
Proceedings Paper:
Al Ghamdi, M. and Gotoh, Y. (2018) Graph-based correlated topic model for trajectory clustering in crowded videos. In: 2018 IEEE Winter Conference on Applications of Computer Vision. 2018 IEEE Winter Conference on Applications of Computer Vision, 12-15 Mar 2018, Lake Tahoe, NV/CA. IEEE, pp. 1029-1037. ISBN: 978-1-5386-4886-5.

https://doi.org/10.1109/WACV.2018.00118

Graph-based Correlated Topic Model for Trajectory Clustering in Crowded Videos

Manal Al Ghamdi
Umm Al-Qura University
Department of Computer Science
Saudi Arabia
maalghamdi@uqu.edu.sa

Yoshihiko Gotoh
University of Sheffield
Department of Computer Science
United Kingdom
y.gotoh@sheffield.ac.uk

Abstract

This paper presents a graph-based correlated topic model (GCTM) to learn and analyse motion patterns by trajectory clustering in a highly cluttered and crowded environment. Unlike previous works that depend on scenes prior, we extract trajectories and apply a spatio-temporal graph (STG) to uncover the spatial and temporal coherence between the trajectories during the learning process. It advances the CTM by integrating a manifold-based clustering as initialization and iterative statistical inference as optimization. The output of GCTM are mid-level features that represent the motion patterns used later to generate trajectory clusters. Experiments on two different datasets show the effectiveness of the approach in trajectory clustering and crowd motion modelling.

1. Introduction

Trajectory clustering and analysis of crowd movements have been vital components of various applications in public surveillance, such as flow estimation. The goal is to analyze individual movements by a trajectory associated with a cluster label, thus representing individuals' paths. A highly crowded scene is particularly challenging because of the density, heavy occlusions and variations in the view. Additionally interaction between individuals can lead to misdetection of body parts [14]. The presence of such challenges makes it difficult to analyze movements using conventional techniques such as background subtraction and motion segmentation, although they may work effectively with less-crowded scenes.

To overcome the shortcomings of conventional techniques, motion patterns have been introduced for processing crowded scenes. In such a scenario, objects are represented by a small number of pixels; there is thus ambiguity in appearance caused by the dense packing [12]. Therefore, defining the motion patterns in the crowd is the key to the problem. Examples of motion pattern techniques include scene structure-based force models [2] and the Bayesian framework with spatio-temporal motion models [9]. These models are based on the assumption that the objects move coherently in one direction throughout a video. This is a major shortcoming, as it fails to represent the complex crowded scenes with multiple dominant crowd behaviours in each location.

Trajectory clustering is fundamental in various applications such as crowd analysis and video surveillance. In many applications, a vast amount of trajectories and motion patterns are extracted and clustered into groups without manually labeled of the data. Lin *et al.*[10] detected motion trajectories in crowd scenes by processing the flow fields. They then applied a two-step clustering process to define semantic regions which is used later to recognize pre-defined activities in the crowd. Lu *et al.*[11] extracted the motion trajectories to investigate the characteristics of pedestrians in unstructured scenes. Trajectories were firstly represented as a four-dimensional vector, then clustered using fuzzy c-means (FCM) algorithm to form the motion patterns using. Sharma and Guho *et al.*[16] proposed a two steps trajectory clustering approach (TCA) for segmenting crowd flow patterns. Trajectory extraction step to detect and track blocks or regions in the video followed by clustering step that utilised shape, location and the density of the trajectory in the neighborhood. Xu *et al.*[21] combined the mean shift clustering and the manifold-based model to improve the trajectory clustering performance. The center of a cluster is defined by a manifold and the motion pattern is defined by the structure of the cluster. They have shown the results of classification using Hidden Markov Model (HMM) as well as of clustering using k-means algorithm.

Many works have been proposed for trajectory clustering based on mid-level features learning. These features are usually observed as paths defined by individuals' move-

ments, which aim to map the segments of trajectories from low-level feature space to their clusters [23]. Trajectory mid-level features can be learnt with hierarchical latent variable Bayesian models, such as latent Dirichlet allocation (LDA) [5] and the correlated topic models (CTM) [4]. These models are known as 'topic models', adopted from the text processing field. They often have hierarchical structures where the latent variables lie at multiple levels. Using these models, documents are represented by trajectories and visual words are given by observations of object trajectories. With these approaches the learnt topics represent mid-level features of trajectories.

The CTM was adopted by Rodriguez *et al.*[14] as a mid-level feature to represent multiple motion behaviours in one scene. Their tracker was weighted to predict a rough displacement using a codebook generated from all the moving pixels in the scene, along with the learnt high-level behaviour. Although CTM is an effective model, it only considers the motion direction at each spatial location and disregards the temporal correlation between sequential motions that naturally occur in crowd scenes; it can not create discriminative mid-level features for multiple clusters.

