

# Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths.

B. Cancela, A. Iglesias, M. Ortega, M. G. Penedo  
VARPA Group, Universidade da Coruña  
Campus de Elviña s/n, A Coruña, Spain

<http://www.varpa.org/>

## Abstract

*This paper presents a novel methodology for modelling pedestrian trajectories over a scene, based in the hypothesis that, when people try to reach a destination, they use the path that takes less time, taking into account environmental information like the type of terrain or what other people did before. Thus, a minimal path approach can be used to model human trajectory behaviour. We develop a modified Fast Marching Method that allows us to include both velocity and orientation in the Front Propagation Approach, without increasing its computational complexity. Combining all the information, we create a time surface that shows the time a target need to reach any given position in the scene. We also create different metrics in order to compare the time surface against the real behaviour. Experimental results over a public dataset prove the initial hypothesis' correctness.*

## 1. Introduction

Modelling human behaviour is a huge field of study in computer vision. One of the most active topics is related with the trajectory analysis, that is, how is the people usual behaviour, having a special interest in areas like video surveillance. However, a lot of questions remain unsolved. For instance, what information makes people decide which path they should use to reach their goal?. Or what makes a movement abnormal? The answer for the questions are crucial in order to obtain a good trajectory modelling.

There exist two different approaches in the literature to model human trajectory behaviour. On one hand there are the Computer Vision techniques. They are mainly focused in classifying all the trajectories in the scene, mostly using clustering techniques, like Hybrid [12], agglomerative [3], where we merge clusters until we obtain the desired number; divisive [1], Graph-based [14], Spectral [11] or direct [17], using techniques such as k-means or fuzzy  $c$  means.

These techniques require every path to have the same number of detections to update the patterns, making the updating procedure very difficult. More recently, new techniques are used, like the use of non-parametric Bayesian models [27], [26], [13] or using models to predict the motion behaviour [6].

On the other hand there are the techniques based in *social force* models [10]. They are based in the idea that some stimuli, like the scene properties and other people interactions, affect the pedestrian trajectory [9], [2]. This approach is often used in computer graphic schemes, developing a set of different *forces* that are added to infer the new movement [23], [20], [24]. The main drawback of these techniques is that although they are good approaches to model the usual human behaviour, there exist infinite solutions to model a normal behaviour. Thus, how can we use these kind of systems to decide whether a trajectory is abnormal or not?

More recently, techniques that try to merge computer vision techniques with social models are arising. For instance, some works introduce the *social force* model to detect abnormal behaviour [15], [19]. Flow models were also included to predict crowd behaviour [16]. More recently, the use of minimal paths were introduced to model the usual behaviour [5], [4]. However, this approach does not take into account velocity patterns or orientation in the propagation procedure.

Based on the latter, we propose a new methodology to model pedestrian trajectory behaviour. Our solution is based in the hypothesis that, typically, when people try to reach a destination, they use the path that takes less time. We take into account both the velocity and the orientation of the usual motion to create a *time surface*, where each node shows the time needed to reach it if the person behaviour is usual. We create a modified Fast Marching Method (FMM) [22] which includes the mentioned extra information without increasing the computational cost. Using this technique, only a potential and a velocity surfaces are needed in order to establish the so-called *time surface*. We also present different metrics to test a path's "degree of abnormality", in

order to experimentally prove that our initial hypothesis is correct. Due to the lack of datasets providing some ground truth, we provide a theoretic statistical probe of correctness, along with some visual confirmation, as other approaches do. No quantitative results against other techniques are provided, since no abnormal detection results were provided in any other paper, which focus in clustering techniques our algorithm do not need.

This paper is organized as follows: section 2 describes our modified FMM and the metrics created to model the pedestrian behaviour; section 3 shows implementation details about our algorithm; section 4 shows some experimental results and section 5 offers conclusions and future work.

## 2. The Governing Equations

In this section we develop our model for pedestrian trajectory modelling. We begin to set the hypothesis that are used to establish our methodology. We subsequently introduce our modified FMM that allows us to create the *time surface*. Finally, we introduce some metrics in order to test the pedestrian behaviour. We defer the implementation of these techniques to section 3. For coherence with previous approaches, we are going to follow the notation introduced in [7].

**Hypothesis 1** *Each person tries to reach a geographic goal.*

Since this model tries to model pedestrian trajectories, it is needed that targets have the intention to reach some goal within or outside the scene. As a consequence, people that are stopped or moving erratically are considered as abnormal movements.

**Hypothesis 2** *The trajectory used to reach the goal is ruled by the common pedestrian behaviour.*

This is a crucial point in this algorithm. In [5], the minimal path is ruled by a potential image that contains, for every node, the number of people that reach it. However, in our algorithm, we also include the velocity of the targets. Despite this, our model will also be driven by the people count estimation.

