Identifying Behaviors in Crowd Scenes Using Stability Analysis for Dynamical Systems

Berkan Solmaz, Brian E. Moore, and Mubarak Shah

Abstract—A method is proposed for identifying five crowd behaviors (bottlenecks, fountainheads, lanes, arches, and blocking) in visual scenes. In the algorithm, a scene is overlaid by a grid of particles initializing a dynamical system defined by the optical flow. Time integration of the dynamical system provides particle trajectories that represent the motion in the scene; these trajectories are used to locate regions of interest in the scene. Linear approximation of the dynamical system provides behavior classification through the Jacobian matrix; the eigenvalues determine the dynamic stability of points in the flow and each type of stability corresponds to one of the five crowd behaviors. The eigenvalues are only considered in the regions of interest, consistent with the linear approximation and the implicated behaviors. The algorithm is repeated over sequential clips of a video in order to record changes in eigenvalues, which may imply changes in behavior. The method was tested on over 60 crowd and traffic videos.

Index Terms—Video Scene Analysis, Dynamical Systems, Crowd Behaviors.

1 Introduction

VIDEOS of crowd scenes present challenging problems in computer vision. High object-densities in real-world situations make individual object recognition and tracking impractical; understanding crowd behaviors, without knowing the actions of individuals, is often advantageous. Automated detection of crowd behaviors has numerous applications, such as prediction of congestion, which may help avoid unnecessary crowding or clogging, and discovery of abnormal behaviors or flow, which may help avoid tragic incidents. The aim of this particular work is to devise an algorithm that identifies five common and specific crowd behaviors, which we call bottlenecks, fountainheads, lanes, arches, and blocking.

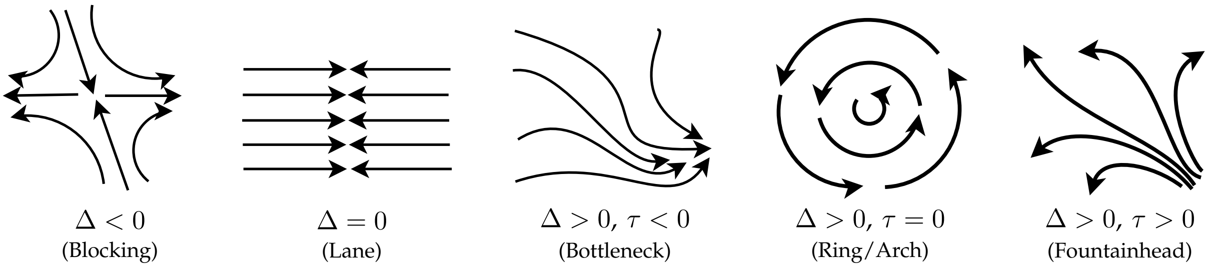
Recent methods for video surveillance (in stations, streets, malls, etc.) lack capabilities to analyze crowd scenes. Limited efforts have addressed problems in high density crowd scene analysis, due to complexity. Most studies are aimed at abnormal behavior detection [1], [2], [3], [4], detecting/tracking individuals in crowds [5], [6], [7], [8], [9], counting people in crowds [10], [11], [12], identifying different regions of motion and segmentation [13], [14], [15], [16], or crowd detection [17], rather than identifying collective crowd behaviors. For a more comprehensive review of behavior analysis in crowd scenes, see the survey [18].

Conventionally, activity analysis and scene understanding involve object detection, tracking and behavior recognition. This approach, requiring low-level motion [19], [20], appearance features [21], [20], or object trajectories [22], [19], performs well in scenes with low object density, but fails in real-world higher density crowd scenes. Tracking is also a hurdle; methods are often not suitable for multiple target tracking, due to computational expense or unreliability. Recent work [23] uses motion trajectories of multiple objects to learn models for segmentation of group motion patterns, but all these approaches require training.

Thus, researchers proposed a holistic approach for activity analysis and scene understanding, which avoids tracking and uses features directly, rather than computing trajectories for representing activities. This approach requires features such as multi-resolution histograms [2], spatio-temporal cuboids [1], appearance or motion descriptors [24], [25] and spatio-temporal volumes, etc. [26]. In another work [3], a representation based on dynamic textures, in which appearance and dynamics are modeled jointly, is used for detecting anomalies in crowd scenes. Though this approach is suitable for recognizing actions and detecting/segmenting activities, it also requires training and manual labeling of activities.

We present a method that combines low-level local motion features, computed by optical flow, with high-level information, obtained by analyzing regions of interest in the scene. It performs well in various crowd scenes as it does not involve object detection and tracking, which may be unreliable in crowd scenes. Our method does not require training, as needed in most current approaches for human action recognition. Our approach is not restricted to isolated activities, but is able to identify multiple behaviors in a single scene. Using a Lagrangian particle dynamics model [27], [28] of crowd scenes, crowds are treated as collections of mutually interacting particles, so our

- *B. Solmaz and M. Shah are with the Department of Electrical Engineering and Computer Science, University of Central Florida, 4000 Central Florida Blvd, Orlando, FL 32816.*
E-mail: bsolmaz@eecs.ucf.edu, shah@eecs.ucf.edu.
- *B.E. Moore is with Department of Mathematics, University of Central Florida, 4000 Central Florida Blvd, Orlando, FL 32816.*
E-mail: bmoore@math.ucf.edu.



