CROWD MOTION ANALYSIS: SEGMENTATION, ANOMALY DETECTION, AND BEHAVIOR CLASSIFICATION

Habib Ullah

UNIVERSITY OF TRENTO - Italy

Department of Information Engineering and Computer Science

Advisor: Nicola Conci, PhD

February 2015

Abstract

The objective of this doctoral study is to develop efficient techniques for flow segmentation, anomaly detection, and behavior classification in crowd scenes. Considering the complexities of occlusion, we focused our study on gathering the motion information at a higher scale, thus not associating it to single objects, but considering the crowd as a single entity. Firstly, we propose methods for flow segmentation based on correlation features, graph cut, Conditional Random Fields (CRF), enthalpy model, and particle mutual influence model. Secondly, methods based on deviant orientation information, Gaussian Mixture Model (GMM), and MLP neural network combined with GoodFeaturesToTrack are proposed to detect two types of anomalies. The first one detects deviant motion of the pedestrians compared to what has been observed beforehand. The second one detects panic situation by adopting the GMM and MLP to learn the behavior of the motion features extracted from a grid of particles and GoodFeaturesToTrack, respectively. Finally, we propose particle-driven and hybrid appraoches to classify the behaviors of crowd in terms of lane, arch/ring, bottleneck, blocking and fountainhead within a region of interest (ROI). For this purpose, the particle-driven approach extracts and fuses spatio-temporal features together. The spatial features represent the density of neighboring particles in the predefined proximity, whereas the temporal features represent the rendering of trajectories traveled by the particles. The hybrid approach exploits a thermal diffusion process combined with an extended variant of the social force model (SFM).

Keywords

[Graph cut, Conditional Random Fields, Optical Flow, Gaussian Mixture Model, Multilayer Perceptron, Spatio-Temporal Features]

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter overviews the research field investigated in this doctoral study. In particular, we describe crowd motion analysis techniques, focusing on segmentation, anomalies detection, and behavior classification. The main objectives and the novel contributions of this thesis are also presented. Finally, we describe the organization of the thesis.

1.1 Overview

According to the report presented by Montgomery [34], more than half of the people of the world live in populated areas. Therefore, automated motion analysis plays an important role in pedestrian flow management and visual surveillance systems. in terms of designing public spaces, visual surveillance systems, and intelligent environments. Applications include the monitoring of pedestrian flows, preventing accidents, as well as implementing evacuation plans necessary in the unlikely event of a fire or in presence of riots in urban areas. In the literature, the research has focused on gathering the motion information at a higher scale, thus not associating it to single objects. These approaches often require low-level features such as multi-resolution histograms [69], spatio-temporal cuboids [23], appearance or motion descriptors [4] [43] and spatio-temporal volumes [25] [6].

Pedestrian flow implies that the the flow can neither be considered as a continuum, nor can its uniform behavior be verified given that individuals are independent, which are key requirements in the existing techniques. For instance, Ali and Shah [2] proposed a Lagrangian Coherent Structure (LCS) approach to segment the flow using the Finite Time Lyapunov Exponent (FTLE) [48], to extract the boundaries between different flow regions in the scene. However, when the optical flow computation is not accurate due to the lack of coherence in motion, the boundaries may be discontinuous. Furthermore, the merge operation based on Lyapunov divergence is mainly suitable for combining adjacent segments, resulting also in this case in over-segmented regions in pedestrian flow scenes. A more recent related work [32] proposed streaklines based on linear dynamical model. However, streaklines are incapable to encapsulate the crowd dynamics, thus failing to group pixels with common motion patterns. In addition, streaklines cannot capture temporal changes, exhibiting choppy motion segmentation in high density crowd scenes.

Moreover, automated motion analysis is also important for designing public spaces and intelligent environments. Real environments often include road networks, pedestrian pathways, and trails. The movement of pedestrians in the aforementioned places is a complex system to study. However, when we consider the environment being very large, all areas of the environment are not equally important. Therefore, a vision-based throttle that relies on the acquired visual data would be desirable in order to improve on the one hand the structure of the environment, for urban design and planning, and on the other hand prevent accidents. For this purpose, Ozturk et al. [38] detect dominant motion flows by exploiting local and global information using SIFT features and Self-Tuning Spectral Clustering [67]. However, SIFT features can be unreliable in representing the characteristic parts of the objects due to redundant information in

the 128-dimensional descriptor [13] [64]. Moreover, the spectral clustering approach fails to simultaneously identify clusters at different scales [35].

Automated detection of anomalous events generated by self-organization phenomena resulting from the unlikely event of a fire or in presence of riots in urban areas, can cause significant hindrance in the flow. This makes necessary to provide more vigilant surveillance, possibly in lieu of, or as an assistance to, human operators. However, there is a lack of empirical studies of crowded scenes where besides basic motion segmentation, also the analysis of more structured behaviors, such as the formation of lanes, or the detection of oscillations at bottlenecks, is decisive for the safety of people during, for example, the access to or exit from mass events, or in situations of emergency evacuation. Congested conditions can possibly trigger crowd disasters arising from the maximum density and irregular flow of crowd. Moreover, the behavior of the crowd may transition from one state of collective behavior to a qualitatively different behavior depending on the density of crowd. Such transitions typically occur when individuals in the crowd accumulate, propagate, or uniformly move with the flow.

1.2 Proposed Solutions

The objective of this doctoral study is to develop efficient techniques for motion segmentation, anomalies detection, and behavior classification considering the complexities of occlusion, foreshortening, and perspective.

Given such requirements, during this doctoral research we contributed in each application scenario proposing the following approaches :

- Motion segmentation;

  In [55], we train a conditional random field (CRF) to segment the motion flow. We first position a grid of particles over the frame and track it using the Lucas-Kanade optical flow. By tracking the particles, we

extract motion patterns, which are used as a-priori information for CRF training. Training is performed by means of the gradient ascent algorithm, so as to maximize the conditional likelihood. Furthermore, the parameters after training are used for CRF to segment the crowd flow in terms of motion directions. In fact, compared to other approaches, such as Hidden Markov Model (HMM), CRF is able to model dense and correlated flow features of crowd since it models the conditional probability allowing relaxation of the strong independence assumptions made by the HMM.

For medium density scenes, we consider intra- and inter-group properties, in [54], representing motion dynamics of pedestrian scenes. Intra-group properties, e.g. slackness and stability, denote internal coordination among members in the same group, whilst inter-group properties, e.g. distributiveness, reflect the external interaction between members in different groups. Groups in the pedestrian flow are represented by slacked individuals lacking firmness. Therefore, we observe that lightly packed pedestrian flows of individuals can be treated as a constituent (block), albeit irregular and inhomogeneous at a coarse scale. This constituent begins to correspond to a harmonic pattern, as is the case of the continuum, at a finer scale. We also observe that, groups of individuals are likely to exhibit an increased level of similarity represented by block-based correlation features based on constituents. Our goal in using correlation features for localized constituents is to estimate recurrent structures in the frames, but with the important distinction that such constituents are not expected to fully contain a person. After analyzing the correlation features, the min cut/max flow algorithm is exploited in order to obtain a regularized representation of the motion field. The inspiration for this algorithm comes from the observation that a pedestrian flow can be

represented as a set of nodes of a graph, where each node corresponds to a constituent of a video frame.

Furthermore, we present a novel method [58] for dominant motion analysis in crowded scenes, based on corner features. For this purpose, we extract the corner features from a video frame and track them using the Lucas-Kanade optical flow. These features are then analyzed through an enthalpy model returning a subset of features of potential interest. Subsequently, we extract orientation information from the corner features and train a random forest to learn the behavior of the crowd, in order to detect dominant motion flows. In fact, random forests deliver a higher level of predictive accuracy automatically, resist to overfitting, diagnose pinpoint multivariate outliers, and exhibit invariance to monotone transformations of variables.

In [46], we detect and track moving entities in wide surveillance videos. Considering the wide area covered by the camera, which makes the detection and tracking of humans, as well as the classification of their motion a complex task and resource consuming, we adopt a particle-based approach to highlight particles of interest and group them based on their motion properties. A cross influence matrix is computed at the particle level identifying the relevant areas of the video, and pruning static particles and outliers. Based on the motion features of the particles marked as interacting with their neighbors, a learning procedure based on an MLP neural network is implemented, in order to create consistent groups, representing the moving entities to be tracked over time.

- Anomaly detection;

  On top of motion segmentation, we investigated an anomaly detection strategy [54], by highlighting deviant motion of the pedestrians com-

pared to what has been observed beforehand. Once the motion flow is extracted from the foreground, an accumulator is constructed on top of each block to create the pedestrian motion model, by collecting evidence regarding the dominant directions of pedestrian motion. The accumulator is updated at every frame, keeping up with the evolution of the pedestrian flow. The pedestrian motion model combined with the output of multi-label optimization and orientation information is exploited to detect anomalies.

In [57], we detect anomaly in term of panic situation. For this purpose, we adopt Gaussian Mixture model (GMM) to learn the behavior of motion features extracted from a grid of particles instead of modeling the values of all the pixels as a mixture of Gaussians. These motion features are exploited to learn repetitive variations of crowd scenes for GMM, which models the normal behavior distribution. If each particle resulted from a particular behavior, a single Gaussian would be sufficient to model the motion feature of it, while accounting for surrounding noise. However, in practice, multiple surfaces often appear in the view frustum of a particular particle. Therefore we use multiple adaptive Gaussians to approximate this process. At each frame the parameters of the Gaussians are updated, and the Gaussians are evaluated using a simple heuristic to hypothesize, which are most likely to be part of the distribution representing the normal crowd behavior.

To consolidate the anomaly detection in term of panic situation, we present a method [59] that adopts multi-layer perceptron (MLP) feed-forward neural network to learn the behavior of motion features extracted from the corner features instead of considering the values of all the pixels. The motion features are exploited to learn the abrupt changes of crowd scenes represented by corner features, thus modeling the abnormal behavior of the crowd. A single motion feature

extracted from an arbitrary corner feature is not sufficient to model the abnormal behavior of crowd due to surrounding noise. Therefore, for each corner feature we extract a set of motion features to robustly model the abnormal behavior of the crowd.

- Behavior classification;

  We identify crowd behaviors in real-time using a particle-driven approach [56]. We focus on three types of behaviors, namely lanes, arches, and bottlenecks. The method exploits a grid of particles uniformly distributed on the video frame, and advected over a temporal window through optical flow tracking. Approximating the moving particles to individuals, spatio-temporal features are extracted at the end of the temporal window for each particle within a region of interest (ROI). The temporal features represent the rendering of trajectories traveled by the particles, whereas the spatial features represent the density of neighboring particles in the predefined proximity. The two features are fused together to model the behavior of the crowd in low to medium density crowd. Furthermore, the feature extraction process is computationally affordable, thus suitable to be applied in real-time applications for behavior analysis in crowded scenes.

  We also present a novel method [60] for crowd behaviors classification within a region of interest (ROI) taking inspiration from dynamic systems. In our method, a motion flow field is obtained from video frames using dense optical flow technique. Then a thermal diffusion process is exploited to turn the motion flow field into a more coherent motion field. Approximating the moving particles to individuals, their interaction forces, represented as force flow, are computed using an extended variant of social force model (E-SFM) to obtain potential particles of interest. Apart from capturing the effect of neighboring in-

dividuals on each other, the E-SFM also takes into account the crowd turbulence usually triggered by regions of high interactions. The approach presents significant performance irrespective of the density of the crowd.

Table 1.1, summarizes our proposed methods covered in this section in terms of analysis and features used for motion segmentation, anomaly detection, and behavior classification.

1.3 Thesis Structure

The thesis is organized in 5 chapters where each Chapter begins with the corresponding state of the art. In Chapter 2, the details of our proposed approaches regarding pedestrian flow segmentation are presented and discussed. In Chapter 3 and Chapter 4, the details of our proposed approaches regarding anomaly detection and behavior classification are presented and discussed, respectively. Moreover, Chapter 5 collects some concluding remarks.

Table 1.1: Summary of our proposed methods for motion segmentation, anomly detection and behavior classification. Methods dealing with any of the three problems are marked Y (Yes) in the corresponding column.

Ref.	**Analysis level**	**Features**	**Motion segmentation**	**Anomaly detection**	**Behavior classification**
[54]	Medium density	Block-based correlation Graph cut	Y	Y	-
[55]	High density	CRF Graph cut	Y	-	-
[46]	High density	Influence matrix MLP	Y	-	-
[58]	High density	Enthalpy measure Random forests	Y	-	-
[57]	High density	GMM	-	Y	-
[59]	High density	Corner features MLP	-	Y	-
[56]	Medium density	Spatio-temporal features	-	-	Y
[60]	High density	TDP E-SFM Dynamic system	-	-	Y

Chapter 2

Motion Segmentation

This chapter begins with the state of the art regarding motion segmentation and then presents our proposed methods. In particular, the techniques based on block-based correlation, graph cut, and conditional random fields (CRF) are presented. Subsequently, methods for analyzing dominant flows and tracking moving entities based on particle influence model in crowded scenes are presented, respectively.

2.1 State of The Art

The literature in crowd motion analysis is becoming rich, and an overview about earlier algorithms in the area and related issues are presented by Jacques et al. [22] and Zhang et al. [68]. In fact, activity analysis and scene understanding entail object detection, tracking and activity recognition. These approaches, requiring low-level motion features [61], appearance features [31], or object trajectories [52], render good performance in low to medium density crowd scenes. However, for higher density scenes, the research has focused on gathering the motion information at a higher scale, thus not associating it to single objects, but considering the crowd as a single entity. These approaches often require low-level features such as multi-resolution histograms [62] [69], spatio-temporal cuboids [23], appearance or motion

descriptors [4] [43] and spatio-temporal volumes [23] [53] [37] [6].

We divide state of the art into three categories based on the density of the flow considered. For example, methods targeting a single individual are under *individual level analysis*, methods targeting two to five individuals are under *low density flow analysis*, and methods targeting more than five individuals are grouped under the term *high density flow analysis*. Methods that rely on individual level analysis and low density flow analysis try to segment individual objects or group of objects in a scene, respectively. These methods tend to produce more accurate results in scenes with a limited number of moving entities. In pedestrian scenes, however, clutter and severe occlusions make the individual or group segmentation an extremely challenging task. In contrast to that, high density flow analysis methods treat the entire scene as a single entity, and usually capable of obtaining coarser-level information, such as the identification of the main flow, disregarding local and finer information.

The methods proposed by Bai and Sapiro [5], Cremers and Soatto [15], and Paragios and Deriche [40], for objects segmentation, fall under individual level analysis. Bai and Sapiro [5] exploit geodesic transforms to encourage spatial regularization and contrast-sensitivity for image and video segmentation. The method assumes given user strokes and imposes an implicit connectivity prior, which forces each region to be connected to one stroke. In the work by Cremers and Soatto [15], the optical flow constraint is exploited to estimate a conditional probability of the spatio-temporal intensity change. Furthermore, motion estimation and segmentation are integrated into a functional minimization strategy based on a Bayesian framework. A mixture model is exploited by Paragios and Deriche [40] to represent the inter-frame difference. The mixture model comprises of two components corresponding to the foreground and background.

In low density flow analysis, Cheriyadat and Radke [14], Chan and Vas-

concelos [11], and Cisar and Kembhavi [29], segment groups of objects. Cheriyadat and Radke [14] exploited low-level features using optical flow, in order to segment or track the dominant motion in the scene. For this purpose, trajectories are clustered based on a distance measure. Chan and Vasconcelos [11] used a mixture of dynamic textures to fit a video sequence and then assigned homogeneous motion regions to the mixture components. Cisar and Kembhavi [29] perform motion segmentation without relying on the optical flow. For this purpose, they exploit a dynamic texture model to measure the similarity between neighboring spatio-temporal patches. These patches are grouped by connected component analysis, resulting into over segmentation in the presence of pedestrian flow, since patches corresponding to individuals moving homogeneously may not be connected.

The approaches proposed by Ali and Shah [2] and Mehran et al. [32] fall under high density flow analysis for motion segmentation. Ali and Shah [2] proposed a Lagrangian Coherent Structure (LCS) approach to segment the flow using the Finite Time Lyapunov Exponent (FTLE) [48], to extract the boundaries between different flow regions in the scene. However, when the optical flow computation is not accurate due to the lack of coherence in motion, the boundaries may be discontinuous. Furthermore, the merge operation based on Lyapunov divergence is mainly suitable for combining adjacent segments, resulting also in this case in over-segmented regions in pedestrian flow scenes. A more recent work proposed by Mehran et al. [32] exploit *streaklines*. *Streaklines* are vector field representations of the flow and are represented through a linear dynamical model. *Streaklines* [32] cannot capture temporal changes, exhibiting choppy motion segmentation in low-density and medium-density crowd scenes. Additionally, both approaches [2] [32] are oriented toward crowd coherency. Thus both methods become unreliable when coherency changes with the density of crowd.

2.2 CRF With Graph Cut

The method we present is modeled in three main stages namely: particle advection, CRF inferencing, and refinement of the motion map using graph cut. During the first stage, a grid of particles is disposed on the video frame. Each particle represents a block of pixels of predefined size. Motion patterns, defined in terms of orientation features, are extracted by tracking the particles using the pyramidal Lucas-Kanade optical flow [66]. During this first step, the orientation features act as a sequential data for inferencing the CRF, resulting into a motion map. The orientations features with the corresponding label sequence are used to learn the CRF parameters during the training stage, and the crowd motion directions are inferred on the test samples. In order to provide a more coherent representation of the crowd motion in the second step, graph cut [9] is used to filter out the residual noise.

2.2.1 Inferencing

After disposing the grid of particles over the video frame, and tracking it by the Lucas-Kanade optical flow, the orientation features of each particle in term of angle of motion are extracted at regular intervals of K frames. The collected orientation features are stored to build a feature vector for each particle. The target of this processing step is to remove and filter out the orientation features that would be possible if considered singularly, but that do not contribute to the identification of the crowd motion direction.

A Conditional Random Field (CRF) is a discriminative model used for labeling sequential data. It provides the probability of a particular label sequence, given the observation sequence. Specifically, $\mathbf{x}$ is our input sequence, consisting in N observations collected within the K frames window (i.e. $\mathbf{x} = x_1, x_2, \ldots, x_N$), containing the orientation features. Given the

observation sequence, the CRF thus signals the most probable label in terms of direction, inferring the output label $\mathbf{y}$ ($\mathbf{y} = y_1, y_2, \ldots, y_M$) of the respective crowd motion direction, and quantized in M possible values.

$$p\left(y/\mathbf{x};\mathbf{w}\right) = \frac{exp\sum_j w_j F_j(\mathbf{x},y)}{Z(\mathbf{x},\mathbf{w})} \tag{2.1}$$

In Eq. (2.1), $F_j(\mathbf{x},y)$ is a feature function, which consists of the paired mapping $F_j : X * Y \rightarrow \Re$. Each feature function renders the score for any output label y in terms of its relevance to the input observation vector $\mathbf{x}$. The flow of inference process is shown in Fig. 2.1 where N represents the total number of particles tracked. The denominator in Eq. (2.1) is a partition function $Z(\mathbf{x},\mathbf{w})$, which ranges over all the label set $\mathbf{y}$.

$$Z\left(\mathbf{x},\mathbf{w}\right) = \sum_{y'} exp\left\{\sum_j w_j F_j(\mathbf{x},y')\right\} \tag{2.2}$$

Hence, the partition function acts as a normalization factor. Given orientation features $\mathbf{x}$, the corresponding label is obtained as:

$$\hat{\mathrm{y}} = argmax_y p\left(y/\mathbf{x};\mathbf{w}\right) = argmax_y \sum_j w_j F_j(\mathbf{x},y) \tag{2.3}$$

For each j, we will obtain different F_j functions, according to the parameter w_j and the test observation sequence $\mathbf{x}$. Our main contention in obtaining the probability score for each label sequence is that it is easy to reveal the most probable direction for each particle, which can segment the crowd motion as the scene dynamically changes over time.

2.2.2 Training

The goal of the training stage is to identify the appropriate values for the parameters w_j, so as to maximize the conditional probability of the training examples. For this purpose we use the stochastic gradient ascent

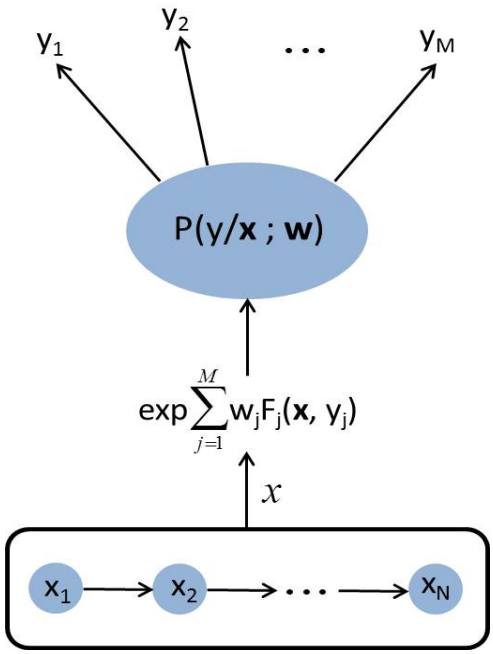


Figure 2.1: CRF Inference

to maximize the conditional log-likelihood (CLL) of the set of training examples:

$$\frac{\partial}{\partial w_j} \log p\left(y/\mathbf{x}; \mathbf{w}\right) = F_j(\mathbf{x}, y) - \frac{\partial}{\partial w_j} \log Z\left(\mathbf{x}, \mathbf{w}\right) \tag{2.4}$$

For each w_j, the partial derivative of CLL is evaluated for single training sequences, i.e., one weight for each feature function F_j. The partial derivative with respect to w_j corresponds to the i-th value of the feature function for its true label y, minus the averaged values of the feature function for all possible labels $\mathbf{y}$. Therefore, Eq. (2.4) can be rewritten as:

$$\frac{\partial}{\partial w_j} \log p\left(y/\mathbf{x}; \mathbf{w}\right) = F_j(\mathbf{x}, y) - \sum_{y'} p\left(y'/\mathbf{x}; \mathbf{w}\right) \left[F_j(\mathbf{x}, y')\right] \tag{2.5}$$

In order to maximize the conditional log-likelihood by stochastic gradient ascent, w_j is updated according to Eq. (2.6) where α is the learning

rate.

$$w_j = w_j + \alpha(F_j(\mathbf{x}, y) - \sum_{y'} p\left(y'/\mathbf{x}; \mathbf{w}\right)\left[F_j(\mathbf{x}, y')\right]) \tag{2.6}$$

2.2.3 Sanitizing the motion map

Although the output of the CRF inference is in general quite accurate in indicating the motion flow, it still includes a non negligible amount of noise. In order to remove this noise and to better present the main motion directions of the crowd flow, we exploited the α-expansion moves based on graph cuts [9], which produce a solution within a known factor of the global minimum of the energy function. The minimization process takes place according to Eq. (2.7)

$$E\left(L\right) = \sum_{p \epsilon P} D_p\left(L_p\right) + \sum_{(p,q) \epsilon N} V_{p,q}(L_p, L_q) \tag{2.7}$$

where D_p is the so-called *data cost* term, and $V_{p,q}$ is the *smooth cost* term. The α-expansion minimizes the energy function for a set of labels under the class of smoothness term, called metric. We exploited both the data cost term and the smooth cost terms so that the resulting labeling fit to the data and accomplishes the desired smoothing. Fig. 2.2 presents the effectiveness of the α-expansion moves. Further detail of this process is provided in Section 2.3.2.

As shown in Fig. 2.2, the α-expansion moves demonstrate a very good capability in suppressing the residual noise left by the preceding processing stages.

2.2.4 Experimental results

In this section we present the results of our approach. Experiments are carried out on benchmark video sequences [2] [54] [32] to thoroughly eval-

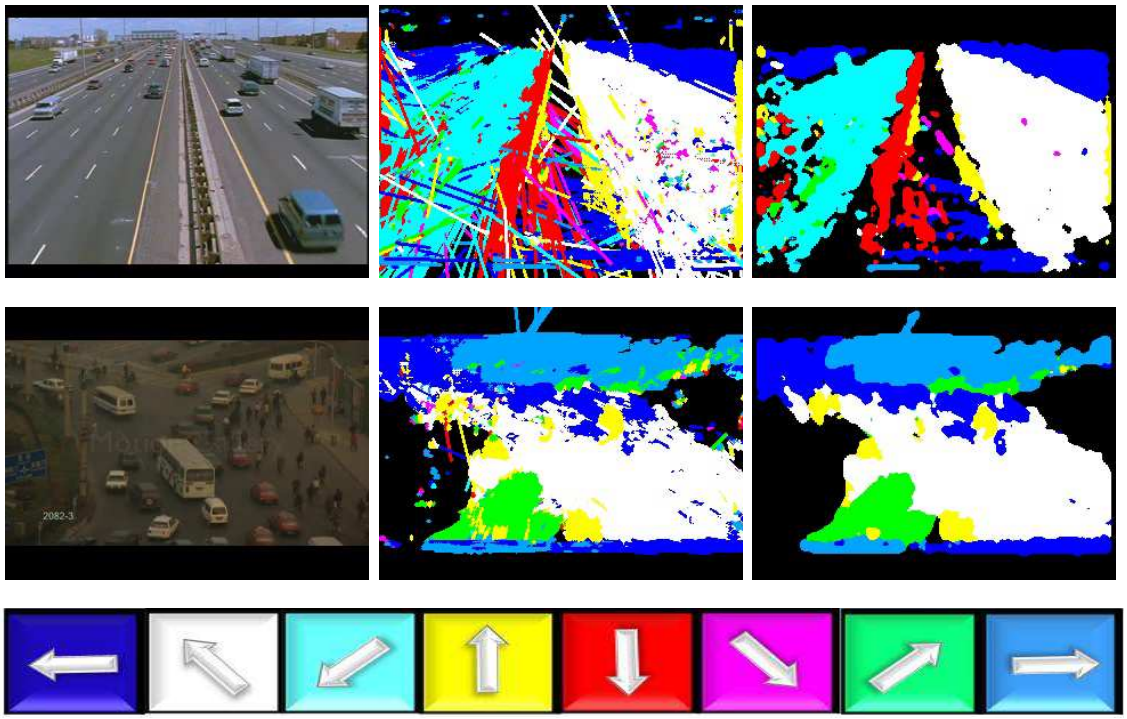


Figure 2.2: Input frames (first column), CRF segmentation (second column), and refinement using graph cut (third column).

uate the effectiveness of our proposed approach. In Fig. 2.3, the first rows present the snapshots of the original video sequences, while the second, third, fourth and fifth rows show the results obtained using (i) pure optical flow, (ii) the method in [54], (iii) streaklines approach [32], and (iv) the proposed approach, respectively.

