Multimedia Tools and Applications (2020) 79:20689–20728
https://doi.org/10.1007/s11042-020-08840-7

Gate and common pathway detection in crowd scenes and anomaly detection using motion units and LSTM predictive models

Abdullah N. Moustafa[1,2] · **Walid Gomaa**[1,3]

Received: 23 January 2019 / Revised: 23 January 2020 / Accepted: 13 March 2020 /
Published online: 23 April 2020

Abstract

In this paper, we propose two approaches to analyze the crowd scenes. The first one is motion units and meta-tracking based approach (MUDAM Approach). In this approach, the scene is divided into a number of dynamic divisions with coherent motion dynamics called the motion units (MUs). By analyzing the relationships between these MUs using a proposed continuation likelihood, the scene entrance and exit gates are retrieved. A meta-tracking procedure is then applied and the scene dominant motion pathways are retrieved. To overcome the limitations of the MUDAM approach, and detect some of the anomalies, that may happen in these scenes, we proposed another new LSTM based approach. In this approach, the scene is divided into a number of static overlapped spatial regions named super regions (SRs), which cover the whole scene. Long Short Term Memory (LSTM) is used in defining a predictive model for each of the scene SRs. Each LSTM predictive model uses its SR tracklets in the training, such that, it can capture the whole motion dynamics of that SR. Using apriori known scene entrance segments, the proposed LSTM predictive models are applied and the scene dominant motion pathways are retrieved. an anomaly metric is formulated to be used with the LSTM predictive models to detect the scene anomalies. Prototypes of our proposed approaches were developed and evaluated on the challenging New York Grand Central station scene, in addition to four other crowded scenes. Four types of anomalies that may happen in the crowded scenes were defined in the context, and our proposed LSTM based approach was used in detecting these anomalies. Experimental results on anomalies detection were applied too on a number of data sets. Ov erall, the proposed approaches managed to outperform the state of the art methods in retrieving the scene gates and common pathways, in addition to detecting motion anomalies.

Keywords Crowd scene analysis · Motion units · MUs · Long short term memory (LSTM) · LSTM predictive models · Tracklets · Trajectory formation · Anomaly detection · Video surveillance

✉ Abdullah N. Moustafa
abdalla.moustafa@ejust.edu.eg

Extended author information available on the last page of the article.

1 Introduction

Although the computer vision and artificial intelligence research communities have achieved significant advances in many problems in the last decades, there are still big challenges in analyzing crowd scenes, and detecting the activities related to them. The need for automatic crowd scene analysis has become an urgent need because of the increase in population, the sharp rise in the complexities of human activities, fighting crime and terrorism, in addition to the decreasing cost and widespread use of video surveillance cameras. Public celebrations, downtown streets, train stations, airports, sports activities, demonstrations, etc., are typical examples of the crowds that we can deal with, analyze, and manage to ensure safety and security of their people.

Automatic surveillance, analysis, prediction, detection, classify, etc., problem can help in crowd management and planning, ensuring the security and safety of people and property, detection and classification of anomalous events (such as people running due to panic situations, people overstock at entrance/exit gates, potential criminal or terrorist act, etc).

The problem of automatic analysis of crowded scenes is plagued by many challenges, which are mainly due to the complex motion patterns and the high dynamics inherent in such scenes (in many situations objects have to detour to avoid obstacles and avoid collision with each other). This makes the process of object detection and tracking very difficult, due to severe occlusions, interference, and dynamic background clutter [1].

Because of these challenges, and the importance of the task, training the computers to give them the ability to analyze the crowded scenes through the mining of large data captured by all available surveillance cameras, has become one of the hot topics in the computer vision and pattern recognition research communities. Solving this problem can be used in space planning, scheduling of public areas pedestrian flow, scheduling of public transport, guiding intelligent tracking systems, detecting and classifying anomalous behaviors and events, etc.

One of the core components of any crowd scene analysis vision-based approach is the selection of a suitable motion representation. In this regard, the literature identified multiple families of motion representations [20, 40]; these families can be generally classified into three as follows. The first is the *'trajectory based representation'*, where motion can be represented by complete trajectories of the moving objects for the whole time duration they appear in the scene [27, 39, 42]. The advantage of this representation is its ability to incorporate information about the moving object for the whole time it appears in the scene. So, it has all the temporal history of the motion. But, unfortunately, it is not practical for crowd scenes, because in this case, the object tracking process is impeded by small object sizes, large propinquity between the objects, and frequent occlusions.

The second family is the *'flow-based representation'* [21], where the motion features are extracted at the pixel level between consecutive frames [22, 26, 34, 36]. The approaches derived from this family work on the lowest level of the image structure (the pixel level). So they can capture the motion dynamics as it is without any approximations. However, these approaches can't capture the temporal history of these motion dynamics, which is very important for crowd scene analysis. Also, they do not handle the spatial changes in the scene dynamics (distinguishing between different moving objects in the scene e.g. pedestrian vs vehicle.), and the last shortcoming arises from its dependency on dense flow representation, which is prominently time-consuming and computationally demanding.

The third family of motion representation is a mix between these two extremes. In this family, several intermediate motion representations can be identified. One of them is

local spatiotemporal (2D/3D volumes) representation [9, 16]. In this representation, the motion features are extracted from two-dimensional (2D) patches or three-dimensional (3D) cuboids to capture long-range spatial, and temporal dependencies [5, 17, 28]. These approaches are mainly limited in their applicability to activity recognition and anomaly detection tasks [14]. Also, working with these approaches is very computationally expensive in the training and quantization stages [19].

Fortunately, in crowd scene analysis, we are interested in what is collectively happening, and not who/what is doing it [20]. Hence, there is no need for retrieving complete trajectories. Another derived motion representation, which is an enhanced version of the first family is the short trajectory fragments representation, called ***"tracklet"*** [39]. Tracklets represent the motion of a tracked object or points over a short time in the scene [6, 18]; that is why it gives a high-level description of the motion flow including a part of its temporal history [30]. Tracklets are stopped when any tracking ambiguities are detected, as a result, they are less likely to drift, and hence they can be more robustly extracted even from highly crowded scenes [20]. For these reasons, we selected tracklets as a robust motion representation in our proposed approaches for crowd scene analysis.

In this work, we aim at 1) analyzing the crowd scenes to discover the common motion pathways of the scene typical moving objects, (2) discovering the scene entrance/exit regions, and 3) building normalcy models for the scene motion dynamics and accordingly using these models in anomaly detection.

Generally, looking at the literature, the research work on crowd scene analysis using tracklets as a motion representation can be divided into ***traditional model-based approaches*** [20] and ***neural network (NN) based approaches*** [31]. The traditional model-based approaches use handcrafted energy functions and apply specific settings based on the type of scene to model the motion dynamics contained in it [13, 23]. The basic problem of this category of approaches is their dependency on the manually formed energy functions instead of learning from the scene tracklets data itself, so they often fail in cases of complicated crowded scenes. On the other hand, motivated by systems that learn from the given data, and due to the non-linearity nature of crowd scenes, deep learning-based methods attracted the computer vision community researchers who implemented numerous convolutional neural networks (CNN) based solutions for crowd behavior analysis [2, 29, 37, 41]. In the next section, we consider the most related work of these methods.

In this work, we propose two crowd scene analysis approaches based on using 'tracklets' as motion representation. In both of them, we assume the stationarity of the scene dynamics, (changing the motion dynamics of the scene over the time is not considered).[1]

The first approach is " Motion units and Meta-tracking based approach (MUDAM Approach)". This one depends on a new proposed representation for the scene motion dynamics (Motion Units (MUs)) in retrieving the scene common pathways and basic entrance / exit gates.[2] Due to limitations in this first approach, and motivated by their successful applications in sequence data processing and their capabilities of learning long-term dependencies, Long short-term memory networks (LSTMs) were recruited to model the scene overall motion dynamics in the second proposed approach (LSTM based approach).

In MUDAM approach, tracklets are collected and then hierarchically clustered together over consecutive time intervals using the non-parametric clustering technique proposed

[1] This point is more discussed at the end of Section 3.2.2.

[2] By retrieving a crowd scene entrance / exit gates, we mean the terminal points of that scene. These gates are shown for all of the scenes used in our experiments in Figs. 8a, c, and 9.

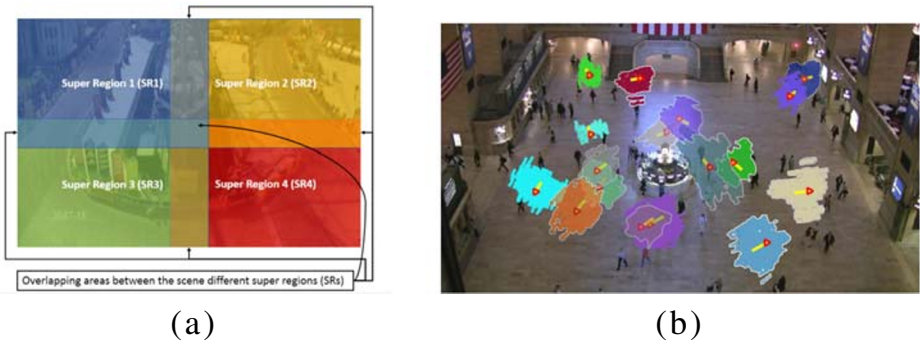

Fig. 1 **a** The Marathon scene divided into four overlapped super regions (SRs). **b** A group of MUs of the Grand Central station scene and their mean tracklets (yellow) and their orientations (red) [23]

in [13, 23].[3] The whole motion dynamics of the scene are then modeled into a network of interconnected, compact, and coherent motion patterns called motion units (MUs) (as explained in our previous work [23]). The relation between these MUs is analyzed so that, the scene entrance/exit gates are retrieved. Common pathways of the scene are also retrieved using a proposed motion units dynamic acquisition model and applying meta-tracking [23].

In the second approach (LSTM Based approach), a novel LSTM based predictive model is formulated for each part of the scene after dividing it into a number of overlapped parts called super regions (SRs), as shown in Fig. 1a, that gives an example of dividing the Marathon data set [13, 23] into four overlapped SRs.[4]

After that for each SR, a proposed LSTM predictive model is trained using its SR tracklets. Using short segments of the scene entrance tracklets (which were retrieved from the scene entrance gates) as inputs to the proposed LSTM predictive models defined over the scene SRs, all the possible trajectories of the scene can be predicted, and then clustered according to their entrance/exit gates to form the whole scene possible common pathways. Here we assume that the spatial layout of the scene gates is known apriori. However, we can use our proposed "MUDAM approach" to retrieve the scene gates spatial extent and their entrance tracklets then using that as an input for the second proposed approach "LSTM based approach"

LSTM based approach was also more tuned to work as an anomaly detection technique. It is used in detecting some of the crowd scene anomalies such as panic situations, gates overstocking, moving in the opposite direction, and moving in prohibited areas. In this approach, due to the training time of the different SRs LSTM predictive models takes most of the time complexity, so the approach trained offline on the scene video batches. After that, the trained LSTM predictive models can be used offline to extract the scene common pathways, or online to detect the scene anomalies.

The evaluation experiments demonstrated that our proposed approaches outperform the state of the art methods on both the structured (well-defined motion patterns) crowd scenes and unstructured (random and overlapping motion patterns) crowd scenes.

[3]This non-parametric clustering technique was selected, as it does presume a predetermined number of clusters, represents a state of the art technique, and the code is publicly available. The final clusters are obtained losing the temporal information as we are only interested in retrieving the overall scene motion dynamics (assuming such dynamics are statistically stationary).

[4]The basic intuition of dividing the scene into a group of SRs is explained later in Section 3.2.2

The main contributions of this work can be summarized as follows.

1. Developing a method for representing motion dynamics inside the crowd scene, by dividing the scene into a number of compact adaptive regions with coherent motion characteristics called the motion units (MUs).
2. Developing a method for discovering the scene gates (entrance/exits terminals) by analyzing the structure of the scene MUs, without the need for detecting the scene pathways first as in the other approaches.
3. Developing a new LSTM based generative models to provide faithful complete synthesized motion trajectories using short segments of the scene entrance tracklets as an input to the proposed LSTM based predictive models.
4. Developing a new anomaly detection criteria, that can differentiate the anomalous tracklets in the scene based on comparing its motion characteristics to those modeled by the corresponding SR LSTM.

This work is an extension of our conference work [23]. So, the first two contributions were previously introduced in the conference paper [23] and briefly mentioned again in this manuscript for clarification of ideas. Also in the current article we expand on details that were not mentioned in the conference paper. The last two mentioned contributions are new.

The rest of the paper is organized as follows. Section 2, outlines the most recent related work in the literature. In Section 3, all the details of our two proposed approaches are covered. Section 4, presents the evaluation experiments and results of our two proposed approaches versus the state of the art. Finally, Section 5 concludes the paper and gives points for future work.

2 Related work

Crowd scene analysis can be broadly divided into two basic categories: traditional model-based approaches and neural networks-based approaches [20, 31]. In many cases, the traditional category of models fails in achieving good analysis due to: I) the complicated nature of the crowd scene and II) the dependence on handcrafted models to simulate the motion characteristics instead of learning such characteristics from the given video data (which is one of the main reasons for the success of the analysis achieved by the neural network approaches).

In the following two subsections, we will review some of the most related work to our proposed approaches, stating their drawbacks, and how our proposed approaches overcome these drawbacks.

2.1 Traditional model based approaches

In this section, we focus on those approaches in the literature, that are most related to ours. So we mention techniques based on tracklets as a motion representation and trying to discover the scene basic structures (entrance/exit gates, and the common motion pathways of the scene).

