



Queensland University of Technology
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Xu, Jingxin, Denman, Simon, Fookes, Clinton B., & Sridharan, Sridha (2012) Activity analysis in complicated scenes using DFT coefficients of particle trajectories. In *9th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2012)*, 18-21 September 2012, Beijing, China. (In Press)

This file was downloaded from: <http://eprints.qut.edu.au/51041/>

© Copyright 2012 Please consult the authors.

Notice: *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

Activity Analysis in Complicated Scenes Using DFT Coefficients of Particle Trajectories

Jingxin Xu, Simon Denman, Sridha Sridharan, Clinton Fookes
Image and Video Laboratory, Queensland University of Technology
GPO Box 2434, Brisbane 4001, Australia
{j15.xu, s.denman, c.fookes, s.sridharan}@qut.edu.au

Abstract

Modelling activities in crowded scenes is very challenging as object tracking is not robust in complicated scenes and optical flow does not capture long range motion. We propose a novel approach to analyse activities in crowded scenes using a “bag of particle trajectories”. Particle trajectories are extracted from foreground regions within short video clips using particle video, which estimates long range motion in contrast to optical flow which is only concerned with inter-frame motion. Our applications include temporal video segmentation and anomaly detection, and we perform our evaluation on several real-world datasets containing complicated scenes. We show that our approaches achieve state-of-the-art performance for both tasks.

1. Introduction

The large number of cameras present in public environments (i.e. airports, shopping centres) means that automatic solutions are required to adequately monitor all incoming feeds. Recently, there has been an increasing interest in modelling activities in crowded scenes, and detecting abnormal events [1, 3, 7, 18, 21]. Due to the number of events that are possible in a surveillance scene, it is rarely practical to list all events beforehand or label the abnormal events in the training dataset. Many algorithms [1, 3, 7, 9, 19, 21] apply unsupervised learning approaches and assume that unusual events are those that have low likelihoods within a training dataset. The definition of the unusual events, however, is closely linked to the features used by the algorithms.

Due to the challenges of tracking objects in a crowded scene, many algorithms [7, 10, 11, 19, 17] extract local features that are not directly associated with individual objects, such as a Mixture of Dynamic Textures (MDTs) [10], motion vectors [17] or optical flow feature representations [7, 19, 1].

The features outlined above can be used with a variety of learning models to detect abnormalities, such as Gaus-

sian Mixture Models (GMM) [13], Hidden Markov Models [3], Probabilistic Topic Models [9, 7, 19, 24], Sparse Coding [22, 23] and graph based non-linear subspace models [17]. In [9], Li et al. proposed to represent events in a 10 dimensional feature vector, including information on location, shape and motion. A GMM is trained to cluster the events into discrete codewords. Cascade topic models are proposed to model both regional and global context. Applications such as unusual event detection and temporal video segmentation are shown using this technique. However, the motion features are derived from optical flow, which only captures motion for successive frames, and is inaccurate in textureless regions.

We propose a novel approach to model activities in complicated scenes such as those in [9], by analysing trajectories over short video clips (i.e. 10 seconds). Within each clip, we first perform a background subtraction to locate moving objects [8]. Particle trajectories for foreground regions are extracted using the particle video algorithm [14]. A Discrete Fourier Transform (DFT) is applied to the trajectory, taking the first few coefficients of the transformed trajectory as our feature vector. K-means is used to quantize the features and build a codebook. We demonstrate the proposed feature for anomaly detection and temporal video segmentation using Latent Dirichlet Allocation (LDA) and a combination of LDA and compressed sensing (CS).

This Section introduces our algorithm in detail. From Section 2.1 to Section 2.4, we present the particle trajectory construction, trajectory clustering using DFT coefficients, anomaly detection and temporal video segmentation respectively.