To neglect regions without motion, we discard small magnitude optical flow. For the extraction of the orientation features for each particle, the resolution of the grid is kept half of the resolution of the video frame. For each particle, the orientation features consist of a vector of $N = 4$ observations, where each element of the vector corresponds to the orientation information extracted after each $K = 8$ frames. The possible output directions are $M = 8$, one label every $45°$. When applying the graph cut, each frame processed by the CRF is divided into blocks 2×2 pixels. Each block is considered as a single element and scanning is carried out from top-left to bottom-right. For each central block, the spatial neighborhood is set to 5×5 blocks. For the training phase, we used 800 samples. Each training sample is selected randomly, so that the trained model reflects a relevant and accurate representation of the training data.

We track the particles for 8 consecutive frames by using the Lucas-Kanade optical flow. Then, the obtained *tracklets* are drawn according to the selected eight possible output directions. It is evident from the segmentation map, that the simple optical flow representation is not powerful enough to segment the crowd motion. Also, when comparing with the method presented in [54], we can notice inconsistencies in the crowd motion in Fig. 2.3, and this is evident especially in the first and third video sequences in the first row, where the crowd is moving in semi-circle direction.

In Fig. 2.3, streaklines [32] exhibits choppy motion segmentation, whereas our approach is more consistent. Furthermore, the approach in [54] presents

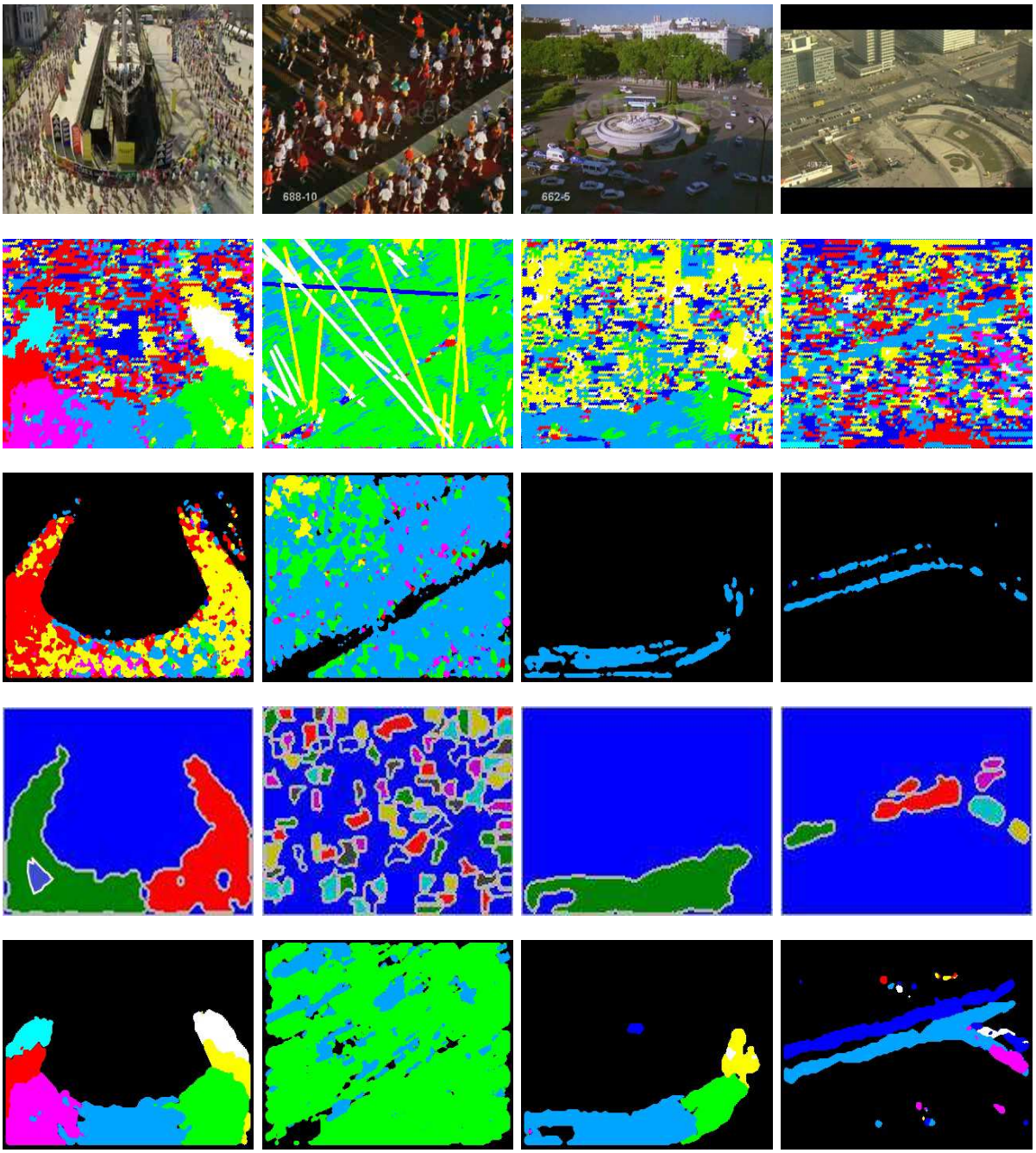


Figure 2.3: Segmentation. Frames from video sequences (first row); pure optical flow (second row), correlation [54](third row), streaklines [32] (fourth row), and our approach (last row)

.

quite unreliable results for all video sequences comparing to our approach. We are able to outperform the three methods thanks to the CRF inference, mainly aimed at learning the temporal evolution of the crowd motion, and the graph cut, which consolidates the output in the spatial dimension.

2.3 Block-Based Correlation

We characterize pedestrian flow segmentation from the computer vision point of view by considering the pedestrian flow beyond just a collection of spatially proximate individuals, but also as a dynamic unit that exhibits various properties. In our approach, we have selected different directions of motion where each direction is represented by a label. We represent the input video, foreground frame, orientation information, motion segmentation, and anomaly detection with the symbols V(t), F(t), O(t), S(t), and A(t), respectively. The motion segmentation is obtained by operating on the union of the correlation information with the orientation information, as formulated in Eq. 2.8.

$$S(t) = B(t) \cup O(t) \tag{2.8}$$

An overview of the whole processing chain is shown in Fig. 2.4. The proposed method operates on the foreground region. Therefore, foreground is first extracted from each input frame of the video sequence through the Gaussian mixture model [51]. This is represented as change detection in the first box on the left side of Fig. 2.4. We then correlate the information of the pedestrian flow by applying a block-based correlation technique in the spatio-temporal domain, returning the preliminary segmentation map $B(t)$. We also apply a multi-label optimization technique to reduce the noise introduced in the flow output, so as to obtain a regularized representation of the motion field, and highlighting only the most relevant

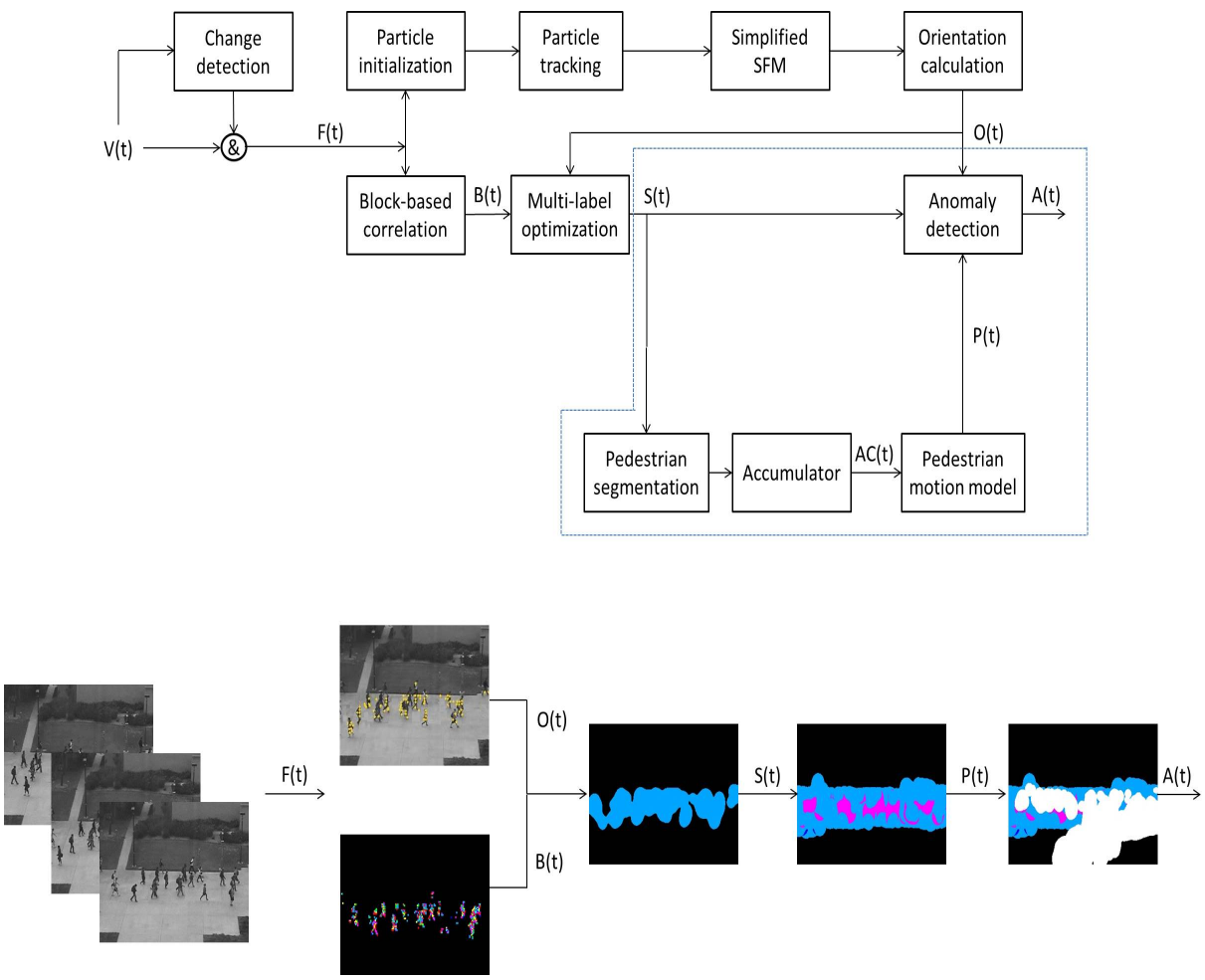


Figure 2.4: Block diagram of the proposed approach representing the pedestrian flow segmentation and anomaly detection. The output of the major processing stages are indicated using symbolic notations and the output is shown in the lower part of the figure with the help of pictures. For example, $V(t)$, $F(t)$, $B(t)$, $S(t)$, $P(t)$, $AC(t)$, and $A(t)$ represent the input video, foreground map, block-based correlation, segmentation, pedestrian motion model, accumulator, and anomaly detection, respectively.

orientations that characterize the pedestrian motion. However, since the multi-label optimization might still contain a residual noise, we integrate the orientation information of the particles, in order to consolidate the output and provide a more consistent representation of pedestrian motion $S(t)$ as shown in Fig. 2.4. To this aim, we initialize a grid of particles on the foreground region. Each particle represents the position of a pixel, and is tracked using the pyramidal Lucas-Kanade optical flow [66]. In order to identify only the particles that exhibit a relevant motion, we exploit a *simplified* variant of the Social Force Model (SFM) [33]. The SFM describes the motion of particles as if they are subject to social forces. Therefore, the model is able to discard the noise-driven particles, as shown in Fig. 2.5. Each block may be overlapped with an arbitrary number of particles since these particles are not directly associated with blocks. The direction information of the particles is then integrated with the multi-label optimization technique in order to provide a more consistent representation of the pedestrian motion, as detailed in the next sections. In fact, a pedestrian flow can be represented as a set of nodes of a graph, where each node corresponds to a region (block) of the video frame. Considering that the characterization of nodes is relatively simple, we transform the problem of pedestrian flow segmentation into a problem of graph-based optimization.

In next sections, we provide the details regarding the steps of the proposed algorithm for motion segmentation, namely block-based correlation, multi-label optimization, and *simplified* social force model.

2.3.1 Block-based correlation

In order to compute the correlation, the image acquired by the camera is first divided into blocks of fixed size. For each block, correlation is computed among successive frames by comparing the current block with the 8-connected blocks in the previous frame so as to find the best match that

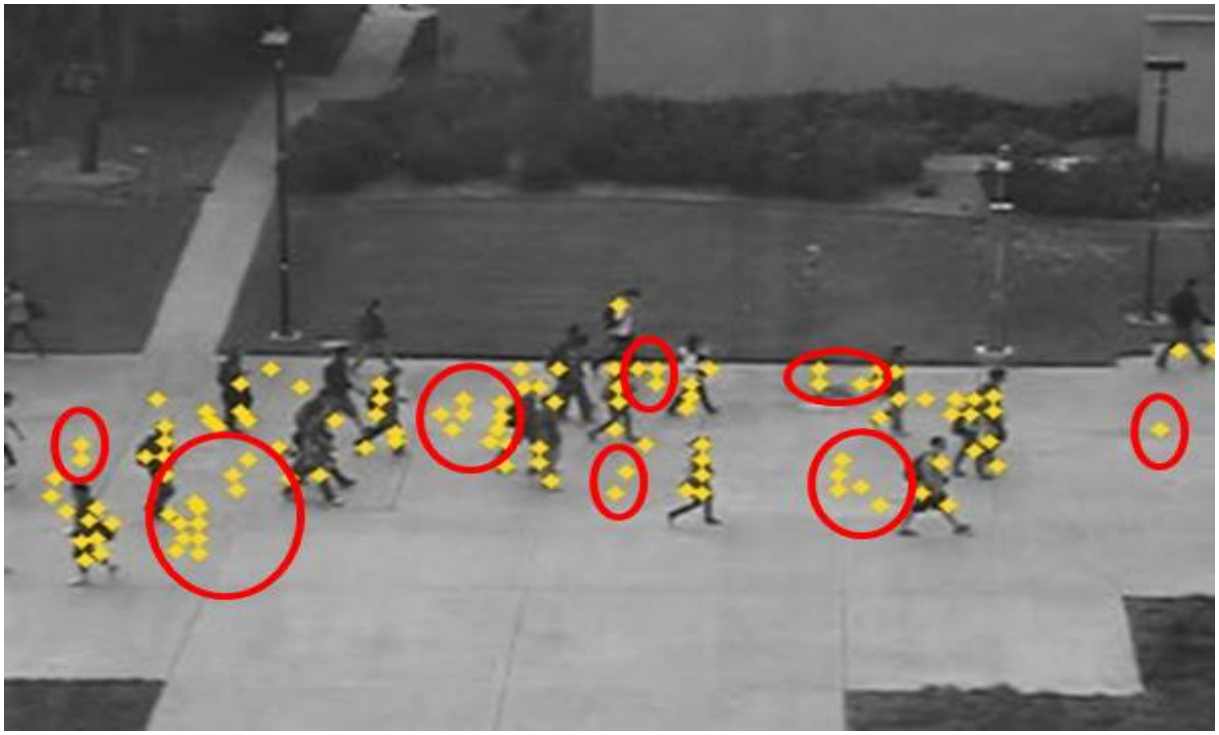


Figure 2.5: Noise driven particles, annotated with circles.

describes the most likely displacement across two successive time instants. Instead of a pixel-based approach, our choice for the block-based approach is motivated by the fact that it is robust to illumination variations and dynamic background [44].

In order to efficiently exploit the correlation information, an accumulator is implemented to store the evolution of each block over time. The comparison of the reference block with each neighboring block in the previous frame is computed on a pixel basis, and formulated according to Eq. (2.9):

$$C_{block} = \sum_{i,j} \frac{1}{1 + |p_t(i,j) - p_{t-1}(i,j)|} \tag{2.9}$$

where $p_t(i,j)$ and $p_{t-1}(i,j)$ represent the gray scale value of the pixel in the reference block and in the neighboring block, at the coordinates (i,j),

respectively. For each block, the dominant direction is stored in terms of absolute angle as formulated in Eq. (2.10):

$$\Psi_b(\theta) = \sum_{blocks=b}^{B} \sum_{directions=i}^{D} [\theta == i], \tag{2.10}$$
$$Where \qquad \forall b \in \{1, \ldots, B\},$$
$$\forall i \in \{1, \ldots, D\}$$

where B and D represent the number of blocks in a frame and possible directions, respectively. The correlation information about the motion direction will be used as input for the pedestrian flow segmentation based on graph-cut, as will be described in Section 2.3.2.

Our choice of implementing block-based correlation to extract a preliminary motion map, is preferred to particle advection through optical flow (as in [2] [32]), since the latter might not be appropriate for pedestrian scenes where the background is by definition dynamic, and in which clutter and complicated occlusions often occur. Moreover, optical flow techniques do not provide accurate measures of motion. On the contrary, when observed at block level, pixel intensities in blocks show strong correlation across consecutive frames, thus giving the opportunity to better highlight motion patterns in the pedestrian flow.

2.3.2 Multi-label optimization

In order to reduce the noise introduced in the flow output, we adopt a procedure based on graph cut. The distinguishing feature of graph cut, compared to LCS [2] and *streaklines* [32], is in the adoption of energy minimization for accurate segmentation in local regions with complex motion patterns. In this procedure, we have followed the implementation of the

min cut/max flow algorithm proposed by Boykov et al. in [8]. In particular, due to the segmentation problem based on orientations, we used the α-expansion moves based on graph cuts [9]. The α-expansion moves are formulated in terms of energy minimization process according to Eq. (2.11), where D_p is the so-called *data cost* term, and $V_{p,q}$ is the *smooth cost* term.

$$E(L) = \sum_{p \epsilon M} D_p(L_p) + \sum_{(p,q) \epsilon N} V_{p,q}(L_p, L_q), \tag{2.11}$$

In this process, the objective is finding the label that minimizes the energy in Eq. (2.11). The data cost represents the appropriateness of a label for the pixel p given the observed data, whereas the smooth cost represents the extent to which labeling is not piecewise smooth. In Eq. (2.11), M and N are the sets of interacting pairs of pixels denoted by p and q, L represents the label, and L_p and L_q are the labels associated to pixel p and pixel q, respectively.

We show the performance of the α-expansion moves with the help of an example. Considering a sample situation with eight labels (from 0 to 7) as input, the output of the algorithm is shown in Fig. 2.6. As can be seen, 0 and 6 are absorbed by 7 because of the short distance in terms of minimization of the energy function.

Input					Output				
7	7	7	7	7	7	7	7	7	7
7	0	0	0	7	7	7	7	7	7
3	3	0	6	6	2	2	7	7	7
2	2	1	1	1	2	2	1	1	1
5	5	5	5	5	5	5	5	5	5

Figure 2.6: Application of α-expansion.

In our implementation, we exploited both the data cost term and the smooth cost term so that the resulting labeling fits the data and accomplishes the desired smoothing. Each block is labeled in the frame according to the angle information, meaning that each label represents a different motion direction. Without loss of generality, and similarly to other approaches proposed in literature, we have selected $W = 8$ different directions quantized with a step of 45 degrees. However, any arbitrary number of elements can be chosen. The data cost assigns different weights to each motion direction extracted by block correlation according to the distance between them. The higher the distance, the higher the data cost. The angle is then compared with the 8 angles of our label set, searching in a neighborhood window of 5x5 blocks.

Given W labels we calculate the minimum distance R between labels as in Eq. (2.12).

$$R = \frac{360}{W} \tag{2.12}$$

The data cost is then computed for each node according to Eq. (2.13).

$$D\left(\theta_l\right) = min\left(\left|\frac{\theta_l}{R} - \frac{\theta}{R}\right|, N - \left|\frac{\theta_l}{R} - \frac{\theta}{R}\right|\right). \tag{2.13}$$

In Eq. (2.13), θ_l is the angle (motion direction) in our label set, and θ is the current angle computed as discussed in Section 2.3.1. The angle θ is then compared with all the angles in the label set and the one minimizing the energy function (Eq. (2.11)) is chosen. Furthermore, the smooth cost term is calculated according to Eq. (2.14).

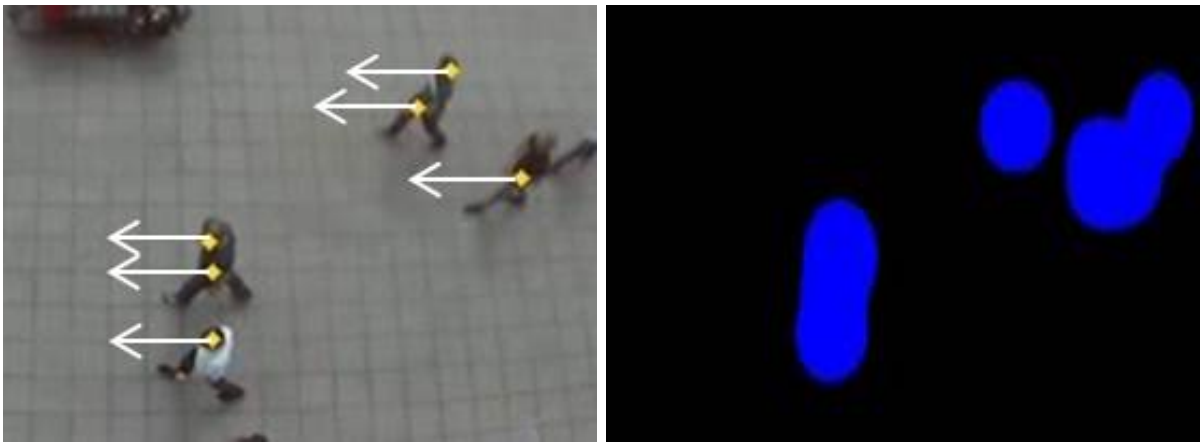
$$V\left(\theta_l, \theta_{l-1}\right) = \left|\left(\theta_l - \theta_{l-1}\right)\right| \tag{2.14}$$

The inspiration for this algorithm comes from the observation that pedestrian motion in a local region is generally simple and can be closely resembled using a graph-based optimization method. Furthermore, the

energy minimization of graph cut is an effective way to fuse similar motion regions, thus limiting the effect of over-segmentation in pedestrian flows. Compared to the state of the art works presented by Ali and Shah [2] and Mehran et al. [32], this representation introduces some important benefits. For example, FTLE [2] identifies LCS as ridges in the pedestrian scenes. These ridges correspond to the boundaries segmenting the flow. All the particles within each region are considered as showing the same behavior. Following a similar paradigm, *streaklines* [32] are defined as the loci of particles that have earlier passed through a prescribed point. *Streaklines* are clustered on the basis of their similarity, to identify segments of the video with similar motion. However, both the boundaries of LCS [2] and *streaklines* [32] are delineated to combine adjacent segments of the dense scene, often resulting in over segmentation of pedestrian flows.

2.3.3 Simplified social force model

Although the α-expansion approach results in a very coherent representation of the major directions reflecting the pedestrian motion behavior, some noise may still be persistent in the segmentation map. In order to consolidate the α-expansion output, we introduced an additional source of information, by initializing a grid of particles on the foreground map. Particles are uniformly spread over the video and tracked over a fixed temporal window using the Lucas-Kanade optical flow technique. However, some of these particles are noisy. Therefore, we exploit the Social Force Model [33], which describes the behavior of the pedestrians based on motion dynamics resulting from the interaction of individuals. This model reflects that an individual keeps a certain distance to other individuals and borders, avoids obstacles and intends to achieve the desired velocity of motion. In this context, the motion of particles is described as if they are subject to social forces, thus providing a mechanism to discard noise-



(a) (b)

Figure 2.7: Movement of the particles from right to left (a) result of the α-expansion is shown (b) since particles are moving in the same direction.

driven particles, because they do not satisfy these requirements. Therefore, a set of potential particles are extracted by exploiting a *simplified* variant of the Social Force Model (SFM) [33], which measures the internal motivations of the individual particle to perform certain movements, and take into account the influence of the other particle surrounding it.

According to the SFM, the velocity of each particle k with mass m_k obeys to Eq. (2.15).

$$m_k \frac{dv_k}{dt} = SF_k = SF_{p,k} + SF_{int,k} \tag{2.15}$$

where SF_k is a combination of the personal desire force $SF_{p,k}$ and the interaction force term $SF_{int,k}$. Considering that each particle in the SFM is treated as an individual in the pedestrian flow, it is assumed that each particle pursues certain goals. Therefore, the personal force of a particle is formulated according to Eq. (2.16).

$$SF_{p,k} = \frac{1}{\lambda}|v_k^p - v_k| \tag{2.16}$$

where λ is the relaxation parameter, v_k^p is the desired velocity, and v_k is the actual velocity of the particle. The desired velocity v_k^p is calculated using the Euclidean distance where the initial position and final position of the particle k are considered. The actual velocity v_k represents the average velocity calculated over T observations in a fixed temporal window.

The interaction force $SF_{int,k}$ consists of the repulsive force SF_{rep} (to ensure a certain distance between particles) and an environment force SF_{env}, to avoid obstacles. In our case, however, we seek to extract potential particles associated to pedestrian motion instead of detecting panic behaviors of the dense crowd [33]. Therefore, we formulate the interaction force $SF_{int,k}$ of a particle k according to Eq. (2.17).

