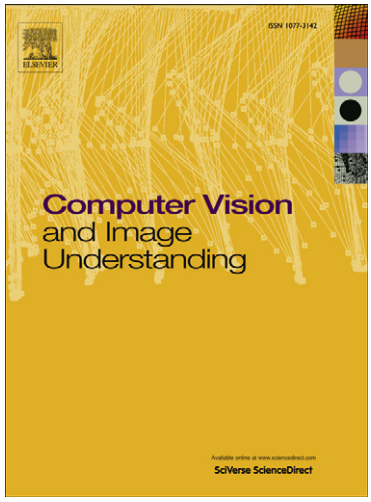


TRASMIL: A Local Anomaly Detection Framework Based on Trajectory Segmentation and Multi-instance Learning

Wanqi Yang, Yang Gao, Longbing Cao

PII: S1077-3142(12)00165-8
DOI: http://dx.doi.org/10.1016/j.cviu.2012.08.010
Reference: YCVIU 1915

To appear in: *Computer Vision and Image Understanding*

Received Date: 9 March 2012
Revised Date: 26 July 2012
Accepted Date: 24 August 2012

Please cite this article as: W. Yang, Y. Gao, L. Cao, TRASMIL: A Local Anomaly Detection Framework Based on Trajectory Segmentation and Multi-instance Learning, *Computer Vision and Image Understanding* (2012), doi: http://dx.doi.org/10.1016/j.cviu.2012.08.010

TRASMIL: A Local Anomaly Detection Framework Based on Trajectory Segmentation and Multi-instance Learning

Wanqi Yang[a], Yang Gao[1a], Longbing Cao[b]

[a] *State Key Laboratory for Novel Software Technology, Nanjing University, China*
[b] *Advanced Analytics Institute, University of Technology Sydney, Australia*

Abstract

Local anomaly detection refers to detecting small anomalies or outliers that exist in some subsegments of events or behaviors. Such local anomalies are easily overlooked by most of the existing approaches since they are designed for detecting global or large anomalies. In this paper, an accurate and flexible three-phase framework TRASMIL is proposed for local anomaly detection based on TRAjectory Segmentation and Multi-Instance Learning. Firstly, every motion trajectory is segmented into independent sub-trajectories, and a metric with *Diversity* and *Granularity* is proposed to measure the quality of segmentation. Secondly, the segmented sub-trajectories are modeled by a sequence learning model. Finally, multi-instance learning is applied to detect abnormal trajectories and sub-trajectories which are viewed as bags and instances, respectively. We validate the TRASMIL framework in terms of 16 different algorithms built on the three-phase framework. Substantial experiments show that algorithms based on the TRASMIL framework outperform existing methods in effectively detecting the trajectories with local anomalies in terms of the whole trajectory. In particular, the MDL-C algorithm (the combination of HDP-HMM with MDL segmentation and Citation kNN) achieves the highest accuracy and recall rates. We further show that TRASMIL is generic enough to adopt other algorithms for identifying local anomalies.

Keywords: Local anomaly detection, Trajectory segmentation, Trajectory representation, Multi-instance learning, HDP-HMM

1. Introduction

Abnormal event detection is a critical research topic in visual surveillance. Basically, the abnormal events are defined as the events which are largely deviated from normal ones. So the goal of the abnormal event detection is to

[1] corresponding author, email: gaoy@nju.edu.cn

automatically discover the potential abnormal events from observations. Due to the unpredictability and diversity of abnormal events, it is usually difficult and infeasible to build a particular classifier for abnormal events. The remaining core problem of abnormal event detection is how to train a good classifier for detecting anomalies, by fully using the large amount of available normal events and possibly few available abnormal events.

These years, many efforts have been expended on the abnormal event detection. Existing work on abnormal event detection can be roughly classified into two categories: motion-based approaches and trajectory-based approaches. The former ones usually extract low-level motion features, e.g. optical flow [1] and motion history image [2], from fixed spatial-temporal cubic, and then apply machine learning techniques for classification to discover possible abnormal events. The latter ones usually first extract the trajectory for moving objects in the scene, and then build temporal classifiers, e.g. HMM, HDP-HMM, for trajectory classification to find potential abnormal events. The limitation of the motion-based approaches is that simple features are usually adopted, which only reflect the relatively coarse motion information and may not be suitable for the detection in more complex scenes. Our proposed method belongs to the trajectory-based approaches. With the development of the object tracking, trajectory-based approaches are widely applied in traffic surveillance for detecting abnormal vehicle trajectory. They are also used in crowded scenes for identifying complex motion patterns.

Although many trajectory-based approaches achieve many successes in detecting abnormal events, most of them are usually developed for detecting global or large anomalies, and may ignore the local anomalies that exist in some subsegments of events or behaviors. To this end, we propose in this paper a novel framework for local anomaly detection based on TRAjectory Segmentation and Multi-Instance Learning, called TRASMIL, which consists of three phases: trajectory segmentation, trajectory representation and multi-instance learning (MIL).

In trajectory segmentation phase, we consider partitioning each trajectory into several sub-trajectories, to avoid ignoring local anomalies. The criterion for the segmentation usually includes two folds: for each trajectories, (i) every segmented sub-trajectories are encouraged to be independent from others with the goal to find simple motions from complex trajectory, and (ii) the number of sub-trajectories is encouraged to be small with avoiding over-segmentation. In this paper, four types of different segmentation algorithms are investigated, which includes: Maximum Acceleration (MA) [3], Minimum Description Length (MDL) [4, 5], Log-likelihood of Regression Function (LRF) [6], and Heterogeneous Curvature Distribution (HCD) [7]. To evaluate the quality of trajectory segmentation, we propose a metric, namely *QMeasure*, considering *Diversity* and *Granularity* of sub-trajectories, which are corresponding to the two folds of above mentioned criterion respectively. In our experiments, we found that *QMeasure* can help determine the optimal parameter of segmentation algorithms as well as choose appropriate segmentation algorithms, which will be validated in the experiments of Subsection 5.5.

In trajectory representation phase, the goal is to measure the similarity of any given two sub-trajectories. We first model any two sub-trajectories by sequential probabilistic models, and then calculate the distance by using the KL-divergency method [8] on the obtained two corresponding models. In our work, two sequential probabilistic models: Hidden Markov Model (HMM) and Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) are selected due to their effectiveness in modeling time-series data. The results of the HMM and HDP-HMM are listed and compared in experiments of Subsection 5.3.

In MIL detection phase, trajectories and segmented sub-trajectories are viewed as bags and instances in multi-instance learning, respectively. To solve the MIL problem, we use two conventional MIL methods, Diversity Density [9] and Citation kNN [10] in the phase. Their detection results are compared and analyzed in the experiment section. Finally, to validate the superiority of the proposed TRASMIL framework, all the combination methods in three phases are experimented and compared with two whole-trajectory-based methods.

The main contribution of the proposed TRASMIL is that it can successfully incorporates trajectory segmentation, modeling-based representation and multi-instance learning, which is able to effectively detect the trajectories with local anomalies that cannot be identified by existing methods. We combine this TRASMIL framework with 16 algorithms for different phases and test the performance. Substantial experiments show that the TRASMIL-based algorithms are effective in local anomaly detection, in which the combination of HDP-HMM with MDL segmentation and Citation kNN, MDL-C algorithm, performs the best.

1.1. Organization

The remainder of the paper is organized as follows. Section 2 gives a brief overview of related work. Problem statement and framework structure are described in Section 3. The working mechanism of TRASMIL and its combination with 16 algorithms are introduced in Section 4. Section 5 presents experimental results, which are compared to two whole-trajectory detection algorithms. Discussions are presented in Section 6, followed by conclusions in Section 7.

2. Related Work

Recent years, many works have been proposed for abnormal event detection. The related works can be roughly classified into two categories: motion-based approaches and trajectory-based approaches. Our proposed method belongs to the trajectory-based approaches, so we will pay more attention to the previous works about trajectory-based abnormal event detection.

For motion-based approaches, low-level features, e.g. optical flow, gradient, are first extracted from local spatial-temporal patches, then the extracted features of the normal events will be learned as normal model by using certain classifier, e.g. Gaussian Mixture Model [11, 12], Markov Random Field [13, 14] and Social Force Model [15]. After obtaining the normal model, for the

new coming frames, the abnormal events will be detected for the events, which are largely deviated from normal model by computing the likelihood. Although motion-based approaches can achieve satisfactory results in some datasets, most of them are only suitable for simple motion patterns since they usually model the variation of speed and direction at pixel/patch-level. The results for detecting complex abnormal events are undesirable.

For trajectory-based approaches, most of them rely on trajectories obtained from object detection and tracking. Much previous research works have been developed for trajectory-based approaches. The trajectory-based approaches mainly focus on trajectory representation and trajectory classification.

In terms of trajectory representation, related algorithms are listed in Table 1. As can be seen from the table, trajectory representation methods can be classified into three types: sequence of flow vectors, sequence of other features, and modeling-based representation. Firstly, flow vectors are usually represented as four-dimensional vectors (x, y, dx, dy) that contain spatial coordinates and velocities such as that claimed in literature [16]. Since abnormal behaviors are triggered by motion objects, Wang *et al* [17] add some features of objects, such as the size, into the flow vectors. Secondly, there are other sequences of features which differ from the above flow vectors. For example, semantic spatial primitives are proposed by Chan *et al* [18] to encode trajectories with binarized distance relations among objects. Pelekis *et al* [19] propose a local trajectory descriptor by computing local density and trajectory similarity to represent line segments. Lastly, modeling-based representation is becoming popular for representation due to its statistical description of trajectory distribution. Hidden Markov Model (HMM), a method of sequence modeling is adopted by Lester *et al* [20] for activity modeling. Since HDP-HMM can automatically adjust the state number of sequences, unlike HMM, which cannot, it is also widely applied for trajectory modeling by Hu *et al* [21] and Zhang *et al* [22].