$\Delta < 0$ (Blocking)

$\Delta = 0$ (Lane)

$\Delta > 0$, $\tau < 0$ (Bottleneck)

$\Delta > 0$, $\tau = 0$ (Ring/Arch)

$\Delta > 0$, $\tau > 0$ (Fountainhead)

Fig. 1. Five flows corresponding to Δ and τ, along with the related crowd behaviors.

method is well-suited for small and large crowds [29].

Closely related work [7], [30] uses Lagrangian particle dynamics, based on optical flow, for analyzing flow in crowd scenes. The method of [30] uses the learned normal behavior in the scene to detect abnormal behavior, but our method is concerned with detecting and identifying particular crowd behaviors. The method of [7] uses the flow of the crowd to aid in tracking individuals, but the method is specific to dense crowds with uniform motion, and our method is less restrictive on scene type, requiring only a characteristic flow. Finally, [31], [32] present methods for learning motion patterns in crowd scenes, but our method locates specific instances of crowd behavior and does not require learning the typical flow.

In other related work, description of orientation fields by phase portraits [33] and detection of critical points [34] are proposed, and [35] introduced a fluid-dynamic model for simulating the movement of pedestrians and showed the phenomenon of lane formation which may occur in dense real crowds. [36] analyzed the spread of particles near the singularities regarding the problem of oceanic diffusion. Yet, none of these studies connect flow fields with crowd behaviors; to the best of our knowledge, this is the first attempt in computer vision to identify specific crowd behaviors. Our main contributions are: (1) Use of linear dynamical systems and the Jacobian matrix, defined by the optical flow, to analyze crowd flow in videos. (2) A local analysis that detects/identifies specific global behaviors i.e. bottlenecks, fountainheads, lanes, arches, and blocking. (3) A simple, yet effective, method to detect these behaviors, that does not require object detection, tracking, nor training. (4) A modular framework that can identify multiple crowd behaviors in one scene.

2 Behaviors from Dynamical Systems

2.1 Stability Analysis

Consider a continuous dynamical system

$$\dot{w} = F(w), \tag{1}$$

with $w(t) = [x(t), y(t)]^T$ and $F(w) = [u(w), v(w)]^T$. Here, x and y are particle positions, and u and v represent particle velocities in the x and y directions, respectively. (In our application to video sequences, u and v are obtained from optical flow.) A first step in understanding solution behavior for (1) is finding critical points w^* such that $F(w^*) = 0$. Behavior of trajectories near a point w^* is determined by linearizing the system about w^*. To find a linearization (see, for example, [37]) let $z = w - w^*$, which means $\dot{z} = \dot{w} = F(w) = F(w^* + z)$. By Taylor's theorem

$$F(w^* + z) = F(w^*) + J_F(w^*)z + \mathcal{O}(z^2), \tag{2}$$

where J_F denotes the Jacobian matrix for F,

$$J_F = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}, \tag{3}$$

$F(w^*) = 0$ implies a linearization of (1) about w^*,

$$\dot{z} = J_F(w^*)z. \tag{4}$$

The solutions of (4) are completely defined by the initial conditions and the eigenvalues of the matrix J_F, which are solutions of a characteristic equation $\lambda^2 - \tau\lambda + \Delta = 0$, where τ is the trace and Δ is the determinant of the matrix. It is easy to show that

$$\lambda_{1,2} = \frac{1}{2}\left(\tau \pm \sqrt{\tau^2 - 4\Delta}\right) \tag{5}$$

with

$$\Delta = \lambda_1\lambda_2 \quad \text{and} \quad \tau = \lambda_1 + \lambda_2, \tag{6}$$

where λ_1 and λ_2 are the eigenvalues, yielding important information about the flow, as depicted in Fig. 1.

$\Delta < 0$ implies w^* is a saddle, and particle trajectories are pulled toward the point in two directions, but pushed away in other directions.

$\Delta = 0$ implies at least one eigenvalue is zero, and critical points are non-isolated.

$\Delta > 0$ implies the eigenvalues are real with same sign or complex conjugates. If $\tau < 0$, then w^* is stable, acting as a sink for nearby trajectories. If $\tau > 0$, then w^* is unstable, acting as a source for nearby trajectories. Purely imaginary complex conjugate eigenvalues ($\tau = 0$) implies w^* is a center and near-by trajectories orbit the point indefinitely.

It is important to notice that the overall flow in a crowd scene can not be expected to conform to the dynamical system (4), with global flow patterns depicted in Fig. 1.

This may be understood by noting that there is not one global function F that defines the entire flow, but instead each particle has a function F defining its motion. As a result, we can only expect the flow patterns of Fig. 1 to represent the crowd flow locally, but there are global aspects of crowd flow that may be recognized by such a comparison.

2.2 Crowd Behaviors

Now consider the flows arising from J_F, as depicted in Fig. 1, in connection with specific crowd behaviors.

Bottlenecks. If $\Delta > 0$ and $\tau < 0$, then particle trajectories from many points converge to one location, i.e. many pedestrians or vehicles from various locations enter through one narrow passage. Hence, we define a *bottleneck* to be the mouth of any narrow passage through which pedestrians regularly pass. This liberal definition allows consideration of many flows. Yet, it makes no distinction between bottlenecks that occur in normal situations and those that result in clogging, typical of panic situations when many people simultaneously try to exit through one narrow passage. Though the present framework does not enable detection of panic, it may identify clogged bottlenecks as a special case of the blocking behavior.