Tracklets were collected and clustered twice using a non-parametric clustering algorithm in [13, 14] to form the crowd scene common pathways. On the terminals of the formed pathways, the scene gates were spatially identified using some geometry analysis for the shapes of the pathways. In such work, the retrieved pathways were not guaranteed to start nor end on a real entrance/exit scene gates, so this approach frequently failed in retrieving

the true scene gates locations. Also, the spatial extent of the formed pathways was not accurate, in many cases, the retrieved pathways are multiple pathways merged together, or just a part of a true scene common pathway. In [14], these problems have been identified and tackled by adding the source and sink gate probabilities for each tracklet as a prior and incorporating this prior information into the likelihood function. This has enhanced the clustering performance, and so obtaining better results. But, this requires adding prior knowledge by manually determining the spatial extent of the gates, in addition to some inaccuracies in determining the common motion pathways.

In our MUDAM approach [23], we first detect the scene gates by a way that guarantees their locations at the end of the motion area of the scene and then discovers the motion pathways extending between the obtained gates. That can address one of the main weaknesses of [13, 14], as will be discussed in more details later.

One of the trials to retrieve the crowd scene structure elements is based on *meta-tracking* [15]. In such work, the motion histogram was computed at each scene pixel, and orientation distribution functions (ODFs) were created using these motion histograms to summarize the direction of the flow at each pixel. A meta-tracking procedure was then launched to drive particles through the scene following the dominant flow characteristics expressed by these ODFs forming a number of particle trajectories called meta-tracks. These meta-tracks were then clustered to form the general motion patterns of the scene. Using the generated meta-tracks, the authors proposed a method for retrieving the scene entrance/exit gates. The weaknesses of this method can be summarized as follows: a) great time complexity in producing the ODFs since they are computed at the pixel resolution and b) poor analysis results for complex scenes, due to missing the temporal history of motion (temporal dependencies), and so the particles are very likely to drift when being meta-tracked by the created ODFs.

Unlike that work, in our proposed approaches we take the benefit of dividing the scene into coarser blocks, we as well consider the temporal motion history, which is very important especially in the cases of unstructured crowded scenes as we work on the scene tracklets instead of the scene pixels. Also, the time complexity of our LSTM based approach is very small compared to that of meta-tracking, by training the models of our scene divisions in parallel, which saves a lot of time compared to the sequential processing used by the other approaches. Finally, the results achieved by our approaches are much better both visually and in terms of objective measures.

2.2 Neural networks based approaches

Recurrent Neural Networks (RNNs) are neural networks that can deal with sequence data; in addition, their variations, specifically LSTM, are designed to handle data with long term dependencies alleviating the problem of *gradient vanishing or gradient explosion* [24]. Considering the tracklets extracted from the crowded scene as a time sequence data, LSTM models can be used to predict the next points or steps of these tracklets, which can be used in analyzing the crowd scene.

In [2] a model based on analyzing the social human behaviors in the crowd was proposed and called Social-LSTM model. This work aimed to predict the pedestrian complete trajectory based on analyzing some of the social activities such as avoiding collisions with other pedestrians or joining them if they were forming a group. In this model, LSTM was defined for each person in the crowd scene, and then a social pooling layer was introduced to describe the interactions between the different LSTMs inside the scene. However,

this approach is very computationally demanding in crowded scenarios because it needs an LSTM for each person; in addition to other problems such as occlusion, clutter, etc. On the contrary, in our proposed LSTM based approach the complexity of our LSTMs is very low compared to this social approach because we just use one LSTM for each super-region (SR) of the crowd scene. And the dependencies of motions across different regions are captured by a coarse overlapping of the super-regions, and hence, a coarse overlapping in the training of the LSTMS responsible for these regions.

Based on LSTM, the authors in [29] provided a two stacked coherent LSTM (cLSTM) model that aims to model the nonlinear characteristics of the crowd motion behaviors. The model was used in detecting the collective properties based on adding a regularization term. Also by training the model using the scene tracklets to learn their hidden features, it was then used in predicting the future paths of other tracklets based on the assumption that there is a collective relationship between them. Unlike that work, our proposed LSTM approach is based on creating a model for each SR, that model can learn from the tracklets passed through this region over the whole time, and then can generally predict a similar path through this region based on some partial given history about the predicted path.

In [37], the authors proposed bidirectional LSTM as a model to predict human trajectory in the crowd. In this work, the proposed model is based on the bidirectional-LSTM which takes into consideration the previous history of the trajectory points as well as all the possible scene destinations, In their approach, they divide the scene into a number of regions and use an LSTM for each region to give all the possible future directions of the trajectory entering this region. Another group of LSTMs different than the last mentioned is then used in predicting the trajectory based on its history points producing a number of probabilistic trajectory continuations. The basic drawback of this proposed algorithm is the complexity of their approach. That is mainly because of using two different layers of LSTMs, and each layer includes a number of LSTMs to cover all the scene divisions they did. On the contrary, our proposed LSTM based approach is more simple and its complexity is lower due to training the LSTMs in parallel.

3 Scene motion analysis

Sometimes it is very important to know what the typical motion in a local region of the scene looks like (regardless of the time), which is called the *stationary motion dynamics of the scene*. In our work, we are going to discover these dynamics (scene entrance/exit gates and common motion pathways) using two different proposed approaches MUDAM approach, and an LSTM based approach).

3.1 Motion units and meta-tracking based approach (MUDAM approach)

In this approach to discover the crowd scene dynamics (initially proposed in [23]) we first collect the scene tracklets over an extended period of time, and then cluster them hierarchically.[5]

[5]In our implementation we used the first stage of the non-parametric clustering algorithm used by [13], but any other clustering algorithm can be used.

After that, based on the scene motion flow, the scene is spatially divided into a network of interconnected Motion Units (MUs) [23].

3.1.1 Motion units (MUs)

Based on the scene motion dynamics, the whole scene can be mapped into a network of compact, coherent, and interconnected MUs. MUs are spatially-localized regions with coherent motion dynamics, which are specified by the compact and coherent tracklet clusters obtained by the hierarchical non-parametric clustering step [23]. After obtaining the whole scene tracklets, they are all clustered together regardless of their arise time in the scene. The clustering process is done by collecting all tracklets going in the same direction and very near in the distance to each other in one cluster [13, 23]. Our MUs are spatially represented by the obtained clusters, MUs directions are represented by the direction of the mean tracklet of each cluster tracklets as fully explained in [23].

Figure 1b shows a group of Grand Central station - fully described in Section 4.1- scene MUs, which are formed from the corresponding tracklets clusters. As shown from the figure, each MU consists of a number of tracklets that share the same motion orientation, and are very close to each other in the spatial distance. Each MU has an overall motion direction represented by the direction of its mean tracklet, and the MUs can be spatially overlapped.[6] By analyzing the relationship between these MUs, we retrieved the scene entrance/exit gates, and then discover the common motion pathways.

3.1.2 Tracklet continuation likelihood

To perform the MUs needed analysis, and find the relationships among them, which is used directly in detecting the scene dynamics or structures (entrance/exit gates, and common pathways), we defined two relations (connectivity and transition relations) as following:[7]

Tracklet connectivity relation This binary relation can be defined between any two tracklets T_i and T_j to see whether or not one of them can be considered as an acceptable continuation for the other. Figure 2a shows a number of tracklets (in green) that can be considered as acceptable continuations to the tracklet T_i since they satisfy a set of necessary and intuitive conditions as follows. For a specific tracklet T_i, any tracklet T_j's that satisfies the following three conditions is an acceptable continuation for T_i:

- T_j's tail lies within the angular field of view of T_i's head.
- T_j's tail is within a distance tolerance δ from T_i's head.
- T_j's orientation is within a tolerance angle ϕ from T_i's.

By applying this tracklet connectivity relation between the MUs mean tracklets, we can detect the entrance/exit gates of the crowd scene as explained in the following section.

[6] More details about the MUs spatial area, overall orientation, and the representing mean tracklets are discussed in details in [23]

[7] This section is explained in more details in [23].

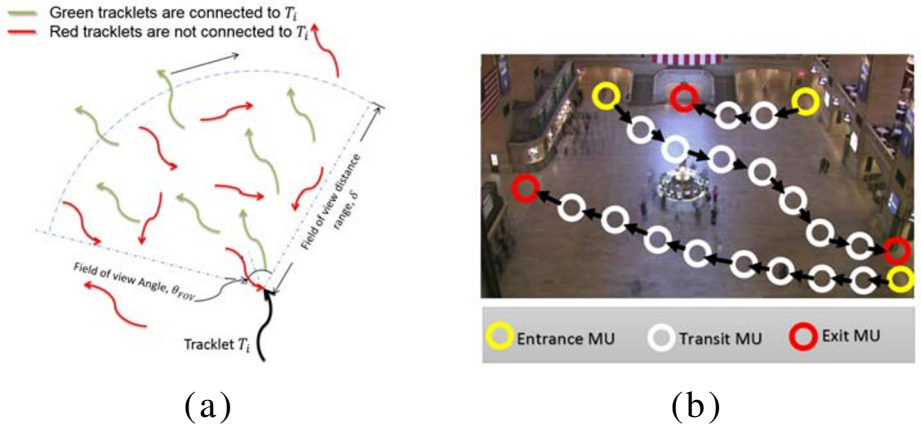

Fig. 2 **a** The field of view of a hypothetical tracklet T_i, and a group of acceptable continuation tracklets (green tracklets). δ and θ_{FOV} are the distance and angle parameters of the field of view, respectively [23]. **b** A hypothetical figure shows the type of MU (represented by a circular node) according to the connectivity relationship (represented by the black arrow) between it and the surrounding MUs

Scene gates discovery using the connectivity relation In many scenarios, the task of automatic gate detection is very important, regardless of knowing the typical pattern of motion pathways [23].[8]

The connectivity relation discussed above can be applied between the scene MUs in order to classify them into: Entrance MUs (Yellow), Transit MUs (White), and Exit MUs (Red)[9] according to the conditions mentioned in Table 1. Also see Fig. 2b.

After classifying the MUs into entrance/exit MUs, the terminal gates are detected by clustering the tails of the entrance tracklets (all tracklets of Entrance MUs) and the heads of exit tracklets (all tracklets of Exit MUs). The clustering, in this case, is done using the mean shift clustering algorithm [7, 35].[10]

The gate geometrical extent, in this case, is represented by the spatial extent of the shape including all the entrance and exit points of that gates. where entrance points are the tails of the entrance tracklets, and exit points are the heads of the exit tracklets contained in that gate.

Tracklet transition relation To model the motion dynamics of the scene to discover the common motion pathways. The transition relation between a tracklet T_i and a neighboring

[8]Typical examples shows the importance of the automatic gate detection: (1) Outdoor scenes, where typically, gates are not well defined. (2) Gates outside the camera's field of view, however, their presence can be recognized from the motion dynamics (for example, the gates in the bottom part of the Grand Central scene. (3) Dynamic scenes, due to construction work. (4) alleviating the burden of annual annotation. Our proposed MUDAM approach can detect the gates before discovering the scene pathways. Therefore, in situations when the gates are a priori known, this information can be incorporated into our framework and pathways can be directly detected.

[9]The figure shows the scene MUs as circles only for clarification, but in reality, the MUs can take any irregular shape as shown in Fig. 2b.

[10]Mean shift was selected since it can automatically estimate the number of clusters. Also, it is computationally efficient in case of dealing with points that lie on a 2D Euclidean space (which is our case).

Table 1 MUs classification conditions

	MU is classified as:	
Entrance MU	Exit MU	Transit MU
- Has at least 1 neighboring MU.[a] - Not a neighbor of any MU.	- Doesn't have any neighboring MUs. - A neighbor for at least 1 MU.	- A neighbor of at least 1 MU. - Has at least 1 neighbor MU.

[a]This means it has at least one valid MU as a continuation, which is represented in Fig. 2b by an arrow out of the MU

tracklet T_j is a real-valued relation that gives a score of the likelihood of how much that tracklet T_j can be a continuation of tracklet T_i. To set the scores between each MU and its neighbour MUs, we use the MU mean tracklet as a representative for that MU.[11] So, the transition score between any two mean tracklets T_i and T_j of two motion units MU_i and MU_j, denoted by T_{ij}, is calculated as follows:

$$T_{ij} = (1 - D_{ij}) + O_{ij} \tag{1}$$

where D_{ij} is the Euclidean distance between the head point of tracklet T_i and the tail point of tracklet T_j, after being normalized such that $D_{ij} \in [0, 1]\ \forall T_j \in N(T_i)$ (the neighborhood of T_i).

D_{ij} value was normalized according to the following equation:

$$D_{ij} = \frac{d_{ij} - d_{min}}{d_{max} - d_{min}} \tag{2}$$

where, d_{ij} is the euclidean distance between the head point of tracklet T_i and the tail point of tracklet T_j. d_{min} and d_{max} are the minimum and the maximum values of the distance in the scene. d_{min} is always equal to (0 pixel), and d_{max} is the maximum distance of the crowd scene (the diagonal distance).[12]

O_{ij} is the orientation similarity between the two tracklets, which is calculated as cosine the angle between their unit directions. Where, the tracklet direction is the direction of the vector going from the tracklet starting point to its end point.

This transition relation is used in detecting the common motion pathways of the crowd scene with the aid of the motion unit dynamic acquisition model described in the next subsection.

3.1.3 Motion unit dynamics acquisition model (MUDAM)

To discover the scene common motion pathways, we have to consider the motion inside the MUs. MUDAM is a proposed model to understand and simulate the motion inside the MU.

By advecting a particle in the crowd scene, MUDAM is the model used in guiding that particle inside the different scene MUs. To do that, MUDAM models the motion dynamics

[11]The Mean tracklet of an MU is defined as the average of all the tracklets contained in that MU, i.e. the i^{th} point in the mean tracklet is the average of the i^{th} points of all tracklets belonging to that MU.

[12]In our experiments for example, New York Grand Central Data set resolution is 1920 x 1080 pixel. So, $d_{max} = \sqrt{1920^2 + 1080^2} = 2203\ pixel$.

within an MU as a *linear dynamical system* by computing the linear transformation that controls the advancement of the motion exhibited by the tracklets contained in that MU. This linear transformation is represented by a non-reflective similarity transformation that moves each point on any tracklet belonging to an MU to the subsequent point on that tracklet. Such model is justified by the fact that tracklets are short trajectories, and for normally moving people at a regular place, the motion can be approximated locally (over the tracklet) by a line segment with white Gaussian noise.

