Video Trajectory Analysis

Jiang Bian

Faculty of Engineering and Information Technology

University of Technology Sydney

A thesis submitted for the degree of

Doctor of Philosophy

2019

To my loving parents
Zuosen Bian and *Yanhui He*
my wife
Yiran Ding
and my children
Xuyuan Bian and *Jingbo Bian*

Certificate of Original Authorship

I, Jiang Bian declare that this thesis, is submitted in fulfilment of the requirements for award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by an Australian Government Research Training Program.

Jiang Bian

Acknowledgements

I would like to express my special appreciation and thanks to my supervisor Professor Dacheng Tao and Professor Maolin Huang. Dacheng accepted my application four years ago and gave me the chance to change my life. In first 45 months, he consistently taught me how to research. From reading literatures to writing my own papers, from implementing others' methods to devising my own methods, I learned a lot under his instructions. Maolin gave me the chance to finish my PhD degree and taught me how to write papers and submit one conference paper to review under his supervision. He gave his total support to help me improve myself.

I have to say thank you to my wife Yiran Ding, she tried her best to support me and gave me all her have to help me finish my degree without disturbing. I also wish to give special thanks to Chaoyue Wang, Guoliang Kang, Dayong Tian and Jiayan Qiu. Without their helps, my life in Sydney would not be so easy. I am grateful to Zijing Chen. Our work stations are neighboring. Her opinions deeply impressed me and gave me more space to explore. I would like to give my gratitude to Jun Li and Maoying Qiao for their helps on my research.

I would like to give thanks to my friends I met in Sydney, especially to Zhenkai Hao. Zhenkai is an graduated student from University of Sydney. He shared lots of his experiences on learning English to me, and life skills.

Finally, I would like to express my gratitude to my family, my parents and my children, for their encouragement and support.

Abstract

Considering the critical role of trajectory data mining in modern intelligent systems for surveillance security, abnormal behavior detection, crowd behavior analysis, and traffic control. Furthermore, with the widely spreading of camera, trajectories are recorded by camera, so trajectory analysis including trajectory clustering in computer vision is of great use for a lot of works. However, video trajectories analysis is also a hard work, because its limited information to generate trajectories and few representation methods are available. Thus, the better performance could be reached if more reliable motion information is employed. A lot of characterizations are contained in trajectory data that can be useful and powerful in trajectory clustering including distance, speed, direction, relative displacement and some other features. Finally, in the case that a large number of trajectory data need to be cluster into small number of categorizes which are hidden "groups", an unsupervised clustering model is also required to implement the goal.

In addition, with more and more lecture videos are available on the Internet, on-line learning and e-learning are getting increasing concerns because of many advantages such as high degree of interactivity. The semantic content discovery for lecture video is very important. However, every lecture video contains a lot of semantic information including spoken language and lecture notes, so how to use all these features is a key problem to improve the performance of e-learning.

Therefore, a novel method is proposed in this paper. Reference points are detected and the scale-invariant feature transform (SIFT) descriptor is used to represent the image patches around the points. In addition, SIFT is a descriptor that is fast and robust to match. In order

to unify the lengths of trajectories, Discrete Fourier Transformation (DFT) transforms trajectories into frequency domain with a fixed length, so that pattern information is retained. Furthermore, one more feature type is involved to describe object motion that presents the motion of object relative to the camera, and the difference between the static objects and moving objects can be figured out.

Latent Dirichlet Allocation (LDA) has great performance on natural language processing, but it prefers to model discrete words only. However, another different kind of semantic feature, continues feature, involves in, so we derive a novel clustering model called derived LDA model which the word-topic distribution following Multivariate distribution. After derived LDA, we derive dual-variable LDA model that processes two different features parallel. Furthermore, a detailed derivative process is given to support our model.

In the experiment, we applied our model into two data sets including lecture video and KITTI data set. In lecture video data set, the speaking content and the notes on presentation slides are extracted from the lecture videos, and dual-variable LDA model involves to cluster the videos. For KITTI data set, derived LDA model is applied to consider continue feature only, and dual-variable LDA model is employed to process two kinds of features. The experimental results show that the proposed method can effectively discover the meaningful semantic characters of the lecture videos.

Contents

List of Figures

Chapter 1

Introduction

With the development of tracking and surveillance devices, a tremendous amount of object trajectory data has been collected, which makes extracting useful information both imperative and challenging. Trajectory clustering is an efficient method of analyzing trajectory data that has been applied to pattern recognition, data analysis, machine learning, and many other areas. One of the benefits of trajectory clustering is its ability to reveal the spatiotemporal information contained in trajectory data. Hence, it has become ubiquitous in some fields of application, such as object motion prediction [24], traffic monitoring [6] [48] [81], activity understanding [8] [135] [151], abnormal detection [20] [141] [156] [163], 3 dimensional reconstruction [70], weather forecasting [38] and geography [93].

In technical details, trajectory clustering aims to recognize objects through a unique motion status or track. This requires measuring multiple features, each representing different characteristics. Furthermore, it is worth noting that the selected clustering method should consider a trade-off between the extracted features in Fig.1.1. For example, a positional feature should be coupled with a PCA or densely-based method to improve clustering performance.

