Journal of Circuits, Systems, and Computers
Vol. 26, No. 7 (2017) 1750110 (50 pages)
© The Author(s)
DOI: 10.1142/S0218126617501109

An Eigenvalue Approach to Detect Flows and Events in Crowd Videos*

Md. Haidar Sharif

Computer Sciences and Engineering,
Faculty of Engineering and Natural Sciences,
International University of Sarajevo,
Hrasnička cesta 15, 71210 Ilidža, Sarajevo, Bosnia and Herzegovina
haidar@ius.edu.ba

Received 13 April 2016
Accepted 9 December 2016
Published 17 February 2017

Analysis of flows in crowd videos is a remarkable topic with practical implementations in many different areas. In this paper, we present a wide overview of this topic along with our own approach to this problem. Our approach treats the difficulty of crowd flow analysis by distinguishing single versus multiple flows in a scene. Spatiotemporal features of two consecutive frames are extracted by optical flows to create a three-dimensional tensor, which retains appearance and velocity information. Tensor's upper left minor matrix captures intensity structure. A normalized continuous rank-increase measure for each frame is calculated by a generalized interlacing property of the eigenvalues of these matrices. In essence, measure values put through the knowledge of existing flows. Yet they do not go into effect desirably due to optical flow estimation error and some other factors. A proper set of the degree of polynomial fitting functions decodes their existence. But how can we estimate that set? Its detailed study is performed. Zero flow, single flow, multiple flows, and interesting events are detected as frame basis using thresholds on the polynomial fitting measure values. Plausible mean outputs of recall rate (88.9%), precision rate (86.7%), area under the receiver operating characteristic curve (98.9%), and accuracy (92.9%) reported from conducted experiments on PETS2009 and UMN benchmark datasets make clear and visible that our method gains high-quality results to detect flows and events in crowd videos in terms of both robustness and potency.

Keywords: Accuracy; anomaly; eigenvalue; optical flow; polynomial fitting; precision.

1. Introduction

Analysis of crowd flows and events is a key point to ensure public safety, automatic detection of anomalies, estimating of number of people entering into public and

*This paper was recommended by Regional Editor Tongquan Wei.

private spaces. As a result, a great variety of approaches along with their contemporary and comprehensive surveys[1–7] can be found to deal with crowd flow analysis and behavior understanding. In computer vision research, those approaches include people tracking, crowd counting, crowd density estimation, numerous events detection, validation, and simulation. But available surveys implicitly categorize flows and events understanding on crowd videos. First of all, in a tabular fashion, as shown in Table 1, we have created a more focused and structured yet short review of recent publications related to the detection of flows and events in crowd videos. The area under the receiver operating characteristic (ROC) curve, equal error rate, accuracy, and the number of individual false alarms noncontiguous to positive ground truth have been abbreviated as AUC, EER, ACC, and ϖ, respectively. A larger AUC bespeaks a better performance in robustness of positive ground truth events detection, whereas a larger ACC shows the effectiveness of detection. However, we have categorized the up-to-date algorithms on the basis of flow models, which consist of pixel-wise and patch-wise flows. In pixel-wise flow, flow vectors of a pixel are considered as either two consecutive frames only or more than two frames until its clear existence in the scene. In patch-wise flow, flow vectors of a patch or block are estimated as two consecutive frames using either phase correlation or other related techniques. Both pixel-wise and patch-wise flow models are further divided into supervised, semi-supervised, and unsupervised learning-based approaches. Supervised learning-based approach supposes the available training dataset should be labeled for both normal and anomalies cases. Semi-supervised learning-based approach expects the available training dataset or thresholds should be labeled only for normal cases. Unsupervised learning-based approach does not count on training data.

Main gain of supervised learning-based approaches[8–15] clearly lies in the fact that normal and abnormal cases are known precisely. But it is essential to get all normal and abnormal models before detection. Their others pros and cons depend on the selected features of individual methods. For instance, the tracking results of Tracklets[10] were penalized by incomplete fragments of trajectories, object missing, and new object entering. Hence 13 false alarms were encountered from UMN[16] dataset. In ShearForce,[13] interaction force was estimated directly from shear force in viscous fluid by studying peculiarity of local motion patterns. If interaction force estimation can assess erroneously, false alarms are increased. ShearForce[13] model was validated by randomly selecting 40% frames from each scenario for parameter optimizing during the learning procedure, and used the rest for testing. The reliable learning of unknown parameters using random samples was not precisely possible and thus eight false alarms were reported from UMN[16] dataset. Semi-supervised learning approaches concerning both pixel-wise flow[17–35] and patch-wise flow[36–38] do not require labels for anomaly cases. Accordingly, their applications are more wider than those of supervised approaches. For semi-supervised learning approaches Table 1 hints that as compared to pixel-wise flow-based models, patch-wise flow-based models possess a higher tendency to turn real-time operational algorithms. But

Table 1. The qualitative and quantitative comparison among the state-of-the-art detection methods regarding flows and events in crowd videos. "✓" and "–" signify corresponding intents included and excluded, respectively.

Years	State-of-the-art methods	Flow field models						Foremost features										Functions				Benchmark datasets					
			Pixel-wise		Learning			Low level						Mid level				Events				UMN[16]		PETS2009[49]		UCSDped1[50]	
																						Performance		Performance		Performance	
		Patch-wise	2 frames	2+ frames	Supervised	Semi-supervised	Unsupervised	Corner	Segment	Texture	Intensity	Gradient	Motion	Histogram	Volume	Trajectory	Ratio	Detecting	Locating	Real-time	ϖ	AUC	ACC	AUC	ACC	EER	AUC
2008	HOG3D[8, 13]	–	✓	–	✓	–	–	✓	–	–	✓	✓	✓	✓	✓	–	✓	✓	–	–	11	0.771	73.3%	0.728	77.2%	–	–
	HOG/HOF[9, 13]	–	✓	–	✓	–	–	✓	–	–	✓	✓	✓	✓	✓	–	✓	✓	–	–	7	0.875	84.5%	0.785	86.6%	–	–
	DirectionMap[17, 25]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	✓	–	–	✓	✓	–	✓	9	–	–	–	–	–	–
2009	Cuboid3D[13, 18]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	✓	–	–	✓	✓	–	10	0.801	73.8%	0.748	79.2%	–	–
	RankIncrease[19]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	–	–	✓	✓	–	–	–	–	–	–	–	–	–
	SocialForce[13, 20, 30]	–	✓	✓	–	✓	–	✓	–	–	✓	✓	✓	–	✓	✓	–	✓	✓	–	12	0.960	80.2%	0.857	81.7%	31.0%	0.179
	LocalMotion[13, 39]	–	✓	–	–	–	✓	✓	–	–	✓	✓	✓	✓	✓	–	✓	✓	✓	–	10	0.789	76.2%	0.719	78.9%	–	–
2010	ChaoticInvariant[21]	–	✓	✓	–	✓	–	✓	–	–	✓	✓	✓	–	✓	✓	–	✓	✓	–	0	0.990	–	–	–	–	–
	Streakline[22, 27]	–	✓	✓	–	✓	–	✓	–	–	✓	✓	✓	–	–	✓	–	✓	✓	–	–	0.900	–	–	–	–	–
	PhaseCorrelation[36]	✓	–	–	–	✓	–	–	–	–	–	–	✓	–	✓	–	–	✓	✓	✓	2	0.893	–	–	–	–	–
2011	Tracklets[10, 13]	–	✓	✓	✓	–	–	✓	✓	–	✓	✓	✓	✓	✓	✓	–	✓	✓	–	13	0.919	85.2%	0.807	80.3%	–	–
	PotentialEnergy[23, 27]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	–	✓	–	✓	–	–	–	0.985	–	–	–	–	–
	FourierCoefficient[37]	✓	–	–	–	✓	–	–	–	✓	–	–	✓	–	–	–	✓	✓	✓	✓	0	–	–	–	96.4%	–	–
	TopologicalShape[40]	–	✓	–	–	–	✓	✓	–	–	✓	✓	✓	–	–	✓	–	✓	✓	–	–	0.987	–	–	–	–	–
2012	LocalPressure[11]	–	✓	–	✓	–	–	✓	–	–	✓	✓	✓	✓	✓	–	–	✓	–	–	–	0.978	–	–	–	–	–
	StatisticalSums[24]	–	✓	–	–	✓	–	✓	✓	–	✓	✓	✓	✓	–	–	✓	✓	✓	–	–	0.985	–	–	–	16.0%	0.927
	NormalizedEntropy[25]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	–	–	✓	✓	–	–	0	–	–	–	–	–	–
	BehaviorEntropy[26]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	–	–	✓	✓	✓	–	–	0.990	–	–	–	–	0.795
	LogisticClassifier[41]	–	✓	–	–	–	✓	✓	–	–	✓	✓	✓	✓	✓	–	✓	✓	–	–	0	0.988	–	–	–	–	–
2013	ThermalDiffusion[12]	–	✓	✓	✓	–	–	✓	–	–	✓	✓	✓	–	–	✓	–	✓	–	–	1	–	–	–	–	–	–
	ShearForce[13]	–	✓	–	✓	–	–	✓	✓	–	✓	✓	✓	–	✓	–	–	✓	–	–	8	0.929	89.7%	0.883	93.3%	–	–
	SparseCoding[27]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	✓	–	–	✓	✓	–	–	0.989	–	–	–	–	0.823
	CovarianceMatrix[28]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	✓	–	–	–	✓	–	–	–	0.933	–	–	–	–	–
	LaplacianMap[29]	–	✓	–	–	✓	–	✓	✓	–	✓	✓	✓	✓	✓	–	–	✓	✓	–	–	0.977	–	–	–	22.0%	–
	SparseCosting[30]	–	✓	–	–	✓	–	✓	✓	–	✓	✓	✓	✓	✓	–	–	✓	–	–	10	0.978	–	–	–	19.0%	0.461
	SimilarityMetric[42]	–	✓	–	–	–	✓	✓	–	–	✓	✓	✓	✓	–	–	✓	✓	–	–	0	–	–	–	–	–	–
2014	SparseTracking[14]	–	✓	✓	✓	–	–	✓	–	–	✓	✓	✓	✓	–	✓	–	✓	–	–	0	–	–	–	99.5%	–	–
	NearestNeighbor[31]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	✓	✓	–	–	✓	–	–	–	0.986	–	–	–	27.9%	–
	DynamicTexture[38]	✓	–	–	–	✓	–	✓	✓	✓	–	–	✓	–	✓	–	✓	✓	✓	–	–	0.995	–	–	–	22.9%	0.696
	EarthMover[43]	–	✓	–	–	–	✓	✓	–	–	✓	✓	✓	✓	✓	–	–	✓	✓	–	–	0.997	–	–	–	15.0%	–
	NatureSetting[44]	–	✓	–	–	–	✓	✓	–	✓	–	✓	✓	✓	✓	–	–	✓	–	–	7	0.983	–	–	–	15.0%	0.924
2015	GaussianRegression[15]	–	✓	–	✓	–	–	✓	–	–	✓	✓	✓	–	✓	–	✓	✓	✓	–	–	–	–	–	–	23.7%	0.838
	ProjectionVector[32]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	✓	–	–	–	✓	–	–	38	–	93.7%	–	95.0%	–	–
	KernelAnalysis[33]	–	✓	–	–	✓	–	✓	✓	–	✓	✓	✓	✓	–	–	–	✓	–	–	–	0.995	97.8%	–	89.5%	–	–
	WeightedEntropy[47]	✓	–	–	–	–	✓	–	–	✓	–	–	✓	✓	–	–	✓	✓	–	✓	–	0.950	–	–	–	–	–
2016	AdaptiveCutbin[34]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	✓	✓	–	–	✓	✓	–	–	0.983	–	–	–	–	0.650
	HashingFilter[35]	–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	✓	–	–	✓	✓	✓	–	–	0.987	–	–	–	20.0%	0.870
	AverageEnergy[45]	–	✓	✓	–	–	✓	✓	✓	–	✓	✓	✓	–	–	✓	✓	✓	–	✓	0	0.981	–	–	–	–	–
	LightIntensity[46]	–	✓	–	–	–	✓	✓	✓	–	✓	✓	✓	–	–	✓	–	✓	–	–	–	0.984	–	–	–	–	–
Our proposed method		–	✓	–	–	✓	–	✓	–	–	✓	✓	✓	–	–	–	✓	✓	–	–	0	0.978	92.6%	0.999	93.1%	–	–

they suffer from the difficulty to get a training dataset which would cover all possible normal behaviors that can occur in the data. For instances, SocialForce[20] model detected and localized abnormal behaviors in crowd videos using social forces. The estimated social force was capable of detecting the governing dynamics of the abnormal behavior, even in the scenes that it was not trained. But 12 false alarms were resulted from UMN[16] dataset as an incorrect estimation of social forces and some unreliable learning of unknown parameters. SparseCosting[30] model detected abnormal events via a sparse reconstruction over the normal bases. Anomaly of a testing sample was decided by its sparse reconstruction cost, through a weighted linear reconstruction of the over-completed normal bases. Yet 10 false alarms were resulted from UMN[16] dataset as an incorrect estimation of sparse reconstruction cost and unreliable learning of unknown parameters. Unsupervised learning approaches regarding both pixel-wise flow[39–46] and patch-wise flow[47] do not need any prior training data. Hence a database of normal events is built up without prior training samples to detect aberration. Table 1 clues that both pixel-wise and patch-wise flows-based unsupervised learning approaches show high tendency to get real-time operational algorithms. But they make the implicit assumption that normal cases are far more frequent than anomalies in the test data. If this surmise is false then they suffer from extensive false alarms. For instance, LocalMotion[39] model is completely unsupervised, making no prior assumptions of what abnormal events may look like. But 10 false alarms were reported from UMN[16] dataset for this model[13] as the estimated flows sometimes failed to exploit the spatial characteristic of crowd behavior. Besides, it assumes that abnormal events come up rarely, normal events occur often and will be got easily in the initial time.

Why do a vast majority of the existing approaches come from pixel-wise flow based semi-supervised learning? In general, pixel-wise flow-based approaches include several special amenities. For example, explicit segmentation of objects is not necessary, problems related to detecting and tracking of individual objects (e.g., object missing, new object entering, incomplete fragments of object trajectories, etc.) are overcome, and any density mover scenes can be applied. Conversely, if semi-supervised learning-based approaches do not fail badly, their performance is better than that of unsupervised learning. A tentative proof of this proposition would be found from the statistics of Table 1. For example, using UMN[16] benchmark dataset mean performances of supervised-, semi-supervised-, and unsupervised-based approaches can be estimated as AUC $= 0.894$ with ACC $= 83.2\%$, AUC $= 0.961$ with ACC $= 86.4\%$, and AUC $= 0.957$ with ACC $= 76.2\%$, respectively. It seems that semi-supervised-based approaches performed the best. Table 1 also infers the mean performance of AUC ≈ 0.9486, ACC $\approx 83.8\%$, and $\varpi \approx 7$ for methods that used UMN[16] dataset. This overall performance of existing methods may fall into the workable range of many real-world applications but should be improved a bit more for desired level of high performance. Although some methods showed more than average performance, their internal algorithmic assumptions (e.g., in optical flow each pixel goes somewhere, etc.) and estimations would potentially limit to use in many real-world applications. Probably, these two simple demands lead to a central motivation for many computer vision researchers to seek for further reliable and applicable approaches to detect interesting events in crowd videos. In addition, multiple flows be the root of any kind of interesting event or behavior or activity. But it is very unlike to find any existing method where the problem of crowd flow analysis in a scene is distinguished based on single flow and multiple flows.

Our aim is to detect various flows from crowd videos that could be used to benefit a wide number of computer vision applications, including pedestrian traffic and detection of potential panic events when crowd motion patterns become anomalous. We have presented a simple pixel-wise flow-based semi-supervised learning method to detect various flows and interesting events in crowd videos based on eigenvalue analysis of spatiotemporal features (STFs). It points to the problem of crowd flow analysis by distinguishing single versus multiple crowd flows in a scene. Our cardinal contribution is the polynomial fitting algorithm to determine appropriate set of thresholds for distinguishing between single and multiple flows. Figure 1 depicts the flow diagram of our method. It extracts spatiotemporal information from crowd videos deeming two consecutive frames to construct a three-dimensional (3D) tensor, which holds appearance and velocity information. Tensor's upper left minor matrix pertains to intensity structure of the frame region of interest. A normalized continuous rank-increase measure δ_r is calculated for each frame using a generalized interlacing property of the eigenvalues of both tensor and its upper left minor matrix. The use of eigenvalues to recognize crowd events is not new.[13,19] To recognize large-

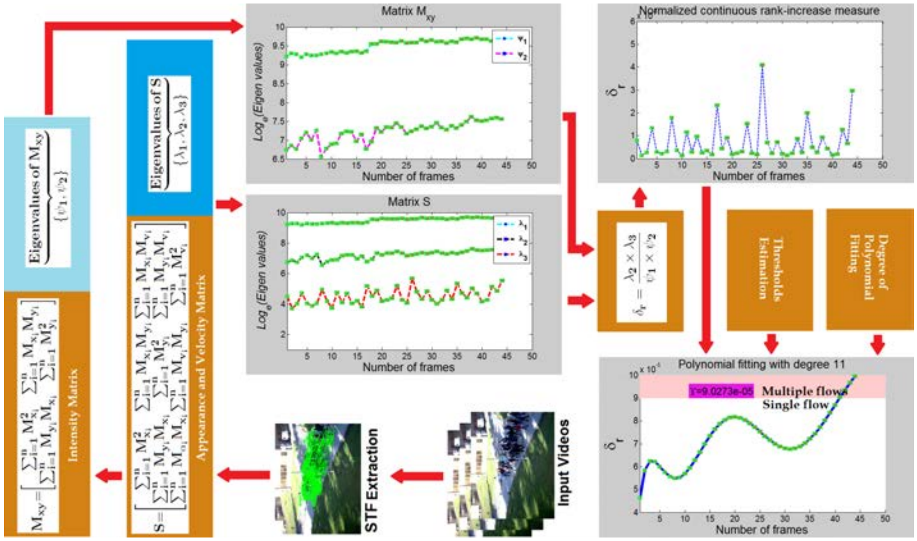

Matrix M_xy
Log_e(Eigen values)
Number of frames
ψ1
ψ2
Matrix S
λ1
λ2
λ3
Normalized continuous rank-increase measure
δr
Eigenvalues of M_xy {ψ1, ψ2}
Eigenvalues of S {λ1, λ2, λ3}
Intensity Matrix
Appearance and Velocity Matrix
δr = (λ2 × λ3)/(ψ1 × ψ2)
Thresholds Estimation
Degree of Polynomial Fitting
Polynomial fitting with degree 11
Y=9.0273e-05
Multiple flows
Single flow
STF Extraction
Input Videos

Fig. 1. Flow diagram of our proposed approach.

scale crowd events, eigenvalue analysis was applied on spatiotemporal variation matrix to extract the principal fluctuations resulting in an abstract fluid field.[13] Our method is treated as an incremental improvement based on the method presented by Sharif–Djeraba.[19] Along with other stuffs, a scrutiny of the alternatives of polynomial fitting degrees and many experiments have been performed on real-world videos to know its robustness and efficacy. If there exist various flows in the crowd video, then δ_r values increase significantly (e.g., see Fig. 3(d)). Due to optical flow estimation error (e.g., problems of illumination and motion discontinuities[48]) and some other factors, it is impracticable to identify their existence (e.g., see Fig. 3(e)). But a right set of the degree of polynomial fitting functions on δ_r values can flawlessly detect their existence. How to estimate that set has been interposed. Applying thresholds on the polynomial fitting δ_r values for various flows (e.g., zero flow, single flow, and multiple flows) as well as interesting events can be recognized as frame basis. Multiple flows be the necessary but would not be the sufficient conditions for anomalies. A threshold Υ applied on the polynomial fitting δ_r values can discriminate between single flow and multiple flows. Another threshold ξ, where $\xi \geq \Upsilon$, can extract interesting or aberrant events from multiple flows.

Although multiple flows are the necessary condition for causing interesting events, sufficiently large number of existing methods did not discretely demonstrate them. Therefore, it is hard to give a comparison of our reported algorithmic outputs of various flows in videos with recorded results of state-of-the-art methods. In whatever way, to make clear how our method is better or different from existing methods, a full numerical comparison of various interesting events (e.g., crowd splitting, running, and evacuation) detection results reported from diverse state-of-the-art methods using PETS2009[49] and UMN[16] datasets has been put forth. The recorded results for detecting interesting events from the same videos made clear and visible that our method is likely a bit superior to its alternative methods, e.g., Fourier-Coefficient,[37] SimilarityMetric[42] (see Fig. 19), SocialForce,[20] HOG/HOF (HOG: Histogram of Oriented Gradient; HOF: Histogram of Optical Flow),[9] HOG3D,[8] Cuboid3D,[18] LocalMotion,[39] Tracklets,[10] ShearForce[13] (see Fig. 20), etc.

