Detection of static groups and crowds gathered in open spaces by texture classification

Marco Manfredi, Roberto Vezzani, Simone Calderara, Rita Cucchiara

DIEF - University of Modena and Reggio Emilia
Tel. +39 0592056270 Fax. +39 0592056129
{*marco.manfredi, roberto.vezzani, simone.calderara, rita.cucchiara*}*@unimore.it*

Abstract

A surveillance system specifically developed to manage crowded scenes is described in this paper. In particular we focused on static crowds, composed by groups of people gathered and stayed in the same place for a while. The detection and spatial localization of static crowd situations is performed by means of a One Class Support Vector Machine, working on texture features extracted at patch level. Spatial regions containing crowds are identified and filtered using motion information to prevent noise and false alarms due to moving flows of people. By means of one class classification and inner texture descriptors, we are able to obtain, from a single training set, a sufficiently general crowd model that can be used for all the scenarios that shares a similar viewpoint. Tests on public datasets and real setups validate the proposed system.

Keywords: Crowd Detection, Surveillance, One Class SVM

1. Introduction

Streets of our cities are day by days more crowded, impacting our well-being in comfortable environments, affecting our sense of safety and also creating serious problems of security. Studies on crowds and individuals in a crowd are critical for surveillance and real-time proactive control of safe and smart cities, since crowds could cause or be caused by violent events [1]. Several times in the past organized groups of hooligans met up in public areas, scheduled a plan and, armed, they went to the stadium just to look for a brawl; groups of rioting people met each other to create an untamable flow of violence. Conversely, violent events, accidents and unexpected situations can create crowds (for instance at the exit of undergrounds, at the gates of building, etc.) which, in turn, induce additional public safety problems.

The pioneering definitions of crowd discuss the simultaneous presence of densely distributed high number of individuals. In this context studies from Le Bon and Freud in early 20th century treat the crowd as an emotional mass with a intuitive loss of consciousness from single individuals [2, 3]. We refer to this type of crowd as *dense crowd*. Theory of dense crowd of Gustave Le Bon is based on the "contagion phenomenon", in which the crowd goal and formation mechanisms are emerging from a global crowd will, where individuality is lost.

Recent studies observe the collective behavior of crowd under a different perspective. In Turner and Killian [4] observation, the crowd phenomenon can emerge even from small gatherings of people diffuse in a large area, authors refer to this case as the *diffuse crowd* case.

In diffuse crowd, individuality is more observable than in dense crowd. For this reason, the "crowd will" towards a common goal is often not directly verifiable. The behavioral pattern emerging in this situation follows the "convergence theory": the individual will prevails over the collective, resulting on a crowd formation due to the gathering of people sharing a common goal.

Under these premises automatic tools for crowd analysis should take into account the type of the phenomenon object of the study.

The solution proposed in this paper focuses on the surveillance of open environments where the gathering of groups of people generates a diffuse crowd. The purpose of the system is to aid the security officers to timely identify security alerts and eventually acquire the identities of the authors of the crime [5]. In the aforementioned scenario a single or a set of Pan-Tilt-Zoom cam-

eras cooperate to detect groups of people and to capture high quality details useful for identification and recognition tasks.

In particular, the aim of the proposed system is to detect *static crowds*, which can be defined as groups of people that gather together and remain in the same location for a while (see Fig. 1).

Since the people groups we aim to detect are static, they will be characterized by slow and chaotic motion inside a restricted area. The motion direction or other flow-based features are not the most discriminating information as in the case of dense crowds [6]. Instead, the groups of people visually appear differently from the surrounding. This happens only in diffuse crowd where the density is not excessively high. For this motivation, we adopted appearance features, which can be effectively modeled through texture elements.

Texture features are discriminative enough to distinguish crowds inside urban scenes captured from typical surveillance cameras [7], even in case of trees, vehicles, bicycles and other artifacts cluttering the scene. Thus, we propose a specific patch-level descriptor based on Gradient Orientation Co-occurrence Matrix (GOCM) features [8, 7], which takes into account gradients and their spatial distribution. Rectangular patches are extracted from the video sequence and described by GOCM. Patches are then classified as *potential crowd* by means of a One Class SVM, that separates the patches belonging to the crowd class from all the other possible classes belonging to background. In addition, we propose a validation step that exploits motion features, in order to filter out misclassified patches. The idea behind the validation step is to recognize and enhance small movements typical of static groups, thus removing stationary patches. Our proposal exhibits the advantage of being both fast and effective, that can be deduced by several tests performed in different scenarios with various types of crowds; moreover, it has been included in an online surveillance framework to trigger PTZ cameras devoted to capture details and identifying elements of monitored people. Our proposal exhibits

(a) (b)

Figure 1: Examples of static crowds.

the advantage of being both fast and effective, that can be deduced by several tests performed in different scenarios with various types of crowds; moreover, it has been included in an online surveillance framework to trigger PTZ cameras devoted to capture details and identifying elements of monitored people.

The rationale behind our proposal is that, in the cases of diffused crowds, crowd patches can be described by means of their internal textures. This description shares some peculiar characteristics that are independent from the scenario but they depend mostly on the camera viewpoint. Although binary classification approaches appear to be a suitable solution, we hypothesize that in urban surveillance, the negative patches (e.g. patches belonging to the background of the scenes or moving objects such as car or bicycles) cannot be generalized when the scene location varies. This aspect highlights the need of rebuilding the training set for every surveilled location that is both time-costly and often not practicable in real surveillance systems such in the case of public municipalities surveillance networks. We propose to employ the combination of a fast, although general enough, textural description for crowd patches and the one-class classification framework in order to obtain a system capable of high performances with few positive training examples being generally applicable, without re-training, to a plethora of urban scenarios where the camera viewpoint does not dramatically change w.r.t. the training scenario.