In order to measure similarities among different types of trajectory data, data representation, feature extraction and distance metric selection are critical preliminary works of trajectory clustering. For example, trajectories can be represented as a vector and downsampled to a unified length, so Euclidean distance is used [98]. Trajectories also can be treated as samples of a probabilistic distribution. Hence, Bhattacharyya Distance [79] is used to measure the distance

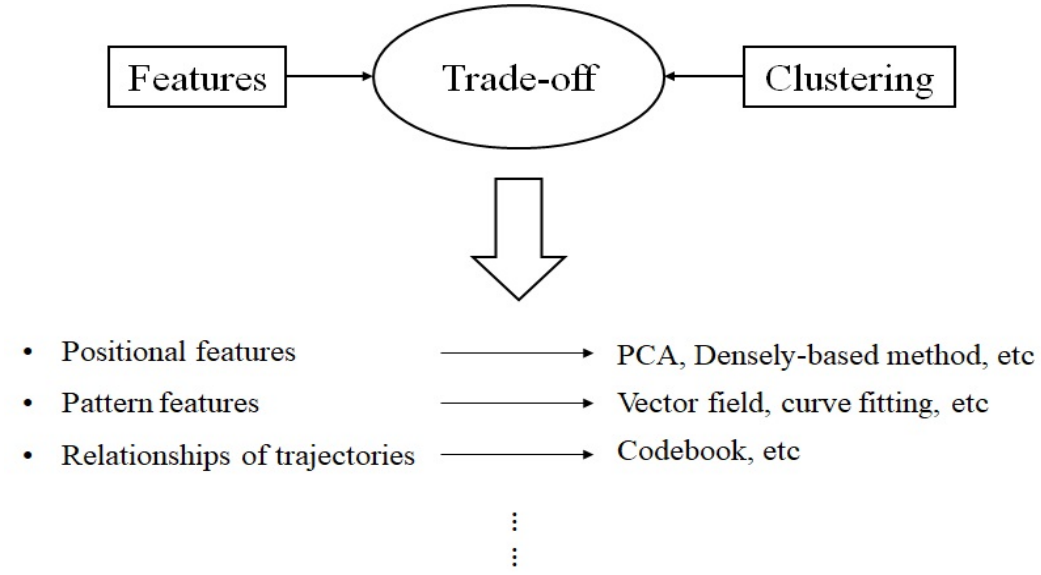

Figure 1.1: The trade-off between features and clustering.

between two distributions.

Hence, in more details, it is critical to generate trajectory data from video dataset robust, because the objects are influenced by illumination, environment and the performance of recording device. Then, a feature descriptor is also needed to fix the varying trajectories lengths and keep more information. Furthermore, the descriptor should considers the format of trajectory data, either. For example, 3D positional information are recored by GPS device, but only 2D positional information are applied in video trajectories. Finally, a proper clustering method is implemented here which may considers different feature types. Thus, the thesis is organized as followed, the related works are reviewed in section 2, trajectory generation and feature extraction are described in section 3 and 4, and section 5 discuss our novel video trajectory analysis methods.

1.1 Research Challenge

This study explores three main research questions (RQ):

RQ 1: How to generate trajectories?

RQ 2: Which features are extracted?

RQ 3: How to cluster trajectories?

To answer these research questions, three corresponding research objectives are given.

RO 1: Trajectory generation

The images from existing positioning systems, such GPS, only contain pixel information but an appropriate method for tracking the key points of objects is still needed. This demands a fast and robust feature descriptor that can accurately match key points in different frames.

RO 2: Feature extraction

Once a trajectory has been generated, it is critical that each trajectory has a different the length. Further, since so much information is required for clustering, e.g., shape, position, length, direction, variation, etc., a more flexible feature is required as opposed to simply resampling or substituting a trajectory with sub-trajectories.

RO 3: Trajectory clustering

Clustering is trajectory analysis technique that groups different parts of a dataset according to similarity. Given the success of LDA in discrete data clustering tasks, like natural language processing, an LDA model that can cluster trajectories based on complex features is needed.

1.2 Research Objectives

I aim to develop trajectory clustering method to deal with video data set by using SIFT, DFT and novel LDA called dual-variable LDA model.

- Objective one: Trajectory generation
  Limited information is available in a video dataset for generating trajectories after the pixels have been chosen as tracking points. Therefore, a fast, robust algorithm is needed to match pixels in consecutive frames. Many studies have turned to the SIFT algorithm to detect and describe a pixel's

local features because of its robustness in extracting features and its speed in matching arbitrary pixels. Further, SIFT generates a new trajectory when an arbitrary pixel cannot be matched in the next frame.

- Objective two: Feature extraction
  As previously mentioned, it is critical that the length of each generated trajectory is different. Moreover, the spatiotemporal, shape and position information contained in a trajectory is key to clustering. Therefore, a better feature extraction method is vital, and DFT has emerged as a promising approach. DFT can describe a trajectory by transforming an original sequence into the frequency domain with an arbitrary fixed length. This technique only requires one further feature that describes the object's movement condition to determine the motion of the object's trajectory. For example, if a frame is split into 3x3 patches, an object is recognized as moving when the trajectory travels through different patches, but static when it stays in the same patch throughout the entire video relative to the camera.

- Objective three: Trajectory clustering
  LDA is a generative statistical model that groups similar parts of a dataset using unobserved criterion. It is a technique that has been used in natural language processing many times. LDA only processes discrete data, using multinomial distribution, but some of the features in trajectories are discrete; the others are continuous. Thus, a novel model is needed to simulate both types of features and combine their corresponding distributions. An dual-variable LDA model is proposed to address this problem.
  In dual-variable LDA, discrete data such as object movement feature are modeled by Multinomial distribution. Dirichlet distribution parameterized by α simulates the prior on per-document topic distribution, and the distribution parameterized by β simulates the prior on pre-topic word distribution. For continuous features, DFT coefficients, the features belonging to same frequency domain comprise word dictionary. Each word is modeled by Multivariate distribution because each word obeys Gaussian distribution.

The prior on per-topic word distribution is used to compute the parameters of the multivariate distribution, μ and Σ. Furthermore, the parameters distributions are conjugate prior distribution.