If we consider a tracklet T_i that belongs to a motion unit MU_j, with two consecutive points p_{i_k} and $p_{i_{(k+1)}}$ on that tracklet. Then, MUDAM represents the temporal relationship between the two points as follows:

$$p_{i_{(k+1)}} = A_j \times p_{i_k} \tag{3}$$

where A_j is a non-reflective similarity transformation matrix estimated from the whole set of tracklets belonging to the motion unit MU_j as explained in details in [23]. Once A_j is estimated, it can be used to move any particle through that MU.

Common pathways detection using MUDAM and the transition relation To discover the common motion pathways of the scene we apply the steps shown in Algorithm 1, which are fully discussed in [23].

Algorithm 1 Generating a complete trajectory using the proposed MUDAM Approach [26].

```
Advect a particle in the scene;[13]
Apply the meta-tracking procedure as follows:
Select the MU that contains the particle location;
while Particle doesn't reach a scene exit gate do
    while Particle within the MU spatial extent do
        Apply MUDAM of the MU to predict the next location of the particle;
    end
    Use the connectivity relation to select the neighboring MUs of the current MU;[14]
    Use the transition relation to select one MU of the set of neighboring MUs with
    the highest transition likelihood;
end
```

After synthesizing large enough number of trajectories using the obtained entrance points (tails of entrance tracklets), they are clustered based on the entrance-exit gates to retrieve all the scene possible motion pathways and their spatial layout. This proposed approach showed acceptable results for the detection of the scene common motion pathways, and also it can perform better if we increased the number of the synthesized trajectories.[15] The results of applying that approach are shown in the experimental results section.

Generally, this proposed algorithm has some limitations:

[13] Particles advection locations are defined by the tails of the obtained entrance tracklets.

[14] Considering the current MU mean tracklet, we can select its neighborhood from the set of mean tracklets of all the MUs whose spatial layouts contain the current position of the particle, and hence, we have all the neighboring MUs of the current MU.

[15] Increasing the number of synthesized trajectories is done by increasing the number of particles advected in the scene. One possible solution is advecting particles at the obtained entrance point and around them instead of using only the exact retrieved point location.

1. It takes a lot of time to advect particles in the whole scene to retrieve all the scene possible common pathways. So it is computationally demanding.
2. Since all the obtained trajectories are just synthesized, they only reflect the motion shape and direction in a definite scene area. But they can't be used, to detect the scene anomalies, since they don't consider important properties of the scene motion dynamics (ex, the motion typical speed in a definite scene region).
3. This method has some failure cases such as in the Rush Hour2 and Streetlight data sets [23], where the obtained MUs were too large in the spatial extent compared to the scene dimensions. That causes a very large entrance gate in case of Rush Hour2 data set and a very large exit gate in case of Streetlight data set.
4. Another failure that was noticed in some of the obtained common pathways results is shown in Fig. 3. As shown in the figure, because the particle doesn't consider its history motion pattern, it may change its motion direction making a loop back into its start point or close to it.

As a result, we need a new approach that can accurately and robustly model the motion dynamics in any specific area of the scene, and ensuring the ability of that approach to achieve the following new requirements:

1. Overcoming the drawbacks and limitations of our previous MUDAM approach.
2. Having the ability to synthesize motion trajectories at any part of the scene based on the motion dynamics observed in this particular scene area.
3. Providing real information and characteristics in the synthesized trajectories such as the motion speed and direction (which are very important in building normalcy models and consequently detecting some abnormal situations such as running persons, or slowing persons due to gates over congestion.)

So the next new LSTM based approach is proposed.

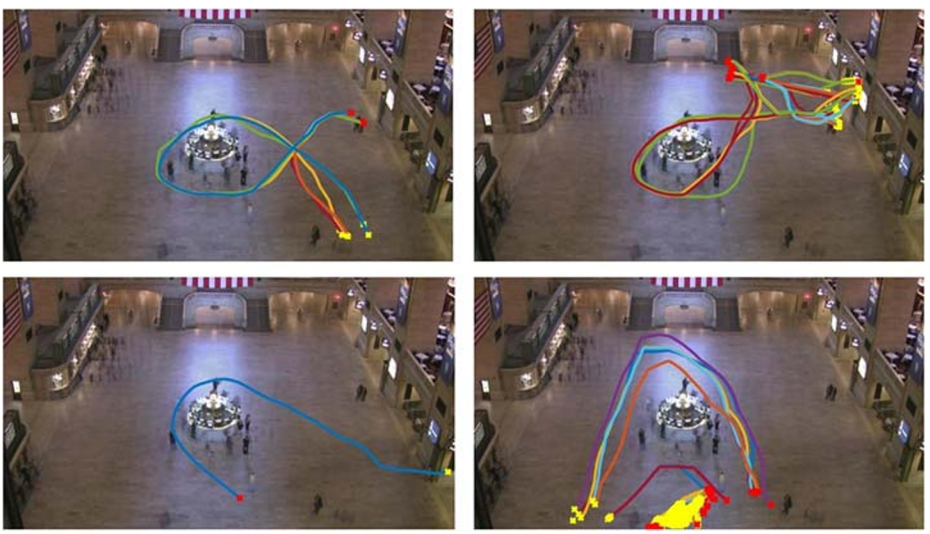

Fig. 3 Samples of the retrieved trajectories using our proposed MUDAM approach, that shows the loop back failure case [23]. The trajectories are starting at the yellow points and ending at the red points

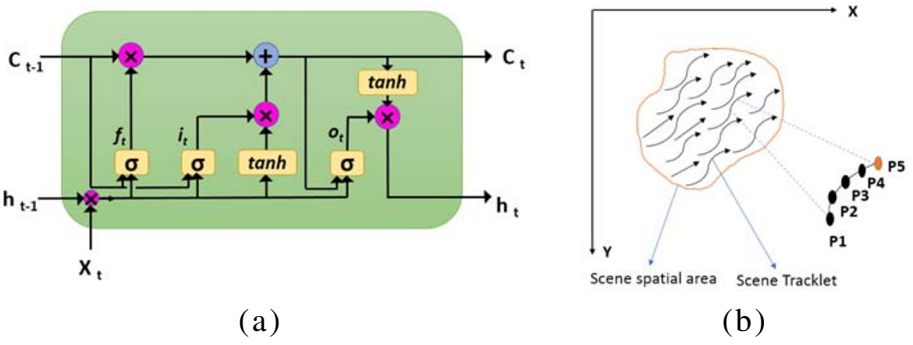

Fig. 4 **a** The architecture of an LSTM unit (peephole LSTM unit) [11]. The circles with $\times$, and $+$ inside represent an element-wise multiplication and summation respectively. the shapes with the sigmoid symbol or *tanh* are activation functions. **b** A hypothetical scene area containing some hypothetical tracklets (black curvy arrows). Hypothetically this area can be considered as a super-region despite in our experiments we are considering the SRs of uniform shapes (Rectangular shapes), which will be more discussed in the experimental section

3.2 LSTM based approach

Taking advantage of the current advances in using LSTM (a form of recurrent neural networks that avoids gradient vanishing and explosion) with time sequence data, we proposed a new LSTM based approach to analyze and model crowded scenes to detect the scene common motion pathways. This proposed LSTM approach assumes knowledge of the scene entrance segments[16] - considering that these segments are given as a prior - and then predicting a complete trajectory in the crowd scene based on the scene motion dynamics learned by LSTM networks.

In our experiments, we used our first proposed approach (MUDAM approach) to discover the scene gates and entrance tracklets. After that, the retrieved entrance tracklets were used in forming the entrance segments needed for the LSTM based approach.

3.2.1 Long-short-term memory (LSTM)

LSTM is a derived architecture from RNN, which can learn long-range temporal dependencies [12]. Figure 4a shows the typical architecture of a peephole LSTM unit (one of the popular variants of LSTM). It contains three types of gates: an input gate i_t, a forget gate f_t, and an output gate o_t; a memory cell c_t, and an output h_t. The input and forget gates are connected to the memory cell c_t to control how much information of the input, x_t and h_{t-1}, will pass to the cell, and how much will be delivered out to the output gate. Output gate o_t governs how much information will be passed to the output h_t.

The problem of vanishing gradient, that makes training very difficult in RNNs, has been solved in the LSTM architecture by making a self-connected recurrent edge of weight 1 over each LSTM unit. This connection ensures that the gradient can go across many training steps without vanishing. For a peephole LSTM unit, the equations that control the recursive computations of the memory cell activation and the three gates can be referenced at [37].

[16] An entrance segment is a small part at the start of an entrance tracklet with a definite length. For example, a segment of length 4 points, means the starting four points of an entrance tracklet.

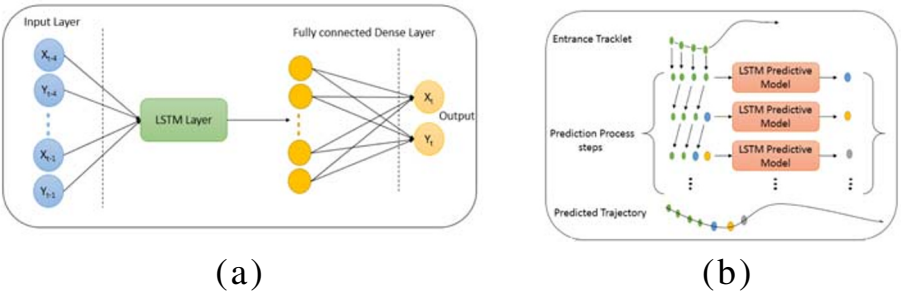

Fig. 5 **a** The architecture of the proposed LSTM predictive model. **b** A hypothetical figure showing how the trajectory generation process is applied

3.2.2 LSTM predictive model

In this section, we use LSTMs to train and build a predictive model that capture the motion dynamics of the scene SRs, which are fixed and overlapped areas that cover the whole scene. This predictive model will be used later to predict the motion dynamics within these super regions whenever it is needed, and also will be the basic elements of the proposed anomaly detection criteria.

If we have a hypothetical spatial area as shown in Fig. 4b. The motion dynamics within this area are expressed completely by the tracklets localized inside it. So by using these tracklets, as temporal sequences, to train our predictive LSTM model, we can have a model that captures the motion dynamics of this particular spatial area.

The architecture of the proposed LSTM predictive model is shown in Fig. 5a. It consists of an LSTM layer that takes four normalized image points $P_1, \cdots, P_4$, each represented by (x, y) coordinates, as an input to the input layer. The LSTM layer is then fully connected to the output through a dense layer. At the output layer, we can obtain the next normalized predicted point in the sequence which corresponds to P_5 in our hypothetical generated tracklet.

The points $P_1, \cdots, P_4$ are normalized with respect to the scene origin point(upper left corner point (0,0)) using the following normalization equations:

$$p_{i_{x_{norm}}} = \frac{p_{i_x} - x_{min}}{x_{max} - x_{min}}, \; p_{i_{y_{norm}}} = \frac{p_{i_y} - y_{min}}{y_{max} - y_{min}} \tag{4}$$

For the normalized point $P_{i_{norm}}(p_{i_{x_{norm}}}, p_{i_{y_{norm}}})$, p_{i_x} is the x value of the point p_i, p_{i_y} is the y value of the point p_i, x_{min} and y_{min} are the minimum x and y values of the scene (both of them $= 0$ because the scene origin is $(0, 0)$), x_{max} is the maximum x value of the scene, and the same is for y term of the point.[17]

After training each LSTM model using all the tracklets of its SR, the model can capture all the motion dynamics of that SR. Then, giving that trained model any four consecutive

[17]Data points are normalized before training, because having a large scale values of the the different features when training the LSTM model makes it weight these features not equally and so false priorities of the features over the others happens. So to avoid this false prioritization, we pass all the features data in a normalized form to train the model.

points (entrance segment) that lie in the spatial extent of its associated SR, it should be able to predict the fifth one reflecting the true dynamics in this SR.[18]

Due to our data - represented by the scene tracklets - is a univariate sequence, so we divided our sequence (tracklet) into multiple input/output samples where; 4 time steps (4 points) are used as input, and one time step (5^{th} point) is used as output for the one-step prediction. In our implementation, we used an LSTM model that has a single hidden layer of LSTM units, and an output layer connected through a dense layer to make the prediction.

The hidden LSTM layer is composed of 4 LSTM units, that take our last-mentioned data samples, and then the model is fit using the Adam version of stochastic gradient descent optimizer, and the mean squared error loss function for 20 epochs with a batch size of 1 and 10 % of the training samples for validation.[19] Due to the batch size is a gradient descent parameter that determines the number of training samples before updating the trained model parameters. So selecting it with the value of 1 in our training means that the model will updates its internal parameters after each sample of the training data. Also, the number of epochs is another parameter of the gradient descent, that controls the number of complete iterations over the whole training samples.

Validation accuracy is calculated after each epoch of the training process and the model is saved, which means that at the end of the training process - after 20 epochs- we will have 20 different model with different 20 values of the validation accuracy. The model of the best validation accuracy is then selected to be the LSTM predictive model that represent the motion dynamics of that SR.

In the following, we show how we used the trained LSTM models (corresponding to the constituent SRs of the scene) in discovering the scene common motion pathways.

LSTM predictive models for common pathways discovery In the LSTM based approach, in order to retrieve the scene common motion pathways, we apply the steps shown in Fig. 6.

First the scene is partitioned into a number of regular, and overlapped spatial areas called super regions (SRs), that cover all the scene areas containing any motion dynamics. The SRs should be overlapped to guarantee a smooth transition for the predicted trajectory between the different SRs, by letting each LSTM predictive model to learn about its surrounding other SRs via the overlap areas.[20]

Our intuitions of dividing the scene into a number of SRs are:

– Dividing the scene in this way, and handling the overlap between these divisions makes each LSTM predictive model (with a simple structure of 4 hidden units and a single layer) working on its division (SR) only not the whole scene. So, for each SR of the scene there will be a dedicated LSTM predictive model, which guarantees that the proposed simple LSTM predictive model structure can work effectively in the required task.