A scene prior belief based correlated topic model (BCTM) [23] was then proposed to construct a mid-level features for trajectory clustering. A feature tracker was firstly employed to generate the trajectory. A spanning tree method was then used to define the initial cluster information. The mid-level features were generated using BCTM followed by a hierarchical clustering algorithm to produce the final clusters. Their experiment shows that the BCTM as a trajectory clustering method outperforms the CTM, but it could only be applied if the scenes prior were available.

Zhou *et al.* [22] proposed a random field topic (RFT) model to perform trajectory clustering in a crowd scene. It extended the LDA models by integrating scene prior and using a Markov random field (MRF) algorithm. RFT significantly improve the clustering performance over LDA models; however, the performance can drop in crowded scenes with correlated topics, where topics are shared with multiple clusters, and where clusters are also shared with multiple topics.

Despite the effectiveness of the above models, most of them ignored the temporal relationship within the crowded scenes and also the distribution of data. Therefore, they required a complex parameter estimation and variable inference procedure. This paper presents a graph-based correlated topic model (GCTM) for analysing crowd movements and clustering trajectory in a complex crowd scene. It advances a CTM by integrating a spatio-temporal graph (STG) to enforce the spatial and temporal coherence between trajectories during the learning process. The goal of this work is to address the problem of trajectory clustering and motion pattern analysis in high-density crowds without using any prior knowledge of the motion pattern or the scene. Different from previous works, GCTM has a manifold-based cluster initialization step followed by iterative optimization with Bayesian inference. The initialization step helps our approach to generate topics or motion patterns (mid-level features) that effectively reflect data distribution and cluster information. After the iterative optimization, the generated topics are discriminative where different trajectories are clustered separately in the manifold space.

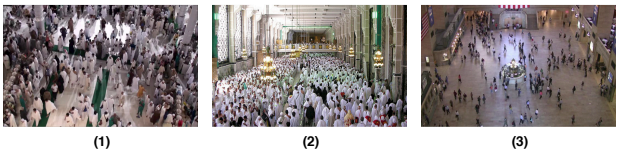


Figure 1. Sample frames from indoor scenes at (1) (2) Al-Masjid Al-Haram [3] and (3) New York's Grand Central Station [22].

We firstly apply the Kanade–Lucas–Tomasi (KLT) tracker [17] to extract trajectories points used later by the locality-constrained linear coding (LLC) technique [20] to generate a set of visual codes as low-level features. The STG is then constructed to uncover the spatio-temporal relations between the trajectories and projected to lower-dimensional space to initialize clusters in a manifold embedding space. Using cluster labels, topics are learnt by the GCTM for final trajectory clustering. Experiments on two different video datasets – one collected at the crowded Grand Central station in New York [22] and the other collected from multiple locations at Al-Masjid Al-Haram [3], both of which are well known for crowded and busy scenes (Figure 1) – show the effectiveness of the presented approach.

The remainder of the paper is organized as follows: a review of the original CTM and the proposed GCTM model are introduced in Section 2. The initial and final trajectory clustering are presented in Section 3. Datasets and experiments set-up are presented in Section 4. We discuss our results in Section 5 and conclude the paper in Section 6.

2. Our Approach

This section outlines how the mid-level features (topics) are learnt as motion patterns (paths) by GCTM parameters estimation. To make the paper self-contained, we start by reviewing the conventional CTM (Section 2.1) followed by the proposed GCTM (Section 2.2).

2.1. Correlated Topic Model

Figure 2(a) shows the graphical representation of the CTM that was originally developed in the text-processing field [4]. Let M, N and K denote the number of documents, the number of words in a document and the number of hidden variables (or 'topics') in the model, respectively.

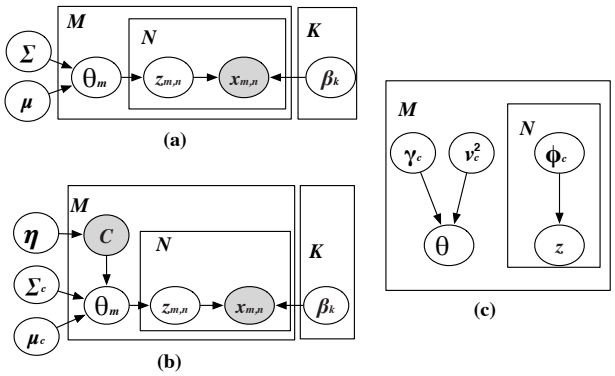


Figure 2. CTM and GCTM models. (a) Graphical representation of CTM [4]. (b) Graphical representation of GCTM. (c) Graphical representation of approximate distribution of GCTM.