1.3 Summary of Contributions

The main contribution of this thesis is incorporating different features into video trajectory analysis. A novel topic model is presented that classifies trajectories by considering multiple features. A brief outline of the thesis is provided in the following paragraphs.

Chapter 2 includes a comprehensive review and discussion of the state-of-the-art techniques in trajectory generation, trajectory feature extraction, and trajectory clustering methods.

Chapter 3 explains how the SURF detector and the SIFT descriptor have been implemented to track the characteristic points in objects.

Chapter 4 describes the spatiotemporal information extraction process for different measurement spaces and motion statuses. These features uniquely represent each trajectory and serve as comparisons for other features in the trajectory data.

Chapter 5 provides a brief review of the topic models and presents the clustering algorithm step-by-step. This chapter also includes the multi-modal LDA model that processes two features of the same type, and the dual-variable LDA model that processes two feature of different types.

And my algorithm flow chart is shown in Fig.1.2.

1.4 Outline

1.4.1 Significance

SIFT's robustness and scale invariance properties have been validated in numerous computer vision tasks, make it the preferred choice over other techniques

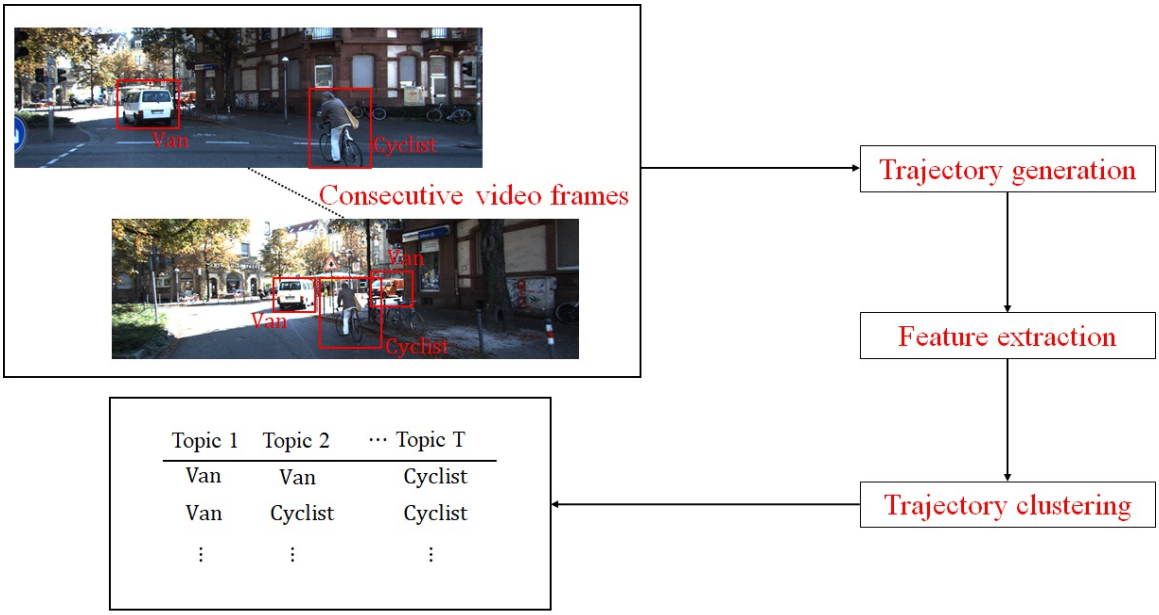

Figure 1.2: Algorithm procedure. Objects generate multiple trajectories which start with the characterize point and classify into categories, each category has its unique semantic information representing only one type object.

for many scholars. However, SIFT could be extended to consider more features, including the relative motion of objects and DFT coefficients. Relative motion is a continuous feature that describes the movement condition, while DFT coefficients represent motion patterns by projecting high-dimensional data into a low-dimensional construct. The novel LDA model presented in this thesis balances both features to generate richer trajectories.

1.4.2 Innovations

SIFT tracks reference points for objects in video dataset, and trajectories are represented and characterised by a DFT feature space and the relative motion of the object. Hence, motion patterns and motion gestures can be generated.

This thesis presents a novel LDA model that incorporates two different types of features and a Gibbs sampler to correct the probability over many states.

1.5 Publications Related to This Thesis

1. **Jiang, B.**, Dayong, T., Yuanyan T., and Dacheng T. (2019). *Trajectory Data Classification: A Review.* ACM Transactions on Intelligent Systems and Technology (TIST), *accepted.*

2. **Jiang, B.**, Maolin H. (2019). *Semantic Topic Discovery for Lecture Video.* Intelligent Systems, *accepted.*

Chapter 2

Related works

This chapter establishes the definitions for key concepts in the field of video trajectories, including trajectory generation, characterizations methods and clustering models, and a review relating to trajectory clustering is also given [**?**] With this background information in place, the overall goal and broad procedure for extracting trajectory data from consecutive video frames is outlined. This includes the issues such as appropriate feature description, identifying the type of object associated with a trajectory, and classifying trajectories into categories.

Before the following discussion, all abbreviations are listed in Table.2.1.

2.1 Background of Trajectory Generation

Generally, trajectories are generated from 2D location information that has been recorded on a device, such as Global Positioning System (GPS). However, 2-dimensional data lacks a great deal of detail that can influence the accuracy of clustering, for example, the scale of the object, the range of movement, how quickly the object moved. Hence, ideally, trajectory data should contain 3D coordinates and spatiotemporal information.