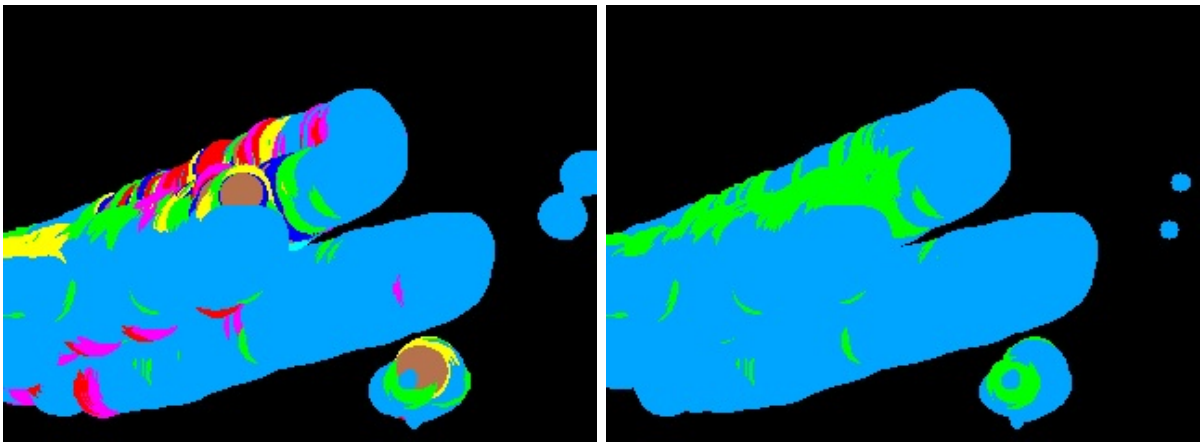
$$SF_{int,k} = \langle v_k \rangle \tag{2.17}$$

where $\langle v_k \rangle$ is the average velocity calculated over a fixed spatio-temporal window. The size of the spatial window for the neighboring particles is currently set to 3×3. Further details of the SFM are not in the interest of this paper and can be found in relevant citations in [19] [20] [33] for a more comprehensive discussion. To this end, the *simplified* social force model can be summarized as in Eq. (2.18):

$$m_k \frac{dv_k}{dt} = SF_k = \frac{1}{\lambda}|v_k^p - v_k| + \langle v_k \rangle. \tag{2.18}$$

In our model we set both the relaxation parameter λ and mass m_k of a particle k to 1 since all particles can be assumed of the same size.

According to the output of the *simplified* SFM, the output returned by the α-expansion is accepted only if a number of particles (defined a priori) is moving in the same direction as shown in Fig. 2.7, or otherwise the incoming block maintains the previous orientation. In fact, driving a set of potential particles, validated by the *simplified* SFM, can contribute to a



(a) (b)

Figure 2.8: Segmentation output before (a) and after (b) particle advection representing significant improvement in the performance.

considerable improvement to the pedestrian flow segmentation, as can be seen in Fig. 2.8.

2.3.4 Experimental results

To validate the performance of our approach, we have conducted the experiments on benchmark datasets such as PETS2009 [42] and UCSD [30]. We have also tested the method on video sequences from our UCD dataset [54]. In the PETS2009 dataset, people are moving from bottom right to left and left to bottom right in the video sequences S1 and S2, and video sequences S3 and S4, respectively. The UCSD dataset is acquired with a stationary camera overlooking pedestrian walkways. The video sequences contain the circulation of pedestrians, bikers, skaters, small carts, and people walking across a walkway or in the grass that surrounds it. In the UCSD dataset, the majority of people are moving from left to right in the video sequences S5, S6, S7, and S8. Our UCD dataset contains four video sequences (S10 to S13) representing flows of students moving outdoor across two buildings. The duration of our dataset is long enough to collect evidence over time

Table 2.1: Summary of the video sequences, in terms of average number of objects, frames per second, number of frames, and resolution, from the UCD dataset.

Video sequences	Avg. objects	FPS	No. of frames	Resolution
S10	15.93	29	1422	320x240
S11	9.37	29	870	
S12	5.02	29	1067	
S13	5.17	29	933	

for both flow segmentation and anomaly detection, as compared to other benchmark datasets lasting only a few seconds. The details of the video sequences are reported in Table 2.1. To evaluate the motion segmentation performance of our approach, we compared it with the state of the art recently proposed by Ali and Shah [2] and Mehran et al. [32].

Fig. 2.9 shows the results of the flow segmentation, where video frames are overlaid by colored regions resulting from the segmentation. The column (a) presents the sample frames taken from the original video sequences, while the central columns (b) to (d) illustrate the results obtained using the Lagrangian method presented by Ali and Shah [2], the *streaklines* method presented by Mehran et al. [32], and the proposed approach, respectively. It is worth noting that the reference approaches [2] [32] in column (b) and (c) , and our method in column (d), use different colors for labeling, therefore results should be interpreted in terms of the segmentation quality, regardless of the color used for visualization. For conciseness reasons only two video sequences from each of the three datasets are depicted in Fig. 2.9. The first two rows show the results for two video sequences taken from PETS2009 [42], where people are moving from the bottom right corner to left, and left to bottom right, respectively. The third and fourth

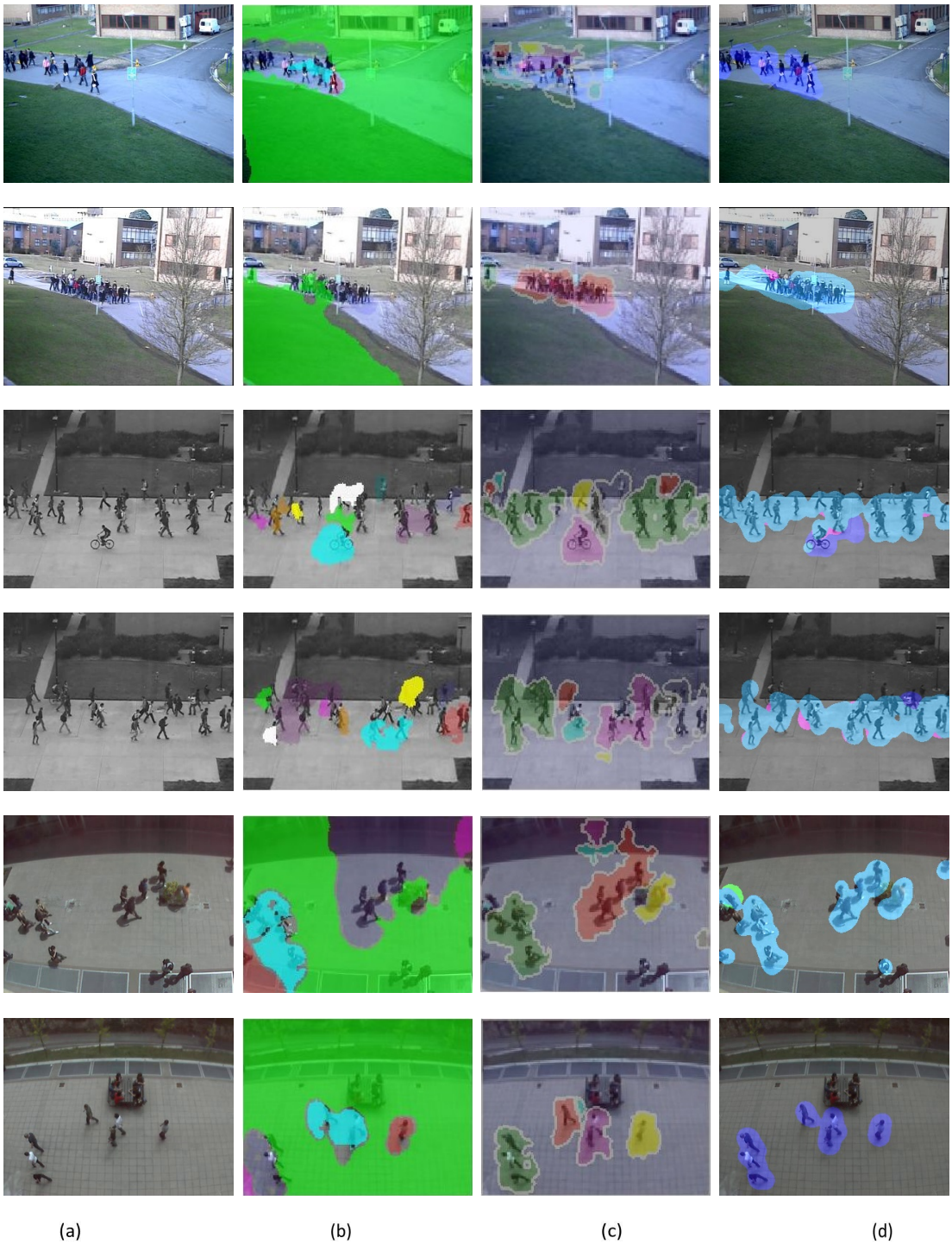


Figure 2.9: Pedestrian flow segmentation. A selection of the results are reported in the figure. The first two rows show the performance on the PETS2009 dataset, and the central and last two rows report the testing on the UCSD and UCD sequences, respectively. Input frames are shown in column (a), Lagrangian approach [2] in column (b); Streaklines approach [32] in column (c); Our approach in column (d).

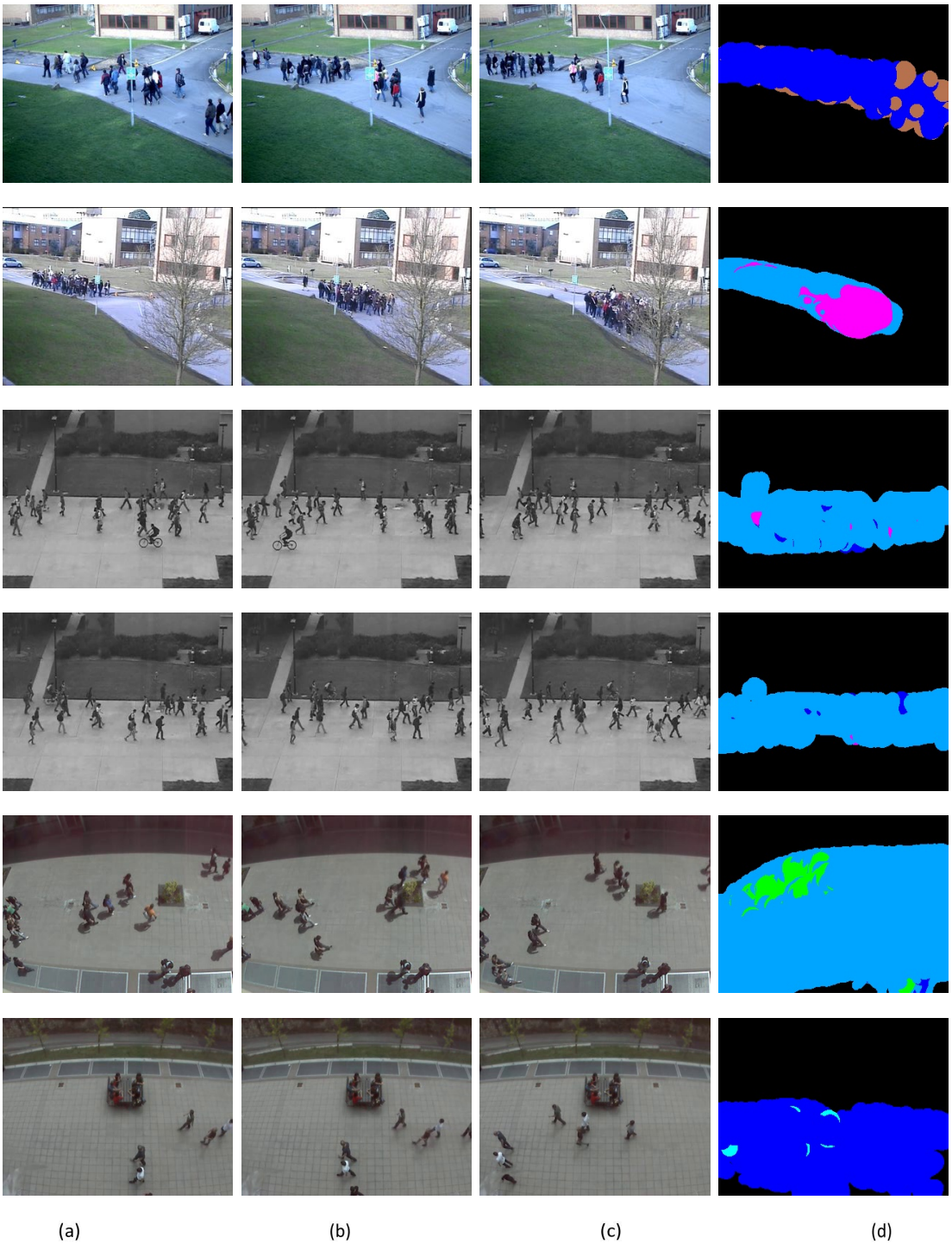


Figure 2.10: Accumulating motion segmentation. Three frames from two different sequences in the PETS2009 dataset are shown in the first two rows, UCSD dataset in middle two rows, and UCD dataset in last two rows (columns (a) to (c)); the accumulated results of our approach are shown in column (d).

.

Table 2.2: Quantitative comparison of the reference methods and the proposed method against the ground truth in pedestrian flow segmentation regarding PETS2009, UCSD, and UCD datasets, respectively. The F-scores for individual video sequences, the average F-score for each datasets, and the average F-scores for all the dataset are provided. The F-scores are shown in bold letters where we outperform the reference methods.

Dataset	**Seq. No.**	**Lagrangian [2]**	**Streaklines [32]**	**Our method**
PETS2009	S1	0.21	0.38	**0.54**
	S2	0.20	0.44	**0.52**
	S3	0.55	0.48	**0.58**
	S4	0.12	0.52	**0.53**
Average		0.27	0.45	**0.54**
UCSD	S5	0.30	0.41	**0.47**
	S6	0.27	0.31	**0.39**
	S7	0.30	0.37	**0.42**
	S8	0.23	0.34	**0.44**
	S9	**0.43**	0.40	0.41
Average		0.30	0.36	**0.42**
UCD	S10	0.20	0.31	**0.48**
	S11	0.29	0.30	**0.42**
	S12	**0.31**	0.28	0.27
	S13	0.23	0.28	**0.42**
Average		0.25	0.29	**0.39**
Average		0.28	0.37	**0.45**

rows show the results of two video sequences taken from the UCSD dataset [30], where people are mainly moving from left to right. Furthermore, the last two rows show the results obtained for two video sequences taken from our UCD dataset. The scene refers to a continuous flow of people moving from bottom left to right and right to left, respectively.

As can be seen in Fig. 2.9, both the Lagrangian approach [2] and the *streaklines* [32] exhibit irregular motion segmentation, especially for the first two video sequences, whereas our approach is spatially and temporally consistent, as well as more accurate. The Lagrangian method [2] tends to segment the motion also when the boundaries in the optical flow field are not salient. The *streaklines* approach [32], instead, is mainly based on spatial correlation with a frailly temporal component, which turns out to be a discriminant factor. Moreover, *streaklines* [32] create stilted time lag and cannot detect local spatial changes, hence leaving spatial crevices in flow and abrupt transitions between frames (column (c) in the last two rows of Fig. 2.9). Furthermore, the Lagrangian method [2] can not cope with video sequences where the pedestrian motion is occurring concurrently in different directions. This can be seen in the third and fourth rows, column (b), of Fig. 2.9. Our approach in the column (d) shows that the obtained results are visually consistent with the pedestrian flow. Fig. 2.10 reports the accumulated results obtained using the proposed approach.

For quantitative analysis, the F-score is calculated for each method. Regarding motion segmentation, the segmentation masks are annotated for the reference methods [2] [32] and our proposed method. These segmentation masks are compared against the ground truth mask. For calculating the F-score, we annotated each tenth frame of a video sequence, in order to save time and resources. According to our approach, the pedestrian flow is segmented in eight possible directions. To this end, we initialize a grid of particles and advect them over a temporal window. Quantitative results

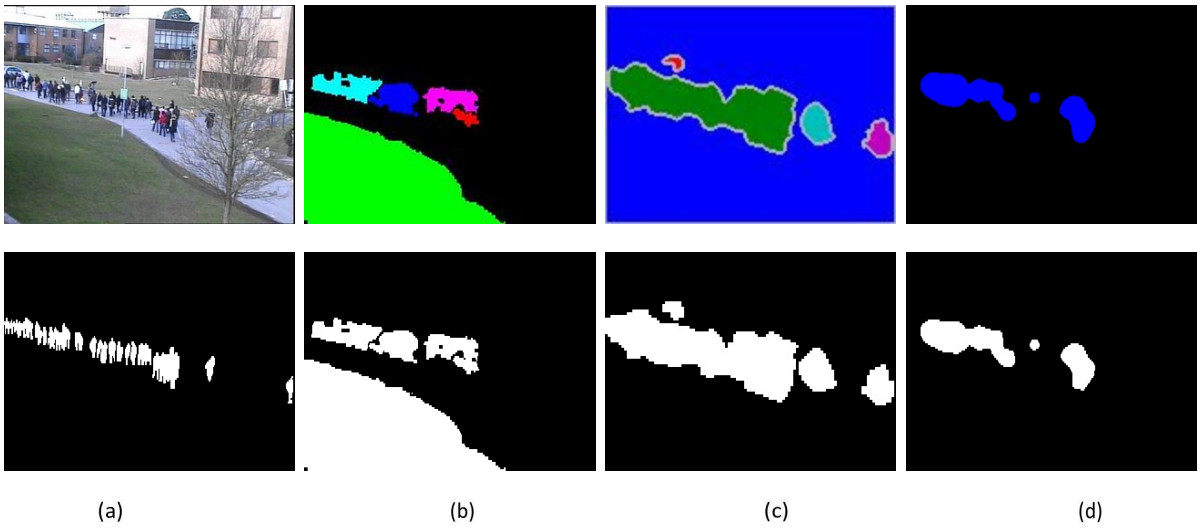


Figure 2.11: Explanation of ground truth calculation. Input frame and ground truth mask in column (a); Lagrangian results [2] and segmentation mask in column (b); Streaklines results [32] and segmentation mask in column (c); the proposed method and the corresponding segmentation mask in column (d).

for flow segmentation are presented in Table 2.2 for each dataset, where the performance evaluation is carried out by comparing our results against the collected ground truth. It is worth noting that most of the datasets do not come with an associated ground truth, as far as the flow segmentation is concerned, and mostly qualitative evaluation is used to validate the approaches. However, in order to further demonstrate the validity of our approach, we have collected the ground truth by manually annotating individuals in each video in the pedestrian scene using the RATSNAKE annotation tool [21]. For instance, the ground truth for a video frame, from the PETS2009 dataset, is annotated in the column (a) of Fig. 2.11. The same annotation tool is used to generate the binary masks for the reference methods and the proposed method (column (b) to (d)).

Comparing to reference methods [2] [32], the superior performance of our method is demonstrated in Table 2.2. This difference in perform-

ance is mainly due to the fact that the Lagrangian approach [2] is more oriented towards coherence in pedestrian flow; as the density of the pedestrian changes over time in a video sequence, the coherence changes as well, making the results less reliable. Similarly, our approach also outperforms the *streaklines* [32] (fourth column). Significant achievement in the average performance of the the proposed approach can be seen in the last row of Table 2.2.

Sensitivity Analysis

Our method is associated with a few parameters. Therefore we have used different parameter configurations, listed in Table 2.3, for all the tests, in order to demonstrate the robustness of our approach. These configurations are encoded in the experiments based on three sets of block sizes: 2x2, 4x4, and 8x8. For block size equal to 2x2, we have used different temporal windows and thresholds ranging from 5 to 15 and from 10 to 20, respectively. In order to investigate the performance of our approach by changing block sizes to 4x4 and 8x8, different thresholds are combined with the temporal window equal to 10. In Tables 2.4, results of our method based on fifteen configurations (C1 to C15) are shown, along with average results for each dataset and the average results for all datasets.

As can be seen the change in the F-score is negligible from configuration C1 to configuration C9 for video sequences in all datasets. However, significant performance decline can be noticed from configuration C9 to configuration C10 for most of the video sequences. For instance, the F-score for video sequence S3 from PETS2009 dataset gravitate from 0.58 to 0.31. The same decline can be noticed for video sequences S8 and S9, and S11 and S12 from UCSD and UCD datasets, respectively. In fact, the performance of our method does not change significantly by changing other parameters except the block size. We also plotted the average along

Table 2.3: Configuration set for sensitivity analysis for our method.

Parameters	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
Threshold	10	15	20	10	15	20	10	15	20	10	15	20	10	15	20
Temporal window	5			10			15			10					
Block size	2x2									4x4			8x8		

with standard deviation for each configuration in column (a) of Fig. 2.12. The plot represents consistent variations from the averages for most of the configurations.

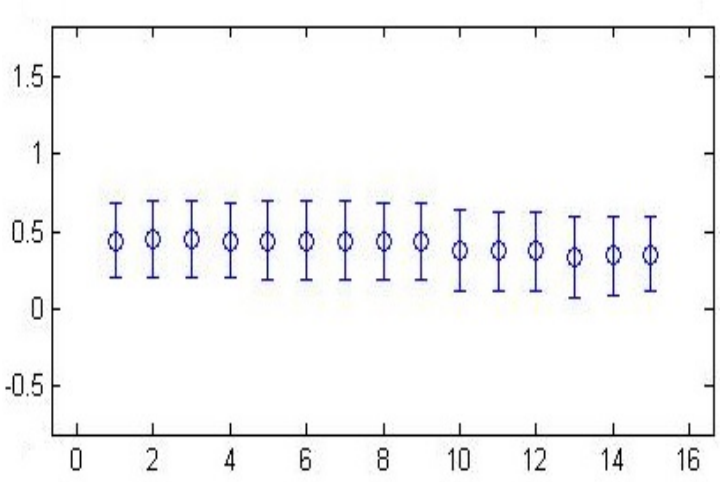
2.4 Enthalpy Model

The method we propose, for dominant motion analysis, consists of three main processing blocks namely: corner features extraction, corner features snipping with an enthalpy model, and random forest inferencing. During the first stage, corner features are extracted from a video frame. Motion patterns, defined in terms of velocity magnitudes, are extracted by tracking the particles using the pyramidal Lucas-Kanade optical flow [66]. In our approach we assume that each corner feature corresponds to an entity and has reactive forces upon other corner features surrounding it. Under this hypothesis, each feature can be classified not only on the basis of its own motion characteristics, but also in relation to the context, in this case provided by its neighbors. Therefore, we incorporate an enthalpy model from thermodynamics to identify potential features of interest only, since the emergent motion patterns in crowd dynamics have dynamical and physical interpretations in thermodynamics. During the last stage,

Table 2.4: Quantitative analysis for our method based on different parameter configurations is provided in pedestrian flow segmentation regarding PETS2009, UCSD, and UCD datasets, respectively. The F-scores for individual video sequences, the average F-score for each dataset, and the average F-scores for all the datasets are provided.

Dataset	**Seq.**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**	**C15**
PETS2009	S1	.54	.54	.54	.45	.54	.54	.54	.54	.54	.52	.51	.51	.48	.47	.48
	S2	.53	.52	.53	.52	.52	.52	.51	.50	.51	.48	.48	.48	.41	.41	.42
	S3	.57	.58	.57	.57	.57	.57	.58	.58	.58	.31	.29	.29	.34	.35	.38
	S4	.52	.53	.51	.51	.51	.49	.50	.50	.49	.45	.45	.42	.46	.49	.49
Average		.54	.54	.53	.51	.53	.53	.53	.53	.53	.44	.43	.42	.42	.43	.44
UCSD	S5	.45	.47	.48	.49	.47	.48	.46	.46	.44	.43	.43	.43	.29	.29	.34
	S6	.38	.39	.38	.38	.38	.37	.37	.37	.37	.31	.31	.32	.27	.28	.29
	S7	.42	.42	.42	.42	.44	.42	.41	.41	.40	.40	.38	.37	.32	.34	.36
	S8	.43	.44	.43	.42	.39	.39	.42	.42	.42	.35	.35	.36	.29	.29	.27
	S9	.41	.41	.41	.41	.38	.42	.38	.39	.41	.33	.33	.36	.39	.37	.37
Average		.41	.42	.42	.42	.41	.41	.40	.41	.40	.36	.36	.36	.31	.31	.32
UCD	S10	.48	.48	.48	.48	.48	.48	.48	.47	.48	.45	.45	.45	.38	.39	.38
	S11	.42	.42	.42	.42	.42	.42	.42	.42	.41	.36	.35	.33	.25	.27	.31
	S12	.27	.27	.26	.26	.26	.26	.26	.26	.25	.20	.20	.19	.23	.23	.29
	S13	.42	.42	.42	.42	.42	.42	.42	.42	.41	.40	.40	.39	.27	.27	.28
Average		.39	.39	.39	.39	.39	.39	.39	.39	.38	.35	.35	.34	.28	.29	.31
Average		.44	.45	.45	.44	.44	.44	.44	.44	.43	.38	.37	.37	.33	.34	.35

the orientation features of the corner features act as input data to the random forest, so as to infer the dominant flows. The orientation features



(a)

Figure 2.12: Results of the pedestrian flow segmentation with respect to different configurations. Each 'o' symbol presents the average calculated over all video sequences of three datasets. Standard deviations are also plotted representing variations from the averages.

and the corresponding label sequence are used to learn the random forest parameters during the training stage, and the dominant flows are inferred on the test samples.

2.4.1 Corner features extraction

We selected corners as the main feature to analyze, since they represent peculiar elements in the scene and can be easily tracked in dense crowded scenes, leading to better consistency and accuracy in tracking, especially in scenes representing complex motion. The corner features are extracted from the video frame as shown in Fig. 2.13. To detect them, the function formulated in Eq. (2.19) is maximized.

$$E\left(u,v\right) \approx \sum_{xy} w(x,y)[I(x+u,y+v) - I(x,y)]^2 \tag{2.19}$$

In Eq. (2.19), $w(x,y)$ is the window at position (x,y), $I(x,y)$ is the

Figure 2.13: Corner features initialization. Frame from an irregular crowd video sequence (Left); the same frame with corner features driven (Right).

.

intensity at (x, y), and $I(x+u, y+v)$ is the intensity at the moved window $(x+u, y+v)$. The function in Eq. (2.19) can be reformulated as in Eq. (2.20).

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.20}$$

Where u is the displacement of the window w along x, and v is the displacement of the window w along y. The score R for a corner feature can be determined from the eigenvalues of the matrix M as formulated in Eq. (2.21).