Fountainheads. When $\Delta > 0$ and $\tau > 0$, particle trajectories diverge from one location. This behavior is noticed when pedestrians leave a narrow passage, persisting in many separate directions, and we call the mouth of such a passage a *fountainhead*. This behavior is the opposite of a bottleneck, so fountainheads are detected as bottlenecks in backward time.

Lane Formation. In crowd situations, lanes of flow in opposite directions naturally form, as pedestrians moving against the flow step aside to avoid collision and end up moving with other pedestrians with the same general direction and speed. In such instances, the motion near an individual is negligible, relative to other nearby individuals, because they are all moving together. This is precisely the behavior we see in what we call a *lane*, and the behavior is well-described by non-isolated critical points, rendering $\Delta = 0$ along the path of the lane. Clearly, $\Delta = 0$ if the objects in the scene are stationary, then the optical flow is zero. But, we are not interested in this trivial case, and it is distinguished from the case in which many pedestrians or vehicles are moving at the same speeds in the same direction (a straight line). In addition, it should be noted that a single object moving in a unique direction is not considered a lane.

Ring/Arch Formation. Motion described by $\Delta > 0$ and $\tau = 0$ is characteristic of crowd flow that is curved or circular. This behavior may be typical of a crowd scene in which pedestrians must maneuver around obstacles, forming an *arch*. It may also be observed in less typical scenes such as people dancing or traffic in a round-about, forming a *ring*. In either case, the eigenvalues of the Jacobian matrix are complex conjugates, and we look for this eigenvalue response along oblique paths over which many trajectories may pass.

Blocking. Local flows in which particles are bouncing off of each other in somewhat random directions, unable to proceed in the direction desired, is represented by $\Delta < 0$. This is characteristic behavior of people in densely populated scenes where the surrounding crowd prevents the desired motion of many individuals. We define this behavior as *blocking*, because pedestrians moving in opposite directions block each other as crowd density increases, preventing advancement from either group. In some situations the density of the crowd may lead to gridlock and no particle motion, in which case the optical flow is zero. These instances can still be recognized as blocking because they are always preceded by some type of regular flow. In other words, regions with regular movement in a high density crowd that become void of motion are best explained by blocking.

3 Implementation

A key factor of the analysis described in Section 2.1 is location of a critical point. In video scene analysis, we locate, instead, a *region of interest* (ROI), which locally corresponds to a critical point, and check the eigenvalues of J_F at points in these regions. For multiple behaviors in a scene we have multiple ROI. It is possible that not all ROI have significance for understanding the behaviors in the scene, but this can be determined through J_F. This framework constitutes two main tasks, which may be executed in parallel and are summarized by the following computations.

-*Regions of interest* (ROI) are defined according to the behaviors we observe, and thereby consist of one of the three following locations.

- Candidate Points for Bottlenecks/Fountainheads
- Candidate Paths for Lanes and Rings/Arches
- Candidate Precincts for Blocking

-*Eigenvalue maps* are defined by the signs of Δ and τ for each point in a ROI.

Our method is based on advection of particles by (1), defined by the optical flow, and implementation begins with computation of Lucas-Kanade optical flow [38]. Since optical flow corresponding to some individuals may be different than the general crowd flow, we apply median filter. (For an image of 360x480 typical size of median filter is 40x40.) A flow chart describing implementation of our method is depicted in Fig. 2, showing computation of ROI and eigenvalue maps from the optical flow. It is essential to complete both tasks, because identifying the given behaviors is not possible using the particle trajectories alone, and the eigenvalues loose their significance without the ROI, which correspond to critical points of the dynamical system. Once these tasks are completed for a given video clip, the type of flow observed in the scene is determined by checking the eigenvalue map

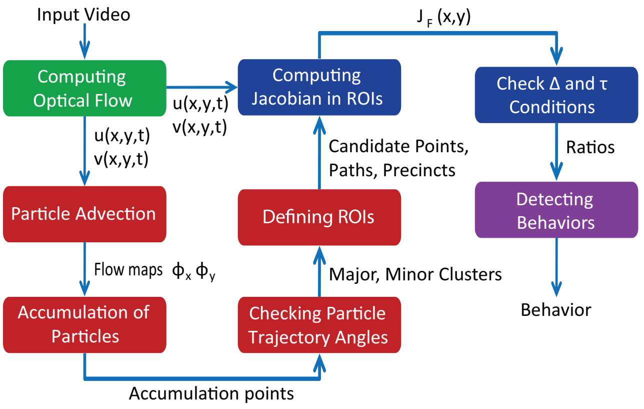


Fig. 2. Overview of the Framework.

in ROI. Details on implementation are provided in Section 3.1 and 3.2, respectively, and the implementation process is fully depicted in Figs. 3, 4, and 5.

Notice, our definition of a ROI is not equivalent to the definition of a critical point as given by dynamical systems theory and given by w^* in Section 2.1. This is because we do not want to consider *all* points where the optical flow is zero. Instead we consider points that are affected by the flow in ways that are consistent with the behaviors considered, and this is consistent with our set-up, as the system (4) can only be used to give local information about the flow. To be more specific, $F(w^*) = 0$ is nearly satisfied in 1) regions with blocking, as the motion of the individuals is prohibited, 2) bottleneck and fountainhead regions, as those points act as sinks or sources for the flow, and 3) the paths of lanes, as the motion of an individual is nearly zero relative to other near-by individuals.