Trajectory data are recorded in different formats according to device types, object motion or even purposes. For instance, GPS tracking devices generate a trajectory by tracking object movement as $Trajectory = (Tr_1, Tr_2, \cdots, Tr_N)$, which is a consecutive sequence of points in geographical space, and Tr_i denotes

Table 2.1: Multimodal-LDA variable list

abbreviation	model name
SIFT	scale-invariant feature transform
SURF	speeded up robust features
PCA	principal component analysis
SVM	support-vector machines
DFT	discrete fourier transform
MDL	minimum description length
MBR	minimum bounding rectangle
EM	expectation–maximization
KLT	Kanade–Lucas–Tomasi feature tracker
FCM	Fuzzy C-means
HITS	hypertext included topic search
TAD	test-and-divide
SVD	singular-value decomposition
k-NN	k-nearest neighbors
GMM	Gaussian mixture model
MCMC	Markov chain Monte Carlo
DP	Dirichlet process
DPMM	Dirichlet process mixture model
HDP	Hierarchical Dirichlet process
CNN	Convolutional Neural Networks
MLP	Multilayer perceptron
DNN	Deep neural network
OCR	optical character recognition
ASR	automatic speech recognition
DBSCAN	Density-based spatial clustering of applications with noise

a combination of coordinates and time stamp like $Tr_i = (x_i, y_i, t_i)$, as shown in Fig.2.1. In some specific circumstances, other properties relevant to object movement are added, such as velocity, direction, acceleration or geographic information [153] [154].

Different from GPS devices, which record position information of trajectory data only, trajectory data also can be generated from image data or video data. In some papers, the interest points are located as initial points of trajectories, and models are used to track the interest points in the following images. As

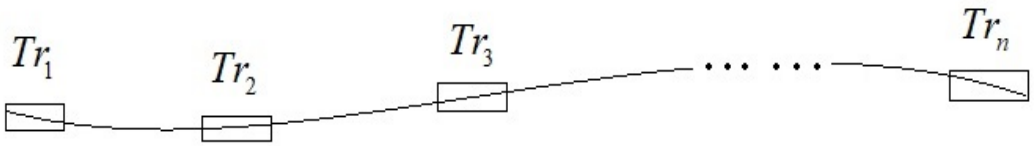

Figure 2.1: Trajectory generated by GPS tracking devices

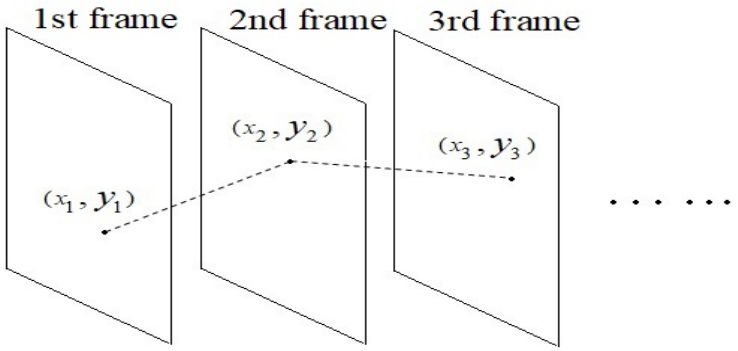

Figure 2.2: Trajectory generated from camera device

shown in Fig.2.2, for image data, a sequence of pixels in consecutive frames form up a trajectory, which is similar to optical flow [18] [138]. Furthermore, scale-invariant feature descriptors are employed to track the points in video data set as well [60] [130]. With the trajectory data generated from images or videos, spatiotemporal and image information including pixels or scale is employed. On the other hand, the semantic trajectory data are attracted more and more attention recently [28] [164], because they contain more information to improve classification accuracy, and can be used directly and hence save more time.

Therefore, the aim of this study is to generate a trajectory by comparing and matching image patches or pixels from two image points in consecutive frames [113]. Several scholars have already developed trajectory methods along these lines: the iterative closest point algorithm [25], the robust point matching algorithm [29] and more popular method, optical flow algorithm [54] which is shown in Fig.2.3 and Fig.2.4. These methods can robustly track points in video data, but they cannot extract characteristic points, i.e., unique object representations represented as trajectories. Further, other factors may influence trajectory

generation, such as occlusion, changes in scale, and illumination. To overcome these issues, pixel-level feature detection and tracking methods are needed.