1.1. Particle Trajectory Construction

To construct the particle trajectories, we use the particle video algorithm proposed in [14]. Particle video is a technique for motion estimation using long range point trajectories. A preprocessing step for particle video is optical flow computation using [5]. There are five steps in parti-

cle video: propagation, linking, optimisation, pruning and addition.

The propagation step is almost equivalent to the ‘‘particle advection’’ step in [11, 21]. Let $u(x, y, t - 1)$ and $v(x, y, t - 1)$ be the horizontal and vertical pixel (x, y) velocities respectively at frame $t - 1$. Particle i is propagated to frame t using,

$$x_i(t) = x_i(t - 1) + u_i(x_i(t - 1), y_i(t - 1), t - 1), \quad (1)$$

$$y_i(t) = y_i(t - 1) + v_i(x_i(t - 1), y_i(t - 1), t - 1). \quad (2)$$

Particles located on the same surface will have similar trajectories. The linking algorithm merges similar particle trajectories in a neighbourhood into a single trajectory. Then, a particle optimisation algorithm is used to determine the final location where appearance features such as colour and gradient are also considered. This step can also be used to remove the non-rigid decomposition of the movements. Particles associated with stationary or occluded objects are removed through particle pruning. The algorithm also adds particles in the gaps between existing particles to allow for new regions of motion to be monitored.

However, in any given scene there are a large number of particles located in the background regions, which should be removed. In our application, we first perform a background subtraction using online GMMs [8].

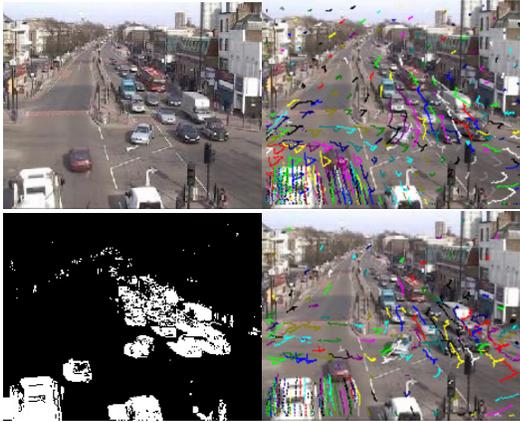


Figure 1. Particle Trajectories in Foreground Regions. Top Left: original image; Top Right: particle trajectories without background subtraction; Bottom Left: foreground image; Bottom Right: particle trajectories with background subtraction.

Figure 1 shows the effect of background subtraction. It can be seen that by considering only trajectories that lie in foreground regions, a large amount of noise is removed. However, since the background subtraction algorithm does detect some false motion, there are still some trajectories in the background that need to be removed. Given a trajectory Z , let x_{min} and x_{max} be the minimum and maximum of the horizontal location; y_{min} and y_{max} be the minimum and maximum of the vertical location; and T be the duration of the trajectory (i.e. the number of frames it is tracked

over). We remove trajectories that satisfy

$$(T < \varepsilon) \cup (|x_{max} - x_{min}| + |y_{max} - y_{min}| < \sigma), \quad (3)$$

where ε and σ are thresholds.

1.2. Trajectory Clustering using DFT Coefficients

Following the trajectory construction, the trajectories are encoded to create a codebook. Here, we make use of the Discrete Fourier Transform (DFT). The DFT has long been used for information retrieval of time series data in sequence databases [2]. Given that the trajectories can be viewed as two dimensional time series data, this technique has been used in trajectory clustering [12]. We pad each sequence with 0s to fix the length to n points. A trajectory, Z , is described by a sequence of locations in time. We separate the sequence into a horizontal series and a vertical series, which are denoted as $X = [x_1, x_2, \dots, x_N]$ and $Y = [y_1, y_2, \dots, y_N]$. We take the Fourier Transform separately on the two signals,

$$X_f = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} x_i \exp\left(\frac{-j2\pi ft}{n}\right), \quad f = 0, 1, \dots, n - 1 \quad (4)$$

$$Y_f = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} y_i \exp\left(\frac{-j2\pi ft}{n}\right). \quad f = 0, 1, \dots, n - 1 \quad (5)$$

The DC components, X_0 and Y_0 , are removed (this differs from [12]), and we select the first M frequency components. Since X_f and Y_f are complex values, we can represent them in two vectors with a size of $2M$ (M real and M imaginary coefficients). We concatenate the Fourier coefficients for X and Y , creating a feature vector of length $4M$. K-means is used to cluster the features and build the codebook.