**Hypothesis 3** *People move at the maximum speed possible.*

That is, the speed of every person is defined by a velocity field.

Having this hypothesis in mind, we can conclude that the usual path can be modelled as a minimal path approach. We propose a modified FMM in order to create a *time surface*, which is created taking into account the information about both the people frequency and their velocity. In algorithm 1 the method is explained. In essence, the structure is similar to the FMM. It only differs in the updating procedure, since

---

### Algorithm 1 Time Surface Fast Marching method

---

Definitions:

- $p_0$ : the initial point, the first time a target is tracked.
- $U$ : surface of minimal action, driven by the people frequency.
- $T$ : time surface: every node contains the time needed to reach it starting in the initial point.
- *Alive* set: points of the grid for which  $U$  has been computed and it will not be modified.
- *Trial* set: next points in the grid to be examined (4-connectivity) for which a estimation of  $U$  is computed using the points in *alive* set.
- *Far* set: the remaining points of the grid for which there is not an estimate for  $U$ .

Initialization:

- For each point in the grid, let  $U_{i,j} = \infty, T_{i,j} = \infty$  (large positive value).  
Put all points in the *far* set.
- Set the start point  $(i, j) = p_0$  to be zero:  
 $U_{p_0} = 0, T_{p_0} = 0$ , and put it in the *trial* set.

Marching loop:

- Select  $p = (i_{min}, j_{min})$  from *trial* with the lowest value of  $U$ .
  - Put  $p$  in *alive* and remove it from the *trial* set.
  - For each of the  $p$ 's neighbours  $(k, l)$  of  $(i_{min}, j_{min})$ :
    - If  $(k, l)$  belongs to *far* set, then put  $(k, l)$  in *trial* set.
    - If  $(k, l)$  is not in *alive* set, then set  $U_{k,l}$  with Equation 3, and  $T_{k,l}$  with Equation 4.
- 

now we have two different output maps:  $U$ , the classical minimal path surface; and  $T$ , the *time surface*, where the time needed to reach every point is stored.  $T$  surface is driven by the  $U$  surface. That means we use the classical minpath updating procedure in order to update the surface. Thus, we need the best horizontal and vertical neighbours to do that. So, having the point  $p_N = (i, j)$  to be updated we define the points  $p_H = p \in \{(i+1, j), (i-1, j)\} | \min U(p)$  and  $p_V = p \in \{(i, j+1), (i, j-1)\} | \min U(p)$ . Then, we have the points

$$p_a = p \in \{p_H, p_V\} | \min U(p) \quad (1)$$

$$p_b = p \in \{p_H, p_V\} | \max U(p), \quad (2)$$

which are the two points used to perform the updating procedure. Hence, to update the minimal action surface  $U$  we

use the following equation

$$U_{p_N} = \begin{cases} \frac{U_{p_a} + U_{p_b} + \sqrt{\Delta_1}}{2} & \text{if } \tilde{P}_{p_N} > (U_{p_b} - U_{p_a}) \\ U_{p_a} + \tilde{P}_{p_N} & \text{otherwise} \end{cases}, \quad (3)$$

being  $\tilde{P}(p_N)$  the potential surface, which satisfies that  $\tilde{P} > 0$ ; and  $\Delta_1 = 2\tilde{P}_{p_N}^2 - (U_{p_b} - U_{p_a})^2$  the discriminant. To update the time surface, we use the same points  $p_a$  and  $p_b$  defined in the minimal action surface procedure. The equation is similar to the previous equation, that is,

$$T_{p_N} = \begin{cases} \frac{T_{p_a} + T_{p_b} + \sqrt{\Delta_2}}{2} & \text{if } \tilde{P}_{p_N} > (U_{p_b} - U_{p_a}) \\ T_{p_a} + v_{p_N} & \text{otherwise} \end{cases}, \quad (4)$$

being  $v_{p_N} = \frac{1}{v_{p_N}}$  the velocity inverse; and  $\Delta_2 = 2v_{p_N}^2 - (T_{p_b} - T_{p_a})^2$  the discriminant. Note the condition  $\tilde{P}_{p_N} > (U_{p_b} - U_{p_a})$  refers to the first equation, which guarantees that  $\Delta_1 > 0$ . However, in the second equation we cannot guarantee this situation in the discriminant. To solve that, we introduce a restriction in the procedure that allows the *time surface* to change the minimal action surface  $U$ . If we found that  $\Delta_2 < 0$ , we update both the minimal action surface  $U$  and the *time surface*  $T$  with the default condition. The computational complexity of the algorithm remains equal to the FMM, that is,  $\mathcal{O}(M \log M)$ , being  $M$  the numbers of nodes in the potential surface  $\tilde{P}$ .