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2) \tag{2.21}$$

In the equation, k is a free parameter. A window with the greatest R is considered as a corner feature.

2.4.2 Enthalpy model

The objective of this processing stage is to isolate and filter out the corner features that do not contribute to the identification of the dominant crowd flow detection. Motion information, defined in terms of velocity magnitudes, is extracted at regular intervals of K frames by tracking the corner features using the Lucas-Kanade optical flow [66].

The motion patterns observed in a crowded scene can be well modeled through a common thermodynamic measure, the enthalpy. Compared to the entropy model, which measures the disorder of a process, the enthalpy is a measure of the total energy of a thermodynamic system.

In thermodynamics, the enthalpy of a system with respect to temperature T and pressure P is formulated in Eq. (2.22).

$$dH = \left(\frac{\partial H}{\partial T}\right)_P dT + \left(\frac{\partial H}{\partial P}\right)_T dp \tag{2.22}$$

In a thermodynamic system, energy is measured with respect to some reference energy. Therefore, the internal energy U is calculated as a variation in U, instead of an absolute value as formulated in Eq. (2.23).

$$dU = \left(\frac{\partial U}{\partial T}\right)_V dT + \left(\frac{\partial U}{\partial V}\right)_T dV \tag{2.23}$$

It is worth mentioning that, compared to a thermodynamic system, the crowd dynamics represents a homogeneous system, which is clearly independent from the temperature. We consider the crowd as a continuum, simultaneously being able to capture motion properties of each corner feature at the individual level. It allows us to treat corner features as constituents (subpopulations) of the large crowd, each having its own motion properties. We thus have the possibility to examine the interactive behaviour between subpopulations, in the spatial neighborhood, which have

distinct characteristics represented by the enthalpy model as formulated in Eq. (2.24).

$$H = U + pV \tag{2.24}$$

Here, U is the internal energy, p is the pressure, and V is the volume of the system. We exploit the kinetic energy in terms of internal energy, since we are only interested in motile corner features. *Pressure* is defined as $p = Force/Area$ and $Force$ is $F = mass * acceleration$. For acceleration, we calculate the average velocity $\langle v \rangle$ in the spatial neighborhood over time, whereas the area A is the total number of corner features in the spatial neighborhood. Mass and volume of each corner feature may be associated with its contribution in the corresponding subpopulation, in the spatial neighborhood. However, we set them to 1 in our case to maintain consistency. Our enthalpy model is thus formulated in Eq. (2.25).

$$H = \frac{1}{2}mv^2 + \left(\frac{\partial\langle v \rangle}{\partial t}\right)\left(\frac{1}{A}\right) \tag{2.25}$$

Figure 2.14: Interaction flow. The extracted corner features (left column); the same frame with the interaction flow overlayed (right column).

.

After evoking the relevant corner features using the enthalpy model, as

depicted in Fig. 2.14, the orientation information of each corner feature in terms of angle of motion is extracted at regular intervals of K frames. We have selected 8 different directions quantized with a step of 45 degrees as depicted in Fig. 2.15, where R, TR, T, TL, L, BL, B, and BR stand for right, top right, top, top left, left, bottom left, bottom, and bottom right, respectively. The collected orientation features are stored to construct a feature vector for each corner feature. The feature vector is fed to the random forest classifier as an input (details are provided below) that in turn signals the corresponding label for the direction. To this end, a *tracklet* is drawn from the initial position to the final position of the corner feature where each pixel in the *tracklet* is assigned the same label. An example of a *tracklet* is shown in Fig. 2.16.

2.4.3 Random forest

A random forest [10] is a classifier consisting of a set of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1,.....K\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$. Given an ensemble of classifiers $h_1(\mathbf{x})$, $h_2(\mathbf{x})$, . . . ,$h_K(\mathbf{x})$, the margin function for the random forest over the input vector $\mathbf{x}$ and the label y is formulated in Eq. (2.26).

$$\begin{aligned} mg(\mathbf{x}, y) = av_K I(h_k\mathbf{x} = y) - \\ max_{j \neq y} av_k I(h_k(\mathbf{x}) = j) \end{aligned} \tag{2.26}$$

In Eq. (2.26), $I(\cdot)$ is the indicator function. The margin measures the extent to which the average number of votes at an input $\mathbf{x}$ for the right class y exceeds the average vote for any other class. The larger the margin, the higher the confidence in the classification. The generalization error is given by Eq. (2.27).

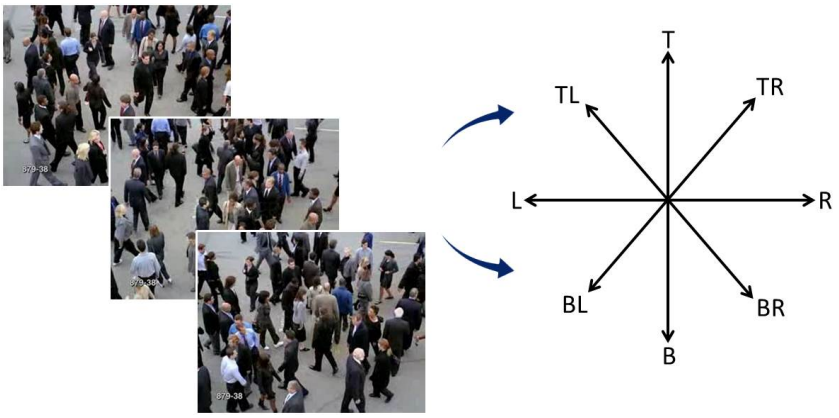


Figure 2.15: Orientation-based dominant crowd flows detection. We analyze the crowd flows in eight possible directions according to the annotations on the left.

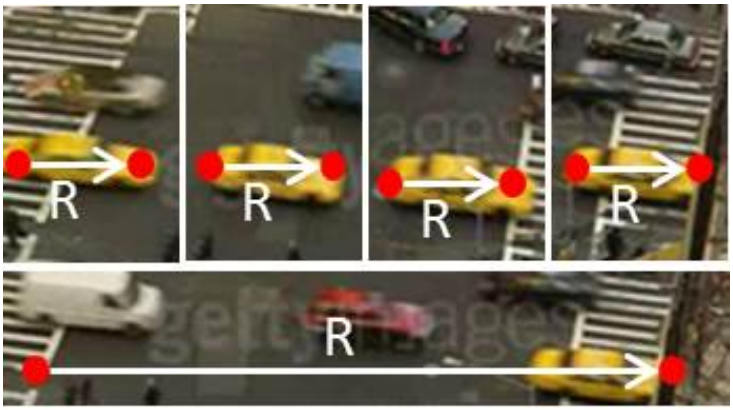


Figure 2.16: Example. The top four frames show the motion of a corner feature to the right side of the image, while the bottom frame shows the computed tracklet.

.

$$PE = P_{\mathbf{x},y}(mg(\mathbf{x}y) < 0) \tag{2.27}$$

Where the subscripts $\mathbf{x}$, y indicate that the probability is over the $\mathbf{x}$ and y space. When the number of trees increases, the generalization error PE converges as in Eq. (2.28) for all the parameters Θ_1.....Θ_K.

$$\begin{aligned} P_{\mathbf{x},y}(P_\Theta(h(\mathbf{x},\Theta) = y) - \\ max_{j \neq y} P_\Theta(h(\mathbf{x},\Theta) = j) < 0) \end{aligned} \tag{2.28}$$

This means that random forests do not overfit as more trees are added, but produce a limiting value of the generalization error. A random forest specifies a particular label, given the observation sequence. Specifically, $\mathbf{x}$ is our input sequence, consisting in N observations collected within the K frames window (i.e. $\mathbf{x} = x_1, x_2, \ldots, x_N$), containing the orientation features. Given the observation sequence, the random forest signals the most probable label in terms of direction, inferring the output label y_m ($y_m = y_1, y_2, \ldots, y_M$) of the respective crowd motion direction.

During training, all the trees exploit the same parameters but on different training sets. These sets are generated from the original training set using the bootstrap procedure: for each training set, the same number of vectors are selected randomly as in the original set. Moreover, the vectors are chosen with replacement, meaning that some vectors will occur more than once and some will be absent. Only a random subset of variables are used to find the best split at each node of each trained tree. With each node a new subset is engendered. However, its size is fixed for all the nodes and all the trees.

2.4.4 Experimental results

The experiments are carried out on video sequences from benchmark datasets such as UCF [2] and UCD [54]. The video sequences in the UCF dataset are originally taken from Getty-Images, Photo-Search and Google Video. To test the generalization properties of our proposed method, we crawled two video sequences from YouTube (shown in the last two columns of Fig. 2.17.). For each corner feature, the orientation features consist of a vector of $N = 4$ observations, where each element of the vector corresponds to the orientation information extracted after every $K = 8$ frames. The possible output directions are $M = 8$, one label every 45°. We do not consider corner features with with small motion magnitudes. To eval-

uate the performance of our proposed method, we compared it the optical flow (as a baseline method), as well as the segmentation methods proposed by [54] and [55] in Table 2.5. The first column renders the original video sequences, while columns (2 - 6) present the ground truth, and the results obtained using the optical flow, the method proposed in [54], the method proposed in [55], and our proposed method, respectively.

Table 2.5: Comparison of our approach with the reference approaches in dominant crowd flows detection. The first column presents the original video sequences and the second column shows the ground truth in terms of four dominant directions and the number of people moving in each dominant direction, respectively. Columns {3-6} present the reference approaches and the proposed approach.

No.	Ground truth	Optical flow	ICPRw[18]	ICIP[19]	Proposed
1	TL-R-TR-L 80-54-24-19	1 25.76-18.33-8.07-21.41	0 7.75-79.68-0-11.91	2 43.81-18.88-11.64-16.53	4 52.38-15.3-13.19-12.26
2	R-L-TR-T/B 40-35-15-12/12	1 17.74-17.82-15-17.86/6	2 46-13.4-1.89-4/11	4 41.64-29.78-8-5/3.63	2 45.87-33-2.98-3/5.23
3	R-BR-L-B 70-34-28-15	2 34.66-20.40-21.82-6.97	4 62.50-27.99-5.66-2.53	4 48.5-27.76-20.4-1.57	4 43.87-29.66-24.63-1.09
4	R-BR-TL-TR 100-60-57-29	2 32.48-7.17-8.86-9.81	2 47.59-26.23-2.87-8.51	2 52.26-21.58-7.43-11.38	4 73.78-13.1-5.9-2.65
5	R-L-TL-TR 39-34-5-1	0 25.16-25.26-4.36-5.60	2 65.5-11.36-0-0	2 43.62-40.52-0.73-5.11	2 46.69-45.31-0.17-0.76
6	R-TR-L 37-30-2	1 32.56-9.88-17.78	1 100-0-0-0	3 77.37-17.44-3.3	3 85.62-11.25-2.37
7	B-TL-BL-T 58-42-9-5	1 17.97-24.33-3.44-26.47	2 13.73-3.72-9.34-1.79	2 43.43-44.4-4.85-1.39	4 45.83-37.13-8.66-1.37
8	R-T-L-B 71-46-31-12	1 19.5-26.14-20.37-8.7	2 37.54-22.5-4.83-7.67	4 41.31-35.84-14.51-1.31	4 45.35-33.62-14.69-0.99

In our experiments, the ground truth consists of the number of individuals moving in each direction. By examining the ground truth, we identify

Table 2.6: Quantitative comparison of the reference approaches and the proposed approach with the ground truth in terms of accuracies. The first column shows a total number of 31 dominant directions, while other columns present number of correctly detected dominant directions along with percent accuracies by the reference approaches and the proposed approach.

Total	Optical flow		ICPRw[18]		ICIP[19]		Proposed	
	Correct	Accuracy	Correct	Accuracy	Correct	Accuracy	Correct	Accuracy
31	9	29.03%	15	48.38%	23	74.19%	27	87.09%

that a significant number of people is moving only in four possible directions instead of all eight directions. Therefore, we perform analysis only in four directions, where most of the people are moving, for the purpose of evaluation. For instance, the ground truth, TL-R-TR-L, for the first video sequence shows that most of the people i.e. 80 are moving in the top-left direction, while 54 people moving in the right direction stood second. There are 24 people moving in the top-right direction and 19 people moving in the left direction. To compare against the ground truth, orientation information is collected at each temporal window and accumulated over time for each video sequence for the reference approaches and the proposed approach. To further elaborate, frames from video sequences are shown in the first row and the orientation information are annotated with different colors for the sake of visualization in the second row of Fig. 2.17, from the proposed method. In Table 2.5, the number of correctly identified directions along with orientation information in terms of percentages are provided for the reference approaches and the proposed approach. For the first video sequence, the pure optical flow collects 25.76% orientation information in the top-left direction, while 18.33% in the right direction, 8.07% in the top-right direction, and 21.41% in the left direction, respectively. Therefore, the pure optical flow correctly identifies one dominant direction, since the orientation information collected only in the top-left direction corresponds

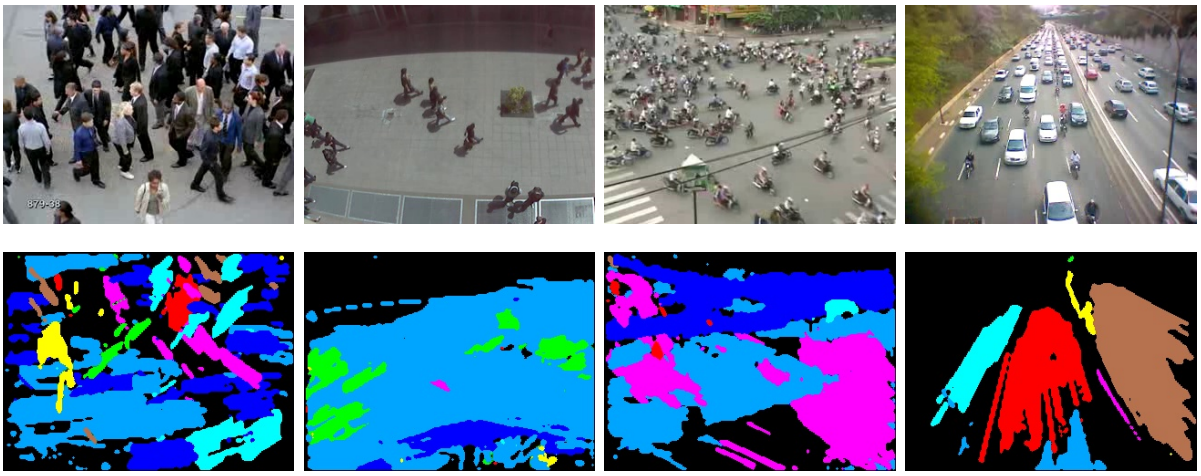


Figure 2.17: Orientation information. Input frames from video sequences (first row); Orientation information annotated with different colors (second row), where each color is associated with a specific direction.

.

with the ground truth in terms of highest numbers in the same positions. Comparing our results with the reference approaches, we notice that our approach performs better or equally for most of the video sequences. In particular, our approach outperforms the reference approaches in video sequences, one, four, and seven, where it correctly identifies all four dominant flows. In Table 2.6, the number of correctly identified dominant directions along with the percent accuracies are presented by the reference approaches and the proposed approach, respectively. The first column presents the total number of dominant directions for all video sequences. The evidence for the significant performance of our approach over the reference methods lies in the fact that on the one hand the corner features combined with the enthalpy measure, highlights characteristic areas in the crowd, and on the other hand the random forest presents significant predictive performance to identify dominant flows.

2.5 Entity Grouping

Detectors and trackers are likely to fail in severe occlusions when the number of moving subjects in the scene increase. Therefore, more generic approaches based on the motion flow, commonly exists in the crowded scenes, can be exploited in such scenarios. These approaches ignore the notion of person, however, it is still possible to estimate, for example, the density of people, and the aggregation points in the monitored environment. This turns out to be an efficient pre-processing step for any further and more detailed analysis. Our approach [46] considers each particle as a single entity where each particle represents the position of a pixel. In our work, particles are generated through the GoodFeaturesToTrack algorithm, and tracked by the Lucas-Kanade optical flow. Each particle is characterized by its own motion properties and its influence over the neighboring particles. Therefore, we exploit particle mutual influence model to extract potential particles of interest and filter out rest of the particles. Regarding my contribution, I extract features from the potential particles and feed them into a MLP neural network to form coherent groups of entities sharing similar motion properties.

2.5.1 Mutual influence

The first step of our proposed approach relies on particles dynamic properties. Each particle corresponds to an entity and has attractive and repulsive forces upon other particles surrounding it. Under this hypothesis, each particle can be classified on the basis of its own and its neighbors motion characteristics. The influence among particles can be expressed by a stochastic matrix called influence matrix as proposed by Pan et al. [39]. This stage prunes the particles marked as *alone* and considers just the *grouped* particles relevant for further processing. Fig. 2.18 shows the ex-

tracted particles of interest annotated in red.

Figure 2.18: An example of particle initialization (left) and after pruning (right).

2.5.2 Feature extraction

The objective of the features extraction process is to identify low-level information relative to the particles interaction. Features are extracted only for the particles obtained from the mutual influence model.

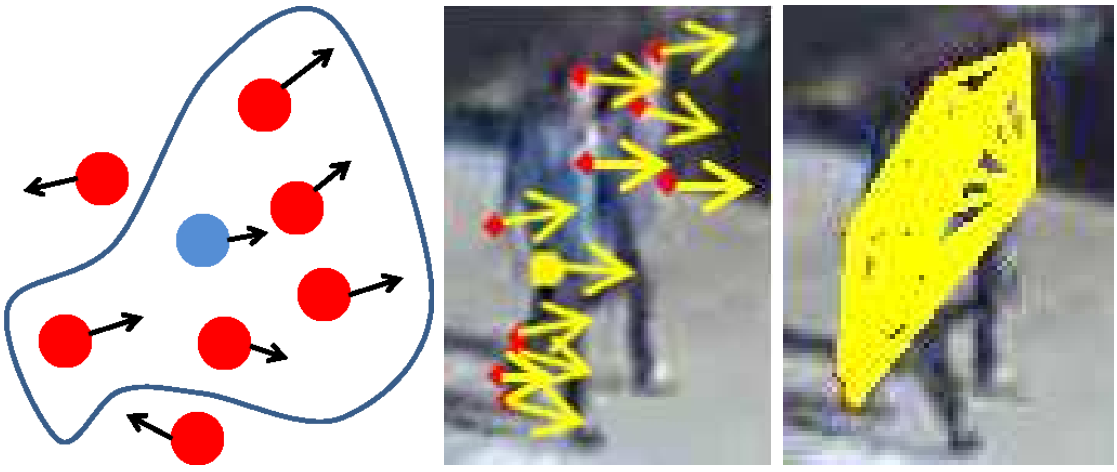
In our approach we have selected the average distance among the particles and their density as two representative elements to infer the interaction among particles. In fact, proximity, which is partially exploited also in the influence model measures the instantaneous relationship among neighboring entities. At the same time, the higher the density of the particles, the higher the chance for them to interact.

For both features, orientation is used as a prior, meaning that particles are considered in the same group, only if their relative offset in terms of direction of motion fall in a predefined range. In Fig. 2.19 (a), a set of synthetic entities are shown where a reference entity, annotated in blue, is grouped with the neighboring entities annotated in red. On the contrary, two entites are not included in the same group since their orientations do

not conform to the orientation of the reference entity. Moreover, a reference entity annotated in yellow and neighboring entities annotated in red are shown in Fig. 2.19 (b). These entities constitute a group, as shown in (c), according to the compliance in terms of density and mutual distances with the reference entity.

2.5.3 Classification

In order to weight the features we have selected for entity grouping, we have trained an MLP neural network described in detail in Section 3.4.2. To combine the particles from the preceding stage of the mutual influence model, the average distance of a reference particle with its neighbors is accumulated and averaged. A particle is only considered for grouping with a reference particle if its relative orientation is compliant with the orientation of a reference particle. The density and average distance of the reference particle are fed as an input to the MLP.



(a) (b) (c)

Figure 2.19: Entities grouping. Synthetic example of moving entities (a), moving entities obtained from the particles mutual influence model (b) and grouping implemented according to the motion and density features (c).

In Fig. 2.20, the tracked groups of entities are depicted. Two groups, annotated in cyan (left) and yellow (right) respectively, are zoomed and shown in the third row. Initially, entities are pruned with the particles mutual influence model and propagated over a predefined temporal window to associate them in groups in accodance with the features. At the same time, these groups are then mapped to a new set of pruned entities, with mutual influence model, which are then tracked over the same temporal window and the re-association process is repeated over time.

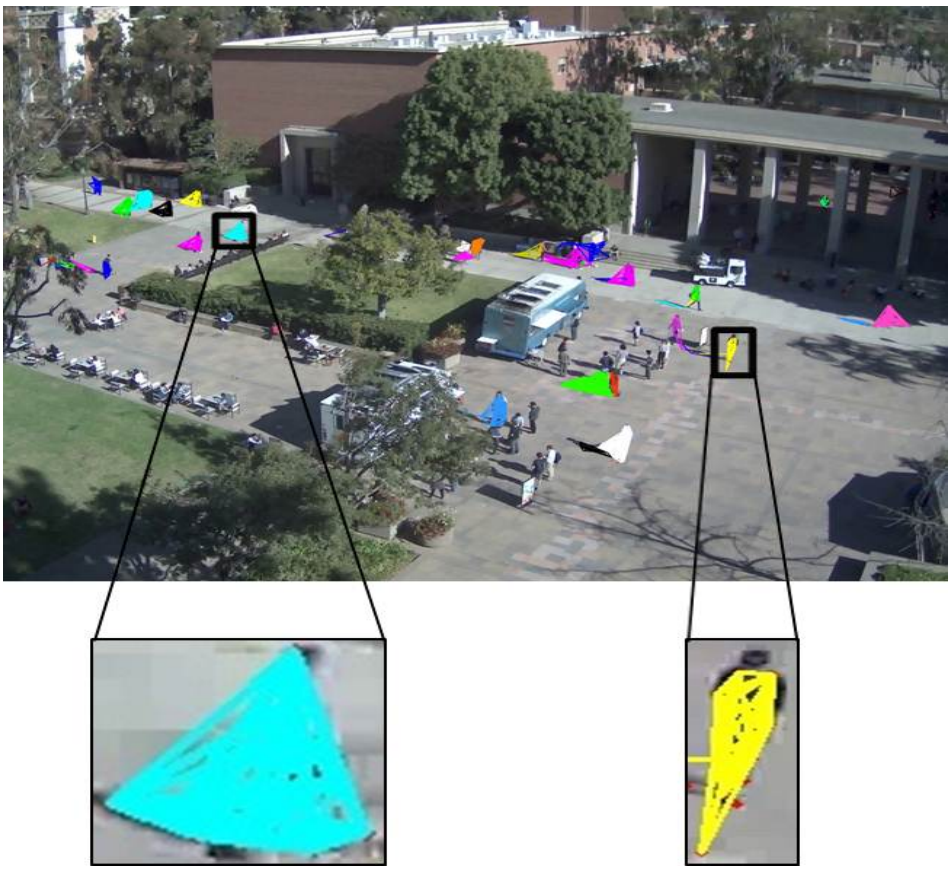
2.5.4 Experimental results

For the experiments, we consider the UCLA [3] and the BIWI [41] datasets. The UCLA dataset presents human activities including walking, talking, riding-skateboard, riding-bike and driving car. We consider only the ETH sequence from the BIWI dataset because of the exclusive presence of pedestrians. For the influence model, the length of the time window is set to 45 frames. The neural network has been configured considering one input layer, two hidden layers and one output layer. The input layer consists of two neurons, each hidden layer consists of three neurons, and a single neuron is allocated to the output layer. To extract the input features, the relative orientation with a reference particle is set to ± 30 degrees. Furthermore, the distance threshold from the reference particle is set to 80 pixels. For the purpose of training, we exploited 1000 training samples, where each sample is a vector of two observations consisting of average distance and density of particles. These parameters are kept constant for both sequences to demonstrate the capability of generalization.

The obtained results are depicted in Fig. 2.21. The first image (a) shows an example of the method applied on the UCLA dataset, where we can notice a very clear group composition, especially for zones A, D and E. In zone B the number of particles is not dense as in the previous cases,

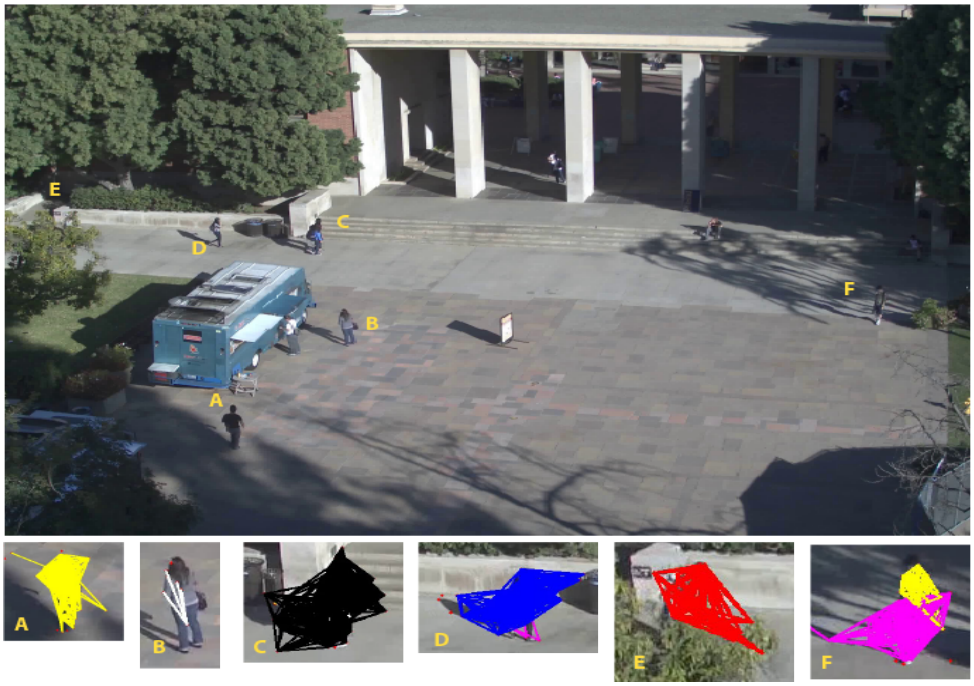
but still grouping is possible since the distance and density features of the entities are sufficient for the neural network. In zone F, two groups have been detected instead of a single one; this can be ascribed to the severe shadowing, in which the pedestrian is located where, in fact, features of the entities are mainly segmented into two groups. The quality of the results is also depicted in (b) by the ETH sequence, where the groups are well defined (zones A, B, and C). However, in this case a few mistakes are also present (zone D). This is most probably connected to the limited resolution and the compression artifacts. The UCLA dataset have much better results in terms of grouping not only because of the resolution but also because the bird eye view is less accentuated and the illumination conditions are considerably better.