We argue that this approach has the following benefits: 1) Projecting data into an orthogonal basis (i.e the Fourier Transform basis) can achieve decorrelation of the data, benefiting the K-means algorithm where the Euclidean distance is used. 2) Using the first M low frequency components is equivalent to dimension reduction; as the trajectories have variable lengths, it is not easy to perform dimension reduction in the time domain; 3) Because the DC components are sensitive to the location and the temporal duration of the trajectories; by removing them we can better capture motion characteristics.

Figure 2 demonstrates that the first 11 frequency components can be used to distinguish between the horizontal and vertical trajectories. Furthermore, it can be seen that by removing the DC component, the two trajectories are more easily distinguishable.

1.3. Anomaly Detection

This section will introduce our application for anomaly detection. The first step is dimension reduction using Latent

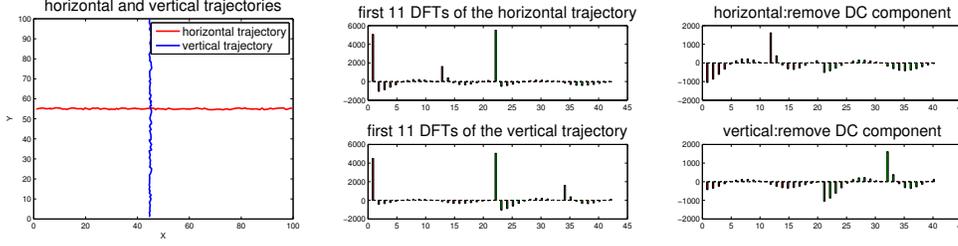


Figure 2. DFT coefficients of trajectory. Left: horizontal and vertical trajectories. Middle: the first 11 DFT coefficients (including DC). Right: DFT coefficients after removing DC component. Removing the DC component emphasises the difference in the trajectories.

Dirichlet Allocation [4]. Each trajectory is quantised into a “word” using K-means as discussed in Section 2.2. A short video clip is represented by a histogram of words (“document”). The long video sequence from which the “documents” are taken, is the “corpus”. A “topic” is a multinomial distribution of words, which are likely to occur within a short video clip. The number of topics, K , is set manually. The distribution of topics, θ , draws from a Dirichlet distribution which is controlled by the Dirichlet parameter, α . In order to be consistent with [4], we denote w as the words, and z as the labels of words to the K topics. Let the word probabilities for the topics be denoted by β , which is a $k \times V$ matrix where V is the size of the vocabulary and $\beta_{ij} = p(w^j | z^i)$. Parameters such as α and β are trained by a variational inference EM algorithm [4], with the variational parameters Φ and γ , where γ is the posterior Dirichlet parameter with the same size as θ , and Φ is the multinomial parameter with the same size as β . They have the following relationship:

$$\Phi_{ni} \propto \beta_{iw_n} \exp\{E_q[\log(\theta_i) | \gamma]\}; \quad \gamma_i = \alpha_i + \sum_{n=1}^N \Phi_{ni}. \quad (6)$$

The document likelihood is, however, sensitive to the total number of words in the document. Due to the variational EM algorithm [4], documents with more words tends to produce a lower likelihood. To solve this problem, we seek to represent the documents in the topic simplex. In the topic simplex, a document can be represented by either θ or γ . In this way, LDA is used for dimension reduction [4] by representing the document as γ .

