An Improved Approach to Crowd Event Detection by Reducing Data Dimensions

Aravinda S. Rao, Jayavardhana Gubbi and Marimuthu Palaniswami

Abstract Crowd monitoring is a critical application in video surveillance. Crowd events such as running, walking, merging, splitting, dispersion, and evacuation inform crowd management about the behavior of groups of people. For an effective crowd management, detection of crowd events provides an early sign of the behavior of the people. However, crowd event detection using videos is a highly challenging task because of several challenges such as non-rigid human body motions, occlusions, unavailability of distinguishing features due to occlusions, unpredictability in people movements, and other. In addition, the video itself is a high-dimensional data and analyzing to detect events becomes further complicated. One way of tackling the huge volume of video data is to represent a video using low-dimensional equivalent. However, reducing the video data size needs to consider the complex data structure and events embedded in a video. To this extent, we focus on detection of crowd events using the Isometric Mapping (ISOMAP) and Support Vector Machine (SVM). The ISOMAP is used to construct the low-dimensional representation of the feature vectors, and then an SVM is used for training and classification. The proposed approach uses Haar wavelets to extract Gray Level Coefficient Matrix (GLCM). Later, the approach extracts four statistical features (contrast, correlating, energy, and homogeneity) at different levels of Haar wavelet decomposition. Experiment results suggest that the proposed approach is shown to perform better when compared with existing approaches.

Aravinda S. Rao,
The University of Melbourne, Department of Electrical and Electronic Engineering, Parkville Campus, VIC - 3010, Australia. e-mail: aravinda@student.unimelb.edu.au

Jayavardhana Gubbi,
TCS Innovation Labs, Tata Consultancy Services, Bengaluru, Karnataka - 560066, India. e-mail: j.gubbi@tcs.com

Marimuthu Palaniswami
The University of Melbourne, Department of Electrical and Electronic Engineering, Parkville Campus, VIC - 3010, Australia. e-mail: palani@unimelb.edu.au

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-28658-7_8

1 Introduction

Crowd behavior analysis of video surveillance applications has gained significant interests in computer vision research. Analyzing crowd behavior requires several levels of video processing: low- and high-level [11, 8]. The low-level processing involves extracting features at pixel levels and high-level semantics are often derived by combining other features relevant to crowd behavior. Analyzing individual movements is important for applications where tracking of an individual in required. However, analyzing crowd requires a holistic view of the scene. In the case of crowds, multi-person interaction must be considered and this becomes a challenging task because of undefined motion patterns involved in crowd movements. The human visual system's advancements mask the difficulty and challenges for humans in finding the motion patterns. In contrast, crowd video analytics is still under development to reaching the stage of being fully automated.

The need for automated crowd behavior analysis stems from the fact that the crowd monitoring applications are highly important. Applications of crowd monitoring include counting, density estimation, detecting crowd events (such as walking, running, etc.), and understanding crowd behavior. These applications often find their importance in crowd monitoring at public places, such as public transport hubs, airports, subway stations, and other. The fundamental steps for any crowd analytics algorithms include preprocessing the video frames, object detection and tracking. However, the challenges of object detection and tracking include [33]: (1) unavailability of depth information, (2) unaccounted video noise generated during the formation of video frames, (3) articulated motions of the moving objects, (4) occluded scenarios, and (5) illumination variations. Compared with computer vision algorithms developed for detecting rigid objects, crowd analytics suffers severely because of articulated human motions and occlusions.

The crowd behavior analysis provides a holistic view of crowd under surveillance. Several features are aggregated at different levels of processing and behavior is inferred. One of the indicators used in behavior are the crowd events. Crowd events consist of interaction of people and their activities [16] and one of the main end-user application for officials concerned with crowd management [23]. Analyzing crowd motion pattern is one of highly challenging tasks in computer vision. To the best our knowledge, the literature consists of six crowd events defined using the Performance Evaluation of Tracking and Surveillance (PETS) 2009 dataset [10]: walking, running, merging, splitting, dispersion, and evacuation. Fig. 1 shows the six crowd events from PETS 2009 dataset. The objective of event detection at a broader level is to analyze the video and detect events such that frames with similar events are clustered together. However, object detection itself has been a research challenge for many years. Although several detection methods have been proposed in the literature, crowd event detection has received less attention. This is primarily due to the complexities involved in detection humans and the crowd.