## 2.1. Behavioural Metrics

Once we have computed the surface  $T$ , we have to establish whether a trajectory is abnormal or not. We can define a path as  $P = \{p_0, p_1, \dots, p_M\}$ , where each point represents positions that are reached for the target, and its associated time as

$$P_t = \{t_{p_0}, t_{p_1}, \dots, t_{p_M}\}. \quad (5)$$

Intuitively, without any information about the environment, the human brain detect as abnormal behaviour erratic movements, such as sudden orientation changes or zigzag movements. However, taking into account the scene properties, it may be the only way to reach the target, causing the path to be usual. An example of this behaviour could be a mountain road, climbing to the top like a snake. However, in our assumption, we only take into account the initial and final point of the trajectory. According with our assumptions, the average time required to reach a goal in the scene, starting at any given position, is stored in the surface  $T$ . Thus, a new hypothesis is established, that is,

**Hypothesis 4** *If a path  $P$  have an usual behaviour, then  $\forall p \in P, T_p \approx P_t(p)$ ,*

that is, the relation between the real and the expected time is  $\frac{T_p}{P_t(p)} \approx 1$ . Note that this idea is similar to the distance

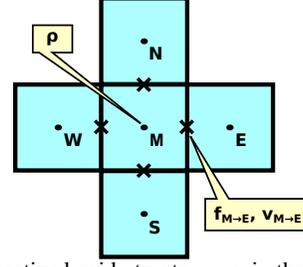


Figure 1. Discretized grid structure.  $\rho$  is the number of times a target reaches the node, whilst  $f$  and  $v$  are both the number of times a target crosses the path in each direction, and their most common speed.

hypothesis performed in [4]. However, we think that the metrics used are not well suited, since the division between the two factors could obtain lower results when the numerator is too small compared with the denominator. Having this in mind, we develop a metric that allows us to measure a path's "abnormality". We called it Time Log-Likelihood (TL), and is defined by

$$TL(p_N) = \|\log(T_{p_N}) - \log(t_{p_N} - t_{p_0})\|, \quad (6)$$

where  $1 < N \leq M$ . Values close to 0 mean the path is correct, while higher values could indicate an abnormal behaviour. Note that this final point mentioned in this metric is not necessarily the moment when the target leaves the scene. It is only a moment when we decide to evaluate a trajectory. This strong advantage allows this method to be used in real-time systems, since the computational complexity of the metric, once the surface  $T$  is computed, is  $\mathcal{O}(1)$ .

## 3. Implementation

The model described in the previous section needs an implementation of the different structures that are used in the algorithm. Specifically, the potential  $\tilde{P}$  and the velocity  $V$  surfaces. To compute these surfaces, we follow a similar approach that can be found in [23]. Since this method uses digital images, we discretized space into a regular grid. For each node/pixel, we store a set of properties, according to the schema shown in Fig. 1, into a 2D array we call  $P$ , being  $\rho$  the number of people that reach the node. We store anisotropic fields with four floats per cell corresponding to the east, north, west and south faces of each pixel ( $\theta = \{0, 90, 180, 270\}$ ).  $f_{M \rightarrow \{E, N, W, S\}}$  are the number of people that crosses each one of the pixel faces, while  $v_{M \rightarrow \{E, N, W, S\}}$  are their most common speed. With this information, we are able to create  $\tilde{P}$  and  $V$  surfaces.

To select the most common speed velocity, we use a Kernel Density Estimator [18]. To estimate the bandwidth, we use the *Rule of Thumb* method [8], [21]. Once we have the probability density function, we select the most common speed as the value with higher probability.

### 3.1. Potential Surface $\tilde{P}$

To model the potential surface we use the  $\rho$  parameter. Thus, having a point  $p$ , its potential value is defined by

$$\tilde{P}(p) = \frac{1}{\rho_p} + \omega, \quad (7)$$

being  $\omega$  the regularization parameter, typically  $\omega = 0$ . We also have to define the  $p$  neighbourhood. The FMM uses a 4-connectivity procedure. However, in our case, we restrict the front-propagation technique to the directions that are commonly used, that is, being  $M = \{p_E, p_N, p_W, p_S\}$  the usual  $p$  4-connectivity neighbours,

$$\forall m \in M, m \text{ is } p\text{'s neighbour} \Leftrightarrow f_{p \rightarrow m} > \alpha, \quad (8)$$

being  $\alpha$  a manual threshold. Note that this sentence does not imply the opposite. For instance,  $p_S$  can be a neighbour of  $p$  but  $p$  could not be a neighbour of  $p_S$ . With this restriction we can include orientation in the front-propagation procedure.