Table 1: Related work of trajectory representation

Trajectory Representation	**Related Algorithms**
Sequence of flow vectors	Johnson *et al* [16], Hu *et al* [23], Wang *et al* [17] etc.
Sequence of other features	Semantic spatial primitives [18] Local trajectory descriptor [19] etc.
Modeling-based representation	HMM [20], HDP-HMM [21, 22] etc.

In terms of trajectory classification, the related work is usually based on two metrics, e.g., similarity-based and model-based metrics. The similarity-based methods [21, 24, 25] compute the similarity between trajectories in terms of Euclidean distance, Hausdorff distance or Dynamic Time Warping. Zhou *et al* [24] propose a supervised algorithm of trajectory edit-distance for learning trajectory similarity function. Those trajectories having a distance to normal ones larger than a given threshold are detected as abnormal. Fu *et al* [25] use hierarchical clustering to find dominant paths and lanes. Test trajectories are

predicted to a cluster where the spatial constraints and velocity constraints are checked to detect anomalies. Different from Fu's work, Hu *et al* [21] use Fuzzy Self-Organizing Neural Network to learn activity patterns. Then, the Euclidean distance between test trajectories and the best matching neuron is calculated. Model-based methods [16, 23, 26] model the observations of normal trajectories with no temporal order. Johnson *et al* develop a model of the probability density functions (pdfs) for object trajectories. Neural network based vector quantisation is used to learn the pdfs of flow vectors. Hu *et al* use HDP-HMM and One-class SVM to learn the model of normal trajectories. Anomalies can be detected via fitting the normal activity models. Different from Hu's work, sequential patterns are mined in feature selection phase by Lee *et al* [26], and then a classifier model is built for trajectory classification.

In our previous work, HDP-HMM and ensemble learning were used in [22] to generate a normal activity model for detecting anomalies. Abnormal activity models, which are derived from normal activity models, are applied to correct misclassified normal activities. This algorithm effectively detects emerging or unknown events and reduces the False Positive (FP) rate. Also, [27] creatively proposes a specific detection method based on multi-instance learning. After trajectories are segmented and modeled, the method finds anomalies according to their nearest neighbors. HDP-HMM demonstrates good performance experimentally; however, the solution of how to detect local anomalies has not been addressed generally or deeply in [27]. In this paper, a three-phase framework TRASMIL for local anomaly detection is proposed and 16 different algorithms are employed in the framework. We show that other algorithms can also be adopted in the framework for different applications. In addition, a novel metric with *Diversity* and *Granularity QMeasure* is proposed to compare different segmentation methods and measure the quality of trajectory segmentation. Substantial experiments are conducted and show the generalization and superiority of the framework. The MDL-C method, in particular, is the best combination for the highest accuracy and recall.

3. The Framework for Local Anomaly Detection

3.1. Problem Statement

A trajectory in videos may be quite long and complicated; however, anomalies usually occur in local subsegments. Therefore, it is understandable that local anomalies cannot be targeted by existing methods that are designed for the whole trajectory. In other words, previous detection methods are not effective when a trajectory has only local anomalies. Models and similarity measures designed for whole trajectories only focus on the global variations of movement and overlook the characteristics of local anomalies. As a result, the global trajectory based approaches are more suitable for cases in which huge differences exist between the normal and the abnormal activities. Taking Figure 1 as an example, there are four trajectories, of which only TR_3 trajectory is abnormal. The four trajectories are very similar except that the segment of TR_3 in the

rectangle differs from others. If modeling was conducted with the global trajectory, it is very likely that TR_3 would be mistakenly detected as normal, so the problem is how to effectively detect trajectories with such local anomalies.

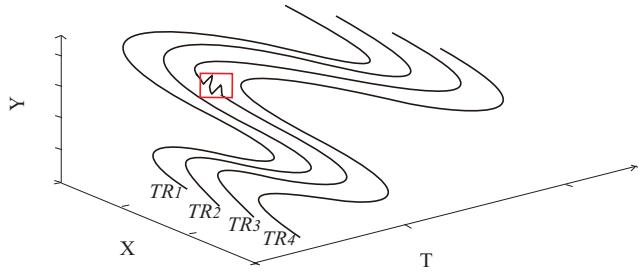


Figure 1: A short abnormal sub-trajectory in a long trajectory

To address the above problem, it is essential to segment whole trajectories into independent sub-trajectories (see Section 4). However, the detection of those trajectories which contain segmented sub-trajectories is different from that for the global trajectory. If a trajectory is abnormal, it has at least one abnormal sub-trajectory; otherwise, all sub-trajectories are normal. Likewise, in multi-instance learning, labeled bags consist of unlabeled instances. A positive bag has at least one positive instance; by contrary, all the instances in a negative bag are negative [28]. Based on the above analysis, local anomaly detection based on trajectory segmentation has a similar nature to multi-instance learning. Thus, multi-instance learning can be applied to local anomaly detection. Applied methods and their relationship are depicted in Figure 2, in which the whole trajectories are deemed as bags, and normal bags are negative, while abnormal ones are positive, and sub-trajectories are instances in the bags.

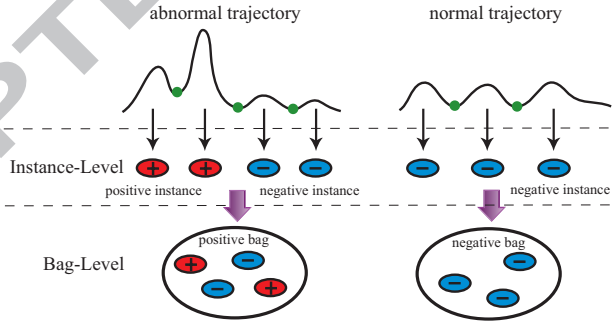


Figure 2: The relationship between multi-instance learning and local anomaly detection

3.2. The Structure of the Framework

In this paper, a novel detection framework is proposed to effectively detect local anomalies. Figure 3 shows the structure of the framework in three phases. The diagram (a) in Figure 3 shows that a trajectory is firstly partitioned into several independent sub-trajectories, such as $1.1, 1.2, 1.3$. Diagram (b) shows that each sub-trajectory is modeled via sequence learning models such as HDP-HMM. In (c), whole trajectories and sub-trajectories are viewed as bags and instances respectively. They are detected via multi-instance learning algorithms. The proposed TRASMIL framework is a general solution to detect local anomalies that are challenging to detect by traditional methods.

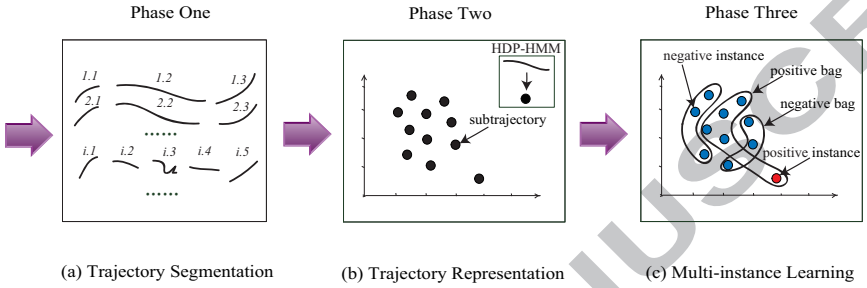


Figure 3: The working mechanism of the TRASMIL framework

With the proposed framework, many different algorithms can be fed into the three stages to form different combinations for detecting local anomalies. We create and test the following 16 combinations (as shown in Table 3).

1. Trajectory segmentation. There are algorithms that segment a trajectory into several independent sub-trajectories for image retrieval or trajectory clustering. We choose four of them: Maximum Acceleration (MA) [3], Minimum Description Length (MDL) [4, 5], Log-likelihood of Regression Function (LRF) [6], and Heterogeneous Curvature Distribution (HCD) [7]. Furthermore, a segmentation metric *QMeasure* (see Equation 1 in Section 4.1) is proposed to evaluate the four algorithms.
2. Trajectory representation. Representation of the sub-trajectories is critical for measuring the distance between them. In addition to be represented as sequences of feature vectors, sub-trajectories can be represented by sequence learning models such as HMM and HDP-DMM. The distance between two sequences can then be computed by the KL divergence [8].
3. Multi-instance learning. To learn a classifier for trajectories and sub-trajectories, we apply the classical multi-instance learning algorithms such as Diversity Density (DD) [9], Citation kNN (CkNN) [10], and their ensemble versions. Finally, the trajectories which are predicted to be positive are viewed as abnormal.

With the above combinations, the following three questions need to be addressed:

1. Is the three-phase framework effective for local anomaly detection?
2. Which combination performs the best?
3. How does the proposed approach compare to alternative approaches?

We discuss them in the sections below.