Our system is developed in association with the local police department, and is meant to be used for forensic investigation.

2. Related work

Crowd situations have been usually analyzed as a whole in order to catch the global or local flows, or to estimate the people density [9]. A very interesting survey on crowd analysis has been published some years ago by Zhan et al. [10]. As detailed in the survey, different approaches can be followed based on the application and the scene representation type. If the capturing setup allows the detection of individuals [11], people trajectories can be extracted and used to detect anomalous events [12, 13, 14].

In case of very large crowd it is usually impossible to detect each individual, but only statistical measures could be extracted. For example, the work by Ali and Shah [15] is based on segmentation and classification of crowd flows for large gatherings of people, at events such as religious festivals. An HOG based

tracker is used in conjunction with a scenario modeling stage by Garate et al. [16] to categorize crowds events.

Real time systems aiming at estimating the crowd density have already been installed in real scenarios, e.g., London [17] and Genoa [18].

Kim and Grauman [19] exploited the local optical flow in a Markov Random Field framework to detect abnormal activities. Similarly, Hassner et al. [20] exploited motion information to detect anomalies and, in particular, violent crowds.

Motion features were used by Reisman et al. [21] to detect pedestrians from a moving platform, no appearance model is used and static groups are thus ignored. Most of the proposed solutions deal with the absence of a clear view of the target full body, due to crowd clutter, by focusing on the upper body part [22], eventually raising the classification accuracy injecting in the model crowd density information or spatial properties of the scene [23].

Arandjelovic et al. [24] exploited a statistical model of SIFT descriptors occurrences for the detection of crowds from still images; while the results seems promising, the focus of the paper was on high density crowds (e.g. public events) and seems unsuitable for the detection of smaller groups.

A texture-based approach has been proposed by Marana et al. [7] to estimate the crowd density using gray-level dependence matrices. Similarly, Ma et al. [25] described a framework to estimate the crowd density, where a set of features based on the GOCM matrix is computed and exploited in a bag of words classifier. The output is provided at frame level and motion information are completely neglected, generating false positive events in correspondence of textured background regions.

One Class SVMs were introduced by [26] to estimate the support of probability densities in high dimensional spaces, and recently used for speaker diarisation [27] and remote sensing image classification [28].

3. The system architecture

The overall scheme of the proposed system is depicted in Fig. 2, where crowded zones are firstly detected by means of a frame by frame step. The basic framework is composed by two cameras, working in a master-slave configuration: the master camera is fixed and used to detect the crowd zones, while a PTZ slave camera is exploited to extract useful details on the interesting zones. Frames acquired by the master cameras are divided in patches; for each one, a Gradient Orientation Co-occurrence Matrix is computed and provided to

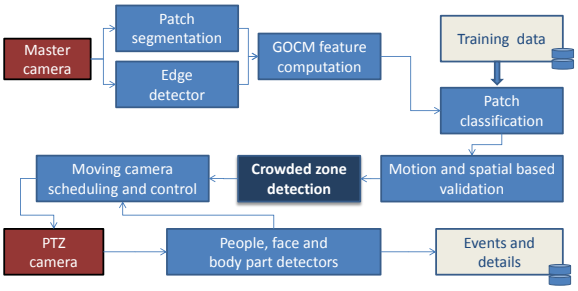

Figure 2: Block diagram of the system architecture

a patch-wise crowd classifier. To this aim, a block-wise One Class Support Vector Machine classifier based on GOCM has been adopted, that proved to be very effective for crowd pattern discrimination. Motion and temporal information are then used to validate each patch: only static crowds are selected by filtering out people flows and short aggregations. The output of the detection system triggers the Pan-Tilt-Zoom camera manager which selects and analyzes more deeply a region of interest, with the aim of extracting and storing the highest number of individuals and visual details as possible. In the implemented system, the depicted block for "people, face and body part detection" has been obtained with a pipeline of off-the-shelf detectors. In particular, the Opencv's [29] HoG people detection [30] and Haar cascade face detection [31] have been integrated and used to process the zoomed regions. However, these last tasks are not detailed since they are beyond the scope of this paper.

4. Texture description and modeling

The core step of the crowd detection is the patch based classifier. Given the input frame at time t, I_t, horizontal and vertical gradients are computed and converted to magnitude $G_m(x, y)$ and orientation $G_\theta(x, y)$ values for each pixel. The frame is split into a set of N overlapping rectangular blocks $R_t^i, i = 1 \dots N$, hereinafter referred as patches. The size of the patches should account for the people size in the image. In particular, a good choice is to generate patches large enough to contain more than one person (see Fig. 5 for some examples). The height can be limited to capture half of the person or the full figure. As a consequence, the overlap among patches is the result of both computational resource availability and the patch size. In the provided experiments, a 50% overlapping rate has been observed as a good tradeoff. To describe the texture we discard color information that may be subjected