The circles in the figure are random variables or model parameters, and the edges specify the probabilistic dependencies (or the conditional independences) among them; boxes, with M, N and K, are compact notations for multiple instances of the variables or parameters. Shaded variables represent the observed variables, while unshaded variables indicate the latent variables. The CTM assumes that each document is a mixture of words based on a set of hidden topics, and in turn each topic is determined by a distribution over the entire vocabulary. In the figure, θ_m (or θ) is a K-dimensional vector, specifying the topic priors for each document; $z_{m,n}$ (or z_n) is a hidden variable, following a parameterized multinomial distribution $Mult(\theta)$; $x_{m,n}$ (or x_n) is the random variable whose value is the observed word (*i.e.*, 'feature'); and β is a hyper-parameter, corresponding to the mid-level features. Finally μ and Σ are the mean and the covariance matrix of the multivariate Gaussian process. The generative process of the CTM is outlined as follows:

- Draw $\theta|\{\mu, \Sigma\} \sim N(\mu, \Sigma)$
- Draw the document-specific topic proportions π as $\pi = \frac{exp(\theta)}{\sum_i^K \theta_i}$
- For each visual word $x_n, n \in \{1, \ldots, N\}$:
  1. Choose a topic assignment $z_n|\theta$ from $Mult(\pi)$;
  2. Choose a word $x_n|\{z_n, \beta_{1:K}\}$ according to $p(x_n|z_n, \beta)$.

  According to this model, the document probability given topic variable θ, word x and individual topic assignment z is:

$$p(\theta, z, x|\mu, \Sigma, \beta) = p(\theta|\mu, \Sigma) \prod_{n=1}^{N} p(z_n|\theta)p(x_n|z_n, \beta) \quad (1)$$

  Notice that the topic-level information given by θ and z is hidden, while the word-level representation is observed.

An approximate method (variational approximation) has been used to estimate the likelihood of performing training and to estimate the most likely topic proportions θ and topic assignments z. Further details can be found in [4].

2.2. Graph-based Correlated Topic Model

Corpus, document, topic and words (for text data) in CTM are replaced with path, trajectory, motion pattern (or topic for simplicity) and visual codes (for video data) in GCTM. The topic mixture of a document corresponds to a set of different motion patterns in a trajectory. GCTM learns crowd movements by clustering trajectories. The graphical representation of GCTM is presented in Figure 2(b). Observed visual codes (low-level features) and the initial clusters are the inputs for GCTM. Section 3 describes the construction of the visual codes and initial clusters as low-level features.

We begin with some notations and definitions for the presented Figure 2(b):

- M, the number of trajectories in the path, each of which is modelled as a mixture of K topics. $m = 1, 2, ..., M$, the index of an individual trajectory in the path.
- N, the total number of visual occurrences in a trajectory m. $n = 1, 2, ..., N$, the index of a visual code occurrence in document m.
- K is the number of hidden topics in the model, where each topic is a distribution over a code set given by the hyper-parameter β_k.
- $c \sim p(c|\eta)$, $c = \{1, \ldots, C\}$, the initial cluster that has to be defined for each trajectory, where C is the total number of initial clusters, and η is a C-dimensional vector of a multinomial distribution.
- θ_m (or θ) is a continuous variable sampled from a Gaussian distribution for choosing the topic $p(\theta_m|\mu, \Sigma, c)$.
- μ is K-vector and Σ is a $K \times K$ covariance matrix, the parameters of a multivariate Gaussian process.
- $z_{m,n}$ (or z_n) is a hidden variable assigned to a visual code x_n drawn from a multinomial distribution.
- $x_{n,m}$ (or x_n) is the visual code n in the trajectory m.

Given the parameters Σ, μ, η and β we can now write the full generative equation of the model. The joint probability

of a topic mixture θ, a set of N topic z, a set of N visual codes x and the cluster c is:

$$P(x, z, \theta, c|\eta, \beta, \mu, \Sigma) = p(c|\eta)p(\theta|\mu, \Sigma, c)\prod_{n=1}^{N} P(z_n|\theta)P(x_n|z_n, \beta) \quad (2)$$

$$p(\theta|\mu, \Sigma, c) = \prod_{c=1}^{C} Mult(\theta|\mu_c, \Sigma_c) \quad (3)$$

$$p\left(c|\eta\right) = Mult\left(c|\eta\right) \quad (4)$$

$$p\left(z_n|\theta\right) = Mult\left(z_n|\theta\right) \quad (5)$$

where $Mult(\cdot)$ is a Multinomial distribution based on parameters μ_c and Σ_c. The distribution of $p(c|\eta)$ is always assumed to be a fixed uniform distribution in which $p(c) = 1/C$. Therefore, we will leave out the estimation of η.

The log probability for x is given as:

$$p(x|\mu, \Sigma, \beta, c) = \int p(\theta|\mu, \Sigma, c) \left(\sum_{z}\left[\prod_{n=1}^{N} p(x_n|z_n, \beta)p(z_n|\theta)\right]\right) d\theta \quad (6)$$

In order to estimate parameters for GCTM, we used parts of video sequences as training data and adopt the variational expectation maximization (EM) algorithm to do variable inference and parameter estimation [4]. Figure 2(c) is the graphical representation of the approximate distribution of the GCTM where $\gamma_{M\times K}$, $v_{M\times K}$ and Φ are variational parameters. Therefore, the log-likelihood for a document m is given by:

$$log\, p(x|\mu, \Sigma, \beta, c) = L(\gamma_c, v_c, \phi_c; \mu_c, \Sigma_c, \beta) + KL(q(\theta, z|\gamma_c, v_c, \phi_c)||p(\theta, z|x, \mu_c, \Sigma_c, \beta)) \quad (7)$$