4. The Working Mechanism of the TRASMIL Framework

4.1. Trajectory Segmentation

Many trajectory segmentation algorithms [3–7, 29, 30] have been proposed for video indexing and retrieval, in which object motion retrieval in videos usually relies on object detection and tracking to extract the trajectories. In order to effectively extract local features of trajectories for retrieval, it is necessary to segment them into subsegments in terms of a certain evaluation metric. However, the same evaluation metrics cannot be used in local anomaly detection. Two additional aspects should be considered in evaluating segmentation methods. One is the independence between sub-trajectories, which means that movement patterns among different sub-trajectories can be distinguished from each other. In trajectory detection phase, these independent movement patterns are viewed as instances of bags in multi-instance learning. The other is the *Granularity* of segmentation, which measures the total segmented number of each trajectory for controlling over-segmentation, since too many small sub-trajectories segmented from a trajectory are harmful for classification with ignoring the full information of each sub-trajectory. The above two aspects need to be measured to evaluate the quality of trajectory segmentation.

Let TR_1 be a trajectory, which is often sampled by a sequence of discrete feature points $\overrightarrow{p_1}, \overrightarrow{p_2}, ..., \overrightarrow{p_{n_1}}$, thus in general $TR_1 =< \overrightarrow{p_1}, \overrightarrow{p_2}, ..., \overrightarrow{p_{n_1}} >$. Assume that TR_1 can be partitioned into N sub-trajectories, which constitute a set $SUBs = \{TR_{1.1}, TR_{1.2}...TR_{1.N}\}$, the set of segmentation points are also correspondingly labeled as $SEG = \{\overrightarrow{p_{s_1}}, \overrightarrow{p_{s_2}}, ..., \overrightarrow{p_{s_{N-1}}} | (1 \leq s_1 \leq s_2 \leq ... \leq s_{N-1} \leq n_i)\}$. Thus, different SEG sets reflect different kinds of segmentation. It is obvious that $TR_{1.1} =< \overrightarrow{p_1}, \overrightarrow{p_2}, ..., \overrightarrow{p_{s_1}} >$. Figure 4 shows the sub-trajectories and segmentation points associated with the trajectory TR_1.

Since it is difficult to depict and measure the independence between sub-trajectories, the *Diversity* among the sub-trajectories is alternatively measured in our work. We propose a metric *QMeasure* to evaluate the results for trajectory segmentation by considering both the *Diversity* and the *Granularity* of sub-trajectories, whose function is shown in Equation (1). The motivation of *QMeasure* comes from two aspects: 1) the *Diversity* (the first term in *QMeasure* function (1)) among different sub-trajectories is encouraged to be large since large *Diversity* among instances contributes to large variance, which is found to be useful when applying multi-instance learning in literature [31]. 2) the *Granularity* (the second term in *QMeasure* function (1)) is introduced to control the total numbers of segmented sub-trajectories, whose goal is to prevent the

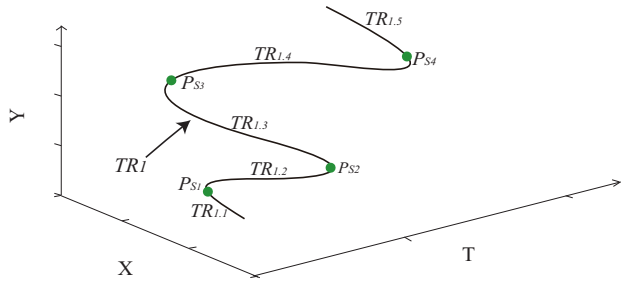


Figure 4: An example of sub-trajectories and segmentation points for trajectory TR_1

trajectory over-segmentation with too many small segmented sub-trajectories. The *QMeasure* function is computed as follows:

$$QMeasure = \frac{1}{N^2}\sum_{p=1}^{N}\sum_{q=1}^{N} Dist(TR_{i.p}, TR_{i.q}) + \frac{\lambda}{N}\sum_{k=1}^{N} Len(TR_{i.k}), \quad (1)$$

where p, q and k are the index of different sub-trajectories of trajectory TR_i, N is the number of the sub-trajectories. $Len(TR_{i.k})$ is the length of sub-trajectory $TR_{i.k}$. Parameter λ is the weight of *Granularity* and it is set as 0.0001 for the purpose of normalization, so the two terms in Eq. (1) will be comparable, which will be seen in Table 2. Generally, higher *QMeasure* means better segmentation which contributes to better classification result. *Diversity* is embodied through the distance between sub-trajectories $Dist(TR_{i.p}, TR_{i.q})$, which can be the distance of two multi-dimensional vectors, or the difference of their distribution with the KL divergence. *Granularity* is measured in terms of the average length $Len(TR_{i.k})$ of sub-trajectories, where the *Granularity* term is encouraged to be large. As can be seen in the equation, the summation of sub-trajectory lengths is approximate to the length of the trajectory. However, the summation is often less than or more than the trajectory length because of removed noise or overlapped segmentation points. So we use the summation of the sub-trajectory lengths to avoid over-segmentation.

In the trajectory segmentation phase, we select four types of different segmentation algorithms: Maximum Acceleration (MA) segmentation, Minimum Description Length (MDL), Segmentation with Log-likelihood of Regression Function (LRF), and Heterogeneous Curvature Distribution (HCD).

Maximum Acceleration (MA) segmentation is proposed by Chen *et al* [3]. After trajectory smoothing, a sequence of varied acceleration is computed. Then segmentation is performed at points of maximum acceleration at different scales. Trajectory smoothing and multi-scale segmentation make motion matching flexible and robust, and sub-trajectories have different motion patterns with respect to acceleration.

Minimum Description Length (MDL) is used by Lee *et al* [4, 5] to partition

a trajectory. Characteristic points where the motion changes rapidly are first detected. Then, from these points, the optimal segmentation points are found by minimizing the MDL cost. The optimization result improves the conciseness and preciseness of segmentation. MDL segmentation aims to find that the simple representation of motion and its conciseness is similar to the *Granularity* we measure.

Segmentation with Log-likelihood of Regression Function (LRF) is proposed by Zhou *et al* [6]. They model each trajectory as a regression function $f(T)$ with time variable T and compute the log-likelihood of each of them. Then segmentation points are sequentially determined at the position where the log-likelihood of a subsegment is smaller than the predefined threshold.

The Heterogeneous Curvature Distribution (HCD) method has been devised as follows by Bashir *et al* [7, 30] to segment trajectories. Two adjacent windows X and Y with equal width slide on a trajectory. Firstly, curvature data and its distribution of each window are calculated. Secondly, likelihood ratio test is adopted to determine if data of the windows X and Y are drawn from the same distribution. If the negative log-likelihood ratio, denoted as LD, is larger than the predefined threshold, then LD value and the point $\overrightarrow{P_s}$ partitioning two neighbor windows are added into a candidate list of segmentation points; otherwise, the two windows would be slid by constant length which is less than the sliding window size and the first and second steps are repeated. Finally, from the candidate list, distinct maxima are determined which are located as the local peak of LD sequence, and the corresponding points $\overrightarrow{P_s}$ are then treated as segmentation points of the trajectory. This algorithm partitions complex motion into individual curve segments of uniform motion.

To clearly understand the four segmentations, we take a trajectory for an example, which is segmented into many simple sub-trajectories by the four algorithms. Since the trajectory is long and complex so that it is not suitable for full display, a part of its sub-trajectories for the four segmentations are depicted respectively in Figure 5(1)-(4). As can be seen from the figure, the segmentations in Figure 5(2) and 5(3) are better than that in Figure 5(1) and 5(4) since the latter two are under-segmented and over-partitioned respectively. According to the *QMeasure* metric, Figure 5(2) is slightly better than Figure 5(3) since the *Diversity* among sub-trajectories via MDL is larger than that among sub-trajectories via LRF with the same number of segmentation points. To clearly understand the relationship of *QMeasure* values and segmentation results, we list the *QMeasure* values of the example trajectory shown in Figure 5 and their two terms, *Diversity* term and *Granularity*, in Table 2. In the table, λ is set as 10^{-4} that makes the two terms comparable. For the example trajectory, *QMeasure* values are different among four segmentations and the *QMeasure* value of MDL segmentation is the highest than others. In terms of the *Diversity* and *Granularity*, MDL has the highest value of *Diversity* while MA has the highest value of *Granularity*. The *Diversity* of LRF is the lowest because most segmented sub-trajectories are similar with each other, which can be seen in Figure 5(3). The lowest *Granularity* of HCD is due to the most seg-

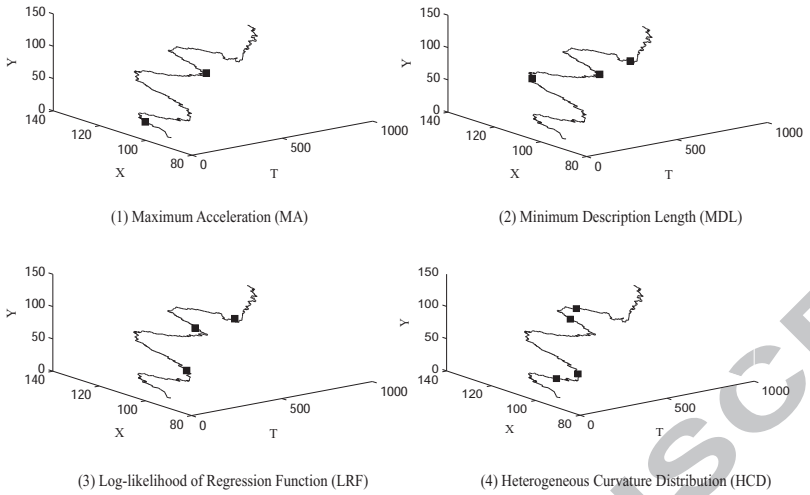


Figure 5: The results of a trajectory segmented by four methods

Table 2: *QMeasure* values of the example trajectory in Figure 5 where λ is set as 10^{-4}

Segmentation Algorithms	**Diversity**	**Granularity**	**QMeasure**
Maximum Acceleration	0.2074	0.2349	0.4423
Minimum Description Length	0.6279	0.1889	0.8178
Log-likelihood of Regression Function	0.1787	0.2285	0.4073
Heterogeneous Curvature Distribution	0.6052	0.1731	0.7783

mentation points in Figure 5(4). Thus, the values of Table 2 measure the two aspects *Diversity* and *Granularity* of trajectory segmentation in Figure 5. Moreover, Figure 5 and Table 2 indicate that both the *Diversity* and *Granularity* are necessary and important. From the figure, HCD segmentation of the example trajectory with more segments is harmful to preserve the motion information, which indicates that the *Granularity* is important. From the table, we can find that MA is with higher diversity and equivalent granularity compared with LRF, and the *QMeasure* results of MA are better than LRF, which indicates that the *Diversity* plays an important role in *QMeasure* as well as classification results. In the three-phase TRASMIL framework, trajectory segmentation phase is significant since it has a key influence on detection results, which will be shown in the experiments of Section 5. The proposed *QMeasure* is viewed as an evaluation metric of segmentation results, and helps select appropriate segmentation algorithms.