The rest of the paper is organized as follows: Section 2 confers implementation steps; Sec. 3 reports a detailed detection abilities of our method followed by few clues for further investigation; and Sec. 4 makes conclusion.

2. Implementation Steps

2.1. *Calculation of spatiotemporal features*

For calculation of *spatiotemporal features*, we apply image velocity (optical flow) technique. To calculate the optical flow between successive video frames, we treat the well-known Pyramid-LK algorithm or pyramidal implementation of the Lucas–Kanade feature tracker.[51–53] Once we define n (typically 10,000) points of interest, we track those points over the next frames using Pyramid-LK algorithm.[51–53] Images in Fig. 2 give evidence of the set of vectors gained with interest points tracking. Upon matching interest points between frames, we can get a set of optical flow vectors, which formulate a matrix $M_{n\times 3}$ by dint of

$$M_{n\times 3} = \begin{bmatrix} M_{x_1} & M_{y_1} & M_{v_1} \\ \vdots & \vdots & \vdots \\ M_{x_i} & M_{y_i} & M_{v_i} \\ \vdots & \vdots & \vdots \\ M_{x_n} & M_{y_n} & M_{v_n} \end{bmatrix}_{n\times 3}, \tag{1}$$

where $M_{x_i} \mapsto$ x-coordinate of any feature element i and $i \in n$, $M_{y_i} \mapsto$ y-coordinate of i, and $M_{v_i} \mapsto$ velocity of i.

2.2. *Analysis of spatiotemporal features*

Deeming that optical flow is constant within frame, then optical flow of the frame be can estimated by solving the following 3D scatter matrix S as

$$S \times \begin{bmatrix} x \\ y \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3\times 1}. \tag{2}$$

And S can be formulated by multiplying both sides of Eq. (1) by M^T (the transpose of M), more explicitly:

$$S = M^T M = \begin{bmatrix} \sum_{i=1}^{n} M_{x_i}^2 & \sum_{i=1}^{n} M_{x_i} M_{y_i} & \sum_{i=1}^{n} M_{x_i} M_{v_i} \\ \sum_{i=1}^{n} M_{y_i} M_{x_i} & \sum_{i=1}^{n} M_{y_i}^2 & \sum_{i=1}^{n} M_{y_i} M_{v_i} \\ \sum_{i=1}^{n} M_{\alpha_i} M_{x_i} & \sum_{i=1}^{n} M_{v_i} M_{y_i} & \sum_{i=1}^{n} M_{v_i}^2 \end{bmatrix}_{3\times 3}. \tag{3}$$

The 3×3 matrix S belongs to the spatiotemporal features. Consequently, this 3D structural tensor S contains information on both appearance and velocity simultaneously. All possible ranks of S would be 0, 1, and 2. In general, Eq. (2) is a set of linear equations. Hence, if the optical flows are constant within the frame region, there will be a nonzero solution of Eq. (2). As a result, coefficient matrix S should be rank deficient, i.e., $\text{rank}(S) \leq 2$. Explicitly, if S is not rank deficient, i.e., its smallest eigenvalue $\lambda_{\min}$ is not equal to zero ($\lambda_{\min}(S) \neq 0$), then there will not exist any motion consistency within the frame region of interest. Coefficient matrix S contains information about appearance and motion. Consequently, the rank of S can be used to analyze the brightness distribution and motion types within the frame region. In case of $\mathrm{rank}(S) = 3$, there will be multiple motions within the frame region. A distributed spatial brightness structure moves at a constant motion when $\mathrm{rank}(S) = 2$. If we examine all possible ranks of S, which contains only uniform motion, then to come to know the spatial properties of the frame region of interest we could consider the upper left minor matrix M_{xy} of the tensor S as

$$M_{xy} = \begin{bmatrix} \sum_{i=1}^{n} M_{x_i}^2 & \sum_{i=1}^{n} M_{x_i} M_{y_i} \\ \sum_{i=1}^{n} M_{y_i} M_{x_i} & \sum_{i=1}^{n} M_{y_i}^2 \end{bmatrix}_{2\times 2} . \tag{4}$$

Matrix M_{xy} captures the intensity structure of the frame region of interest. Let us assume that ψ_1 and ψ_2 be the eigenvalues of matrix M_{xy}. The eigenvalues form a rotationally invariant description. There are three cases to be considered:

- Rank(S) = 0: If both ψ_1 and ψ_2 are small, so that the auto-correlation function of S is having a uniform colored image region, i.e., little change in S (frame region of interest) in any direction, then the flat image region is of approximately constant intensity.
- Rank(S) = 1: If one eigenvalue is high and the other is low, so that the auto-correlation function of S is posing a rigid-shaped image region, then only shifts along the ridge (i.e., along the edge) cause little change in S (frame region of interest) and significant change in the orthogonal direction.
- Rank(S) = 2: If both ψ_1 and ψ_2 are high, so that the auto-correlation function of S is assuming a posture as for the character of a sharply-peaked image, then shifts in any direction result in a significant increase in S (frame region of interest).

2.3. *Flow analysis*

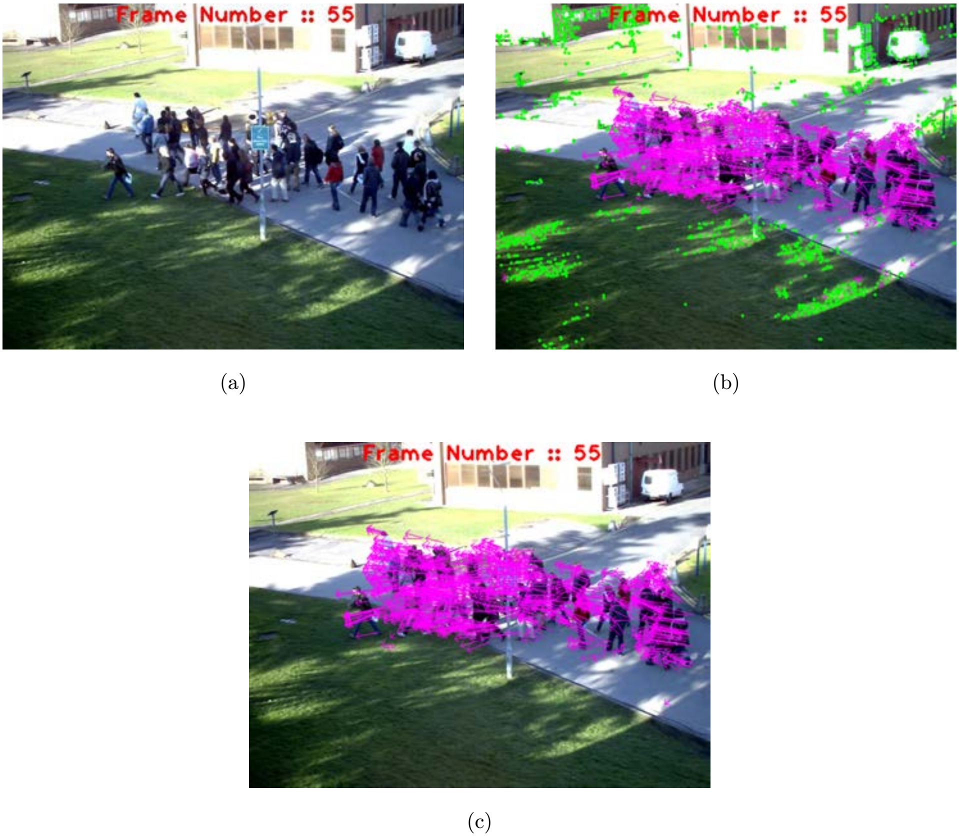
When there is only uniform flow within the frame region of interest, the added temporal component at the third row and column does not introduce any increase in rank. This condition satisfies $\mathrm{rank}(S) = \mathrm{rank}(M_{xy})$. Nonetheless, condition does not satisfy when the flow is not along a single straight line. In such cases, the added temporal component introduces an increase in the rank, $\mathrm{rank}(S) = \mathrm{rank}(M_{xy}) + 1$. The difference in rank cannot be more than 1, because only one column/row is added in the transition from M_{xy} to S. Then measuring the rank-increase δ between S and M_{xy} reveals whether the frame region of interest contains a single or multiple flows:

$$\delta = \mathrm{rank}(S) - \mathrm{rank}(M_{xy}) = \begin{cases} 0 & \text{single flow}\,, \\ 1 & \text{multiple flows}\,. \end{cases} \tag{5}$$

The simple way to estimate the rank-increase from rank(M_{xy}) to S is to compute their individual ranks and then take the difference, which provides either 0 or 1. The rank of a matrix is determined by the number of its nonzero eigenvalues. Notwithstanding, in the presence of noise, eigenvalues are never zero. Usage of a threshold to the eigenvalues is mainly data-dependent. Besides a wrong choice of a threshold would lead to faulty rank values. If two motions are very similar but not identical, are they consistent or not? As a consequence, a normalized and continuous measure is needed to quantify the matrix deficiency.

2.4. *Normalized continuous rank-increase measure δ_r*

Figure 2(a) demos one of 86 frames from the V_1 of PETS2009[49] as sampled in Fig. 10. Five more frames of V_1 and their corresponding optical flows have been displayed

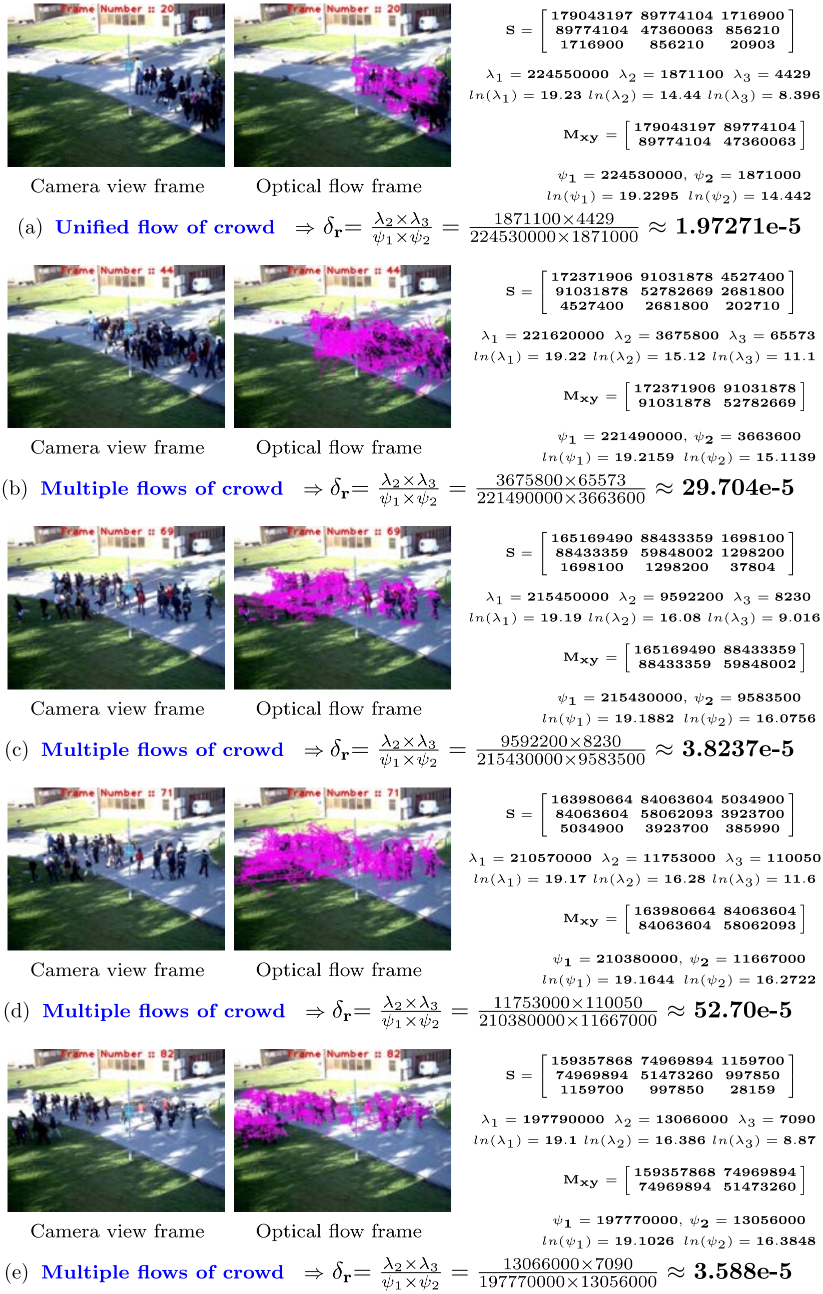


(a)

(b)

(c)

Fig. 2. Panel (a) evinces the original frame. Green points and pink arrows in (b) exhibit static corners and optical flow, respectively. Panel (c) demonstrates the optical flow without static corners. Crude elements of (c) are used to model the matrix $M_{n\times3}$ with a view to detecting potential flows and events (color online).



Camera view frame Optical flow frame

$$\mathbf{S} = \begin{bmatrix} 179043197 & 89774104 & 1716900 \\ 89774104 & 47360063 & 856210 \\ 1716900 & 856210 & 20903 \end{bmatrix}$$

$\lambda_1 = 224550000$ $\lambda_2 = 1871100$ $\lambda_3 = 4429$
$ln(\lambda_1) = 19.23$ $ln(\lambda_2) = 14.44$ $ln(\lambda_3) = 8.396$

$$\mathbf{M_{xy}} = \begin{bmatrix} 179043197 & 89774104 \\ 89774104 & 47360063 \end{bmatrix}$$

$\psi_1 = 224530000$, $\psi_2 = 1871000$
$ln(\psi_1) = 19.2295$ $ln(\psi_2) = 14.442$

(a) **Unified flow of crowd** $\Rightarrow \delta_\mathbf{r} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{1871100 \times 4429}{224530000 \times 1871000} \approx \mathbf{1.97271e\text{-}5}$

Camera view frame Optical flow frame

$$\mathbf{S} = \begin{bmatrix} 172371906 & 91031878 & 4527400 \\ 91031878 & 52782669 & 2681800 \\ 4527400 & 2681800 & 202710 \end{bmatrix}$$

$\lambda_1 = 221620000$ $\lambda_2 = 3675800$ $\lambda_3 = 65573$
$ln(\lambda_1) = 19.22$ $ln(\lambda_2) = 15.12$ $ln(\lambda_3) = 11.1$

$$\mathbf{M_{xy}} = \begin{bmatrix} 172371906 & 91031878 \\ 91031878 & 52782669 \end{bmatrix}$$

$\psi_1 = 221490000$, $\psi_2 = 3663600$
$ln(\psi_1) = 19.2159$ $ln(\psi_2) = 15.1139$

(b) **Multiple flows of crowd** $\Rightarrow \delta_\mathbf{r} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{3675800 \times 65573}{221490000 \times 3663600} \approx \mathbf{29.704e\text{-}5}$

Camera view frame Optical flow frame

$$\mathbf{S} = \begin{bmatrix} 165169490 & 88433359 & 1698100 \\ 88433359 & 59848002 & 1298200 \\ 1698100 & 1298200 & 37804 \end{bmatrix}$$

$\lambda_1 = 215450000$ $\lambda_2 = 9592200$ $\lambda_3 = 8230$
$ln(\lambda_1) = 19.19$ $ln(\lambda_2) = 16.08$ $ln(\lambda_3) = 9.016$

$$\mathbf{M_{xy}} = \begin{bmatrix} 165169490 & 88433359 \\ 88433359 & 59848002 \end{bmatrix}$$

$\psi_1 = 215430000$, $\psi_2 = 9583500$
$ln(\psi_1) = 19.1882$ $ln(\psi_2) = 16.0756$

(c) **Multiple flows of crowd** $\Rightarrow \delta_\mathbf{r} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{9592200 \times 8230}{215430000 \times 9583500} \approx \mathbf{3.8237e\text{-}5}$

Camera view frame Optical flow frame

$$\mathbf{S} = \begin{bmatrix} 163980664 & 84063604 & 5034900 \\ 84063604 & 58062093 & 3923700 \\ 5034900 & 3923700 & 385990 \end{bmatrix}$$

$\lambda_1 = 210570000$ $\lambda_2 = 11753000$ $\lambda_3 = 110050$
$ln(\lambda_1) = 19.17$ $ln(\lambda_2) = 16.28$ $ln(\lambda_3) = 11.6$

$$\mathbf{M_{xy}} = \begin{bmatrix} 163980664 & 84063604 \\ 84063604 & 58062093 \end{bmatrix}$$

$\psi_1 = 210380000$, $\psi_2 = 11667000$
$ln(\psi_1) = 19.1644$ $ln(\psi_2) = 16.2722$

(d) **Multiple flows of crowd** $\Rightarrow \delta_\mathbf{r} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{11753000 \times 110050}{210380000 \times 11667000} \approx \mathbf{52.70e\text{-}5}$

Camera view frame Optical flow frame

$$\mathbf{S} = \begin{bmatrix} 159357868 & 74969894 & 1159700 \\ 74969894 & 51473260 & 997850 \\ 1159700 & 997850 & 28159 \end{bmatrix}$$

$\lambda_1 = 197790000$ $\lambda_2 = 13066000$ $\lambda_3 = 7090$
$ln(\lambda_1) = 19.1$ $ln(\lambda_2) = 16.386$ $ln(\lambda_3) = 8.87$

$$\mathbf{M_{xy}} = \begin{bmatrix} 159357868 & 74969894 \\ 74969894 & 51473260 \end{bmatrix}$$

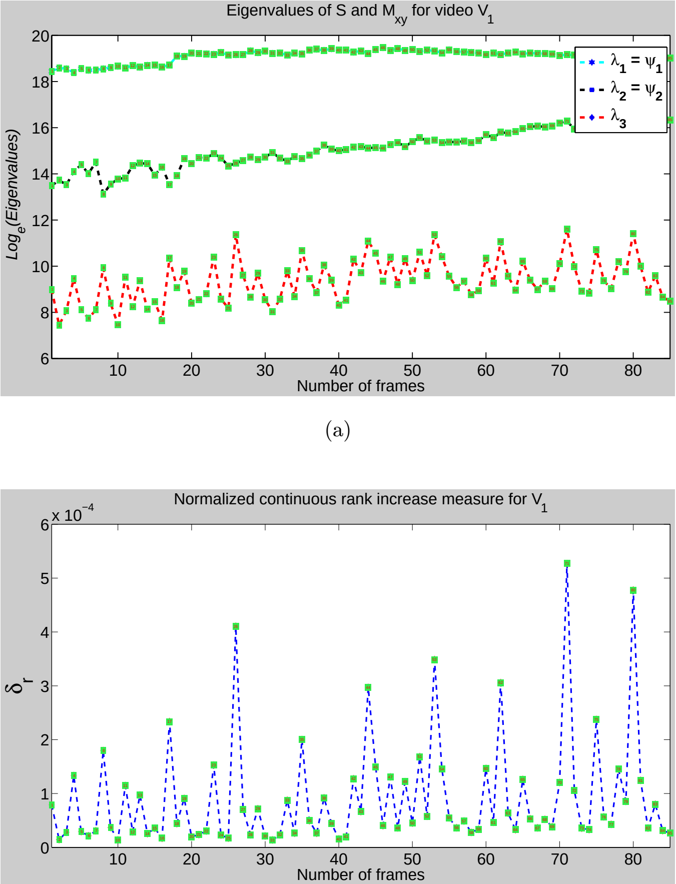
$\psi_1 = 197770000$, $\psi_2 = 13056000$
$ln(\psi_1) = 19.1026$ $ln(\psi_2) = 16.3848$

(e) **Multiple flows of crowd** $\Rightarrow \delta_\mathbf{r} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{13066000 \times 7090}{197770000 \times 13056000} \approx \mathbf{3.588e\text{-}5}$

Fig. 3. Five randomly selected sample camera view frames of V_1 as well as their corresponding optical flows along with numerical calculations of S, λ_1, λ_2, λ_3, M_{xy}, ψ_1, ψ_2, and δ_r. Panel (a) shows single unified flow of crowd, whereas Panel (b) denotes initial stage of multiple flows. Panels (c), (d), and (e) depict split and multiple flows of crowd. Yet calculated δ_r values for (c) and (e) did not reflect the real cases, i.e., a correction must be performed. But how would we perform the correction?

in Fig. 3. Frame number 20 exhibits a single uniform flow of crowd, whereas frame number 44 parades the uniform flow has been split into two flows, i.e., two different flows had existed. Frames 69, 71, and 82 depict the three ruptured different flows, i.e., there exist multiple flows. How can we detect those multiple flows?

