term trajectories using the estimated model parameters.

This section discusses the proposed method in detail. It includes introduction to linear cyclic pursuit followed by parameter estimation, prediction method, camera calibration and distance measurement.

A. Linear Cyclic Pursuit

We discuss the concept of the linear cyclic pursuit [11] in this section. In this model, each agent follows the weighted centroid of other agents' positions. In this paper, an agent refers to a feature point corresponding to an individual/vehicle which is being temporally followed. It could be a feature point on the head of the individual or on a vehicle. The method for extracting these feature points is discussed later.

Consider a group of N agents in $\mathbf{R}^2$ i.e. (X, Y). The motion in each dimension is assumed to be independent of that in the other dimension. Therefore motion parameters can be estimated for each dimension separately. Let the initial positions of all agents be known. Now consider the motion along the X dimension. Motion equations in the other dimension can be obtained similarly. For the i^{th} agent, motion equation at time t_j is given as:

$$\dot{x}_i(t_j) = k_i[\sum_{k=1}^{N-1} \eta_k x_{(i+k)_N}(t_j) - x_i(t_j)] \tag{1}$$

where k_i is the gain for the i^{th} agent, η_k is the weight of the k^{th} agent and $(a)_N$ denotes a mod N. Also $\eta_k \geq 0$ and $\sum_{k=1}^{N} \eta_k = 1$. The parameters k_i and η_k decide the agent's motion and hence are called motion parameters. Therefore in matrix form, the motion along the X direction becomes

$$\dot{\mathbf{X}}(t) = A_x \mathbf{X}(t) \tag{2}$$

where $\mathbf{X}(t) = [x_1(t)\ x_2(t)......x_N(t)]^T$ and

$$A_x = \begin{bmatrix} -k_1 & k_1\eta_2 & \dots & k_1\eta_N \\ k_2\eta_1 & -k_2 & \dots & k_2\eta_N \\ \dots & \dots & \dots & \dots \\ k_N\eta_1 & k_N\eta_N & \dots & -k_N \end{bmatrix} \tag{3}$$

Since we are interested in short-term prediction, the analysis regarding the point of convergence and stability for the above motion model is not required. However a discussion on these issues can be found in [11]. Solving the first order differential equation along (2) the X and Y directions gives

$$\begin{aligned} \mathbf{X}(t) &= \exp^{A_x t} \mathbf{X}(0) \\ \mathbf{Y}(t) &= \exp^{A_y t} \mathbf{Y}(0) \end{aligned} \tag{4}$$

where $\mathbf{X}(0) = [x_1(0)\ x_2(0)......x_N(0)]^T$ and $\mathbf{Y}(0) = [y_1(0)\ y_2(0)......y_N(0)]^T$ denote the initial positions of the agents.

B. Motion parameter estimation and trajectory prediction

To be able to predict the trajectories using the LCP model, we need to learn the parameter matrices A_x and A_y. Assume that we know the positions of all N agents for M consecutive frames in a video, where $M > N$. For the i^{th} agent, M motion equations of the form (1) can be obtained and written in a matrix form as

$$\begin{bmatrix} \dot{x}_i(t_1) \\ \dot{x}_i(t_2) \\ \dots \\ \dot{x}_i(t_M) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_N(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_N(t_2) \\ \dots & \dots & \dots & \dots \\ x_1(t_M) & x_2(t_M) & \dots & x_N(t_M) \end{bmatrix} \begin{bmatrix} \alpha_{1i} \\ \alpha_{2i} \\ \dots \\ \alpha_{Ni} \end{bmatrix} \tag{5}$$

or equivalently,

$$\dot{X}_i = X_i \vec{a}_i \qquad \text{for all } 1 \leq i \leq N, \tag{6}$$

where $\vec{a}_i = [\alpha_{1i}, \alpha_{2i}, ..., \alpha_{Ni}]^T$ with $\alpha_{ii} = -k_i$ and $\alpha_{ji} = k_i\eta_j$.