We iteratively maximize the term $L(.)$ instead of $p(x|\mu, \Sigma, \beta, c)$, which results in the minimum of difference between the distribution in Figure 2(b) and Figure 2(c). For details of computation, please refer to [4]. We give modified parameters and variables as:

$$\phi_{ki}^{c} \propto exp\{\gamma_k^c\}\beta_k \quad (8)$$

$$\beta_k \propto \sum_i \phi_{ki}^{c} n_i \quad (9)$$

$$\mu = \frac{1}{M}\sum_m \gamma_m^c \quad (10)$$

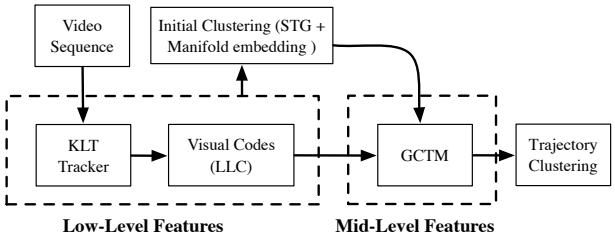


Figure 3. Flow chart of the crowd behaviours modelling framework with GCTM.

$$\Sigma = \frac{1}{M}\sum_m diag\left(v_m^c\right) + \left(\gamma_m^c - \mu_c\right)\left(\gamma_m^c - \mu_c\right)^T \quad (11)$$

where m is used to index the trajectory, i to index the word and k to index a topic. ϕ_{ki} denotes the probability that the ith word belongs to the kth topic, n_i is the word count and β_k is the kth topic's representation in the word space.

3. Low-level Features

In trajectory clustering, the first step is to generate the low-level features by extracting the trajectory segments and representing them with a collection of visual codes (*i.e.* words). The second step is to apply a spatio-temporal graph on the visual codes to uncover spatio-temporal relations among trajectories and embed them in the lower-dimensional space to define the initial cluster. Given initial clusters and the set of visual codes, the final step is to learn the mid-level features by GCTM (Section 2.2) and to produce the final trajectory clustering. The framework is shown by a flow chart in Figure 3.

Low-level Features Given a video sequence, the KLT tracker [17] is firstly applied to calculate M trajectories. The LLC algorithm [20] is then employed to represent each trajectory with a set of visual codes X as low-level features. LLC is a coding scheme proposed by Wang *et al.* [20] to project features onto their respective local coordinate systems and encode them using fewer codebook basis in the high-dimensional feature space.

Given a trajectory m with a set of points $m = \{t_1, \ldots, t_N\}$, the set of codes $X = \{x_1, \ldots, x_N\}$ are derived by firstly constructing a neighbourhood graph based on the geodesic distances between the trajectory points and the codebook, then computing the shortest path, performing a kNN search, and finally solving the following constrained least square fitting problem:

$$\min_X \sum_{i=1}^{N} \|t_i - Bx_i\|^2 + \lambda\|d_i \odot x_i\|^2 \quad st.\ 1^\top x_i = 1,\ \forall i \quad (12)$$

where $\odot$ is the element-wise multiplication, B is a codebook, and λ is a sparsity regularization term. Furthermore,

'$1^\top x_i = 1,\ \forall i$' means the shift-invariant requirements for the LLC code. The locality-constrained parameter d_i represents each basis vector with different freedom based on its shortest path to the trajectory point t_i. The final step uses the multi-scale max pooling [15], where the sets of codes computed for each trajectory are grouped together to create the corresponding pooled representation X.

Initial Clustering To obtain the initial clusters C for the trajectories, we applied the STG algorithm [1] to uncover spatio-temporal relations among trajectories and connect them as initial clusters. The structure in the high-dimensional space is transferred to a spatio-temporal distance graph with nodes representing LLC representations. The method reconstructs the order of the LLC representations based on their spatio-temporal relationship and recalculates distances along them to ensure the shortest distance. First the similarity matrix S is calculated between the LLC representations using the Euclidean distance. The value of S_{ij} defines the distance between X_i and X_j of two trajectories ($i, j = 1, \ldots, M$). Then for each instance X_i ($i = 1, \ldots, M$):

1. L codes, whose distance is the closest to X_i, are connected. They are referred to as spatial neighbours (sn):

$$sn_{X_i} = \left\{ X_{j1}, \ldots, X_{jL} \mid \underset{j}{\operatorname{argmin}}^{L}(S_{ij}) \right\} \quad (13)$$

   where $\underset{j}{\operatorname{argmin}}^{L}$ implies L node indices with the shortest distances.