[18]In this paragraph we are supposing 4 points to predict the fifth one because that guarantees to keep a good amount of the motion history before that point. Also, 4 is not a fixed number it is only mentioned here for clarification.

[19]The selected values of the optimizer, loss function, number of epochs, and batch size parameters were selected after many experiments to define the most appropriate parameters for our problem over various data sets.

[20]The selected value for the overlap area is mentioned later in Section 4.6

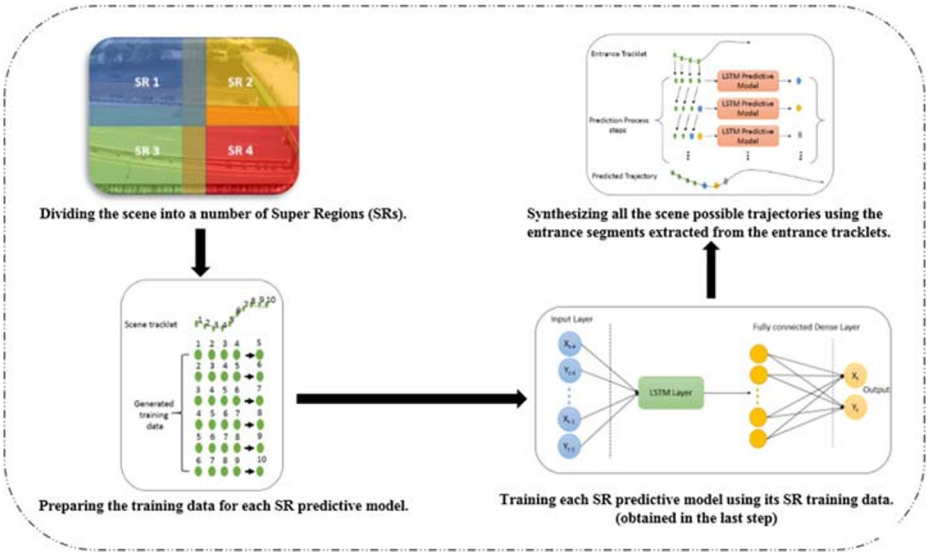

Fig. 6 The basic steps applied by the LSTM based approach to retrieve the scene common motion PWs

– To enhance the overall performance of our proposed technique, by minimizing the total training time of the different predictive models corresponding to different SRs by training these models in parallel, as these SRs can be considered independent from the perspective of motion dynamics.
– To have robust results, by proposing a suitable overlapping criterion between these SR models. That guarantees a smooth transition between the different SRs spatial areas during synthesizing the scene complete trajectories.

The second step is preparing the training data needed for each SR LSTM model. In this step, each tracklet in the SR is divided into a number of training samples - using all the points of the tracklet -. each sample consists of a number of locations (four locations shown in the figure) as an input, and the location of the next point as an output.[21]

All the obtained samples are then used in training our proposed LSTM model as shown in the third step and Fig. 5a. Our basic intuition of training the scene LSTM predictive models, is to give it the ability to predict the next point on the generated trajectory at any scene position, whenever that trajectory reaches this position based on its history (the last 4 points on the trajectory). To achieve that purpose, we need a number of training samples that cover the whole motion area of the scene. The more training samples of the scene, the better the obtained system accuracy of the predicted trajectory.

After that, the fourth step which is the trajectory synthesizing step is applied on each entrance tracklet of the scene (these are known as prior or can be obtained by applying the tracklet connectivity relation of our MUDAM approach). In this step, the LSTM predictive model is initialized by many points from an entrance tracklet equal to that number used during the training. Then, a prediction process step is applied to predict the next point on the synthesized trajectory as shown in Fig. 5b.

[21] The number of input locations used in our experiments is given and justified in Section 4.6

The steps of the generation are repeated iteratively on the previous points of the predicted trajectory until it reaches a valid scene exit gate. During the iterative prediction process of the pathway, each predicted point is tested if it lies in the motion map of the scene (Shown in Fig. 15b for Marathon data set) or not so that:

- In case of the predicted point is out of the motion map, in this case, the prediction process will be stopped and the proposed method will start a new generation process for another trajectory.
- If the point is still inside the motion map the prediction process continues until reaching a valid scene exit gate and the predicted trajectory is counted as a true predicted one.

It is also worth mentioning that, the LSTM predictive model used during the trajectory generation steps, shown in Fig. 5b (which is applied to predict the new trajectory point given the previously predicted points), is that one of the SR that contains in its spatial extent the coordinates of the predicted point obtained from the previously generated points. Also, in case that the newly predicted/generated point of the previous step is located in an overlapping area (such as those shown in Fig. 1a), all the LSTM models associated with the overlapped SRs are applied on the trajectory previously obtained points to predict a new point. The trajectory next point, in this case, is the average of all the predicted points by the overlapped SR predictive models.

Finally, after generating all the scene possible trajectories (by iteratively applying the generation process shown in Fig. 5b on the whole scene entrance tracklets), the synthesized trajectories are then clustered according to the scene entrance and exit gates (given as prior information, or can be obtained using the proposed MUDAM approach) to obtain all the scene possible common motion pathways.

In this work, we assumed the stationarity of the scene dynamics, which means that the motion dynamics are similar in the scene over time. In case of changing these motion dynamics with the time (changing dominant pathways over the time -different times of the day and different days-) there are two ways to handle that:

- The first is to allow our system to be trained on a video bathes collected over different times of the day and different days. In this case, our system will detect the general common pathways of the scene regardless of a specific time of the day or a specific day. That is the current operating way of our system. In this case, what we detect is an aggregate of the universal motion patterns over time.
- The second is to allow our system to produce different weights for the trained SRs models based on the time of the day, or the day of the week. After that, when using the SRs models to detect the scene common pathways, according to the time of the day - or the day of the week - the appropriate weights will be used.

Actually, changing the dominant pathways over different time (non-stationarity of the motion) is not our focus in this work, but it is one of our future directions.

LSTM predictive models for anomaly detection To detect crowd scene anomalies, we first capture the normal scene motion characteristics using the proposed LSTM predictive models described above. So these LSTMs act as normalcy models for the scene motion dynamics. After that, the following anomalous scenarios can be detected:

- ***Panic situation (people running):*** This situation is shown in the second row of Fig. 16, which shows three panic scenes from the UMN data set [33]. In this scenario, one or more persons are running suddenly, which is reflected in the scene dynamics by

very fast tracklets; when the distance between the tracklet points is larger than normal tracklets it will be a fast tracklet.
- ***Gates over congestion (slow motion):*** In this case, due to people crowding at one gate, so their movement will be slower. That is reflected in the scene motion dynamics by slow tracklets.
- ***Opposite direction motion:*** One person - or more - is moving opposite to the *typical* motion direction.
- ***Moving in a prohibited area:*** due to moving in a scene area that didn't have any previous motion dynamics.

By dividing the scene into a number of overlapped SRs, and training each SR LSTM predictive model using the extracted tracklets contained within its SR spatial extent, we will have a network of overlapped predictive models that cover the whole motion dynamics contained in the scene. The proposed LSTM predictive models can capture the general motion direction and speed at any point within the spatial extent of their SR.

So, in order to detect any of the last defined anomalies in a sequence of frames, we use the scene collective LSTM motion models as normalcy model of the whole scene - after being trained on the normal scene tracklets without any anomalies - in testing all the tracklets extracted from these frames.[22] as follows (Algoritm 2):

Algorithm 2 Anomaly detection steps using LSTM predictive models and the anomaly metric.

```
Collect the tracklets of all the frame sequence Tr ;
foreach tracklet tr_test ∈ Tr do
    if tr is out the motion area then
        Report tr as an anomaly tracklet with anomaly type: Moving in prohibited
        area;
    end
    else
        Make a segment Seg_tr_test of the first 4 locations of tr_test;
        Synthesize a tracklet tr_syn using the LSTM predictive model of the SR
        containing tr_test starting by Seg_tr_test ;[23]
        Apply the Anomaly Metric (discussed in the following subsection);
    end
end
```

Anomaly metric To test a tracklet tr_{test}, whether it is anomalous or not, using the scene appropriate LSTM predictive model (that one associated with the SR that contains in its spatial extent the tracklet) a similar tracklet tr_{syn} with the same number of points are synthesized using the starting points of tr_{test} as shown in Algorithm 2. After that, the steps shown in Algorithm 3 are applied.

[22] Each tracklet is tested using its SR LSTM model.

[23] The SR containing the tracklet tr_{test} is that one which have in its spatial extent all the points of that tracklet. In case of the tracklet spans multiple SRs, in this case, any of them can be used. That is mainly because of the overlap proposed criterion. In our experiments, we set the overlap distance to be larger than any of the scene tracklets. So in case of a tracklet that lies between two SRs, It will be lying in the overlap area. That area is considered at the training of the LSTM models of all the SRs sharing this overlap area. So any of the containing SRs can be used.

Algorithm 3 Anomaly metric steps.

```
Compute the average direction of tr_test;
Compute average direction of tr_syn ;
if The two average directions are in a counter direction (The difference between the
two angles >= 135° and <= 180 °) then
    Report tr_test as an anomalous tracklet;
    Anomaly type: Opposite direction moving tracklet;
end
else
    Compute the matching score S_m of tr_test;
    Obtain the overall gradient direction ∇_avg for tr_test and tr_syn;
    Obtain the average direction of the tracklet tr_test;
    if S_m >0.1 then
        Report tr_test as an anomaly tracklet;
        if ∇_avg and tr_test average direction in the same direction (The difference
        between the two angles >= 0° and <= 45 °) then
            Anomaly type: Panic situation (fast tracklet);
        end
        else
            Anomaly type: Gate over congestion (slow tracklet);
        end
    end
    else
        Report tr_test as a normal tracklet;
    end
end
```

As shown, first the average direction of both tracklets (tr_{test}, and tr_{syn}) is computed using (5). Comparing the obtained average directions, we can know if the tracklet is going in the opposite direction or not.

If the both tracklet are in the same direction, then a matching score S_m between them is computed as follows:

$$S_m = Slope_{tr} - Slope_{sr-avg} \tag{5}$$

where;

$Slope_{tr}$ is the slope of the line shown in Fig. 7b. The line is formed by the values of the euclidean distances between each point on the tracklet tr_{test} and its corresponding point on tr_{syn} shown in Fig. 7a.

To compute the $Slope_{sr-avg}$ value, all the SR (the one that includes tr_{test} in its spatial extent .) tracklets are passed to the SR LSTM predictive model. For each tracklet, a synthesized one of the same length -the same number of points- is created using the SR predictive model and the tracklet starting points as discussed previously in Fig. 5b. After that a line similar to that one in Fig. 7b. is formed from each SR tracklet and its synthesised one, then the slope of that line is computed. Averaging all the slopes of the lines corresponding to the SR tracklets, we have the value of $Slope_{sr-avg}$.[24] Due to the LSTM predictive model of the SR is trained using the SR tracklets, so passing these tracklets again to the model, it will

[24]The process of computing the $Slope_{sr-avg}$ for each SR, occurs offline only once for all the scene SRs.

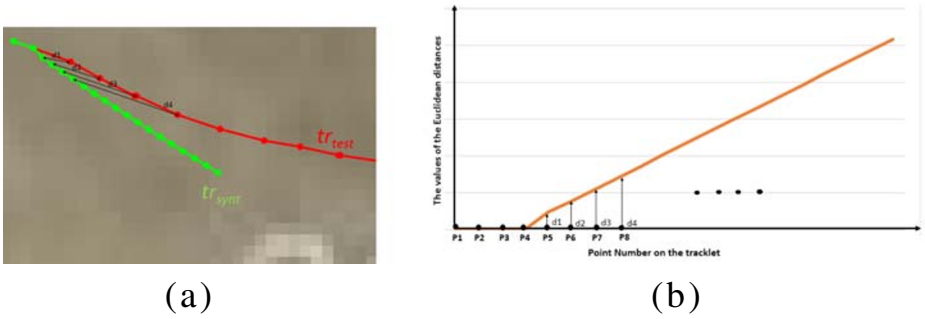

Fig. 7 **a** Test tracklet tr_{test} (red tracklet), synthesized tracklet tr_{syn} (green tracklet), and the Euclidean distances between each point on the former and its corresponding one on the latter (d1, d2,...). **b** The line representing the values of the Euclidean distances

synthesize a very similar tracklets with very small distances ($d_1, d_2, d_3, ...$). So the value of the $Slope_{sr-avg}$ is always $\in [0, 0.1]$.

The value of S_m is always $\in [0, 1]$. The values lower than 0.1 represent normal tracklets, and those greater than 0.1 represent anomalous tracklet (0.1 value was empirically set after applying many experiments on different scenes and found that the maximum value of the $Slope_{sr-avg}$ is 0.1).

To identify the anomaly type of the test tracklet (if it decided anomalous), we compute an overall gradient direction ∇_{avg} between the two tracklets (tr_{test}, and tr_{syn}), and also an average direction for the test tracklet tr_{test}.

The overall gradient direction is computed according to the following equation:

$$\nabla_{avg} = \frac{\sum_{i=1}^{n} \nabla_i}{n} \tag{6}$$

Where,

n: is the number of the tracklet points.

∇_i is the gradient direction between each point i on the tracklet tr_{test}, and its corresponding one on tr_{syn}.

The average direction of the tracklet tr_{test} is obtained as:

$$tr_{test_{averagedirection}} = \frac{\sum_{i=1}^{n-1} \nabla_{ab}}{n} \tag{7}$$

Where,

n: is the number of the tracklet points.