to light variations and change from crowd to crowd. Indeed, color is not a key cue in representing crowd since people dresses may vary as well as environmental conditions that change, radically, color perception. Texture descriptors have been widely used in the past to detect and analyze crowds [32, 7]. Among possible texture descriptors, image gradients have been adopted to effectively describe texture [33]. In particular, gradients orientation carries important information about the texture of the patch. Common descriptors for gradient orientation are the well known Histograms of Oriented Gradients (HOG) that have been profitably adopted in several fields from people detection [30] to image retrieval and segmentation [34]. Given an image region, an histogram of gradient directions is computed for every pixel inside the region. Additionally, histograms may be stacked together to constitute the feature vector of an image composed by several regions. In order to include gradients magnitude in the descriptor, a pixel may cast a vote for a direction proportionally to its gradient magnitude. By including magnitude it is possible to strengthen the information carried by strong gradients, while considering less important the one from points with a weak gradient. Disadvantage of HOG is that it gives only statistics on orientation of each single pixel. Spatial information (the relationship) between pairs of pixels is not taken into account. If spatial information is used, more shape information of object can be captured. This information can be computed by the co-occurrence matrix of oriented gradients (GOCM).

The unnormalized GOCM is a matrix with each element $P^*(i, j)$ computed as:

$$P^*(i,j) = \sum_{x,y \in R_t} G_m(x,y) \quad \text{where} \tag{1}$$
$$i = G_\theta(x,y)$$
$$j = G_\theta(\mathbf{v}(\mathbf{x},\mathbf{y}))$$

with $\mathbf{v}(\cdot)$ the neighboring function.

A common choice for the neighboring function is the fixed value neighborhood [8], where, given a vector $\mathbf{u}$, $\mathbf{v}(\mathbf{x}, \mathbf{y}) = (x, y) + \mathbf{u}$. The major drawback of fixed value neighbor is that multiple vectors $\mathbf{u}$ should be adopted in order to completely describe the texture, this leads to the computation of several GOCM, one for each vector. In order to keep the texture descriptor more compact without losing its descriptive power, we adopt the following magnitude based function that allow to obtain a single matrix for a texture region R_t^i:

$$\mathbf{v}(\mathbf{x},\mathbf{y}) = \underset{(x_i,y_i)\in N_r}{\operatorname{argmin}} \|G_m(x_i,y_i) - G_m(x,y)\| \tag{2}$$

where N_r is a circular region centered in (x, y) with fixed radius D (set to 1 in our experiments, thus considering the 8-neighborhood of each pixel). For the computation of the GOCM matrices (P and its unnormalized version P^*), orientations are quantized into 9 bins, leading to a GOCM matrix of 9×9 cells. Algorithm 1 depicts the computation of the GOCM for the patch R_t^i.

Algorithm 1 Gradient Orientation Co-Occurrence Matrix

Require: $R_t^i, G_m(x,y), G_\theta(x,y)$, ▷ patch gradient magnitude and orientation
Initialize $P^*(i,j) = 0 \quad \forall i,j$;
1: **for all** $p_r = (x_r, y_r) \in R_t$ **do**
2: $N_r = \{p_i \in R_t | \|p_i - p_r\| < D\}$ ▷ Neighborhood N_r of p_r
3: $p_b = \mathbf{v}(\mathbf{x_r}, \mathbf{y_r}) = \operatorname{argmin}_{p_i \in N_r} \|G_m(p_i) - G_m(P_r)\|$ ▷ Point $p_b \in R_t$ most similar to p_r
4: $o_r = G_\theta(x_r, y_r)$
5: $o_b = G_\theta(\mathbf{v}(\mathbf{x_r}, \mathbf{y_r}))$
6: $P^*(o_r, o_b) = P^*(o_r, o_b) + G_m(x_r, y_r)$ ▷ update of P^*
7: **end for**
8: $S_p = \sum_{i,j} P^*(i,j)$ ▷ Matrix normalization
9: $P(i,j) = \dfrac{P^*(i,j)}{S_p}$

From the GOCM matrices, the following descriptors are extracted and concatenated to generate the final feature vector:

1. *Sum*: Sum of the elements of P^*, before the normalization (S_p in Algorithm 1, Step 8);
2. $Hom = \sum_{i,j} P_{i,j} / \left(1 + (i-j)^2\right)$: sum of $P_{i,j}$ weighted by their distance to the principal diagonal; high values means low contrasts;
3. *Max*: highest element of P;
4. *Diag*: vector of the principal diagonal values of P. A high element in the diagonal of P indicates the presence of neighbor pixels with similar gradients;
5. $Mcon_i = \sum_j j \cdot P_{i,j}$: the Conditional Mean indicates the expected orientation of a neighbor pixel;
6. *Mean reference* E_r *and mean neighbor* E_n: Mean based on reference and neighbor pixel;
7. *Variance*: Variance based on the reference V_r and the neighbor V_n pixel;
8. $Correlation = \sum_{i,j} P_{i,j} \cdot \frac{(i-E_r)\cdot(j-E_n)}{V_r \cdot V_n}$

Each feature is independently normalized, using reference values extracted from the training set, and the 26-dimensional feature vector $f(R_t^i) \in \mathbb{R}^{26}$ is defined as a compact texture descriptor for the patch R_t^i.