In this study, SIFT algorithm is presented as a solution for detecting and

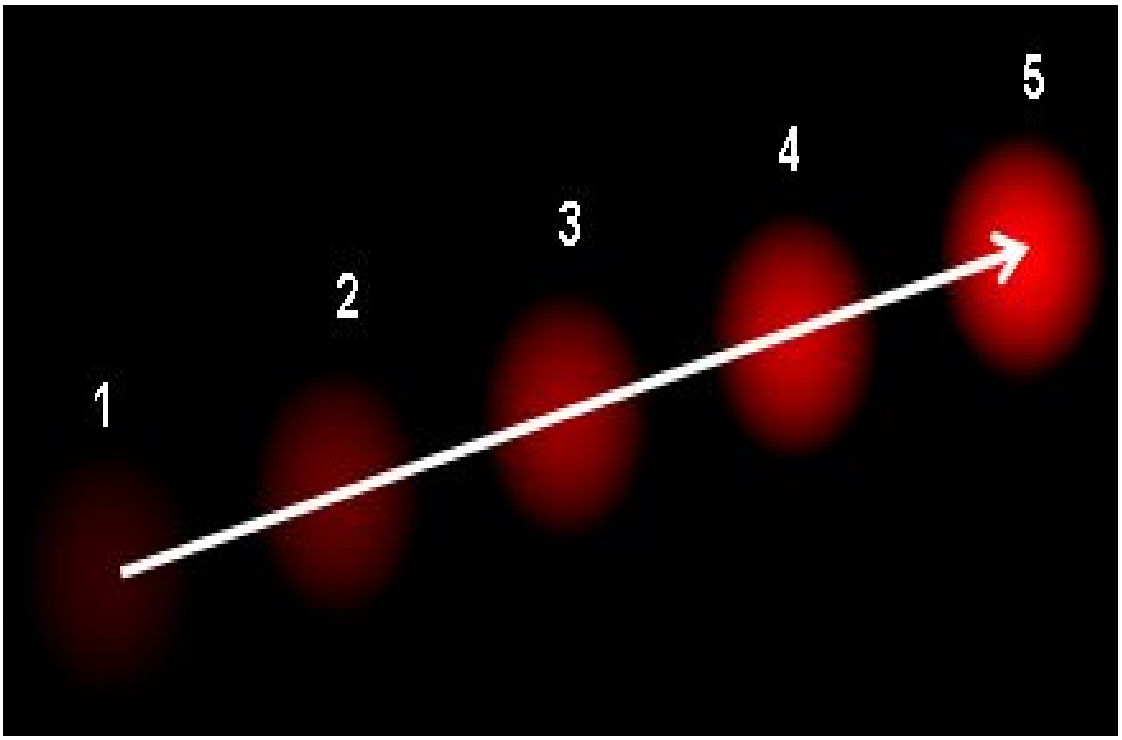

Figure 2.3: Optical Flow algorithm: Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera

characterizing local features in an image [87] [145], because SIFT is a feature detection algorithm to detect and describe local feature of image data. It identifies and tracks a point of interest for each object using a feature description. Hence, a feature extracted from a training frame can then be detected in the query frame, whether or not the scale, noise, or illumination of the image changes. Another advantage of SIFT algorithm is its ability to track the relative positions between points of interest by tracking the feature points in consecutive frames. The experiments in later chapters prove this is an efficient method for generating trajectories.

To ensure the SIFT algorithm performing, the SURF algorithm is used to

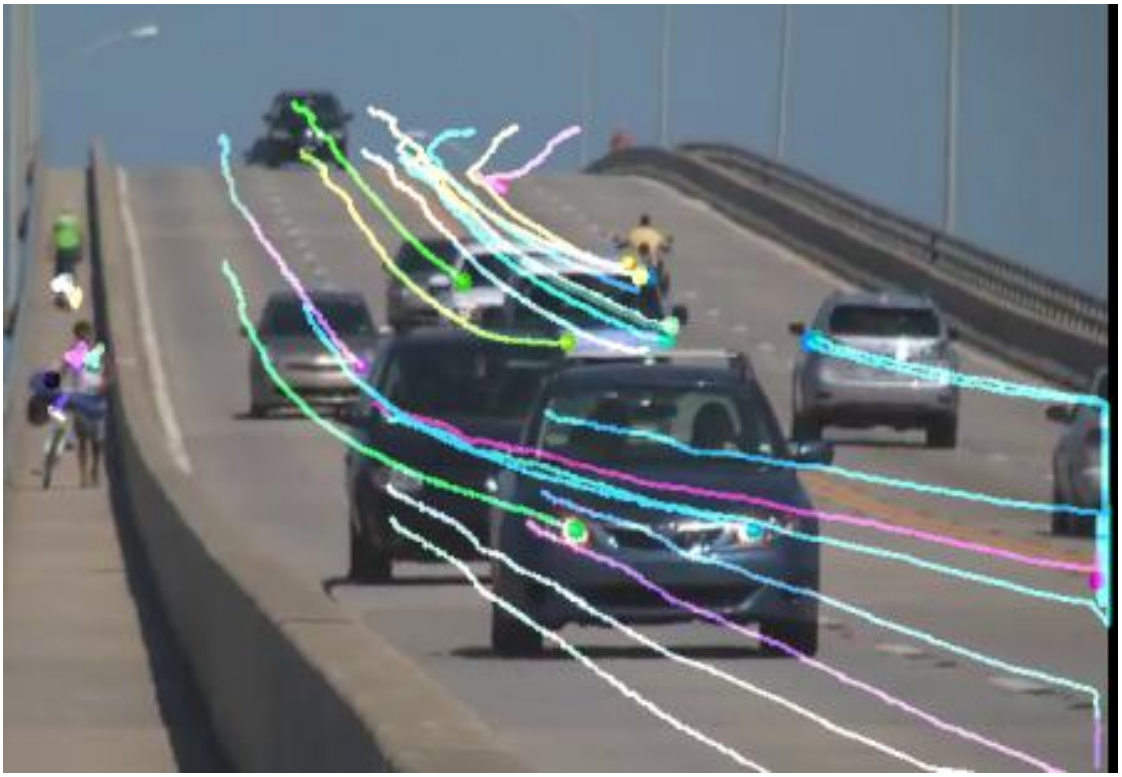

Figure 2.4: Optical Flow algorithm in real world application

locate the characteristic points. SURF is a patented local feature detector and descriptor and has been widely used in object recognition, image registration, image classification, and 3D reconstruction [11] [50] [115]. Although the two algorithms are similar, SURF is only used for point detection in the generation step, as it is faster and more robust than SIFT in the applications explored in this study. Hence, the first step is to use SURF to detect the characteristic points in one object and represent them as multiple characteristic points. Once extracted, SIFT tracks those points, connecting consecutive frames in order, to generate the trajectories. In addition, the method relies on 3D world coordinates since these data contain spatiotemporal information, which improves clustering performance.

However, given the goal is to identify the type of object the trajectory data is describing, more characteristic information is needed. For example, measuring velocity helps to distinguish vehicles from pedestrians. Hence, a more appropriate method for generating trajectory data would record the most useful information

for generating trajectory data from video footage.