Using Eqs. (1), (3), and (4) we have constructed 85 different S and M_{xy} matrices from V_1 video sequence. Let us assume that $\lambda_1, \lambda_2, \lambda_3$ and ψ_1, ψ_2 be the eigenvalues of S and M_{xy}, respectively. The eigenvalues of S and M_{xy} matrices for all 85 frames have been plotted in Fig. 4(a). Eigenvalues satisfy the following inequalities: $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\psi_1 \geq \psi_2$. It is noticeable that λ_1 and λ_2 coincide on ψ_1 and ψ_2, respectively. Consequently, λ_3 becomes the leading factor to analyze the multiple flows.



(a)

(b)

Fig. 4. Numerical numbers inscribed in the green rectangles of curves indicate frame numbers. Panel (a) shows plotting of the eigenvalues of S and M_{xy} for V_1. Panel (b) depicts δ_r values for different frames using Eq. (9). From this shape it is hard to detect that after frame number 41 crowd has been split, i.e., existing multiple flows cannot be detected. But how would we detect those multiple flows? (color online).

Thus from this observation a generalized interlacing property of eigenvalues in S and M_{xy} matrices can be structured as: $\lambda_1 \geq \psi_1 \geq \lambda_2 \geq \psi_2 \geq \lambda_3$. This yields to the following observations:

$$\lambda_1 \geq \frac{\lambda_1 \times \lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{\det(S)}{\det(M_{xy})} \geq \lambda_3\,, \tag{6}$$

$$1 \geq \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} \geq \frac{\lambda_3}{\lambda_1} \geq 0\,, \tag{7}$$

$$1 \geq \delta_r \geq \frac{\lambda_3}{\lambda_1} \geq 0\,, \tag{8}$$

where the *normalized continuous rank-increase measure* δ_r follows $0 \leq \delta_r \leq 1$ and

$$\delta_r = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2}\,. \tag{9}$$

Based on application eigenvalues may be very high, a logarithmic function would be used. The case of $\delta_r = 0$ is an ideal case of no rank-increase. But there is a clear rank-increase when $\delta_r = 1$. In spite of that, δ_r allows to handle noisy data and provides diverge degrees of rank-increases for varying degrees of motion consistencies.

Calculations of S, λ_1, λ_2, λ_3, M_{xy}, ψ_1, ψ_2, and δ_r values for five randomly selected sample frames of V_1 have been presented in Fig. 3. It is noticeable that $\delta_r = 1.9727$E–5 for frame number 20 in Fig. 3(a) which includes single uniform flow of crowd, whereas $\delta_r = 29.704$E–5 for frame number 44 in Fig. 3(b) which shows initial stage of multiple flows. Similarly, $\delta_r = 3.8237$E–5, $\delta_r = 52.7$E–5, and $\delta_r = 3.588$E–5 should indicate multiple flows for frames 69, 71, and 82, respectively. But δ_r values for frames 69 and 82 (i.e., Figs. 3(c) and 3(e)) are too low to represent multiple flows of crowd, i.e., incorrect estimation. Not only these two frames but also there are other frames where calculation went wrong. Figure 4(b) depicts the measure of δ_r values using the calculated data as displayed in Fig. 4(a). Anyhow the brought off graph shows extremely low performance to analyze flows. Because in crowded scenes, the extracted low-level features, such as optical flows or spatio-temporal interest points, are inevitably uncertain and noisy. Nonetheless, a polynomial fitting function can minimize the existing problem with a great extent.

2.5. *Polynomial fitting*

We can apply a threshold on δ_r values to have a decision about multiple flows. But any discrete value of δ_r which exceeds a predefined threshold is not a clear evidence of the existence of multiple flows. At least one attribute may have been severely corrupted by mistake or tracking calculation errors, which would lead to an erroneous decision on multiple flows. To minimize this dismay, a polynomial fitting can be an acceptable solution. In addition, polynomial models are among the most frequently used empirical models to fit functions and are computationally easy to use having moderate flexibility of shapes. Take on any θ-degree polynomial fitting of δ_r values of any frame η that is denoted by $f_\theta(\eta)$ where $\theta \in \{2, 3, \ldots, 30\}$ and $\eta \in \{1, 2, 3, \ldots, 85\}$. Various θ-degree polynomial fittings of the δ_r values of 85 distinct frames that succeeded from Fig. 4(b) are acted for Ω_θ diverse sets as

$$\begin{bmatrix} \Omega_2 = & \{f_2(1) & f_2(2) & f_2(3) & f_2(4) & \cdots & f_2(85) \\ \Omega_3 = & \{f_3(1) & f_3(2) & f_3(3) & f_3(4) & \cdots & f_3(85) \\ \Omega_4 = & \{f_4(1) & f_4(2) & f_4(3) & f_4(4) & \cdots & f_4(85) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Omega_{30} = & \{f_{30}(1) & f_{30}(2) & f_{30}(3) & f_{30}(4) & \cdots & f_{30}(85) \end{bmatrix}. \tag{10}$$

Figure 5 plots different datasets looking on Eq. (10). Nevertheless, from Fig. 5 it is extremely hard to make a decision on which degree of polynomial performs well.

2.6. *Simulation to get the knowledge of polynomial fitting degree*

How does the choice of polynomial fitting degree affect the obtained results? One of the main limitations of polynomial model is that it is significant to make a shape or degree tradeoff. To model data with a complex structure, the degree of the model should be high, indicating that the associated number of parameters to be estimated will be high too. This would result in highly unstable model. It is fascinating to find a

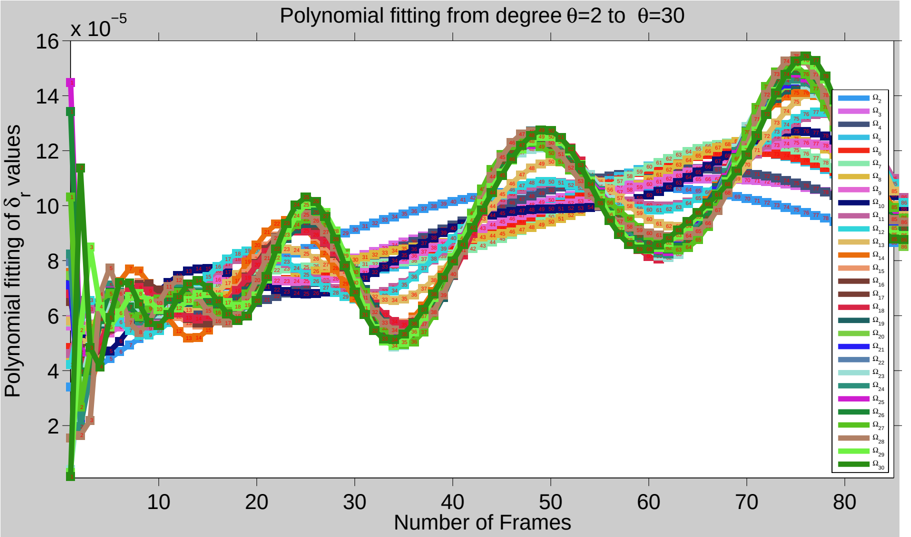


Fig. 5. The two-dimensional (2D) plotting of miscellaneous θ-degree polynomial fittings of the δ_r values of 85 separate frames came after Fig. 4 as formulated in Eq. (10). Numerical numbers inscribed in the rectangles of curves mark frame numbers. Curves are too congested to extract the informative range of polynomial fitting degrees. But how would we get the right range of polynomial fitting degrees?

set of polynomial degrees to fit δ_r values to keep pace with lower and higher order polynomials. Runge's phenomenon[54] affirms that lower order polynomials are normally to be preferred instead of augmenting the degree of interpolation polynomial, even if some of the badness of this interpolation may be overcome by using Chebyshev polynomials[55] instead of equidistant points.[25] However, we wish to find a set of polynomial fitting degrees that give very stable results.

We assume that any two datasets, irrespective of their values, will come very close to each other or make a convergence in some degrees of polynomial fitting. Our target is to find that range of polynomial fitting degrees which exhibit such manner of behaving. If we could find out that set of polynomial fitting degrees, we can easily solve our degree of polynomial fitting problem on the gained δ_r values from any video. With this vein, we have performed a bit more calculation on the obtained different degrees of polynomial fitted δ_r values of 85 frames. We have reorganized the fitted δ_r values got from Fig. 10 as $\ell_\jmath$ different sets where $\jmath = \{1, 2, 3, \ldots, 85\}$ as

$$\left[\begin{array}{lllllll}
\ell_1 = & \{f_2(1) & f_3(1) & f_4(1) & f_5(1) & \cdots & f_{30}(1) \\
\ell_2 = & \{f_2(2) & f_3(2) & f_4(2) & f_5(2) & \cdots & f_{30}(2) \\
\ell_3 = & \{f_2(3) & f_3(3) & f_4(3) & f_5(3) & \cdots & f_{30}(3) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\ell_{85} = & \{f_2(85) & f_3(85) & f_4(85) & f_5(85) & \cdots & f_{30}(85)
\end{array}\right]. \tag{11}$$

We have applied a polynomial fitting on each of $\ell_\jmath$ sets deeming a very high degree of polynomial (let us say 28). High degree of polynomial has been taken into account to get an exact or as good as possible match of data points. On applying 28-degree polynomial fitting on each of $\ell_\jmath$ sets, the new datasets $\ell'_\jmath$ are denoted as:

$$\left[\begin{array}{lllllll}
\ell'_1 = & \{f'_2(1) & f'_3(1) & f'_4(1) & f'_5(1) & \cdots & f'_{30}(1) \\
\ell'_2 = & \{f'_2(2) & f'_3(2) & f'_4(2) & f'_5(2) & \cdots & f'_{30}(2) \\
\ell'_3 = & \{f'_2(3) & f'_3(3) & f'_4(3) & f'_5(3) & \cdots & f'_{30}(3) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\ell'_{85} = & \{f'_2(85) & f'_3(85) & f'_4(85) & f'_5(85) & \cdots & f'_{30}(85)
\end{array}\right]. \tag{12}$$

For example, $\ell_1 =$ [0.1398E–4, 0.1398E–4, 0.5184E−4, 0.401E−4, 0.5696E−4, 0.518E−4, 0.4147E−4, 0.5004E−4, 0.5148E−4, 0.5282E−4, 0.5374E−4, 0.5374E−4, 0.5374E−4, 0.5374E−4, 0.5374E−4, 0.5374E−4, 0.5374E−4, 0.5374E−4, 0.3038E−4, 0.1738E−4, 0.2581E−4, 0.1536E−4, 0.1432E−4, 0.1152E−4, −0.0716E−4, −0.1085E−4, 0.1463E−4, 0.0607E−4, −0.8642E−4. On applying 28-degree polynomial fitting on ℓ_1 set we have obtained the new values of ℓ_1 as: $\ell'_1 =$ {0.3645, −0.1437, −0.0152, 0.0061, −3.0742E−4, 1.6584E−4, 1.3203E−5, 5.0151E−5, 6.0527E−5, 5.3049E−5, 5.5629E−5, 5.6492E−5, 5.5165E−5, 4.8166E−5, 3.0616E−5, −3.6352E−5, 9.7210E−5, 1.4182E−4, −2.4913E−4,

$-5.607\text{E}{-4}$, 0.0011, 0.0021, -0.0101, -0.0449, -0.032, -0.086, 0.0087, 0.331, $1.153\}$. Figure 6 plots together every value of all $\ell'_\jmath$ sets acquired from Eq. (12). From Fig. 6 it is distinct that all $\ell'_\jmath$ datasets have tried to make a convergence with polynomial fitting degree from $\theta = 8$ to $\theta = 17$. But all $\ell'_\jmath$ datasets have made a divergence with rest of degrees. These phenomena also prove our assumption that any two $\ell_\jmath$ sets, irrespective of their values, will come very close to each other or make a convergence in some degrees of polynomial fitting.

Although Fig. 6 makes visible a clear range of polynomial fitting degrees, if the cumulative distribution function of a discrete probability distribution (e.g., negative binomial cumulative distribution function) at each value of $\ell'_\jmath$ as shown in Eq. (12) would be applied, then a clearer range of polynomial fitting degrees can be observed. Widely known and esteemed discrete probability distributions are binomial, negative binomial, Bernoulli, Poisson, uniform (discrete), geometric, hypergeometric, and multinomial. Among these models, in general, the Poisson distribution is appropriate for applications that involve counting the number of times a random event occurs in a given amount of time, volume, area, distance, and so on. Nevertheless, the Poisson model assumes that the mean and variance of the errors are equal. Although the variance of the errors can be smaller, usually in practice it is larger than the mean. The negative binomial distribution has a variance that is greater than its mean, making it suitable for counting data that do not meet the assumptions of the Poisson distribution. Thus it is more general than the Poisson distribution, and is often

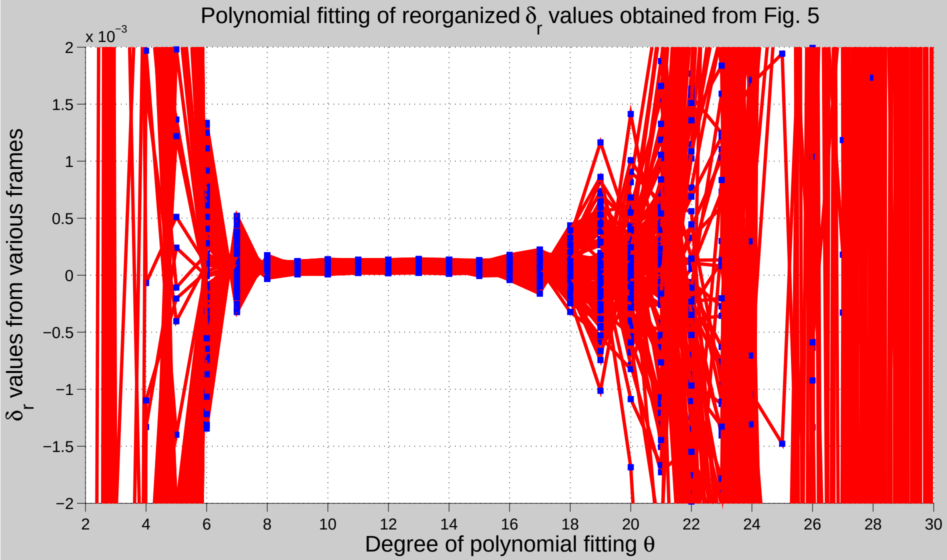


Fig. 6. The 2D plotting of the numerical values of all $\ell'_\jmath$ sets together as structured in Eq. (12). This plotting of data gives evidence of a clear data convergence with polynomial fitting degree from $\theta = 8$ to $\theta = 17$ exclusively, i.e., the problem in Fig. 5 has been solved as $\theta \in \{8, 9, 10, \ldots, 17\}$.

suitable for counting data when the Poisson model proves insufficient. It has no location or scale parameters. It can take on a variety of shapes, ranging from very skewed to nearly symmetric, depending on the value of $\ell'_\jmath$. The negative binomial cumulative distribution function $N_{bcdf}(\beta, \gamma, p)$ is defined as

$$N_{bcdf}(\beta,\gamma,p) = \begin{cases} 0\,, & \beta < 0\,, \\ p^{\gamma}\displaystyle\sum_{\epsilon=0}^{\beta} \frac{(\gamma+\epsilon-1)!}{\epsilon!(\gamma-1)!}(1-p)^{\epsilon}\,, & \beta \geq 0\,, \end{cases} \tag{13}$$

where β be a value from each set of $\ell'_\jmath$, γ be a numeric value that counts the number of successes meeting the condition of $\gamma > 0$, and p be a numeric probability of success with satisfying the condition of $0 \leq p \leq 1$. Using Eq. (13) it is easy to calculate the cumulative probability of any value from Eq. (12). For example, if $\beta = 1.1529$, $\gamma = 1.53$, and $p = 0.75$, then on applying Eq. (13) we get: $N_{bcdf}(1.1529, 1.53, 0.75) = (0.75)^{1.53}[\frac{(1.53+0-1)!}{0!(1.53-1)!}(1-0.75)^0 + \frac{(1.53+1-1)!}{1!(1.53-1)!}(1-0.75)^1] = 0.643937[1+ 0.3825] = 0.89024$. If we apply each value of ℓ'_1 repeatedly in Eq. (13), then we can calculate their negative binomial cumulative distribution as: $N_{bcdf}(\ell'_1, 1.53, 0.75) = \{0.6439, 0, 0, 0.6439, 0, 0.6439, 0.6439, 0.6439, 0.6439, 0.6439, 0.6439, 0.6439, 0.6439, 0.6439, 0.6439, 0, 0.6439, 0.6439, 0, 0, 0.6439, 0.6439, 0, 0, 0, 0, 0.6439, 0.6439, 0.8902\}$. In the similar manner, it is easy to calculate $N_{bcdf}(\ell'_2, 1.53, 0.75)$, $N_{bcdf}(\ell'_3, 1.53, 0.75)$, $N_{bcdf}(\ell'_4, 1.53, 0.75)$, $N_{bcdf}(\ell'_5, 1.53, 0.75)$, etc. Figure 7 plots the

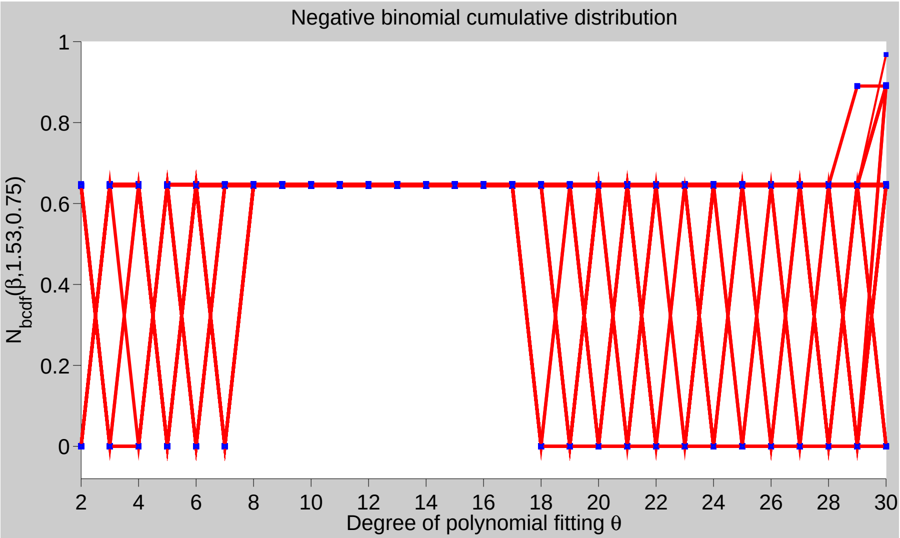
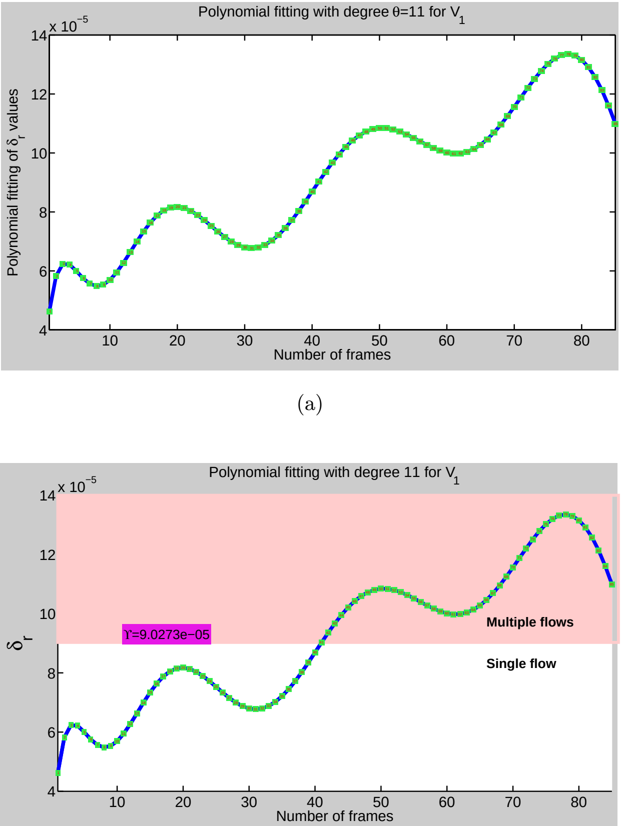


Fig. 7. The 2D plotting of the negative binomial cumulative distribution of all $\ell'_\jmath$ sets together as adverted in Eq. (12). This plotting of data shows a better data convergence with polynomial fitting degree from $\theta = 8$ to $\theta = 17$ only, i.e., the problem in Fig. 5 has been solved as $\theta \in \{8, 9, 10, \ldots, 17\}$.

negative binomial cumulative distribution of all values of all ℓ'_J sets deeming corresponding parameters in $\gamma = 1.53$ and probability of success $p = 0.75$. From Fig. 7 it is noticeable that the negative binomial cumulative distribution function in Eq. (13) has cut off any negative value from ℓ'_J to 0 and made any positive value from ℓ'_J to certain constant upper bound value based on β, $\gamma = 1.53$, and $p = 0.75$. For instance, if β value is less than 1 or greater than or equal to 0, then the upper bound is fixed at 0.6439 considering $\gamma = 1.53$ and $p = 0.75$. All values of ℓ'_J have made a convergence from polynomial fitting degree $\theta = 8$ to $\theta = 17$. This phenomena give us a very clear range of polynomial fitting degree $\theta \in \{8, 9, 10, \ldots, 17\}$. Consequently, we can apply any alternative of polynomial fitting degree θ from the set of $\{8, 9, 10, \ldots, 17\}$ in



(a)

(b)

Fig. 8. Numerical numbers inscribed in the green rectangles of curve denote frame numbers. Panel (a) displays polynomial fitting of δ_r values at set Ω_{11} in Eq. (10) taken in Fig. 5. Panel (b) differentiates between single and multiple flows with $\Upsilon_{V_1} = 9.0273\text{E}{-5}$. On frame number 41 crowd has been split and the existence of multiple flows can be detected, i.e., the problem in Fig. 4 has been solved (color online).