Video frames are a source of high-dimensional data. Let x and y denote position of a pixel in a video frame I. Then a pixel location can be denoted as $I(x,y)$. Let $m \in \mathbb{R}$ and $n \in \mathbb{R}$ denote the number of rows and columns of the video frame $I(x,y)$.

(a) (b) (c) (d) (e) (f)

Fig. 1: Examples of crowd events from the PETS 2009 dataset [10]: (a) walking, (b) running, (c) merging, (d) splitting, (e) dispersion, and (f) evacuation.

Let $(r,g,b) \in \mathbb{R}^3$ denote the three-tuple notation for red, green and blue channels of the camera. Then, the dimensions of a video frame is $\mathbb{R}^{m \times n \times 3}$. When m and n increase considerably, the processing of such high-dimensional frames and later performing the analytics (counting, detecting events) coupled with complex human motions becomes extremely difficult. Dimensionality reduction is a much sought-after approach to reduce video dimensions without losing much of the pertinent data. The core of dimensionality reduction is to represent the high dimensional data in a low-dimensional form. The high-dimensional features are represented in a low-dimensional format such that they can be used to distinguish objects and track them. These features are then used for applications such as counting, density estimation, tracking, detecting crowd events and behavior analysis.

In this work, we focus on detection of crowd events using the Isometric Mapping (ISOMAP) [28] and Support Vector Machine (SVM). In particular, we use the PETS 2009 dataset [10] and extract the Gray Level Coefficient Matrix (GLCM) using the Haar wavelet [14] with up to eight levels of decomposition. With this, the feature vectors span a 100-dimensional features space. The approach uses ISOMAP to find the low-dimensional representation of the feature vectors by constructing a graph distance matrix with feature vectors as nodes of the graph. The approach learns the mapping from the low-dimensional space in the event classes using the SVM. The experiments were conducted for both linear and Radial Basis Function (RBF) kernels. These two different dimensional data were trained and classified using both linear and RBF kernels. The output from the ISOMAP were mapped to one-dimensional and ten-dimensional embedded feature space.

2 Related Work

2.1 Dimensionality Reduction Methods

There are two main types of dimensionality reduction methods: (1) linear and (2) nonlinear. Linear methods assume the feature subspace to be linear, i.e., the feature vectors satisfy the linearity property of the subspace. Principal Component Analysis (PCA) [15] is a good example of linear dimensionality reduction. PCA assumes the data vectors to be lying in a linear feature space. PCA endeavors to maximize the

global variance while discarding the order of the vectors in the put feature space. In addition, the relationship among feature vectors are ignored

On the other hand, the nonlinear methods approximate the global space by locally augmenting the linear subspaces. These algorithms are also called as manifold learning algorithms. They endeavor to find the embedded (lower) dimensions of the high-dimensional features. Manifold is intuitively thought to be a point in some topological space that can be reached without ambiguity [18, 24]. The three common steps involved in finding the low-dimensional feature vectors by nonlinear methods are [20]: (1) construct a weighted graph with nodes denoted by feature input vectors and connections using the neighborhood information, (2) convert the weighted graph to a form suitable to find the low-dimensional embedding, i.e., find the graph distances, and (3) solve the eigenfunction ("spectral embedding") to obtain a set of low-dimensional embedding vectors.

The ISOMAP [28] is an example of nonlinear dimensionality reduction algorithm. Classical Multidimensional Scaling (MDS) [31] aims at maintaining interpoint distances from high-dimensional input to low-dimensional space. The interpoint distances represent objects and MDS calculates the proximity matrix based on the Euclidean distances between points. ISOMAP [28] uses classical MDS that attempts to identify a low-dimensional subspace while preserving the isometry of the input data points. ISOMAP assumes that the points are invariant under transformation and geodesic distances are used. The ISOMAP employs a *k*-Nearest Neighbor ($k-$NN) approach, followed by MDS to construct the graph distances and then apply eigendecomposition. This method is a global approach to finding the low-dimensional embedding and the other manifold learning methods approach the dimensionality reduction from a local observer perspective. Work presented in [28] showed the effectiveness of the ISOMAP to applications, such as handwriting recognition and head-pose estimation. Locally Linear Embedding (LLE) [25] assumes the local geometry of data points to be linear coefficients such that the patches are reconstructed by its neighbors, where it employs a *k*-Nearest Neighbor ($k-$NN) approach.