2.2 Background of Feature Extraction

In the field of trajectories, representation methods describe trajectory data according to its properties. There are many ways to measure the properties of trajectory data, such as directly through a distance measurement [98], with PCA model [8] [9] which is an orthogonal transformation to represent data by a set of principle components or DBSCAN [59] [139] which a data clustering algorithm, or through a range of other algorithms. However, these methods tend to deal with a limited amount of information and therefore only capture a few properties. Moreover, most of these models are used to recognise outliers in trajectory data [108], as shown in Fig.2.5.

Furthermore, trajectories are presented in different lengths at most circum-

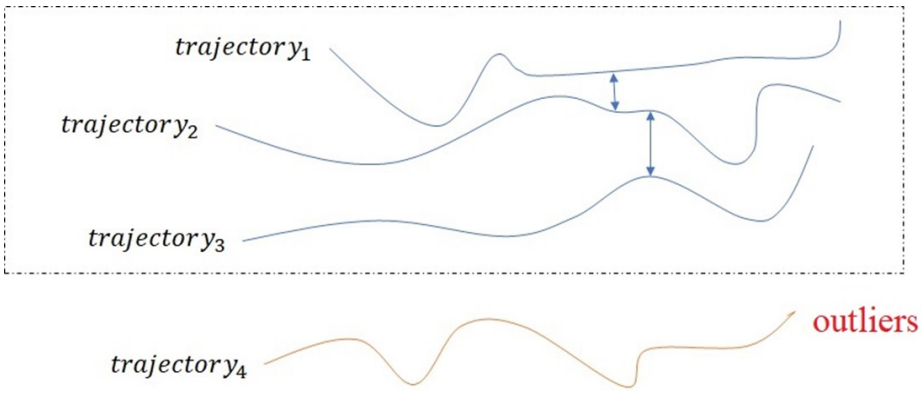

Figure 2.5: The outlier is separated from normal trajectory.

stances. Therefore, one of the goals of this study is to develop a method of trajectory generation that can characterize trajectory data with spatiotemporal information and describe an object's motion status, as well as overcome issues associated with length in Fig.2.6.

In trajectory feature extraction, there are typically many features available to choose. Some features are simple to extract, such as vector field [38], which describes the vector and direction of object, or curving fitting [158], which fits a

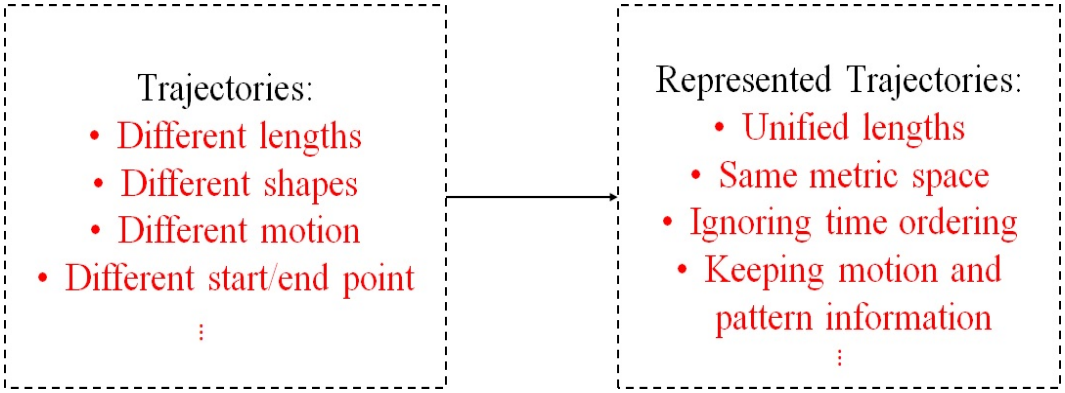

Figure 2.6: The issues need to be fixed in characterizing trajectory data.

trajectory using a fixed number of parameters and representations of several sub-trajectories [8] [9] [76], because segmenting a trajectory can reveal spatiotemporal information. However, clustering and identifying a trajectory is more difficult, particularly with limited information.

To address this challenge, the method presented in this study extracts two different features from trajectory data; one feature represents the tracking information, the other represents motion information. DFT algorithm is used to collect samples from the frequency domain. Specifically, the original trajectory data in the time domain are treated as signals of different lengths with unique characteristics. Then, sine and cosine functions or complex sinusoids can be used to calculate the amplitude and frequency coefficients in the frequency domain. From the opinion of digital signal processing, trajectory data are generated by a function, and the function being any quantity or signal that varies over time. Further, using a fixed number of parameters could be an efficient way of representing trajectory data of various lengths after the DFT process. With this approach, the unique feature types in the trajectory data could be compared in the same space. For instance, Zavarehei and Vaseghi [157] used a DFT to analyse voice trajectories in this way, while Naftel and Khalid [97] used a DFT to learn motion trajectories. DFT has also been used to extract one or more features for clustering trajectory data [56].

That collaborating the coefficients of DFT allows more information related to an object's motion to be collected. However, this technique created a new prob-

lem objects that appear to be moving when they should be static. The trajectory data are recorded when the ego-platform moves while recording. Therefore, several papers have incorporated relative motion into their methods [60] [136].