2. Another L chronologically ordered neighbours around each code X_i are set as temporal neighbours (tn):

$$tn_{X_i} = \left\{ X_{j-\frac{L}{2}}, \ldots, X_{j-1}, X_{j+1}, \ldots, X_{j+\frac{L}{2}} \right\} \quad (14)$$

3. Optimally (tn_{sn}) is selected from temporal neighbours of spatial neighbours as:

$$tn_{sn_{X_i}} = \left\{ tn_{X_{j1}} \cup tn_{X_{j2}} \cup \ldots \cup tn_{X_{jL}} \right\} \cap tn_{X_i} \quad (15)$$

4. The union between spatial and temporal sets represents spatio-temporal neighbours (stn_{X_i}) for code X_i as:

$$stn_{X_i} = sn_{X_i} \cup tn_{sn_{X_i}} \quad (16)$$

The above formulation of stn_{X_i} effectively selects X_i's temporal neighbours that are similar, with a good chance, to its spatial neighbours.

Given the spatio-temporal neighbourhood graph, a new correlation δ based on the geodesic distances is defined by applying Dijkstra's distance algorithm between the neighbouring nodes [18]. The $\delta = \omega_{ij}$ value represents the shortest path distance (neighbour weights) between two nodes X_i and X_j. If node X_j is a spatio-temporal neighbour of X_i and $j \in stn_{X_i}$, then $\delta(X_i, X_j) = \omega_{ij}$ and their trajectories have neighbor relations, otherwise, $\delta(X_i, X_j) = 0$.

The manifold embedding is then modelled by applying the multidimensional scaling [6]. It is formed as a transformation of the high-dimensional data in terms of the correlation δ into a new d-dimensional embedded space that best preserves the neighbouring relations of the clusters. In the lower dimensional manifold embedding space, a k-means algorithm [7] is adopted to perform clustering and obtain initial trajectory cluster labels.

Final Clustering After the mid-level features are learnt and the topic probabilities of the trajectories are computed, each trajectory has a set of K topics to choose from. A topic label with the highest probability is assigned to the trajectory.

Given a new trajectory m with an unknown path, LLC representation X is firstly defined with N visual codes and the probability of each cluster is computed as:

$$p(c|x, \mu, \Sigma, \beta, \eta) \propto (x|c, \mu, \Sigma, \beta)p(c|\eta) \propto (x|c, \mu, \Sigma, \beta) \quad (17)$$

where μ, Σ, β and η are parameters learnt by the GCTM model. The decision of the topic is then made by comparing the likelihood of X given each cluster label as

$$\underset{c}{\operatorname{argmax}}\, p(x|\beta, \mu, \Sigma, c) \quad (18)$$

where the term $p(x|\beta, \mu, \Sigma, c)$ is defined as in Eq. 6.

4. Datasets and Experimental Setup

We evaluated the graph-based correlated topic model (GCTM) using a trajectory clustering task in crowded videos. Once the GCTM model is learnt, trajectories are clustered based on the motion pattern they belong to. For each trajectory, the decision of the topic is made to the cluster that gives the highest likelihood probability. Two different datasets were employed for evaluation.

- New York's Grand Central Station [22] — collected from the inside of the Grand Central railway station in New York, USA. It contains multiple entrances and exits where individuals have different paths to follow. Therefore, the crowd presents multiple behaviours (or paths) in various moving directions.

- Al-Masjid Al-Haram [3] — collected from indoor scenes at the holy mosque of Mecca, Saudi Arabia. This dataset involved a number of difficult problems,

Dataset	Resolution	Duration	Codebook size	Trajectories
Al-Masjid (S1)[3]	960×540	$5,600$ sec	$96 \times 54 \times 4$	87,321
Al-Masjid (S2)[3]	960×540	$3,400$ sec	$96 \times 54 \times 4$	61,760
Station [22]	720×480	$1,800$ sec	$72 \times 48 \times 4$	47,866

Table 1. The resolution, duration, codebook size and number of extracted trajectories for each dataset.

such as lighting changes, occlusions, a variety of objects, changes of views and environmental effects. Al-Masjid videos were collected from two scenes. The first was at one of the Tawaf area stairs used to enter or leave the Tawaf. It is a very busy area and needs monitoring to ensure individuals' safety. Multiple paths can be defined at this scene including (1) a direct path to approach the Tawaf, (2) the left and the right side paths leading to the seating areas. Currently this area is monitored and managed by the security officers.

For simplicity, we denote the first dataset as 'Station' and the second one as 'Al-Masjid (S1)' and 'Al-Masjid (S2)'. The details of both datasets are presented in Table 1. For the low-level feature step, the initial codebook B used for the LLC codes was learnt from a random half of the trajectories. In both datasets, the size of the codebook was designed as follows: the $W \times H$ scene was divided into 10×10 cells and the velocities of key-points were quantized into four directions. In both datasets, the pooled representations from the LLC codes were computed for each sub-region (of 4×4, 2×2 and 1×1) and pooled together using the multi-scale max pooling. The following parameters were used: the number of neighbours $k = 5$ and $\lambda = 500$ in Eq. (12). For the initial clustering, we used Elkan's k-means clustering algorithm from the *VLFeat toolbox* [19], which was faster than the standard Lloyd's k-means. The pooled features were then concatenated and normalized using the ℓ^2-norm. For the STG, the similarity matrix was computed using the Euclidean distance and the KNN graph was constructed with $L = 20$.