### 3.2. Velocity Surface $V$

As defined in Eq. 4, having any given point  $p$  to be updated, we have to find the value  $v_p = \frac{1}{V_p}$  in order to update the time surface front. When the default condition is used, that is, only the point  $p_a$  is used to update the front it is easy to obtain the velocity, since it is  $V_p = v_{p_a \rightarrow p}$ .

However, when dealing with the first condition, we have two different velocities that have to be combined,  $v_{p_a \rightarrow p}$  and  $v_{p_b \rightarrow p}$ . In order to establish an accurate solution to this problem, we define another surface  $D$  to be computed. This surface stores, for any given node  $p$ , the distance needed to reach it starting in  $p_0$  and following the front-propagation method defined in Algorithm 1. This surface can be computed at the same time the surface of minimal action  $U$  is computed, as the time surface  $T$  does. To update the distance surface we follow the equation

$$D_{p_N} = \begin{cases} \frac{D_{p_a} + D_{p_b} + \sqrt{\Delta_3}}{2} & \text{if } \tilde{P}_{p_N} > (U_{p_b} - U_{p_a}) \\ D_{p_a} + 1 & \text{otherwise} \end{cases}, \quad (9)$$

being  $\Delta_3 = 2 - (D_{p_b} - D_{p_a})^2$  the discriminant, and  $p_a$  and  $p_b$  the best neighbours defined in the minimal action surface procedure. To obtain the value  $V_p$  we make use of the gradient descendant in  $D$ , that is,

$$V_p = \frac{\nabla D_{p_a} v_{p_a \rightarrow p} + \nabla D_{p_b} v_{p_b \rightarrow p}}{\nabla D_{p_a} + \nabla D_{p_b}}, \quad (10)$$

where  $\nabla D_{p_{\{a,b\}}} = \|D_p - D_{p_{\{a,b\}}}\|$ .

## 4. Experimental Results

When trying to test any trajectory analysis, the same problem arises: there is a total absence of ground truth information, except in the BARD dataset [5]. However, this dataset is too small (only over 600 trajectories) and does not have any time information included. Thus, we decided to use a dataset that included a high number of trajectories, the single camera MIT trajectory dataset [25]. It contains 40,453 different trajectories obtained from a parking lot scene within five days. We use half the trajectories as training, and the other half as testing.

Having all of this information, it is very hard to define whether a trajectory is normal or not. In related papers, they use some visual information to probe its effectiveness [25], [27], [26], [28]. In our approach, we also focus the solution as a statistical problem. Since all the earlier attempts to model the human trajectory behaviour have used some learning methods to determine the usual behaviour, we extract the idea that, having no information about the environment, every method consider the most usual paths as normal movements, being the outliers the abnormal ones.

Ideally, we expect an abnormal behaviour measure to have an asymptotic curve, like  $\frac{1}{x}$ , where the most part of the trajectories are normal, with a few abnormal movements. That is, the more erratic a trajectory is, the lower frequency it has. Thus, we can establish an inverse correlation between this two properties. In the top density function in Fig. 2-(a) we can see our metric has this behaviour. However, we have to check that, as we assume, lower values correspond to normal trajectories, while higher values means the opposite. To that end, we length-normalize every trajectory and perform a Fuzzy C means clustering into a large number of clusters. When we visualize the results, we saw that some trajectories detected as normal has some erratic movements in the middle. So, we decided to include two additional metrics: the Mean TL (MTL), which plots the mean of all the  $TL$  measures within the same trajectory, and the Maximum TL (MaxTL), which indicates its higher value. In Fig. 2-(a) we can see the density function of these metrics. We found that these two metrics perform really bad compared with the TL metrics, especially the MaxTL. Additionally, Fig. 2-(b) shows that the number of results near to  $\infty$  is higher in the new metrics. This is really interesting, because when plotting the results of the clusters, according with the MaxTL value, we see some pattern (Fig. 3). The bad accuracy of the MaxTL metric is related to failures in the tracking system. That is, when due to a tracking failure a bad match is provoked, the system can detect it. This is an outstanding property, as it can be used to increase tracking performance.

### 4.1. Crowded Scenes

We demonstrate how this method can detect and classify every trajectory. However, at this point it can be argued