4.1. Patch classification

At classification time, each patch is independently classified from the others. To maintain the adaptability of the system to different scenarios, we choose to create a training set of crowd patches only. This choice is motivated by the very high background variability, that can range from buildings to roads, from vehicles to trees, making a complete description of the background class unfeasible. When only positive examples are considered, supervised learning can be effectively done using One Class Support Vectors Machines (OC-SVMs). The OC-SVM finds a hyperplane that best separates the feature points corresponding to the training samples from the origin of the features space, with a certain degree of freedom represented by the percentage of outliers that can be ignored when defining the boundary. The separation can be non-linearly performed in a mapped space when kernels are used; this is analogous to standard SVM.

4.1.1. One class SVM

We adopted the standard OC-SVM introduced by Schölkopf et al. [26]. Given a set S (of high dimensional input features) containing n points $\left\{\mathbf{x}_j : j = 1 \dots n\right\}$ where $S \in \mathbf{X}$ and $X \in R^d$, we introduce a mapping $\phi(\mathbf{x})$ where the dot product in the feature space can be computed by a kernel function $K(\mathbf{x_i}, \mathbf{x_j})$.

The aim of the OC-SVM is to find the maximum margin hyperplane that best separates the data points from the origin of the feature space allowing a certain percentage of data points to be wrongly classified. This is achieved by finding the vector $\mathbf{w}$ solving the following constrained minimization problem:

$$\frac{1}{2}\|\mathbf{w}^2\| - \rho + \frac{1}{\nu}\sum_j \xi_j \tag{3}$$

subject to the constraints:

$$\mathbf{w} \cdot \phi(\mathbf{x_j}) \geq \rho - \xi_j \tag{4}$$

$$\xi_j \geq 0 \tag{5}$$

The dot product $\mathbf{w} \cdot \phi(\mathbf{x_j}) = \rho$ represents a hyperplane in the feature space, ξ_j a slack variable, as for two classes SVM, and $\|\cdot\|$ the Euclidean norm. We remark that this formulation does not differ significantly from two classes SVMs. The main difference is that here the class value is absent for every data point since all the data points belong to the same class. Additionally, the parameter ν weights the contribution of the slack variables over the data points margin from the hyperplane of separation. This means that by raising or lowering ν it is possible to control the fraction of data points that may reside on the hyperplane and possibly rejecting strong outliers. Making use of the *Wolfe Duality* the dual Lagrangian problem of minimization (3) becomes:

$$\begin{aligned} &\max_\alpha -\textstyle\sum_{ij} \alpha_i \alpha_j K(\mathbf{x_i}, \mathbf{x_j}), \\ &\text{subject to} \quad 0 \leq \alpha_i \leq \tfrac{1}{\nu}, \quad \textstyle\sum_{i=1}^n \alpha_i = 1. \end{aligned} \tag{6}$$

where we make use of the kernel function $K(\cdot)$ to express the dot product between two points in the non-linearly mapped feature space. The output of the learning step is the set of all the α_i; this can be achieved using standard methods for solving the quadratic programming (QP) optimization such as *SVM-light* or the *SMOStep*, [35, 36]. The training examples that have their relative α greater than zero are called the Support Vectors.

To learn crowd patches we exploited the non-Linear OC-SVM, working in a kernel space using a radial basis function kernel. We use as input vector $\mathbf{x}$ the 26 dimensional vector described in Sec. 4.

At classification time test patches are extracted from the image and the GOCM features are computed on them. The final decision whether a patch is a potential crowd patch or not is taken by means of sign of the decision function of OCSVM:

$$g(\mathbf{x}) = sgn\left(\sum_i \alpha_i K(\mathbf{x_i}, \mathbf{x})\right) \tag{7}$$

where only kernel computations with Support Vectors are necessary, because they are the only examples considered to create the hyperplane, i.e. have a nonzero α value.

4.2. Crowd Validation

Crowd regions are obtained merging together connected patches classified as crowd by the One Class SVM using Eq. (7). However, even small or undesired regions could be detected due to noise or moving crowds and they should be filtered out by a validation step. To this aim, a motion based approach has been introduced, taking into account both the instantaneous and the integral motion as validation masks. Starting from the single difference mask, a morphology closing (3x3 structuring element) is applied to merge neighbor moving pixels. The integral motion mask is obtained validating only pixels that constantly show motion, while it cuts out flowing people and highly-textured vehicles in fast transit.

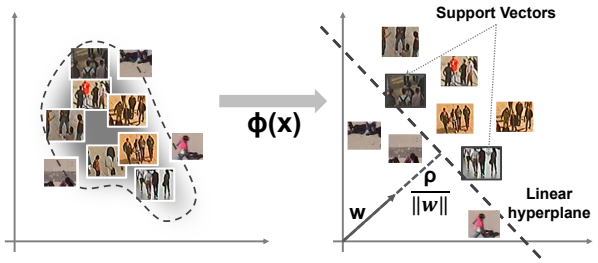

Figure 3: Patch classification using Non Linear One Class SVMs: samples in the feature space are visualized on the left. On the right, support vectors are used in the kernel space to define the separation hyperplane. Samples that lie below the hyperplane are the outliers, whose upper-bound is related to the parameter ν.

Given the single difference mask S, the integral mask I_m at frame n is obtained as follows:

$$I_m(n,i) = \lambda \cdot S(n,i) + (1-\lambda) \cdot I_m(n-1,i) \qquad (8)$$

λ indicates the responsiveness of the integral motion mask to changes in motion patterns, the value of $\lambda = 0.05$ was chosen after several experiments. Finally, the integral motion mask is thresholded using a parameter β obtained from (8), with the aim of considering only regions with a long lasting crowd. Higher is the β, longer is the motion needed for validation. Since Eq. (8) is expressed in terms of frames, the parameter β is related to the frame-rate of the video, a typical value for a video sequence captured at 10 fps is 0.9 (corresponding to approximately one second of continuous motion). Patches classified as crowd are validated if they intersect the binarized integral image.