5. Results

Figures 4(a), (c) and (e) show that crowd movements learnt by GCTM presented clearly discriminative paths in the scene. Each direction of crowd movement was assigned with a different colour. Trajectory clusters, generated by the clustering algorithm, are identified by different colours and presented in Figures 4(b), (d) and (f). In both datasets most trajectory segments were broken; however, spatially distant trajectories could be clustered in one group when they were found to have the same path. For example, the leftmost cluster from Al-Masjid (S1) shown in Figure 4(b) contained trajectories for pedestrians walking towards the left side of the scene. It was not easy to obtain this cluster because occlusion caused by the people sitting on the marble pillar resulted in trajectories observed mostly either at the start or

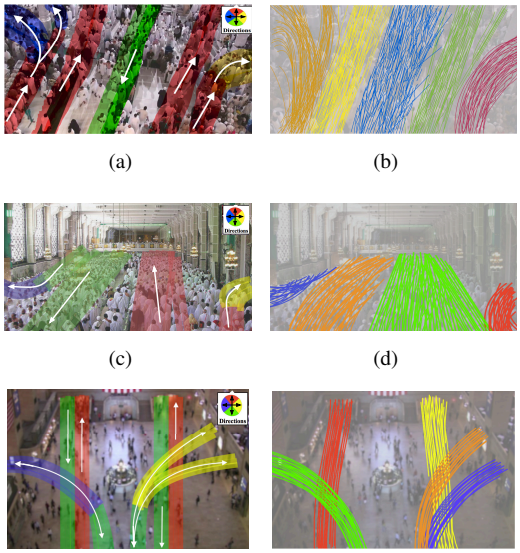


Figure 4. (a), (c) and (e) present learnt topics by GCTM for Al-Masjid (S1), (S2) and Station. (b), (d) and (f) are their trajectory clusters. (Seen better in colour.)

the end of the path. In Figure 4(d), movements were clustered into four groups; one of them was up the left side with an exit and another one was down the right side with another exit. Trajectories were mixed with adjacent paths and occluded by the heavy traffic; however, GCTM was able to identify these paths and their exit positions. Similarly, in Figure 4(f), trajectory segments were clustered into five different paths; two of them were on the right side to exit the station. Trajectories were shared between these two exits, but the GCTM was able to distinguish between their paths.

Figure 5 presents trajectory clusters from Al-Masjid (S1) by various approaches, including GCTM, random field topic[1] (RFT) [22], CTM [14] and spectral clustering (SC) [8]. We implemented the SC using a linear interpolation and the Euclidean distance to measure the similarities. Different colours in the figure represent different clusters (paths). It can be observed that GCTM was able to produce the cleanest trajectory paths and clusters. The other three approaches failed to perform trajectory clustering, which was particularly evident with the side paths towards the exits because of their heavy occlusion. RFT achieved better results for the central paths in comparison to CTM and SC. SC was the worst. It was only able to cluster the trajectory segments at one end of the movements (the starting or ending positions) as one path and the other end as a different path.

For further quantitative evaluation of the clustering performance, we adopted correctness and completeness introduced by [13]. Correctness is the accuracy with which a pair of trajectories from different pathways (with the

[1] We used the publicly available code from the authors' websites.

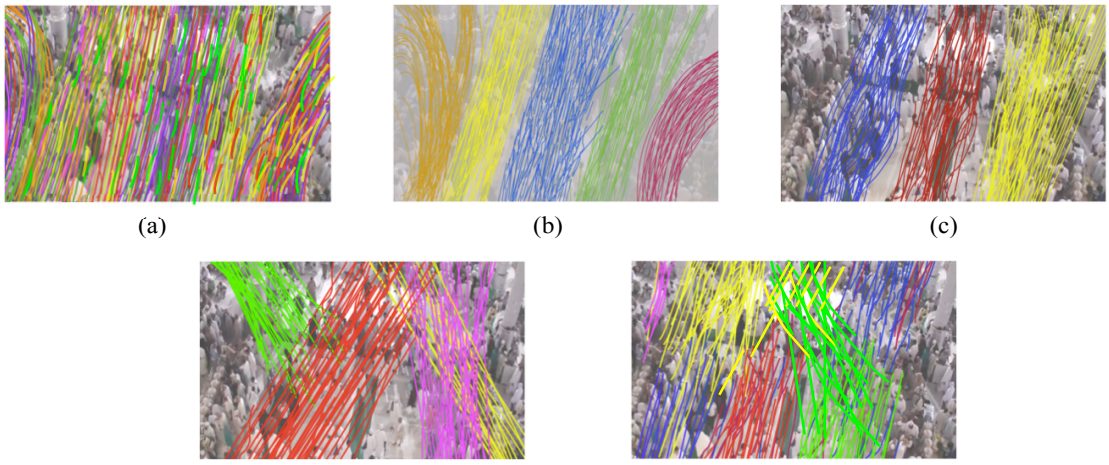
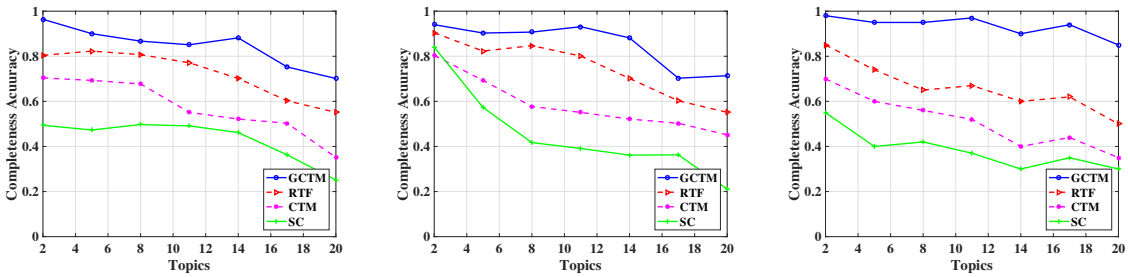