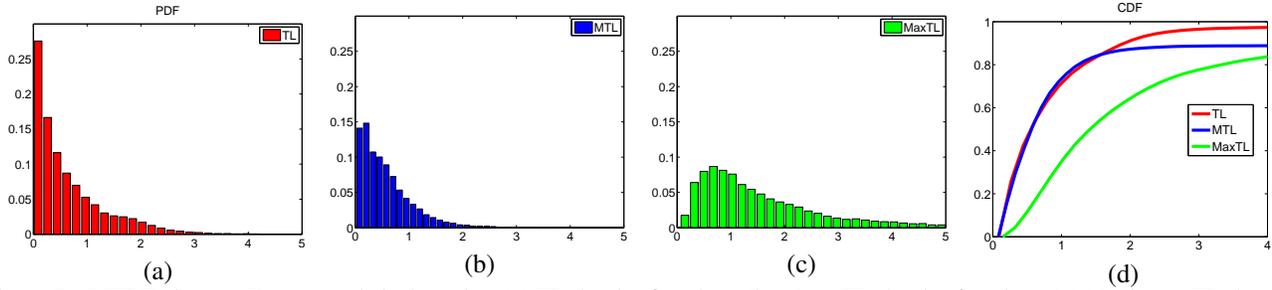


Figure 2. MIT Trajectory Dataset statistical results. (a) TL density function. (b) Mean TL density function. (c) Maximum TL density function. (d) Cumulative density function of each metric. Although the TL metric obtains good results, both its mean and its max value has poor quality. This seems to infer the trajectories are not accurate.

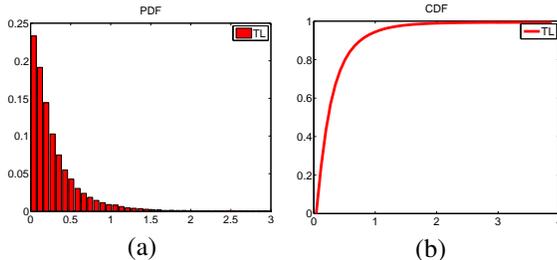


Figure 5. Train Station Dataset statistical results. The results suggest the effect of the rest of the people in a crowded scene has little impact in a target behaviour.

that the method could not work in crowded scenes, where the rest of the targets can affect a target’s behaviour, making that usual direction may not be chosen. To that end, we select the Train Station Dataset [28], where 42,821 trajectories were recorded in only 33 minutes. Performing the same experiments as in the previous dataset we obtain similar results (Fig. 5). That is really interesting, because we can conclude that the effect caused in a target by the rest of the people is not as huge as one may think. And with this hypothesis, we can pre-compute the time surface at the beginning without significantly increasing the metric error. Having this in mind, computing the behaviour at each position can be done in constant time.

In Fig. 4 we can see some trajectories the system detect as normal behaviour. As we can see, some movements were caused because of other people interaction, but the system remains robust against it. In a similar way, in Fig. 6 we can see some abnormal trajectories, and also when the system is changing its decision. This images show how our method can detect the degree of ‘normality’ at every moment a target is detected, providing a very powerful technique for surveillance scenarios.

The results obtained both in normal and high dense scene are better than expected. The fact that the number of people in the scene does not have a high impact in the target’s behaviour makes this system very useful in very different situations. Previous methods group the scene in different

regions with same behaviour [21] or cluster the trajectories in different patterns [25]. In our case, we cluster every ‘time surface’ taking into account each target’s initial position. This idea is very promising, since in the most part of scenes the number of entrances is low. Thus, in future research it would be interesting to create only the surfaces related with these regions, causing the system to detect abnormal behaviours in constant time. To our knowledge, this is the first method that can be able to obtain it, since we do not have to compare the new trajectory against all the different pattern stored in the system, like the other methods do.

## 5. Conclusions

We proposed a novel idea to classify human trajectories, based in the idea that a person uses the path that takes less time to reach its target. Using the information about previous targets, we define a new potential field that is used to create a ‘time surface’ where, starting at any given point in the image, can predict the average time needed to reach any other position, assuming the target has an usual behaviour. Having this information, we introduce a new temporal metric to decide whether a trajectory is abnormal or not. Selecting two complicated scenarios, we prove that our initial hypothesis is correct, having an important contribution to establish a new way to define trajectory behaviour. The lack of datasets providing ground truth make impossible the task of evaluating our algorithm against other state of the art techniques. However, our method offers many advantages against techniques based in clustering or bayesian models, since it is able to compute the degree of ‘normality’ at every trajectory instant, and it is robust against the influence of other targets. Furthermore, it can detect tracking errors, making it suitable to improve any tracking system that can be integrated with our system. In future work we aim to further refine this method, using statistical techniques to model both velocity and time, instead of only using the most usual speed for every position. We also aim to expand this methodology to multiple camera frameworks.