Then the next step is to detect outlier patterns in terms of the training data. Since in many cases the data points may not lie at a single low-dimensional subspace, it is more robust to project the data into a higher dimensional space, termed a union of low dimensional manifolds [15]. For this reason, we use compressed sensing to detect outliers. Following [20] from face recognition, given M documents, we construct the basis set (dictionary) as

$$B = [\gamma_1, \gamma_2, \dots, \gamma_M].$$

An input, y , is considered to be a sparse linear superposition

over the basis set, such that $y = Bx$, where x is the sparse coefficient. In the detection process, given a new observation, y' , we need to compute the sparse coefficients, x' , that can be used to best represent y' . Since the training dataset only contains normal items, an abnormal observation will cause a high reconstruction error, $\|y' - Bx'\|_2$, based on which anomalies are detected. In our application, we use the Dantzig Selector [6] to compute the sparse coefficients,

$$\min \|x\|_{l_1} \quad \text{s.t.} \quad \|B^T(y - Bx)\|_{l_\infty} < \xi, \quad (7)$$

where ξ is a positive scalar.

Alternatively, we can detect the anomalies using LDA only by identifying the low likelihood documents, or use compressive sensing directly without dimension reduction. In the latter approach, the Dantzig selector will become a least squares problem with a sparsity constraint. In Section 3, we will compare these three approaches on several datasets.

1.4. Temporal Video Segmentation

The task of temporal video segmentation is to cluster the video clips into groups that describe the dominant modes within the scene. Like Section 2.3, we use LDA as the learning model. We compute the topic distribution by first marginalising Φ over all documents in the corpus, followed by normalization. We have two approaches for this task. In the first approach, we define a topic as a dominant mode. In this case, the number of topics is set to the number of dominant modes. We find the maximum value of θ for each video clip, and cluster the clip into that group.

The second approach is to use the topic distribution as the feature. K-means is subsequently used to cluster the documents. However, one topic may have a much higher weight than others in all video clips. Thus we first perform normalization over the corpus to ensure a more even distribution of topic weights. Let θ_{id} be the probability of topic i in document d , $\theta_{i,max}$ and $\theta_{i,min}$ be the maximum and minimum values for the probabilities of topic i over all documents in the corpus respectively. We set

$$\theta_{id} = \frac{\theta_{id} - \theta_{i,min}}{\theta_{i,max} - \theta_{i,min}}.$$



Figure 3. Sample images from the dataset for anomaly detection. Left image: normal case. Right image: Abnormal case

2. Experiments

We evaluate our proposed system for the tasks of anomaly detection and temporal video segmentation. The datasets used in our evaluation are presented in Section 3.1. Section 3.2 shows the results for anomaly detection and Section 3.3 shows the results for temporal video segmentation.

2.1. Database Specification

Our experiments are conducted on two datasets, the QMUL traffic datasets¹ and a real world surveillance dataset on a university campus (the Campus dataset)². Figure 3 shows a normal and an abnormal scene for the Campus dataset. Both the training and test data are one hour long video clips, and contain a mixture of crowd densities. The normal activities include pedestrians entering or exiting the building, entering or exiting a lecture theatre (yellow door), and going to the counter at the bottom right of the image. The abnormal events are caused by heavy rain outside, and include people running in from the rain, and people walking towards the door to exit and turning back. We divide the video sequence into clips of 10 seconds duration. Thus both the training and test dataset contains 360 video clips. We manually annotate the video clips during the heavy rain as abnormal video clips. In total, there are 77 abnormal video clips. For the QMUL dataset, we use the same groundtruth as [9], which is also annotated at the clip level. This dataset contains two environments: the Junction dataset and the Roundabout dataset. Each dataset contains about one hour of real world traffic surveillance footage. The abnormal events defined in this dataset are dangerous driving, traffic rule violations, interrupting the traffic flow, and rare maneuvers such as U-turns. The Junction Dataset contains 73 video clips for training and 39 video clips for testing. The Roundabout dataset contains 144 video clips for training and 59 video clips for testing. The QMUL dataset is also used to evaluate the temporal segmentation. In terms of [9], there are two temporal phases: vertical and horizontal flow (See Figure 4).