Fig. 4(b). Figure 8(a) depicts polynomial fitting with $\theta = 11$ of the dataset Ω_{11} brought in Eq. (10), i.e., 11-degree polynomial fitting of the data in Fig. 4(b). Although Fig. 8(a) gives us a clear picture of different flows, another very important problem remains, i.e., it does not indicate how to differentiate between single flow and multiple flows. Henceforth, it is important to estimate a good threshold for each video such that single flow and multiple flows can be separated on the individual frame basis.

2.7. *Threshold estimation*

The decision of single or multiple flows can be taken by comparing with polynomial fitting data of δ_r (e.g., see Fig. 8(a)) with a predefined threshold Υ, which can be estimated from video which contains mainly unified crowd flow. Maximum estimated value from such video is then added with some Gaussian noise to get Υ. A multiple flow frame can be detected if a polynomial fitting of δ_r is greater than Υ. If a polynomial fitting of δ_r value is zero or close to zero, then the corresponding video frame will be considered as zero flows. The Υ depends on the controlled environments (e.g., videos), specifically the remoteness of the camera to the scene, the orientation of the camera, type and position of the camera, lighting system, density of the objects, etc. If we have $\aleph$ videos, which are the case in sites namely airports, banks, hospitals, hotels, concerts, parking places, political events, shopping malls, subways, stations, schools, sporting events, town centers, etc., then we put forward at least $\aleph$ thresholds. If the video stream $V_{\kappa-1}$, where $\kappa - 1 \in \aleph$, leaves for another V_κ, $\kappa \in \aleph$, then Υ of V_κ should be regenerated as

$$\Upsilon_{V_\kappa} = \max_{\eta=1,\ldots,\Gamma}\{\Omega_\theta\}_\eta + \min_{\eta=1,\ldots,\Gamma}\left[\frac{1}{(2\pi)^2}\sum_{k=0}^{\infty}\frac{(-1)^k((\Omega_\theta)_{\max})^{2k+1}}{k!(2k+1)}\right]_\eta, \qquad (14)$$

where Γ be the frame number of the unified crowd flow of video database. For example, by dint of Eq. (14) the estimated threshold of the corresponding video in Fig. 8(a) is $\Upsilon_{V_1} = 9.0273\text{E}{-5}$. All frames above this numerical value will be treated as frames with multiple flows. A bit more, Fig. 8(b) gives a comprehensive picture of various flows. The sharp slope including frame 41 and its following frames indicate the existence of multiple flows (i.e., crowd is splitting) in these video sequences. To see both quantitative and qualitative performances of polynomial fitting, we may have a look back into Fig. 3 where previously calculated δ_r went wrong for frames 69 and 82 to make a decision. But now δ_r values are sufficient enough to demo multiple flows in the crowded scenes. Using Fig. 8(b), Table 2 lists the obtained δ_r values in Fig. 3 with before and after 11-degree polynomial fitting.

Multiple flows in the crowd videos are necessary condition but would not be the sufficient condition for abnormal activities. In consequence, any kind of multiple flows of crowd may not lead to an abnormal event. We may need to apply another threshold ξ to detect interesting/abnormal behaviors. To estimate ξ we can reuse

Table 2. How has the correction in Fig. 3 been performed? How have δ values been affected by polynomial fitting?

	δ values in Fig. 3 before and after 11-degree polynomial fitting				
	Before		After		
Numerous frames in Fig. 3	δ values	Remarks	δ values	Degree of effect	Concluding remarks
Frame 20 in Fig. 3(a)	**19.727E–6**	**Correct**	**79.785E–6**	δ increased by 4.045 times	Unified flow of crowd
Frame 44 in Fig. 3(b)	**297.04E–6**	**Correct**	**102.92E–6**	δ decreased by 2.886 times	Split (multiple flows) of crowd
Frame 69 in Fig. 3(c)	**38.237E–6**	**Incorrect**	**116.89E–6**	δ increased by 3.057 times	Split (multiple flows) of crowd
Frame 71 in Fig. 3(d)	**527.0E–6**	**Correct**	**129.38E–6**	δ decreased by 4.073 times	Split (multiple flows) of crowd
Frame 82 in Fig. 3(e)	**35.877E–6**	**Incorrect**	**134.38E–6**	δ increased by 3.746 times	Split (multiple flows) of crowd

Eq. (14) on keeping under control snobbishly standard activity of crowd from multiple flows. A frame of an interesting/abnormal activity can be detected if the polynomial fitting of δ_r is greater than ξ where $\xi \geq \Upsilon$. Based on the context, the split of a crowd (e.g., Fig. 3) would be considered as abnormal activity. For example, if there exists a sudden impediment in the normal pedestrian lane, flows of crowd may be split and then that would become very interesting for traffic officers.

2.8. *Effect of threshold on sensitivity and specificity*

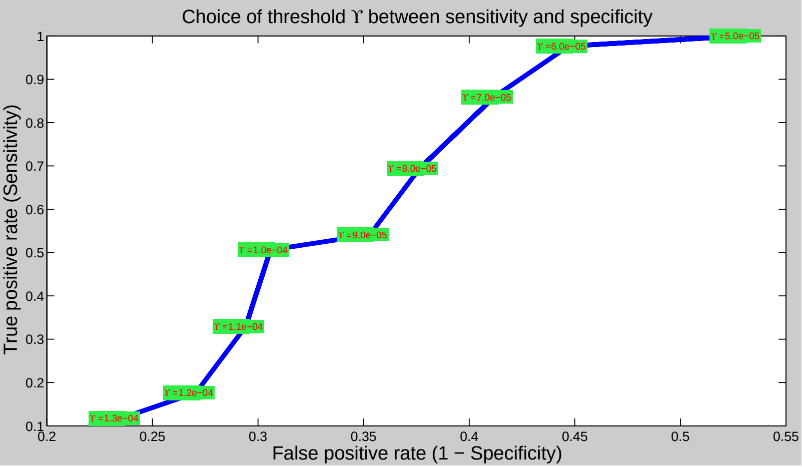
How does threshold affect the two well-known statistical probability measures namely sensitivity and specificity? Sensitivity or true positive rate or recall rate measures the percentage of actual positives which are correctly identified as such condition. It is complementary to the false negative rate. Specificity or true negative rate measures the percentage of negatives which are correctly identified as such condition. It is complementary to the false positive rate. The relation between sensitivity and specificity is that an increase in sensitivity is accompanied by a decrease in specificity and vice versa. We have used data from Figs. 4(b) and 8(a) to illustrate how sensitivity and specificity change depending on the choice of threshold Υ values. Let frames with δ_r values greater than $\Upsilon = 6\text{E}{-5}$ be esteemed as abnormal event's frame. In this vein, as shown in Table 3, the curve of polynomial fitting with 11 degrees shows that numbers of normal and abnormal frames are 2 and 83, respectively. Conversely, the curve without polynomial fitting demos that numbers of normal and abnormal frames are 47 and 38, respectively. With this end, we can verify that the sensitivity is $(83/(2+83)) \approx 0.9765$ and the specificity is $(47/(47+38)) \approx 0.5529$. Assume that we have decided to make the definition of abnormal event's frame less stringent and then let frames with δ_r values greater than

Table 3. Analysis of sensitivity and specificity deeming data from Figs. 4(b) and 8(a).

	11-Degree polynomial fitting				Without polynomial fitting		
Threshold Υ values	N_Υ	A_Υ	Sensitivity	AUC$_\Upsilon$	N_Υ	A_Υ	Specificity
6E−5	2	83	0.9765	0.7590	47	38	0.5529
7E−5	12	73	0.8588	0.7945	50	35	0.5882
8E−5	26	59	0.6941	0.8644	53	32	0.6235
9E−5	39	46	0.5412	0.9674	55	30	0.6471
10E−5	42	43	0.5059	1.0000	59	26	0.6941
11E−5	57	28	0.3294	0.8684	60	25	0.7059
12E−5	70	15	0.1765	0.8000	62	23	0.7294
13E−5	75	10	0.1176	0.7800	65	20	0.7647

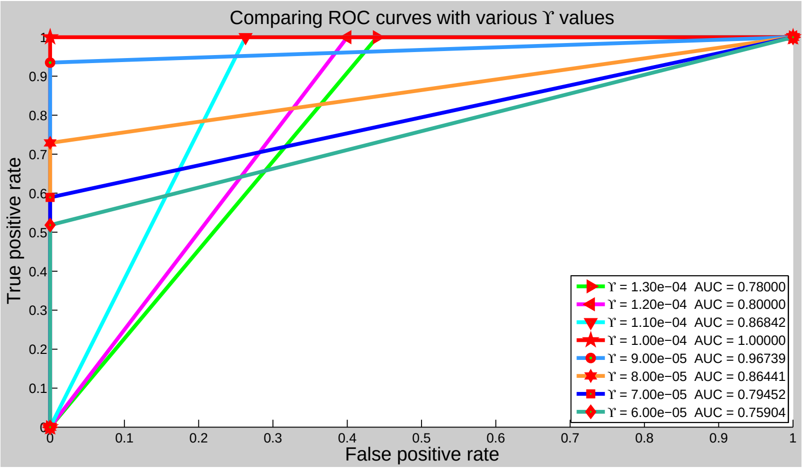
$\Upsilon = 7\text{e}{-5}$ be regarded as abnormal event's frame. With the obtained data, as shown in Table 3, we can easily verify that the sensitivity endures $(73/(12+73)) \approx 0.8588$ and the specificity goes on as $(50/(50+35)) \approx 0.5882$. Similarly, we can move Υ points forward and then observe their effects on sensitivity, AUC, and specificity as recorded in Table 3 as well as shown graphically in Fig. 9. Complex symbols N_Υ, A_Υ, and AUC$_\Upsilon$ in Table 3 represent total number of normal frames, total number of abnormal frames, and AUC put on threshold Υ, respectively.

An ROC curve is a graphical visualization of the true positive rate (plotted on the X-axis) as a function of the false positive rate (plotted on the Y-axis) of a classifier system set of classifiers. ROC curves are useful to visualize and compare the performances of classifier methods. A classifier produces a single ROC point. An AUC is



(a)

Fig. 9. Using the data in Table 3, the diverse choices of thresholds between sensitivity and specificity are depicted in (a); whereas corresponding AUC values of thresholds are exposed in (b).



(b)

Fig. 9. (*Continued*)

a common metric, a measure of test accuracy, that can be used to compare different tests or indicator variables. AUC values will always satisfy the inequality of $0 \leq \text{AUC} \leq 1$. Larger AUC values indicate better classifier performance. An AUC = 1 represents a perfect test. At threshold $\Upsilon = 10\text{E}{-5}$, we can get such result which is the ground truth case. From both Table 3 and Fig. 9, it is notable that we can develop the sensitivity in a positive way by moving the cutoff point to a lower Υ value, i.e., we are acting in a certain way so as to acquire the criterion for a positive test more strict and thorough. We can get a better specificity by moving the cutoff point to a higher Υ value, i.e., we are acting in a certain way so as to acquire the criterion for a positive test less strict and thorough. We can undergo a change for the definition of a positive test to turn to profitable account one but the other will become smaller or lose substance. Thus there should be a tradeoff between sensitivity and specificity. And threshold will depict with an illustration of that tradeoff between them. The effect of threshold on AUC is also important to take into account. Contemplating all stuffs together and rounding them, we would conclude that the threshold Υ_{V_1} in Fig. 8(b) would range from $\Upsilon = 8\text{E}{-5}$ to $\Upsilon = 11\text{E}{-5}$ in the preferable light of sensitivity and specificity tradeoff with AUC.

3. Experimental Results

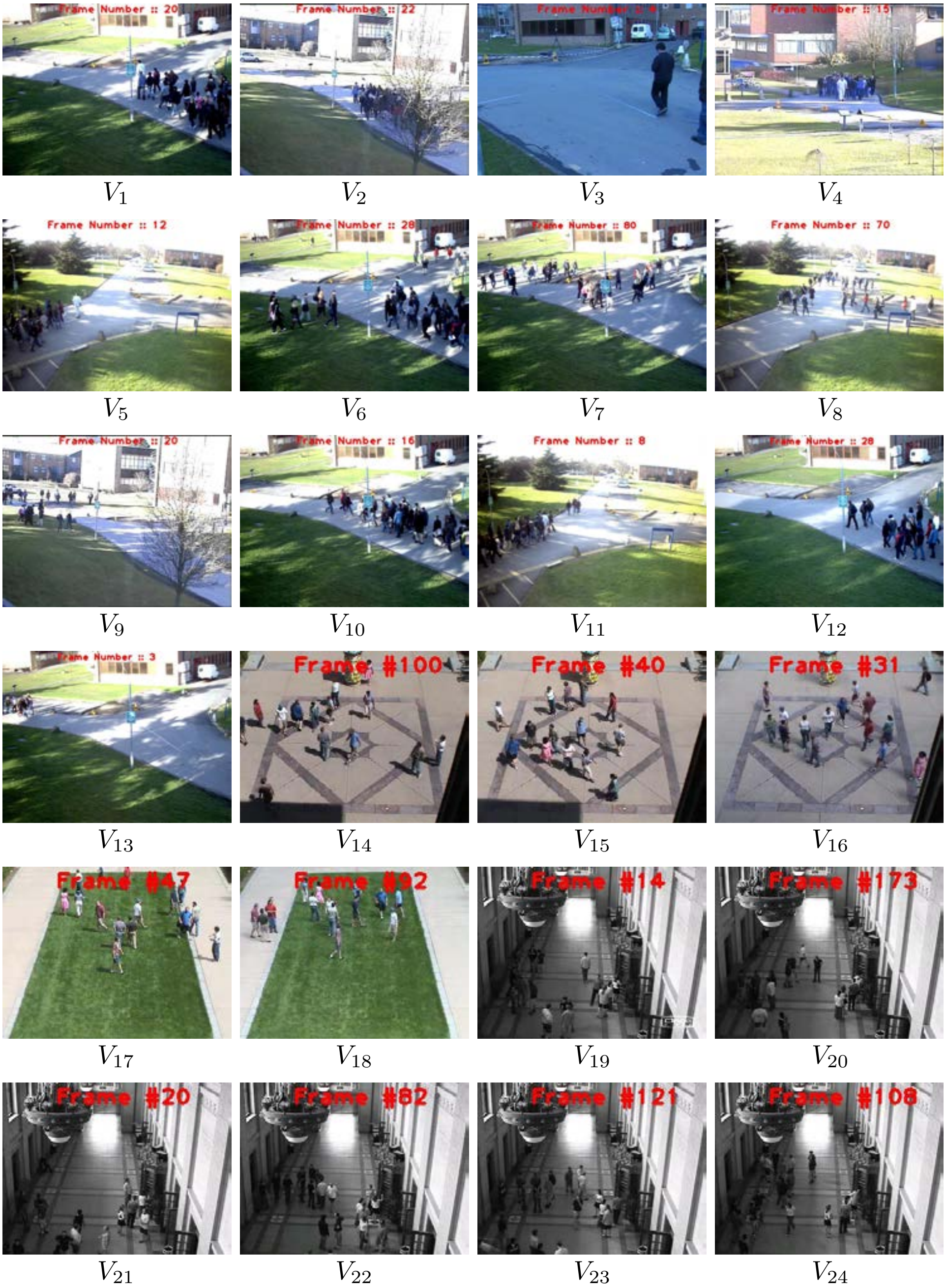
3.1. *Used datasets*

To conduct our experiment we have looked upon two publicly available benchmark datasets namely PETS2009[49] dataset with a frame size of 768 × 576 pixels and

UMN[16] dataset with a frame size of 320×240 pixels. The publicly available PETS2009[49] dataset considers crowd image analysis and includes crowd count and density estimation, tracking of individual(s) within a crowd, and detection of separate flows and specific crowd events.[49] Another publicly available dataset of normal and abnormal crowd videos from University of Minnesota (also called UMN dataset)[16] comprises the videos of 11 different scenarios of escape events in three different indoor and outdoor (i.e., lawn, indoor, and plaza) scenes. A total of 13 flow-detection/event-detection videos (V_1–V_{13} in Fig. 10) from PETS2009[49] and 11 different scenarios (V_{14}–V_{24} in Fig. 10) from UMN[16] datasets have been employed to evaluate the performance of our approach.

3.2. *Detection of various flows and anomalies*

The quantitative results of various flows detection from 24 videos using our algorithm have been presented in Table 4. Detection results of two sample videos namely V_2 and V_7 have been exhibited in Fig. 11. All estimated values of S, M_{xy}, δ_r, and polynomial fitting of δ_r using $\theta = 11$ have been plotted. Numerical numbers inscribed in the green rectangles of curve indicate frame numbers. Video V_2 concerns the unified crowd flow (single flow) and then the splitting of crowd (multiple flows). Initially, there existed singe flow of crowd emerging in the same direction in the video frames, e.g., frame number 22. Then the stream of crowd tended to diverse in different directions, e.g., frames 57 and 94. Since the crowd diverted in dissimilar directions, there must exist more than one flow, e.g., in frame number 94 there existed three distinct flows. From the polynomial fitted δ_r values curve of V_2, it is noticeable that there is a raising peak about frame number 50. This is because initially there was unified flow and estimated δ_r values were low. When the crowd tended to diverse, multiple flows existed and hence got reflected on δ_r values. But original δ_r values did not show significant performance widely due to optical flow tracking errors. Upon polynomial fitting with $\theta = 11$ the curve gives clear evidence of those existing flows. Polynomial fitting curve also shows that when the crowd is further split from two groups (two flows) to three different groups (three flows), the change in δ_r values further increased, i.e., frame number 94. A typical threshold $\Upsilon_{V_2} = 2.3423\text{e}{-5}$, obtained by Eq. (14), at frame level 49 can differentiate single flow and multiple flows. Video V_7 describes several groups of people from different directions entered into the scene with various movements (i.e., multiple flows). As time passed various groups were merged (i.e., single or zero flows) and then split suddenly (i.e., multiple flows). Initially, there existed multiple flows of crowd in different locations in the frame, e.g., frame number 80. People from various places tended to gather in a certain location in the frame. At some moment all people came close to each other on that location, e.g., frame number 207. In such case, there will exist zero flows of crowd. But if some persons would move randomly, then there would be single or multiple flows. After gathering suddenly people left their places in



V_1 V_2 V_3 V_4

V_5 V_6 V_7 V_8

V_9 V_{10} V_{11} V_{12}

V_{13} V_{14} V_{15} V_{16}

V_{17} V_{18} V_{19} V_{20}

V_{21} V_{22} V_{23} V_{24}

Fig. 10. Random sample frames of the used videos from PETS2009 and UMN datasets.

Table 4. Detection records of the used videos as sampled in Fig. 10.