To locate changes in behavior, the total sequence is divided into clips of fixed length, and the algorithm is repeated. Comparing sequential clips and recording appropriate changes in the eigenvalues reveals changes in behavior. This is demonstrated in Fig. 5.

3.1 Regions of Interest

ROI are necessary for finding possible locations of scene behaviors, for removing false positives, and for reducing the amount of computations. We describe computation of the three types of ROI, but common to computation of each ROI is particle advection and the resulting accumulation of particles, which is explained first.

Particle Advection. A grid of particles is overlaid on the initial frame and advected with the flow. Numerically solving the system of equations (1), using

$$w(t+1) = w(t) + F(w(t)), \qquad (7)$$

yields the particle positions over the time interval $[t_0, t_f]$. The evolution of particles through the flow is tracked using particle flow maps

$$\phi_{t_0}^t(w) = w(t; t_0, w_0), \qquad (8)$$

which simply indicate the relation between the initial particle positions and their positions at time $t \in [t_0, t_f]$. We use $P(\phi_{t_0}^t(w))$ to denote the particle corresponding to a particular flow map. The flow maps are initialized in the first frame of the video. As the particles evolve through the flow field in time, they are accumulated in particular regions of the scene.

Accumulation Points. Particles accumulate at bottlenecks or at the ends of lanes resulting in higher particle densities, which are calculated using flow maps. If $D(w_f)$ denotes the density map for particles at position w_f, then it is equal to the cardinality of the set containing all particles at that point, i.e.

$$\mathcal{P}_{w_f} = \{P(\phi_{t_0}^{t_f}(w)) \,|\, w = w_f\} \;\Rightarrow\; D(w_f) = |\mathcal{P}_{w_f}|. \qquad (9)$$

A Gaussian filter is applied to the density map to obtain blobs of high particle densities; the variance of the typical gaussian filter is 1 and the size is 11x11 pixels. The centroids of the blobs, which are the local peaks of the density map, are clustered using mean-shift algorithm [39]. (This approach results in a significant increase in speed over using mean-shift directly on the density map.) The number of particles in each cluster is a significance measure for that cluster, and significant cluster centers are defined as accumulation points.

Finding accumulation points and using them to locate behaviors, may be considered similar to a sink seeking process. For instance, [7] uses particle trajectories to find preferred directions of motion in crowd scenes by finding sinks, which are typically preferred exits or frequently visited regions of the scene. Our approach differs significantly, because we do not use information from neighboring particles to make conclusions about particle paths, meaning our approach is applicable to crowd flows of varying densities, provided there is a characteristic flow. In addition, sink seeking alone implies nothing about types of flow, which is our main concern.

Candidate Points, Paths, and Precincts. After particle advection, trajectories should satisfy two criteria: (a) final position of a particle must be close to an accumulation point, (b) distance between initial and final particle positions should be long enough. This is essential for selecting particles that describe the motion and discarding others. For instance, a trajectory not reaching an accumulation point does not satisfy (a), and a non-motion particle does not satisfy (b).

Trajectories of particles satisfying both criteria are analyzed, accounting for two possibilities as follows.

C1 Particles accumulate unidirectionally.
C2 Particles accumulate multidirectionally.

C1 implies the trajectory region is a candidate path; C2 implies accumulation points are candidate points.

Let d_0 and d_f be unit vectors for particle directions at times t_0 and t_f, respectively. For every trajectory, the angles between vectors $\theta = \arccos(d_0 \cdot d_f)$, are clustered by Mean-shift algorithm [39] and clusters are