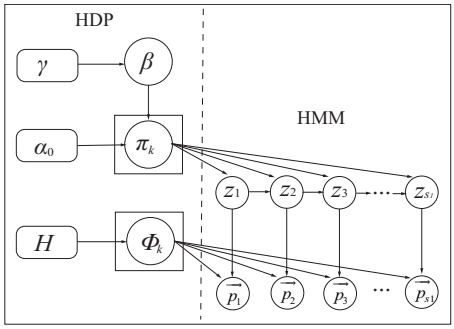


Figure 6: The graphical model of HDP-HMM [33]

4.2. Trajectory Representation using HDP-HMM

In regard to trajectory representation, modeling based representation is appropriate for describing sub-trajectories and measuring the similarity between them. A sub-trajectory can be depicted by a sequence of feature vectors by sampling n points from the original. However, it is inefficient for a complex motion because latent features are difficult to extract. Modeling-based representation can statistically describe the temporal distribution of motion trajectories. Thus, it is preferable to adopt modeling based representation methods, e.g., HMM [32] and HDP-HMM [33], in our framework. The similarity between sub-trajectories is measured via the KL divergence [8] between their models.

A sub-trajectory $TR_{1.1}$, modeled by HMM, is assumed to be a Markov process with hidden states. On the right side of Figure 6, feature points $\overrightarrow{p_1}, \overrightarrow{p_2}, ..., \overrightarrow{p_{s_1}}$ form an observation sequence, and $z_1, z_2, z_3, ..., z_{s_1}$ are a discrete hidden state sequence. State k varies according to state transition probability π_k. Every feature point only depends on the current state, which has a probability distribution over the feature points, denoted as $\overrightarrow{p_t} \sim F(\phi_{z_t})$. However, the downside of HMM is that the state number needs to be determined in advance and is unchangeable during the modeling. This means that HMM performs well only in cases where we have prior knowledge. If significant difference exists between the pre-assigned number of states and the ground truth, it is hard to achieve the real distribution of sub-trajectories, which is proved by experiments in Section 5.

Modeling-based representation by HDP-HMM can automatically determine the number of states, which is accomplished by HDP [34]. Specifically, arbitrary priors are placed on HMM model parameters π_k and ϕ_k.

$$\pi_k \sim DP(\alpha, \beta), \phi_k \sim H \tag{2}$$

where α, β are hyper parameters of the Dirichlet Process, H is the common base distribution. HDP is used to learn the ground truth parameters from data, which is depicted on the left side of Figure 6. The details of HDP are omitted in this paper due to limited space. The two parts in Figure 6 are merged to form the process of the HDP-HMM modeling of $TR_{1.1}$. We then adopt Beam

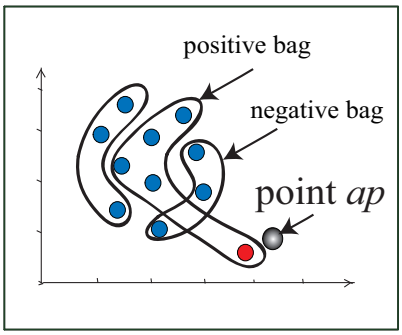


Figure 7: A virtual abnormal point *ap* in DD

sampling [35], which is a MCMC sampling algorithm. (It beats Gibbs sampling for fast convergence, robust performance and straightforward implementation)

4.3. Multi-Instance Learning

In the above two sections, we have discussed two phases of the proposed framework: segmenting trajectories and then modeling sub-trajectories. This section discusses the last phase: detecting abnormal sub-trajectories by multi-instance learning [28]. In multi-instance learning, a training set consists of labeled bags which are sets of instances with no labels. For binary classification, a bag is labeled as positive if it includes at least one positive instance. Otherwise, all the instances in the bag are negative. According to the above rule, a trajectory is viewed as a bag and its sub-trajectories form the instances of the bag. Abnormal TR_i^+ represents a positive bag and $TR_{i.p}^+$ is the p^{th} instance in that bag. Likewise, TR_i^- and $TR_{i.p}^-$ represent a negative bag and its p^{th} instance respectively. To detect anomalies, MIL algorithms are used to find positive bags. Two representative algorithms are adopted in this phase: Diverse Density (DD) [9] and Citation-kNN [10].

In trajectory detection, Diverse Density (DD) aims to find a representative abnormal point *ap* in the feature space of sub-trajectories to represent anomalies in the training data, as shown in Figure 7. It seems close to at least one subsegment from each abnormal trajectory and far away from all the normal trajectories. It can be computed by maximizing the diverse density over all the trajectories. The equation and its derivation are as follows:

$$\begin{aligned} dd(ap) &= \Pr(ap|TR_1^+,\ldots,TR_m^+,TR_1^-,\ldots,TR_n^-), \\ &= \prod_i \Pr(ap|TR_i^+)\prod_j \Pr(ap|TR_j^-). \end{aligned} \tag{3}$$

where $dd(ap)$ means the diverse density of abnormal point *ap*; the second line in Equation (3) assumes that bags are conditionally independent. Intuitively, if one of the subsegments in abnormal trajectory TR_i^+ is close to *ap*, then $\Pr(ap|TR_i^+)$ is high. After the point *ap* is learned, trajectory TR_j can be predicted to be

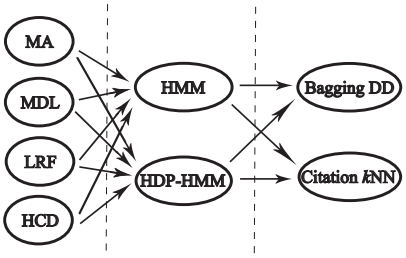


Figure 8: Combinations of algorithms based on the TRASMIL framework

abnormal or not according to the value of $\Pr(ap|TR_j)$. To improve performance, the Bagging DD algorithm [36], the ensemble version of DD, is employed in the experiments in Section 5.

In Citation-kNN [10], each trajectory is taken as a whole. Both its R-nearest neighbors and C-nearest neighbors are computed to predict anomalies. The R-nearest neighbors of trajectory TR_i refer to its nearest neighbors, and C-nearest neighbors are the trajectories that view TR_i as their nearest neighbors. The sum of normal trajectories in the two neighbor sets is counted, and denoted as RC_{normal}. Likewise, $RC_{abnormal}$ represents the sum of abnormal trajectories. If $RC_{abnormal} > RC_{normal}$, we will predict TR_i as being abnormal, otherwise normal. If R-nearest neighbors only (such as kNN) are measured, the majority voting scheme could easily be confused by the noise of false anomalies and the label is often mistakenly predicted. Thus, we use Citation kNN rather than kNN in the framework. In addition, the measure of R and C-nearest neighbors depends on the distances between trajectories. Here, the minimum Hausdorff distance [10] is used to evaluate the distance. It is defined as:

$$Dist(TR_i, TR_j) = \min_{TR_{i.p}} \min_{TR_{j.q}} Dist(TR_{i.p}, TR_{j.q}) \tag{4}$$

where $Dist(TR_j, TR_i) = Dist(TR_i, TR_j)$. The distance between sub-trajectories $Dist(TR_{i.p}, TR_{j.q})$ can be computed by the KL divergence [8].

4.4. Combination Methods

In this paper, we present a three-phase framework TRASMIL for local anomaly detection. In the first phase, four different segmentation methods are adopted to obtain a set of sub-trajectories for each trajectory. They are MA, MDL, LRF and HCD segmentation. In the second phase, HMM and HDP-HMM are then used for modeling sub-trajectories and a distance measure based on the KL divergence [8] is adopted to measure the distance between models in the second phase. In the last phase, two MIL methods, Bagging DD and Citation kNN are employed to detect trajectory bags with many sub-trajectories. During the implementation, 16 combination algorithms are applied in the framework for local anomaly detection. Details of combinations are depicted in Figure 8 and Table 3.