∇_{ab} is the gradient direction from the point $a(x_1, y_1)$ on the tracklet to the next point $b(x_2, y_2)$ with reference to the origin point of the scene. That can be obtained as follows:

$$\nabla_{ab} = \tan\text{inverse}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{8}$$

Comparing the obtained overall gradient direction ∇_{avg} to the average direction of the test tracklet tr_{test} (average direction of tr_{test}), we can classify the anomaly type of the tracklet tr_{test} into:

– *Fast tracklet (Panic situation)*, if the overall gradient direction ∇_{avg}, and the average direction of the test tracklet tr_{test} are in the same direction.

Table 2 Experimental data sets

GC [38][a]	Mar.	RushH.	StreetL.	ChinaS.
1 H	16 Sec.	6 Sec.	2.13 Min.	16 Sec.
1920x1080	720x480	480x360	480x360	480x360
25 FPS	25 FPS	25 FPS	25 FPS	25 FPS
Un S	S	S	S	S
GT	–	–	–	–

Mar. is Marathon data set, RushH: Rush Hour, StreetL: Street Light, and ChinaS: China Street. H is the time in Hour, Sec. is Second, and Min. means Minute. All the data sets dimensions are given in pixels, and FPS means Frame per Second. S: means a Structured Scene and Un S: means Unstructured Scene. GT: means there are a predefined Ground Truth for the data set

[a]The challenging New York's Grand Central station (GC) data set

– *Slow tracklet (Gate overstocking scenario)*, if the overall gradient direction ∇_{avg}, and the average direction of the test tracklet tr_{test} are going in a counter direction to each another.

4 Experimental results

4.1 Experimental data sets charactersics

In this work to verify the effectiveness of our proposed approach, we tested it on the data sets mentioned in Table 2. GC data set was used as our primary evaluation one. It is one of the most complicated scenarios of crowded scenes available on the web (densely crowded and unstructured scene with a wide range of motion dynamics).

Four other data sets (Marathon, Rush Hour, Street Light, and China street) have been used by our baseline state of art approaches (Jodoin et al. (JA) [15] and Hassanein et al.(HA) [13])[25] in addition to our baseline approach MUDAM [23]. So we also used them during our experiments.

4.2 Experiments pre processing Steps

Generally, in our experiments, we first extract the crowd scene tracklets using the KLT tracker [32] to obtain all the scene possible tracklets. Since KLT tracker is a robust, and well-known one in the computer vision research area, we used its default parameters to extract the scene tracklets. After that:

– In case of MUDAM approach [23], all the tracklets are hierarchically incremental clustered together - without considering the time of their rise in the scene - using the first stage of the non-parametric clustering algorithm used in [13].[26] The clustering step

[25]The JA and HA were selected to be our baseline state of art approaches, because they share with us the same goals of discovering the crowd scene basic structure elements (Entrance/Exit gates and the common motion pathways).

[26]Any cluster approaches can be used to cluster the tracklets under the condition of guaranteeing the compactness of the obtained MUs, and also guaranteeing the coherency of them (small variations in orientation between tracklets in the same MU).

is applied to obtain the scene MUs mentioned in Section 3.1.1, which is then used to retrieve the scene Entrance/Exit gates, and after that the scene common motion pathways.
– In the case of the second LSTM based approach, the whole scene tracklets are divided according to the scene super regions (SRs) divisions discussed in Section 4.4.1. After that each SR's LSTM predictive model is trained using the tracklets within the spatial extent of the SR as mentioned in Section 3.2.2 and then used in discovering the scene common motion pathways.

Our experimental results are compared against the two baseline (JA [15] and HA [13]) approaches, and the comparisons are performed in terms of the entrance/exit gate detection rate, common pathways detection rate, and the pathway spatial layout coverage.

4.3 Entrance/exit gate results

Commonly, as proposed by our MUDAM approach in [23], to retrieve any crowd scene entrance/exit gate without the need of prior information about the scene common pathways, the following steps are followed after applying the pre processing steps:

1. The connectivity relation (Section 3.1.2) is applied between the obtained MUs to classify them into Entrance, Transit, or Exit MU.
2. The mean-shift clustering algorithm [7, 35] is then applied on the whole obtained Entrance/Exit MUs together to form the scene expected gates (as shown in Fig. 8b).
3. Finally bipartite graph matching is used in matching the obtained gates (after applying the mean shift algorithm) with those of the ground truth (if available) to know how much did we truly retrieved of the ground truth gates, and how much is false.

The following subsections present the discovered gates of the: a) NYC Grand Central (GC) station data set and b) four other data sets: Marathon, Rush Hour, Street light, and China street; that were used by the other state-of-the-art approaches (JA [15], HA [13], and MUDAM approach [23].

4.3.1 New York's grand central station (GC) gate results

Collecting the tracklets from the GC data set, more than 1,550,000 tracklets are gathered. After applying four stages of the hierarchical clustering 1,338 MUs were obtained. Using these obtained MUs, and based on the scene characteristics, we constructed the tracklet

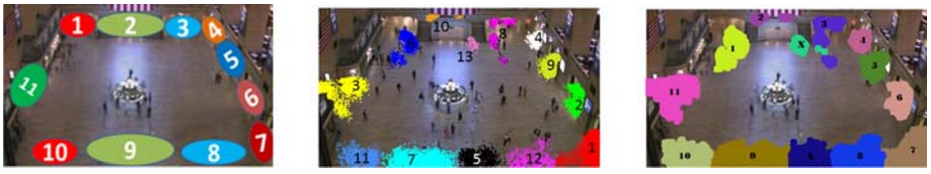

Fig. 8 **a** The manually annotated GT gates of New York's Grand Central station Data set [13]. **b** The obtained entrance/exit gates after applying the mean shift clustering [23]. **c** Our obtained gates after matching to the GT gates (bipartite matching)

Table 3 GC gates results for our proposed approach (MUs plus Meta-tracking) vs. Jodoin et al. (JA) [15] and Hassanein et al. (HA) [13] approaches

	JA	HA	Proposed
No. True detected gates (TD-G)	6	9	**11**
No. False detected gates (FD-G)	7	15	**2**

connectivity relation with the following parameter values: field of view angle $\theta_{FOV} = 50°$ and field of view distance $\delta = 300$ pixels.[27]

To detect the entrance/exit gates in terms of the obtained entrance/exit MUs, the mean-shift clustering algorithm, with a flat kernel, and a bandwidth parameter $BW = 135$ pixels was applied - more details are given in Section 4.6 - . The results of the obtained gates after applying the mean-shift clustering on the GC MUs are shown in Fig. 8b.

To compare our obtained results with those of the manually annotated ground truth (GT) (proposed by [13], and shown in Fig. 8a), we matched our obtained gates with those of the GT using the bipartite graph matching criteria [10], based on the overlap between their spatial extent. The results of the matching are shown in Fig. 8c.[28]

The quantitative results of comparing our obtained gates versus the two baseline approaches are given in Table 3. As shown, our proposed approach (MUDAM approach) outperforms both the two baseline approaches. The proposed approach can truly detect all the 11 gates of the GT, while JA and HA can only detect 6 and 9 gates respectively. Also, the number of false detected gates produced by our approach are the lowest (only 2) compared to 7 and 15 for the JA and HA. Overall, we can say that our proposed approach represents the state of the art in detecting crowd scene gates.

4.3.2 Gate discovery for other data sets

MUDAM approach also applied on the four data sets Marathon, Rush Hour, Street Light, and China Street. The last-mentioned steps of detecting the scene gates were applied with the same parameter values as those of the GC data set (field of view angle $\theta_{FOV} = 50°$ and field of view distance $\delta = 300$ pixels).

The results of the gate detection are shown in Fig. 9b. Comparing the obtained gates (visually) to the manually annotated GT gates on the first column (Fig. 9a), our MUDAM approach can detect all crowd scene gates, whatever that scene complexity (structured scenes, unstructured scenes, or even U shape scenes as the marathon scene). In some cases, gates are very coarse taking almost the whole scene such as the third and fourth scenes, but this problem is due to the obtained scene MUs are very coarse in these scene locations.That is meanly a limitation in the non-parametric clustering algorithm used in forming the scene MUs.

It is also worth mentioning that, For gates comparisons of these data sets, we applied qualitative evaluations that show the results of applying our proposed method on these data sets. Since we don't have a predefined ground truth (GT) for these scenes gates locations nor their spatial extent, so we can't apply quantitative evaluations as we did with the New York

[27]These parameter values are discussed in details in Section 4.6.

[28]The number written on the gate is the GT gate number to which this gate is matched to, and 'X' means false detection (there is no match between the detected 'X' gate and any of the GT gates).

Fig. 9 Qualitative results of applying our MU and meta-tracking based approach on the Marathon, Rush Hour, Street Light, and China Street datasets (from top to bottom respectively). **a** Ground Truth entrance/exit gates and common pathways (green arrow). **b** Detected Entrance (yellow)/Exit (red) gate points

Grand Central Station data set (Counting the number of true detected and false detected gates compared to the GT).

4.4 Discovery of common pathways

To discover the crowd scene common motion pathways (PWs), we previously proposed MUDAM approach [23], which used the transition relation (Section 3.1.2) in discovering these PWs. Due to some weaknesses and also the need for achieving new requirements (as mentioned in Section 3.1.3), we proposed an alternate new LSTM based approach to detect the scene common PWs.

In this section, to evaluate the performance of our proposed approaches (MUDAM and LSTM-based), we compare our obtained results versus: (1) the large scale annotated ground truth (GT) of the GC video dataset [38], that contains 12,684 annotated trajectories, and based on the manually annotated ground truth gates of [13], (2) the two baseline approaches JA [15] and HA [13], and (3) our previous work (MUDAM approach [23]).

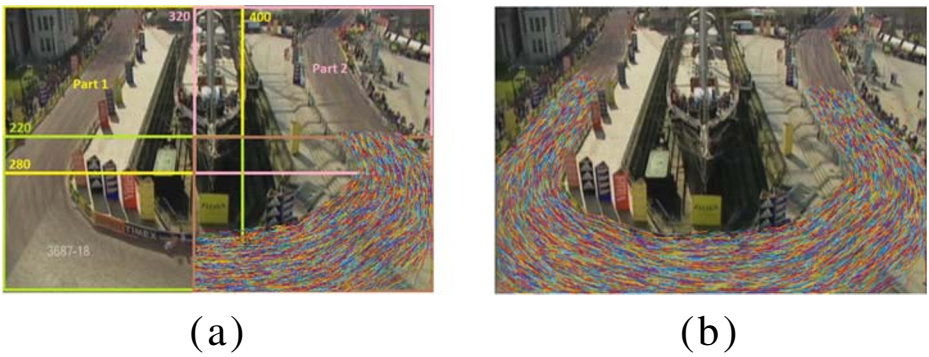

Fig. 10 **a** The tracklets of one of the scene SRs. **b** The Marathon scene collected tracklets

In our proposed LSTM based approach, to discover the scene common pathways, the following steps (showed in Algorithm 4) are applied after collecting the scene tracklets.

Algorithm 4 Common pathways detection steps using our proposed LSTM based approach.

Divide the scene into a number of overlapping SRs as shown in Fig. 1a;[29]
foreach *SR in the Scene* **do**
  Train an LSTM predictive model (as discussed in Section 3.2.2) using its SR tracklets (see Fig. 10a)
end
Synthesize complete trajectories using the scene entrance tracklets and the proper pipelining of the trained LSTM models (the fourth stage in Fig. 5b;
Classify the obtained trajectories according to the scene entrance/exit gates to form all the scene common motion pathways;

To evaluate the proposed approach, we tested it on the GC dataset, in addition to the other four data sets: Marathon, Rush Hour, Street Light, and China Street. However, before that, we analyze the impact of changing the number of SRs of the scene on the performance and the results obtained by our proposed approach.

4.4.1 The resolution of super regions

In this section, Marathon dataset is used to perform analysis of the proper number of SRs that will be used in dividing the scene. To do that, the scene is divided into two, three, four, and six overlapped rectangular parts as shown in Fig. 11a from top to down.

Basically, the rectangular shape is chosen (instead of hexagons for example), because it can cover the whole scene without the need for any padding at the scene boundaries. Also, the rectangular shape allowed us to easily perform the overlapping criteria between the scene different SRs to guarantee a smooth transition between the scene different SRs when synthesizing complete trajectories.

[29]The number of SRs that the scene will be divided into is analyzed in Section 4.4.1.

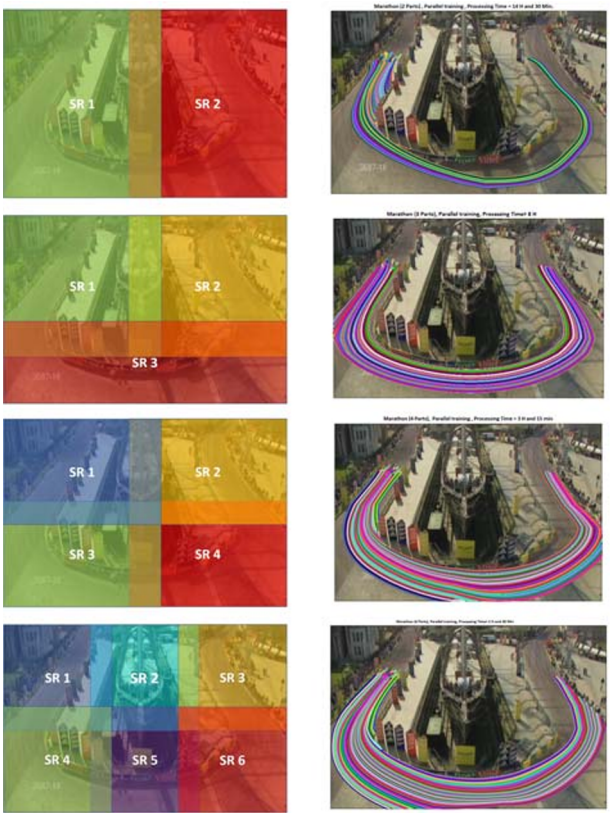

Fig. 11 The Marathon scene is divided into different number of overlapped super regions (SRs) as shown in Column (**a**), and the corresponding obtained common pathways in each case (shown in Column (**b**))

In each of the mentioned scene divisions, a number of predictive LSTM models equal to the number of the scene SRs are trained (each on its SR tracklets). These LSTM models are then applied on the scene entrance tracklets considering the following: (1) Each model is only applied within the spatial extent of its corresponding SR and (2) All the overlapped models are applied simultaneously together over their shared spatial area and their resulting points are averaged to obtain the next trajectory point within the overlapped area. The averaging of the points is executed using the following equation:

$$(x_{new}, y_{new}) = \left(\frac{1}{n} \sum_i x_i, \frac{1}{n} \sum_i y_i \right) \tag{9}$$

Where: (x_{new}, y_{new}): is the coordinate of the new point on the trajectory.

n: is the number of the obtained points using the overlapped models (which is the number of LSTM predictive models that share this overlapping area too).

x_i and y_i: are the x and y locations of these obtained points. Finally, the scene common motion pathways (PWs) are obtained as previously shown in Fig. 5b. The obtained results for each case of division are shown in Fig. 11b.