After the detection of static crowds has been performed, a temporal filter is adopted to select a single Region of Interest (RoI) to control the PTZ camera. The RoI should contain crowd regions which have not been already inspected and thus requires the frame by frame crowd detection and the history of the previous camera movements. Using a round-robin scheduling, the Pan-Tilt-Zoom cameras installed in the same network scan the RoI at a higher zoom level than the master camera. Additional tasks are then performed on the high resolution images, such as a HOG-based people detector [30] or a face detector.

5. Experimental evaluation

To prove the effectiveness of the proposed solution, a set of experiments has been carried out. More precisely, we evaluate the effectiveness of the patch classification algorithm, without the motion validation step. We compare different training procedures varying the training set and classification parameters. We compare OC-SVM against binary SVM to prove the better capability of capturing the crowd model of the former solution. Finally, performance of the complete solution using both motion and textures are evaluated on two publicly available datasets.

Given some sequences of manually annotated videos taken from "Crowds-by-Example" (CbE) [37] and PETS2009 [38] datasets, we tested the accuracy of the crowd detection, including both static and moving crowds. The source code, the training patches and the testing sequences are available online[1]. We compare the proposed texture feature with rectangular HOG, using the same training patches. HOGs are calculated dividing each patch with a 3×3 grid of cells, and grouping them in 4 partially overlapped blocks of 2×2 cells each. Trilinear interpolation between histogram bins and cells was appropriately applied. The HOG feature is computed using 6 bins to quantize the gradient orientation. For each video, a training set of 10 crowd patches is manually extracted. It is important to highlight that both the manual annotations, and the output of the system are at patch level, and are not meant to outline the people with pixel-wise accuracy. Qualitative results on both the datasets are reported in Fig. 4. Given the manually extracted ground truth mask M_{gt} and the output mask of the system M_{out} we can compute some accuracy measures to highlight the mutual agreement of the two. The overlapping ratio O_r is defined as:

$$O_r = \frac{\sum_{p=1}^{P} M_{out}(p) \wedge M_{gt}(p)}{\sum_{p=1}^{P} M_{out}(p) \vee M_{gt}(p)}, \qquad (9)$$

where P is the number of pixels in the image and assuming that the masks take value 1 where crowd is detected. This is commonly used for evaluating segmentation accuracy, thus we expect low values of O_r, due to the coarseness of the output masks. We consider more

[1] http://imagelab.ing.unimore.it/go/crowd

Figure 4: Qualitative results of the patch classification algorithm on the "Crowds by Example" dataset (first row) and PETS2009 dataset (second row).

Table 1: Patch classification results comparing our proposal with HOG features.

Dataset	Density	Groups	Our proposal			HOG		
			R_o	P_o	O_r	R_o	P_o	O_r
PETS2009	low	multi	**0.875**	**1.0**	**0.536**	0.778	0.778	0.456
PETS2009	high	single	0.893	0.953	0.595	**0.893**	**1.0**	**0.614**
CbE dataset	high	multi	**0.823**	**0.698**	**0.472**	0.723	0.694	0.359

reasonable to evaluate the detection accuracy at object level, measuring whether or not a group of people is detected. A detection is correctly provided if the spatial overlapping with the ground truth M_{gt} is higher than a certain threshold (50% of overlap is chosen in this evaluation as suggested by Felzenszwalb et al. [39]). False positive and false negative detections are accordingly defined and the well known precision and recall metrics can be estimated [40]. We call this metric the object-level precision and recall, respectively P_o and R_o. Results are reported in Table 1. GOCM features perform very well on the diffuse crowd scenario (see Sec. 1), in the presence of multiple groups of 2-5 people. Where a single, bigger group of people is present, HOG and GOCM have comparable results. It is important to point out that, at this stage of the method, a high precision is not mandatory, because a further patch validation step will filter out false positives using motion features.

On the same datasets we also tested other kernels for the OC-SVM, in particular the linear and the polynomial kernels. While they are able to reach comparable results in terms of recall when compared to the RBF kernel, the precision values decrease by at least 20%.

Figure 5: Crowd patches used as training.

Table 2: Patch classification results varying the number of training patches.

N Patches	OC SVM		Bin. SVM	
	P_o	R_o	P_o	R_o
10	0.884	0.864	0.668	0.668
20	0.887	0.869	0.691	0.672
30	0.889	0.871	0.766	0.731
50	0.891	0.873	0.812	0.790

5.1. Comparison with Binary SVM

A comparison between the OC-SVM and a binary SVM, trained using patches extracted from the background, is conducted. The same setting used to compare GOCM features and HOG features is used and average values of precision and recall are plotted in Figure 6. The two graphs are obtained varying the parameter C for the binary SVM, and the parameter ν for the OC-SVM. The background patches used to train the binary SVM (100 for each scenario) are taken from the same scenes of the testing videos. The binary SVM seems sensitive to parameter C, and below value 5 no positive crowd patches are found. The OC-SVM is more robust to changes to parameter ν, and overall achieves better results. Positive patches are visually similar, while background patches are scattered in the feature space, also exhibiting crowd-like textures (e.g. trees), this complicates the definition of the boundary of the binary SVM.