¹<http://www.eecs.qmul.ac.uk/~jianli/Junction.html>
<http://www.eecs.qmul.ac.uk/~jianli/Roundabout.html>

²The dataset is available upon request from the authors. A video showing an example of the dataset is provided as supplementary material.



Figure 4. QMUL Traffic Dataset. Top: Junction dataset. Bottom: Roundabout Dataset. Left: Horizontal flow. Middle: Vertical flow. Right: Abnormal events. In the Junction dataset, a fire engine interrupted the normal traffic flow; in the roundabout dataset, a white car break the traffic light.

2.2. Anomaly Detection Evaluation

To extract and encode the trajectories (see Sections 2.1 and 2.2), we perform a 300 point DFT and exact the first 10 low frequency components, disregarding the DC component. K-Means is used to build a codebook of size 500. Given the same feature, we use three approaches to detect anomalies: 1) using LDA only; 2) using CS only; and 3) the combination of LDA and CS outlined in Section 2.3. The number of topics is set to 30 for the Campus dataset, 10 for the Junction dataset, and 48 for the Roundabout dataset.

Figure 5 shows the ROC curves for all datasets. We compare our algorithm with [9] in Table 1. Results demonstrated in Figure 5 and Table 1 are achieved when the parameters are well initialized³. In the Campus dataset, all three approaches have AUCs higher than 0.9, demonstrating the suitability of the proposed feature across a variety of classifiers. In the Junction dataset, using the proposed feature and LDA, we achieve an AUC as 0.5887, which significantly outperforms the equivalent system in [9] (0.3871). Using LDA or CS only with our feature, we are unable to match the performance of Cas-LDA or Cas-pLDA [9]. However, the proposed combination of LDA and CS introduced in Section 2.3, is able to slightly outperform Cas-pLSA and match the performance of Cas-LDA (see Table 1). The number of topics, which is typically is set using Hierarchical Dirichlet Processes (HDP) [16], is selected through experimentation. This is because, for the proposed combination of LDA and CS, the number of topics that fit the LDA does not fit the LDA-CS combination. For example, in the Junction dataset HDP learns the number of topics as 3. The plot in Figure 6 shows that despite this being optimal for LDA, it is far from optimal for the proposed LDA+CS approach. The performance is also highly related to the number of codewords (the number of clusters in the K-means). We observed that significant improvement can

³We randomly initialise the LDA parameters, run experiments ten times, and select the best results.

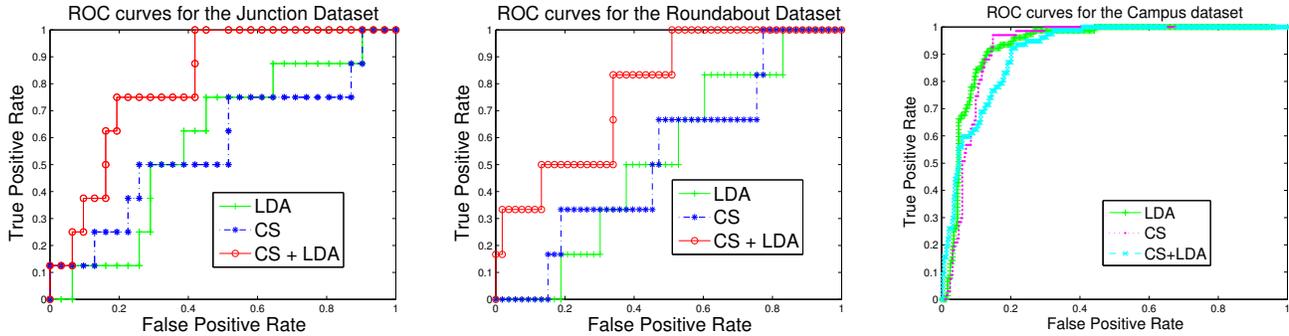


Figure 5. ROC curves. Left: QMUL Junction Dataset. Middle: QMUL Roundabout Dataset. Right: Campus Dataset.