Generally, Fig. 11 illustrates smoother, more natural, and closer to reality pathways with increasing the number of SRs. So, we conclude that, the more scene SRs, the better the obtained common PWs results. But the computational resources needed to process the whole scene by increasing the number of SRs are increased. That can be mitigated by applying the parallelization processing of these SRs to gain more time. Also using more SRs means a finer and smaller SRs, so the training of the corresponding LSTM predictive models will be easier.

To confirm our last subjective conclusion we used a quantitative evaluation (spatial layout coverage criterion [23]). This criterion measures how much the spatial extent of a retrieved PW P_i matches the spatial extent of a ground truth PW P_i^{GT}, by computing the quadratic chi-square χ^2 distance[30] between these two PWs heat maps.[31] To measure that χ^2 distance, we construct two probabilistic distributions (one for P_i, the other for P_i^{GT}), and compute the distance between them using the following equation;

$$S_d = \chi^2(P_i, P_i^{GT}) = \frac{1}{2}\sum_n \frac{(P_{i_n} - P_{i_n}^{GT})^2}{(P_{i_n} + P_{i_n}^{GT})} \tag{10}$$

Where $S_d \in [0, 1]$ for 0 representing the best match between the two PWs (P_i and P_i^{GT}). P_{i_n} is the n^{th} point of the PW P_i, and $P_{i_n}^{GT}$ is the corresponding point of the PW P_i^{GT}.

This measure of the spatial similarity (between the discovered and the GT PWs) between any two PWs has been used by [14] to form their Pathway Matching Score (PMS). PMS measures the amount of matching between two pathways in terms of a spatial matching score $S_{\chi_s^2}$, and a motion orientation matching score $S_{\chi_o^2}$. The spatial extent matching score is the same as our previously proposed measure S_d, (10), subtracted from 1 to give a similarity value of 1 for the best match (conversion between distance and similarity measures). The motion orientation matching score is on the quadratic chi-square distance between the two PWs motion orientation distributions (the motion orientation histograms of the PWs[32]), and then subtract this value from 1 too.

The overall PMS between two PWs (A and B) is then given by averaging the spatial and motion orientation matching scores as follows [14]:

$$PMS(A, B) = 0.5(S_{\chi_s^2} + S_{\chi_o^2}) \tag{11}$$

Where, PMS lies in the unit interval $\in [0, 1]$ for 1 representing the perfect match between the two PWs.

As PMS is currently the most viable objective measure, we applied it to evaluate our obtained pathways and provide quantitative results. To differentiate between our discovered pathways in the Marathon scene, shown in Fig. 11, and as we don't have GT for this scene, we used the scene collected tracklets, shown in Fig. 10b, as if it was a ground truth PW, and applied the PMS measure between our obtained PWs and those tracklets.

The results of applying the PMS metric on four different SR partitioning of the Marathon dataset are shown in Table 4. Since in marathon data set there is only one ground truth

[30] chi-square χ^2 distance is used to test the amount of fit between two distributions [25].

[31] Pathway heat map is a spatial probability map that is basically constructed by overlaying all the pathway trajectories, and accumulating them on top of each other, and then normalize all the values of the whole map to be in the range [0,1], for 1 representing the highest motion dynamics at this point in the map, and 0 for no motion.

[32] Defined by [14], the motion orientation histogram is formed for any specific PW by computing the motion direction between each two consecutive points of that PW trajectories or tracklets, and then quantize them into one histogram. The full mathematical definition and clarification can be found in [14].

Table 4 The results of the discovered PWs of Marathon dataset using LSTM proposed approach after dividing the scene into two, three, four, and six super regions (SRs)

Score	2 SRs	3 SRs	4 SRs	6 SRs
SS	0.2383	0.2446	0.3318	0.3512
MOS	0.7029	0.7855	0.7829	0.7640
Avg.	0.4706	0.5100	0.5574	**0.5576**
$T(hours)$	14.5	8	3.25	**2.5**

SS: Spatial matching Score, MOS: Motion Orientation matching Score, and Avg.: Pathway average matching Score (PMS). T: the time for training one LSTM predictive model in hours. Bold numbers show the best average pathway matching score, and the best processing time needed to discover the scene PWs

common pathway that goes from the input gate to the exit gate. So comparing to the obtained results of our proposed approach in Fig. 11, we found that there is no false detected PWs in all the cases. Also, as shown in the table results, the more scene SRs, the better the obtained common PWs results; the PWs become smoother, more natural, and closer to reality with increasing the number of SRs.

Also, a lower training time T for obtaining the scene PWs is achieved when the number of the scene SRs increased. That is mainly because of two factors: (1) The lower number of tracklets used in training each LSTM predictive model (tracklets are divided on more SRs). (2) The parallelization of training each one independently assuming the tracklets all exist apriori.[33,34]

4.4.2 Pathway discovery in New York's Grand Central station (GC)

To obtain the Grand Central PWs, we divided the scene into 16 overlapped SRs ($4x4$ SRs).[35]

Applying the PW discovery steps mentioned in Algorithm 4 on the scene entrance tracklets of each scene gate as an input to the 16 trained LSTM predictive models of the scene, we obtain all the gate pairwise possible pathways.[36]

To evaluate our obtained results against the state of the art approaches, we compared against the two baseline approaches JA [15] and HA [13]; in addition to our previous work MUDAM approach [23] and the ground truth (GT).

[33]The experiments on the Marathon dataset were tested on a machine with Core i7 processor and 8GB of Ram.

[34]The processing time to obtain the scene pathways is the time needed for training the scene LSTM predictive models in addition to the time needed to synthesize the complete trajectories using the entrance segments. Since the time needed to predict the trajectories is almost fixed and very small compared to the training time, so we didn't consider it in our time complexity comparison. Also, since all LSTM predictive models are trained in parallel, we only reported the time of training one of these models (the largest time).

[35]Controlled by the available hardware resources, this number of SRs was empirically chosen to give good and smooth PWs results in an appropriate processing time. In future work, we will investigate how to automatically set this parameter through a pre processing stage that studies the complexity of motion dynamics inside the scene.

[36]The scene entrance tracklets are those of the entrance gates. For our proposed LSTM based approach to work and retrieve the scene common PWs, it needs entrance segment of the entrance tracklets as an input. Those tracklets should be a priori known, or be obtained using our proposed MUDAM approach by discovering the scene gates first. In our experiments, we used our retrieved gates entrance tracklets of MUDAM approach as an input to the LSTM proposed approach.

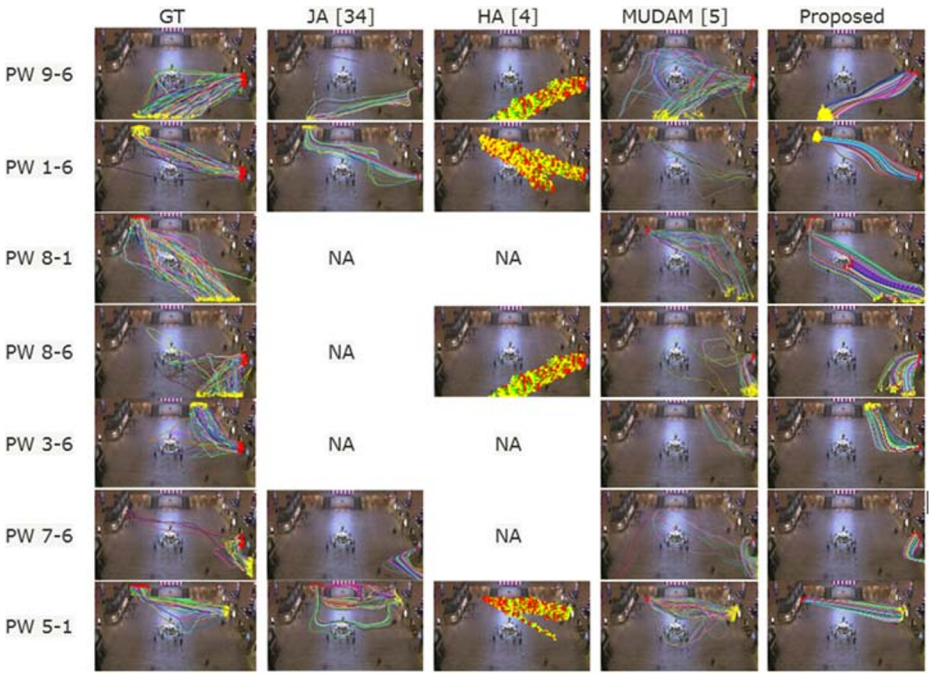

Fig. 12 The 7 richest PWs of the GT, JA [15], HA [13], MUDAM [23], and our LSTM based approaches from left to right respectively

The assessment methodology is borrowed from [13] and [23]. To evaluate our proposed approach, we sorted the discovered PWs of the scene according to their richness,[37] and then counting how many PWs (of the 7 richest GT PWs) are discovered by our LSTM based approach versus JA, HA, and MUDAM. The results are shown in Fig. 12.

As shown, both of our MUDAM and LSTM based approaches can capture all the 7 richest PWs. But the LSTM based approach can retrieve more motion dynamics of the 7 richest PWs with a high degree of smoothness and high visual similarity to the GT PWs than the other approaches.

To prove these results quantitatively, we applied the PMS evaluation criteria on the whole 7 richest PWs of each approach with respect to the GT and compared the obtained results together in Table 5. From the table, we can see that: (1) our proposed LSTM approach retrieves all the 7 richest PWs, (2) It can recall more spatial coverage and motion orientation of the GT PWs (PWs 8-6, 3-6, and 5-1), or slightly lower (second-best rank) with comparable values for almost all the other PWs, and (3) It is the fastest in terms of the processing time to obtain the scene results. As shown in Table 5 results, LSTM based approach only needs 2 days to detect all the common pathways of the scene compared to 5 days for the MUDAM approach.[38]

[37] Richness of a PW is measured in MUDAM approach, MT approach, LSTM approach, and GT by counting the number of the trajectories of that PW. While for the HA approach it is measured by counting the number of the tracklets contained in that PW.

[38] Time comparison was applied on a machine with Intel(R) Xeon(R) CPU E5-2699 v3 @2.30GHZ (2 processors), and 256 GB of RAM

Table 5 New York's Grand Central dataset 7 richest PWs results obtained by JA [15], HA [13], our proposed MUDAM [23], and our Proposed LSTM based approach

PW	Score	JA	HA	MUDAM	LSTM
	SS	0.146	1.000	0.613	0.742
9-6	MOS	0.041	0.946	0.874	0.954
	Avg.	0.094	**0.973**	0.743	0.848
	SS	0.001	0.233	0.293	0.094
1-6	MOS	0.329	0.853	0.718	0.831
	Avg.	0.165	**0.543**	0.506	0.463
	SS			0.659	0.670
8-1	MOS	NA	NA	0.811	0.630
	Avg.			**0.735**	0.650
	SS		1.000	0.201	0.753
8-6	MOS	NA	0.539	0.464	0.908
	Avg.		0.770	0.332	**0.830**
	SS			0.456	0.8350
3-6	MOS	NA	NA	0.817	0.610
	Avg.			0.636	**0.722**
	SS	1.000		0.504	0.681
7-6	MOS	0.705	NA	0.532	0.579
	Avg.	**0.853**		0.518	0.630
	SS	0.001	0.003	0.173	0.307
5-1	MOS	0.510	0.803	0.558	0.457
	Avg.	0.255	0.403	0.366	**0.412**
	T (days)	14	7	5	**2**

SS: Spatial matching Score, MOS: Motion Orientation matching Score, and Avg.: Pathway average matching Score (PMS). Bold values represent the best-obtained matching score between that column approach and the GT

4.4.3 Discovery of PWs in other datasets

In addition to the GC and Marathon data sets, we also tested our LSTM based approach in retrieving the scene PWs on three other data sets: Rush Hour, Street Light, and China Street. For each one of these, the scene was divided into four SRs and then corresponding four LSTM predictive models were trained. Because the three scenes are small in their resolution 480 x 360 pixel so 4 SRs divisions are enough to handle the different dynamics inside the scene.[39]

The obtained LSTM predictive models are then used to retrieve the scene common PWs using the entrance tracklets of the obtained entrance gates (see Section 4.3.2.

Qualitative (visual) results of these data sets are shown in Fig. 13. The figure shows the scenes collected tracklets in the first row, the scenes SR divisions (second row), the

[39]"enough" means that there is no need for more divisions for these scenes. Also more details for choosing this value is given in Section 4.6.