Figure 5. Comparison of trajectory clustering approaches: (a) original trajectory set, (b) GCTM, (c) RFT, (d) CTM and (e) SC.



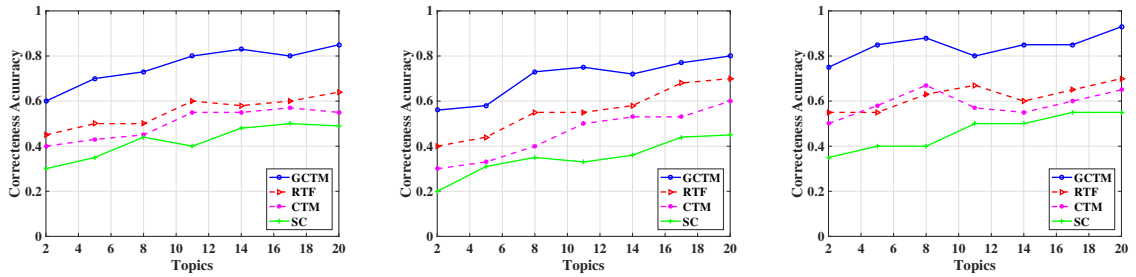
(a) Al-Masjid (S1) (b) Al-Masjid (S2) (c) Station

Figure 6. Completeness accuracies of trajectory clustering approaches.

groundtruth) are clustered into different groups. Completeness is the accuracy with which a pair of trajectories from the same path are clustered into the same group. In an extreme case, a 100% completeness and 0% correctness may be achieved when all the trajectories are clustered into a single group. Another extreme is 0% completeness and 100% correctness, achieved when each trajectory is clustered into a different group. A good clustering algorithm should achieve high percentages in both correctness and completeness. As a groundtruth we manually labelled 2,500 trajectories for correctness and 1,700 for completeness with Al-Masjid (S1), 2,000 trajectories for correctness and 1,500 for completeness with Al-Masjid (S2) and 2,000 trajectories for correctness and 1,500 for completeness with Station.

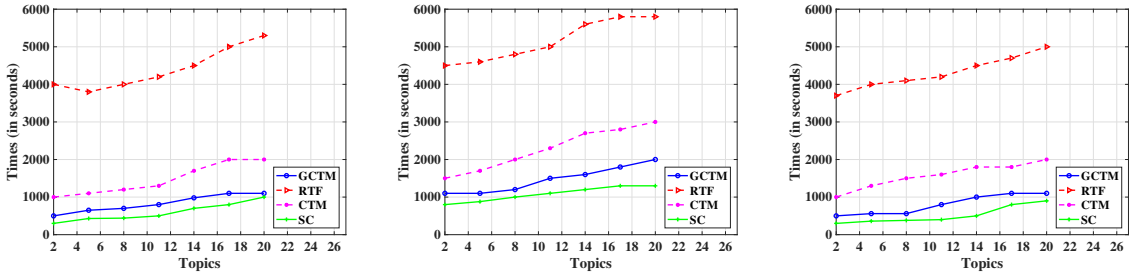
Correctness and completeness for GCTM, RFT, CTM and SC are reported in Figures 6 and 7. The correctness and completeness results show that GCTM outperformed the other three approaches in both datasets with a clear margin. The margin was even wider for completeness when the number of topics was larger. The GCTM with the STG is able to learn discriminative mid-level features better, even with a large number of topics to share the clusters. The other three approaches did not cluster trajectories well because most of these trajectory segments were short and mixed and difficult to be clustered. RFT has advanced the LDA [5] by considering belief priors based on the position and the spatial correlation of trajectories along the video sequence. However, the spatio-temporal correlation between trajectories was disregarded. CTM considered four motion directions at each spatial location, but it ignored the temporal relation between sequential local motions in crowded scenes. SC was adversely affected by the outliers because it relied on the linear distance for clustering and did not consider ordering of points or the direction of moves. All three methods process low-level features of the trajectories in the high-dimensional feature space, which is very sparse, making it difficult to directly perform clustering.