We tested both the OC-SVM and the binary SVM varying the number of training patches from 10 to 50, results are reported in Table 2. The performance of the OC-SVM does not significantly improve, while the binary SVM takes advantage of the increased training set. Our system needs to be easily applicable to new settings and the creation of a large training set is costly; given the slight performance improvement when increasing the training set size, we chose to use only 10 positive patches for training.

5.2. Changing the training set

A fundamental aspect of the proposed solution is the applicability to different scenarios, without the need of retraining the system. Of the proposed datasets, two shares a similar view, PETS2009 and ViSOR. We

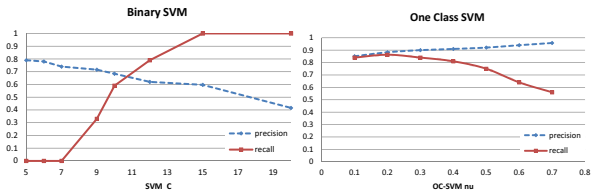

Figure 6: Precision and Recall values on PETS2009 and CbE datasets varying the C and ν parameters of the binary and one class SVMs respectively.

Table 3: Precision and Recall values changing the training set.

Setting	Same Training		Different Training	
	P_o	R_o	P_o	R_o
PETS2009	0.976	0.884	0.93	0.74
CbE	0.731	0.823	0.578	0.434

trained the system using patches of ViSOR and tested it on PETS2009 sequences. On the other hand, the "Crowds by Example" dataset has a completely different viewpoint, but it is included in the experiments to stress the method generality. Table 3 shows that, for similar viewing angles, results' accuracy is not affected negatively by leaving the training set unchanged.

The system has been tested on several real scenarios. The dataset is composed by videos from public repositories such as BEHAVE [41] and ViSOR [42], the latter also contains several videos taken at our campus, with a Pan Tilt Zoom camera placed at 15 meters above the ground watching at a wide public area. Sample frames of the two main setups are reported in the left column of Fig.7. The test set includes urban scenes with different lighting conditions and various types of moving objects like people, bicycles and cars, with challenging situations also for the static crowd classification. The training set has been created using few patches manually selected from the BEHAVE and ViSOR. See Fig. 5 for an excerpt of the training set.

We annotated 25 videos from the above mentioned datasets, containing both static and transiting groups (e.g. the group on the right of Fig. 7, last row) as well as vehicles and cluttered backgrounds as distracting items. The number of people in the sequences range from 3 to 25, with people gathered in small groups of 3-6 individuals. The ground truth and the corresponding evaluation metric are defined at object level, as specified at the beginning of this section. The proposed method is compared with the approach of Ma et al. [25] and with two baselines composed of the patch classification and the motion validation step separately. The comparison with these baselines serves to independently highlight

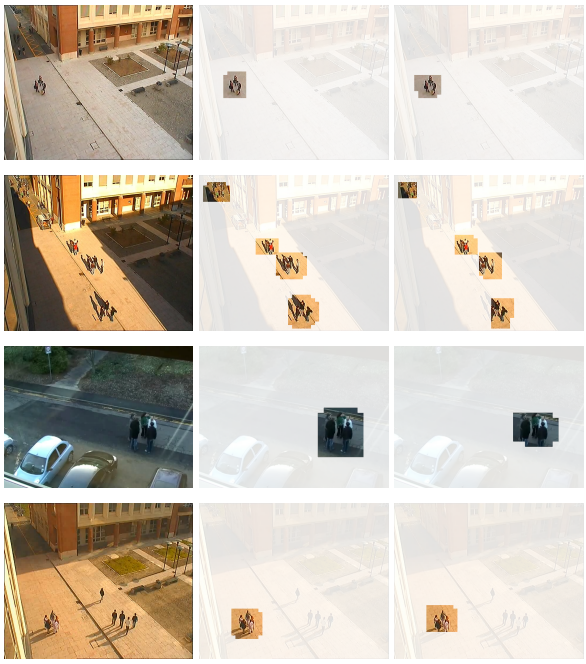

Figure 7: Qualitative results on ViSOR and BEHAVE datasets: input frame (left), ground truth (center) and the system output (right).

Table 4: Compared performances of the proposed and baseline methods.

Method	R_o	P_o
Our approach	0.82	**0.76**
Texture only	0.85	0.48
Motion only	**0.94**	0.32
Ma et al. [25]	0.74	0.69

the contributions of the two steps. Results are reported in Table 4. The system proposed in [25], while reaching a good precision, is not able to correctly generalize the crowd class and several crowd patches are classified as background. The two baselines show similar results, due to the fact that both present a high number of false positives and a very low number of missed crowd patches. To achieve the best tradeoff in terms of precision and recall, the full pipeline is needed. Failure cases of our system are moving groups composed of 10+ people, that are wrongly detected as static crowds. This type of group creates a continuous motion inside a patch, validating it, raising the false positive rate. In Fig. 8 detailed results in terms of precision and recall are reported.