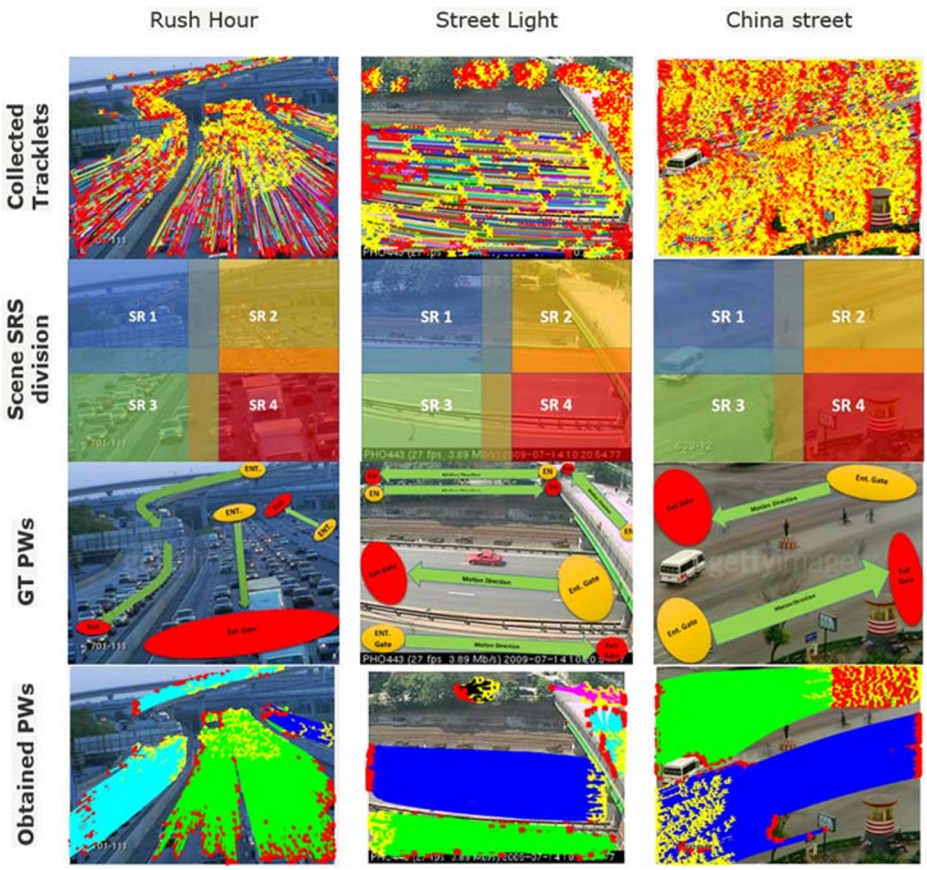

Fig. 13 PW qualitative results of applying our LSTM based approach on the Rush Hour, Street Light, and China Street datasets (from left to right)

manually annotated ground truth PWs direction of motion (third row), and finally the discovered PWs (direction goes from the yellow points to the red points) in the last row.

To quantitatively evaluate these data sets retrieved PWs results, we need a GT to compare with. Due to the lack of the GT pathways count, locations, or their spatial extent, So for these data sets evaluation, we depended on the qualitative one as our basic evaluation.

But, since we have the collected tracklets of these scenes, which represent the scene motion dynamics, we used the PMS to measure how much the retrieved PWs trajectories (all of them together) matched the scenes collected tracklets (Spatial matching using the spatial matching score (SS), and motion orientation matching using motion orientation matching score (MOS) as shown in Table 6. As we mentioned previously, that is not our basic evaluation because the LSTM predictive models are trained using these tracklets. So we couldn't do the same with the other state of the art approaches results to compare with our results. As a result, for these data sets our fair evaluation is the subjective or the qualitative one not the mentioned quantitative one.

Results of Table 6 show that our proposed LSTM based approach can effectively retrieve the scene PWs in case of Rush Hour and Street Light. But for China Street dataset, the PMS

Table 6 Rush Hour, Street Light, and China Street datasets PWs results of our proposed LSTM approach after dividing the scene into four SRsSS: Spatial matching Score, MOS: Motion Orientation matching Score, and Avg.: Pathway average matching Score (PMS)

Score	Rush hour	Street light	China street
SS	1.0000	0.8027	0.3830
MOS	0.9039	0.6873	0.7231
Avg.	0.9519	0.7450	0.5531

The given results are the average across all the scene discovered PWs compared to the scene collected tracklets

results are the worst among the three data sets. That is mainly due to China street contains a lot of discontinuities - due to trees, buildings, or Signboards- in its motion map (black segments circled in red in Fig. 14a). These discontinuities stop the generation process of the scene trajectories at these places. That is the reason of obtaining a short common pathway, that don't match all the scene collected tracklets and so the PMS results are the worst among the other data sets (Rush Hour and Street Light).

One of the possible ways to remedy these effects is to make a preprocessing on the crowd scene to detect these defects and remove them. Dilation, for example, maybe used with different parameters on the scene static regions based on the type of the motion surrounding them, which can discriminate if these static areas are a true static area or a defect in the scene. In our future work, we will consider fixing such defects.

4.5 Anomaly detection

LSTM predictive models can be used in anomaly detectors as explained in Section 3.2.2. To test that, we synthesized four kinds of anomalous tracklets (fast, slow, opposite direction, and out of motion area tracklets), that can reflect the four types of anomalies mentioned in Section 3.2.2; these correspond to panic situation, gate overstocking, opposite direction, and moving in prohibited area respectively as shown in Fig. 15a.

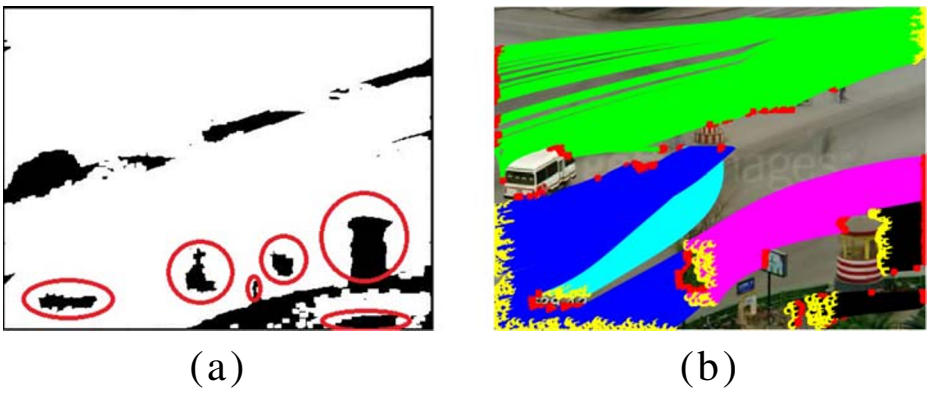

Fig. 14 China Street scene. **a** The scene motion map after circling some of the scene static areas. **b** The scene common pathways generated by our LSTM based approach

Fig. 15 **a** Four types of synthesized tracklets in the Marathon scene: fast tracklet (red), slow tracklet (black), opposite direction tracklet (cyan), and moving in prohibited area tracklet (magenta); versus the original scene tracklet (green). All tracklets are going in the direction from the yellow circle to the red triangle. **b** The motion map of the Marathon dataset. (The details of how we obtained the motion map of any scene is fully explained in our previous work [23])

As mentioned in Section 3.2.2, to detect any of the four defined types of anomalies, the tracklets are extracted from the scene frames, after that for each tracklet of the extracted tracklets, the steps of Algorithm 2 are applied.

4.5.1 Anomaly detection results for different data sets

Since GC data set lacks anomalous data [14], to evaluate our proposed approach in detecting anomalies, we synthesized 100 abnormal tracklets that represent the last-mentioned four types of anomalies (25 tracklet for each type). Because of the complexity of the GC data set motion patterns, we synthesized the tracklets in the Marathon data set as shown in Fig. 15a.

Using the obtained data set motion map[40] (Fig. 15b), and the six LSTM predictive models trained on the scene tracklets, we applied our proposed anomaly metric (discussed in Section 3.2.2) on the 100 synthesized abnormal tracklets, in addition to other 100 normal tracklets of the scene (randomly chosen) to evaluate our model ability to detect anomalous tracklets. The results of this experiment are reported in Table 7. The area under the curve (AUC) is used as an evaluation metric to compare the obtained results versus the ground truth data (the 100 synthesized abnormal tracklets, in addition to the other 100 normal tracklets of the scene.).

Anomaly detection, in this case, is at the tracklet level [14], which means that we are just reporting whether or not the tracklet is anomalous. Another level of anomaly detection is frame level [14]. In this level, the concern is about detecting the frames that contain anomalous behavior. In our proposed anomaly metric, we can also report the anomaly at the frame level by reporting all the frame numbers that contain the points of the anomalous tracklet. In addition to that our proposed anomaly metric can also report the local area of the scene (SR) that contains the locations of the tracklet points as an anomalous area.

[40]Motion map of any scene is a binary image that we create for that scene by marking all the pixels that contain any motion dynamics by 1 and all the other pixels of the image with zero. So as shown in Fig. 15b (which shows the motion map of the Marathon dataset) all the static areas of the scene that don't contain any kind of motion will take the value of zero. We used these motion maps to identify whether or not any tracklet is moving outside the active motion area of the scene.

Table 7 Area under the curve (AUC) results of our proposed anomaly metric on the Marathon dataset, and the three UMN panic datasets [33], versus Hassanein et al. [14] approach in detecting the anomalies in the mentioned datasets

	Marathon	UMN 1	UMN 2	UMN 3
Proposed	0.99	0.97	0.94	0.93
Hassanein et al.	—	0.97	0.91	0.83

Three panic scenes of the UMN data set [33] (shown in Fig. 16 and labeled in our results with names UMN 1, UMN 2, and UMN 3 respectively), were also used to evaluate our proposed anomaly detection criterion. Each of the three scenes contains a number of frames with normal motion (which we used in training four LSTM predictive models for each scene after dividing it into four overlapped SRs), and then followed by another number of frames that contain the panic scenario (tracklets collected from these frames were tested using our proposed anomaly metric).

The AUC results of Marathon and UMN data sets are shown in Table 7. For these three datasets, we compared our obtained results with those of Hassanein et al. [14]. Results illustrate that our proposed LSTM predictive models can robustly detect the abnormal scenarios with high accuracy over different types of scenes.

4.6 Experimental hyper-parameters setting

The setting of the empirical values of the hyper-parameters used in our experiments are shown in Table 8. In the following we will discuss our intuitions of choosing and setting these empirical values.

First, the selected values for the epochs number, batch size, and validation percentage were selected after many experiments to define the most appropriate parameters for our problem over the various data sets. The value of the overlap area between the SR and its neighbor SRs is empirically selected to be 15% of the total SR area in every direction(in

Fig. 16 Three panic scenes from the UMN data set [33]. The first row contains samples of normal behavior, and the second row contains abnormal ones

Table 8 The empirical values of the hyper-parameters used in our experiments

Parameter	Value	Dataset
Epochs number	20	all
Batch size	1	all
Validation %	10%	all
Overlap area	15%	all
input points	9	GC
input points	4	all - GC
θ_{FOV}	50°	all
δ	300 pixel	all
BW (Mean shift)	135 pixel	all
SRs number	16	GC
SRs number	6	Marathon
SRs number	4	3 small data sets

all: all the five data sets. all - GC: all the data sets except GC. 3 small data sets: Rush Hour, Street Light, and China city data sets

all the data sets used in our experiments) after trying multiple values. After these trials, we selected that value to guarantee both the smooth transition between the SRs and the slight affect on the training time complexity of the SR LSTM predictive model.

Regarding the number of input points, that number is mainly depending on the scene resolution. So, after extensive analysis for the appropriate number of these input locations, we empirically set 4 locations for the scenes with resolution lower than or equal $720x480$ pixels, and 9 locations for those greater than that resolution to ensure gathering enough information about the previous history of the trajectory compared to the scene resolution.

For the field of view distance parameter θ_{FOV}, and angle parameter δ, the basic rule of these parameters in our MUDAM approach is to reduce the search space of each MU by defining the neighborhood of a given MU instead of searching the whole scene and increasing the algorithm time complexity. To achieve that, and intuited by the observations of objects movements inside different crowd scenes, we concluded that any moving object tries locally to find its way through the crowd of the scene to reach its destination. During this object movement, it may slightly change its movement direction to avoid collisions, or very crowded places and then revert back to its basic direction to reach its destination. So to handle these observations (direction change), we empirically selected the FOV angle to be $\theta_{FOV} = 50°$ to allow a variety of changes in the motion direction. The change in the value of this parameter is left for our future work.

The value of the field of view distance δ was selected after contemplating the relationship between the number of the obtained entrance and exit MUs, with changing the δ value while fixing the θ_{FOV} value at 50°. This relation is shown in Fig. 17.

As shown in the figure, there is stability in the number of the retrieved entrance and exit MUs for all the values of δ greater than 260 pixels. By increasing the value of the δ parameter, that means increasing the search space of the current MU when trying to select its neighborhood (previously discussed in the connectivity relation in Section 3.1.2) and so more computational cost for the proposed algorithm. That is why, the value of the

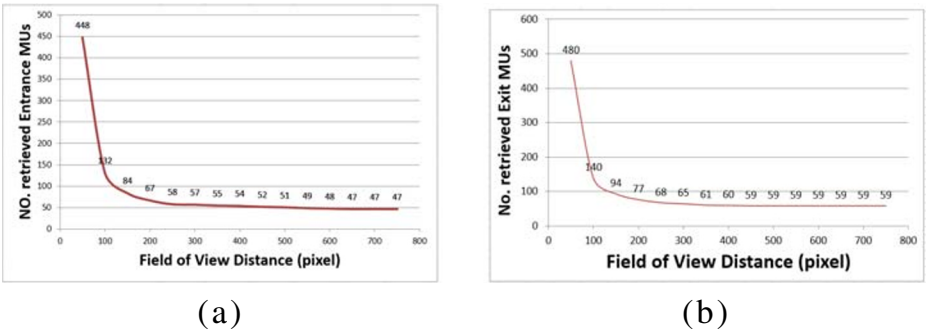

Fig. 17 The relationship between the number of retrieved entrance and exit MUs and the value of the field of view distance δ in pixel

δ parameter was selected to be 300, which guarantees a stable number of the retrieved entrance and exit MUs, and also not too large to minimize the computational cost.[41]

The BW parameter value was empirically selected after many experiments on the GC data set and the other four data sets to have a value that can properly work for the all scenarios and diversity of crowd dynamics. Also, currently, there is research work on the problem of selecting the best value of the mean shift bandwidth parameter such as [3, 4, 8]. So this point specifically will be more considered in our future work.