Souvenir and Pless [26, 27, 22] proposed image distances based on manifold learning that are similar to ISOMAP. The authors contributed to highlight the natural parameterization space of image manifolds. Guo *et al.* [13] proposed a method to learn the "age manifold" from the set of training images. The learned manifold would serve as model to estimate and predict the age of the people from images. Chang *et al.* [5] proposed a new method to model, track and recognize facial expressions using a low-dimensional manifold. Instead of learning a manifold from images, the authors use the facial contours.

2.2 Crowd Event Detection

Chen *et al.* [6] included Haar features related to head and then tracked the objects in the scene. The objects were then treated as agents and information such as direction and speed was derived. Objects were tracked using template matching and

Kalman filtering. Feature vectors were constructed based on agents' movements such as walking, running, jumping and stopping. An SVM was trained using known samples to recognize the actions from a local dataset. Ke *et al.* [17] proposed an approach to recognize events in crowd videos by identifying actions. Event detection was achieved by matching the shapes in spatio-temporal volumes. First, the shape contours are extracted from spatial-temporal patches and next, these shapes were matched in spatial-temporal patches using optical flow features. As a final step in recognizing the events, part-matching of shape templates was performed: instead of matching the whole templates, Ke *et al.* [17] allowed parts to be matched independently.

Gárate *et al.* [12] proposed to use 2D Histogram of Gradients (HOG) descriptors as features. In particular, they utilized the features around a block of nine cells (3×3). They computed gradient vector magnitude and orientation for all the pixels within the block. Then, the orientations were binned and 2D feature vectors were constructed. These feature vectors were later tracked in the next frame using the object speed and a time window. Finally, the crowd events were recognized based on the tracked features over frames.

Li *et al.* [19] focused their work on multi-object activities to characterize the group motion. In this aspect, they proposed a data-driven Discriminative Temporal Interaction Manifold (DTIM) framework. The framework established probability densities on the DTIM based on the interactions between the objects. They used discriminative temporal interaction matrix to arrive at the probability densities. The method was tested on a soccer game video. Benabbas *et al.* [2] approached the problem of crowd event detection by building a direction and a magnitude model using the optical flow motion vectors. Circular clustering was performed to learn the prominent directions. The motion vectors were refined using the online Gaussian Mixture Model (GMM). The magnitude and direction patterns were used to track the objects in the neighborhood. They were then clustered and tracked to detect the crowd events.

Thida *et al.* [29] used Histogram of Optical Flow (HOOF) as features to detect crowd events. They divided each video frame into blocks and extracted the HOOF features. These features were later supplied to LE [1] to find a low-dimensional representation of the features. Using a similarity measure, the crowd events were represented in a low-dimensional representation.

3 Methodology

3.1 Preprocessing and Feature Extraction

Raw frames from the cameras contain high-frequency noise and would contribute to generating errors when information is passed from low-level processing to semantic-level decisions. In addition, low-frequency image signals contain most information

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-28658-7_8

and this can be verified by performing a wavelet decomposition of an image. Therefore, in the first step, a Gaussian low-pass filter of size 7×7 is used to smoothen the images spatially. Next, the bilateral filter [30] is applied to preserve the edges. The literature provides features such as edge, texture, color, motion (optical flow), shapes, and other. In this work, feature extraction included the Haar features [14] through Haar Wavelet Transform (HWT), where the video frames were analyzed at multiple resolutions using the Haar functions. Previous research on people detection (e.g. [7, 9]) provide extensive evidence to use the Haar wavelet as crowd features in this work.