From the computational point of view the system is very fast and real time performances are met. A preliminary and not optimized implementation on a standard

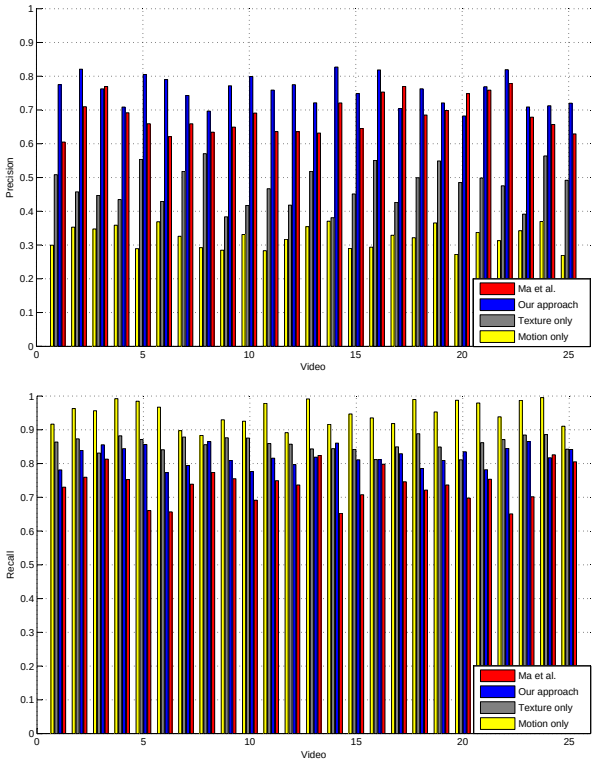

Figure 8: Precision and Recall comparison on 25 annotated videos.

PC is able to process up to five frames per second, with a very limited dependency on the number of training patches adopted.

A software prototype based on the proposed solution has been tested by local police in several urban scenarios. A screenshot of the system is depicted in Fig. 9 where three standing people are correctly detected (a), and a PTZ camera is zooming onto them in order to capture a more detailed image of the subjects (b-c).

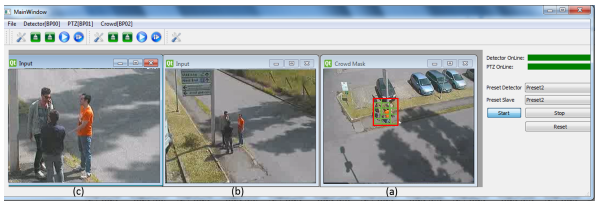

Figure 9: Screenshot of the system working in the Modena surveillance network. (a) Three standing people are detected as a gathering marked in red. (b-c) Two different zooming level performed automatically by a PTZ camera connected to the system

6. Conclusions

In this paper a novel crowd detection technique has been described. Since the specific goal was to identify gatherings of people rather than flows or walking individuals, both texture and motion information have been exploited. We make use of a novel features descriptor that overcomes the performance of well-assessed Histogram of Oriented Gradients. The use of One Class SVM allows the system to learn automatically the needed thresholds making it easily generalizable to settings that shares similar viewing angle.

We aim to show that combining GOCM texture descriptor and one class classification paradigm we are able to reach high performances without the need of retraining the system when the scenario viewpoint does not change. In addition, the OCSVM is able to generalize the crowd model even with few training patches. This is desirable in real systems where the training phase must be as light as possible. We are further investigating both the chance of using classification result to online train the crowd classifier and the use of OCSVMs to bootstrap the training of a binary classifier.

Tests on both publicly available videos and a real setup built in our campus prove the efficacy of the system which is able to reach competitive precision and recall values improving previous state of the art solutions based on texture only. We develop our crowd detection solution in conjunction with the Modena Municipality and will be applied to the public surveillance system of the city for both collecting statistics during events and monitoring potential dangerous situations.

[1] G. B. C. Office, Understanding Crowd Behaviours: V.2: Supporting Theory And Evidence, TSO (The Stationery Office), ISBN 978 0 11 430204 7, 2010.

[2] G. Le Bon, The crowd : a study of the popular mind, T. Fisher Unwin, 1909.

[3] S. Freud, Group psychology and the analysis of the ego, vol. 18 of *Standard Edition of The Complete Psychological Works*, J. Strachey, 1921.

[4] R. Turner, L. Killian, Collective Behavior, 1957.

[5] O. Se, Using Public Surveillance Systems for Crime Control and Prevention, BiblioBazaar, ISBN 9781288383535, URL `http://books.google.it/books?id=cS8KmQEACAAJ`, 2012.

[6] S. Ali, M. Shah, Floor Fields for Tracking in High Density Crowd Scenes, in: D. A. Forsyth, P. H. S. Torr, A. Zisserman (Eds.), ECCV (2), vol. 5303 of *Lecture Notes in Computer Science*, Springer, 1–14, 2008.

[7] A. Marana, S. Velastin, L. da F. Costa, R. Lotufo, Automatic estimation of crowd occupancy using texture and NN classification, Safety Science 28 (3) (1998) 165–175.

[8] T. Watanabe, S. Ito, K. Yokoi, Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection, in: T. Wada, F. Huang, S. Lin (Eds.), Advances in Image and Video Technol-

ogy, vol. 5414 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 37–47, 2009.

[9] J. Jacques Junior, S. Musse, C. Jung, Crowd Analysis Using Computer Vision Techniques, Signal Processing Magazine, IEEE 27 (5) (2010) 66 –77, ISSN 1053-5888.

[10] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, L.-Q. Xu, Crowd analysis: a survey, Mach. Vision Appl. 19 (5-6) (2008) 345–357, ISSN 0932-8092.

[11] D. Simonnet, S. Velastin, E. Turkbeyler, J. Orwell, Backgroundless detection of pedestrians in cluttered conditions based on monocular images: a review, Computer Vision, IET 6 (6) (2012) 540–550, ISSN 1751-9632.