Regarding the SRs number for each data set; As we concluded from our work Section 4.4.1, increasing the number of the scenes SRs will provide a better common PWs results, but the computational resources needed to process the whole scene will be increased too. So guided by that conclusion, and with the small enhancement in the results (SS, MOS, and AVG.) of Marathon dataset shown in Table 4 when using 6 SRs instead of 4 SRs, we preferred to use 4 SRs for the other three data sets (Rush Hour, Street Light, and china street). That is mainly because that number of divisions will be enough to give us good results without increasing the needed computational resources. For GC data set, this number of SRs was empirically chosen to give good and smooth PWs results in an appropriate processing time using the available hardware resources. But generally, in future work, we will investigate how to automatically set this parameter through a pre processing stage that studies the complexity of motion dynamics inside the scene.

Overall, The automatic determination of any empirical parameters mentioned in this work will be investigated in our future work.

[41] The same parameter values of θ_{FOV} and δ can be applied for the other four data sets (Shown in Table 2) too without any conflict, because θ_{FOV} is a parameter that shows how much tolerance we permit for our system to accept a possible variation in its current orientation. As the GC scene is considered the most complicated scene where any moving person is permitted to go in any direction, we can argue that the same parameters that can handle the GC scene can as well work effectively with other video scenes. The δ parameter also shows the distance that each MU searches to find its possible neighboring MUs. For the GC data set the stability of this parameter was obtained at 260 pixels as was shown in Fig. 17, also beyond that value the stability in the number of the retrieved MUs is guaranteed and the only more cost will be a computational one. Relative to the GC scene all the other four data sets are smaller in the dimension. So taking the same parameters of the GC will also achieve the stability in the number of the retrieved MUs.

5 Conclusion

In this paper, we proposed two approaches for crowd scene analysis. The first is based on motion units and meta-tracking [23]. In this approach the scene is divided into dynamic divisions called the Motion Units (MUs) based on the scene local motion characteristics. A connectivity relation is defined to analyze the relationship between these MUs to retrieve the crowd scene entrance/exit gates. A transition relation and a Motion Unit Dynamics Acquisition Model (MUDAM) are derived to apply a meta-tracking procedure on the scene to retrieve the scene common pathways. Due to some limitations in this approach, and in addition to the need of detecting anomalous behaviors that may happen in the crowd scene, a new LSTM based approach for analyzing the crowded scenes is proposed. In this approach we divide the scene into a number of spatially overlapped parts called Super Regions (SRs) and then train an LSTM predictive model for each SR using the extracted tracklets inside that SR. The proposed approach is then used in discovering the scene's common pathways, considering that the scene gates are given. In our experiments, we used the gates obtained by the MUs and meta-tracking technique as an input to the LSTM based approach. An anomaly metric is also proposed to detect four abnormal situations that may happen inside the crowd scene. The two proposed approaches have been assessed against two of the state of art approaches in video content analytics in addition to the ground truth pathways of the challenging New York's Grand central dataset. For more evaluation of the proposed approaches, four other datasets were also used. the proposed LSTM based approach was tested in terms of identifying anomalous activities on several scenes. The experimental results show that our proposed approaches outperform the other state of the art approaches in terms of detecting the scene gates and pathways, and also in detecting anomalous scenarios that may happen. In future work, we will consider fixing the scene defects that affect the performance of our proposed algorithms such as those mentioned in the results of the China street data set. We also will consider the adaptive scene SRs division process, which can be adaptive to the motion dynamics in the scene (more SRs in highly dynamical areas of the scene and vice verse) and the shape of these SRs. The proper number of input sequential points to the LSTM predictive model is another factor that will be investigated too. One, of the important points, also is changing the dominant pathways over different time of the day or different days (non-stationarity of the motion), we also intend to analyze this situation and give our approach the ability to handle such situation. Also, we intend to study more complex anomalous behaviors such as people grouping or splitting in the crowd scene, and the effect of these actions on the scene motion flow. Also, we plan to identify some specific events in the crowd scenes such as putting something on the unattended floor and leaving it, which is very important as a security issue. Also as a need for these new anomalous scenarios, we intend to employ a neural network-based approach for more accurate and adaptive detection of these anomalies.

Acknowledgements This work is Funded by the Science and Technology Development Fund STDF 992 (Egypt); Project id: 42519 - "Automatic Video Surveillance System for Crowd Scenes".

References

1. Ali S, Shah M (2008) Floor fields for tracking in high density crowd scenes. In: European conference on computer vision. Springer, Berlin, pp 1–14

2. Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S (2016) Social lstm: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 961–971
3. Arias-Castro E, Mason D, Pelletier B (2016) On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. J Mach Learn Res 17(1):1487–1514
4. Chacón JE, Monfort P (2013) A comparison of bandwidth selectors for mean shift clustering. arXiv:1310.7855
5. Chen K, Kamarainen JK (2016) Pedestrian density analysis in public scenes with spatiotemporal tensor features. IEEE Trans Intell Transp Syst 17(7):1968–1977
6. Chongjing W, Xu Z, Yi Z, Yuncai L (2013) Analyzing motion patterns in crowded scenes via automatic tracklets clustering. Chin Commun 10(4):144–154
7. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
8. Comaniciu D, Ramesh V, Meer P (2001) The variable bandwidth mean shift and data-driven scale selection. In: Eighth IEEE international conference on computer vision, 2001. Proceedings. ICCV 2001, vol 1. IEEE, pp 438–445
9. Cong Y, Yuan J, Liu J (2013) Abnormal event detection in crowded scenes using sparse representation. Pattern Recogn 46(7):1851–1864
10. Conte D, Foggia P, Sansone C, Vento M (2004) Thirty years of graph matching in pattern recognition. Int J Pattern Recogn Artif Intell 18(03):265–298
11. Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1110–1118
12. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE International conference on Acoustics, speech and signal processing (ICASSP). IEEE, pp 6645–6649
13. Hassanein AS, Hussein ME, Gomaa W (2016) Semantic analysis for crowded scenes based on non-parametric tracklet clustering. In: IJCAI, pp 3389–3395
14. Hassanein AS, Hussein ME, Gomaa W, Makihara Y, Yagi Y (2018) Identifying motion pathways in highly crowded scenes: a non-parametric tracklet clustering approach. Computer Vision and Image Understanding
15. Jodoin PM, Benezeth Y, Wang Y (2013) Meta-tracking for video scene understanding. In: 2013 10th IEEE International conference on advanced video and signal based surveillance (AVSS). IEEE, pp 1–6
16. Kratz L, Nishino K (2009) Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: 2009 IEEE Conference on computer vision and pattern recognition, 2009. CVPR 2009. IEEE, pp 1446–1453
17. Kratz L, Nishino K (2012) Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. IEEE Trans Pattern Anal Mach Intell 34(5):987–1002
18. Kuo CH, Huang C, Nevatia R (2010) Multi-target tracking by on-line learned discriminative appearance models. In: 2010 IEEE Conference on computer vision and pattern recognition (CVPR). IEEE, pp 685–692
19. Li W, Mahadevan V, Vasconcelos N (2014) Anomaly detection and localization in crowded scenes. IEEE Trans Pattern Anal Mach Intell 36(1):18–32
20. Li T, Chang H, Wang M, Ni B, Hong R, Yan S (2015) Crowded scene analysis: a survey. IEEE Trans Circ Syst Video Technol 25(3):367–386
21. Mehran R, Oyama A, Shah M (2009) Abnormal crowd behavior detection using social force model. In: IEEE Conference on computer vision and pattern recognition, 2009. CVPR 2009. IEEE, pp 935–942
22. Mehran R, Moore BE, Shah M (2010) A streakline representation of flow in crowded scenes. In: European conference on computer vision. Springer, Berlin, pp 439–452
23. Moustafa AN, Hussein M, Gomaa W (2017) Gate and common pathway detection in crowd scenes using motion units and meta-tracking. In: 2017 International conference on digital image computing: techniques and applications (DICTA). IEEE, pp 1–8
24. Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: International conference on machine learning, pp 1310–1318
25. Pele O, Werman M (2010) The quadratic-chi histogram distance family. In: European conference on computer vision. Springer, Berlin, pp 749–762
26. Saleemi I, Hartung L, Shah M (2010) Scene understanding by statistical modeling of motion patterns. In: 2010 IEEE Conference on computer vision and pattern recognition (CVPR). IEEE, pp 2069–2076
27. Shao J, Change Loy C, Wang X (2014) Scene-independent group profiling in crowd. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2219–2226

28. Su H, Yang H, Zheng S, Fan Y, Wei S (2013) The large-scale crowd behavior perception based on spatio-temporal viscous fluid field. IEEE Trans Inform Forensics Secur 8(10):1575–1589
29. Su H, Dong Y, Zhu J, Ling H, Zhang B (2016) Crowd scene understanding with coherent recurrent neural networks. IJCAI 1:2
30. Topkaya IS, Erdogan H, Porikli F (2016) Tracklet clustering for robust multiple object tracking using distance dependent Chinese restaurant processes. SIViP 10(5):795–802
31. Tripathi G, Singh K, Vishwakarma DK (2018) Convolutional neural networks for crowd behaviour analysis: a survey. Vis Comput, 1–24
32. Tomasi C, Kanade T (1991) Detection and tracking of point features. School of Computer Science, Carnegie Mellon Univ. Pittsburgh
33. UMN (2006) Unusual crowd activity dataset of University of Minnesota. http://mha.cs.umn.edu/movies/crowdactivity-all.avi. Accessed: 2010-09-30
34. Wang X, Yang X, He X, Teng Q, Gao M (2014) A high accuracy flow segmentation method in crowded scenes based on streakline. Optik-Int J Light Electron Opt 125(3):924–929
35. Wen ZQ, Cai ZX (2006) Mean shift algorithm and its application in tracking of objects. In: 2006 International conference on machine learning and cybernetics. IEEE, pp 4024–4028
36. Wu S, Moore BE, Shah M (2010) Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. In: 2010 IEEE Computer society conference on computer vision and pattern recognition, San Francisco, pp 2054–2060
37. Xue H, Huynh DQ, Reynolds M (2017) Bi-prediction: pedestrian trajectory prediction based on bidirectional LSTM classification. In: 2017 International conference on digital image computing: techniques and applications (DICTA). IEEE, pp 1–8
38. Yi S, Li H, Wang X (2015) Understanding pedestrian behaviors from stationary crowd groups. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3488–3496
39. Zhou B, Wang X, Tang X (2011) Random field topic model for semantic region analysis in crowded scenes from tracklets
40. Zhou B, Wang X, Tang X (2012) Understanding collective crowd behaviors: learning a mixture model of dynamic pedestrian-agents. In: 2012 IEEE Conference on computer vision and pattern recognition (CVPR). IEEE, pp 2871–2878
41. Zhuang N, Ye J, Hua KA (2017) Convolutional DLSTM for crowd scene understanding. In: 2017 IEEE International symposium on multimedia (ISM). IEEE, pp 61–68
42. Zou Y, Zhao X, Liu Y (2015) Detect coherent motions in crowd scenes based on tracklets association. In: 2015 IEEE International conference on image processing (ICIP). IEEE, pp 4456–4460

Abdullah N. Moustafa is a PhD student and research assistant at Egypt - Japan University for science and technology (E-JUST)., He was appointed as a lecturer assistant in Menofia University, Computer Science and Engineering Department, in the Faculty of Electronic Engineering, Menofia 32952, Egypt. He obtained his BSc. and MSc. (both Excellent) in Computer Science and Engineering from Menofia University, in May 2010 and December 2014. He served as a reserve officer in the Egyptian army starting from October 2010 for three years. His main research interests are Digital Image Processing, Medical Imaging, Computer vision, and Machine learning. He worked in a funded project from ITAC. His recent research work is focusing on analyzing the crowd scenes and detecting the anomalies in such scenes.

Walid Gomaa is currently an associate professor in E-JUST (on-leave from Alexandria University). He is the founder and director of the Cyber-Physical Systems lab. Dr. Gomaa obtained his PhD degree in computer science from the University of Maryland College Park, United States in 2007. He then held an INRIA post-doctorate position in Loria lab, France, from 2008 till the end of 2009. During such period his research focus was on theoretical foundations of computation, especially, the study of computable analysis which is the infusion of classical computability theory with mathematical analysis. Dr. Gomaa started his position in E-JUST since 2010 where he developed research directions aiming at essentially data analytics in multiple domains, such as drug design, traffic management, robot localization, etc, based on the state-of-the-art statistical machine learning techniques. He has acquired several funded projects from local agencies such as ITAC, STDF, ASRT, etc. He has strong collaborations with reputable international institutes including the National Institute of Informations in Tokyo Japan where he has been invited several times, the International Center for Theoretical Physics in Trieste Italy where he is an associate, LIX lab in Ecole Polytechnique in France where he has been invited several times as well as in Loria Lab in Nancy France. Dr. Gomaa has got the IBM Faculty award in 2010. His recent research and projects focus essentially on different aspects of human activity recognition using different sensory modalities, in particular, vision and IMU sensors.

Affiliations

Abdullah N. Moustafa[1,2] · Walid Gomaa[1,3]

Walid Gomaa
walid.gomaa@ejust.edu.eg

1 Cyber Physical Systems Laboratory, Egypt-Japan University of Science and Technology, New Borg El-Arab City, Alexandria, Egypt

2 Computer Science and Engineering Department, Faculty of Electronic Engineering, Menofia University, Menofia, Egypt

3 Faculty of Engineering, Alexandria University, Alexandria, Egypt