MID-LEVEL FEATURE SET FOR SPECIFIC EVENT AND ANOMALY DETECTION IN CROWDED SCENES

F. de-la-Calle-Silos, I. González-Díaz, F. Díaz-de-María

Department of Signal Theory and Communications
Universidad Carlos III, Leganés (Madrid), Spain

ABSTRACT

In this paper we propose a system for automatic detection of specific events and abnormal behaviors in crowded scenes. In particular, we focus on the parametrization by proposing a set of mid-level spatio-temporal features that successfully model the characteristic motion of typical events in crowd behaviors. Furthermore, due to the fact that some features are more suitable than others to model specific events of interest, we also present an automatic process for feature selection. Our experiments prove that the suggested feature set works successfully for both explicit event detection and distance-based anomaly detection tasks. The results on PETS for explicit event detection are generally better than those previously reported. Regarding anomaly detection, the proposed method performance is comparable to those of state-of-the-art method for PETS and substantially better than that reported for Web dataset.

Index Terms— Machine Vision, Video processing , Video surveillance, Crowded environments, Clutter environment, Motion analysis

1. INTRODUCTION

Crowd behavior in public areas is currently of great interest to the computer vision community. More and more scenarios involving crowds have to be monitored, such as airports, transport stations, or public events. However, the current video surveillance systems still lack of the desired level of automation, requiring operators who monitor a high number of cameras. For this reason, there is an increasing demand of systems that are capable of either automatically triggering alarms in case of an abnormal situation, or directly identifying explicit events of interest in the surveillance video. The system proposed in this paper aims to detect abnormal behaviors, such as crowd formations or evacuations, allowing the operator to pay more attention to these potentially risky events.

Detection of unusual events in video surveillance has been traditionally addressed following one of these two approaches: a) explicit event detection [1]; and b) anomaly detection [2]. The former considers the problem of modeling a set of predefined events of interest in video surveillance so that they may later be detected in real-time operation. This particular approach for event detection in video surveillance has been promoted by various international challenges, such as TRECVID [3] or PETS [4]. When applied to crowded scenarios (e.g. PETS) these events describe actions such as people walking or running, evacuation, crowd formation, crowd splitting, crowd dispersion, etc.

However, considering a predefined set of events of interest is sometimes inadequate due to the large amount of potential activity patterns of the crowds. In these cases, therefore, a generic detection of anomalies seems to be more appealing. Since these systems simply consider anomalies or unusual events as those that substantially differ from what is considered usual activity, they rely on modeling the usual activities, for which a large amount of training data is available.

Regardless of the selected approach (explicit event or anomaly detection), a typical system for activity recognition in video surveillance (e.g. [5], [6]) usually include some of the following processing steps: (1) *Background subtraction*: areas belonging to objects of interest (pedestrians or crowds) are detected and segmented in each frame in order to reduce computations and improve the precision of the subsequent analysis; (2) *Object tracking and motion estimation*: objects are tracked along the following frames in order to extract motion information; (3) *Feature extraction*: a set of features is extracted from the tracked objects; and (4) *Event/anomaly detection*, a supervised or unsupervised system makes decisions based on the previously computed features.

This paper mainly focuses on the feature extraction step. Specifically, a novel set of mid-level features is presented that models several aspects of crowd behavior. The proposed set was applied to both explicit event detection and anomaly detection tasks. Furthermore, concerning the explicit event detection scenario, some features are more informative than others for each particular event being detected. Even more, there would be features that, whereas become of great importance to detect a specific event, become noise when detecting other. In consequence, this paper also presents a systematic procedure which, given an initial feature pool and a set of explicit events to be detected, automatically selects the most suitable subset of features for each particular event.

The remainder of this paper is organized as follows: Section 2 provides an overview of the state of the art; Section 3 describes the proposed system in detail; Section 4 assesses our proposal in several scenarios and databases. Finally Section 5 draws conclusions and introduces potential future lines of research.