	MOHMM [9]	MCTM [7]	Cas-pLSA [9]	LDA[9]	Cas-LDA[9]	Proposed LDA	Proposed CS	Proposed LDA+CS
Junction	0.6351	0.3911	0.8024	0.3871	0.8589	0.5887	0.5726	0.8105
Round	0.6730	0.5579	0.7154	0.6761	0.7374	0.5283	0.5346	0.7767
Campus	NA	NA	NA	NA	NA	0.9290	0.9202	0.9108

Table 1. AUC for anomaly detection. The proposed LDA+CS outperforms using LDA only and CS only, slightly outperforms Cas-pLDA and match the performance of Cas-LDA.

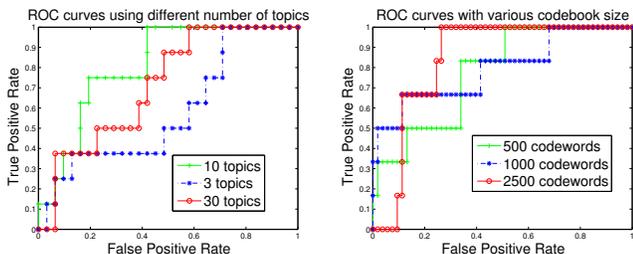


Figure 6. ROC curves using different codebook size and number of topics. The left image shows the ROC curve using various number of topics for the Junction dataset. The right image shows the ROC curve using various codebook size for the roundabout dataset.

be obtained when using a larger size codebook. The right plot in Figure 6 shows the ROC curves for the Roundabout dataset when using a codebook of size 500, 1000, and 2500. When the codebook size is set into 2500, an AUC as high as 0.8428 is achieved, far better from using a codebook of 500 size (0.7767), and exceeding the performance of Cas-LDA (0.7374) and Cas-pLDA (0.7154) in [9]. However, a larger codebook is more computationally expensive.

Since we follow [9] and cut the video into clips, we only have 73 clips in the Junction and 144 in the Roundabout dataset for training; far from enough to train a stable classifier. Thus performance of the experiments is highly related to the initialisation of the LDA parameters, α and β . To look at the effect of the initialisation of parameters, we run the LDA classifier experiments 10 times and compute the average AUCs⁴. In the Roundabout dataset, the joint approach has an average AUC of 0.6491 compared to 0.5245

⁴In the approach using compressed sensing only, we don't perform multiple experiments as there are no model parameters to initialise.

(LDA) and 0.5346 (CS), although we do achieve an average AUC of 0.7211 for the joint approach with a codebook size of 2500. In the Junction dataset, the joint approach has an average AUC of 0.5891 compared to 0.5839 (LDA) and 0.5726 (CS). In the campus dataset, the joint approach has an average AUC as 0.7518, worse than the average AUC of LDA only (0.9290) and compressed sensing (0.9202). We observed that using LDA achieves the best performance for the campus dataset. This is because the unusual events in this dataset are caused by a heavy rain outside the building. The crowd density for the unusual events is higher than the normal case, which results in a larger number of words in those video clips. The EM algorithm in LDA usually reports a lower likelihood with a larger document, resulting in the superior performance. From this evaluation it is clear that the proposed feature system is capable of performing very well, exceeding [9], however performance is dependent on how the system is initialised.

2.3. Temporal Video Segmentation Evaluation

To evaluate temporal video segmentation, we use the QMUL Traffic dataset as used in [9]. We use the same configuration and groundtruth as [9] to segment the video clips into the vertical and horizontal flow phase.