Table 3: The details of 16 different combinations

Index	Algorithms	Combination Details
1	MA-H-DD	MA + HMM + Bagging DD
2	MA-H-C	MA + HMM + Citation kNN
3	MDL-H-DD	MDL + HMM + Bagging DD
4	MDL-H-C	MDL + HMM + Citation kNN
5	LRF-H-DD	LRF + HMM + Bagging DD
6	LRF-H-C	LRF + HMM + Citation kNN
7	HCD-H-DD	HCD + HMM + Bagging DD
8	HCD-H-C	HCD + HMM + Citation kNN
9	MA-DD	MA + HDP-HMM + Bagging DD
10	MA-C	MA + HDP-HMM + Citation kNN
11	MDL-DD	MDL + HDP-HMM + Bagging DD
12	MDL-C	MDL + HDP-HMM + Citation kNN
13	LRF-DD	LRF + HDP-HMM + Bagging DD
14	LRF-C	LRF + HDP-HMM + Citation kNN
15	HCD-DD	HCD + HDP-HMM + Bagging DD
16	HCD-C	HCD + HDP-HMM + Citation kNN

As the experiments show, MDL-C performs the best among the 16 algorithms. Therefore, rather than introducing all algorithms, we only illustrate the MDL-C procedure as shown in Algorithm 1. Line $2-3$ describe the first phase. For every trajectory TR_i, the sub-trajectories set $SUBs$ is obtained via MDL segmentation, whose implementation is omitted due to limited space (please refer to [4]). In the second phase, Line $4-6$ describe HDP-HMM modeling based representation for all sub-trajectories $TR_{i.p}$ Then from Line 8 to Line 17, we compute the distance $Dist$ between HDP-HMM models and the distance $TRDist$ between trajectories. The rest part (from Line 18 to the end) describes the last trajectory detection phase via Citation kNN. Specifically, R-Nearest Neighbors RNN of testing data are firstly found at Line $18-20$ and then C-Nearest Neighbors CNN of testing data are determined. Lastly, anomalies are detected according to its RNN and CNN at Line $27-33$. As can be seen, the MDL-C method is implemented by stages in TRASMIL. The time cost of every phase is accumulated where HDP-HMM modeling phase is relatively time-consuming than others. Moreover, the results of every phase have an influence on the final detection. Thus, it is significant to select the appropriate method in the three phase of TRASMIL.

5. Experimental Results

We test our proposed approach on two publicly available datasets provided by the CAVIAR project [37]. The first dataset consists of video clips filmed with a wide angle camera lens in the entrance lobby of the INRIA Labs at

Algorithm 1 The procedure of the MDL-C algorithm

Input: Trajectory sets TRs, which consist of training set $TRtrain$ and testing set $TRtest$.

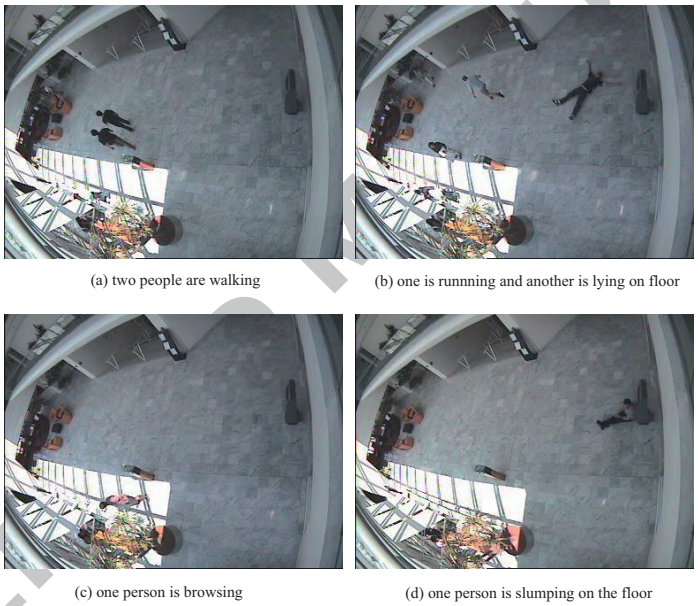
Output: Trajectories which are detected abnormal.

for $TR_i \in TRs$ **do**
  $SEG^* = \arg\min_{SEG} MDL(TR_i)$.
  $SUBs(TR_i) = \{TR_{i.1}, TR_{i.2}...\}$ is obtained by SEG^*.
  for $TR_{i.p} \in SUBs(TR_i)$ **do**
    Model θ_{ip} is built by HDP-HMM with Beam sampling for $TR_{i.p}$.
  end for
end for
for $TR_i \in TRtest$ **do**
  for $TR_j \in TRtrain$ **do**
    for $TR_{i.p} \in SUBs(TR_i)$ **do**
      for $TR_{j.q} \in SUBs(TR_j)$ **do**
        $Dist(TR_{i.p}, TR_{j.q}) = \frac{1}{2} D_{KLSYM}(\theta_{ip}||\theta_{jq})$.
      end for
    end for
    $TRDist(TR_i, TR_j) = \min_{TR_{i.p}} \min_{TR_{j.q}} Dist(TR_{i.p}, TR_{j.q})$.
  end for
end for
for $TR_i \in TRs$ **do**
  R-Nearest Neighbors $RNN(TR_i)$ are found according to $TRDist$.
end for
for $TR_i \in TRtest$ **do**
  for $TR_j \in TRtrain$ **do**
    if $TR_i \in RNN(TR_j)$ **then**
      TR_j is joined in C-Nearest Neighbors set $CNN(TR_i)$, $TR_j \in CNN(TR_i)$.
    end if
  end for
  The sums $RC_{nomal}, RC_{abnormal}$ are counted in the two nearest neighbor sets respectively.
  if $RC_{nomal} < RC_{abnormal}$ **then**
    TR_i is detected abnormal and outputted.
  else
    TR_i is predicted to be normal.
  end if
end for

Grenoble. The second also uses a wide angle lens along and across the hallway in a shopping center in Lisbon. The two datasets are labeled manually, and the corresponding xml files are provided. The trajectories of moving objects can be obtained from these files.

5.1. Experiment 1 on the First CAVIAR Dataset

In the first CAVIAR dataset, there are 30 videos and 265 sequences are extracted from the videos. Six basic scenarios are acted in the entrance lobby of the INRIA Labs, including "One person walking", "Browsing", "Resting, slumping or fainting", "Leaving bags behind", "People/groups meeting, walking together and splitting up" and "Two people fighting". These scenarios consist of four movements "walking", "inactive", "active" and "running". Figure 9 represents four movements in different scenarios. Obviously, Figure 9(a) and (b) depict "walking" and "running" respectively. In Figure 9(c), one person dressed in a red coat is browsing the reception desk, which belongs to "active" movement. In Figure 9(d), a man is slumping on the floor, which is "inactive" movement. We conduct several experiments on the dataset. Firstly,



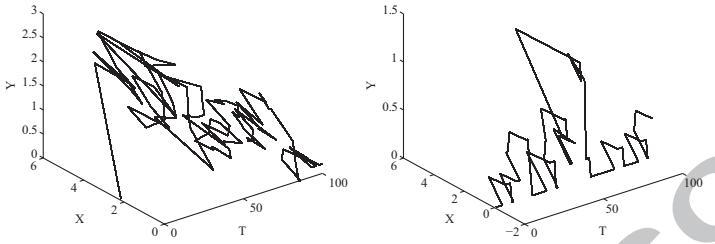
(a) two people are walking (b) one is runnning and another is lying on floor

(c) one person is browsing (d) one person is slumping on the floor

Figure 9: Different scenarios with four movements in the first CAVIAR dataset

two movements "walking" and "inactive" are selected as experimental data. There are 99 "walking" sequences and 22 "inactive" sequences whose lengths are more than 100. To clearly observe movement trajectories, we take examples of movements "walking" and "inactive" in Figure 10. In the figure, (a) and (b) depict "walking" trajectory and "inactive" trajectory with the first 100 points respectively, because the whole trajectories are fairly long and complex.

We define the "inactive" sequences as abnormal trajectories for local anomalies and perform the experiments with 10-fold cross validation. Then, the other two movements "active" and "running" are added into training and testing sets for multi-normal-one-abnormal classification and multi-normal-multi-abnormal classification in Subsection 5.4.



(a) "walking" trajectory (b) "inactive" trajectory

Figure 10: Two example trajectories of movements "walking" and "inactive" with the first 100 points respectively

In the model training phase with HDP-HMM, the training process is terminated when the state number remains unchanged for 20 iterations, which is selected experimentally to make all trajectories convergent. The learned state numbers of sub-trajectories are varied from 1 to 5 because of their different movement patterns. Note that relatively few states are learned since sub-trajectories mean simple motions, which are extracted from long trajectories. We take examples of a sub-trajectory with the state number of 2 in Figure 11. The figure shows the variations of the state number with the pre-assigned numbers 50, 20, 8 and 2 respectively. As can be seen in the figure, the four training processes are converged to the same state number of 2, which is always unchanged by the prior state number. It indicates that HDP-HMM modeling can automatically learn the ground truth of state number.

5.2. Experiment 2 on the Second CAVIAR Dataset

Another experiment is performed on the sequence data extracted from the second dataset. Each clip of the dataset is recorded from two different viewpoints - corridor and frontal. These two video sequences are synchronized frame by frame in terms of timeframe, which can be seen from Figure 12(a) and 12(b) respectively. Figure 12(c) depicts that the couple are standing there and browsing the store when one person is walking along the corridor. Figure 12(d) depicts that a man is going inside a store and his companion is waiting for him. We select the corridor viewpoint with three types of movements and choose two of three to carry out the experiments. There are 124 sequences in the first category and 21 in the second which are regarded as normal and abnormal respectively.