As the Station video is not as crowded as the Al-Masjid



(a) Al-Masjid (S1) (b) Al-Masjid (S2) (c) Station

Figure 7. Correctness accuracies of trajectory clustering approaches.



(a) Al-Masjid (S1) (b) Al-Masjid (S2) (c) Station

Figure 8. Comparison between the model learning time.

videos, it generates higher accuracies via all the approaches. This is because most of the trajectories generated in the Al-Masjid datasets are short and mixed in. Therefore, SC, CTM and RFT often failed to cluster them and achieved lower completeness in Figures 6(b) and (c). In the Al-Masjid (S2) videos, some of the trajectories are lying on the sides (blue and red trajectories in Figure 4(d)), and the other three approaches failed to perform trajectory clustering (Figure 6(b)). In contrast, GCTM (with no scene priors) performs well.

Figure 7 shows that GCTM had better correctness accuracies compared to the others. RFT, with its priors information achieved the second best performance apart from the Station videos, where CTM with five and eight topics in the Station dataset outperforms RFT. This is because the CTM approach could perform well where scenes were not too crowded (*e.g.*, Station, as opposed to Al-Masjid), and thus full and complete trajectories could be generated with its object-tracking algorithm. They were clustered well by the CTM; however, the accuracy dropped as the number of topics increased.

Finally, Figure 8 presents a comparison of GCTM, RFT, CTM and SC with regard to the topic learning time under a different number of topics. These times include the preprocessing time of feature detection, codebook generation, the topic learning and the final clustering on a 2.6 Ghz machine. The figures show that the learning process of the proposed GCTM model is faster than RFT and CTM. Generating the LLC codes as low-level features, defining the STG between the trajectory segments and supporting the topic learning process with initial clusters help to improve the computational aspects of topic modelling. While computing the scenes prior in RFT and tracking individuals with Optical flow in CTM are computationally more expensive. On the other hand, SC has slightly faster processing time than the GCTM. This is expected, since its clustering process does not involve tracking of features nor does it consider spatio-temporal relations between trajectories. However, SC achieved the worst results in analysing the motion patterns.

6. Conclusions

We have proposed a graph-based correlated topic model (GCTM) for learning and clustering crowd movement from trajectory segments. Using a spatio-temporal graph and manifold-based clustering, GCTM can effectively reflect the relations between trajectories, and learn discriminative motion patterns (topics) from crowded scenes. Experiments and comparisons with recent methods have shown that GCTM is faster and more able to learn a crowd topic model and to cluster trajectories. It has been shown that the learnt topics were able (1) to separate different paths at fine scales with a good accuracy, and (2) to capture the global structures of the scenes in long ranges, clearly interpreting crowded movements.

References

[1] M. Al Ghamdi and Y. Gotoh. Video clip retrieval by graph matching. In *ECIR*, 2014.

[2] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *ECCV*, pages 1–14, 2008.

[3] Y. Ali, B. Zafar, and M. Simsim. Estimation of density levels in the holy mosque from a network of cameras. In *Conference on Traffic and Granular Flow*, 2015.

[4] D. Blei and J. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

[6] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[7] C. Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 147–153. AAAI Press, 2003.

[8] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based surveillance video retrieval. *IEEE Transactions on Image Processing*, pages 1168–1181, 2007.

[9] L. Kratz and K. Nishino. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *IEEE Trans. on PAMI*, 34(5):987–1002, 2012.

[10] W. Lin, Y. Mi, W. Wang, J. Wu, J. Wang, and T. Mei. A diffusion and clustering-based approach for finding coherent motions and understanding crowd scenes. *IEEE Transactions on Image Processing*, 25(4):1674–1687, 2016.

[11] W. Lu, X. Wei, W. Xing, and W. Liu. Trajectory-based motion pattern analysis of crowds. *Neurocomputing*, 247:213 – 223, 2017.

[12] W. Luo, J. Xing, X. Zhang, X. Zhao, and T. Kim. Multiple object tracking: a literature review. *CoRR*, 2014.

[13] B. Moberts, A. Vilanova, and J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *IEEE Visualization*, pages 65–72, 2005.

[14] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *ICCV*, pages 1389–1396, 2009.

[15] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proceedings of the IEEE CVPR*, 2005.

[16] R. Sharma and T. Guha. A trajectory clustering approach to crowd flow segmentation in videos. In *The IEEE International Conference on Image Processing (ICIP)*, pages 1200–1204, 2016.

[17] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, 1991.

[18] L. van der Maaten, E. Postma, and H. van den Herik. Dimensionality reduction: a comparative review, 2008.

[19] A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, pages 1469–1472. ACM, 2010.

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010.

[21] H. Xu, Y. Zhou, W. Lin, and H. Zha. Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 4328–4336, Dec 2015.

[22] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *CVPR*, pages 3441–3448, 2011.

[23] J. Zou, Q. Ye, Y. Cui, D. Doermann, and J. Jiao. A belief based correlated topic model for trajectory clustering in crowded video scenes. In *ICPR*, pages 2543–2548, 2014.