	Values of parameters			Flow detection results of our algorithm from PETS2009[49] and UMN[16] benchmark datasets	
Videos	θ	Υ	ξ	Frames of single flow	Frames of multiple flows
V_1	11	0.000090273	—	1–40	41–85
V_2	11	0.000023423	—	1–48	49–112
V_3	8	0.000106679	—	1–22	23–82
V_4	11	0.000013698	—	1–34	35–115
V_5	11	0.000031325	—	112–115	1–111
V_6	11	0.000029788	—	0	1–96
V_7	11	0.000031956	0.0000580799	162–295	1–161, 296–337
V_8	11	0.000015842	0.0000429978	138–219, 262–283	1–137, 220–261, 284–337
V_9	16	0.000016585	0.0000551351	1, 80–112, 160–234, 287–302	2–79, 113–159, 235–286, 303–338
V_{10}	11	0.000025536	—	0	1–140
V_{11}	11	0.000019242	—	61–84	1–60, 85–130
V_{12}	11	0.000028327	0.0000743489	1–4	5–95
V_{13}	8	0.000046137	0.0000845098	0	1–99
V_{14}	13	0.000001958	0.0000084838	1–3, 32–48, 234–293, 374–428, 500–531, 617–622	4–31, 49–233, 294–373, 429–499, 532–616, 623
V_{15}	11	0.000001958	0.0000068333	0	1–659
V_{16}	16	0.000001731	0.0000069212	6–18, 55–81, 401–479, 555–612, 665–698, 771–776	1–5, 19–54, 82–400, 480–554, 613–664, 699–770, 777–779
V_{17}	14	0.000001693	0.0000075468	403–448, 581–588, 606	1–402, 449–580, 589–605
V_{18}	11	0.000001483	0.0000077105	0	1–801
V_{19}	16	0.000001009	0.0000033378	1–9, 16–31, 245–295, 425–433, 479–531	10–15, 32–244, 296–414, 434–478, 532, 533
V_{20}	9	0.000001397	0.0000061651	0	1–608
V_{21}	16	0.000001013	0.0000048277	6–16, 55–67, 407–450, 609–625, 735–739	1–5, 17–54, 68–406, 451–608, 625–734
V_{22}	13	0.000001083	0.0000041014	562, 563	1–561
V_{23}	11	0.000001573	0.0000052263	814–834	1–813, 835–839
V_{24}	16	0.000001011	0.0000034976	12–50, 66–110, 210–329, 390–429, 576–597, 627–642	1–11, 51–65, 111–209, 330–389, 430–575, 598–626, 643–644

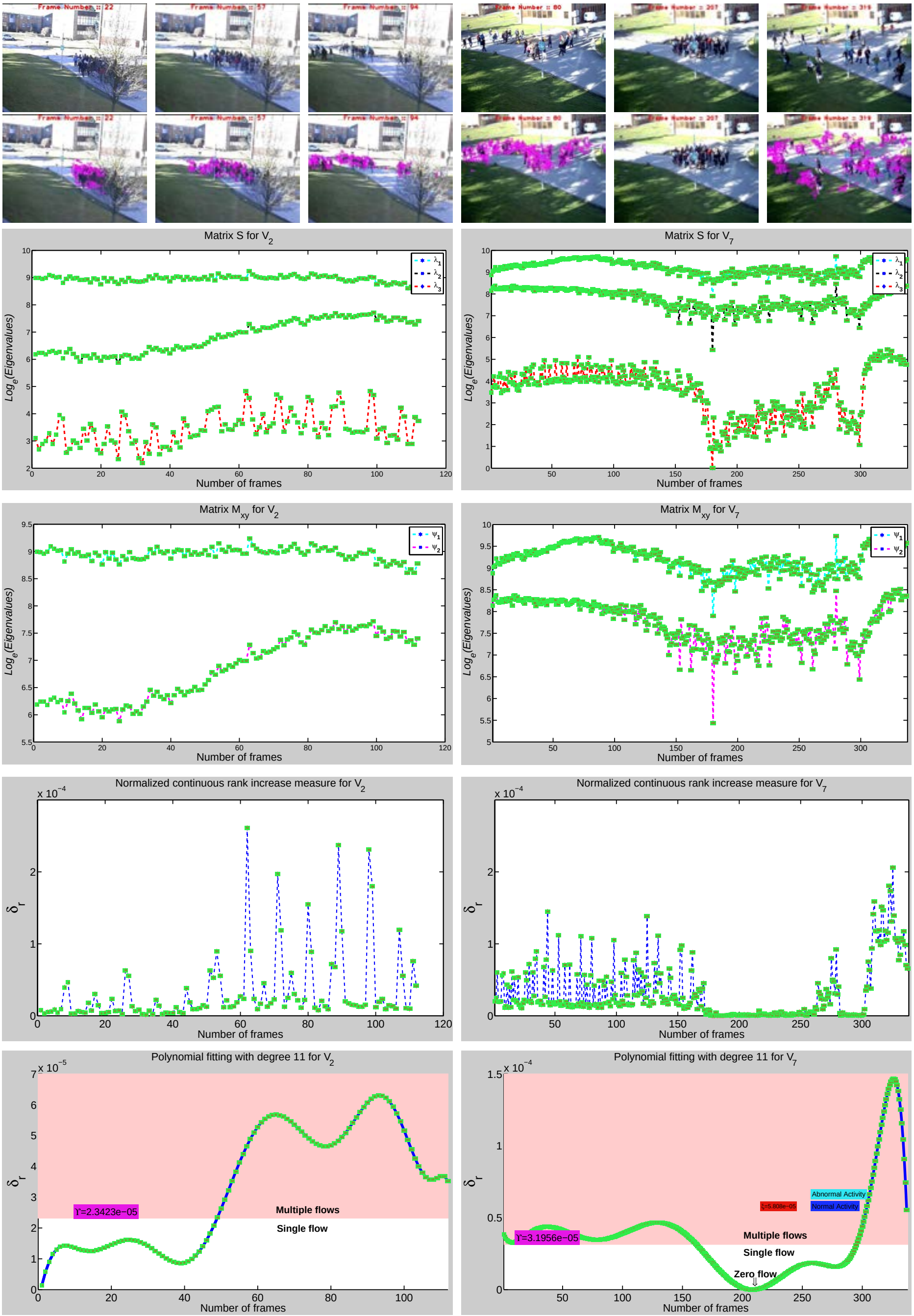


Fig. 11. Sample base frames followed by functioning of our method for the estimated optical flows, eigenvalues of S and M_{xy}, normalized continuous rank increase measure δ_r values, flows as well as events detection results of V_2 and V_7.

random directions, e.g., frame number 319. In this case there existed multiple flows of crowd. The situations have been detected by our algorithm. All estimated values of S, M_{xy}, δ_r, and polynomial fitting of δ_r using $\theta = 11$ have been plotted. A typical threshold $\Upsilon_{V_7} = 3.1956\text{E}{-5}$, obtained by Eq. (14), can be used to make different multiple flows, single flow, and zero flows. It is discernible that such kind of crowd flows can help to detect potential panic situations when the crowd motion patterns become anomalous. For example, a threshold $\xi = 5.808\text{E}{-5}$ can differentiate global abnormal behaviors of crowd from this video.

Detection results of two sample videos namely V_3 and V_{19} have been provided in Fig. 12. All estimated values of S, M_{xy}, δ_r, and polynomial fitting of δ_r with $\theta = 8$ and $\theta = 16$ have been plotted. Video V_3 complies with some scenarios of persons' random movement. A person entered into the scene which exhibited single flow. When other persons entered into the scene with random movements then multiple flows dominated throughout video frames. A typical threshold $\Upsilon_{V_3} = 0.00010668$, attained at Eq. (14), has been differentiated into single flow and multiple flows. Video V_{19} explains some scenarios of crowd random movements and a sudden split, i.e., people were moving randomly and once upon a sudden split they left their places with very quick motion due to sudden panic. Accordingly, multiple flows prevail the whole video sequences followed by a global abnormal behavior. Typical $\Upsilon_{V_{19}} = 1.009\text{e}{-6}$, gained in Eq. (14), can be used to make difference between single flow and multiple flows. Some false alarms also existed due to change of lighting condition variation and shadows. It is visible that in all videos from V_{14} to V_{24} multiple flows of crowd prevail throughout the video sequences and global abnormal activities (escape events) occurred in the subparts of them, which hint that multiple flows are the necessary condition but not the sufficient condition to detect any aberrancy. Threshold Υ has been used to detect multiple flows. But any kind of multiple flows of crowd would not lead to an abnormal event. Consequently, we need to apply another threshold ξ to detect global abnormal events. For instance, $\xi = 3.3378\text{E}{-6}$ has differentiated normal and abnormal activities of crowd in V_{19}. At above $\xi = 3.3378\text{E}{-6}$ all frames belong to abnormal activity.

3.3. *Performance evaluation for detection of various flows*

We have used AUC as the metric to evaluate the performance of multiple flows recognition. A larger AUC conveys a meaning for a better performance in robustness of multiple flows detection. Frames of multiple flows are considered as positive frames while frames of single flow are regarded as negative frames. Figures 13 and 14 make visible ROCs for detecting frames of multiple flows in PETS2009[49] and UMN[16] datasets using a linear variation of seven threshold sets A_k and B_k where $k = \{1, 2, 3, \ldots, 7\}$, respectively. Threshold sets have been used to give expression to the effect of threshold on performance if the threshold estimation in Eq. (14) would be characterized by errors. Applying mean threshold set $\frac{1}{7}\sum_{k=1}^{7} A_k \times 10^5$ on videos

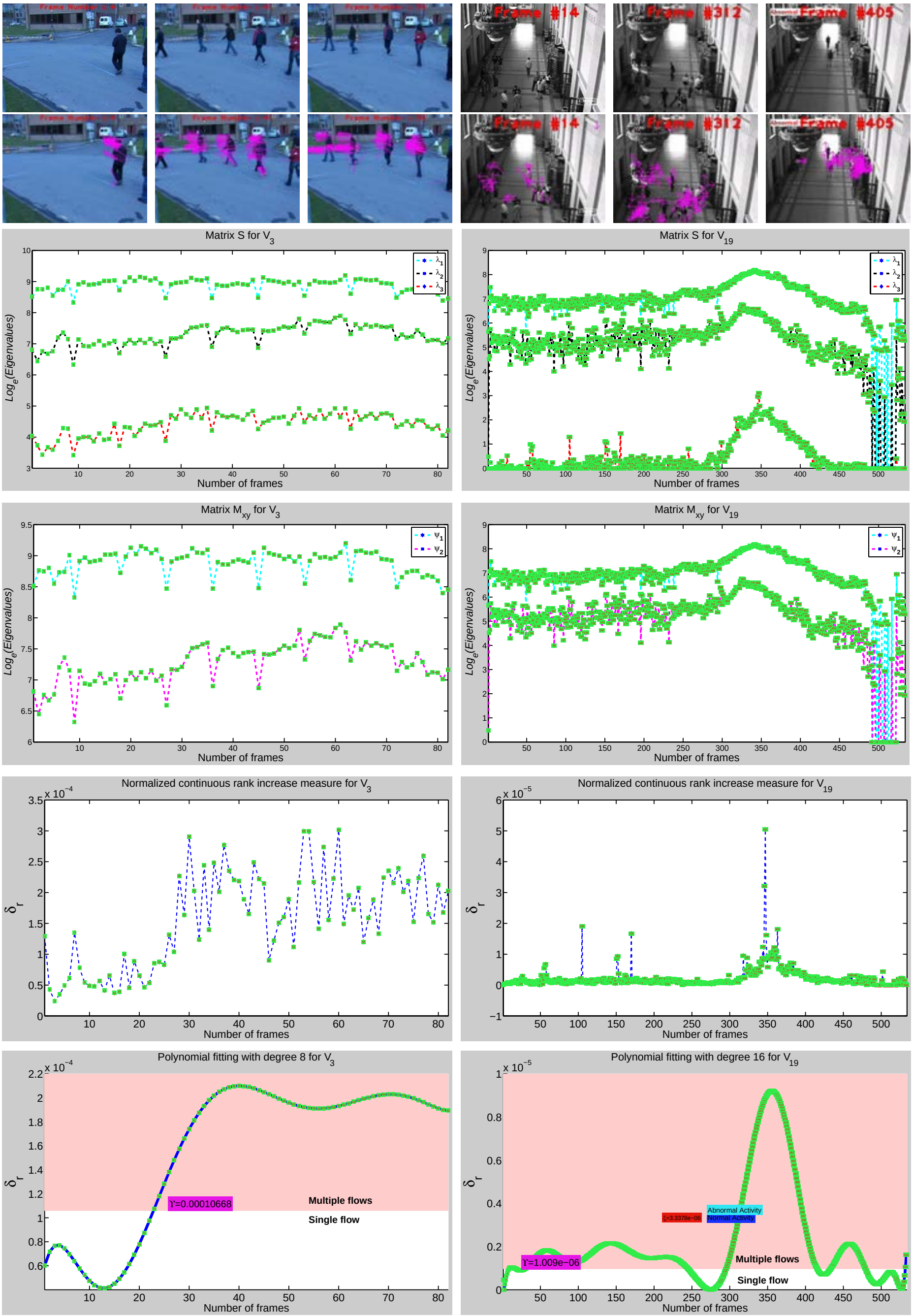


Fig. 12. Sample base frames followed by functioning of our method for the estimated optical flows, eigenvalues of S and M_{xy}, normalized continuous rank increase measure δ_r values, flows as well as events detection results of V_3 and V_{19}.

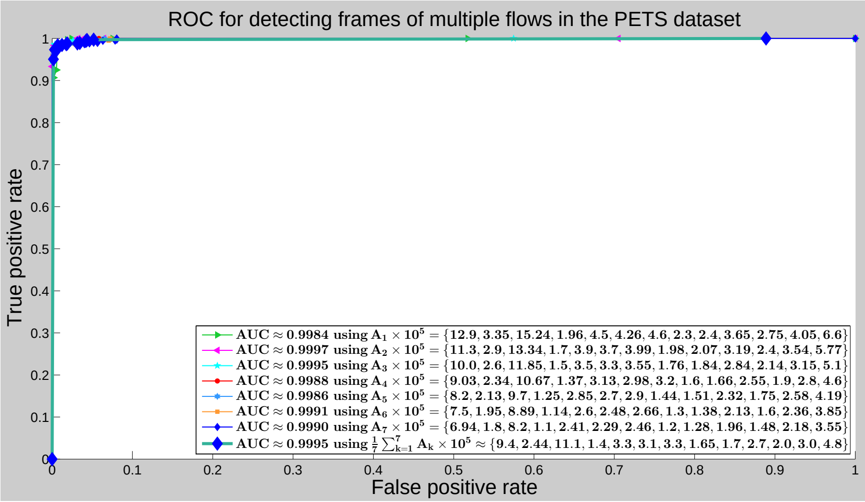


Fig. 13. Threshold effect on AUC to detect frames of multiple flows from PETS2009 dataset.

from V_1 to V_{13}, we have estimated true positive and false positive rates. On sorting true positive rate (recall rate) and false positive rate as well as adding the lower left point [0,0] and upper right point [1,1], we have plotted the ROC curve and estimated $\mathrm{AUC} = 0.9995$ for PETS2009[49] dataset using trapezoidal numerical integration method.[56] Points [0,0] and [1,1] act for the strategies of never issuing positive classification and truly issuing positive classification, respectively. Similarly, laying on mean threshold set $\frac{1}{7}\sum_{k=1}^{7} B_k \times 10^6$ on videos from V_{14} to V_{24}, the ROC curve

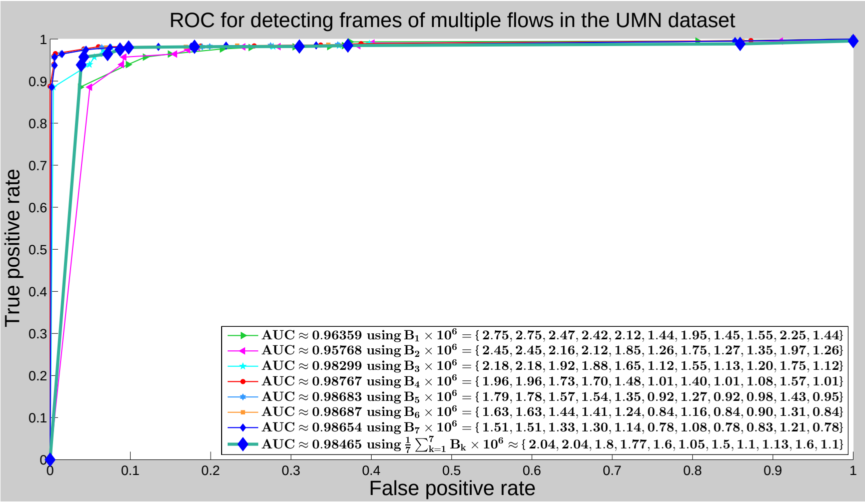


Fig. 14. Threshold effect on AUC to detect frames of multiple flows from UMN dataset.

provides AUC $= 0.98465$ for UMN[16] dataset. In addition, considering mean threshold sets the number of individual false alarm events was recorded as $2 + 2 = 4$ emanated from V_5 and V_9 in PETS2009[49] dataset, while $5 + 6 + 2 + 4 + 4 + 1 + 6 = 28$ flowed from V_{14}, V_{16}, V_{17}, V_{19}, V_{21}, V_{23}, and V_{24} in UMN[16] dataset. In brief, the performance of PETS2009[49] dataset was better than that of UMN,[16] as the false positive rate for UMN[16] dataset was a bit high.

3.4. *Interesting events detection from PETS2009 dataset*

Based on the context an event may be interesting. For example, a walk event would not be stimulating in the supermarket premises but it would be absorbing while a marathon is running. In whatever manner, to detect interesting events, e.g., split, evacuation, and run from PETS2009[49] dataset, we have concatenated final polynomial fitted δ_r values of V_1, V_2, V_4, V_7, V_8, V_9, V_{12}, V_{13} and then overlapped with their ground truths as shown in Fig. 15. Split, evacuation, and run events are marked as positive events and the remaining events, e.g., walk, wait, and gather, are denoted as negative events. Table 5 lists qualitatively and quantitatively the outcomes of Fig. 15. Mean absolute error, mean absolute percentage error, root-mean-square error, and coefficient of variation of the RMSE are shortened by MAE, MAPE, RMSE, and CV(RMSE), respectively. Treating ground truths and mean threshold set $\frac{1}{7}\sum_{i=1}^{7} P_i \times 10^5$, we have estimated various statistical measures, e.g., recall rate, false positive rate, precision rate, accuracy, mean errors for both rising and falling

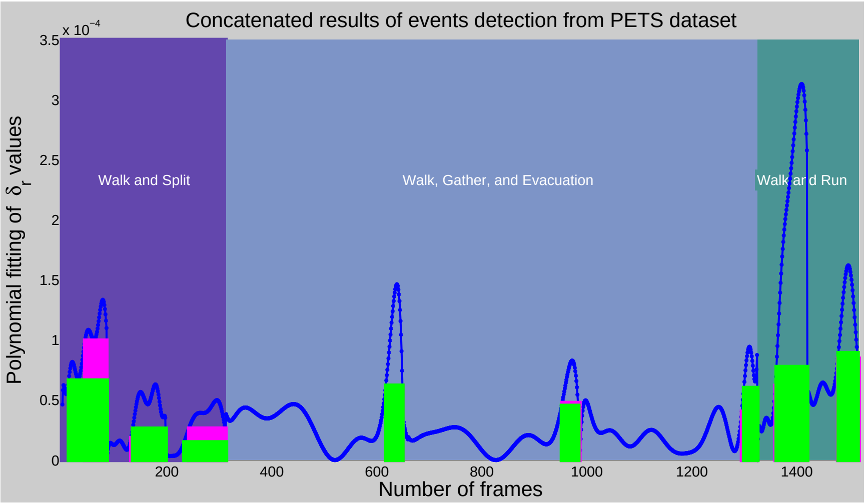


Fig. 15. Concatenated events detection results from PETS2009 dataset. Pink and green bars show ground truth and our approach, respectively. Left and right corners on the width of bar show start and end of a positive event, severally. Positive events are split, evacuation, and run. Numerical analysis of this structure is given in Table 5 (color online).

Table 5. Qualitative and quantitative analysis of Fig. 15.