[12] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, in: Proceedings of the 6th British conference on Machine vision (Vol. 2), BMVC '95, BMVA Press, Surrey, UK, UK, ISBN 0-9521898-2-8, 583–592, 1995.

[13] A. Basharat, A. Gritai, M. Shah, Learning object motion patterns for anomaly detection and improved object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, ISSN 1063-6919, 1 –8, 2008.

[14] S. Calderara, U. Heinemann, A. Prati, R. Cucchiara, N. Tishby, Detecting Anomalies in People Trajectories using Spectral Graph Analysis, Computer Vision and Image Understanding 115 (8) (2011) 1099–1111.

[15] S. Ali, M. Shah, A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis, in: IEEE Conf. on CVPR, 1 –6, 2007.

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

[17] A. Davies, J. H. Yin, S. Velastin, Crowd monitoring using image processing, Electronics Communication Engineering Journal 7 (1) (1995) 37 –47, ISSN 0954-0695.

[18] C. S. Regazzoni, A. Tesei, Distributed data fusion for real-time crowding estimation, Signal Processing 53 (1) (1996) 47 – 63, ISSN 0165-1684.

[19] J. Kim, K. Grauman, Observe locally, Infer Globally: a Space-Time MRF for Detecting Abnormal Activities with Incremental Updates, in: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR), 2921 – 2928, 2009.

[20] T. Hassner, Y. Itcher, O. Kliper-Gross, Violent flows: Real-time detection of violent crowd behavior, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2012), ISSN 2160-7508, 1–6, 2012.

[21] P. Reisman, O. Mano, S. Avidan, A. Shashua, Crowd detection in video sequences, in: IEEE Intelligent Vehicles Symposium, 2004.

[22] W. Ge, R. Collins, R. Ruback, Vision-based Analysis of Small Groups in Pedestrian Crowds, IEEE Trans. on Pattern Anal. and Machine Intel .

[23] M. Rodriguez, I. Laptev, J. Sivic, J.-Y. Audibert, Density-aware person detection and tracking in crowds, in: Proc. of ICCV, 2423 – 2430, 2011.

[24] O. Arandjelovic, Crowd detection from still images, in: British Machine Vision Conference, 2008.

[25] W. Ma, L. Huang, C. Liu, Crowd Density Analysis Using Co-occurrence Texture Features, in: Proc. of Int. Conf. on Computer Sciences and Convergence Information Technology (ICCIT), 170–175, 2010.

[26] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the Support of a High-Dimensional Distribution, Neural Comput. 13 (7) (2001) 1443–1471, ISSN 0899-7667.

[27] B. Fergani, M. Davy, A. Houacine, Speaker diarization using one-class support vector machines, Speech Communication 50 (5) (2008) 355–365, ISSN 0167-6393.

[28] J. Munoz-Mari, F. Bovolo, L. Gomez-Chova, L. Bruzzone, G. Camp-Valls, Semisupervised One-Class Support Vector Machines for Classification of Remote Sensing Data, IEEE Transactions on Geoscience and Remote Sensing 48 (8) (2010) 3188–3197, ISSN 0196-2892.

[29] G. Bradsky, A. Kaehler, Learning OpenCv, O'Reilly, 2008.

[30] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in: In CVPR, 886–893, 2005.

[31] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, in: Proc. of IEEE Conf. on CVPR, vol. I, 511–518, 2001.

[32] A. Marana, L. Costa, R. Lotufo, S. Velastin, On the efficacy of texture analysis for crowd monitoring, in: Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI '98. International Symposium on, 354 –361, 1998.

[33] G.-H. Liu, L. Zhang, Y.-K. Hou, Z.-Y. Li, J.-Y. Yang, Image retrieval based on multi-texton histogram, Pattern Recognition 43 (7) (2010) 2380 – 2389, ISSN 0031-3203.

[34] L. Bertelli, T. Yu, D. Vu, B. Gokturk, Kernelized Structural SVM Learning for Supervised Object Segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2011, 2153–2160, 2011.

[35] T. Joachims, SVM light, http://svmlight.joachims.org, 2002.

[36] T. Joachims, Advances in kernel methods, in: B. Schölkopf, C. J. C. Burges, A. J. Smola (Eds.), Advances in kernel methods, chap. Making large-scale support vector machine learning practical, MIT Press, Cambridge, MA, USA, 169–184, 1999.

[37] A. Lerner, Y. Chrysanthou, D. Lischinski, Crowds by Example, Computer Graphics Forum 26 (3) (2007) 655–664, ISSN 1467-8659.

[38] Pets 2009, PETS 2009, [online] `http://www.cvg.rdg.ac.uk/PETS2009/`, 2009.

[39] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645, ISSN 0162-8828, doi:\bibinfo{doi}{10.1109/TPAMI.2009.167}.

[40] K. Smith, D. Gatica-Perez, J. Odobez, S. Ba, Evaluating Multi-Object Tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, ISSN 1063-6919, 36–36, doi:\bibinfo{doi}{10.1109/CVPR.2005.453}, 2005.

[41] U. of Edinburgh, The BEHAVE Dataset, online, URL `http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/`, 2009.

[42] R. Vezzani, R. Cucchiara, Video Surveillance Online Repository (ViSOR): an integrated framework, Multimedia Tools and Applications 50 (2) (2010) 359–380.