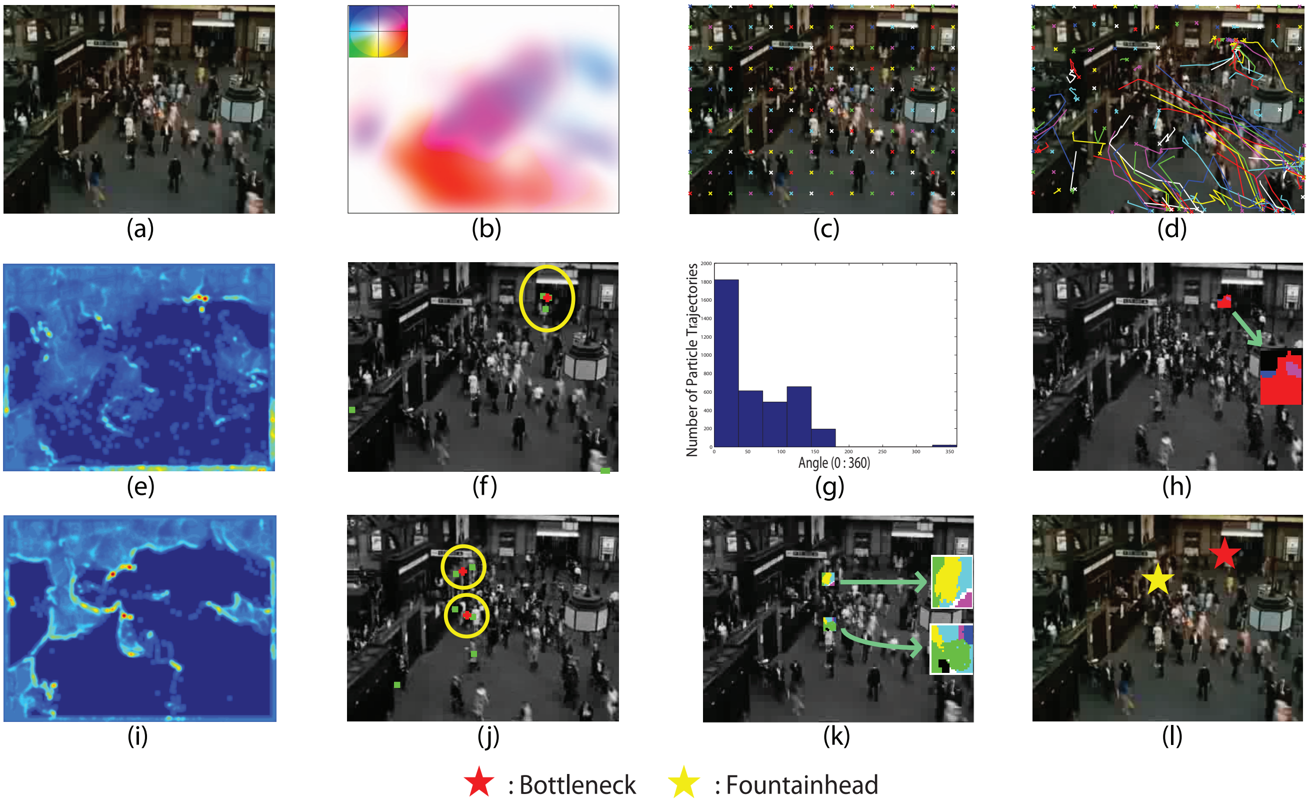


Fig. 3. Process for detecting a Bottleneck and a Fountainhead: (a) Given a video scene, (b) compute optical flow, (c) overlay scene with a grid of particles, (d) advect particles according to the flow, (e) particles accumulate producing a density map, (f) local peaks (green) of the density map are clustered, centroids (red) of clusters are accumulation points, (g) particle trajectories around accumulation points are clustered according to angles, (h) candidate points are determined; these points are checked for bottlenecks using the eigenvalue map, and a bottleneck is correctly detected. (i,j,k) Same approach in backward time enables the correct identification of a fountainhead, (l) behaviors are labeled a bottleneck and a fountainhead with a red and yellow star, respectively.

categorized as major or minor. Typically, a major (resp. minor) cluster contains at least one third (resp. one tenth) of the total number of trajectories at an accumulation point. C1 implies the trajectories correspond to at most two major clusters. C2 implies there are several minor clusters. Since one (possibly two) lane may end at an accumulation point, a candidate path is defined by particle trajectories for major clusters, but several different lanes ending at an accumulation point are better defined by a bottleneck. Thus, an ROI for a bottleneck or fountainhead is the area around a candidate point, which is an accumulation point with three or more minor clusters. Finally, high density regions are labeled candidate precincts, because further increases in density may lead to blocking.

It is conceivable that problems may be encountered at this step due to perspective effects. To clarify, notice that a scene with traffic flow on a long highway, extending into the distance, will appear to have particle trajectories converging at a point in the distance. According to our set-up, the accumulation point may be falsely labeled as a candidate point for a bottleneck, but the behavior is clearly a lane. However, we did not encounter such problems during implementation.

3.2 Eigenvalue Map

To reduce noise and neglect regions without motion, we start by discarding small magnitude optical flow. Then the optical flow is averaged in time and we apply a median filter in space, giving a representation of optical flow for the entire sequence or clip, denoted $(\tilde{u}, \tilde{v})$. Considering only pixels in a given ROI, we analyze the eigenvalues of J_F at each pixel through Δ and τ. Using δ_x and δ_y to denote difference operators in each spatial direction, we compute Δ and τ using

$$\Delta = \delta_x\tilde{u} \cdot \delta_y\tilde{v} - \delta_y\tilde{u} \cdot \delta_x\tilde{v} \qquad (10)$$

$$\tau = \delta_x\tilde{u} + \delta_y\tilde{v}. \qquad (11)$$

Inside a ROI, pixels are colored according to the eigenvalues (up to a tolerance ϵ) as shown in Table 1, and we choose $\epsilon = 0.005$ in practice. The number of pixels satisfying each condition is counted, as shown in Table 1, and we call the total number of pixels in the ROI T. We determine if a ROI is dominated by a behavior using the ratio conditions in Table 2.