Similarly the motion equation for Y dimension is

$$\dot{Y}_i = Y_i \vec{b}_i \qquad \text{for all } 1 \leq i \leq N \tag{7}$$

These set of equations are solved for $\vec{a}_i$ and $\vec{b}_i$ using the least squares method. Observe that fixed A_x and A_y would mean that motion dynamics is not changing which is a very restrictive condition. Since agents are intelligent, they can move on their own which means (4) is valid for a short time. Therefore we update $\vec{a}$ and $\vec{b}$ with each incoming frame. Latest M frames are used for estimating the motion parameters.

C. Camera Calibration

As discussed earlier, we are interested in estimating 2.5D coordinates of the agents which are the feature points representing individuals. The key assumption here is that feature points lie on a known plane parallel to the ground plane. Fig. 1 gives the camera setup. The camera parameters are required to obtain the 2.5D coordintes as discussed in the next section.

This requirement leads to camera calibration. In this section, we discuss the least-squares technique for camera calibration proposed by Tsai [12].

Let (X, Y, Z) be the world coordinate and (x, y) be the corresponding image coordinate. Let P be the transformation matrix which maps world coordinates to image coordinates, i.e.

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{8}$$

or equivalently,

$$\mathbf{q} = P\mathbf{Q} \tag{9}$$

where $\mathbf{q} = [wx\ wy\ w]^T$ and $\mathbf{Q} = [X\ Y\ Z\ 1]^T$ represent homogeneous coordinates in image plane and in 3D respectively. Since P is defined up to a scale factor, the condition $P_{34} = 1$ can be imposed. Consider that a set of $3D$ points (X_i, Y_i, Z_i) and the corresponding image points (x_i, y_i) are known. To estimate the 11 parameters of matrix P, at least 6 points are required. But the measurements $(X_i, Y_i, Z_i, x_i, y_i)$ contain noise so a least squares solution is used with more number of measurements.

D. Distance between two agents

To exploit the spatial distribution of the agents, spatial distances among them are required. Consider the camera setup in Fig. 1 which is a common setup in surveillance applications. The camera is positioned at some height from the floor. Consider the world origin at O_w and camera center at O_c. The floor forms the X-Y plane of the world coordinate system. Let $P1$-$Q1$ and $P2$-$Q2$ be the two agents and the distance between them, i.e. d needs to be calculated.

Let $\mathbf{x}$ be an image point. This image point maps to a ray in space which passes through the camera center. We need at least two points on this ray to get the ray equation. One is O_c which is known because camera is calibrated. The other point can be obtined by $P^+\mathbf{x}$ where P^+ is the pseudo-inverse of P. The point $P^+\mathbf{x}$ will lie on the ray because $PP^+\mathbf{x} = \mathbf{x}$ [13].

Consider two feature points $\mathbf{x}^1$ and $\mathbf{x}^2$ in the image plane corresponding to two individuals with $A = P^+\mathbf{x}^1$ and $B = P^+\mathbf{x}^2$. The ray O_cA can be written as

$$\frac{x - x_c}{x_a - x_c} = \frac{y - y_c}{y_a - y_c} = \frac{z - z_c}{z_a - z_c} \tag{10}$$

where (x_c, y_c, z_c) and (x_a, y_a, z_a) are the coordinates of O_c and A, respectively. The distance measurement is done under the assumption that all agents are of same height which is taken as the average height H equal to $160cm$. The intersections of these rays with the plane $Z = H$ can be found out which gives the 2.5D coordinates of $P1$. For example, 2.5D

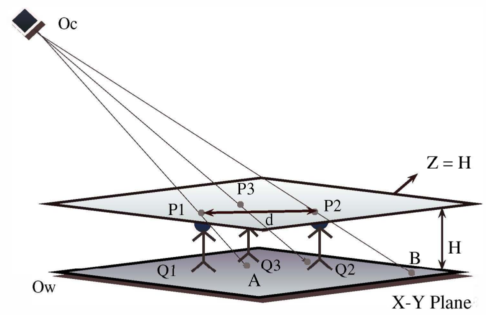


Fig. 1. Experimental setup: Camera center and world coordinate center are at O_c and O_w, respectively. $P1$-$Q1$ and $P2$-$Q2$ represent two agents of height H. The distance between them i.e. d needs to be calculated.