2. RELATED WORK

In this section we briefly review the literature on feature extraction for event recognition in crowds-related video surveillance. Numerous specific approaches have been developed to detect events in crowds. In all of them some selected spatio-temporal features become the basis for the detection process. However, the features themselves should not be considered separately since other elements of the processing pipeline are designed in accordance with them in order to achieve competitive results.

Concerning explicit event detection tasks, the work in [7] presented a system based on holistic (global) spatial features such as the area and the perimeter of the foreground objects, the number of

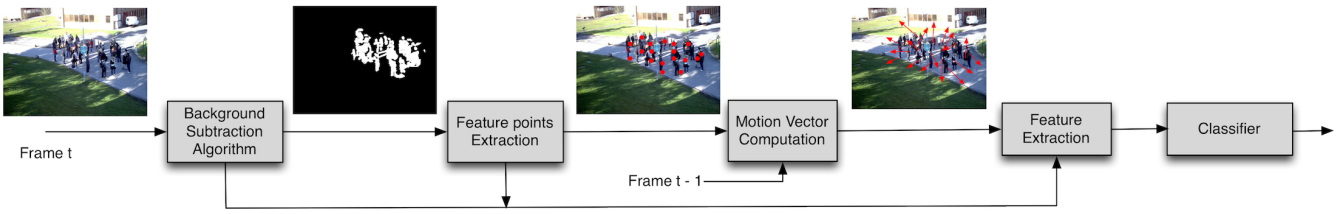


Fig. 1. Flowchart of the processing pipeline of our proposal.

edge pixels, and various texture features, such as homogeneity or entropy. In [8], a simple representation of the crowd motion was built based upon histograms of motion direction alongside an indication of motion speed. A more elaborated approach can be found in [1], where the authors described a method that first computed the optical flow between frames, and then considered histograms of motion magnitude and orientation in multiple coordinate systems. Similarly, the work in [5] built up models for crowd event recognition using features such as the mean velocity, as well as other attributes of the motion direction and the spatial density of the crowd.

Regarding anomaly detection tasks, several features can be found in the literature: the work in [2] proposed to localize abnormal events using a social force model. This model relied on a grid of particles that is placed over the image and advected with the averaged optical flow. A force flow is estimated for every pixel in every frame and randomly selected spatio-temporal volumes of force flow are used to model the normal behavior of the crowd. In [9] a feature set is proposed that is based on data properties in the Fourier domain. Finally, in [10] the authors proposed to work on block-clips representing non-overlapping spatio-temporal patches, and compute their flow field by means of a 2D-mixture of Gaussian.

Although the literature usually proposes specific solutions for explicit event or anomaly detection, we claim that both of them can be successfully approached using detection systems that rely on the same problem parametrization. Hence, our objective is to demonstrate how the same set of features can yield good performance in both scenarios.

3. PROPOSED METHOD

The pipeline of the proposed system, shown in Figure 1, is as follows: first, objects of interest in each frame are detected and masks associated with the foreground areas are computed. Next, a set of salient points is extracted from foreground areas, and then these points are tracked with respect to the previous frame, thus obtaining their motion vectors. Subsequently, a set of mid-level features is computed in the foreground areas. Finally, a detector determines when an event of interest occurs. As already mentioned, although each step is explained below, our work focuses on the feature extraction process, which will be described more in-depth.

Foreground areas that correspond to the objects of interest are computed using a background subtraction algorithm that fuses the results from two methods: a) background models that are specifically learnt for each surveillance camera; and b) temporal updating models. The background models are generated from frames recorded in absence of people in the scene, and thus provide very precise approximations of the background. In particular, and following approaches found in the literature such as [11], an appearance model for each pixel in the image is built by fitting a Mixture of Gaussians (MoG) model, so that various illumination conditions or even non-static backgrounds can be handled. In the test phase, the likelihood of each pixel is computed and compared with a threshold to make a decision.