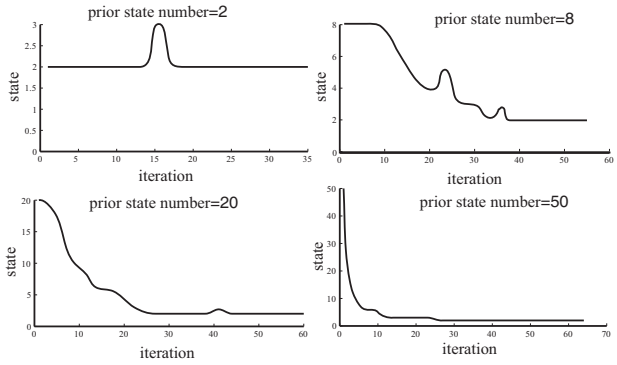


Figure 11: Variation of states with different pre-assigned state numbers during the sampling on the first dataset

(a) couple are walking along corridor and browsing in the corridor view

(b) couple are walking along corridor and browsing in the frontal view

(c) couple are standing there and browing the store

(d) a man is going inside a store, and his companion is waiting outside for him

Figure 12: Different scenarios with movements in the second CAVIAR dataset

Likewise, during the HDP-HMM modeling, we stop training when the state number remains unchanged for 30 iterations, which is also determined experimentally for the convergence of all trajectories . After unchanged 30 iterations, every sub-trajectory determines its convergent state number. In Figure 13, we take the learning process of a sub-trajectory with the state number of 2 for examples. The different prior state numbers 50, 20, 8 and 2 are also assigned in the learning process. From the figure, the state numbers in the four sub-figures are converged to 3, which likewise validates the advantage of HDP-HMM modeling over HMM.

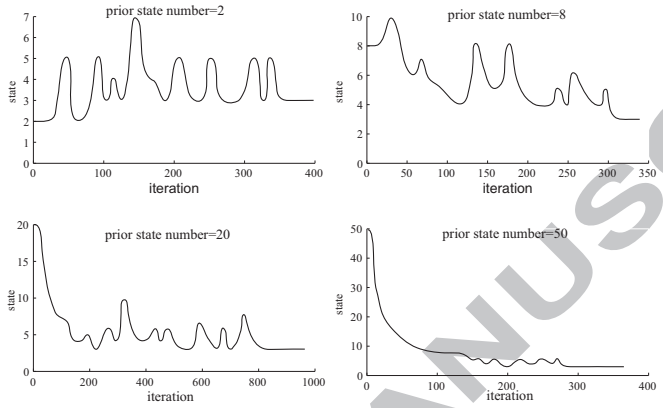


Figure 13: Variation of states with different prior state numbers during the sampling on the second dataset

5.3. Performance Evaluation

We first compare eight combinations (as shown in Table 4) with HDP-HMM trajectory representation and then make comparisons with HMM. We carry out 10 comparative experiments on the CAVIAR datasets, along with two traditional methods based on the whole trajectories.

In multi-instance learning, classification accuracy is usually used to evaluate the performance of different approaches.

$$Accuracy = \frac{TP + TN}{testNum} \tag{5}$$

where TP is the number of the correctly classified abnormal trajectories while TN is the number of normal trajectories, and $testNum$ is the number of trajectories in the testing dataset. In Table 4, we know that the accuracy of most approaches based on the proposed framework is higher than that of the whole trajectory-based methods, which indicates the effectiveness of the framework. In the last two contrast experiments, we use kNN and Regularized Least Squares (RLS) for abnormal detection. The whole trajectory-based modeling is likely to miss local or small anomalies, since a trajectory is long and complicated.

Though the accuracy of kNN and RLS in the first dataset is slightly higher than some of our approaches, other evaluation criteria such as recall and AUC in Table 5 show that these two algorithms actually perform worse than the other algorithms.

In the MIL detection phase, Bagging DD (the ensemble version of Diversity Density) and Citation kNN algorithms are adopted. The result shows the accuracy of Citation kNN is very high, close to 100%. The MDL-C algorithm shows the highest accuracy (98.97%) on the first dataset, and the second best is 96.88%. However, the accuracy of Bagging DD is usually less than that of Citation kNN in Table 4. The main reason is given below: the DD algorithm is used to find an abnormal point to represent all the anomalies and effectively distinguish from normal activities, but trajectories are usually so complex that we rarely find an abnormal point which can accurately reflect all the abnormal activities. Therefore, DD is not suitable for such applications of anomaly trace detection. With the HDP-HMM modeling-based representation and the corresponding similarity measure, Citation kNN is more accurate for measuring the distances between complex trajectories to determine its neighbors. It is based on the labels of R nearest neighbors and C nearest neighbors to predict classification. Since Citation kNN is more suitable for trajectory detection, we use it as the benchmark of MIL classification and conduct further analysis of the performance of the TRASMIL-based detection.

Table 4: The accuracy of different approaches in the two CAVIAR datasets

Algorithms	**Accuracy(%)**	
	Dataset1	**Dataset2**
MDL-DD	91.38	90.63
MDL-C	**98.97**	**96.88**
MA-DD	83.62	85.94
MA-C	96.55	90.63
LRF-DD	89.66	92.19
LRF-C	93.10	82.82
HCD-DD	91.38	92.19
HCD-C	89.96	87.50
Whole + HDP-HMM + RLS	89.96	75.00
Whole + HDP-HMM + kNN	90.63	87.50

In Table 5, we compare the precision, recall and AUC (Area Under ROC) of our approach with two traditional methods. This shows that the first four methods (based on our TRASMIL framework) beat the last two ones in terms of recall and AUC.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

where FN is the number of the misclassified abnormal trajectories. The recall value of 100% means a complete detection of true anomalies. According to the

Table 5: The Precision, Recall and AUC of four TRASMIL-based algorithms vs. two whole-trajectory methods (in Percent)

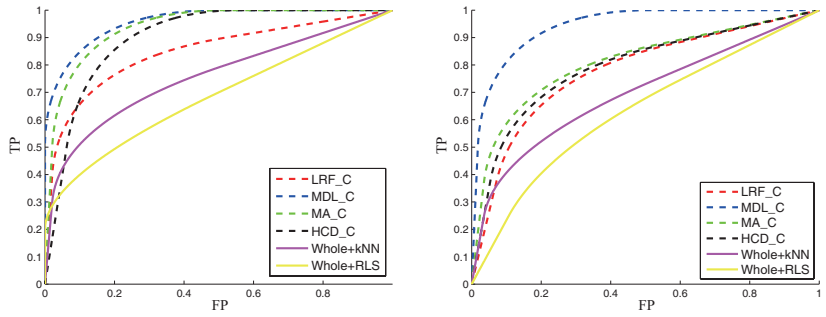
Algorithms	Dateset1			Dateset2		
	Precision	**Recall**	**AUC**	**Precision**	**Recall**	**AUC**
MDL-C	95.00	**100.00**	**99.38**	**90.00**	**87.50**	**92.86**
MA-C	83.33	**100.00**	97.92	63.34	62.50	78.57
LRF-C	80.00	80.00	87.92	38.10	62.50	74.11
HCD-C	62.50	**100.00**	93.75	50.00	62.50	76.79
Whole + HDP-HMM + RLS	**100.00**	40.00	70.00	25.00	50.00	64.29
Whole + HDP-HMM + kNN	75.00	60.00	77.92	50.00	37.50	66.07

performance of kNN and RLS in Tables 4 and 5, we know that they miss many abnormal trajectories and mistakenly predict them as normal. This indicates that the two whole trajectory-based methods kNN and RLS are not desirable. In other words, Table 5 shows that our methods miss fewer true anomalies than others, and are suitable for local anomaly detection. Likewise, AUC also shows the advantages of our approach.

As for datasets, the detection performance in the first dataset exceeds that in the second, because the abnormal samples in the first dataset differ greatly from the normal samples, while the difference in the second dataset is small, which creates a bigger challenge to the anomaly detection. However, the proposed framework applied in the two distinct datasets has better results than the others.

Tables 4 and 5 also show that different combination algorithms based on the framework result in different performance. Specifically, the MDL-C algorithm performs best since MDL conducts concise segmentation effectively for local anomaly detection, according to the proposed metric *QMeasure* Equation (1). In contrast, the LRF-C algorithm is the worst performer in our framework. From this we can see that the choice of segmentation algorithms is very important, and the quality of segmentation directly affects the detection performance. The results by different segmentations will be checked by the proposed QMeasure in Section 5.5. In addition, the ROC curve shown in Figure 14 reflects the relationship of true positive rate (TPR) and false positive rate (FPR) for these methods in Table 5. The approaches based on the proposed framework outperform the two methods based on whole trajectory.

For abnormal detection, there are some simple methods such as speed threshold based method, where the threshold is used to distinguish the abnormal speeds from the normal ones. Firstly, we compute the average speed for every trajectory. Then the threshold parameter is learned by grid search with step length 0.001 in the training set. The optimal value of threshold is selected when it achieves the highest F1-score in training set, where F1-score is a measure of



(a) Dataset1 (b) Dataset2

Figure 14: ROC curve in two datasets

classification results and whose computation is as follows:

$$F1\_score = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{7}$$

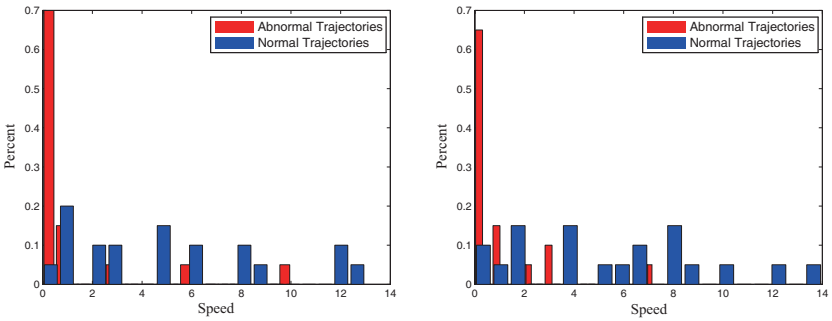
We have experiments on the two CAVIAR datasets. Table 6 shows their classification results. As we can see from the table, the results in the two datasets are not good. In Dataset 1, the recall of anomalies is very low, which means missing

Table 6: The performance of speed-threshold-based approach in the two CAVIAR datasets

Dataset	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)	AUC(%)
Dataset1	86.11	66.67	33.33	44.44	65.00
Dataset2	75.00	33.33	100.00	50.00	85.71

many anomalies. On the contrary, Dataset 2 has the poor precision, which indicates that the learned threshold is inclined to the normal speeds so many normal trajectories are misclassified. As can be seen, speed-threshold-based method is unstable and very sensitive to noises. To clearly understand the training data, we depict the distribution of speed values of trajectories in Figure 15. As can be seen from the feature, speed threshold is not efficient to distinguish them, since there are many overlapping values of the speed for normal and abnormal samples.