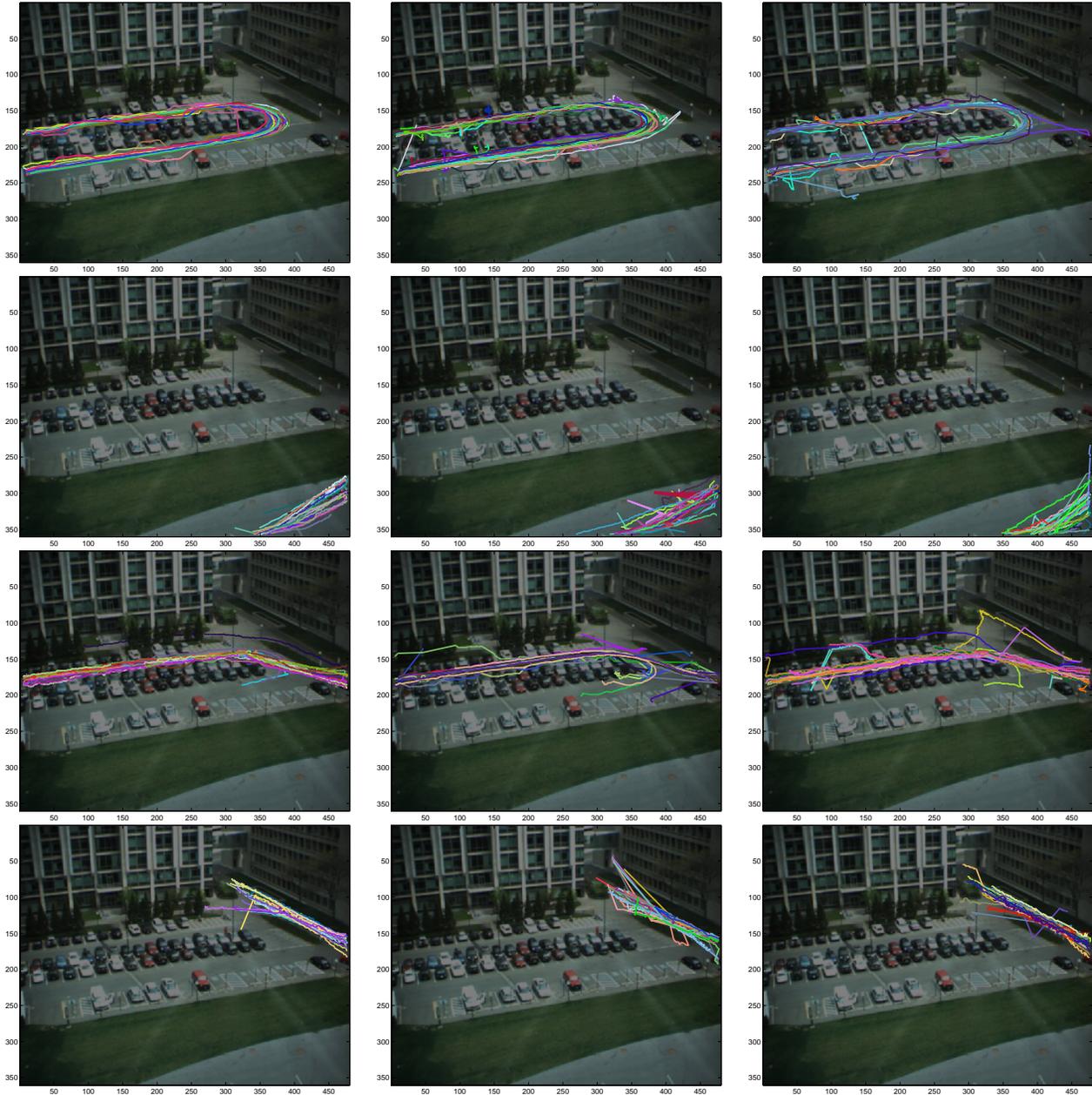


Figure 3. MIT Trajectory Dataset results. In the first column, trajectories with low  $MaxTL$  value. In the second, higher values. In the third, trajectories for which a minimal action surface cannot be created. Trajectory colours are randomly selected.

## References

- [1] D. Biliotti, G. Antonini, and J. P. Thiran. Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)*, volume 2, pages 50–57. IEEE, 2005. **1**
- [2] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525, 2001. **1**
- [3] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 521–524. IEEE, 2004. **1**
- [4] B. Cancela, M. Ortega, A. Fernández, and M. G. Penedo. Trajectory similarity measures using minimal paths. In *Image Analysis and Processing—ICIAP 2013*, pages 400–409. 2013. **1, 3**