This model, however, presents two main drawbacks: first, in some cases there are not available recordings of the empty scene to learn the backgrounds; and second, some illumination changes (mainly due to varying shadows with the time of the day) are not correctly modeled. To overcome these issues, we propose the use of a temporal updating model, which is computed by subtracting the current frame from a reference one obtained as a running average of the previous frames. This second model generates a rough approximation of the foreground areas (using morphological dilations) that, when combined with the background model (using an AND logical operator), provided a refined version of the initial foreground masks by removing areas associated with illumination changes.

Once the foreground masks have been computed, salient points are extracted using FAST [12] just in the regions of interest, what notably reduces the computational cost of the proposed approach. Furthermore, only the best N points (80 in our experiments), those exhibiting the highest corner strengths, are considered for the subsequent tracking process. Next, HOG [13] is used to describe the local area around each detected point, and the Histogram Intersection (HI) [14] is used for feature comparison. The output of this stage is a set of pairs ($\mathbf{p}$,$\mathbf{u}$), where $\mathbf{p}$ represents a key-point and $\mathbf{u}$ its associated motion vector, from which we build the proposed set of features.

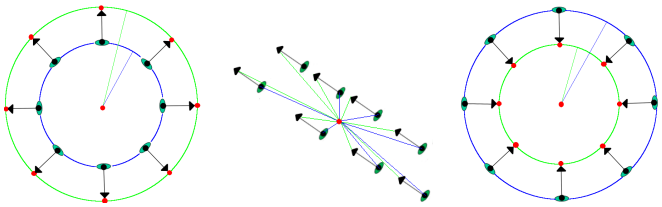
3.1. Proposed features

This section describes our problem parametrization. Based on the detected objects, salient points and motion vectors, we build a set of spatio-temporal mid-level features. One of the desirable properties of the feature set is to be independent of the addressed task or the event to be detected. In the following paragraphs we describe each of the features:

1. **Spatial Location (SL)**: each frame is divided using a 8×6 grid and the number of salient points in each cell is computed. By dividing it by the total number of points in the image, the obtained normalized histogram allows us to model the spatial location (given a specific camera) in which an event tends to occur.

2. **Average Velocity (AV)**: for each frame, the average velocity of the detected objects is computed over a sliding window of 5 (previous) frames. This feature is specially suitable for detecting simple translation events, such as people walking or running.

3. **Dispersion Change (DC)**: defining dispersion as the sum of the distances between the spatial location of each salient point $\mathbf{p}_n(i)$ and the average location $\bar{\mathbf{p}}_n$ in a frame n, this descriptor computes the



(a) $DC > 1$ Evacuation (b) $DC = 1$ Same direction (c) $DC < 1$ Formation

Fig. 2. Distance ratio examples

ratio of the dispersion between two consecutive frames as:

$$DC(n) = \frac{\sum_{i=1}^{N} d(\mathbf{p}_n(i), \bar{\mathbf{p}}_n)}{\sum_{i=1}^{N} d(\mathbf{p}_{n-1}(i), \bar{\mathbf{p}}_{n-1})} \quad (1)$$

where $d(p, q)$ stands for the euclidean distance between two points. It is worth noticing that $\mathbf{p}_{n-1}(i)$ is computed as $\mathbf{p}_{n-1}(i) = \mathbf{p}_n(i) - \mathbf{u}_n(i)$, using the motion vectors obtained in the motion estimation stage. As illustrated in Figure 2 this ratio allows us to discriminate between several crowd motion patterns.