Lastly, we verify that HDP-HMM can automatically determine the appropriate number of states. We compare HDP-HMM with HMM using MDL segmentation and two MIL methods, Bagging DD and Citation kNN. Both HDP-HMM and HMM are based on our TRASMIL framework. The detection accuracy is depicted in Table 7. We use HMM with the state number of 4 for modeling sub-trajectories. As a result, HDP-HMM trajectory modeling has better performance than HMM, as shown in Table 7. This is because the HMM model can hardly reflect the real distribution of the data if the parameter is arbitrarily



(a) Dataset 1 (b) Dataset 2

Figure 15: The distribution of speed value of trajectories in Dataset 1 and Dataset 2

given without the prior knowledge. It indicates that modeling with an appropriate number of states can represent the sub-trajectory well, which is very important for similarity analysis of sub-trajectories and the follow-up anomaly detection.

Table 7: The comparison of the performance of two different modeling-based representation methods HDP-HMM and HMM with the state number 4

TRASMIL-based methods			**Accuracy(%)**	
Trajectory Segmentation	**Detecting Method**	**Sub-trajectory Modeling Methods**	**Dateset1**	**Dateset2**
MDL segmentation	Bagging DD	HDP-HMM	91.38	90.63
		HMM	89.96	81.25
	Citation kNN	HDP-HMM	98.97	96.88
		HMM	96.55	93.75

5.4. The Extension of the First CAVIAR Dataset

To improve the reliability of the TRASMIL framework, more types of movements in the first CAVIAR dataset have been added into the experiments. Firstly, we construct a multi-normal-one-abnormal scenario, where "inactive" sequences are defined as abnormal trajectories while "walking", "active" and "running" sequences are normal ones. Figure 16 depicts the accuracy of 16 different combination methods in the dataset. Most methods are more than 80% with respect to the accuracy. Specifically, HDP-HMM-based methods generally perform better than HMM-based method, which validates the advantage of HDP-HMM again. For the four different segmentations, we can see that MDL segmentation exceeds the others, especially in the 'HMM' group of Figure 16(b). In other groups of Figure 16(a) and (b), MDL is slightly better than the others. For the two MIL detection methods, Citation kNN is generally over than

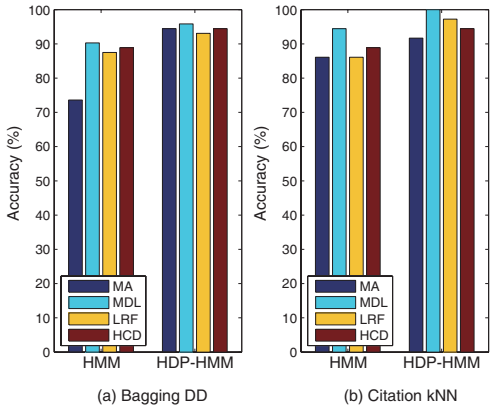


Figure 16: The accuracy of 16 different combination methods in the TRASMIL framework for the multi-normal-one-abnormal scenario: in (a) Bagging DD is used in the MIL detection phase, HMM and HDP-HMM in representation phase, MA, MDL, LRF and HCD in segmentation phase; (b) is similar with (a) except that Citation is used in the MIL detection phase.

Bagging DD. Among the 16 combination methods, MDL-C is the most excellent with one hundred percent of accuracy while MA-H-DD performs the worst with the lowest accuracy (73.61%). In the figure, it indicates that the proposed TRASMIL framework is applied to the multi-normal-one-abnormal scenario.

Secondly, we define movements "walking" and "active" as normal trajectories, "inactive" and "running" as abnormal ones, in order to construct a multi-normal-multi-abnormal scenario. Detection accuracy of the 16 methods is shown in Figure 17. Similar with Figure 16, the accuracy of most methods exceeds 80% expect for MA-H-DD and LRF-H-DD. From the figure, the methods used MDL and HCD segmentation are comparable and excellent, while the performance of LRF-based methods is changed largely between HMM and HDP-HMM. MDL-C method still performs the best in the multi-normal-multi-abnormal experiment for the highest accuracy (95.83%). From the figure, we can see that the TRASMIL framework is suitable for the multi-normal-multi-abnormal scenario.

5.5. QMeasure Evaluation on Trajectory Segmentation

In Section 4.1, we propose *QMeasure* metric for checking the quality of trajectory segmentation. Several experiments are executed on MA segmentation and MDL segmentation to reflect the relationship between *QMeasure* values and classification results. Specifically, we compute a series of *QMeasure* values of testing set as well as the corresponding classification results in MIL phase when the parameter values of segmentation methods vary in a range, which is shown in Figure 18. Note that the *QMeasure* values are computed via the summation of *QMeasure* value of every trajectory in testing set. For varied parameters,

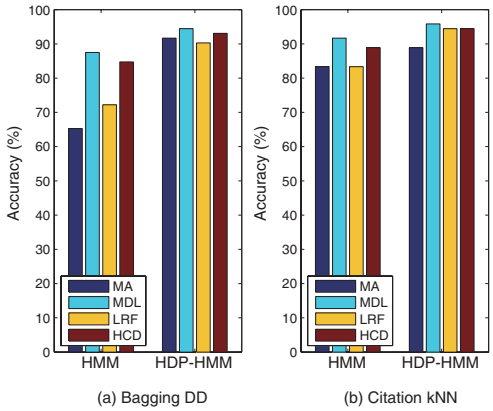


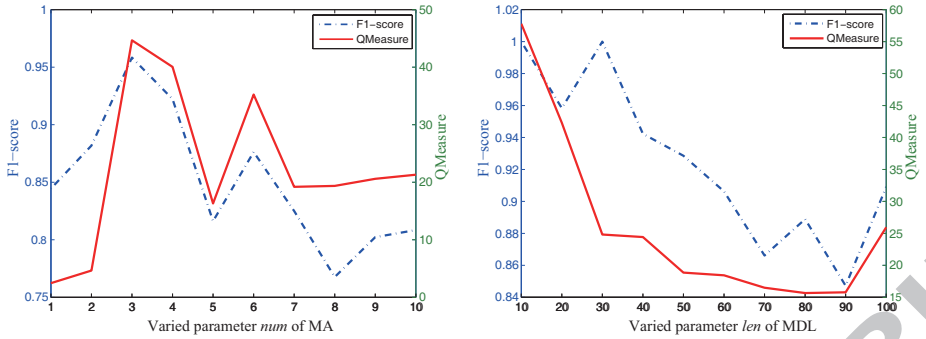
Figure 17: The accuracy of 16 different combination methods in the TRASMIL framework for the multi-normal-multi-abnormal scenario: in (a) Bagging DD is used in the MIL detection phase, HMM and HDP-HMM in representation phase, MA, MDL, LRF and HCD in segmentation phase; (b) is similar with (a) except that Citation is used in the MIL detection phase.

the parameter *num* of MA means the predefined number of subsegments while the parameter *len* of MDL is the step length of searching segmentation points. Figure 18(a) depicts the *QMeasure* and F1-score of MA-C detection when the subsegment number *num* of MA varies from 1 to 10. When *num* is equal to 3, both the *QMeasure* value and the F1-score value achieve the highest point. So the subsegment number *num* of 3 is the optimal parameter for MA segmentation. Likewise, Figure 18(b) depicts the *QMeasure* and F1-score of MDL-C detection when the step length *len* varies from 10 to 100. The step length *len* of 10 is selected as the experimental parameter when the corresponding *QMeasure* is the highest. From Figure 18(a) and (b), we can see that higher *QMeasure* value usually means better classification result. Moreover, when the parameters are the optimal respectively, MDL segmentation performs better than MA segmentation because of higher QMeasure (57.79). Also, for trajectory classification, the F1-score (100%) of MDL-C exceeds that (95.83%) of MA-C. Thus, it indicates that *QMeasure* can help determine the optimal parameter of segmentation algorithms as well as choose appropriate segmentation algorithms.

6. Discussion

The TRASMIL framework is a general solution for detecting local anomalies. The following three aspects can be used in applying the TRASMIL framework.

(1) As shown in the proposed combinations, there is no specific requirement on the combination of algorithms of trajectory segmentation, trajectory representation and multi-instance learning. This implies that many algorithms can be adopted in the framework.



(a) MA-C combination method

(b) MDL-C combination method

Figure 18: The contrast curve between F1-score values and the corresponding QMeasure values with the varied step length *len* of MDL segmentation from 10 to 100 in the first CAVIAR datasets

(2) The combination of three stage-oriented algorithms in TRASMIL is subject to application requirements.