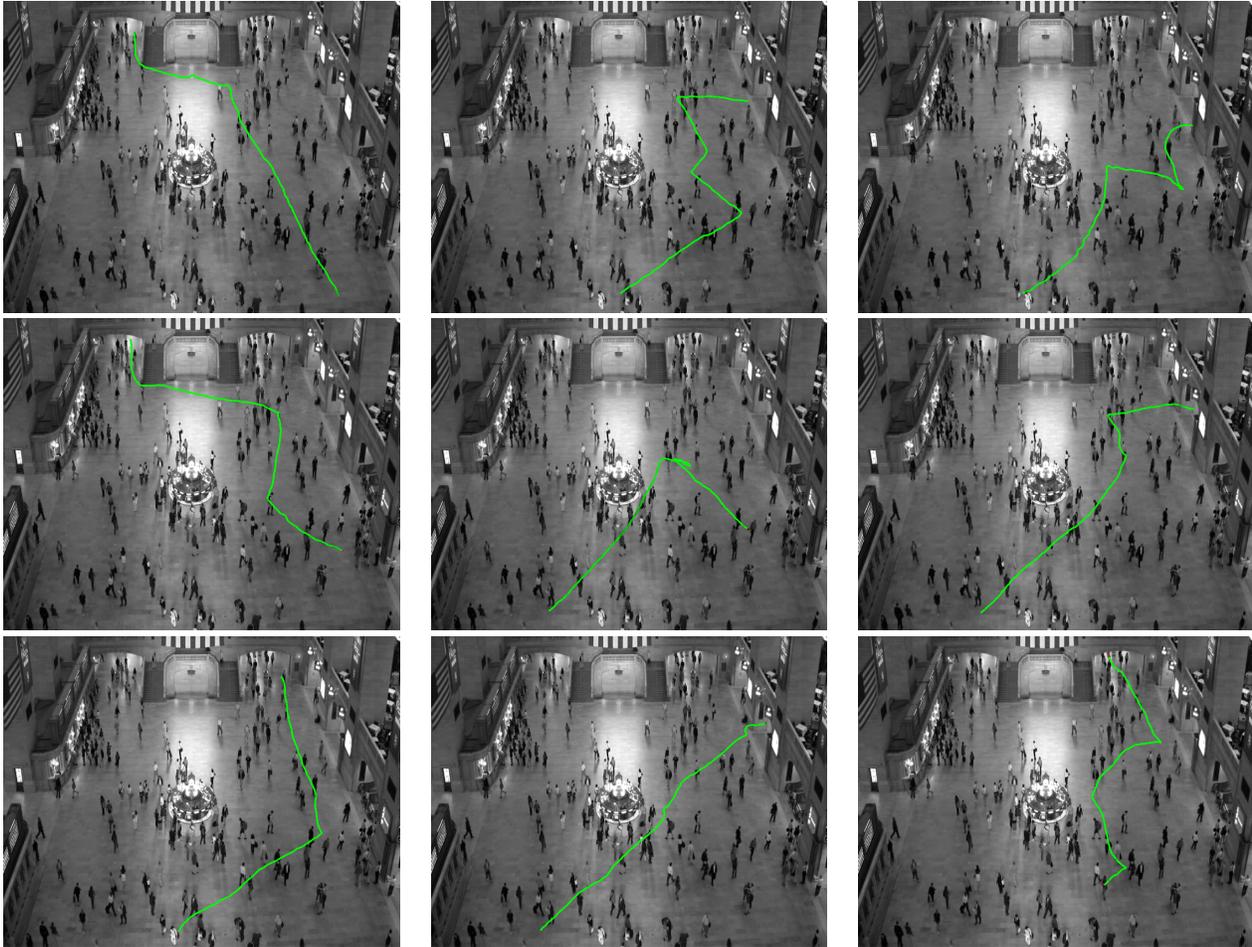


Figure 4. Train Station Trajectory Dataset results. Some examples of trajectories the system marked as normal behaviour.

- [5] B. Cancela, M. Ortega, M. Penedo, J. Novo, and N. Barreira. On the use of a minimal path approach for target trajectory analysis. *Pattern Recognition*, 46(7):2015–2027, 2013. 1, 2, 4
- [6] Z. Chen, L. Wang, and N. H. Yung. Adaptive human motion analysis and prediction. *Pattern Recognition*, 44(12):2902–2914, 2011. 1
- [7] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *International journal of computer vision*, 24(1):57–78, 1997. 2
- [8] K. Dehnad. Density estimation for statistics and data analysis. *Technometrics*, 29(4):495–495, 1987. 3
- [9] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000. 1
- [10] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 1
- [11] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based surveillance video retrieval. *Image Processing, IEEE Transactions on*, 16(4):1168–1181, 2007. 1
- [12] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999. 1
- [13] J. Li, S. Gong, and T. Xiang. Learning behavioural context. *International journal of computer vision*, 97(3):276–304, 2012. 1
- [14] X. Li, W. Hu, and W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 591–594, 2006. 1
- [15] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009. 1
- [16] B. E. Moore, S. Ali, R. Mehran, and M. Shah. Visual crowd surveillance through a hydrodynamics lens. *Communications of the ACM*, 54(12):64–73, 2011. 1
- [17] B. Morris and M. M. Trivedi. An adaptive scene description for activity analysis in surveillance video. In *ICPR*, pages 1–4. IEEE, 2008. 1
- [18] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. 3
- [19] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You’ll never walk alone: Modeling social behavior for multi-target