4. **Divergence (Div)**: Similarly to DC, this feature also focuses on motion; however, both are completely different in nature. In this case the motion of salient points is first parametrized using an Affine Transformation Matrix. In particular, we assume that there exists an affine matrix A that transforms the locations of points in the previous frame into their new locations in the current frame: $P_n = \mathbf{A}P_{n-1}$, where P_n is a 3xN matrix that encompasses the N points $\mathbf{p}_n = \{x_n, y_n, 1\}$ in the frame n, and A can be defined as follows:

$$\mathbf{A} = \begin{bmatrix} \epsilon\cos\theta & -\epsilon\sin\theta & t_x \\ \epsilon\sin\theta & \epsilon\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The matrix that best fits the previously computed motion field $\mathbf{u}$ is estimated using RANSAC [15], which has shown to be very robust against any outlier caused by errors in the tracking step. Finally, the Divergence requires to identify those elements in the matrix that model a change on the scale. This parameter, widely used for modeling camera motion patterns such as zooms [16], can be mathematically expressed as:

$$Div = 2\epsilon\cos\theta \quad (3)$$

This feature models events with divergent motions like crowd formations and evacuations.

5. **Histogram of Motion Orientations (HMO)**: A histogram of motion orientations is computed for each frame based on the previous calculated motion vector field. In particular, 12 orientations have been considered in our approach.

3.2. Feature selection using mutual information

As we have already mentioned, in an explicit event detection scenario, some features are more suitable than others depending on the event being detected. Hence, we have developed an automatic method for feature selection that optimizes the subset of features used to detect each event of interest.[1] The method is based on the measure of the Mutual Information (MI) between the features and a ground truth vector that describes the occurrence of the events. Obviously, due to the need of ground truth labels, this procedure can only be used in supervised scenarios.

Algorithm 1 Feature selection using mutual information.

1: Start with the empty set X^0 and consider an initial mutual information value $MI^0 = 0$.
2: **for** $t = 1 \rightarrow N_f =$ Number of features **do**
3: **for** each feature i not included in the set **do**
4: Compute MI between the extended set $X_i^t = \{X^{t-1}, X_i\}$ and the ground truth vector: $I(X_i^t; Y)$.
5: Compute the increment on the MI as $\Delta MI_i = MI_i^t - MI^{t-1}$.
6: **end for**
7: **if** $\Delta MI_i <= 0$ for every i **then**
8: The feature set X_{t-1} is selected and the algorithm ends.
9: **else**
10: Select the feature $X*$ that maximizes ΔMI_i and add it to the set $X^t = [X^{t-1}, X^*]$
11: **end if**
12: **end for**

Feature	Walk.	Run.	Eva.	Form.	Split.	Disp.
SL	✓	✓	✓	✓	✓	×
AV	✓	✓	✓	✓	✓	✓
DC	×	✓	✓	✓	×	✓
Div	×	×	✓	✓	×	✓
HMO	×	×	×	×	✓	×

Table 1. Selected features for each event using the mutual information method

The mutual information $I(X; Y)$ between two random variables X and Y measures the mutual dependence between them; in other words, it is the amount of uncertainty about Y that is removed by knowing X: $I(X; Y) = H(X) - H(X|Y)$, where H is the entropy of a variable. The mutual information was calculated using MILCA algorithm [17]. More details about the implemented method are given in the Algorithm 1.

The selected features for each event of the PETS dataset are shown in Table 1. As can be observed, the features selected for each particular event intuitively fit its nature; e.g. on the evacuation event the feature selection process includes: Average velocity, Spatial Location, Dispersion Change, and Divergence. All these features are related to the nature of the evacuation event, in which people move quickly from the center to the borders of the scene.

4. EXPERIMENTAL RESULTS

Our proposal has been assessed in two different tasks, explicit event detection and anomaly detection, using several databases for which other state of the art techniques have reported results.