	Videos V_v where $v = \{1, 2, 4, 7, 8, 9, 12, 13\}$								
	V_1	V_2	V_4	V_7	V_8	V_9	V_{12}	V_{13}	
Measures	Split			Evacuation			Run		⌈Mean⌉
Ground truth frame start (S_{V_v})	44	47	45	305	304	309	35	59	144
Ground truth frame end (E_{V_v})	85	112	115	337	337	338	95	99	190
Threshold ($\xi_{V_v} = \frac{1}{7}\sum_{i=1}^{7} P_i \times 10^5$)	9.416	2.440	1.430	6.060	4.484	5.750	7.750	8.814	5.769
ϖ	0	0	0	0	0	0	0	0	0
First detected abnormal frame (F_{V_v})	12	49	35	304	302	312	36	59	139
Last detected abnormal frame (L_{V_v})	85	112	115	336	334	338	95	96	189
Number of false positive frames (f_p)	32	2	10	1	2	3	1	0	7
Number of true positive frames (t_p)	41	65	70	31	30	29	60	37	46
Number of false negative frames (f_n)	0	0	0	1	3	0	0	3	1
Number of true negative frames (t_n)	12	45	35	304	302	306	34	59	138
Sum ($T_f = t_p + t_n + f_p + f_n$)	85	112	115	337	337	338	95	99	190
Recall rate ($t_p/(t_p + f_n)$)	1.0000	1.0000	1.0000	0.9688	0.9091	1.0000	1.0000	0.9250	0.9754
Specificity ($t_n/(t_n + f_p)$)	0.2727	0.9574	0.7778	0.9967	0.9934	0.9903	0.9714	1.0000	0.8700
False positive rate (1-specificity)	0.7273	0.0426	0.2222	0.0033	0.0066	0.0097	0.0286	0	0.1300
Precision rate ($t_p/(t_p + f_p)$)	0.5616	0.9701	0.8750	0.9688	0.9375	0.9063	0.9836	1.0000	0.9004
Accuracy (ACC $= (t_p + t_n)/T_f$)	0.6235	0.9821	0.9130	0.9941	0.9852	0.9911	0.9895	0.9697	0.9310
$\lvert S_{V_v} - F_{V_v}\rvert$	32	2	10	1	2	3	1	0	7
$\frac{\lvert S_{V_v} - F_{V_v}\rvert}{S_{V_v}}$	0.7272	0.0426	0.2222	0.0033	0.0066	0.0097	0.0286	0	0.1300
$(S_{V_v} - F_{V_v})^2$	1,024	4	100	1	4	9	1	0	143
$\lvert E_{V_v} - L_{V_v}\rvert$	0	0	0	1	3	0	0	3	1

Table 5. (*Continued*)

	Videos V_v where $v = \{1, 2, 4, 7, 8, 9, 12, 13\}$								
	V_1	V_2	V_4	V_7	V_8	V_9	V_{12}	V_{13}	
Measures	Split			Evacuation			Run		$\lceil$Mean$\rceil$
$\dfrac{\lvert E_{V_v} - L_{V_v}\rvert}{E_{V_v}}$	0	0	0	0.0030	0.0089	0	0	0.0303	0.0053
$(E_{V_v} - L_{V_v})^2$	0	0	0	1	9	0	0	9	1
MAE at rising edge	$\Lambda_r = \frac{1}{8} \sum_{v=1}^{13} \lvert S_{V_v} - F_{V_v}\rvert = \frac{51}{8} \approx 7$								
MAPE at rising edge	$\Lambda'_r = \frac{100}{8} \sum_{v=1}^{13} \left(\frac{\lvert S_{V_v} - F_{V_v}\rvert}{S_{V_v}}\right) \approx 13\%$								
RMSE at rising edge	$\Gamma_r = \sqrt{\frac{1}{11} \sum_{v=1}^{13} (S_{V_v} - F_{V_v})^2} = \sqrt{\frac{1143}{8}} \approx 12$								
CV(Γ_r) at rising edge	$\mathrm{CV}(\Gamma_r) = \frac{\Gamma_r}{\frac{1}{8} \sum_{v=1}^{13} S_{V_v}} = \frac{12}{144} \approx 0.0833$								
MAE at falling edge	$\Lambda_f = \frac{1}{8} \sum_{v=1}^{13} \lvert E_{V_v} - L_{V_v}\rvert = \frac{7}{8} \approx 1$								
MAPE at falling edge	$\Lambda'_f = \frac{100}{8} \sum_{v=1}^{13} \left(\frac{\lvert E_{V_v} - L_{V_v}\rvert}{E_{V_v}}\right) \approx 0.53\%$								
RMSE at falling edge	$\Gamma_f = \sqrt{\frac{1}{8} \sum_{v=1}^{13} (E_{V_v} - L_{V_v})^2} = \sqrt{\frac{19}{8}} \approx 2$								
CV(Γ_f) at falling edge	$\mathrm{CV}(\Gamma_f) = \frac{\Gamma_f}{\frac{1}{8} \sum_{v=1}^{13} E_{V_v}} = \frac{3}{190} \approx 0.0158$								
ROC curve analysis	*Area under the curve*, AUC $\approx$ 0.9995 from Fig. 16								

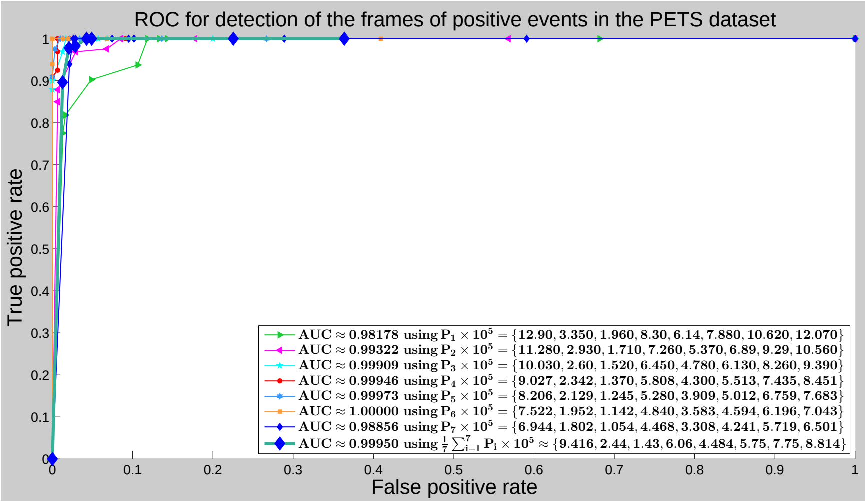


Fig. 16. Threshold effect on AUC to detect frames of positive events from PETS2009 dataset.

edges, etc. Since we are dealing with only positive values, the coefficient of variation of the root-mean-square error, a unitless measure, has been estimated as well. The effects of threshold changes (e.g., sample seven threshold sets from P_1 to P_7) on the performance to detect frames of interesting events from PETS2009[49] dataset have been exemplified in Fig. 16. Mean ROC curve (marked bold) in Fig. 16 estimates AUC $= 0.9995$ from PETS2009[49] dataset.

3.5. *Escape events detection from UMN dataset*

To detect escape events from UMN[16] dataset, we have concatenated final polynomial fitted δ_r values from V_{14} to V_{24} and then overlapped with their ground truths as shown in Fig. 17. Escape events are marked as positive events and the rest of the events, e.g., wanderings of people, are denoted as negative events. Table 6 records the outcomes of qualitative and quantitative analysis of Fig. 17. Betaking ground truths and mean threshold set $\frac{1}{7}\sum_{i=1}^{7} U_i \times 10^6$, we have estimated various statistical measures as before. The consequences of threshold alternations (e.g., sample seven threshold sets from U_1 to U_7) on performance to detect frames of escape events from UMN[16] dataset have been illustrated in Fig. 18. Mean ROC curve (marked bold) in Fig. 18 estimates AUC $= 0.97833$ for UMN[16] dataset. Table 7 depicts the mean numerical values of several measures to detect interesting events using PETS2009[49] and UMN[16] datasets. Averaged numerical values show our method's effectiveness to detect interesting events directly without segmentation in the crowd scenes.

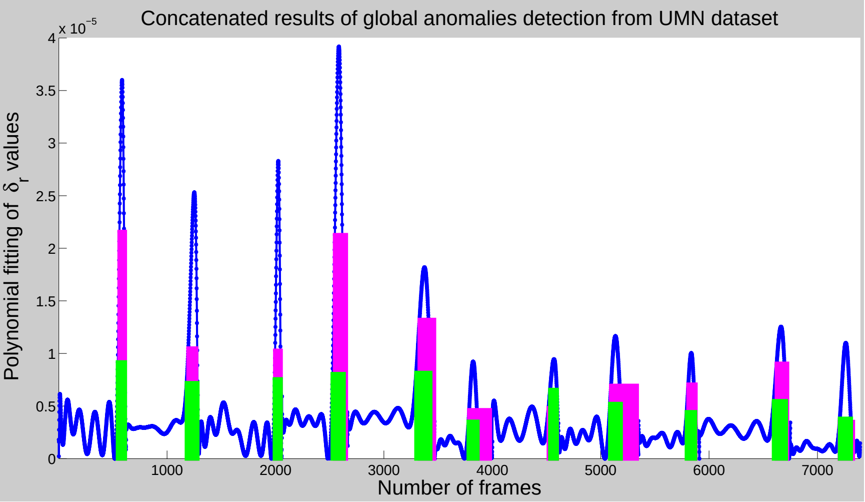


Fig. 17. Blue curve shows the concatenated detection results from UMN dataset. Pink and green bars denote ground truth and our method, severally. Left and right corners on the width of bar hint start and end of a positive event, respectively. Positive events are escape events. Numerical analysis of this structure is given in Table 6 (color online).

3.6. *Performance comparison to find events in PETS2009 dataset*

Briassouli and Kompatsiaris[37] presented an approach based on properties of the data in the Fourier domain to detect new events in crowds. Random crowd motion was encoded in the phase of the Fourier coefficients, and statistical sequential change detection methods were applied to detect events in crowds. To detect behavior changes in crowded scenes, Almeida and Jung[42] proposed an approach based on the extraction of foreground blobs with shadow suppression to identify crowd members, and large-scale optical flow to obtain the displacement vector field. As a performance comparison both FourierCoefficient[37] and SimilarityMetric[42] were tested against videos V_7, V_{12}, and V_{13} of PETS2009[49] dataset. The quantitative results analysis and comparison of the global abnormal behavior detection for our method as well as methods of FourierCoefficient[37] and SimilarityMetric[42] have been demonstrated in Fig. 19. Upmost images of V_7 in Fig. 19 illustrate the ground truth and various detection results of different methods. The frame number 301 points to the ground truth for a change in crowd behavior. FourierCoefficient,[37] SimilarityMetric,[42] and our proposed model detected the change in crowd behavior at frame numbers 311, 308, and 306, respectively. Schematic diagram depicts the video sequence line with the ground truth and detection frames using all three approaches. Green and red colors demonstrate the normal and abnormal frames, respectively. Along with schematic diagrams the middle images of V_{12} with ground truth frame 34 as well as underneath images of V_{13} with ground truth frame 54 in Fig. 19 depict the detection

Table 6. Qualitative and quantitative analysis of Fig. 17.

	Various videos V_v where $v = \{14, 15, 16, \ldots, 24\}$											
Different measures	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}	V_{24}	⌈**Mean**⌉
Ground truth frame start (S_{V_v})	561	573	715	487	662	319	519	487	461	713	455	542
Ground truth frame end (E_{V_v})	612	647	767	590	797	511	596	725	528	810	578	651
Threshold $\left(\xi_{V_v} = \frac{1}{7}\sum_{i=1}^{7} U_i \times 10^6\right)$	8.84	7.12	7.14	7.90	8.04	3.50	6.50	5.04	4.30	5.50	3.65	6.14
ϖ	0	0	0	0	0	0	0	0	0	0	0	0
First detected abnormal frame (F_{V_v})	544	555	709	465	628	312	530	476	443	685	455	528
First detected abnormal frame (F_{V_v})	544	555	709	465	628	312	530	476	443	685	455	528
Last detected abnormal frame (L_{V_v})	611	657	766	569	762	397	596	577	524	799	559	620
Number of false positive frames (f_p)	17	18	6	22	34	7	11	11	18	28	0	16
Number of true positive frames (t_p)	50	64	51	82	100	78	77	90	63	86	104	77
Number of false negative frames (f_n)	1	10	1	21	35	114	0	148	4	11	19	34
Number of true negative frames (t_n)	555	567	721	481	632	334	520	490	478	714	521	547
Sum ($T_f = t_p + t_n + f_p + f_n$)	623	659	779	606	801	533	608	739	563	839	644	673
Recall rate ($t_p/(t_p + f_n)$)	0.9804	0.8649	0.9808	0.7961	0.7407	0.4063	1.0000	0.3782	0.9403	0.8866	0.8455	0.8018
Specificity ($t_n/(t_n + f_p)$)	0.9703	0.9692	0.9917	0.9563	0.9489	0.9795	0.9793	0.9780	0.9637	0.9623	1.0000	0.9727
False positive rate (1-specificity)	0.0297	0.0308	0.0083	0.0437	0.0511	0.0205	0.0207	0.0220	0.0363	0.0377	0.0000	0.0274
Precision rate ($t_p/(t_p + f_p)$)	0.7463	0.7805	0.8947	0.7885	0.7463	0.9176	0.8750	0.8911	0.7778	0.7544	1.0000	0.8339
Accuracy (ACC $= (t_p + t_n)/T_f$)	0.9711	0.9575	0.9910	0.9290	0.9139	0.7730	0.9819	0.7848	0.9609	0.9535	0.9705	0.9261
$\lvert S_{V_v} - F_{V_v}\rvert$	17	18	6	22	34	7	11	11	18	28	0	14
$\frac{\lvert S_{V_v} - F_{V_v}\rvert}{S_{V_v}}$	0.0303	0.03141	0.0084	0.0452	0.0514	0.0219	0.0212	0.0226	0.0391	0.0393	0	0.0283
$(S_{V_v} - F_{V_v})^2$	289	324	36	484	1,156	49	121	121	324	784	0	196
$\lvert E_{V_v} - L_{V_v}\rvert$	1	10	1	21	35	114	0	148	4	11	19	31
$\frac{\lvert E_{V_v} - L_{V_v}\rvert}{E_{V_v}}$	0.0017	0.0155	0.0013	0.0356	0.0440	0.2231	0	0.2042	0.0076	0.0136	0.0329	0.0527
$(E_{V_v} - L_{V_v})^2$	1	100	1	441	1,225	12,996	0	21,904	16	121	361	961

Table 6. (*Continued*)

	Various videos V_v where $v = \{14, 15, 16, \ldots, 24\}$										
Different measures	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}	V_{24}	$\lceil$**Mean**$\rceil$
MAE at rising edge	$\Lambda_r = \frac{1}{11}\sum_{v=14}^{24} \lvert S_{V_v} - F_{V_v}\rvert = \frac{172}{11} \approx 16$											
MAPE at rising edge	$\Lambda'_r = \frac{100}{11}\sum_{v=14}^{24}\left(\frac{\lvert S_{V_v} - F_{V_v}\rvert}{S_{V_v}}\right) \approx 2.83\%$											
RMSE at rising edge	$\Gamma_r = \sqrt{\frac{1}{11}\sum_{v=14}^{24}(S_{V_v} - F_{V_v})^2} = \sqrt{\frac{3688}{11}} \approx 19$											
CV(Γ_r) at rising edge	$CV(\Gamma_r) = \frac{\Gamma_r}{\frac{1}{11}\sum_{v=14}^{24} S_{V_v}} = \frac{19}{542} \approx 0.0351$											
MAE at falling edge	$\Lambda_f = \frac{1}{11}\sum_{v=14}^{24} \lvert E_{V_v} - L_{V_v}\rvert = \frac{364}{11} \approx 34$											
MAPE at falling edge	$\Lambda'_f = \frac{100}{11}\sum_{v=14}^{24}\left(\frac{\lvert E_{V_v} - L_{V_v}\rvert}{E_{V_v}}\right) \approx 5.27\%$											
RMSE at falling edge	$\Gamma_f = \sqrt{\frac{1}{11}\sum_{v=14}^{24}(E_{V_v} - L_{V_v})^2} = \sqrt{\frac{37166}{11}} \approx 59$											
CV(Γ_f) at falling edge	$CV(\Gamma_f) = \frac{\Gamma_f}{\frac{1}{11}\sum_{v=14}^{24} E_{V_v}} = \frac{59}{651} \approx 0.0907$											
ROC curve analysis	*Area under the curve*, AUC $\approx$ 0.97833 from Fig. 18											

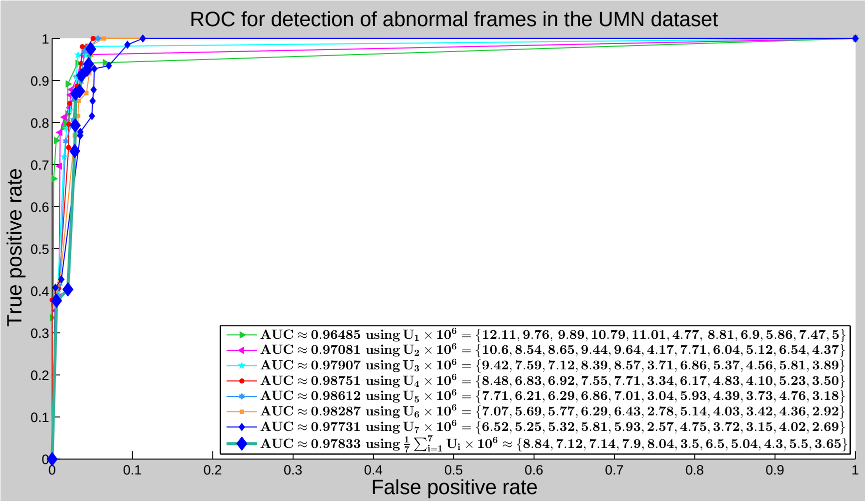


Fig. 18. Threshold effect on AUC to detect frames of positive events from UMN dataset.

results of three methods. Using video V_{12}, FourierCoefficient,[37] SimilarityMetric,[42] and proposed model can detect the start of abnormal crowd behavior at frame numbers 52, 48, and 38, respectively; whereas from video V_{13} the start of abnormal behavior can be detected at frame numbers 83, 75, and 60, respectively. In brief, our method detects the change in crowd activities at frames close to the ground truths, outperforming both FourierCoefficient[37] and SimilarityMetric.[42]

Su *et al.*[13] proposed a shear force-based method to recognize the large-scale crowd behavior from both the appearance and driven factor perspectives. As a comparative study using several videos from PETS2009[49] dataset, Su *et al.*[13] showed that their proposed shear force-based algorithm outperforms the alternative algorithms, e.g., SocialForce,[20] HOG/HOF,[9] HOG3D,[8] Cuboid3D,[18] LocalMotion,[39] and Tracklets.[10] Figure 20 demonstrates the comparison of the use of the ground truth, state-of-the-art methods, and our method to detect positive events from V_1, V_7, V_{12}, and V_{13} in PETS2009[49] dataset. Color bars are utilized to represent the recognized results of each frame in the sequences obtained by the corresponding approaches. Green and red colors represent the frames of negative (e.g., walk and gather) and positive

Table 7. Overall performance of our method to detect interesting events from PETS2009 and UMN datasets.

Datasets	Mean recall rate	Mean precision rate	AUC	Mean ACC
PETS2009[49]	**0.9754** (Table 5)	**0.9004** (Table 5)	**0.9995** (Table 5)	**93.10%** (Table 5)
UMN[16]	**0.8018** (Table 6)	**0.8339** (Table 6)	**0.9783** (Table 6)	**92.61%** (Table 6)
Grand average	**0.8886**	**0.8671**	**0.9889**	**92.86%**

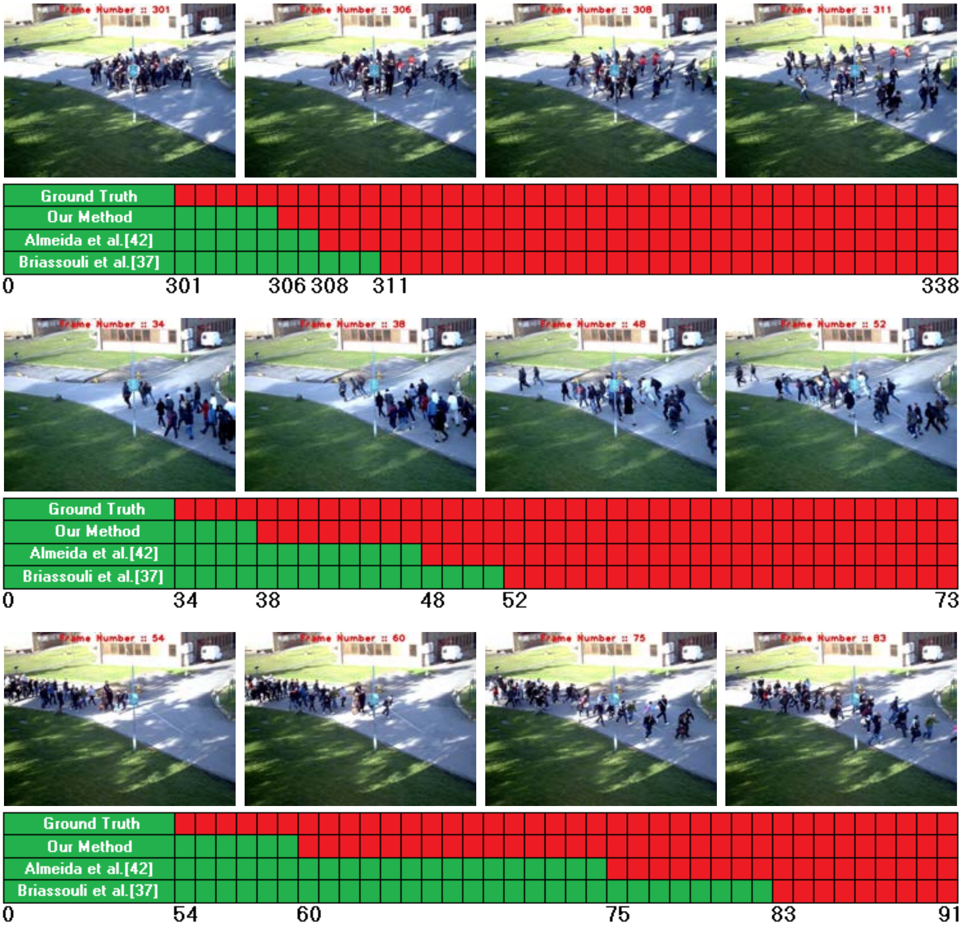


Fig. 19. The comparison of the use of ground truth, our method, and other existing methods to detect global abnormal behaviors deeming V_7, V_{12}, and V_{13} from PETS2009 dataset. Green and red colors portray normal and abnormal frames, respectively. Bold black numerical number connotes frame number. Outputs of our algorithm enhance the appearance of the best neighborhood of ground truths. In an explicit manner, our proposed method outperforms its options (color online).