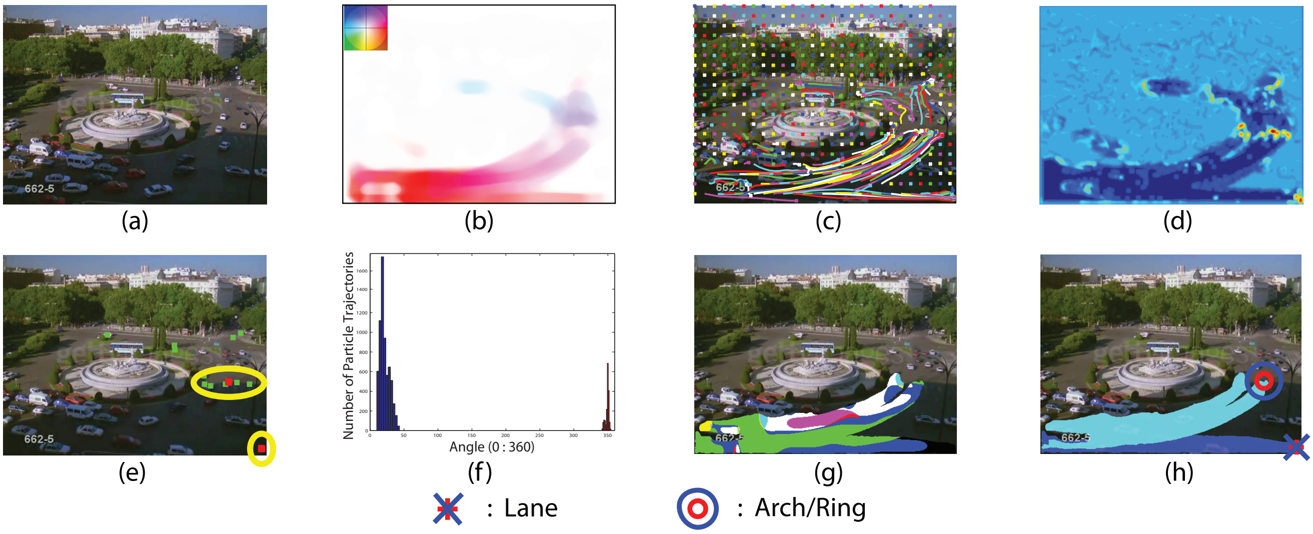


Fig. 4. Process for detecting a Lane and an Arch: Steps are the same as Fig. 3. Angles of trajectories (f) around accumulation points (e) reveal two candidate paths. According to the eigenvalue map (g) along each path, an arch and a lane are correctly detected and labeled (h).

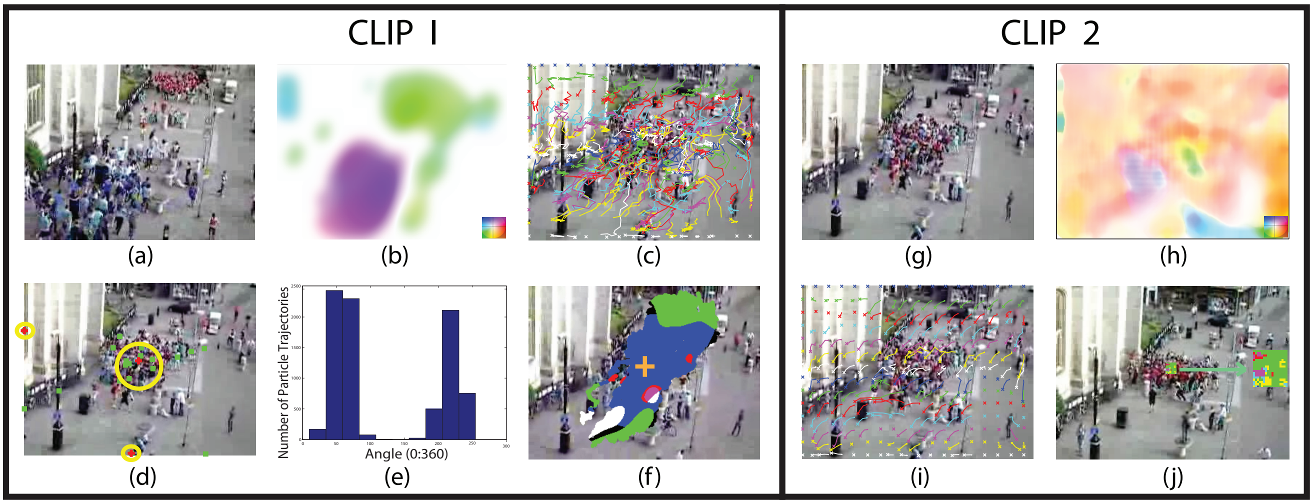


Fig. 5. Process to detect Blocking: A sequence is divided into sequential clips and the process in Figs. 3 and 4 is applied to each clip. In Clip 1 two opposing lanes are detected, as the angle between them is near $180°$ (e), so a candidate precinct (f) is saved for Clip 2. The process is repeated in Clip 2; the eigenvalue map around the candidate precinct (j) shows a majority of points with $\Delta < 0$, and blocking is correctly detected.

4 Experimental Results

The method was tested on real video sequences downloaded from the web (Getty-Images, BBC Motion Gallery, Youtube, Thought Equity) and on sequences from the Performance Evaluation of Tracking and Surveillance (PETS) 2009 Dataset, representing crowd and traffic scenes. The number of overlaid particles is equal to the number of pixels. The videos have different fields of view, resolutions, frame rates, and duration, yet our method performed well in most cases. Performance was measured on more than 60 video sequences, which contain single or multiple behaviors, as shown in Table 3.

To evaluate method performance, we compared detection against manually generated groundtruth, consisting of points for bottlenecks, fountainheads and blockings, and regions for lanes and arches on all videos, and the results are shown in Table 3. (The ground truth was manually generated for each video by an independent computer vision researcher, based on the behavior definitions provided in Section 2.2.) Following the PASCAL VOC challenge [40], detection accuracy is based on overlap of the detected region and groundtruth. For lanes and arches we require an overlap of more than 40%, a relaxation of the Pascal measure appropriate for our problem. Similarly, the region around points that identify bottlenecks, fountainheads, or blocking is required to overlap with the

TABLE 1
Eigenvalue responses and designated labels. *Count* is the number of pixels in a ROI satisfying a condition. $\Delta > \epsilon^2$ for each condition unless stated otherwise.