4.1. Explicit event detection

The explicit event detection has been evaluated on the PETS 2010 dataset [4], which contains videos representing 6 different crowd

[1] Source code at: `http://www.tsc.uc3m.es/~fsilos/code/code.html`

Method	Walk.	Run.	Eva.	Form.	Split.	Disp.
			ACC			
Proposed	**0.98**	**0.98**	**0.99**	0.98	**0.98**	**0.97**
Proposed w/o FS	0.97	0.95	0.96	0.97	0.98	0.95
[1]	0.97	**0.98**	**0.99**	**0.99**	**0.98**	0.94
[7]	0.87	0.88	0.94	0.68	0.77	0.80
			F			
Proposed	**0.92**	**0.85**	**0.86**	**0.88**	**0.98**	**0.78**
[8]	—	0.75	0.20	0.41	0.59	0.30
[5]	0.83	0.72	0.75	0.34	0.49	0.30

Table 2. Results for explicit event detection in PETS dataset

DB	[9]	[10]	[7]	[2]	Ours
PETS (ACC)	0.96	**0.97**	0.81	—	0.96
Web (AUC)	—	—	—	0.89	**0.96**

Table 3. Results for anomaly detection in PETS and Web datasets

events: walking, running, evacuation, formation, splitting and dispersion. Since PETS is a relatively small database, we have followed a 4-fold cross-validation approach. Thus, the database was divided into 4 subsets, 3 of them being used to train a SVM with RBF-Kernel and the remaining one used in test to assess the performance. This experiment was repeated 4 times, one for each test subset, and the average results were considered for comparison.

Table 2 shows the results of our proposal in comparison to state-of-the-art approaches previously reported for this database. In particular, we have compared two different versions of the proposed method: with (Proposed) and without feature selection (Proposed-w/o FS), so that the contribution of the feature selection process can be evaluated. Two metrics have been used for evaluation: detection accuracy and F-measure. We computed both to provide appropriate comparisons to already reported results (which used one or another). As it can be seen our method performs almost always better than current state of art techniques. Furthermore, it is worth noting the improvement achieved by our method for the "Dispersion" event. In our opinion, this result is due to the proper motion parameterization performed in the DC and Div features for this type of event.

The direct comparison of the two versions of the proposed approach allows us to conclude the effectiveness of the feature selection process described in Section 3.2. Additionally, the feature selection process allows for reducing the computational cost of the final system. Finally, some visual examples are shown in Figure 3.

4.2. Anomaly detection

The performance of proposed system in the anomaly detection task has been tested in two datasets: PETS and Web dataset [2]. In PETS, we considered any event as an anomaly except for "walking". In Web dataset there are 12 sequences of normal scenes, such as pedestrian walking or running, and 8 scenes of abnormal events, like crowd fighting, escape panics, and protesters.

The classification approach is different for this task. In particular, it is assumed that either all or, at least, most of the samples used for training correspond to normal scene. Therefore, we substituted the SVM by a Mixture of Gaussians (MoG) that learnt the distribution of data in normal scenarios. Then, in the test stage, our system computed the likelihood of the data with respect to the learnt MoG, and made its decision by comparing it with a threshold (break-even-point). For training purposes, in PETS we used all the available "walking" videos, whereas for Web dataset we followed the experimental protocol described in [2].

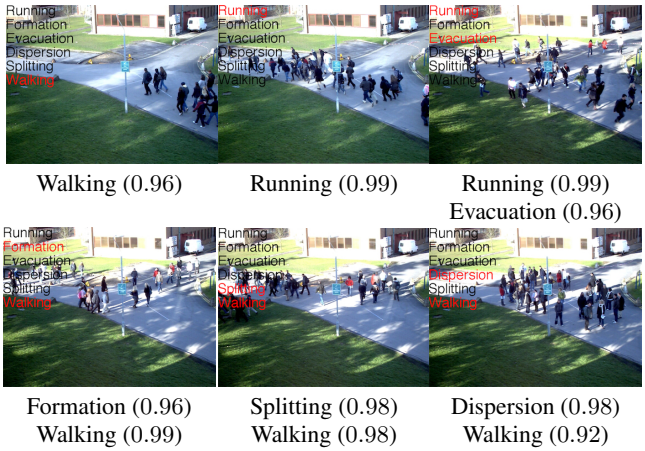


Fig. 3. Examples of the detection of several events on PETS [4]. The probability of each detected event is indicated below the images; furthermore, when an event is detected the font color changes to red.