(a)

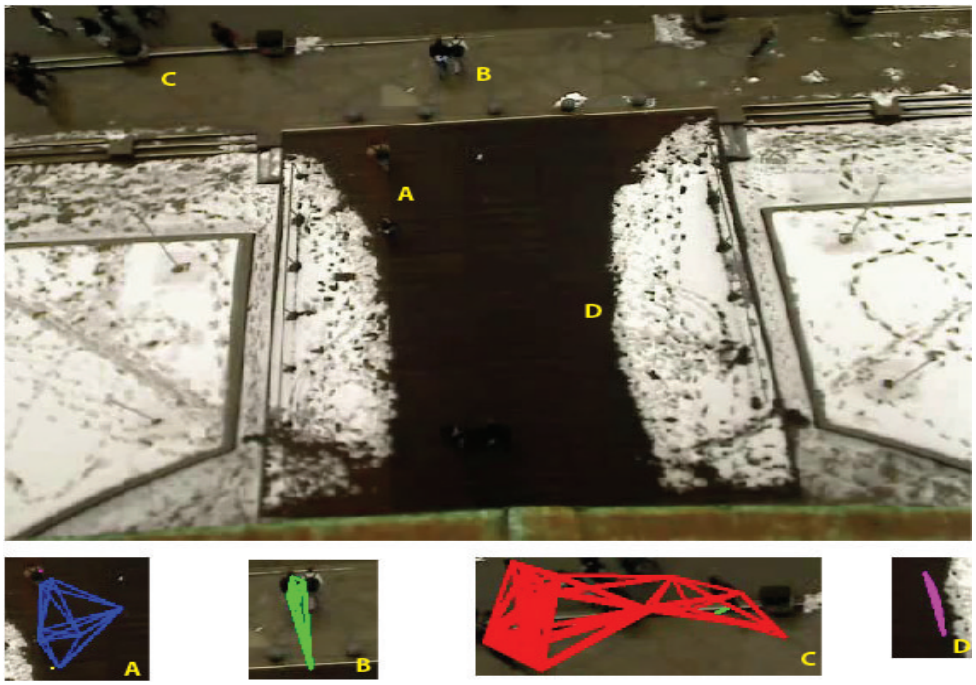


(b)

Figure 2.20: Input frame (a), entities grouping with the zoom on two sample groups (b).



(a)



(b)

Figure 2.21: Particle influence and entity grouping. Results obtained on the UCLA dataset (a) and on the BIWI dataset (b). For visibility, labels are super-imposed on the original frame and the corresponding grouped entities are zoomed in lower row.

Chapter 3

Anomaly Detection

This chapter presents state of the art regarding anomaly detection and then presents our proposed methods. In particular, the techniques based on deviant orientation information, Gaussian mixture model, and corner features are presented.

3.1 State of The Art

Most of the methods for anomaly detection fall under the category of high density flow analysis. Anomaly detection is applicable in a variety of domains, including intrusion detection, traffic monitoring, and behavior analysis. Krausz and Bauckhage [24] detect anomaly in terms of stampede. For this purpose, they analyzed video footages from the Loveparade music festival in Duisburg, Germany. They exploit the dense optical flow to compute the two dimensional histograms of motion magnitude and motion direction of the flow vectors. Then a Non-Negative Matrix Factorization, proposed by Lee and Seung [27], is applied to decompose the histograms, so as to extract motion patterns that can highlight congestions and stampedes. Mahadevan et al. [30], Seidenari et al. [47], and Bertini et al. [7] detect anomalies in terms of circulation of non-pedestrian entities in the scene, by considering the variations of objects appearance to infer abnor-

mality information (e.g. the appearance of a biker is different from the appearance of a pedestrian). For this purpose, temporal normalcy and spatial normalcy are exploited by Mahadevan et al. [30]. Temporal normalcy is modeled with a mixture of dynamic textures whereas spatial normalcy is modeled through a discriminant saliency detector. Seidenari et al. [47] and Bertini et al. [7] exploit a non-parametric approach based on local spatio-temporal features to detect and localize anomalies. Ullah et al. [57], Mehran et al. [33], and Cui et al. [16] detect abnormal events in terms of escape panics. For this purpose, the social force model (SFM), proposed by Helbing and Molnar [20], is exploited by Mehran et al. [33]. After the superposition of a fixed grid of particles on each frame, the SFM is used to estimate the interaction forces associated to the pedestrian behavior. After that, a bag of words method and a Latent Dirichlet Allocation are exploited to discriminate between normal and abnormal frames, localizing the abnormal areas as those representing the highest force magnitude. Cui et al. [16] use spatio-temporal interest points, proposed by Laptev et al. [26], to detect the behavior of the pedestrians. For each interest point, an energy potential is calculated based on the positions and velocities of its neighbor points.

3.2 Deviant Information

Here the anomaly is based on the segmentation information obtained in the Section 2.3. Once the motion flow is extracted from the foreground, an accumulator $AC(t)$ is constructed on top of each block, in order to create the pedestrian motion model (represented by $P(t)$ on the right side of Fig. 2.4), by collecting evidence regarding the dominant directions of pedestrian motion. The accumulator is updated at every frame, keeping up with the evolution of the pedestrian flow. The pedestrian motion model $P(t)$

combined with the output of multi-label optimization $S(t)$ and orientation information $O(t)$ is exploited to detect anomalies $A(t)$, annotated in white as shown in Fig. 2.4. The union of motion segmentation and orientation subject to pedestrian motion model, represented by P(t), allows retrieving the presence of anomalies as formulated in Eq. 3.1.

$$A(t) = \{S(t) \cup O(t)\}\Big|_{P(t)} \tag{3.1}$$

To further elaborate the anomaly detection, we can create a histogram using the motion segmentation model described in Section 2.3.2, which, updated on a frame basis and computed on a block basis, represents the frequency (occurrence) of each of the selected directions (D) in each block, as formulated in Eq. (3.2).

$$H_{m,b}(\theta) = \sum_{frames=m}^{F} \sum_{blocks=b}^{B} \sum_{directions=i}^{D} [\theta == i], \tag{3.2}$$

$$\begin{aligned} Where \quad & \forall m \in \{1, \ldots, F\}, \\ & \forall b \in \{1, \ldots, B\}, \\ & \forall i \in \{1, \ldots, D\} \end{aligned}$$

where F, B, and D represent the number of frames, the number of blocks in each frame, and the number of possible directions, respectively. The magnitudes of the histogram are compared against a threshold (T_0) to determine the most representative directions of the motion for the specific block, thus creating the reference motion model P: people moving in these directions will neither be signaled nor identified as anomalies as shown in Fig. 3.1. The first row in Fig. 3.1, represents people moving from left to right while the second row represents people moving from bottom right to top left.

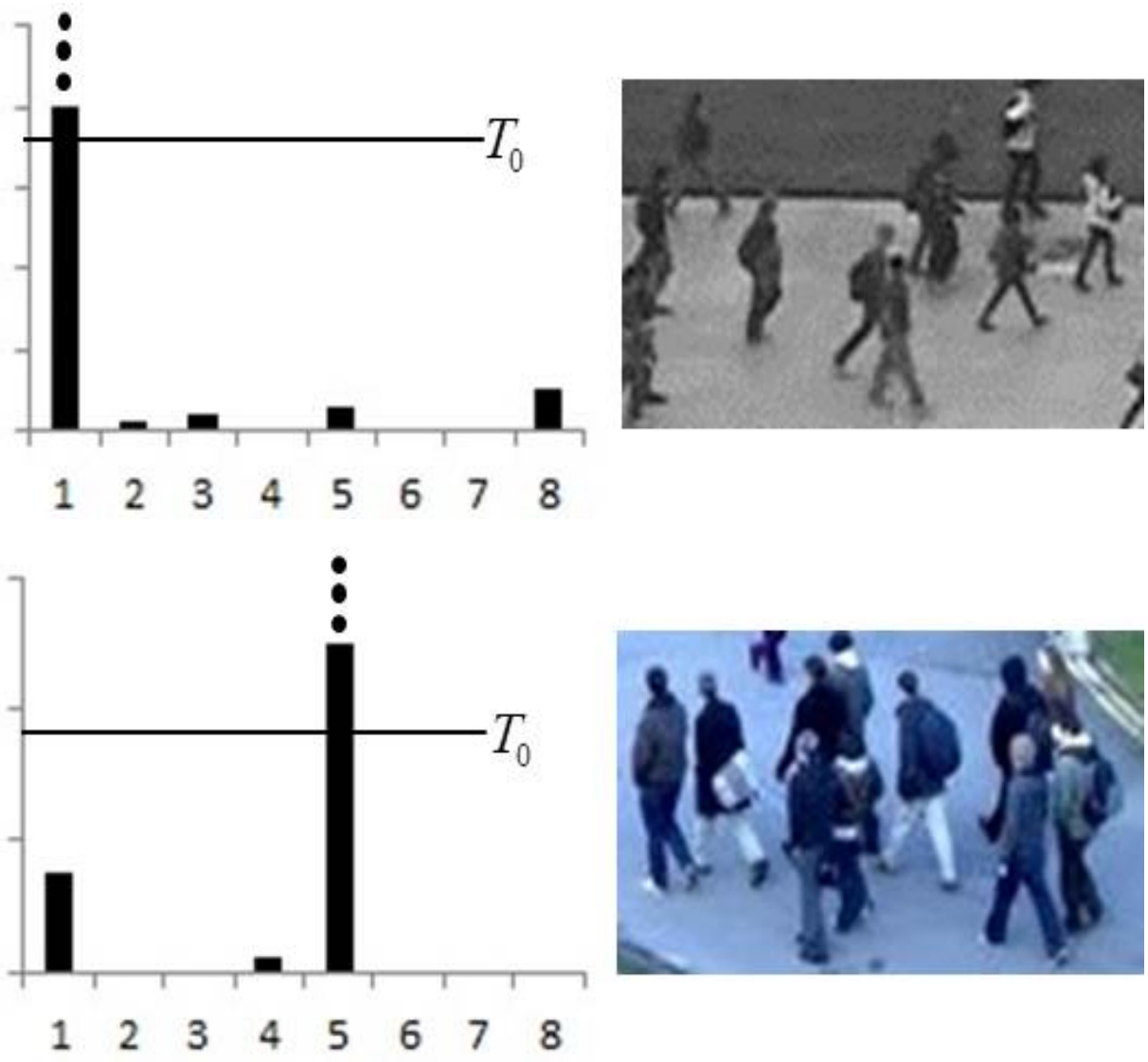


Figure 3.1: Most representative orientations of motion for two video sequences for the assessment of the pedestrian motion model P. Orientations are numbered from 1 to 8, on horizontal axes, representing the angle from 0 to 360 at steps of 45 degrees. Vertical axes represent orientation information accumulated over time.

To further elaborate, the anomaly is detected through a two-step procedure. During the operation of the algorithm, the deviant direction information is accumulated over a temporal window as formulated in Eq. (3.3). The consistency of the accumulated information is then evaluated by comparing it with the motion of the particles in the area. If this condition is verified and the accumulator exceeds a predefined threshold, the anomaly is signaled.

$$\forall i,\ P(i) = \begin{cases} 1 & \text{if} \quad H(i) > T_0 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

3.2.1 Experimental results

To validate the performance of our approach, we have conducted the experiments on benchmark datasets such as PETS2009 [42] and UCSD [30] and our UCD [54]. To evaluate the anomaly detection performance of our approach, we compare it against two baseline methods i.e. SIFT features [28] and GoodFeaturesToTrack [49]. We extract these features and advect them over a temporal window, using the same optical flow technique [66] to maintain consistency with our method, to collect orientation information. The orientation information is exploited to build the pedestrian motion model initially and we identify the daviant motion information later. For this purpose, the same procedure is followed as we did for our method. For quantitative analysis, the F-score is calculated for each method.

In order to detect the anomaly, the threshold T is calculated over a temporal window. For anomaly detection, the histogram is updated on a block-by-block basis in the segmentation stage (consisting of the initial 50% of the video frames in these experiments) to determine the dominant motion directions of the pedestrian flow. Then, a threshold T_0 is applied to highlight only the most evident events. Considering that all the three

datasets only exhibit one main direction of motion for each video segment, we have merged two video sequences, so as to simulate people walking in different directions. For this purpose, we generated all possible combinations of video sequences in each dataset to validate the robustness of our approach. The results for the anomaly detection are shown in Fig. 3.2 for two video sequences from each dataset. Column (a) depicts a sample of the original frames and column (b) highlights the detected anomalies (annotated in white) on top of the pedestrian motion model.

Quantitative results for anomaly detection are presented in Table 3.1. The performance evaluation is carried out by comparing our results against two baseline methods, SIFT features [28] and GoodFeaturesToTrack [49]. To identify dominant directions of motion, pedestrian motion model has been built. The anomaly is signaled in the scene when people or objects start moving differently from the pedestrian motion model. For this purpose, we calculate the F-score, provided in Table 3.1, for the basline methods and our method. The significant performance increase over the two baseline methods is evident in most of the video sequences. It is worth noting that the F-score for all video sequences in UCSD dataset are the same for both the baseline methods. In fact, similar motion patterns are exhibited over the same number of frames in these video sequences. Both the baseline methods are failed to detect anomalies in the video sequences of UCD dataset (except S10-S12 and S11-S13 video sequences in case of SIFT). These methods cannot collect enough evidence over time, due to the unstable motion of the extracted features, in term of orientation information to signal anomalies. We also provided the average F-score for each dataset and the average F-score for all the datasets for the baseline methods and our method. Quantitative results demonstrate that the proposed approach is robust enough to detect anomalies.

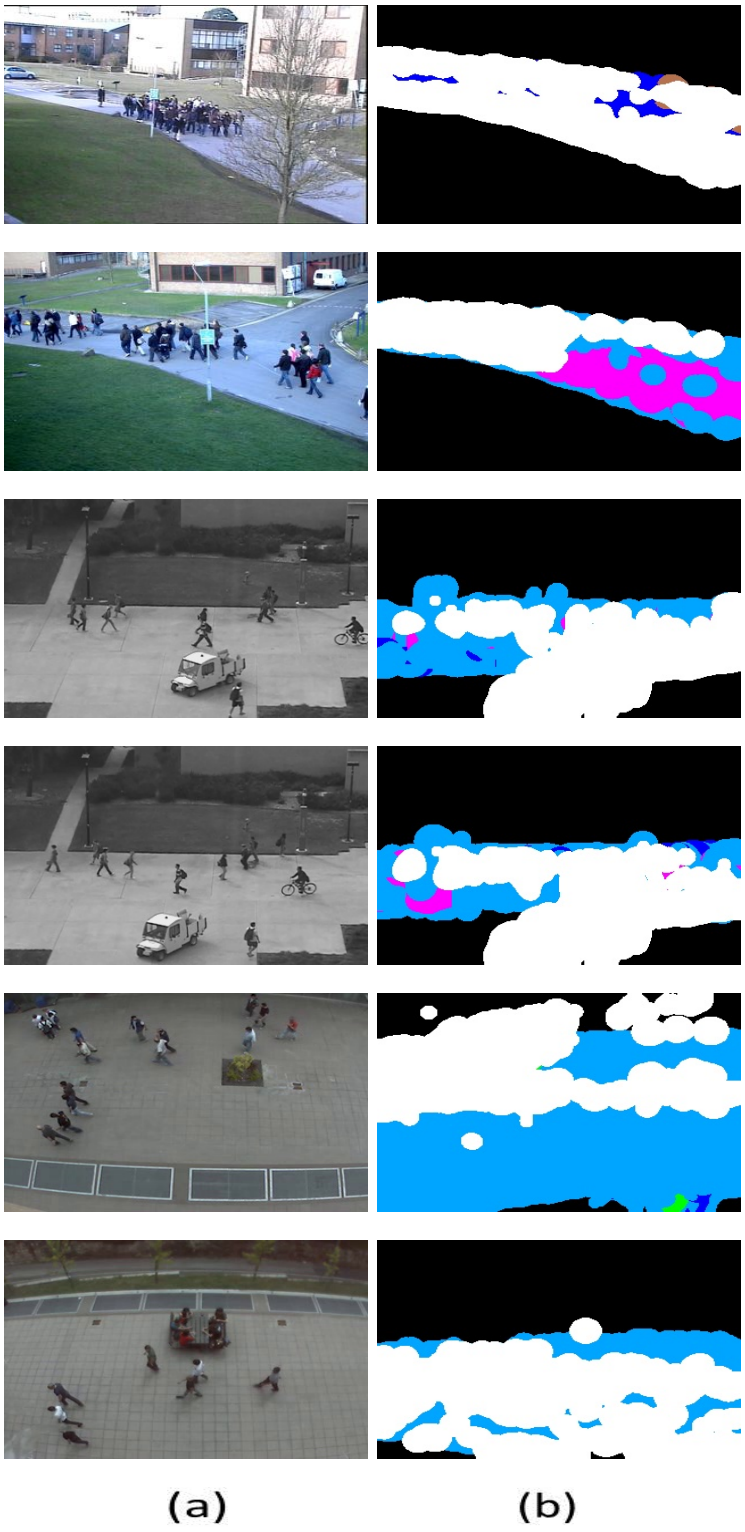


Figure 3.2: Anomaly detection. Input frames from two video sequences are provided from the datasets: PETS2009 (first two rows), UCSD (middle two rows), and UCD (last two rows) in column (a), whereas detected anomalies are shown in column (b).

Table 3.1: Comparison of our method with the baseline methods.

Dataset	Seq. No	SIFT	G.F.T	Our method
PETS2009	S1-S2	0.66	0.66	**0.95**
	S1-S3	**0.78**	0.71	0.68
	S1-S4	0.68	0.66	**0.87**
	S2-S1	0	0	**0.97**
	S2-S3	0.66	0	**0.89**
	S2-S4	0.66	0	**0.84**
	S3-S1	0.75	0.70	**0.95**
	S3-S2	0.70	0.66	**0.90**
	S3-S4	0.66	0.66	**0.80**
	S4-S1	0	0	**0.97**
	S4-S2	0	0	**0.90**
	S4-S3	0	0	**0.97**
Average		0.46	0.33	**0.89**
UCSD	S5-S6	0.66	0.66	**0.96**
	S5-S7	0	0	**0.96**
	S5-S8	0.66	0.66	**0.96**
	S5-S9	0.66	0.66	**0.93**
	S6-S5	0.66	0.66	**0.96**
	S6-S7	0.66	0.66	**0.97**
	S6-S8	0.66	0.66	**0.97**
	S6-S9	0.66	0.66	**0.97**
	S7-S5	0.66	0.66	**0.96**
	S7-S6	0.66	0.66	**0.97**
	S7-S8	0.66	0.66	**0.97**
	S7-S9	0.66	0.66	**0.90**
	S8-S5	0.66	0.66	**0.96**
	S8-S6	0.66	0.66	**0.97**
	S8-S7	0.66	0.66	**0.97**
	S8-S9	0.66	0.66	**0.97**
	S9-S5	0.66	0.66	**0.96**
	S9-S6	0.66	0.66	**0.97**
	S9-S7	0.66	0.66	**0.97**
	S9-S8	0.66	0.66	**0.97**
Average		0.62	0.62	**0.96**
UCD	S10-S11	0	0	**0.81**
	S10-S12	0.58	0	**0.77**
	S10-S13	0	0	**0.84**
	S11-S10	0	0	**0.99**
	S11-S12	0	0	**0.84**
	S11-S13	0.66	0	**0.78**
	S12-S10	0	0	**1.00**
	S12-S11	0	0	**0.77**
	S12-S13	0	0	**0.64**
	S13-S10	0	0	**0.99**
	S13-S11	0	0	**0.82**
	S13-S12	0	0	**0.75**
Average		0.1	0	**0.83**
Average		0.43	0.37	**0.90**

Sensitivity Analysis

For the anomaly detection, we have used different parameter configurations, listed in Table 2.3, for all the tests, in order to demonstrate the robustness of our approach. Quantitative results for these configurations are presented in Tables 3.2. The average F-score for each dataset and the average F-scores for all datasets associated with each configuration are reported. The improvement in performance can be noticed from configuration C1 to Configuration C15 in cases of averages for each dataset and the averages for all datasets. The performance of our method, regarding anomaly detection, does not change significantly by changing other parameters except the block size. To conclude, quantitative results demonstrate that the proposed approach is robust enough to detect anomalous occurrences in video sequences.

We also plotted the averages along with standard deviations for all datasets corresponding to each configuration in Fig. 3.3, where consistent variations from averages can be noticed from C1 to C12. However, the variations from averages are reduced in cases of C13 to C15, arising due to the change in the block size.

3.3 Gaussian Mixture Model

We propose an approach for anomaly detection in term of panic situation. For this purpose, we adopt GMM to learn the behavior of motion features extracted from the particles instead of modeling the values of all the pixels as a mixture of Gaussians. These motion features are exploited to learn repetitive variations of crowd scenes for GMM, which models the normal behavior distribution. If each particle resulted from a particular behavior, a single Gaussian would be sufficient to model the motion feature of it, while accounting for surrounding noise. However, in practice, multiple surfaces

Table 3.2: Quantitative analysis for our method based on different configurations is provided in anomaly detection regarding PETS2009, UCSD, and UCD datasets, respectively. The average F-score for each dataset and the average F-scores for all the datasets are provided.

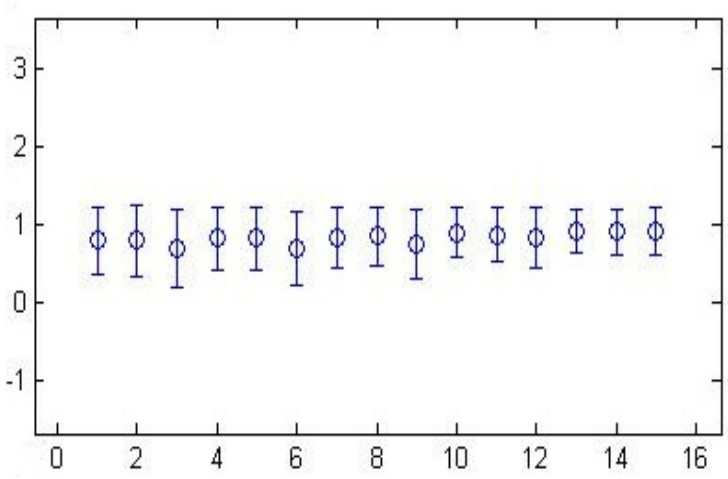
Dataset	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**	**C15**
PETS2009-Average	.75	.76	.69	.76	.77	.71	.84	.84	.83	.88	.87	.79	.89	.93	.93
UCSD - Average	.89	.93	.72	.93	.95	.79	.89	.91	.79	.96	.96	.95	.96	.96	.96
UCD - Average	.65	.60	.60	.66	.69	.63	.69	.71	.84	.74	.69	.65	.83	.79	.77
Average	.79	.78	.68	.81	.81	.68	.82	.84	.73	.88	.86	.83	.90	.90	.90

often appear in the view frustum of a particular particle. Therefore we use multiple adaptive Gaussians to approximate this process. At each frame the parameters of the Gaussians are updated, and the Gaussians are evaluated using a simple heuristic to hypothesize, which are most likely to be part of the distribution representing the normal crowd behavior.

3.3.1 Extracting motion features

As discussed before, the GMM is adopted to learn the behaviour of motion features extracted from the particles, therefore, a grid of particles is disposed on the video frame, which is repeatedly initialized over a temporal window of a video sequence as shown in Fig. 3.4.

Motion features, defined in terms of velocity magnitudes, are extracted by tracking the particles using the pyramidal Lucas-Kanade optical flow [66]. We do not consider the particles having motion features with low magnitudes.



(a)

Figure 3.3: Results of the pedestrian flow segmentation (a) and anomaly detection (b) with respect to different configurations. Each 'o' symbol presents the average calculated over all video sequences of three datasets. Standard deviations are also plotted representing variations from the averages.

3.3.2 Crowd model

Our crowd model deals robustly with repetitive motions of scene elements arising from crowd dynamics. According to the GMM framework, every new motion feature is checked against the existing distributions for that particle, and incorporated into the distribution if a match is found, or, otherwise, forms a new distribution indicating a new cluster. This forms the basis of our adaptive model.

At any time t , what we know is the history of the motion features (in term of velocity magnitude v) of a particle at location $(i, j)'s$:

$$H_{velocity}\,(i, j, n) = \{v_0, v_1, ..., v_t\} \tag{3.4}$$

For each particle at location (i, j) , the crowd model at time t stores K Gaussian distributions, along with their weights $wt_{K,t}$. This is called the Gaussian mixture $\Omega(i, j, t)$, which can be represented by the set as:

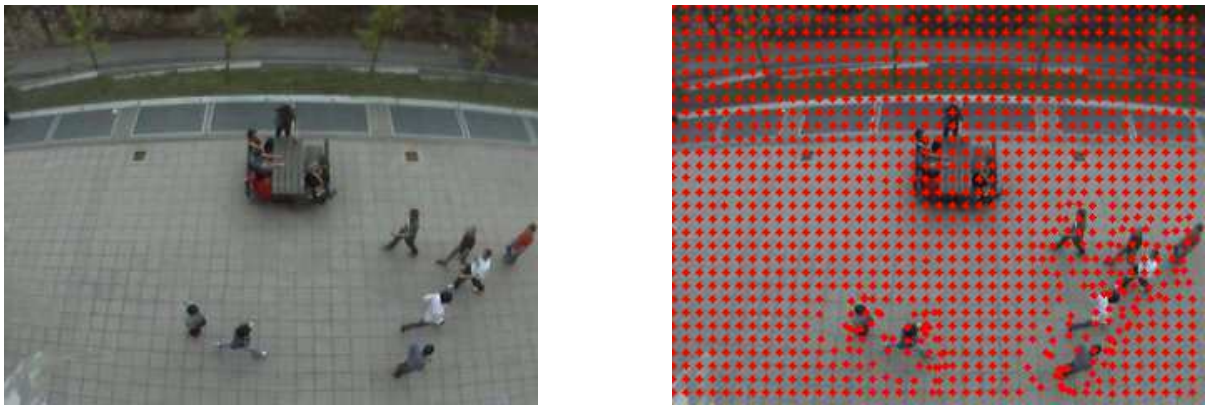


Figure 3.4: Particles initialization. Frame from video sequence (Left); frame from video sequence with particles driven (Right).