2.3 Background of Trajectory Clustering

According to the availability of labeled data, trajectory clustering methods are divided in three categories: unsupervised, supervised, semi-supervised. Unsupervised models aim at clustering data without human experts supervision or labeled data. An inference function has been drawn by analyzing unlabeled data sets [35] [38] [143] [148]. Supervised models are learned prior to trajectory clustering. Furthermore, with training data set, supervised clustering models are classification models, and semi-supervised classification models as well. However, all trajectory classification and clustering models are called trajectory clustering models here. Generally, labeled data are used to learn a function mapping data to their labels, i.e. clusters. The clusters of unlabeled data are predicted by this function, then [44] [156] [146] [26]. Labeling data need a heavy burden of manual works by human experts. It is unfeasible for large data sets. Semi-supervised compromises the previous two types of models. It is trained by labeled data and tuned by unlabeled data [48] [141] [156].

The rest of this section is organized as follows. Preliminary works are introduced as follow, and then the models based on unsupervised algorithms are described. A description of the models under supervised algorithms are presented in the following. Finally, I discusses some models based on semi-supervised algorithms. Conclusions are made in the last.

2.3.1 Preliminaries

2.3.1.1 Trajectory Clustering Preparation

In some clustering models [56] [97] [120] [158], trajectory data are required to be set as a unified length so that they could be measured. However, as shown in Fig.2.7, for two arbitrary trajectories, their lengths may be largely different from

each other. Therefore, representing trajectories in a unified length with little loss of information is a major preliminary work of these models. This procedure is called clustering preparation.

For some methods, original data are represented in other space with the same

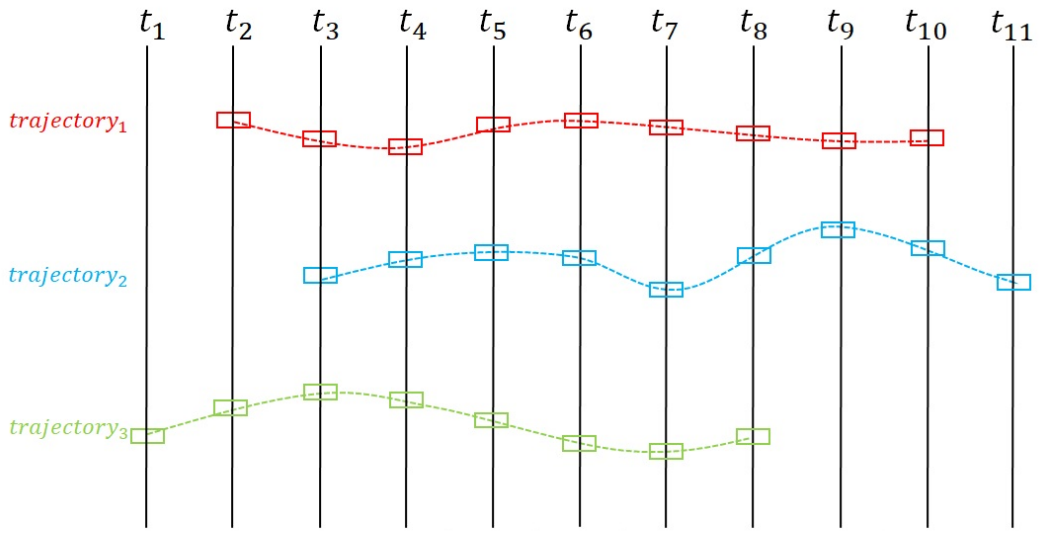

Figure 2.7: For arbitrary trajectory data set, the lengths of trajectories are different from each other.

length. For instance, trajectory data are projected into a subspace [55]. Linear transformation algorithms aim at representing trajectory as a combination of basis trajectories [2]. Curve fitting is another method to approximate trajectories by a parameterized quadratic curve [158]. In order to distinguish similar curves, the direction of the last trajectory point is chosen as an additional parameter. In [120], trajectory data are approximated by a uniform cubic B-spline curve, so that a representation capable of encoding both the shape and the spatiotemporal profile of trajectory data is obtained. In addition, the lengths of trajectories are added to distinguish the trajectories with similar shapes. According to the fact that trajectory data contain a lot kinds of positional information, such as coordinates, speed and directions, vector fields are employed to represent trajectory data [38]. Vector fields give trajectory a smooth streamline and induce a notion of similarity of trajectories. PCA is a statistical procedure to compute a set of linearly uncorrelated variables called principal components by orthogonal

transformation. To avoid partially extracted information, a number of organized segmentations substitute for the corresponding trajectory in [8] and [9]. The time ordering data are transformed and represented in frequency domain by DFT, so a trajectory can be represented as a fixed length vector comprised of Fourier coefficients in [97] and [56]. In [60], the interaction of trajectories are encoded and set as elements of codebook, so camera motion is ignored and the model's robustness is improved.

Re-sampling methods choose trajectory points by sampling rule to unify trajectory lengths. Trajectory data are segmented as sub-trajectories, and all of them are re-sampled to a fixed length so that sub-trajectories are arranged as matrix [9]. In a complex scene such as hand writing data set, Equidistant sampling fixes the problem that two same characters are recorded in different temporal sequence because of different writing speeds [116]. Since re-sampled trajectory points are discontinuous, it is critical that normalization should be involved after re-sampling [84]. It has been widely acknowledged that re-sampling method causes information loss [109]. Therefore, sparsity regularization is used in [19], [33], [101] and [140].