(e.g., split, evacuation, and run) events, respectively. The quantitative evaluation results for detecting those positive events are illustrated in Table 8. In terms of precision rate and recall rate, ShearForce[13] and SocialForce[20] gained a comparable result with our proposed method to detect abnormal events, e.g., crowd splitting and evacuation. In whatever degree, ϖ value of SocialForce[20] was higher than that of ShearForce,[13] because the social force is not obvious in crowd walking cases.[13] HOG/HOF[9] gained balanced results in both normal (negative) and abnormal (positive) events detection, but it faces difficulties if the motion is not obvious such as crowd gathering.[13] The performance of HOG3D[8] was not as desirable as the other

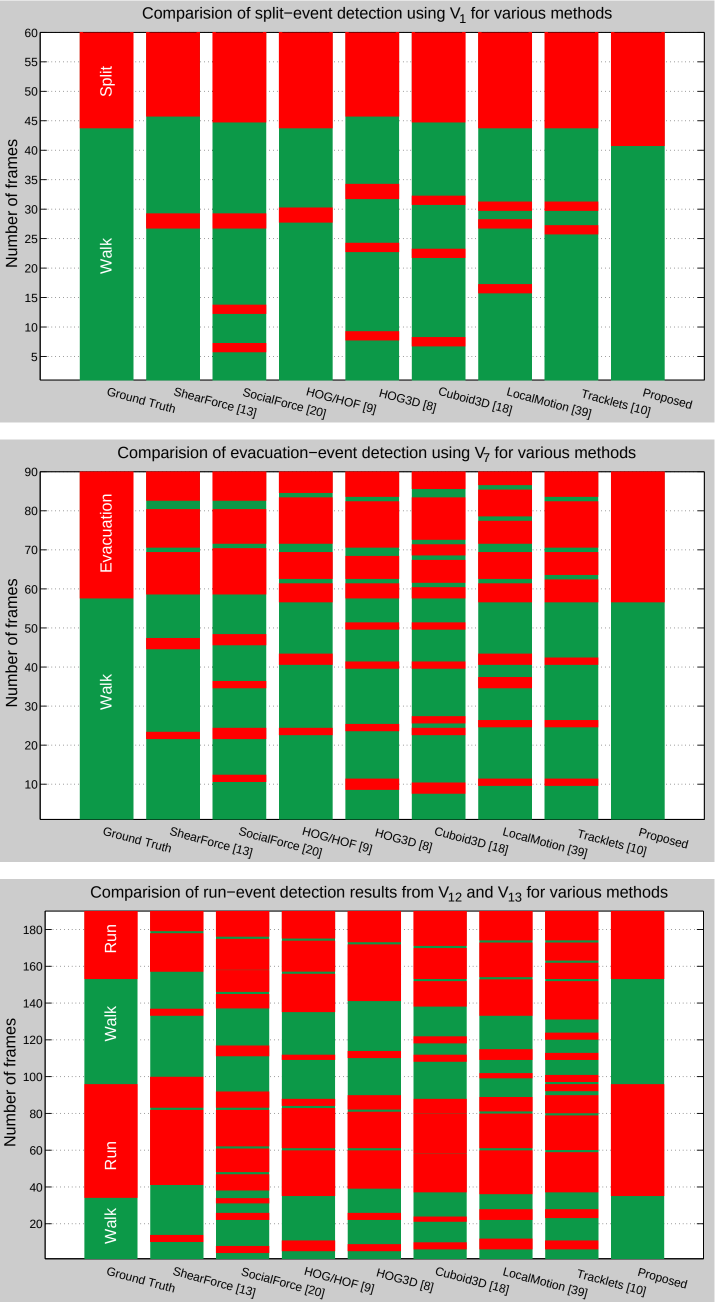


Fig. 20. The comparison of the use of the ground truth, state-of-the-art methods, and our method to detect positive events from V_1, V_7, V_{12}, and V_{13} in the PETS2009 dataset. Green and red colors portray frames from negative and positive events, respectively. Outputs of our algorithm make visible the best proximity to the ground truths, i.e., our method outperforms its key alternatives (color online).

Table 8. Performance comparison of various methods using videos of PETS2009 dataset.

	Positive events: Split, evacuation, and run						
	ϖ esteeming Fig. 20					Average	
Approaches	V_1	V_7	V_{12}	V_{13}	Total	Precision rate	Recall rate
ShearForce[13]	1	2	1	2	6	0.9206	0.9040
SocialForce[13,20]	3	4	3	1	11	0.8000	0.8827
HOG/HOF[9,13]	1	2	1	1	5	0.9100	0.8663
HOG3D[8,13]	3	4	2	1	10	0.8203	0.8233
Cuboid3D[13,18]	3	5	2	2	12	0.8357	0.8320
LocalMotion[13,39]	3	4	2	2	11	0.8537	0.8336
Tracklets[10,13]	2	3	2	3	10	0.8653	0.8683
Proposed [Table 5]	0	0	0	0	0	0.9004	0.9754

alternative methods. Because the spatial feature is overweight for this descriptor and the spatiotemporal volume construction for HOG3D analysis also leads to a lag in detection.[13] The overweight issue also exists for Cuboid3D,[18] i.e., the variation can be significant even if there is no meaningful motion. As a result, the result of Cuboid3D[18] is not desirable for the crowd with medium density.[13] LocalMotion[39] fails to exploit the spatial characteristic of the behavior, and the performance is also inferior to ShearForce.[13] Moreover, Tracklets[10] model attains a comparable result if the occlusion is not serious, e.g., crowd evacuation. But it suffers from performance degradation when new pedestrians get in the scene ceaselessly, e.g., for the crowd running and gathering cases it fails to detect the interaction among pedestrians.[13] ShearForce[13] model gains a comparable result in detecting crowd splitting, evacuation, and run events with our proposed method. Nevertheless, its ϖ value is very high as compared to our method.

3.7. *Performance comparison to detect anomalies in UMN dataset*

From Table 1 it is perceptible that UMN[16] dataset has twirled a round a common benchmark for almost all existing methods to evaluate their robustness and efficacy correlation with AUC and/or ACC. For instance, as a comparative study of events detection from UMN[16] dataset, Su *et al.*[13] showed that their shear force model surpassed SocialForce,[20] HOG/HOF,[9] HOG3D,[8] Cuboid3D,[18] LocalMotion,[39] and Tracklets.[10] The overall performance evaluation of all positive and negative ground truth events detection from UMN[16] dataset is pointed at Fig. 21. It is ascertained that our method outperformed its alternatives in terms of ϖ, AUC, and ACC. ShearForce,[13] SocialForce,[20] HOG/HOF,[9] HOG3D,[8] Cuboid3D,[18] LocalMotion,[39] Tracklets,[10] and our eigenvalue-based model (e.g., Table 6) had reported a total of $\varpi = 8$, $\varpi = 12$, $\varpi = 7$, $\varpi = 11$, $\varpi = 10$, $\varpi = 10$, $\varpi = 13$, and $\varpi = 0$, respectively, to detect anomalies from V_{14}, V_{15}, V_{17}, and V_{18}. ShearForce[13] produced a comparable result to our method for detecting aberrant positive ground truth events but it had

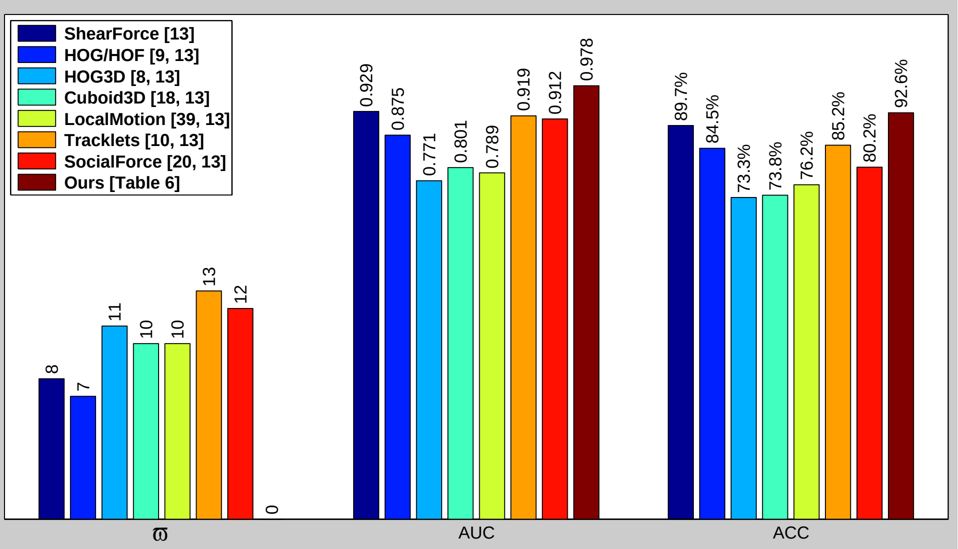


Fig. 21. Performance comparison of the state-of-the-art methods and our method to detect abnormal behaviors from UMN dataset. It reveals that our obtained results are the best in line with ground truths, i.e., our method outperforms its key alternatives in terms of ϖ, AUC, and ACC.

hit sundry detections for negative ground truth events as positive ground truth events. SocialForce[20] had a comparable AUC index too, but its ACC was much lower. It is due to the difficulty in estimating social force for normal behavior, which leads to some error detections. HOG/HOF[9] had a good performance in terms of both AUC and ACC, but like ShearForce[13] it had hit many detections for negative ground truth events as positive ground truth events. Main reason is that it failed to exploit the interaction between pedestrians in dense crowd environment. Cuboid3D,[18] LocalMotion,[39] and Tracklets[10] models underwent momentous degradation for complex motion patterns of crowd and had hit numerous detections for negative ground truth events as positive ground truth events as reflected on their AUCs and ACCs. HOG3D[8] performed inferior to others in terms of both AUC and ACC, because inside its algorithm spatial feature was too much emphasized. By contrast, our method effectuated a much better performance by exploiting both appearance and interaction of crowd behaviors, which had been gained by uniting both eigenvalue-based analysis of spatiotemporal features and a correct degree of polynomial fitting function.

As a concluding remark, beholding Figs. 19–21, it substantiates that our method outperforms some star substitute state-of-the-art methods, i.e., our obtained results are more in line with ground truths. The exceedingly leading grounds of this performance include: (i) Eigenvalue analysis of spatiotemporal features is very robust to exploit appearance and interaction of crowd behaviors; and (ii) apt degree of polynomial fitting function on δ_r values augments its performance markedly over unwanted feature points.

3.8. *Precision of algorithm and accuracy of experimental datasets*

To get our experimental datasets, we have heavily depended on the outputs of Pyramid-LK algorithm.[51–53] In this subsection, we have discussed the precision of this algorithm and the accuracy of our experimental datasets. Hereby experimental data denote the output optical flow data of an optical flow algorithm.

Experimentation frequently produces multiple measurements of the same thing, i.e., replicate measurements, which are subject to error. In statistics, an error is not a mistake. Variableness is a natural part of individuals being measured and of the measurement method. The terms precision and accuracy are sometimes used interchangeably, but they are completely two different specifications. Precision measures how well our machines are functioning, not what they are outputting. Accuracy checks how correct their output data are. In numerical analysis, precision is the resolution of a representation, typically ascertained by the number of binary or decimal digits; while accuracy is also the closeness of a reckoning to the actual value. In mathematics and science, calculating precision is essential to determine whether our instruments and measurements are working well or not. In general, precision means that a measurement gets similar results every single time it is used under unchanged conditions. But precision is limited by the random errors. It may generally be determined by repeating the measurements. Accuracy is normally defined in terms of error or inaccuracy. The precision of an optical flow algorithm gives the knowledge that whether it is just functioning well or not, irrespective of its right or wrong output data. Precision also hints that the algorithm outputs similar data every single time it is used under unchanged conditions. Accuracy checks how right the output optical flow data of the optical flow algorithm are. Both high accuracy and high precision are significant for high quality of experimental dataset. But high precision does not give evidence of high accuracy nor does high accuracy point out high precision. Figure 22 shows the statistical distribution for precision versus the nearness to a ground truth for accuracy. A more precise algorithm has a narrower distribution and a more accurate algorithm is closer to the ground truth. An algorithm is said to be valid if it is both precise and accurate.

If we wish to check the precision of Pyramid-LK algorithm,[51–53] we could run it several times without changing its parameters and record its output optical flow data every time. Note that to calculate precision, we cannot run several optical flow algorithms and compare their outputs. For simplicity, we can run Pyramid-LK algorithm[51–53] several (say five) times by considering 86 frames from V_1 of PETS2009[49] and get 85 distinct δ_r values such as in Fig. 4(b). And then we can calculate and record the geometric mean of those δ_r values for each run. In this vein, geometric means of 5.5925E−05, 5.5925E−05, 5.5925E−05, 5.5925E−05, and 5.5925E−05 can

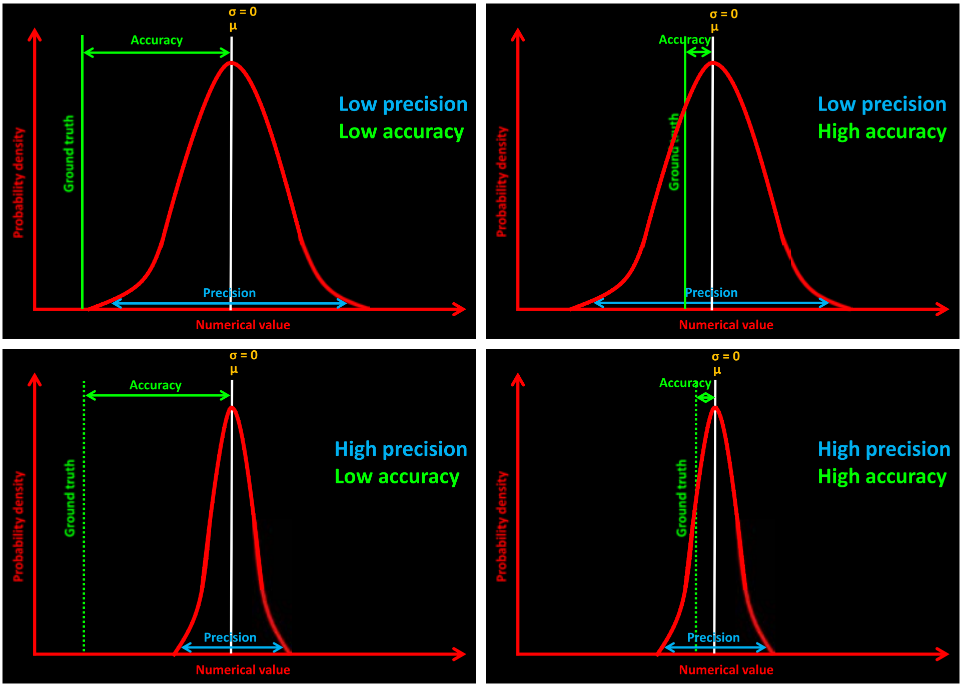


Fig. 22. The relationship between precision and accuracy. Precision implies the repeatability or reproducibility of the experimental data. Accuracy hints the proximity of the experimental data to the ground truths. μ be the mean or expectation of the distribution, whereas σ be the standard deviation. $\sigma = \pm 0$ indicates 100% precision. High precision and high accuracy are always expected.

be recorded from first-run, second-run, third-run, fourth-run, and fifth-run, respectively. To make sense of the changes in precision, we can compare these data to their center point or mean μ value, which makes a good yardstick. It is easy to find that $\mu = 5.5925\text{E}{-05}$. We can apply the standard range for a simple precision estimation. The range is the simplest way to estimate precision. To estimate it, we simply find the highest value and subtract the lowest value from it. In this manner, the estimated range is just $(5.5925\text{E}{-05} - 5.5925\text{E}{-05}) = 0$. This verifies that all geometric means were fitted within $5.5925\text{E}{-05} \pm 0$, where ± 0 is the precision of Pyramid-LK algorithm.[51–53] It shows that this algorithm is only precise within a $0 + 0 = 0$ range (i.e., $+0$ higher and -0 lower). This simple technique is used to estimate precision frequently but ironically it is not very precise always. To estimate precision more accurately, it is important to estimate either mean absolute deviation (MAD) or standard deviation σ. MAD is using L^1 norm (it is also known as rectilinear distance or city block distance or Manhattan distance or Manhattan length), whereas σ is using L^2 norm (it is also called Euclidean distance). The distinction between L^1 and L^2 norms is that L^1 norm is exclusively looking at the absolute difference, whereas L^2 norm is calculating the square of the difference. Consequently, large outliers can produce a higher dispersion when applying σ instead of MAD. But σ is most frequently used. The most important reason is that σ has good properties when the data are normally distributed. The central limit theorem says that as the number of variables in a sum increases the distribution of the sum of random variables approaches to the normal distribution irrespective of the shape of the distribution of the individual random variables. So under this assumption, it is advised to use σ. If data are not normally distributed, we can still apply σ but the results should be carefully interpreted. To whatever extent, without question we can calculate both measures of dispersion as MAD $= 0$ and $\sigma = \pm 0$ from the aforementioned data. These results express briefly that δ_r value of a frame in any time run of the Pyramid-LK algorithm[51–53] will be having zero difference from the δ_r value of that frame obtained from any other time run of the algorithm. At this point, the precision of the Pyramid-LK algorithm[51–53] can be known by asking a simple question as: Are we getting similar output always from the same frames, while we are measuring the precision of this algorithm? Yes, we came into possession of not only the similar output but also exactly the identical output every time, i.e., MAD $= 0$ or $\sigma = \pm 0$, indicating the highest precision of the Pyramid-LK algorithm.[51–53] If we would have obtained high values of σ or MAD, then it would signify that the Pyramid-LK algorithm[51–53] was not functioning efficiently, probably due to some error that existed during experimentation or calculation.