.

$$\Omega\left(i,j,t\right) = \{wt_{0,t}.G_0, wt_{1,t}.G_1, ..., wt_{k,t}.G_k\} \tag{3.5}$$

where $G_K = N(\mu, \sigma)$ are normal distributions for $K = \{0\ ,\ .....\ ,\ k\}$. The crowd model (C_m) at time t, can be represented by an $m \times n$ matrix of Gaussian mixtures $\Omega(i, j, t)$, where $0 \leq i \leq m$ and $0 \leq j \leq n$.

$$\begin{pmatrix} \Omega(0,0,t) & \dots & \dots & \Omega(m,0,t) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \Omega(0,n,t) & \dots & \dots & \Omega(m,n,t) \end{pmatrix}$$

At time $t = 0$, we start with the empty crowd model where Gaussian mixtures are $\Omega(i, j, 0) = \{wt_{0,0}\ .\ G_0\ ,\ .....\ ,\ wt_{k,0}\ .\ G_k\}$, $wt_{0,0} = wt_{1,0} = wt_{k,0} = 0$, and $G_0 = G_1 = G_k = N(0\,,0)$. Since, we use more than a single Gaussian for each particle, therefore, each Gaussian is assigned an individual weight. A weight wt is a value that indicates how often the motion feature has occurred for the particle, in the past. A new motion feature observed is given a low weight, whereas, a motion feature that occurs frequently gradually attains a high weight.

Based on the persistence and the variance of each Gaussian distribution, we determine which Gaussians can be associated to the crowd model. Consider a particle p at location (i, j) at time $t = t_0$ where $v(p)$ is its motion feature, the following criteria is evaluated for any distribution in the Gaussian mixtures:

$$\exists G_m(\mu, \sigma) \ni |v(p) - \mu| \leq n\sigma \tag{3.6}$$

where $G_m \ \epsilon \ \Omega(i, j, t_0)$ and $0 \leq m \leq k$. If the condition is satisfied, then the weight, the mean and the variance are updated for the matched distribution G_m as:

$$wt_{m,T} = (1 - \alpha).wt_{m,T-1} + \alpha \tag{3.7}$$

$$\mu_T = (1 - \rho).\mu_{T-1} + \rho.v(p) \tag{3.8}$$

$$\sigma_T^2 = (1 - \rho).\sigma_{T-1}^2 + \rho.(v(p) - \mu_T)^2 \tag{3.9}$$

where α is the weight-learning rate, and ρ is the mean/variance-learning rate. The weight-learning rate is the rate at which new motion features of a particle should be incorporated into the existing model. To this end, a low weight-learning rate means that the new motion feature will be incorporated slowly into the model. For the unmatched distributions G_n, where $n \neq m$, the weight is updated but mean and variance remain unchanged.

$$wt_{n,T} = (1 - \alpha).wt_{n,T-1} \tag{3.10}$$

$$\mu_T = \mu_{T-1} \tag{3.11}$$

$$\sigma_T^2 = \sigma_{T-1}^2 \tag{3.12}$$

Furthermore, for a particle we consider that there is no matching distribution which is evaluated according to following criteria:

$$\forall G_n(\mu,\sigma)\epsilon\Omega(i,j,T) \ni |v(p)-\mu|>n\sigma \tag{3.13}$$

If the condition is satisfied, then the least probable distribution i.e. with the minimum $\frac{wt}{\sigma}$ ratio is replaced with a new distribtution with the mean set to $v(p)$ and a high variance.

First, the Gaussians are ordered by the value of $\frac{wt}{\sigma}$. This value increases both as a distribution gains more evidence and as the variance decreases. After re-calculating the parameters of the mixture, it is sufficient to sort from the matched distribution towards the most probable normal crowd distribution, because only the relative value of matched models will have changed. This ordering of the model is a list, where the most likely normal behavior distributions remain on top and the less probable ephemeral normal crowd distributions lean towards the bottom and are finally replaced by new distributions. Then, the first C distributions are chosen as the crowd model satisfying the condition as:

$$C = argmin_c\left(\sum_{k=1}^{c} wt_k > Th\right) \tag{3.14}$$

where Th is a measure of the minimum portion of the data that should be accounted for by the crowd model. This takes the best distributions until a certain portion, Th, of the recent data has been accounted for. If a small value for Th is chosen, the crowd model is usually unimodal. If Th is higher, a multi-modal distribution caused by crowd dynamics (e.g. people walking, and people chatting etc.) results in more than one modality being included in the crowd model.

3.3.3 Experimental results

We evaluate the performance of our method on both UMN [1] as well as our own UCD [54] datasets. The UMN dataset consists of four video sequences acquired in both indoor and outdoor scenes. All these sequences represent an escape panic scenario and hence, they start with the normal behavior frames followed by the abnormal situation. Normal scene is identified as crowd walking while abnormal situation is identified as sudden change in terms of crowd running in random directions.

As we adopt GMM to follow the behavior of motion features extracted from the particles, a grid of particles is overlayed on the video frame, which is repeatedly initialized after 3 frames. Motion features are extracted in terms of velocity magnitudes by tracking the particles using the pyramidal Lucas-Kanade optical flow [66]. For extracting the motion features, the particle density (i.e. the number of particles) is kept constant at 12.5% of the number of pixels i.e. a particle every $8x8$ pixels.

Fig. 3.5 shows the normal and abnormal crowd behavior frames from the UMN dataset in the top row and bottom row, respectively. The people in the video sequences in the top row walk in random directions. In the third and last columns in the top row, there are few red circles, which is the result of noise arising from illumination and light changes. In the bottom row, people starts running resulting in an anomaly as represented by red circles. The red circles identify particles not fitting in the GMM crowd model, thus shown as anomolous.

Similarly, Fig. 3.6 demonstates the results obtained on our UCD dataset. Frames from video sequences where crowd show normal behavior are shown in the top row while the bottom row shows examples where part of the crowd shows an anomalous behavior. In the first and second columns, three students are running from left to right and right to left, respectively,

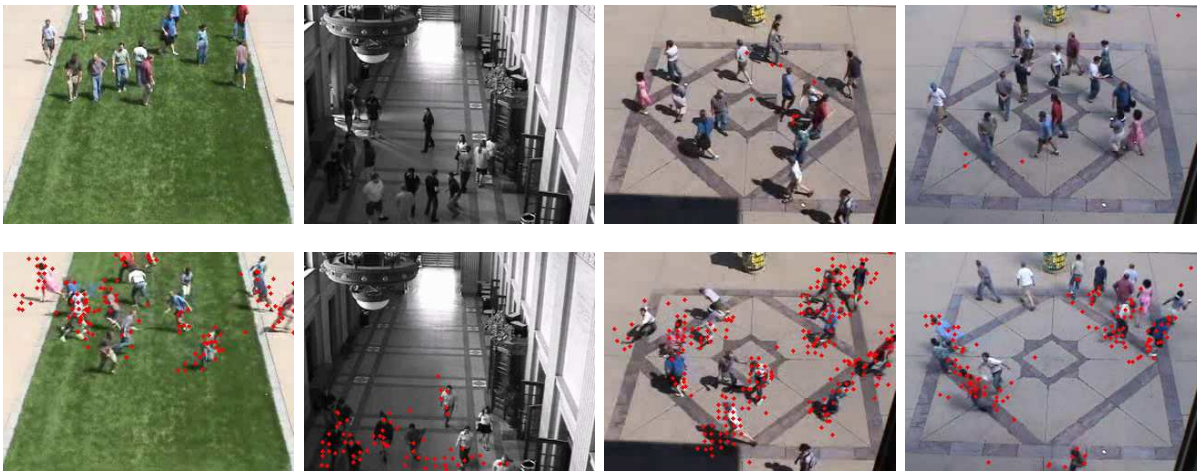


Figure 3.5: Anomaly detection in UMN dataset. Frames taken from four video sequences representing normal behavior of crowd (first row); frames taken from four video sequences representing abnormal behavior of crowd (second row).

.

in the mid of the crowd. Also the behavior is identified as anomalous by particles annotated in red. In the third column, a student is running from the bottom left corner to the top right corner and is identified as anomolous as well since deviating from the learned crowd model. Furthermore, four students running from left to right are successfully identified as anomolous in the last column.

3.4 GoodFeatureToTrack and MLP

To consolidate the anomaly detection in term of panic situation, we adopt multi-layer perceptron (MLP) feed-forward neural network to learn the behavior of motion features extracted from the corner features [49] instead of considering the values of all the pixels. The motion features are exploited to learn the abrupt changes of crowd scenes represented by corner features, thus modeling the abnormal behavior of the crowd. A single motion feature extracted from an arbitrary corner feature is not sufficient to model the

Figure 3.6: Anomaly detection in our UCD dataset. Frames taken from four video sequences representing normal behavior of crowd (first row); frames taken from four video sequences representing abnormal behavior of crowd (second row).

.

abnormal behavior of crowd due to surrounding noise. Therefore, for each corner feature we extract a set of motion features to robustly model the abnormal behavior of the crowd.

3.4.1 Extracting features

The MLP neural network is adopted to learn the motion features, in terms of velocity magnitudes of the corner features. In our approach, we selected corner features, since they represent dominant motion parts in the scene. Therefore, the consistency and accuracy in tracking are higher in crowd scenes representing complex motion. These corner features are extracted from the video frame as shown in Fig. 3.7. Motion information, defined in terms of velocity magnitudes, are extracted at regular intervals of K frames by tracking the corner features using the pyramidal Lucas-Kanade optical flow. The collected motion features are stored to construct a feature vector for each particle as formulated in Eq. (3.15).

$$V_p = \{v_1.....v_N\} \tag{3.15}$$

The objective of this processing stage is to filter out the motion features that do not contribute to the identification of the crowd anomaly detection. We do not consider the corner features having motion information with low magnitudes.

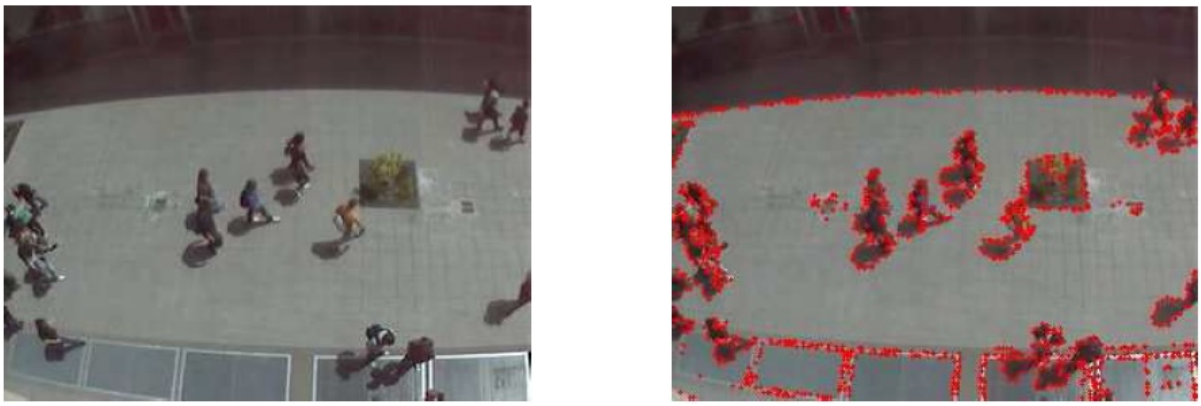


Figure 3.7: Corner features initialization. Frame from a UCD video sequence where students are walking from left to right (Left column); the same frame from UCD video sequence with corner features driven (Right column)

.

3.4.2 MLP neural network

In order to properly weight the motion features, we have trained an MLP neural network. The motivation for exploiting MLP is in its substantial ability, through backpropagation, to resist to noise, and the dexterity to generalize. The motion properties, extracted from the corner features, are fed as an input to the MLP.

The output y is obtained by propagating the motion features as an input vector through the hidden layers, as shown in Eq. (3.16), where y^0 is an input vector.

$$y^0 \xrightarrow{W^1,b^1} y^1 \xrightarrow{W^2,b^2} ..... \xrightarrow{W^L,b^L} y^L \tag{3.16}$$

In MLP networks, there are $L+1$ layers of neurons, and L layers of weights. During the training stage, the weights W and biases b are updated so that the actual output y^L becomes closer to the desired output d. For this purpose, a cost function is defined as in Eq. (3.17).

$$E(W,b) = \frac{1}{2}\sum_{i=1}^{nl}(d_i - y_i^L)^2 \tag{3.17}$$

The cost function measures the squared error between the desired and actual output vectors and the backpropagation is gradient descent on the cost function in Eq. (3.17). Therefore, during the training stage, weights and biases updated. The backpropagation algorithm begins with the forward pass where the input vector y^0 is converted into output y^L. The difference between the desired output d and the actual output y^L is computed to estimate the error. During the backward pass, the estimated error at the output units is propagated backwards through the entire network. The weights and biases are updated using the results of the forward and backward passes. The learned weights and biases from the training stage are used to predict the corner features associated with the abnormal crowd behavior from the input motion information during testing.

3.4.3 Experimental results

For the purpose of performance evaluation, we carred out experiments on both UMN [1] and our own UCD [54] datasets. The neural network we exploited in our approach consists of one input layer, two hidden layers and one output layer. The input layer consists of three neurons, each hidden layer consists of three neurons, and a single neuron is allocated to the

output layer. We adopt an MLP neural network to understand the behavior of motion features extracted from the corner features. These corner features are tracked by the Lucas-Kanade optical flow, and the motion features of each particle in term of velocity magnitude are extracted. For each particle, the motion features consist of a vector of N = 3 observations, where each element of the vector corresponds to the motion information extracted every K = 5 frames.

Fig. 3.8 shows the normal and abnormal crowd behaviors frames from the UMN dataset in the top two rows and bottom two rows, respectively. The crowd in the video sequences in the top two rows walk in random directions. The first row demonstrates the results of our approach described in Section 3.3 and the second row demonstrates the results of our approach in Section 3.4. In the top row, there are a few red circles, representing the noise susceptibility of the method 3.3 comparing to the method in Section 3.4 in the second row. In the bottom two rows, the crowd starts running resulting in an anomaly as represented by red circles. The red circles are more pronounced in the method in Section 3.4 as compared to the method in Section 3.3 in the third row.

Similarly, Fig. 3.9 shows the results obtained on our UCD dataset. Frames from video sequences where crowd shows normal behavior are shown in the top two rows, while the bottom two rows show examples where only part of the crowd shows an anomalous behavior. The first and third rows represent results from the method in Section 3.3 where the second and last rows represent results from the method in Section 3.4. There are a few red circles in the first column of the first row denoting again the noise susceptibility of the method in Section 3.3. In the first and last columns (bottom two rows), four students are running from left to right and right to left, respectively, in the mid of the crowd. In the second column of bottom two rows, a student is running from bottom left to top

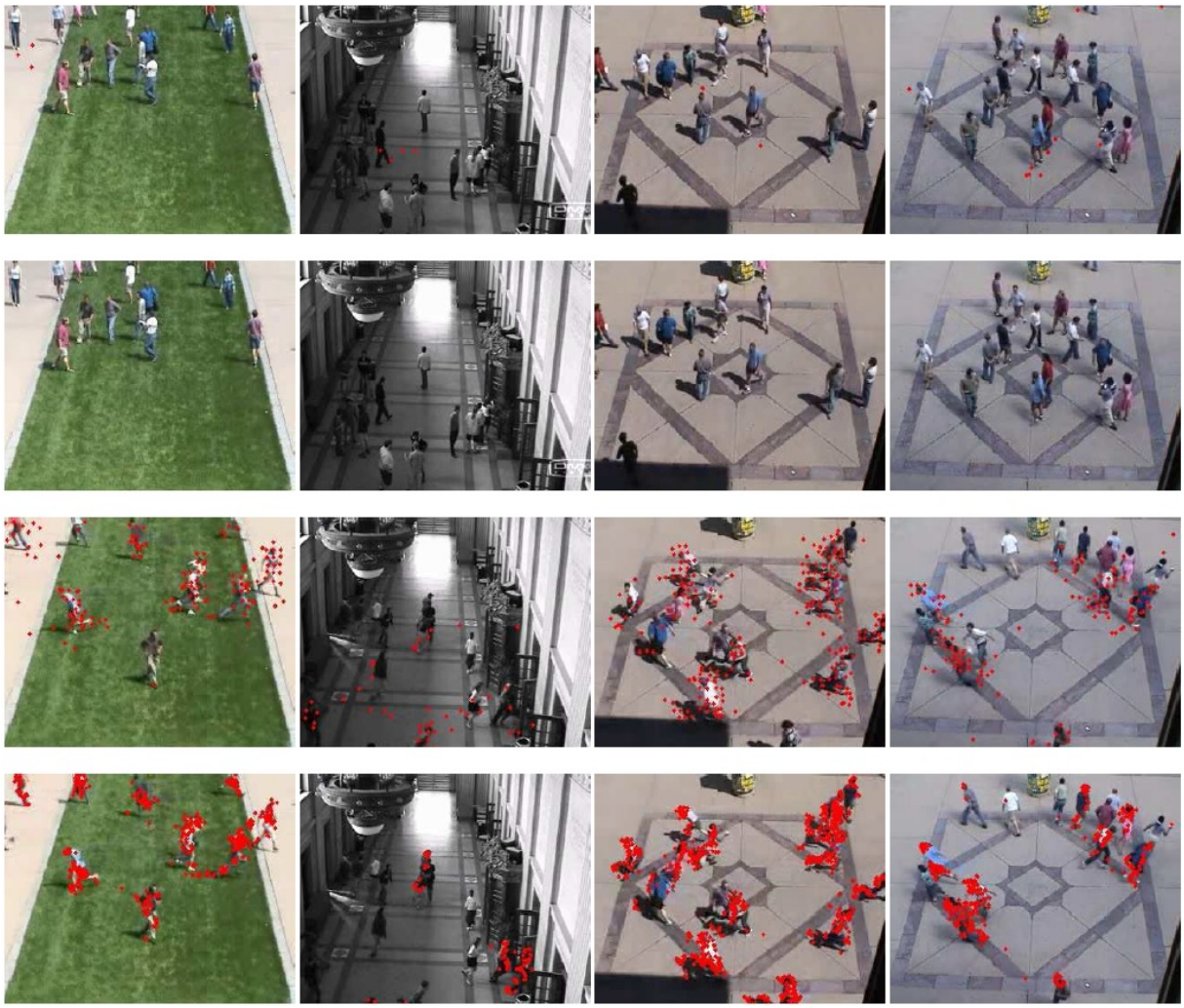


Figure 3.8: Anomaly detection in UMN dataset. Frames taken from four video sequences representing normal behavior of crowd for the reference method and our proposed method (first and second rows, respectively); frames taken from four video sequences representing abnormal behavior of crowd for the reference method and our proposed method (third and fourth rows, respectively).

right. Furthermore, four students are running from left to right in the third column. Also in these cases, the behaviors are identified as anomalous, by the particles annotated in red. However, the annotation in term of red circles is consolidated in case of the method in Section 3.4 comparing to the method in Section 3.3, representing the robustness of the method in Section 3.4.

Figure 3.9: Anomaly detection in our UCD dataset. Frames taken from four video sequences representing normal behavior of crowd for the reference method and our proposed method (first and second rows, respectively); frames taken from four video sequences representing abnormal behavior of crowd for the reference method and our proposed method (third and fourth rows, respectively).

Chapter 4

Behavior Classification

This chapter presents state of the art and our proposed methods regarding behavior classification. In particular, particle-driven and hybrid approaches are presented.

4.1 State of The Art

The State of the art we present it this section also fall under the category of high density flow analysis. Rodrigues et al. [45] propose a tracking approach by minimizing an energy function to jointly optimize the estimates of the density and locations of individual people in the crowd. Ge et al. [18] detect small groups of people traveling together in the crowd. A hierarchical clustering algorithm is exploited by considering a generalized, symmetric Hausdorff distance defined with respect to pairwise proximity and velocity. However, these approaches require training and multiple target tracking. The method proposed by Solmaz et al. [50] classifies crowd behaviors, in terms of lane, rings/arches, bottleneck, fountainhead, and blocking, using time integration of the dynamical system defined by the optical flow. However, this approach results sometimes in significant errors arising from its inability to deal with crowd dynamics in low to medium density crowd scenes. Furthermore, the approach works offline due to com-

putational overheads, thus making it inapplicable to real-time applications.

4.2 A particle-driven approach

In our work, we address the problem of crowd behavior analysis by proposing an approach based on temporal features and spatial features, which does not require neither tracking nor training. The temporal features represent particles trajectories over a fixed interval of time whereas the spatial features represent density of particles in the predefined proximity. Unlike the method in [50], our approach works online, since both features are computationally affordable. We dispose a grid of particles over the video frame and advect them over a fixed time window using the optical flow technique. We subsequently collect spatio-temporal features related to particles moving within a predefined region of interest.

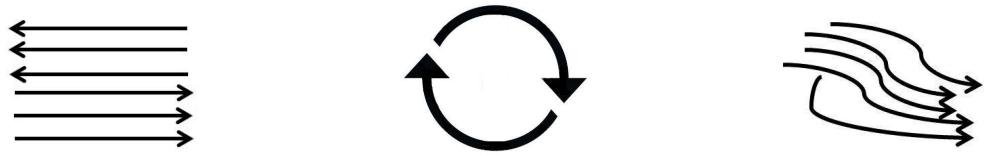


Figure 4.1: Crowd behaviors. Crowd individuals moving in straight directions representing lanes (first column); individuals moving in curved directions representing rings (middle column); individuals from different points accumulating at single location representing bottleneck (last column).

4.2.1 Crowd behaviors

The crowd behaviors identification is carried out starting from a manual selection of the region of interest (ROI), and extracting the spatio-temporal features associated to the particles inside the ROI. To this aim, a grid of particles is initialized on the first video frame, and tracked over a fixed interval of time using the pyramidal Lucas-Kanade optical flow [66]. Our

approach is targeted at the identification of three major crowd behaviors as listed below as:

Lane. In crowd situations, lanes formation take place when individuals are uniformly moving in undeviating and straight directions. In lanes, individuals move with comparable speed and direction of motion, so as to avoid collisions with their neighbors.

Ring/Arch. Motion in the ring (or arch) is characterized by a curved or circular direction. Ring/arch formations take place in typical scenes such as traffic or pedestrian flowing, when following road paths or when avoiding obstacles.

Bottleneck. It encompasses the presence of a narrow passage, through which crowd individuals from many places intersect. The bottleneck represents the condition where many individuals try to go through an exit, in ordinary situations (as entrance gates in crowded places) or in presence of potentially dangerous events, such as a panic situation.

One sample scenario for each of the mentioned behaviors is shown in Fig. 4.1.

4.2.2 Particle advection

We dispose a grid of particles over the first frame of the video sequence. Each particle is associated with a spatial position on the image plane as formulated in Eq. 4.1.

$$P(t) = \begin{bmatrix} x_1(t) & y_1(t) \\ \vdots & \vdots \\ x_N(t) & y_N(t) \end{bmatrix} \tag{4.1}$$

P is a function representing the state of the grid of particles, where $x_i(t)$ and $y_i(t)$ represent the coordinates of the i-th ($i = \{1 \dots N\}$) particle at each time instant. Particles are initially positioned so as to be equally

spaced one from the other, and are then advected over a temporal window on a frame-by-frame basis using the Lucas-Kanade optical flow as formulated in Eq. 4.2, where OF stands for optical flow.