Eigenvalues	Condition	Label	Count
real, $\lambda_1 > 0$, $\lambda_2 < 0$	$\Delta < -\epsilon^2$	Green	G
real, both positive	$\tau < -2\epsilon$, $\tau^2 > 4\Delta$	Red	R
real, both negative	$\tau > 2\epsilon$, $\tau^2 > 4\Delta$	Yellow	Y
complex conjugate, positive real part	$\tau < -2\epsilon$, $\tau^2 < 4\Delta$	Magenta	A
complex conjugate, negative real part	$\tau > 2\epsilon$, $\tau^2 < 4\Delta$	Cyan	C
purely imaginary	$\lvert\tau\rvert < 2\epsilon$	White	W
at least one zero	$\lvert\Delta\rvert < \epsilon^2$	Blue	B
all zero (no motion)	$J_F = 0$	Black	K

TABLE 2
Ratio conditions determine dominance of a ROI by an eigenvalue response, corresponding to a behavior. (Tolerance L is chosen through experimentation.)

Identified Behavior	Ratio Condition
Lane	$B/T > L$
Blocking	$G/T > L$
Bottleneck	$(R + A)/T > L$
Fountainhead	$(Y + C)/T > L$
Arch/Ring	$(W + A + C)/T > L$

analogous region from groundtruth; we require that the Euclidean distance between the detected point and groundtruth be sufficiently small, typically within 40 pixels. Fig. 6 shows ROC curves with True Positive Rate vs False Positive Per Video for four behaviors obtained by varying the tolerance L from Table 2.

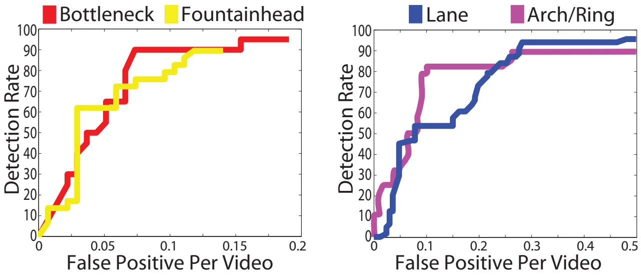


Fig. 6. ROC curves for four behaviors.

5 Conclusion and Future Work

We have proposed a framework to identify multiple crowd behaviors (bottlenecks, fountainheads lanes, arches, and blocking) through stability analysis for dynamical systems, without the need of object detection, tracking or training. Results are illustrated in Fig. 7, demonstrating the capability and flexibility of the method for a wide variety of scenes. In light of its strengths, the method does have shortcomings, which are listed here and open for future work. Our model is deterministic and can not capture the randomness inherent in the problem without a stochastic component. Our model can only identify five behaviors, which is an oversimplification of the complexities encountered in crowds. Our method is not useful when significant overlap of motion patterns is present in the scene, or when there is lack of consistent characteristic flow.

TABLE 3
Crowd Behavior Detection Results

Behavior	Total # of Behaviors	# of Detections	# of Missed	# of False
Lane	66	56	10	11
Blocking	3	3	0	0
Bottleneck	20	16	4	3
Fountainhead	29	23	7	5
Arch/Ring	28	23	5	6

Acknowledgments

This research was supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-09-1-0255.

References

[1] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009.*

[2] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004.*

[3] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition, 2010. CVPR 2010.*

[4] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition, 2010. CVPR 2010.*

[5] G. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *Computer Vision and Pattern Recognition, 2006. CVPR 2006.*

[6] L. Kratz and K. Nishino, "Tracking with local spatio-temporal motion patterns in extremely crowded scenes," in *Computer Vision and Pattern Recognition, 2010. CVPR 2010.*

[7] S. Ali and M. Shah, "Floor fields for tracking in high density crowd scenes," in *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, 2008, pp. 1–14.

[8] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," *2009 IEEE 12th International Conference on Computer Vision*, 2009.

[9] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004.*

[10] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Computer Vision and Pattern Recognition, 2006. CVPR 2006.*

[11] D. Yang, H. Gonzalez-Banos, and L. Guibas, "Counting people in crowds with a real-time network of simple image sensors," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 13-16 2003, pp. 122 –129 vol.1.

[12] A. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *Computer Vision, 2009 IEEE 12th International Conference on*, sept. 2009, pp. 545 –551.

[13] A. B. Chan and N. Vasconcelos, "Mixtures of dynamic textures," in *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005, pp. 641–647.