Hitherto, we have checked the precision of Pyramid-LK algorithm,[51–53] i.e., how well the Pyramid-LK algorithm[51–53] outputs the same data over repeated runs. Now we wish to check the accuracy of our experimental datasets, i.e., how close the output data of Pyramid-LK algorithm[51–53] are to the ground truths. To do so, we need optical flow ground truths. To obtain any optical flow ground truth from V_1 of PETS2009[49] is a costly option. Easy way is to find an optical flow benchmark dataset with given ground truths, then experimental data can be produced by using that dataset. And then we can compare the experimental data with the given ground truths to know the accuracy of the experimental data. Fortunately, Baker *et al.*[57] presented a collection of optical flow datasets with ground truths, called Middlebury optical flow benchmark datasets and ground truths, to evaluate the accuracy of experimental data and interpolation accuracy of the existing optical flow algorithms. The online evaluation available (at the time of writing this paper) on the website (http://vision.middlebury.edu/flow/) provides a snapshot of the state of the art in optical flow algorithms. By any chance, Baker *et al.*[57] already checked the accuracy of the experimental data of Pyramid-LK algorithm,[51–53] i.e., how right the experimental data of Pyramid-LK algorithm[51–53] are. Using same ground truths, they also showed accuracies of others experimental data. Considering Middlebury optical flow benchmark datasets and ground truths, Fig. 23 shows the evaluation results of the experimental data of 20 state-of-the-art optical flow algorithms.[53,58–76] The runtime in seconds was estimated using Urban dataset but it was not normalized by CPU speed or type. The eight sequences with ground truths from datasets of Army,

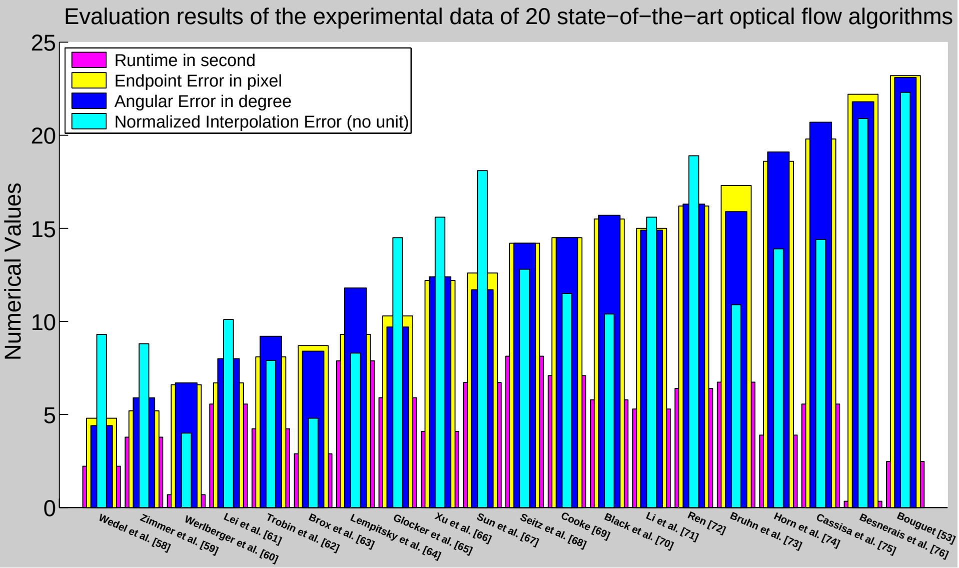


Fig. 23. Runtimes, experimental data accuracies, and interpolation accuracies of 20 state-of-the-art optical flow algorithms considering the Middlebury optical flow benchmark datasets and ground truths. Logarithmic scale has been used for runtime to visualize several orders of magnitude.

Mequon, Schefflera, Wooden, Grove, Urban, Yosemite, and Teddy were used to estimate endpoint error and angular error, which measure the accuracy of experimental data. The datasets of Mequon, Schefflera, Urban, Teddy, Backyard, Basketball, Dumptruck, and Evergreen were used to estimate interpolation error, which measures interpolation accuracy.[57] The most commonly used measure of accuracy for experimental data is the angular error. The goal of the angular error is to provide a relative measure of performance that avoids the divide by zero problem for zero flows. Errors in large flows are penalized less in angular error than errors in small flows. Although the use of angular error is common, the endpoint error measure is probably more appropriate for most applications.[57] In whatever way, the average values over all statistical angular errors and endpoint errors of the experimental data of Pyramid-LK algorithm[51–53] were reported as 23.1 degrees and 23.2 pixels, respectively.[57] This hints that our used experimental datasets are also subject to such type of average errors. As we have observed, the precision of Pyramid-LK algorithm[51–53] is 100% but the accuracy of our used experimental datasets is less than 100%. Another well-known optical flow algorithm is Horn–Schunck.[74] A modern MATLAB implementation of the Horn–Schunck method[74] by Sun *et al.*[77] had encountered the average values over all statistical angular errors and endpoint errors as 19.1 degrees and 18.6 pixels, respectively.[57] Yet the implementation of Pyramid-LK algorithm[51–53] is about four times faster than that of the Horn–Schunck algorithm.[57]

Based on the applications, we can choose an optical flow algorithm to produce reliable experimental datasets by compromising either runtime or error.

3.9. *Future works*

Our method deals with multiple separate instances of single flow occurring in videos with temporal overlap. For example, considering V_3 in Fig. 12, a person entered into a scene and started walking from east to north, i.e., single flow was detected. A bit while other six persons entered into the scene one by one and they started walking in parallel from east to north, etc. There existed multiple separate instances of single flow occurring at the same time (i.e., up to five parallel flows). But our method detected only the existence of multiple flows. It did not give us information about five separate instances of single flow. It did not give us any information whether those separate instances of single flow were opposing flows or parallel flows. It did not tell the locations of the existing parallel flows too. Conversely, it would be interested to know how the algorithm will deal with multiple separate instances of single flow occurring in the video with no temporal overlap. For example, a crowd of people walk from north-west to south-east, followed by a crowd walking from south to north. The flows observed in these two scenarios will be very different and will require different threshold values. How would the algorithm decide that it needs to reevaluate the threshold value? Future study will consider these issues. By way of addition, our current implementation is not working in real-time. For coding both C/C++ and MATLAB environments were used. Although C/C++ part worked in real-time, MATLAB portion was evaluated on offline fashion. This could be another future issue to bring about the algorithm in online mode. Since number of frames execution per unit time depends on computer architectures (e.g., number of CPUs, number of issued instructions by CPU,[78,79] and available memories[80]), theoretical time complexity (e.g., Big O notation[81]) of the algorithm would be analyzed.

4. Conclusion

We presented a wide overview of the analysis of flows in real-world crowd videos along with our own method. In our method, spatiotemporal features of two successive frames were extracted by optical flows to construct a 3D tensor, which grasped appearance and velocity information. Tensor's upper left minor matrix concerned about the intensity structure. A normalized continuous rank-increase measure δ_r for each frame was calculated by applying a generalized interlacing property of the eigenvalues of these matrices. But δ_r values did not work desirably due to optical flow estimation error and some other factors. An appropriate set of the degree of polynomial fitting functions was interposed to make visible the existing flows. Various flows and interesting events were recognized as frame basis using thresholds on the polynomial fitting δ_r values. The convincing results got from conducted experiments on 24 videos of two benchmark datasets reported that our method owned high-quality outputs to detect flows and events in terms of both robustness and efficacy. This is because of the following: (i) Eigenvalue analysis of spatiotemporal features is very robust to exploit appearance and interaction of crowd activities; and (ii) a correct degree of polynomial fitting function on δ_r values magnifies the performance extensively over unwanted feature points. Yet the method did not locate and distinguish between various types of multiple flows. Future study would focus on these issues along with some smarter threshold estimation process.

Acknowledgments

Author acknowledges the anonymous reviewers for their appreciative and constructive comments on the draft of this paper.

References

1. B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin and L. Q. Xu, Crowd analysis: A survey, *Mach. Vis. Appl.* **19** (2008) 345–357.
2. J. C. S. J. Junior, S. R. Musse and C. R. Jung, Crowd analysis using computer vision techniques, *IEEE Signal Process. Mag.* **27** (2010) 66–77.
3. M. Thida, Y. L. Yong, P. C. Pérez, H. Eng and P. Remagnino, A literature review on video analytics of crowded scenes, *Intelligent Multimedia Surveillance: Current Trends and Research*, eds. P. K. Atrey, M. S. Kankanhalli and A. Cavallaro (Springer, Berlin, 2013), pp. 17–36.
4. C. C. Loy, K. Chen, S. Gong and T. Xiang, Crowd counting and profiling: Methodology and evaluation, *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective*, eds. S. Ali, K. Nishino, D. Manocha and M. Shah (Springer, New York, 2013), pp. 347–382.
5. T. Li, H. Chang, M. Wang, B. Ni, R. Hong and S. Yan, Crowded scene analysis: A survey, *IEEE Trans. Circuits Syst. Video Technol.* **25** (2015) 367–386.
6. V. J. Kok, M. K. Lim and C. S. Chan, Crowd behavior analysis: A review where physics meets biology, *Neurocomputing* **177** (2016) 342–362.
7. M. S. Zitouni, H. Bhaskar, J. M. M. Dias and M. E. Al-Mualla, Advances and trends in visual crowd analysis: A systematic survey and evaluation of crowd modelling techniques, *Neurocomputing* **186** (2016) 139–159.
8. A. Kläser, M. Marszalek and C. Schmid, A spatio-temporal descriptor based on 3D-gradients, *Proc. British Machine Vision Conf. (BMVC)*, Leeds, UK (2008), pp. 1–10.
9. I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, Learning realistic human actions from movies, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Alaska, USA (2008), pp. 1–8.
10. B. Zhou, X. Wang and X. Tang, Random field topic model for semantic region analysis in crowded scenes from tracklets, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Colorado, USA (2011), pp. 3441–3448.
11. H. Yang, Y. Cao, S. Wu, W. Lin, S. Zheng and Z. Yu, Abnormal crowd behavior detection based on local pressure model, *Proc. Asia-Pacific Signal and Information Processing Association Annu. Summit and Conf. (APSIPA ASC)*, California, USA (2012), pp. 1–4.

12. W. Lin, H. Chu, J. Wu, B. Sheng and Z. Chen, A heat-map-based algorithm for recognizing group activities in videos, *IEEE Trans. Circuits Syst. Video Technol.* **23** (2013) 1980–1992.
13. H. Su, H. Yang, S. Zheng, Y. Fan and S. Wei, The large-scale crowd behavior perception based on spatio-temporal viscous fluid field, *IEEE Trans. Inf. Forensics Sec.* **8** (2013) 1575–1589.
14. H. Fradi and J. L. Dugelay, Sparse feature tracking for crowd change detection and event recognition, *Proc. Int. Conf. Pattern Recognition (ICPR)*, Stockholm, Sweden (2014), pp. 4116–4121.
15. K. W. Cheng, Y. T. Chen and W. H. Fang, Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Massachusetts, USA (2015), pp. 2909–2917.
16. UMN, Unusual crowd activity dataset of University of Minnesota (2006), http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi.
17. N. Ihaddadene and C. Djeraba, Real-time crowd motion analysis, *Proc. Int. Conf. Pattern Recognition (ICPR)*, Florida, USA (2008), pp. 1–4.
18. L. Kratz and K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Florida, USA (2009), pp. 1446–1453.
19. M. H. Sharif and C. Djeraba, Exceptional motion frames detection by means of spatio-temporal region of interest features, *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Cairo, Egypt (2009), pp. 981–984.
20. R. Mehran, A. Oyama and M. Shah, Abnormal crowd behavior detection using social force model, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Florida, USA (2009), pp. 935–942.
21. S. Wu, B. E. Moore and M. Shah, Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, California, USA (2010), pp. 2054–2060.
22. R. Mehran, B. E. Moore and M. Shah, A streakline representation of flow in crowded scenes, *Proc. European Conf. Computer Vision (ECCV)*, Crete, Greece (2010), pp. 439–452.
23. X. Cui, Q. Liu, M. Gao and D. N. Metaxas, Abnormal detection using interaction energy potentials, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Colorado, USA (2011), pp. 3161–3167.
24. V. Saligrama and Z. Chen, Video anomaly detection based on local statistical aggregates, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Rhode Island, USA (2012), pp. 2112–2119.
25. M. H. Sharif and C. Djeraba, An entropy approach for abnormal activities detection in video streams, *Pattern Recognit.* **45** (2012) 2543–2561.
26. W. Y. Ren, G. H. Li, J. Chen and H. Z. Liang, Abnormal crowd behavior detection using behavior entropy model, *Proc. Int. Conf. Wavelet Analysis and Pattern Recognition (ICWAPR)*, Shaanxi, China (2012), pp. 212–221.
27. X. Tang, S. Zhang and H. Yao, Sparse coding based motion attention for abnormal event detection, *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Melbourne, Australia (2013), pp. 3602–3606.
28. T. Wang, J. Chen and H. Snoussi, Online detection of abnormal events in video streams, *J. Electr. Comput. Eng.* **2013** (2013) 837275.

29. M. Thida, H. L. Eng and P. Remagnino, Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes, *IEEE Trans. Cybern.* **43** (2013) 2147–2156.
30. Y. Cong, J. Yuan and J. Liu, Abnormal event detection in crowded scenes using sparse representation, *Pattern Recognit.* **46** (2013) 1851–1864.
31. X. Hu, S. Hu, X. Zhang, H. Zhang and L. Luo, Anomaly detection based on local nearest neighbor distance descriptor in crowded scenes, *Sci. World J.* **2014** (2014) 632575.
32. A. Li, Z. Miao, Y. Cen, T. Wang and V. V. Voronin, Histogram of maximal optical flow projection for abnormal events detection in crowded scenes, *Int. J. Distrib. Sens. Netw.* **2015** (2015) 406941.
33. T. Wang and H. Snoussi, Detection of abnormal events via optical flow feature analysis, *Sensors (Basel)* **15** (2015) 7156–7171.
34. Y. Zhang, H. Lu, L. Zhang and X. Ruan, Combining motion and appearance cues for anomaly detection, *Pattern Recognit.* **51** (2016) 443–452.
35. Y. Zhang, H. Lu, L. Zhang, X. Ruan and S. Sakai, Video anomaly detection based on locality sensitive hashing filters, *Pattern Recognit.* **59** (2016) 302–311.
36. Y. Shi, Y. Gao and R. Wang, Real-time abnormal event detection in complicated scenes, *Proc. Int. Conf. Pattern Recognition (ICPR)*, Istanbul, Turkey (2010), pp. 3653–3656.
37. A. Briassouli and I. Kompatsiaris, Spatiotemporally localized new event detection in crowds, *Proc. Int. Conf. Computer Vision (ICCV) Workshops*, Barcelona, Spain (2011), pp. 928–933.
38. W. Li, V. Mahadevan and N. Vasconcelos, Anomaly detection and localization in crowded scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* **36** (2014) 18–32.
39. X. Wang, X. Ma and W. E. L. Grimson, Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models, *IEEE Trans. Pattern Anal. Mach. Intell.* **31** (2009) 539–555.
40. N. Li and Z. Zhang, Abnormal crowd behavior detection using topological methods, *Proc. ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking, and Parallel/ Distributed Computing (SNPD)*, Sydney, Australia (2011), pp. 13–18.
41. Y. Ito, K. M. Kitani, J. A. Bagnell and M. Hebert, Detecting interesting events using unsupervised density ratio estimation, *Proc. Int. Workshop Analysis and Retrieval of Tracked Events and Motion in Imagery Streams at ECCV*, Florence, Italy (2012), pp. 151–161.
42. I. R. Almeida and C. R. Jung, Change detection in human crowds, *Proc. Conf. Graphics, Patterns and Images (SIBGRAPI)*, Arequipa, Peru (2013), pp. 63–69.
43. X. Zhu, J. Liu, J. Wang, C. Li and H. Lu, Sparse representation for robust abnormality detection in crowded scenes, *Pattern Recognit.* **47** (2014) 1791–1799.
44. Y. Liu, Y. Li and X. Ji, Abnormal event detection in nature settings, *Int. J. Signal Process. Image Process. Pattern Recognit.* **7** (2014) 115–126.
45. S. Xie and Y. Guan, Motion instability based unsupervised online abnormal behaviors detection, *Multimedia Tools Appl.* **75** (2016) 7423–7444.
46. G. Mariem, E. Ridha and Z. Mourad, Detection of abnormal movements of a crowd in a video scene, *Int. J. Comput. Theory Eng.* **8** (2016) 398–402.
47. S. Susan and M. Hanmandlu, Unsupervised detection of nonlinearity in motion using weighted average of non-extensive entropies, *Signal Image Video Process.* **9** (2015) 511–525.
48. Y. H. Kim and A. C. Kak, Error analysis of robust optical flow estimation by least median of squares methods for the varying illumination model, *IEEE Trans. Pattern. Anal. Mach. Intell.* **28** (2006) 1418–1435.

49. PETS, Benchmark crowd dataset of performance evaluation of tracking and surveillance (PETS) (2009), http://www.cvg.reading.ac.uk/PETS2009/a.html [http://www.cvg.reading.ac.uk/PETS2012/a.html].
50. UCSDped1, Anomaly detection dataset introduced by University of California, San Diego (2013), http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm.
51. B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, Vancouver, Canada (1981), pp. 674–679.
52. J. Shi and C. Tomasi, Good features to track, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA (1994), pp. 593–600.
53. J. Y. Bouguet, Pyramidal implementation of the Lucas–Kanade feature traker description of the algorithm, A part of OpenCV Documentation, Intel Corporation, Microprocessor Research Labs (2000).
54. C. Runge, Über empirische Funktionen und die interpolation zwischen äquidistanten Ordinaten, *Z. Math. Phys.* **46** (1901) 224–243.
55. P. L. Chebyshev, Théorie des mécanismes connus sous le nom de parallélogrammes, *Mém. Acad. Sci. St. Pétersb.* **7** (1854) 539–568.
56. Q. I. Rahman and G. Schmeisser, Characterization of the speed of convergence of the trapezoidal rule, *Numer. Math.* **57** (1990) 123–138.
57. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black and R. Szeliski, A database and evaluation methodology for optical flow, *Int. J. Comput. Vis.* **92** (2011) 1–31.
58. A. Wedel, D. Cremers, T. Pock and H. Bischof, Structure- and motion-adaptive regularization for high accuracy optic flow, *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Kyoto, Japan (2009), pp. 1663–1668.
59. H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn and H. P. Seidel, Complementary optic flow, *Proc. Int. Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, Bonn, Germany (2009), pp. 207–220.
60. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers and H. Bischof, Anisotropic Huber-L1 optical flow, *Proc. British Machine Vision Conf. (BMVC)*, London, UK (2009), pp. 1–11.
61. C. Lei and Y. H. Yang, Optical flow estimation on coarse-to-fine region-trees using discrete optimization, *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Kyoto, Japan (2009), pp. 1562–1569.
62. W. Trobin, T. Pock, D. Cremers and H. Bischof, Continuous energy minimization via repeated binary fusion, *Proc. European Conf. Computer Vision (ECCV)*, Marseille, France (2008), pp. 677–690.
63. T. Brox, A. Bruhn, N. Papenberg and J. Weickert, High accuracy optical flow estimation based on a theory for warping, *Proc. European Conf. Computer Vision (ECCV)*, Prague, Czech Republic (2004), pp. 25–36.
64. V. Lempitsky, S. Roth and C. Rother, Fusionflow: Discrete-continuous optimization for optical flow estimation, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Alaska, USA (2008), pp. 1–8.
65. B. Glocker, N. Paragios, N. Komodakis, G. Tziritas and N. Navab, Optical flow estimation with uncertainties through dynamic MRFs, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Alaska, USA (2008), pp. 1–8.
66. L. Xu, J. Chen and J. Jia, A segmentation based variational model for accurate optical flow estimation, *Proc. European Conf. Computer Vision (ECCV)*, Marseille, France (2008), pp. 671–684.

67. D. Sun, S. Roth, J. P. Lewis and M. J. Black, Learning optical flow, *Proc. European Conf. Computer Vision (ECCV)*, Marseille, France (2008), pp. 83–97.
68. S. M. Seitz and S. Baker, Filter flow, *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Kyoto, Japan (2009), pp. 143–150.
69. T. Cooke, Two applications of graph-cuts to image processing, *Proc. Int. Conf. Digital Image Computing: Techniques and Applications (DICTA)*, Canberra, Australia (2008), pp. 498–504.
70. M. J. Black and P. Anandan, The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields, *Comput. Vis. Image Underst.* **63** (1996) 75–104.
71. Y. Li and D. P. Huttenlocher, Learning for optical flow using stochastic optimization, *Proc. European Conf. Computer Vision (ECCV)*, Marseille, France (2008), pp. 379–391.
72. X. Ren, Local grouping for optical flow, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Alaska, USA (2008), pp. 1–8.
73. A. Bruhn, J. Weickert and C. Schnörr, Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods, *Int. J. Comput. Vis.* **61** (2005) 211–231.
74. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.* **17** (1981) 185–203.
75. C. Cassisa, S. Simoens and V. Prinet, Two-frame optical flow formulation in an unwarping multiresolution scheme, *Proc. Iberoamerican Conf. Pattern Recognition (CIARP)*, Guadalajara, Mexico (2009), pp. 790–797.
76. G. L. Besnerais and F. Champagnat, Dense optical flow by iterative local window registration, *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Genoa, Italy (2005), pp. 137–140.
77. D. Sun, S. Roth and M. J. Black, Secrets of optical flow estimation and their principles, *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA (2010), pp. 2432–2439.
78. M. H. Sharif, High performance computing of $1/\sqrt{x}$ and $e^{\pm x}$ in Alpha, Intel, and Opteron processors, Master's thesis, University of Duisburg-Essen, Duisburg (2006), http://hdl.handle.net/11858/00-001M-0000-0015-5BCE-4.
79. M. H. Sharif, A. Basermann, C. Seidel and A. Hunger, High-performance computing of $1/\sqrt{x_i}$ and $\exp(\pm x_i)$ for a vector of inputs x_i on Alpha and IA-64, *J. Syst. Archit.* **54** (2008) 638–650.
80. M. H. Sharif, High-performance mathematical functions for single-core architectures, *J. Circuits Syst. Comput.* **23** (2014) 1450051.
81. M. H. Sharif, A numerical approach for tracking unknown number of individual targets in videos, *Digit. Signal Process.* **57** (2016) 106–127.