As outlined in Section 2.4, we use two approaches for video segmentation: using LDA only (1) and using both LDA and K-means (2). We conduct experiments 10 times to get the average segmentation accuracies, which are reported in Table 2 and compared with [9]. If the same classifier (LDA) is used, our approach (1) outperforms [9] significantly, especially for the Junction dataset where we achieve

	LDA [9]	Cas- LDA [9]	Proposed (1)	Proposed (2)
Junction	61.5%	87.2%	92.31%	91.28%
Round	55.9%	74.6%	60.17%	76.25%

Table 2. Temporal Video Segmentation Accuracy. The term, Our Approach (1), means the way only using LDA; the term, Our Approach (2), means the way using both LDA and K-means



Figure 7. A video clip often mismatched. Sample images from Junction test dataset Clip 30. Left: the first frame; Right: the last frame.

an average segmentation accuracy of 92.31% (compared to Cas-LDA which achieves 87.2%).

When combining LDA and K-means, we set the number of topics into 10. We cluster the topic distributions into 2 clusters using K-means⁵. We have an average accuracy of 91.28% for the Junction dataset, and 76.25% for the Roundabout Dataset, outperforming the Cas-LDA approach of [9]. We observe that the majority of the classification errors are caused by ambiguities in the dataset. In clip 30 of the Junction dataset, there are several vehicles moving horizontally in the first 200 frames, and a lot of vehicles in the far field begin to move vertically in the last 100 frames. Our algorithm often classifies it as horizontal flow, though the ground truth annotates it as vertical. Figure 7 shows sample images from this clip. If we only use LDA, the result will be more stable. If we use LDA and K-means, the results are more dependant initialisation of the K centres in K-Means, but provided the centres are well initialised the approach can achieve superior performance.

3. Conclusion

In this paper, we have proposed a novel approach to analysing activities in crowded scenes using the DFT coefficients of particle trajectories. The proposed approach has been applied to both anomaly detection and temporal video segmentation, and state-of-the-art performance is achieved. To the best of the authors knowledge, the feature we used has not been used for activity surveillance before. We also observe that that by using Latent Dirichlet Allocation for dimension reduction and then detecting outliers by projecting the data into a high dimensional space, improved performance for anomaly detection can be achieved.

⁵We initialise the K-means centers with a vector by randomly setting half the elements to 1 and the other half to 0

References

- [1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *TPAMI*, 2008.
- [2] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. *Foundations of Data Organization and Algorithms*, 1993.
- [3] E. L. Andrade, S. Blunsden, and R. B. Fisher. Hidden markov models for optical flow analysis in crowds. In *ICPR*, 2006.
- [4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.
- [5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 2004.
- [6] E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 2007.
- [7] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *ICCV*, 2009.
- [8] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. European Workshop Advanced Video Based Surveillance Systems*, 2001.
- [9] J. Li, S. Gong, and T. Xiang. Learning behavioural context. *IJCV*, 2011.
- [10] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.
- [11] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, 2009.
- [12] A. Naftel and S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 2006.
- [13] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Textures of optical flow for real-time anomaly detection in crowds. In *AVSS*, 2011.
- [14] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 2008.
- [15] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *CoRR*, abs/1112.4258, 2011.
- [16] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 2006.
- [17] I. Tziakos, A. Cavallaro, and L.-Q. Xu. Event monitoring via local motion abnormality detection in non-linear subspace. *Neurocomputing*, 2010.
- [18] B. Wang, M. Ye, X. Li, F. Zhao, and J. Ding. Abnormal crowd behavior detection using high-frequency and spatio-temporal features. *Machine Vision and Applications*.
- [19] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *TPAMI*.
- [20] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*.
- [21] S. Wu, B. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes.
- [22] J. L. Yang Cong, Junsong Yuan. Sparse reconstruction cost for abnormal event detection. In *CVPR*, 2011.
- [23] G. Zen and E. Ricci. Earth mover’s prototypes: A convex learning approach for discovering activity patterns in dynamic scenes. In *CVPR*, 2011.
- [24] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. *CVPR*, 2011.