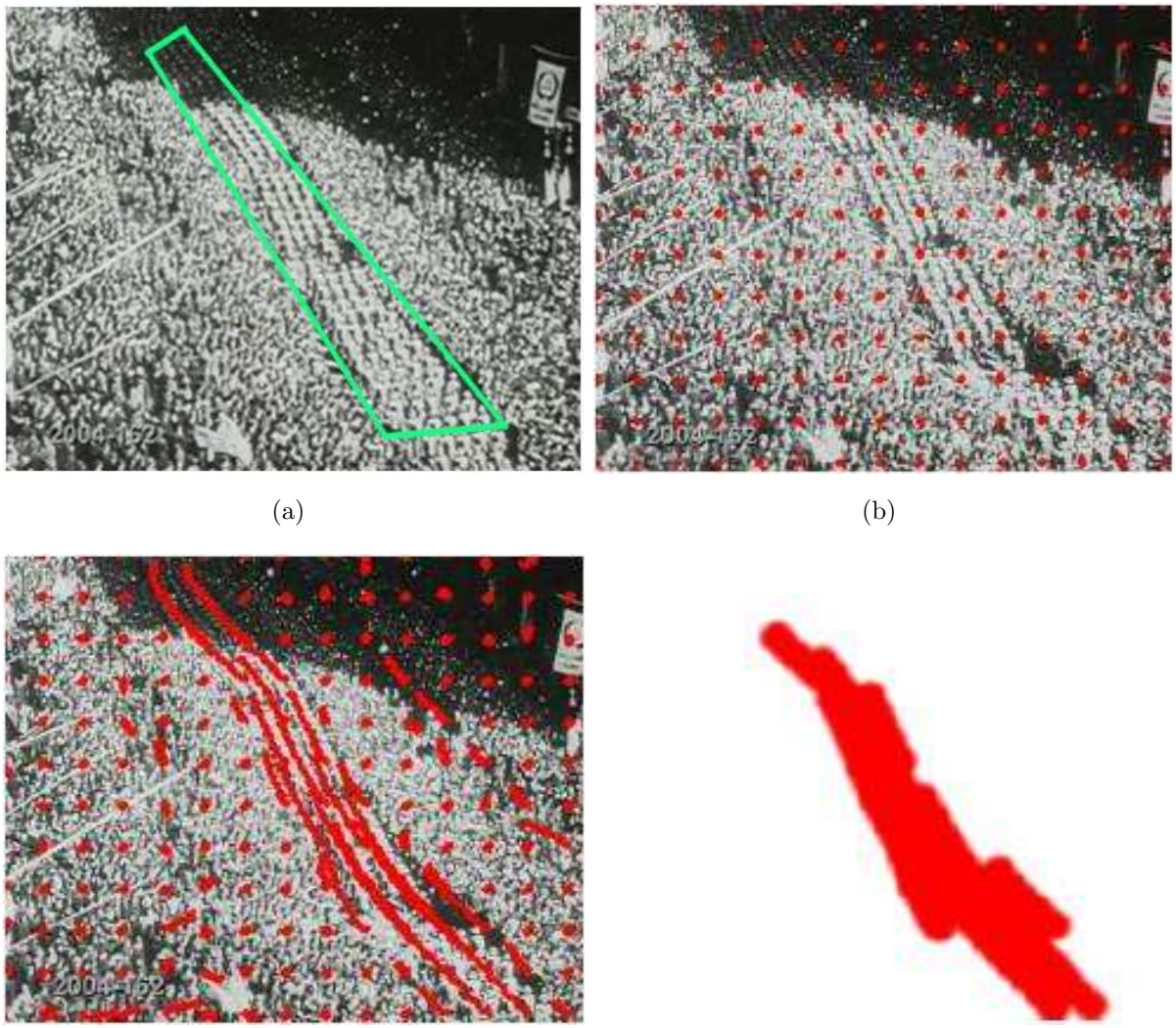
$$\forall i \in \{1, \ldots, n\},\ F(p_i(t+1)) = OF(F(p_i(t)) \tag{4.2}$$

Considering the definition of the ROI, in which we would like to analyze the crowd behavior, only the particles located within the ROI are advected. Our motivation for manual ROI selection comes from the observation that in most videos only a limited portion of the observable space is typically of interest, especially in traffic scenes (lanes, parking lots) and surveillance videos.

We compute the optical flow on a frame by frame basis and the particles are exalted according to the optical flow. At the end of particles advection, the particle paths are highlighted, discarding those particles with low motion. Additionally, we apply a Gaussain filter to the highlighted path, to consolidate the motion map. The workflow of ROI selection and path highlighting is shown in Fig. 4.2, where the ROI annotated in green is drawn manually for both the proposed method and the reference method [50].

4.2.3 Behaviors identification

In the context of behavior identification under complex crowded scenes, the analysis of the spatio-temporal features, and the definition of the metrics to distinguish among different behaviors, are key elements to obtain reliable results. Given the impossibility of tracking single entities moving in the video, the analysis of the optical flow has demonstrated to be a very good baseline to start. The features we have considered are defined pairwise for each particle and include a spatio-temporal component related to the particles position and motion, jointly with the density information of



(a) (b)

(c) (d)

Figure 4.2: ROI selection. Drawing a region of interest (a); A grid of particles disposed over the video frame (b); Particle advection (c); Highlighted paths of particles.

the surrounding particles. The spatio-temporal features represent the trajectories traveled by the particles. However, trajectories may significantly vary also in the same scenario due to illumination changes, resulting into noisy paths. In Fig. 4.4, an example of a particle advection is shown. This is a representative particle showing the accumulative behavior of particles in a ROI. The first column denotes noise-free advection of the particle from the initial to the final location (annotated in green) representing the lane/bottleneck behavior of crowd. However, the same behavior is represented by noisy advection of the particle, as shown in the middle column. The third column represents the ring behavior of a crowd as the particle is advected in the curved shape.

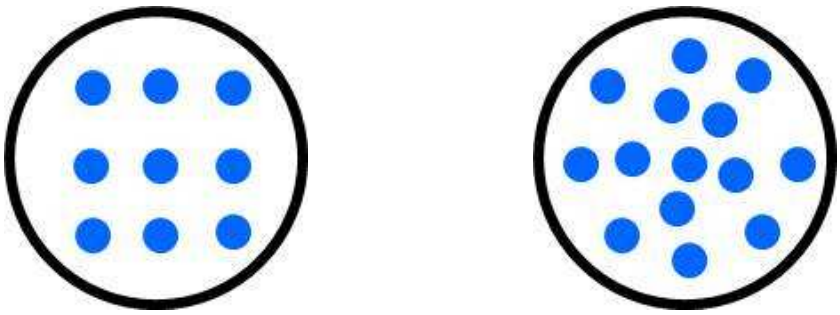


Figure 4.3: Densities of particles at the end of particle advection. Density of particles in the proximity remains the same as before representing lane or arch (left); density of particles increased in the proximity representing bottleneck (right).

In order to compensate this problem, first we accumulate the particles trajectories at the end of the advection phase as formulated in Eq. 4.3:

$$\forall i \in \{1, \ldots, n\},\ T1 = \int_{Initial}^{final} F(p_i(t))dt \tag{4.3}$$

where each particle trajectory is computed in terms of distance from initial to final location (locations are annotated with red circles in Fig. 4.4).

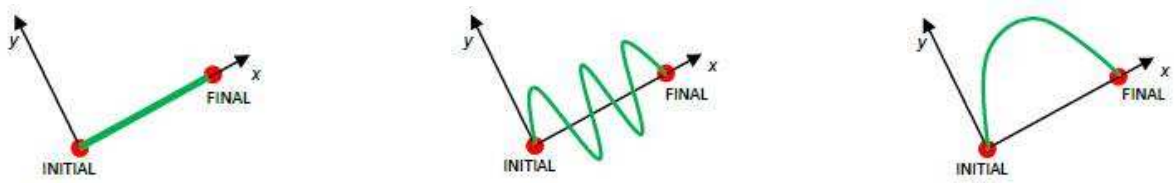


Figure 4.4: Particle advection. Crowd individuals moving in straight directions representing lane or bottleneck (first column); Individuals moving in a noisy straight direction representing lane or bottleneck (middle column); Individuals moving in curved direction representing ring (last column).

The same procedure is subsequently repeated to accumulate the particles trajectories. However, each particle trajectory results as the concatenation of trajectories calculated over K consecutive frames as formulated in Eq. 4.4.

$$\forall i \in \{1, \ldots, n\},\ T2 = \int_{Initial}^{final} \int_{k=1}^{K} F(p_i(t))dt \tag{4.4}$$

In order to identify the category of behaviors, we calculate the non-trivial magnitude of noise as in Eq. 4.5.

$$I = |T1 - T2| \tag{4.5}$$

Hence, we can infer either of the two categories of behaviors from the temporal features as formulated in Eq. 4.6.

$$Behavior = \begin{cases} L \ or \ B \ \ \text{if} \ \ I \approx Th \\ \\ A \ or \ B \ \ \text{if} \ \ I \gg Th \end{cases} \tag{4.6}$$

For the sake of simplicity, lane, arch, and bottleneck are represented by L, A, and B respectively. In Eq. 4.6, Th is set to 5 determined empirically.

The temporal features, all alone, cannot distinguish among the three crowd behaviors, and the particle density information in a predefined neigh-

borhood is also exploited. To this aim, the identification process comprises of two stages:

- Temporal features are collected at the end of particle advection to identify lane/bottleneck or arch/bottleneck;
- Spatial features are collected to identify the particular behavior.

Primarily, inference from the spatial features combined with the temporal features is interpreted in two stages. In the first stage, the density of particles (i.e. number of particles) within a circle with radius r, is computed before the particle advection as in Eq. 4.7:

$$S1 = \sum_{i=1}^{n} p_i(t) \tag{4.7}$$

where n is the number of particles. In the second stage, the same process is repeated after the particle advection to compute $S2$. For instance, the density of the particles is almost the same at the end of particle advection in the first column of Fig. 4.3, hence representing the lane or arch behavior of crowd. Furthermore, the density of the particles increased at the end of advection in the second column representing the bottleneck.

Subsequently, we calculate a ratio between $S1$ and $S2$ as in Eq. 4.8

$$Ra = S1/S2 \tag{4.8}$$

The behavior is determined by fusing the two features pairwise as in Eq. 4.9.

$$Behavior = \begin{cases} L \;\; \text{if} \;\; I \approx Th \; and \; Th_h > \;\; Ra > Th_l \\ A \;\; \text{if} \;\; I \gg Th \; and \; Th_h > \;\; Ra > Th_l \\ B \;\; \text{if} \;\; Ra \leq Th_l \end{cases} \tag{4.9}$$

Where $Th_h = 1 + \varepsilon$ and $Th_l = 1 - \varepsilon$. The ε represents the stability of particles spatially. This stability is associated with the movement of individuals in the video sequences. Therefore, the choice of these values is empirical and we set it to 0.75 and 0.5 for videos with normal motion and swift motion, respectively.

An example applied to a video displaying a lane is shown in Fig. 4.5, where army is parading on a thoroughfare.

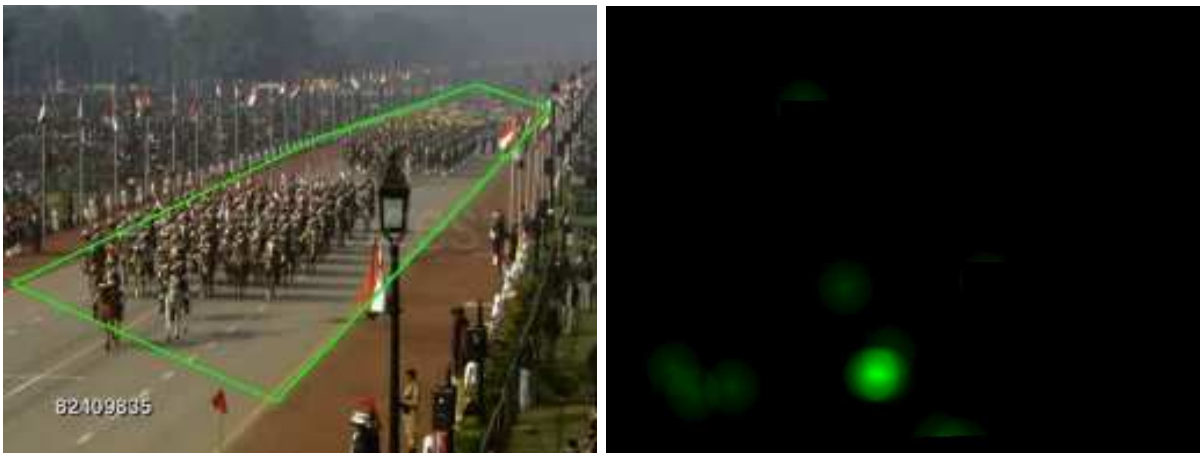


Figure 4.5: Lane. Drawing region of interest manually (first column); Density of particles converged at the end of particle advection (second column).

4.2.4 Experimental results

To validate the performance of our approach, we have conducted the experiments on a set of 22 crowd video sequences extracted from PETS2009 [42], UCSD [30], and our own UCD datasets. We have also considered video sequences from [50]. All video sequences extracted from the aforementioned datasets consist of low to medium density crowds. For the extraction of the spatio-temporal features for each particle, the resolution of the grid is set to one sixteenth of the resolution of the video frame. Additionally, each video sequence is partitioned into segments of fixed-length, set at 160

frames (about 6 seconds depending on the frame rate). However, movement of individuals in 6 video sequence is very swift, representing strong transitions between consecutive frames. Therefore, these video sequences are partitioned into segments of 60 frames each instead. There numbers are determined empirically as shown in Fig. 4.6 and Fig. 4.7, where different thresholds for all video sequences from 60 to 200 are tested at a step size of 20 frames. We noticed that for video sequences with normal motion of individuals, the performance improvement is significant in terms of behaviors detection when the segment size is set to 160 frames. Similarly, the performance achievement is significant with 60 frames for video sequences with swift motion of individuals.

Figure 4.6: Crowd sequences with normal motion. The first and the third video sequences represent traffic flow and the middle one represents marathon flow. The threshold set to 160 correctly detects the behaviors.

The qualitative results are divided into two categories of video sequences. Fig. 4.8 refers to video sequences consisting of pedestrian flows, while Fig. 4.9 reports the results obtained from video sequences consisting of marathon and traffic scenes. In both figures, first columns present the samples frames taken from the original video sequences, middle columns illustrate the density of particles after applying the Gaussian filtering at the end of particle advection, and last columns present peak extraction. For each particle, a two-dimensional Gaussian filter, with variance 1 and size 11 x

Figure 4.7: Crowd sequences with swift motion. The first video sequence represents the traffic flow and the second video sequence represents the gathering of people from different directions. The third video sequence represents crowd of people entering a gate. The threshold set to 60 correctly detects the behaviors.

11, is applied to reduce noise and engender a consistent density map at the end of particle advection. To extract the peak, a blob detector is applied and the centroid of the blob is recognized as a peak.

In Fig. 4.8, for conciseness reasons only two video sequences from PETS2009 dataset are depicted in the top two rows, one video sequence from UCSD dataset is depicted in the third row, and one video sequence from UCD dataset is depicted in the last row. The analysis of the extracted peaks from all the video sequences show lane behaviors as the crowd inside any ROI follows a straight path in an arbitrary direction. The video sequences depicted in Fig. 4.9 are taken from [50]. The analysis of the extracted peak for both the video sequences in the first row and the second row represent bottlenecks. All the fishes and vehicles converge to a single location in both video sequences. The third and the fourth rows represent arch behaviors, respectively.

To evaluate the performance of our approach, we compared it with the method recently proposed by [50]. The comparison of the obtained behavior detection results is shown in Table 4.1, where the second column represents the ground truth in terms of total number of occurrences associated with each behavior in the first column. The third and the fourth columns

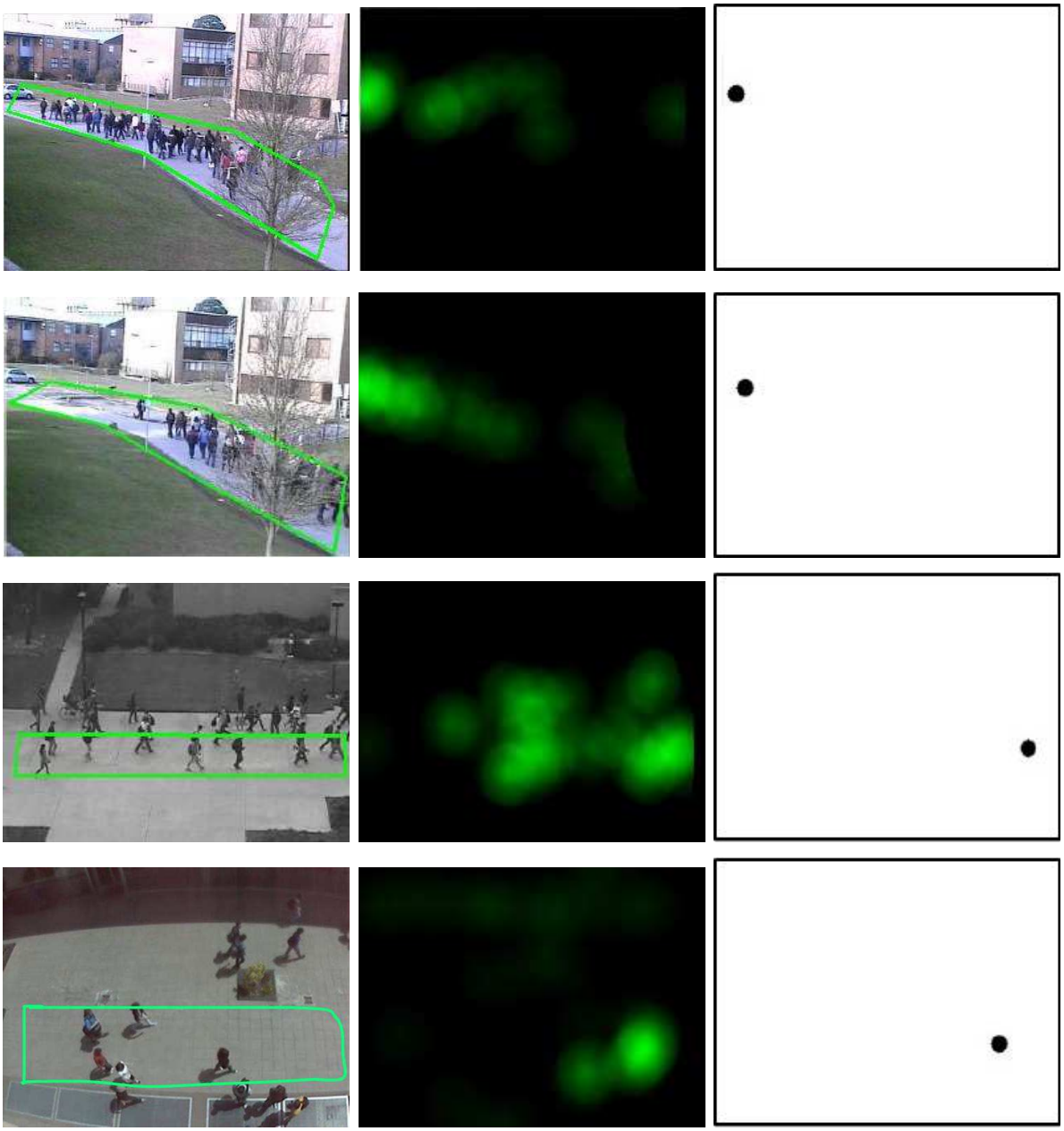


Figure 4.8: Crowd behaviors. Drawing region of interest manually (first column); Density map of particles converging at the end of particle advection (middle column); Peak extraction (last column).

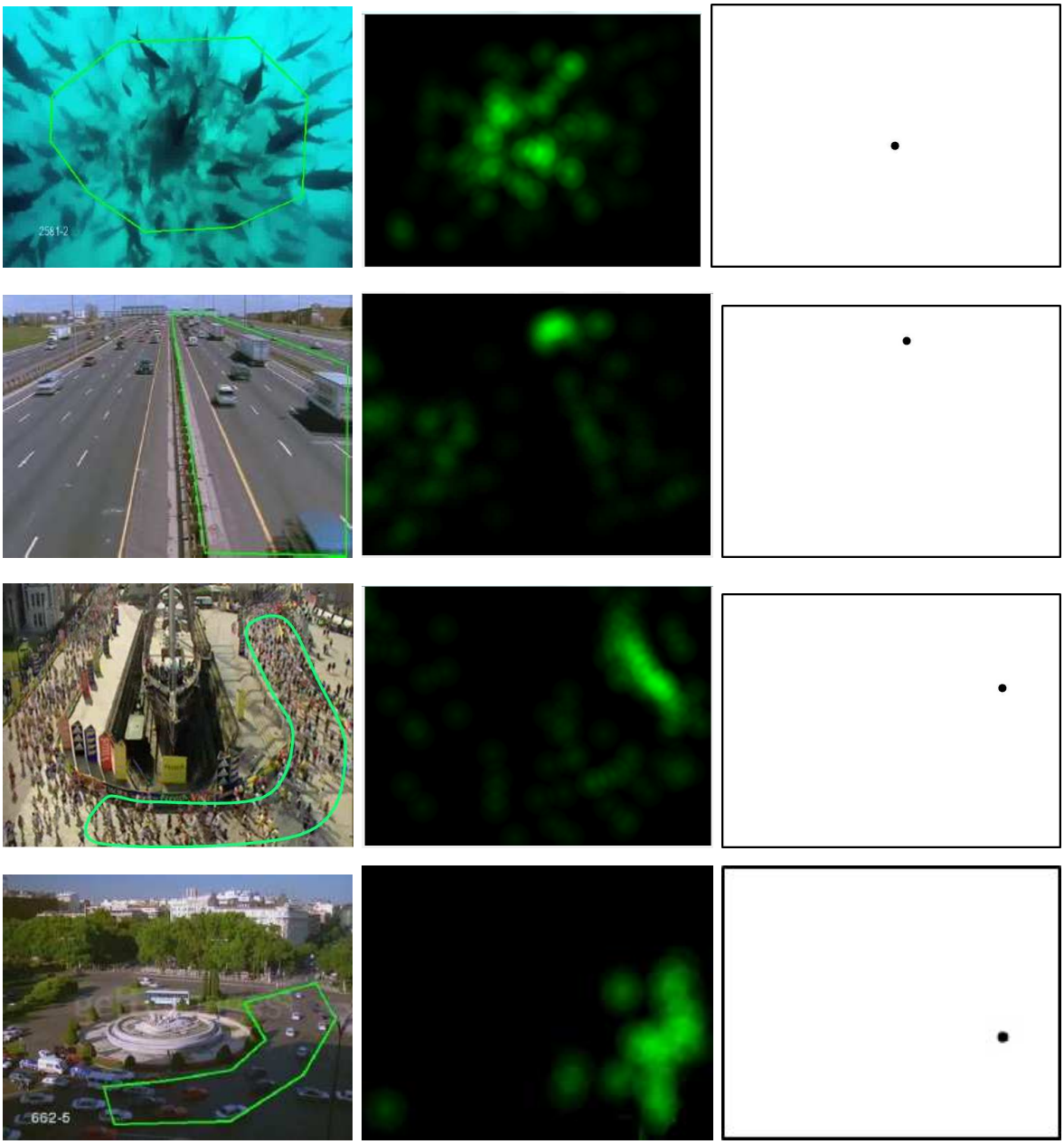


Figure 4.9: Crowd behaviors. Drawing region of interest manually (first column); Density map of particles converging at the end of particle advection (middle column); Peak extraction (last column).

Table 4.1: Comparison of our method with the reference method in behavior detection.

Behaviors	Total	Our method	Ref. method	Improvement
Lane	15	15	13	13.3%
Arch/Ring	9	2	1	11.1%
Bottleneck	6	5	4	16.6%

represent the number of accurately detected behaviors by the proposed method and the reference method, respectively. The last column shows the improvement, in term of percentage, that our approach brings over the reference approach [50].

All experiments were conducted on an Intel(R) Core(TM) i5-2400 3.10 GHz machine with 4GB of RAM. The proposed approach executes at approximately 29 frames per second. Considering the low computational complexity of the proposed approach enables it for real-time surveillance and monitoring applications.

4.3 A hybrid approach

We described in detail each stage of our proposed method based on the representation of a dynamic system. The crowd behaviors are classified in terms of lane, arch/ring, bottleneck, blocking, and fountainhead in the ROI selected manually. For this purpose, a motion flow field is extracted from video frames using the Farnback optical flow technique [17]. We then exploit thermal diffusion process [12] [63] that fuses both motion correlation among particles and motion trends of individual particles, thus transforming the input motion field into a more accurate coherent motion field. Each particle represents the position of a pixel in the video frame. Furthermore, we introduce an extended variant of social force model [65] to isolate and filter out the particles that do not contribute in the classification process.

Unlike the conventional Social Force Model [20] [19], the *Extended* variant also captures the turbulent dynamics arising from high interactions.

4.3.1 Thermal diffusion process

It describes that the energy of the particles propagate to their neighborhoods spontanously. Therefore, we exploit it to find a coherent motion flow as formulated in Eq. (4.10).

$$\frac{\partial \boldsymbol{D}_{P,l}}{\partial l} = \gamma_P^2 \left(\frac{\partial^2 \boldsymbol{D}_{P,l}}{\partial x^2} + \frac{\partial^2 \boldsymbol{D}_{P,l}}{\partial y^2} \right) + \boldsymbol{V}_P \tag{4.10}$$

where $\boldsymbol{D}_{P,l} = (\boldsymbol{D}^x_{P,l}, \boldsymbol{D}^y_{P,l})$ is the accumulated thermal energy diffused from the neighboring particles for a particle $P = (p^x, p^y)$ for l seconds. $\boldsymbol{V}_P = (v^x_P, v^y_P)$ is the motion vector of the particle P and γ_P is a constant. The first term in Eq. (4.10) boosts the spatial correlation among particles. The second term V_P is an external force added on the particle to affect its diffusion behavior while preserving the original motion patterns at the same time. Without the second term, Eq. (4.10) can be solved by:

$$\boldsymbol{D}_{P,l} = \frac{1}{wh} \sum_{S \in I, S \neq P} E_{P,l}(\boldsymbol{S}) \tag{4.11}$$

where I is the set of all particles in the predefined spatial window K, w and h are the width and height of the window. Eq. (4.11) states that the diffused thermal energy is the summation from all the neighboring particles encoding the correlation among them. The individual thermal energy $E_{P,t}(\boldsymbol{S}) = (E^x_{P,l}(\boldsymbol{S}), E^y_{P,l}(\boldsymbol{S}))$ is diffused from the neighbor particle $\boldsymbol{S} = (s^x, s^y)$ to the particle P located in the center of the window K, after l seconds as:

$$E^\beta_{P,l}(\boldsymbol{S}) = N^\beta_{\boldsymbol{S}} . e^{\frac{-\gamma_P}{l} ||\boldsymbol{P} - \boldsymbol{S}||^2} \tag{4.12}$$

where $\beta \in (x, y)$, $N_{\boldsymbol{S}} = (n^x_{\boldsymbol{S}}, n^y_{\boldsymbol{S}})$ is the current motion pattern for the

neighbor particle $\boldsymbol{S}$ and it is initialized by $N_{\boldsymbol{S}} = V_{\boldsymbol{S}}$. $||\boldsymbol{P} - \boldsymbol{S}||$ is the distance between particles $\boldsymbol{P}$ and $\boldsymbol{S}$. In this paper, we fix l to be 1 to eliminate its effect. When $\boldsymbol{V_p}$ in Eq. (4.10) is non-zero, it is difficult to get the exact solution for (4.10). Therefore, an additional term $e^{-\gamma_P|\boldsymbol{V}_S.(\boldsymbol{P}-\boldsymbol{S})|}$ is introduced to approximate the influence of $\boldsymbol{V}_S$ where γ_P is a force propagation factor. In order to prevent unrelated particles from accepting too much energy from $\boldsymbol{S}$, we restrict that only highly correlated particles will propagate energies to each other. The final individual thermal energy from $\boldsymbol{S}$ to $\boldsymbol{P}$ is formulated in Eq. (4.13).

$$E^{\beta}_{P,l}(S) = \begin{cases} N^{\beta}_{S} \times e^{-\gamma_P \|P-S\|^2} \times & \text{if } \cos(V_P, V_S) \geq \theta_c \\ e^{-\alpha_m |V_S.(P-S)|}, & \\ 0, & \text{otherwise} \end{cases} \tag{4.13}$$

where $\boldsymbol{V}_P$ and $\boldsymbol{V}_S$ are the input motion vectors of the current particle $\boldsymbol{P}$ and the neighbor particle $\boldsymbol{S}$, and $\cos(\boldsymbol{V}_P, \boldsymbol{V}_S)$ is the similarity measure conditioning that the particle $\boldsymbol{P}$ will not accept energy from $\boldsymbol{S}$ if their input motion vectors are not coherent subject to the threshold θ_c. The first term in Eq. (4.13) preserves the motion pattern of the energy source. The second term considers the spatial correlation between the source and central particles and the third term guarantees that particles along the motion direction of the heat source receives more thermal energies. An example of TDP is depicted in Fig. 4.10.