3.2 Haar Wavelet Transform

The Haar piecewise constant function (wavelet) is given by [14, 21]

$$\Psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1/2 \\ -1 & \text{if } 1/2 \leq t < 1 \\ 0 & \text{otherwise} , \end{cases} \tag{1}$$

which generates an orthonormal basis on dilations (scalings) and translations (shiftings) given by([14, 21])

$$\Psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \Psi\Big(\frac{t - 2^j n}{2^j}\Big), \tag{2}$$

where $(j,n) \in \mathbb{Z}^2$. The orthonormal basis also forms the basis of a space of finite energy signals given by

$$\|f\|^2 = \int_{-\infty}^{\infty} |f(t)|^2 \, dt \ \ < \infty, \tag{3}$$

which can be represented in the inner product form as

$$\langle f, \Psi_{j,n} \rangle = \int_{-\infty}^{\infty} f(t) \Psi_{j,n} \, dt, \tag{4}$$

where $\langle f, \Psi_{j,n} \rangle$ are called the Haar coefficients. In the case of images, the Haar coefficients correspond to different levels of decomposition. Using the Haar Wavelet, each frame was decomposed into eight levels. For each level, horizontal, vertical and diagonal components were computed. Furthermore, for each of the three components, a GLCM was calculated. The four statistical measures—contrast, homogeneity, energy and correlation—were extracted from the GLCM. In addition, the four statistical measures of the approximation coefficient were extracted. Therefore, for each frame, the feature vector length would be $N_D \times N_S \times N + N_A \times N_S = 3 \times 4 \times 8 + 1 \times 4 = 100$, where N is the number of levels of decomposition, N_D is the

number of detailed coefficients for each level, N_A is the number of approximation coefficient of the N^{th} level and N_S is the statistical measures for each of N_D or N_A. A high dimensional feature matrix ($\mathbb{R}^{m\times n}$) was constructed using the texture features, where m represents the number of features (texture) for each frame and n denotes the frames.

3.3 Dimensionality Reduction and Classification

The general dimensionality reduction algorithm can be described as follows: given a set (data) $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ with $\mathbf{x}_i \in \mathbb{R}^m$, find a set $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ with $\mathbf{y}_i \in \mathbb{R}^d$ that represents X such that $d \ll m$ [1]. Most of the dimensionality reduction methods are proposed based on meeting certain objectives. Hence, most the methods employ a certain form of optimization. ISOMAP [28] uses three steps to complete feature mappings: (1) determine neighbors based on the distance $d(i, j)$ between points (i, j) in the input space ($X \in \mathbb{R}^m$) are represented as a weighted graph. (2) geodesic distance between all the points on the manifold are computed using the shortest path over weighted graph, and (3) apply classical MDS to the graph distance matrix to construct a d-dimensional embedding from the m-dimensional input space X, where $d \ll m$. The neighborhood can be in the first step can be either fixed ε-neighborhood or k-NN.

Support Vector Machine (SVM) is a supervised learning algorithm used for binary classification. The idea behind the SVM is to find the hyperplane such that the distance between two classes separated from the closest points to the hyperplane is maximized and is given by

$$\min_{\omega,b,\xi} \frac{1}{2}\|\omega\|^2 + \frac{C}{n}\sum_{i=1}^{n} l(\xi^k), \tag{5}$$

subject to $y_i(\omega \cdot x_i + b) \geq 1 - \xi_i, \forall \quad i \in 1, \ldots, n$, where C is a positive regularization constant and ξ is the slack term. The points that lie on the boundaries are called as *support vectors*. In cases where the data are not linearly separable using the hyperplane, the kernel techniques are used to project the data points to high-dimensional space so that the data can be separated. In our approach, the output of the ISOMAP algorithms were used to train the SVM. Both linear and nonlinear approaches were used. For nonlinear approach, the RBF kernel was used to project the data to higher dimensions and is given by

$$K(x, x') = e^{\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)}, \tag{6}$$

where $K(\cdot, \cdot)$ is the RBF kernel, x and x' are feature samples in the input space, and a free parameter σ. The positive regularization constant C and the kernel parameter σ were learned using the grid search optimization algorithm [4].

4 Results and Discussion

The proposed method was implemented in MATLAB on a Windows 7 machine (64-bit) equipped with 4 GB RAM and Intel® i7 – 2600 CPU running at 3.4 GHz.