(3) In particular, whether a segmentation method is suitable can be checked by the proposed metric *QMeasure*.

In the case that the running time is not demanding, HDP-HMM is a good choice because of its good performance, while in the other cases HMM does better since the modeling with HMM is less time-consuming than it is with HDP-HMM. In multi-instance learning, many alternative algorithms can be used. There are two assumptions proposed in [38]: one is that a key instance exists in a positive bag, such as DD and mi-SVM; the other is that there is no key instance and every instance contributes to the bag labeling such as MILES [39]. The definition of abnormal events and their differences from abnormal events can be analyzed; the assumption the detection task should take can then be determined. Subsequently, the TRASMIL framework can be used as a generic approach for local abnormal event detection.

The performance of local anomaly detection based on the TRASMIL framework suffers from the selection of specific methods in the three phases. Specifically, trajectory segmentation methods have a direct impact on it. For example, if trajectories are incorrectly partitioned and a local anomaly is divided into two smaller ones which locate at the endpoints of sub-trajectories, the abnormal trajectory tends to be missed. Thus, it is important to choose the right method. This paper is based on available trajectory data, but this is often not the case in real life data. This challenges trajectory-based anomaly detection. The recent use of wireless sensors in collecting activity trajectories may be helpful for this.

7. Conclusions

Most of the existing algorithms for abnormal event detection tend to miss local anomalies. To address this problem, a three-phase framework TRASMIL based on trajectory segmentation and multi-instance learning has been proposed for detecting local anomalies. The three phases consist of trajectory segmentation, trajectory representation and multi-instance learning. We have tested 16 combinations of different algorithms for the three stages to validate the effectiveness of the framework. The trajectories are first partitioned in many sub-trajectories with four different segmentation algorithms; a metric *QMeasure* is devised to measure the quality of segmentation. A sequence learning process is built for each sub-trajectory with HMM and HDP-HMM respectively. Finally, a MIL classifier with Bagging DD and Citation kNN is designed to predict the labels of the sub-trajectories. Substantial experimental results have shown that the TRASMIL-based combinations beat the traditional methods and achieve better accuracy and recall. Furthermore, the combination of MDL, HDP-HMM with Citation kNN (MDL-C) performs the best among all algorithms. We discuss that the fact that TRASMIL is generic and can be applied for local anomaly detection with consideration of the characteristics of target applications.

Acknowledgment

We would like to acknowledge the support for this work from the National Science Foundation of China (61035003,61175042, 61021062), the National 973 Program of China (2009CB320702), the 973 Program of Jiangsu, China (BK2011005, BE2010638), the Program for New Century Excellent Talents in University (NCET-10-0476), Australian Research Council Grants (DP1096218), and Linkage grant (LP100200774 and LP120100566). Also, we would like to acknowledge the CAVIAR data as coming from the EC Funded CAVIAR project/IST 2001 37540.

References

[1] R. Roberts, C. Potthast, F. Dellaert, Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies, in: Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 57–64.

[2] J. Davis, Hierarchical motion history images for recognizing human motion, in: Proc. IEEE Workshop on Detection and Recognition of Events in Video, IEEE, 2001, pp. 39–46.

[3] W. Chen, S. Chang, Motion trajectory matching of video objects, Storage and Retrieval for Media Databases 3972 (2000) 544–553.

[4] J. Lee, J. Han, K. Whang, Trajectory clustering: a partition-and-group framework, in: Proc. Int. Conf. Management of Data (SIGMOD), ACM, 2007, pp. 593–604.

[5] J. Lee, J. Han, X. Li, Trajectory outlier detection: a partition-and-detect framework, in: Proc. Int. Conf. Data Engineering (ICDE), IEEE, 2008, pp. 140–149.

[6] Y. Zhou, T. Huang, Bag of segments for motion trajectory analysis, in: Proc. Int. Conf. Image Processing (ICIP), IEEE, 2008, pp. 757–760.

[7] F. Bashir, A. Khokhar, D. Schonfeld, A hybrid system for affine-invariant trajectory retrieval, in: Proc. ACM SIGMM Int. Workshop on Multimedia Information Retrieval, ACM, 2004, pp. 235–242.

[8] D. García-García, E. Hernández, F. Diaz de Maria, A new distance measure for model-based sequence clustering, IEEE Trans. Pattern Analysis and Machine Intelligence 31 (2009) 1325–1331.

[9] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, Advances in Neural Information Processing Systems (1998) 570–576.

[10] J. Wang, J. Zucker, Solving multiple-instance problem: A lazy learning approach, in: Proc. Int. Conf. Machine Learning (ICML), Morgan Kaufmann, 2000, pp. 1119–1125.

[11] T. Xiang, S. Gong, Incremental and adaptive abnormal behaviour detection, Computer Vision and Image Understanding 111 (2008) 59–73.

[12] Y. Shi, Y. Gao, R. Wang, Real-time abnormal event detection in complicated scenes, in: Proc. Int. Conf. Pattern Recognition (ICPR), IEEE, 2010, pp. 3653–3656.

[13] Y. Benezeth, P. Jodoin, V. Saligrama, C. Rosenberger, Abnormal events detection based on spatio-temporal co-occurences, in: Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 2458–2465.

[14] J. Kim, K. Grauman, Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates, in: Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 2921–2928.

[15] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 935–942.

[16] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Image and Vision Computing 14 (1996) 609–615.

[17] X. Wang, K. Tieu, E. Grimson, Learning semantic scene models by trajectory analysis, Computer Vision–ECCV 2006 3953 (2006) 110–123.

[18] M. Chan, A. Hoogs, J. Schmiederer, M. Petersen, Detecting rare events in video using semantic primitives with HMM, in: Proc. Int. Conf. Pattern Recognition (ICPR), volume 4, IEEE, 2004, pp. 150–154.

[19] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, Y. Theodoridis, Segmentation and sampling of moving object trajectories based on representativeness, IEEE Trans. Knowledge and Data Engineering 24 (2012) 1328–1343.

[20] J. Lester, T. Choudhury, N. Kern, G. Borriello, B. Hannaford, A hybrid discriminative/generative approach for modeling human activities, in: Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), Morgan Kaufmann, 2005, pp. 766–772.

[21] W. Hu, D. Xie, T. Tan, S. Maybank, Learning activity patterns using fuzzy self-organizing neural network, IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics 34 (2004) 1618–1626.

[22] X. Zhang, H. Liu, Y. Gao, D. Hu, Detecting abnormal events via hierarchical Dirichlet processes, Advances in Knowledge Discovery and Data Mining 5476 (2009) 278–289.

[23] D. Hu, X. Zhang, J. Yin, V. Zheng, Q. Yang, Abnormal activity recognition based on HDP-HMM models, in: Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), Morgan Kaufmann, 2009, pp. 1715–1720.

[24] Y. Zhou, S. Yan, T. Huang, Detecting anomaly in videos from trajectory similarity analysis, in: Proc. Int. Conf. Multimedia and Expo, IEEE, 2007, pp. 1087–1090.

[25] Z. Fu, W. Hu, T. Tan, Similarity based vehicle trajectory clustering and anomaly detection, in: Proc. Int. Conf. Image Processing (ICIP), volume 2, IEEE, 2005, pp. 602–605.

[26] J. Lee, J. Han, X. Li, H. Cheng, Mining discriminative patterns for classifying trajectories on road networks, IEEE Trans. Knowledge and Data Engineering 23 (2010) 713–726.

[27] Y. Cui, Y. Gao, Study on abnormal event detection based on the multi-instance learning, Pattern Recognition and Artificial Intelligence 6 (2010) 4–78.

[28] T. Dietterich, R. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1997) 31–71.

[29] R. Mann, A. Jepson, T. El-Maraghi, Trajectory segmentation using dynamic programming, in: Proc. Int. Conf. Pattern Recognition (ICPR), volume 1, IEEE, 2002, pp. 331–334.

[30] F. Bashir, A. Khokhar, D. Schonfeld, Segmented trajectory based indexing and retrieval of video data, Proc. Int. Conf. Image Processing (ICIP) 2 (2003) 623.

[31] Y. Han, Q. Tao, J. Wang, Avoiding false positive in multi-instance learning, Advances in Neural Information Processing Systems (2010) 811–819.

[32] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. of the IEEE 77 (1989) 257–286.

[33] E. Fox, E. Sudderth, M. Jordan, A. Willsky, An HDP-HMM for systems with state persistence, in: Proc. Int. Conf. Machine Learning (ICML), ACM, 2008, pp. 312–319.

[34] Y. Teh, M. Jordan, M. Beal, D. Blei, Hierarchical Dirichlet processes, J. American Statistical Association 101 (2006) 1566–1581.

[35] J. Van Gael, Y. Saatci, Y. Teh, Z. Ghahramani, Beam sampling for the infinite hidden Markov model, Proc. Int. conf. on Machine learning (ICML) (2008) 1088–1095.

[36] Z. Zhou, M. Zhang, Ensembles of multi-instance learners, Machine Learning: ECML 2003 2837 (2003) 492–502.

[37] CAVIAR, http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[38] Z. Zhou, M. Zhang, S. Huang, Y. Li, Multi-instance multi-label learning, Artificial Intelligence 176 (2011) 2291–2320.

[39] Y. Chen, J. Bi, J. Wang, Miles: Multiple-instance learning via embedded instance selection, IEEE Trans. Pattern Analysis and Machine Intelligence 28 (2006) 1931–1947.