coordinates of $P1$ is given as

$$\begin{aligned} x_{p1} &= \frac{H - z_c}{z_a - z_c}(x_a - x_c) + x_c \\ y_{p1} &= \frac{H - z_c}{z_a - z_c}(y_a - y_c) + y_c \\ z_{p1} &= H \end{aligned} \tag{11}$$

Similar calculations can be done for other agents. Once the agent locations are available, Euclidean distance between any two agents $P1$ and $P2$ can be measured as

$$d = \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2} \tag{12}$$

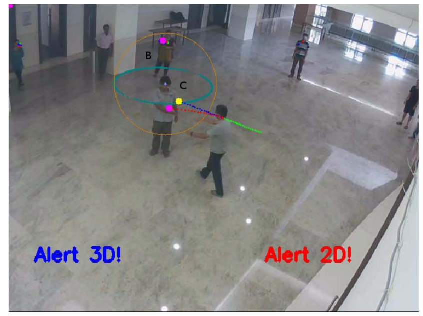
The next section discusses the experimentation under various scenarios.

IV. Experiments and Discussions

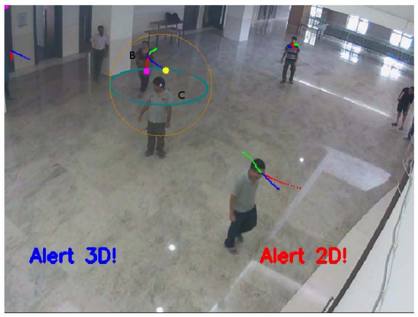
The staged gathering videos are taken inside the IIT campus premise. The videos are created with the help of volunteers. The figures Fig. 2 and Fig. 3 show frames from the staged videos. In the first example, a set of agents approach an agent **C** standing in the center. They approach him one at a time. Let us consider that we are concerned with his security. An alert is sent to security instantaneously if someone violates his personal space. We would like to get an alert early enough to take timely security intervention if needed. To get an early alert, it is required to predict individual trajectories and then space violation. We are predicting agent locations for next 35-40 frames which amounts to around 1.5 seconds. In another example as shown in figure Fig. 3, the group is diverging and then converging. The concerened person is also moving in this case.

Individual participants in the video are represented as feature points in the image. To initialize the feature point, we define a rectangular region on the agent and apply the algorithm given by [14] to select a feature point in that region. We select the same region (for example agent's head) for each agent to make sure that feature points are on the same plane parallel to the ground plane. These points are then tracked using the Lucas-Kanade tracking algorithm [15]. Once the image coordinates of these feature points are available, we find the 2.5D coordinates as discussed earlier. The prediction