4.3.2 Extended social force model

The motion of particles is described as if they are subject to *social forces*. Social forces are a measure for the internal motivations of the individual particle to perform certain movements, and take into account the influence of the other particle surrounding it. Therefore, the force concept turns into a model based on plausible interactions among particles. According to this

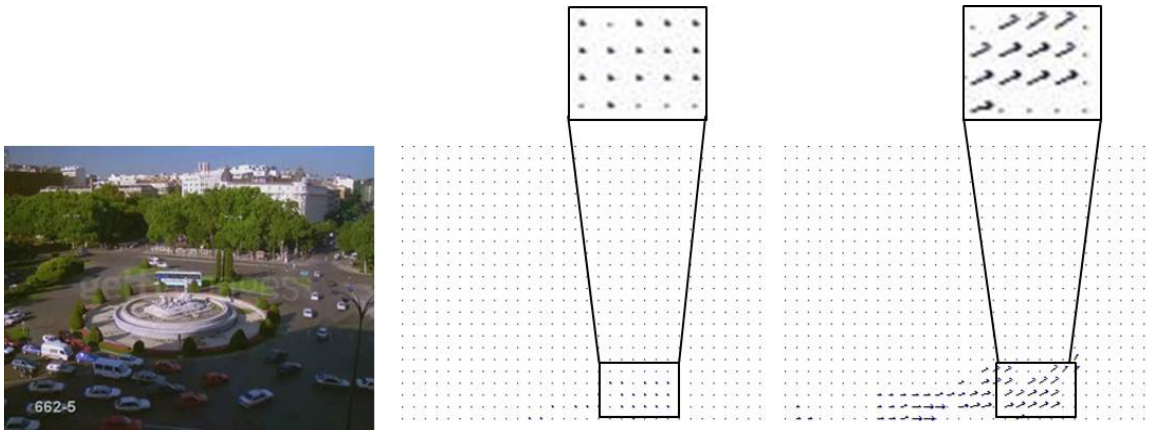


Figure 4.10: TDP. The original frame (first column); the motion flow field (second column) and the coherent motion flow field (third column) after applying TDP.

model the velocity of each particle k with mass m_k obeys to Eq. (4.14), where F_a represents the acceleration force, expressed into two major parts, namely the personal force F_p and the repulsive force F_{rep}, respectively, as in Eq. (4.15).

$$m_k \frac{dv_k}{dt} = F_a \tag{4.14}$$

$$F_a = F_p + F_{rep} \tag{4.15}$$

Here, F_p represents the attempt of a particle to seek certain goal and destination. Therefore, it is plausible to consider that each particle has a desired velocity v_k^p as in Eq. (4.16).

$$F_p = \frac{1}{\tau}(v_k^p - v_k) \tag{4.16}$$

However, for each portion of the video, the crowd motion is resembled by the movement of a particle, where current velocity v_k differs from the desired velocity. Therefore, the desired personal velocity is replaced with v_k^q as in Eq. (4.17), where p_k is a panic parameter. If a particle k exhibits

an individualistic action then p_k decreases. Consequently, the personal force F_p is given in Eq. (4.18).

$$v_k^q = (1 - p_k)v_k^p + p_k v_k \tag{4.17}$$

$$F_p = \frac{1}{\tau}(v_k^q - v_k) \tag{4.18}$$

The repulsive force F_{rep} represents both the attempt of particle k to keep a certain safety distance from other particles, and the desire to gain more space in very crowded situations.

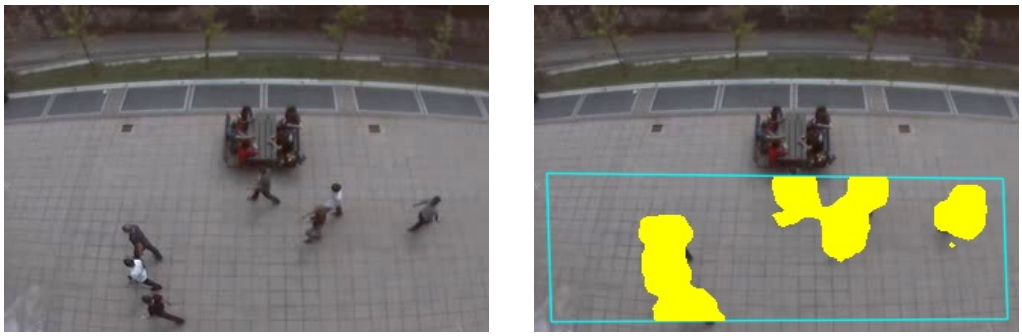
$$F_{rep} = v_k exp(\frac{-S_{avg}}{D_0} + \frac{D_1}{S_{avg}}) \tag{4.19}$$

In Eq. (4.19) S_{avg} represents the average distance of particle k from its neighboring particles over a fixed spatial window. It is reasonable to model particles such that they keep small distances from the surrounding particles, to which they are related or attracted to, and keep far distances from discomforting particles. Therefore, when S_{avg} is very small, particles are squeezed and the repulsive force will increase significantly, reflecting the strong reactions of those located in areas of high interactions. Overall, the *Extended* Social Force Model can be summarized as in Eq. (4.20), where τ is the relaxation parameter and n, D_0 and D_1 are constants.

$$m_k \frac{dv_k}{dt} = \frac{1}{\tau}(v_k^p - v_k) + v_k exp[\frac{-S_{avg}}{D_0} + (\frac{D_1}{S_{avg}})^n] \tag{4.20}$$

However, although parameters vary individually, and in order to avoid model artifacts, we chose fixed values for τ, n, D_0, and D_1 empirically, so as to achieve better calibration and stronger robustness, and excluding irregular outflows because of parameters variations. Since all particles are of the same sizes, therefore we set $m_k = 1$. Fig. 4.11 depicts a frame from a video sequence (a) where the particles of interest are annotated in yellow

within the ROI (b). Further illustration of the Social Force Model and the *Extended* variant is not in the interest of this paper, therefore readers are referred to [65] [20] [19] for comprehensive details.



(a) (b)

Figure 4.11: Extended social force model. The original frame from a video sequence (a); the potential particles are annotated in yellow (b).

.

4.3.3 Dynamic system

A dynamical system describes how a point in a space depends on time. According to [50], the behavior of particles in the crowd scene can be formulated by this system. Therefore, the coherent motion flow field can be treated as a continous dynamic system as formulated in Eq. (4.21).

$$\dot{\Psi} = F(\Psi) \tag{4.21}$$

where $\Psi(t) = [x(t), y(t)]^T$ and $F(\Psi) = [u(\Psi), v(\Psi)]^T$ represent the position and velocity of each particle, respectively. In general, a dynamic system is represented by a differential equation that can be approximated by using infinite series to identify a particular behaviour of the crowd. Therefore, we expand the taylor series around the critical point $\dot{\Psi}$ as formulated in Eq. (4.22).

$$F(\overset{*}{\Psi} + \delta) = F(\overset{*}{\Psi}) + J_F(\overset{*}{\Psi})\delta + \frac{1}{2}H_F(\overset{*}{\Psi})\delta^2 + \mathcal{O} \tag{4.22}$$

where $F(\overset{*}{\Psi}) = 0$ and $\delta(t) = \Psi - \overset{*}{\Psi}$ is a small agitation away from $\overset{*}{\Psi}$. In Eq. (4.22), $J_F(\dot{\Psi})$ and $H_F(\dot{\Psi})$ are the Jacobian and Hessian matrices, respectively. A critical point of $F(\overset{*}{\Psi})$ is a point where the rank of the Jacobian matrix is not maximal. Jacobian matrix is the linear approximation of the function F near the point Ψ and Hessian Matrix is the second order derivative near the critical point $\overset{*}{\Psi}$ that characterizes the local curvature of F. The Hessian matrix contains worthy information consolidating the dynamic system to detect the behavior accurately. In particular, the Hessian matrix renders useful information regarding video sequences containing high interactions and swift motions. Therefore, unlike [50] that exploits the Jacobian matrix only, we fuse both matrices together. We replace the off-diognal elements of Hessian matrix with $\frac{\partial^2 u}{\partial y^2}$ and $\frac{\partial^2 v}{\partial x^2}$ since we are interested only in the second order derivative in the same plane.

$$JH = \begin{pmatrix} \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} & \frac{\partial u}{\partial y} + \frac{\partial^2 u}{\partial y^2} \\ \frac{\partial v}{\partial x} + \frac{\partial^2 v}{\partial x^2} & \frac{\partial v}{\partial y} + \frac{\partial^2 v}{\partial y^2} \end{pmatrix} \tag{4.23}$$

We calculate the trace and determinant from Eq. (4.23) for each particle in the ROI and accumulate them according to [50] to identify the behavior of crowd.

4.3.4 Experimental results

To validate the performance of our approach, we have conducted the experiments on a set of 50 video sequences from benchmark dataset [50] and our UCD dataset [54]. These video sequences exhibit 14 lane, 15 arch/ring, 7 bottleneck, 5 blocking, and 9 fountainhead behaviors. To evaluate the performance of our approach, we compared it with the method recently

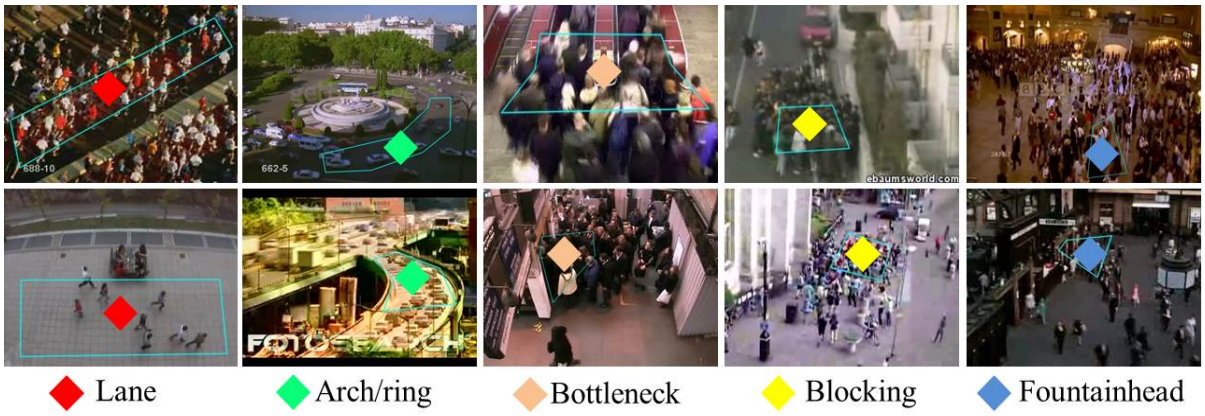


Figure 4.12: Crowd behaviors. Lanes are annotated in red (first column), arches/rings are annotated in green (second column), bottlenecks are annotated in brown (third column), blockings are annotated in yellow (fourth column), and fountainheads are annotated in blue (last column).

proposed by [50]. The ROI is selected manually for both the reference method [50] and the proposed method to maintain consistency in the evaluation process. For quantitative analysis the average F-score for each behavior is calculated for the reference [50] and proposed method. The F-score reaches its best score at 1 and worst score at 0.

Two sample frames for two video sequences for each behavior within the ROI is depicted in Fig. 4.12. The first column presents lanes annotated in red, the second column presents arches/rings annotated in green, the third column presents bottlenecks annotated in brown, the fourth column presents blockings annotated in yellow, and the last column presents the fountainheads annotated in blue, respectively.

The comparison of the obtained behavior detection results is shown in Table 4.2, where the second and third columns represent the average F-scores calculated over all video sequences for each behavior for the reference method [50] and the proposed method, respectively. We outperform the reference method [50] in four behaviors: namely arch, bottleneck, blocking

and fountainhead. However, we do not perform better than the reference method [50] in detecting the lane behavior. Hence, the quantitative results demonstrate that the proposed approach is robust enough to detect four crowd behaviors except lane comparing against the state of the art technique [50].

Table 4.2: Comparison of our method with the reference method in behavior detection. The average F-scores for each behavior is presented below for the reference method and the proposed method, respectively.

Behaviors	**Ref. method [50]**	**Our method**
Lane	0.625	0.415
Arch/Ring	0.698	0.941
Bottleneck	0.118	0.686
Blocking	0.048	0.107
Fountainhead	0.183	0.440

Chapter 5

Conclusion

According to the report by the United Nations [36], the urban population of the world has grown rapidly since 1950, from 746 million to 3.9 billion in 2014. This trend urges to define automatic tools to analyze crowd scenes for people safety. Therefore, in this doctoral study, we developed techniques dealing with pedestrian flows commonly exist in urban areas. For this purpose, we proposed flow segmentation method based on block-based correlation and α-expansion based on graph cut. On top of the segmention map, we investigated an anomaly detection strategy, by highlighting deviant motion of the pedestrians compared to what has been observed beforehand. We also proposed an approach for segmenting motion in crowded scenes using CRF. For this purpose, we extracted the orientation features by exploiting the optical flow evaluated on a set of particles uniformly distributed on the image plane. The orientation features are used as a-priori to train the CRF.

Moreover, we proposed a method to detect dominant flows in crowd videos. The approach, comprising of three stages, extracts first corner features from a video frame, and then exploits the enthalpy model to analyze the corner features based on their motion properties. Orientation information is then extracted from the corner features and exploited to train

a random forest. Dominant crowd flows are successively obtained in the testing stage. We also proposed an approach for detecting and tracking moving entities in surveillance videos based on the cross influence matrix and MLP neural netwok.

Considering the importance of anomalies, in term of panic situations, in crowded scenes, we proposed an approach using Gaussian mixture model. We demonstrated the capability of our approach in capturing the crowd dynamics by disposing a grid of particles over the video frame. The motion features of the particles adopt the GMM to learn the behavior of the crowd. The GMM model for anomaly detection is updated at each frame, in order to absorb the variations of the crowded scene arising from changes of scene context and crowd dynamics over time. To come up with improved performance for anomaly detection, we proposed another approach using corner features and an MLP feed-forward neural network. We demonstrated the capability of our approach in capturing the crowd dynamics by extracting corner features of a video frame. These corner features are exalted over time using the optical flow technique. The motion information of the corner features adopt the MLP neural network to learn the behavior of the crowd. The main advantage of the proposed method is that it considers crowd as a single entity, thus it does not require the tracking of individuals. This further justifies the applicability of our scheme for real time applications. The corner features for anomaly detection are extracted over a fixed temporal window, in order to make a vector of motion magnitudes, consisting of three observations, for each corner feature.

For behavior classification in low to medium density crowd, we proposed a particle-driven approach based on the initialization of a grid of particles uniformly distributed on the image plane. These particles are advected over a temporal window using the optical flow technique. We obtain the spatio-temporal features for these particles, which are combined pairwise to

identify the behavior of crowd within a region of interest. Additionally, we presented another approach for behavior classification irrespective of the density of the crowd based on the representation of a dynamic system. For this purpose, we find the motion flow field using the optical flow technique. Then a thermal diffusion process is applied to find a more coherent motion flow field. Subsequently, an extented social force model is exploited to filter out irrelevant particles. Then a matrix, formed by fusing Jacobian and Hessian matrices , is exploited to identify crowd behaviors within a region of interest selected manually.

Experimental results on video sequences from benchmark datasets as well as our own dataset, demonstrated that our proposed methods outperform other state of the art techniques in motion segmentation and behavior classification.

The future work should consider the problem of fusing the proposed approaches together, since each proposed approach can be applied in different situations. To this end, more features should be investigated to enable the proposed methods in the anomaly detection section to cope with other types of anomalies e.g., the presence of non-pedestrian entities in the crowded scenes.

Bibliography

[1] mha.cs.umn.edu/movies/crowd-activity-all.avi. UMN dataset.

[2] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 1–6, 2007.

[3] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S. C. Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *European conference on computer vision, Springer ECCV*, 2012.

[4] E.L. Andrade, S. Blunsden, and R.B. Fisher. Modelling crowd scenes for event detection. In *International Conference on pattern recognition, IEEE ICPR*, pages 175–178, 2006.

[5] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *International conference on computer vision. IEEE ICCV*, pages 1–8, 2007.

[6] A. Basharat, Y. Zhai, and M. Shah. Content based video matching using spatiotemporal volumes. *Computer vision and image understanding, Elsevier CVIU*, 110(3):360–377, 2008.

[7] M. Bertini, A. Del Bimbo, and L. Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization.

Computer vision and image understanding, Elsevier CVIU, pages 320–329, 2012.

[8] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Transactions on pattern analysis and machine intelligence, IEEE PAMI*, 26(9):1124–1137, 2004.

[9] Y. Boykov, O. Vekser, and R. Zabi. Fast approximate energy minimization via graph cuts. *Transactions on pattern analysis and machine intelligence, IEEE PAMI*, 23(11):1222–1239, 2001.

[10] L. Breiman. Random forests. *Machine learning, Springer*, 45(1):5–32, 2001.

[11] A.B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *Transactions on pattern analysis and machine intelligence, IEEE PAMI*, 30(5):909–926, 2008.

[12] S. Chapman and F.W. Dootson. A note on thermal diffusion. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 33(195):248–253, 1917.

[13] W. Chen, Y. Zhao, W. Xie, and N. Sang. An improved sift algorithm for image feature-matching. In *International conference on multimedia technology, IEEE ICMT*, pages 197–200, 2011.

[14] A.M. Cheriyadat and R.J. Radke. Detecting dominant motions in dense crowds. *Journal of Selected Topics in Signal Processing, IEEE JSTSP*, 2(4):568–581, 2008.

[15] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International journal of computer vision, Springer IJCV*, 62(3):249–265, 2005.

[16] X. Cui, Q. Liu, M. Gao, and D.N. Metaxas. Abnormal detection using interaction energy potentials. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 3161–3167, 2011.

[17] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis, Springer IA*, pages 363–370. 2003.

[18] W. Ge, R. Collins, and R. Ruback. Vision-based analysis of small groups in pedestrian crowds. *Transactions on pattern analysis and machine intelligence, IEEE PAMI*, 34(99):1–1, 2011.

[19] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.

[20] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[21] D.K. Iakovidis, T. Goudas, C. Smailis, and I. Maglogiannis. Ratsnake: A versatile image annotation tool with application to computer-aided diagnosis. *The Scientific World Journal*, 2014, 2014.

[22] J.C.S. Jacques Junior, S.R. Musse, and C.R. Jung. Crowd analysis using computer vision techniques. *Signal Processing Magazine, IEEE SPM*, 27(5):66–77, 2010.

[23] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 1446–1453, 2009.

[24] B. Krausz and C. Bauchage. Loveparade 2010: Automatic video analysis of a crowd disaster. *Computer vision and image understanding, Elsevier CVIU*, 116(3):307–319, 2012.

[25] I. Laptev. On space-time interest points. *International journal of computer vision, Springer IJCV*, 64(2-3):107–123, 2005.

[26] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 1–8, 2008.

[27] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision, Springer IJCV*, 60(2):91–110, 2004.

[29] Y. Ma, P. Cisar, and A. Kembhavi. Motion segmentation and activity representation in crowds. *International journal of imaging systems and technology*, 19(2):80–90, 2009.

[30] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 1975–1981, 2010.

[31] J.S. Marques, P.M. Jorge, A.J. Abrantes, and J.M. Lemos. Tracking groups of pedestrians in video sequences. In *International conference on computer vision and pattern recognition workshop, IEEE CVPRw*, pages 101–101, 2003.

[32] R. Mehran, B.E. Moore, and M. Shah. A streakline representation of flow in crowded scenes. In *European conference on computer vision, Springer ECCV*, pages 439–452, 2010.

[33] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 935–942, 2009.

[34] M.R. Montgomery. The urban transformation of the developing world. *Science*, 319(5864):761–764, 2008.

[35] B. Nadler and M. Galun. Fundamental limitations of spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1017–1024, 2006.

[36] United Nations. World urbanization prospects: The 2014 revision. 2014.

[37] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *European conference on computer vision, Springer ECCV*, pages 737–752. 2014.

[38] O. Ozturk, T. Yamasaki, and K. Aizawa. Detecting dominant motion flows in unstructured/structured crowd scenes. In *International conference on pattern recognition, IEEE ICPR*, pages 3533–3536, 2010.

[39] W. Pan, W. Dong, M. Cebrian, T. Kim, J. H. Fowler, and A. S. Pentland. Modeling dynamical influence in human interaction: Using data to make better inferences about influence within social systems. *Signal Processing Magazine, IEEE SPM*, 29:77–86, 2012.

[40] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *Transactions on pattern analysis and machine intelligence, IEEE PAMI*, 22(3):266–280, 2000.

[41] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *International conference on computer vision, IEEE ICCV*, pages 261–268, 2009.

[42] PETSdataset. http://www.cvg.rdg.ac.uk/pets2009/a.html. 2009.

[43] M. Pittore, M. Campani, and A. Verri. Learning to recognize visual dynamic events from examples. *International journal of computer vision, Springer IJCV*, 38(1):35–44, 2000.

[44] V. Reddy, C. Sanderson, and B.C. Lovell. Improved foreground detection via block-based classifier cascade with probabilistic decision integration. *Transactions on circuit and systems for video technology, IEEE CSVT*, 2013.

[45] M. Rodriguez, I. Laptev, J. Sivic, and J-Y Audibert. Density-aware person detection and tracking in crowds. In *International conference on computer vision, IEEE ICCV*, pages 2423–2430, 2011.

[46] P. Rota, H. Ullah, N. Conci, N. Sebe, and F.G.B. De Natale. Particles cross-influence for entity grouping. In *Signal Processing Conference, IEEE EUSIPCO*, 2013.

[47] L. Seidenari and M. Bertini. Non-parametric anomaly detection exploiting space-time features. In *International conference on Multimedia, ACM ICM*, pages 1139–1142, 2010.

[48] S.C. Shadden, F. Lekien, and J.E. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena, Elsevier*, 212(3):271–304, 2005.

[49] J. Shi and C. Tomasi. Good features to track. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 593–600, 1994.

[50] B. Solmaz, B. E. Moore, and M. Shah. Identifying behaviors in crowd scenes using stability analysis for dynamical systems. *Transactions on*

pattern analysis and machine intelligence, IEEE PAMI, 34(10):2064–2070, 2012.

[51] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages 246–252, 1999.

[52] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *Transactions on pattern analysis and machine intelligence, IEEE PAMI*, 22(8):747–757, 2000.

[53] R. Trichet and R. Nevatia. Video segmentation descriptors for event recognition. In *International conference on pattern recognition, IEEE ICPR*, pages 1940–1945, 2014.

[54] H. Ullah and N. Conci. Crowd motion segmentation and anomaly detection via multi-label optimization. In *ICPR workshop on Pattern Recognition and Crowd Analysis*, 2012.

[55] H. Ullah and N. Conci. Structured learning for crowd motion segmentation. In *International conference on image processing, IEEE ICIP*, 2013.

[56] H. Ullah and N. Conci. Real-time crowd behavior identification using a particle-driven approach. In *International conference on image processing, IEEE ICIP (**submitted**)*, 2015.

[57] H. Ullah, L. Tenuti, and N. Conci. Gaussian mixtures for anomaly detection in crowded scenes. In *IS&T/SPIE Electronic Imaging*, pages 866303–866303. International Society for Optics and Photonics, 2013.

[58] H. Ullah, M. Ullah, and N. Conci. Dominant motion analysis in regular and irregular crowd scenes. In *ECCV workshop on Human Behavior Understanding, Springer*, pages 62–72, 2014.

[59] H. Ullah, M. Ullah, and N. Conci. Real-time anomaly detection in dense crowded scenes. In *IS&T/SPIE Electronic Imaging*, pages 902608–902608. International Society for Optics and Photonics, 2014.

[60] M. Ullah, H. Ullah, and N. Conci. Crowd behavior classification through a hybrid lens. In *International conference on image processing, IEEE ICIP (**submitted**)*, 2015.

[61] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *International conference on computer vision, IEEE ICCV*, pages 734–741, 2003.

[62] H. Wang, A. Kläser, C. Schmid, and C.L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision, Springer IJCV*, 103(1):60–79, 2013.

[63] W. Wang, W. Lin, Y. Chen, J. Wu, J. Wang, and B. Sheng. Finding coherent motions and semantic regions in crowd scenes: A diffusion and clustering approach. In *European Conference on Computer Vision, Springer ECCV*, pages 756–771. 2014.

[64] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong. A comparative study of sift and its variants. *Measurement Science Review*, 13(3):122–131, 2013.

[65] W. Yu and A. Johansson. Modeling crowd turbulence by many-particle simulations. *Physical Review E*, 76(4):046105, 2007.

[66] B.J. Yves. Pyramidal implementation of the lucas-kanade feature tracker. *Microsoft Res. Labs, Tech. Rep*, 1999.

[67] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems, NIPS*, pages 1601–1608, 2004.

[68] B. Zhan, D.N. Monekosso, P. Remagnino, S.A. Velastin, and L.Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications, Springer MVA*, 19(5):345–357, 2008.

[69] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *International conference on computer vision and pattern recognition, IEEE CVPR*, pages II–819, 2004.