Fig. 4. Examples of anomaly detection results from Web dataset [2]. Top row contains normal scenes whereas bottom row shows abnormal scenes

Again, we evaluated our approach using two metrics, accuracy and area under the curve (AUC), to provide appropriate comparisons. The results are shown in Table 3. For PETS, our system performed nearly as well or better than previously described systems ([9], [10] and [7]). Furthermore, for Web the performance of our method was well above the state-of-the-art results reported in [2].

In addition to these results, some visual examples of the output of the anomaly detection system are shown in Figure 4.

Finally, some selected videos demonstrative of the achieved results have been made available in [18], as well as some illustrative videos of the intermediate results generated at output of each step of the processing pipeline.

5. CONCLUSIONS AND FURTHER WORK

In this paper we have presented a method for detection of explicit events and abnormal behavior in crowded scenes. We have focused on the design of a set of spatio-temporal mid-level features that allow us to successfully model crowd behaviors. Furthermore, we have suggested an automatic feature selection approach. We have assessed the proposed feature set for explicit even detection on the PETS dataset and our results compare favourably to the state-of-the art. We have also evaluate the feature set for anomaly detection on two datasets, PETS and Web dataset, obtaining again quite competitive performance. Future work will focus on the classifier and on a more comprehensive evaluation.

6. REFERENCES

[1] Teng Xu, Peixi Peng, Xiaoyu Fang, Chi Su, Yaowei Wang, Yonghong Tian, Wei Zeng, and Tiejun Huang, "Single and multiple view detection, tracking and video analysis in crowded environments," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, sept. 2012, pp. 494 –499.

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

[3] M. Michel, J. Fiscus, and P. Over, "Trecvid 2010 video surveillance event detection task," in *TRECVID 2010 Workshop*, 2010.

[4] IEEE Computer Society, IEEE Signal Processing Society, and Boston University., "Performance evaluation of tracking and surveillance 2010 database," `http://www.cvg.rdg.ac.uk/PETS2010/`, 2010.

[5] C. Garate, P. Bilinsky, and F. Bremond, "Crowd event recognition using hog tracker," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, dec. 2009, pp. 1 –6.

[6] Min Hu, S. Ali, and M. Shah, "Detecting global motion patterns in complex videos," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, dec. 2008, pp. 1 –5.

[7] A. B. Chan, M. Morrow, and N. Vasconcelos., "Analysis of crowded scenes using holistic properties," in *In IEEE Intl. Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2009)*, june 2009.

[8] H.M. Dee and A. Caplier, "Crowd behaviour analysis using histograms of motion direction," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, sept. 2010, pp. 1545 –1548.

[9] A. Briassouli and I. Kompatsiaris, "Spatiotemporally localized new event detection in crowds," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, nov. 2011, pp. 928 –933.

[10] S.S. Pathan, A. Al-Hamadi, and B. Michaelis, "Crowd behavior detection by statistical modeling of motion patterns," in *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*, dec. 2010, pp. 81 –86.

[11] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, Los Alamitos, CA, USA, Aug. 1999, vol. 2, pp. 246–252 Vol. 2, IEEE.

[12] Edward Rosten, Reid Porter, and Tom Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, 2010.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, june 2005, vol. 1, pp. 886 –893 vol. 1.

[14] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, sept. 2003, vol. 3, pp. III – 513–16 vol.2.

[15] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[16] R.S. Jadon, Santanu Chaudhury, and K.K. Biswas, "A fuzzy theoretic approach to camera motion detection," *Proceedings of 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2002.

[17] Harald Stögbauer, Alexander Kraskov, Sergey A. Astakhov, and Peter Grassberger, "Least-dependent-component analysis based on mutual information," *Phys. Rev. E*, vol. 70, pp. 066123, Dec 2004.

[18] Fernando de la Calle Silos, "Videos of the system," `http://www.tsc.uc3m.es/~fsilos/crowds.html`.