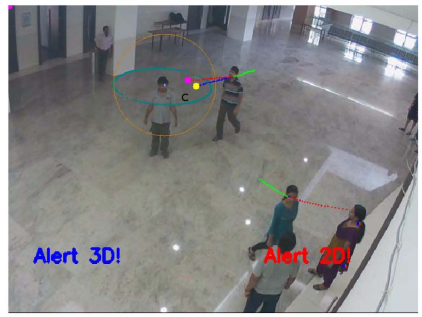
(a) Frame 35



(b) Frame 59



(c) Frame 234

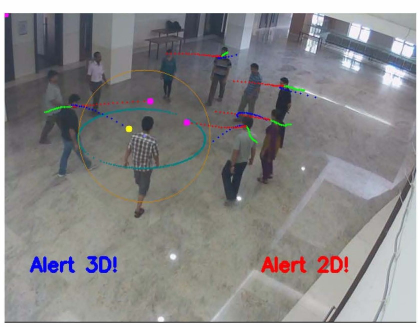


(d) Frame 334

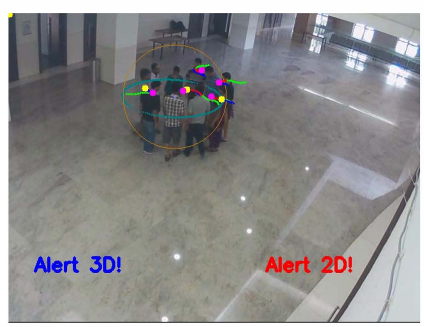
Fig. 2. Results of short-term prediction of space violation: Green lines indicates the data points used for estimating motion parameters. Red and Blue lines are the predicted trajectories in $2D$ and in $2.5D$, respectively. Circle and ellipse represent the personal spaces in $2D$ and in $2.5D$, respectively.

(a) Frame 10

(b) Frame 48



(c) Frame 258



(d) Frame 330

Fig. 3. Another example of short-term prediction of space violation where the group is diverging first and then converging.

is done in the 2.5D coordinate system with the third coordinate fixed as given in (11).

The ellipses and circles in the figures represent the user defined personal spaces of the persons in 2.5D and in 2D, respectively. In 2.5D coordinate system, we define personal space as the area covered by the circle of a predefined radius with the agent's 2.5D point as the center on a plane parallel to ground plane. This circle in space is mapped to an ellipse on the image plane. In the image plane, personal space is simply defined as the circle with the feature point coordinate as the center. The personal space is violated when someone enters inside this circle.

We also compare the prediction done in image plane with the prediction in 2.5D. In Fig. 2, the green trajectory represents the locations of latest M frames used for motion parameter estimation. The red and blue represent the predicted trajectories in the image plane and in 2.5D, respectively. For visualization, 2.5D trajectories are projected back on the image plane using the calibration matrix P. Observe that the distance measurement is better in 2.5D than in 2D. In Fig. 3a and 3b, the agent **B** at the back seems to be inside the 2D personal space region which indicates personal space violation. In reality it is not the case and is rightly depicted the in 2.5D coordinate system as proposed.

V. Conclusion

In this paper, we have presented a simple method for predicting evolution of agents by exploiting the spatio-temporal distribution of the agents. We track agents in a 2.5D coordinate system instead of the image plane. This gives a better judgement of the locations of the agents. Given these locations, we use linear cyclic pursuit based motion model to represent individual motion and hence the trajectory. We analyse the predicted trajectories and attempt to predict the security threat to a person.

References

[1] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 345–357, 2008.

[2] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 935–942.

[3] S. Ali and M. Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–6.

[4] E. L. Andrade, S. Blunsden, and R. B. Fisher, "Modelling crowd scenes for event detection," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 175–178.

[5] W. Choi, K. Shahid, and S. Savarese, "What are they doing?: Collective activity classification using spatio-temporal relationship among people," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1282–1289.

[6] R. Bodor, B. Jackson, and N. Papanikolopoulos, "Vision-based human tracking and activity recognition," in *Proc. of the 11th Mediterranean Conf. on Control and Automation*, vol. 1. Citeseer, 2003.

[7] A. F. M. S. Ali and K. Terada, "A general framework for multi-human tracking using kalman filter and fast mean shift algorithms," *Journal of Universal Computers Science*, vol. 16, no. 6, pp. 921–937, 2010.

[8] E. L. Andrade and R. B. Fisher, "Simulation of crowd problems for computer vision," in *First International Workshop on Crowd Simulation*, vol. 3, 2005, pp. 71–80.

[9] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[10] V. Srikrishnan and S. Chaudhuri, "Crowd motion analysis using linear cyclic pursuit," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 3340–3343.

[11] A. Sinha, "Multi-agent consensus using generalized cyclic pursuit strategies," 2009.

[12] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986*, 1986.

[13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.

[14] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[15] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.