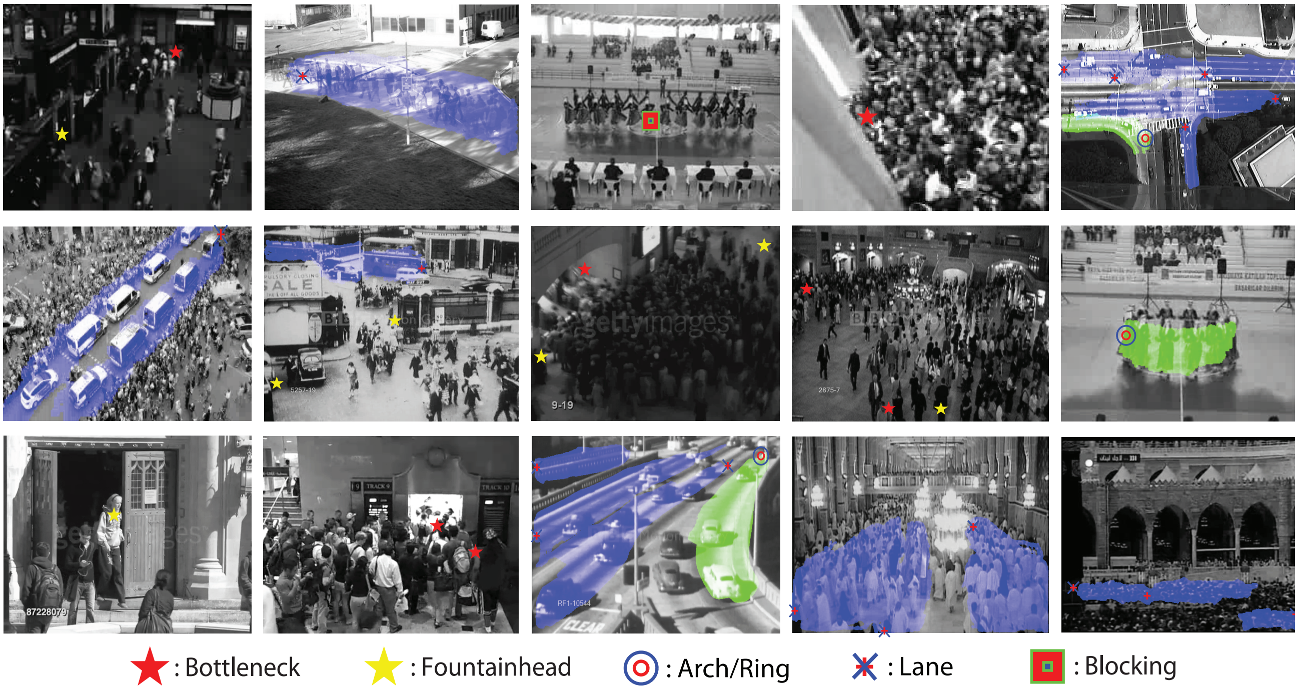


Fig. 7. Scenes from 15 real video sequences, each showing the behaviors that are detected by the method.

[14] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," in *Computer Vision and Pattern Recognition, 2006. CVPR 2006.*

[15] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, and T. Yu, "Unified crowd segmentation," in *ECCV*, 2008.

[16] J. Wright and R. Pless, "Analysis of persistent motion patterns using the 3d structure tensor," in *Motion and Video Computing, 2005. WACV/MOTIONS '05. IEEE Workshop on*, vol. 2, jan. 2005, pp. 14 –19.

[17] R. Pini, M. Ofer, A. Shai, and S. Amnon, "Crowd detection in video sequences," *IEEE Intelligent Vehicles Symposium 2004*, pp. 66–71, 2004.

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

[19] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 747 –757, aug 2000.

[20] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, vol. 2, 13-16 2003, pp. 734–741.

[21] J. S. Marques, P. M. Jorge, A. J. Abrantes, and J. M. Lemos, "Tracking groups of pedestrians in video sequences," *Computer Vision and Pattern Recognition Workshop*, vol. 9, p. 101, 2003.

[22] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *BMVC '95: Proceedings of the 6th British conference on Machine vision (Vol. 2)*. BMVA Press, 1995, pp. 583–592.

[23] R. Li and R. Chellappa, "Group motion segmentation using a spatio-temporal driving force model," in *Computer Vision and Pattern Recognition, 2010. CVPR 2010.*

[24] E. L. Andrade, S. Blunsden, and R. B. Fisher, "Modelling crowd scenes for event detection," in *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*. IEEE Computer Society, 2006, pp. 175–178.

[25] M. Pittore, M. Campani, and A. Verri, "Learning to recognize visual dynamic events from examples," *Int. J. Comput. Vision*, vol. 38, no. 1, pp. 35–44, 2000.

[26] I. Laptev, "On space-time interest points," *Int. J. Comput. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[27] S. Ali and M. Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis."

[28] G. Haller and G. Yuan, "Lagrangian coherent structures and mixing in two-dimensional turbulence," *Phys. D*, vol. 147, no. 3-4, pp. 352–370, 2000.

[29] R. L. Hughes, "A continuum theory for the flow of pedestrians," *Transportation Research Part B: Methodological*, vol. 36, no. 6, pp. 507–535, 2002.

[30] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *CVPR*, 2009, pp. 935–942.

[31] M. Hu, S. Ali, and M. Shah, "Learning motion patterns in crowded scenes using motion flow field," in *International Conference on Pattern Recognition, 2008. ICPR 2008.*

[32] P. Widhalm and N. Brandle, "Learning major pedestrian flows in crowded scenes," *International Conference on Pattern Recognition, 2010. ICPR 2010.*

[33] A. R. Rao and R. C. Jain, "Computerized flow field analysis: Oriented texture fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 693–709, July 1992.

[34] R. Ford, "Critical point detection in fluid flow images using dynamical system properties," *Pattern Recognition*, vol. 30, no. 12, pp. 1991–2000, 1997.

[35] D. Helbing, "A Fluid Dynamic Model for the Movement of Pedestrians," *Complex Systems*, vol. 6, pp. 391–415, May 1992.

[36] A. Okubo, "Horizontal dispersion of floatable trajectories in the vicinity of velocity singularities such as convergencies," *Deep Sea Res*, vol. 17, pp. 445–454, 1970.

[37] S. H. Strogatz, *Nonlinear dynamics and chaos : with applications to physics, biology, chemistry, and engineering*. Addison-Wesley, 1994.

[38] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: Proceedings of the 7th international joint conference on Artificial intelligence*, 1981, pp. 674–679.

[39] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 32–40, 1975.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.