4.1 Dataset

The PETS 2009 [10] dataset was used to evaluate the proposed method. The crowd events are categorized under the PETS 2009 [10] dataset (S3) with four different timings ($14-16, 14-27, 14-31$ and $14-33$). The timings refer to the hour-minute format. For each timing, there are four different views $(001, 002, 003,$ and $004)$. To the best of our knowledge, only the PETS 2009 [10] dataset has the events where all six crowd events can be clearly evaluated based on human motion analysis. The proposed approach was evaluated on a total of 16 different sequences. The annotations were done manually for all the video sequences. The video frames were prepossessed based on the approach provided in [23].

4.2 Experiment

Table 1 provides the results of the proposed approach. The data were split into 70% training and 30% testing sets. Tenfold cross validation was performed to train the model. The columns with † indicate the approach using one-dimensional vector inputs to SVM. The ISOMAP neighborhood parameter k was set 7. The texture 100-dimensional feature vectors are represented as one-dimensional vectors using the ISOMAP. These one-dimensional vectors represent the video frames, which are used for learning the SVM mapping function. Similarly, the columns with ‡ indicate the approach using 10-dimensional vector input to SVM. Furthermore, the SVM-Linear indicates the use of SVM with linear kernel and SVM-RBF indicates the use of SVM with RBF kernel.

From Table 1, the ISOMAP+SVM-Linear (†) and ISOMAP+SVM-RBF (†), both performed equally well in classifying the merging events (precision: 0.78, recall: 0.99, F-score: 0.87). However, ISOMAP+SVM-RBF (†) showed better performance in detecting and classifying the splitting events (precision: 1.00, recall: 0.78, F-score: 0.87). Overall, both ISOMAP+SVM-Linear (†) and ISOMAP+SVM-RBF (†) performances were similar. The SVM-Linear (‡) and SVM-RBF (‡) both performed equally well in detecting and classifying the splitting events with SVM-RBF (‡) performing slightly better than SVM-Linear (‡). However, the ISOMAP+SVM-Linear (‡) performed better in case of dispersion and evacuation events. The ISOMAP+SVM-RBF (‡) showed excellent performance in the case of merging event as compared with ISOMAP+SVM-Linear (‡).

Compared with existing methods from Table 1, running event was best detected by statistical filtering approach [32] with precision, recall, and F-score of 0.99 each. The statistical filtering approach [32] also achieved a recall score of 1 in splitting events. One of the critical point to note here about the approach presented in [32] is that the experiment design included only two classes—running and splitting. This established a high score for running and splitting as both are clearly distinguishable among crowd events. The Random Forest and Motion Pattern approach described in [2] performed well to detect walking events in crowd videos. The method presented in [2] models the crowd motion using the optical flow features and refines the motion patterns. This enables the method to provide better walking detection scores.

In [24], the authors presented a crowd event detection approach using Riemannian manifolds. In [24] the approach models the crowd movement based on the optical flow features. The scheme, which is an unsupervised approach, finds the location of the crowd groups at any time based on the temporal evolution of the crowd locations, which is determined by the localization of crowd groups on Riemannian manifolds. This localization provided better results to detect dispersion events. In contrast, the supervised approach presented in this work, performed well in detecting merging, splitting, dispersion, and evacuation.

One of the main problems with the crowd event detection is the processing of large amounts of video data. Linear dimensionality reductions such as PCA violate the data nonlinearity. Nonlinear approach such as ISOMAP preserve the nonlinear structure of the data by first constructing the graph using frames as vertices and then finding the low-dimensional embedding based on global optimization such that the isometry of the data is preserved in both higher and lower dimensions. The work presented reinstates that using statistical features from Haar wavelet is suitable to detect crowd events. This work also shows that using manifold learning algorithms, the high-dimensional feature space of videos can be reduced to a low-dimensional representation and still detect the events with a trained classifier. Future work includes experiments to have an unsupervised learning system that can automatically detect the manifold parameters, such as k (neighborhood), and also to detect the events from low-dimensional feature vectors without using supervised classifiers. More research is also required to determine as to what kind of features are better suited to detect crowd events and in general crowd monitoring. In this work, only the PETS 2009 dataset was used to detect crowd events as only the PETS 2009 dataset had the crowd events defined in the literature. The research community should also focus on sharing annotated crowd movement data to develop robust algorithms.