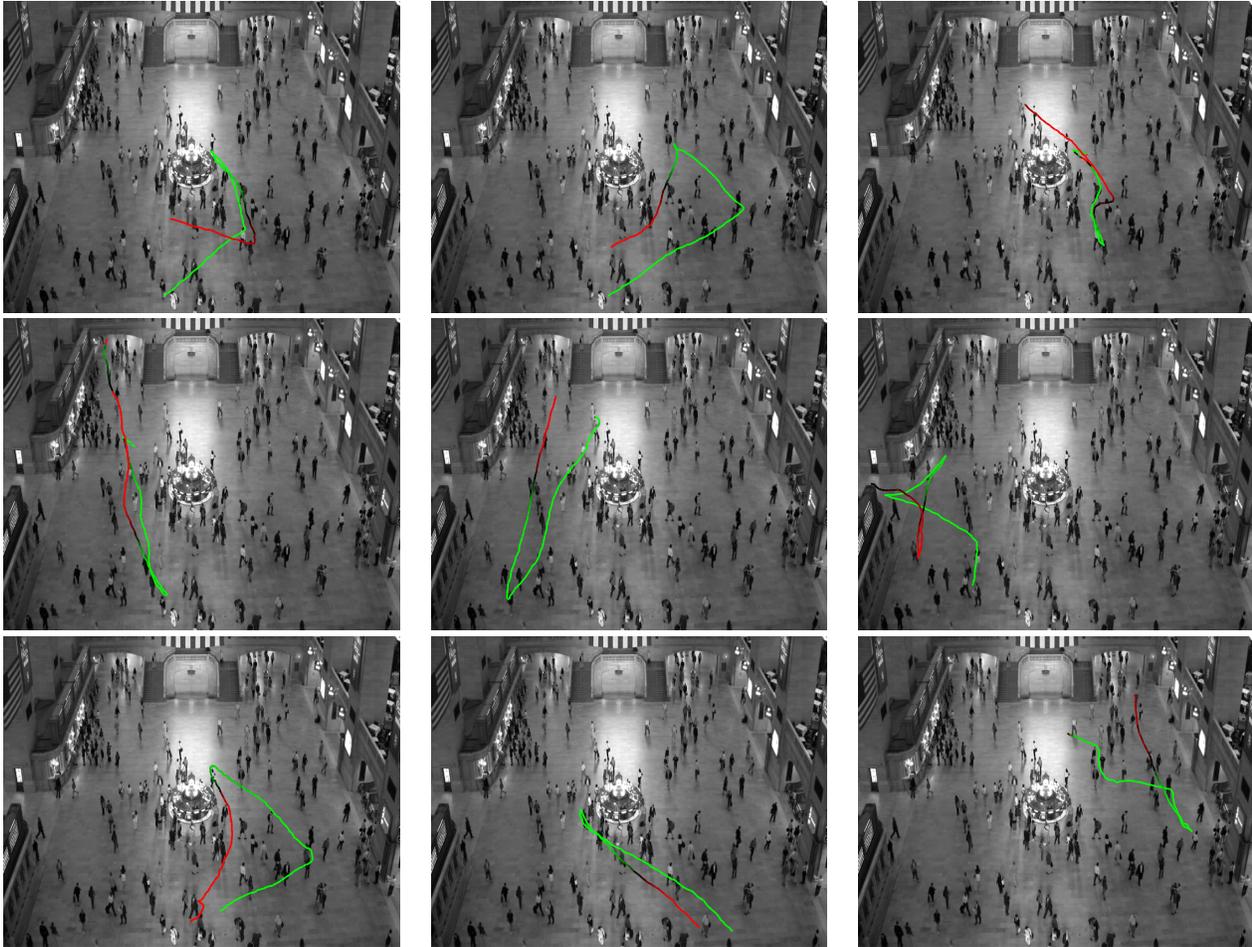


Figure 6. Train Station Trajectory Dataset results. Some examples of trajectories the system marked as abnormal behaviour. The system is able to determine, at any time, the degree of a bnormality (red means the trajectory is abnormal).

- tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009. **1**
- [20] C. Reynolds. Big fast crowds on ps3. In *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, pages 113–121. ACM, 2006. **1**
- [21] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1472–1485, 2009. **3, 5**
- [22] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. **1**
- [23] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1160–1168. ACM, 2006. **1, 3**
- [24] J. van den Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin. Interactive navigation of multiple agents in crowded environments. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 139–147. ACM, 2008. **1**
- [25] X. Wang, K. T. Ma, G.-W. Ng, and W. Grimson. Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008. **4, 5**
- [26] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *International journal of computer vision*, 95(3):287–312, 2011. **1, 4**
- [27] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):539–555, 2009. **1, 4**
- [28] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2871–2878, 2012. **4, 5**