5 Conclusion

In this work, an approach to detect crowd events such as running, walking, merging, splitting, dispersion and evacuation, was presented. However, crowd event detection using videos is a highly challenging task, and the video itself is a high-dimensional

data and analyzing to detect events becomes further complicated. In this work, the huge volume of video data was represented using low-dimensional equivalent. The ISOMAP was used to reduce the high-dimensional nature of the video to a low-dimensional representation. Later, an SVM was trained to detect the six crowd events based on the low-dimensional feature vectors. The experiment used the PETS 2009 dataset, which consists of the six crowd events identified in the literature. The proposed approach used Haar wavelets to generate GLCM from which the four statistical features (contrast, correlating, energy, and homogeneity) at different levels of Haar wavelet decomposition were extracted. The proposed approach performed better in detecting merging, splitting, dispersion, and evacuation, compared with existing approaches. The work presented reinstates that using statistical features from Haar wavelet is suitable to detect crowd events. This work also shows that using manifold learning algorithms, the high-dimensional feature space of videos can be reduced to a low-dimensional representation and still detect the events with a trained classifier.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6), 1373–1396 (2003). DOI 10.1162/089976603321780317
2. Benabbas, Y., Ihaddadene, N., Djeraba, C.: Motion pattern extraction and event detection for automatic visual surveillance. J. Image Video Process. **2011**, 1–15 (2011). DOI 10.1155/2011/163682
3. Chan, A.B., Morrow, M., Vasconcelos, N.: Analysis of crowded scenes using holistic properties. In: Performance Evaluation of Tracking and Surveillance workshop at CVPR, pp. 101–108. IEEE (2009)
4. Chang, C.C., Lin, C.J.:
5. Chang, Y., Hu, C., Feris, R., Turk, M.: Manifold based analysis of facial expression. Image and Vision Computing **24**(6), 605–614 (2006)
6. Chen, Y., Zhong, Z., Ka Keung, L., Yangsheng, X.: Multi-agent based surveillance. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2810–2815. IEEE (2006). DOI 10.1109/iros.2006.282064
7. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Computer Vision–ECCV 2006, pp. 428–441. Springer (2006)
8. Ekin, A., Mehrotra, R., et al.: Automatic soccer video analysis and summarization. IEEE Transactions on Image Processing **12**(7), 796–807 (2003)
9. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(12), 2179–2195 (2009)
10. Ferryman, J.: PETS 2009 benchmark data. `http://www.cvg.rdg.ac.uk/PETS2009/a.html` (2009). [Online; accessed 19-May-2014]
11. Foresti, G.L., Micheloni, C., Snidaro, L., Remagnino, P., Ellis, T.: Active video-based surveillance system: the low-level image and video processing techniques needed for implementation. IEEE Signal Processing Magazine **22**(2), 25–37 (2005)
12. Gárate, C., Bilinsky, P., Bremond, F.: Crowd event recognition using hog tracker. In: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), pp. 1–6. IEEE (2009). DOI 10.1109/pets-winter.2009.5399727
13. Guo, G., Fu, Y., Dyer, C.R., Huang, T.S.: Image-based human age estimation by manifold learning and locally adjusted robust regression. IEEE Transactions on Image Processing **17**(7), 1178–1188 (2008)

14. Haar, A.: Zur theorie der orthogonalen funktionensysteme. Mathematische Annalen **69**(3), 331–371 (1910)
15. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology **24**(6), 417–441 (1933). DOI 10.1037/h0071325
16. Hughes, R.L.: A continuum theory for the flow of pedestrians. Transportation Research Part B: Methodological **36**(6), 507–535 (2002)
17. Ke, Y., Sukthankar, R., Hebert, M.: Volumetric features for video event detection. Interantional Journal of Computer Vision **88**(3), 339–362 (2010). DOI 10.1007/s11263-009-0308-z
18. Lee, J.M.: Riemannian Manifolds: An Introduction to Curvature, vol. 176. Springer (1997)
19. Li, R., Chellappa, R., Zhou, S.K.: Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2450–2457. IEEE (2009). DOI 10.1109/cvpr.2009.5206676
20. Ma, Y., Fu, Y.: Manifold Learning Theory and Applications. CRC Press, Inc. (2011)
21. Mallat, S.: A wavelet tour of signal processing: the sparse way. Academic press (2008)
22. Pless, R., Souvenir, R.: A survey of manifold learning for images. IPSJ Transactions on Computer Vision and Applications **1**, 83–94 (2009)
23. Rao, A.S., Gubbi, J., Marusic, S., Palaniswami, M.: Estimation of crowd density by clustering motion cues. The Visual Computer pp. 1–20. (2014). DOI 10.1007/s00371-014-1032-4. 10.1007/s00371-014-1032-4
24. Rao, A.S., Gubbi, J., Marusic, S., Palaniswami, M.: Probabilistic detection of crowd events on riemannian manifolds. In: 2014 International Conference on Digital lmage Computing: Techniques and Applications (DlCTA), pp. 1–8. IEEE (2014). DOI 10.1109/DICTA.2014.7008124
25. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000). DOI 10.1126/science.290.5500.2323
26. Souvenir, R., Pless, R.: Manifold clustering. In: Tenth IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 648–653. IEEE
27. Souvenir, R., Pless, R.: Image distance functions for manifold learning. Image and Vision Computing **25**(3), 365–373 (2007)
28. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000). DOI 10.1126/science.290.5500.2319
29. Thida, M., How-Lung, E., Remagnino, P.: Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes. IEEE Transactions on Cybernetics **43**(6), 2147–2156 (2013). DOI 10.1109/TCYB.2013.2242059
30. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, pp. 839–846. IEEE (1998)
31. Torgerson, W.: Multidimensional scaling: I. theory and method. Psychometrika **17**(4), 401–419 (1952). DOI 10.1007/BF02288916
32. Utasi, Á., Kiss, Á., Szirányi, T.: Statistical filters for crowd image analysis. In: Proceedings of the 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (in conjunction with CVPR 2009). IEEE (2009)
33. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys **38**(4) (2006). DOI 10.1145/1177352.1177355

Table 1: The last four columns of the table provide the results of the proposed approach. The bold fonts indicate the best results obtained for each event. The (†) indicates the approaches using the one-dimensional vector input to SVM and (‡) indicates the approaches using a ten-dimensional vector input to SVM.

Crowd Event	Measure	Statistical Filters [32]	Holistic Approach [3]	Random Forest [2]	Motion Pattern [2]	Riemannian Manifolds [24]	ISOMAP + SVM-Linear (†)	ISOMAP + SVM-RBF (†)	ISOMAP + SVM-Linear (‡)	ISOMAP + SVM-RBF (‡)
Walking	Precision	-	0.87	0.96	**0.97**	0.61	0.61	0.64	0.65	0.86
	Recall	-	-	**0.99**	0.96	0.75	0.81	0.76	0.81	0.87
	F-score	-	-	**0.97**	0.96	0.67	0.69	0.68	0.8	0.86
Running	Precision	**0.99**	0.75	0.86	0.75	0.78	0.81	0.76	0.87	0.87
	Recall	**0.99**	-	0.68	0.81	0.63	0.61	0.64	0.86	0.86
	F-score	**0.99**	-	0.75	0.77	0.69	0.69	0.68	0.86	0.86
Merging	Precision	-	0.68	0.65	0.59	0.85	0.78	0.78	0.88	**0.96**
	Recall	-	-	0.46	0.45	0.88	**0.99**	**0.99**	0.97	0.98
	F-score	-	-	0.53	0.51	0.86	0.87	0.87	0.93	**0.96**
Splitting	Precision	0.65	0.74	0.73	0.47	0.66	0.99	**1.00**	0.97	0.98
	Recall	**1.00**	-	0.92	0.47	0.6	0.78	0.78	0.88	0.96
	F-score	0.78	-	0.81	0.47	0.62	0.87	0.87	**0.96**	**0.96**
Dispersion	Precision	-	0.8	0.58	0.67	0.9	0.59	0.79	**0.98**	0.96
	Recall	-	-	0.48	0.45	**0.94**	0.84	0.64	0.92	0.90
	F-score	-	-	0.52	0.53	0.91	0.69	0.70	**0.95**	0.93
Evacuation	Precision	-	**0.94**	0.83	0.69	0.75	0.84	0.64	0.92	0.90
	Recall	-	-	**1.0**	0.82	0.65	0.59	0.79	0.96	0.96
	F-score	-	-	0.90	0.74	0.69	0.69	0.70	**0.93**	0.92

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-28658-7_8