Sub-trajectories hold partial and hidden information of original trajectory data [56] [76], so they are put together and describe trajectory with more flexibility. For instance, the latent motion rule beneath hurricane trajectories is figured out and a certain hurricane trend chart is printed by analyzing sub-trajectories of past hurricane trajectories in [38]. Sub-trajectories also lead to simplified trajectories which represent trajectory data as some smaller, less complex primitives suitable for storage and retrieval purposes [3]. In [150], sub-trajectories are generated by pre-defined policies based on facility performance, time range or distance range. In [8] and [9], trajectory is segmented at the so-called changing points at which direction or speed changes dramatically. Curvature describes direction information, and it could be extracted if a trajectory is treated as a curve by connecting consecutive trajectory points. Curvatures are computed by transforming 3-dimensional position coordinates of points into spherical system and quantized as *up, down, left, right* [36], then a trajectory is segmented at the points where curvature changes. In addition, MDL principle traces the sub-trajectories to estimate trajectory motion by minimizing the differences between sub-trajectories

and the corresponding trajectories in [76]. MBR separates trajectories under occlusion and optimize the inter-object separability [3]. It is an algorithm optimizing the bounding rectangles containing sub-trajectories to ensure that the distance between two rectangles are closer than the distance of trajectories.

Some specific regions of surveillance area hold special semantic information and attract more attention so Regional Segmenting method is implemented. The whole scene is split into several regions and boundaries of the regions segment trajectories [160]. As independent motion pattern, sub-trajectories characterize more information while original trajectory presents limited information.

Some specific regions of surveillance area hold special semantic information. Thus, the points inside the special regions are used to represent trajectory or scene in [129] and all these points are called Points of Interest (POI). The points outside the regions are ignored because they are short of useful information. For instance, activity analysis is a key part in surveillance application to seek low-level situational awareness by understanding and characterizing behaviors of objects in the scene [95], so it is critical to extract POI in the special regions. In topographical map, POI inside the special regions are represented as a single node. For example, two types of POI are introduced in [95] where the first one is the points in entry/exit zones and the second one is the points at the scene landmarks that objects intend to approach, move away or stay for a long time. Except for the special areas, points are represented by a node if their speed are less than a threshold in [13] and [96]. The importance of points can be measured and high-scored ones are selected in [169]. For video data, POI are obtained by Pyramid Representation [136]. In addition, optical flow is another popular implementation by estimating trajectory motion in [39] and [138].

In image frames, more robust and representative features are needed rather than only positional information of trajectory points in [60] and [136]. In [136], histograms of oriented gradients (HOG) and histograms of optical flow (HOF) features are used to describe static appearance information and local motion information of trajectories, respectively. HOG feature computes orientation information to keep scale-invariant property of tracking point and it is fast to implement [60] [66] [71] [88] [89] [144]. Furthermore, SIFT descriptor represents image patch around tracking point [125] [130] [132] [134] [137], and computes

scale and orientation information of image patches to localize tracking object in consecutive frames. As a feature extraction method , KLT tracker is used to find trajectory points and SIFT is applied to represent them [130]. In [134], Difference-of-Gaussian (DOG) detector is used to detecting trajectory points instead of KLT in [130].

2.3.1.2 Common Distance Measurements

Essentially, trajectory are allocated into cohesive groups according to their mutual similarities. An appropriate metric is necessary [7] [94] [159].

Euclidean Distance: Euclidean distance requires that lengths of trajectories should be unified and the distances between the corresponding trajectories points should be summed up,

$$D(A,B) = \frac{1}{N}\sum[(a_n^x - b_n^x)^2 + (a_n^y - b_n^y)^2]^{\frac{1}{2}}, \tag{2.1}$$

where a_n^x and a_n^y indicate the nth point of trajectory A on Cartesian coordinate. N is the total number of points. Euclidean distance is used to measure the distance of trajectories in [98].

Hausdorff Distance: Hausdorff distance measures the similarities by considering how close every point of one trajectory to some points of the other one, and it measures trajectories A and B without unifying the lengths in [85] [22],

$$D(A,B) = \max\{d(A,B), d(B,A)\}, \tag{2.2}$$

$$\begin{cases} d(A,B) = \max\limits_{a\in A}\min\limits_{b\in B}||a-b|| \\ d(B,A) = \max\limits_{b\in B}\min\limits_{a\in A}||b-a||, \end{cases} \tag{2.3}$$

Bhattacharyya Distance: Bhattacharyya distance measures how closely of two probability distributions. In [79], it is employed to measures similarities of quantized directions of points,

$$D(A,B) = -\ln(BC(A,B)), \tag{2.4}$$

where $BC(A,B) = \sum_{t=1}^{T} \sqrt{a_t \cdot b_t}$ and it is used to measure the separability of A and B. a_t and b_t are quantized directions.

Frechet distance: Frechet distance measures similarity between two curves by taking into account location and time ordering. After obtaining the curve approximations of trajectories A and B, their curves map unit interval into metric space S, and a re-parameterization is added to make sure t cannot be backtracked. Frechet distance is defined as

$$D(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \{d\left(A(\alpha(t)), B(\beta(t))\right)\}, \tag{2.5}$$

where d is distance function of S, α, β are continuous and non-decreasing re-parameterization.

Dynamic Time Warping (DTW) Distance: DTW is a sequence alignment method to find an optimal matching between two trajectories and measure the similarity without considering lengths and time ordering [10] [118].

$$W(A,B) = \min_{f} \frac{1}{n} \sum_{i=1}^{n} ||a_i - b_{f(i)}||_2, \tag{2.6}$$

where A has n points and B has m points, all mappings $f : [1,n] \rightarrow [1,m]$ should satisfy the requirements that $f(1) = 1,\ f(n) = m$ and $f(i) \leq f(j),\ for\ all\ 1 \leq i \leq j \leq n$.

Longest Common Subsequence (LCSS) Distance: LCSS aims at finding the longest common subsequence in all sequences, and the length of the longest subsequence could be the similarity between two arbitrary trajectories with different lengths. The distance $LCSS_{\epsilon,\delta}(A,B)$ is written as

$$LCSS_{\epsilon,\delta}(A,B) = \begin{cases} 0,\ if\ A\ or\ B\ is\ empty \\ 1 + LCSS_{\epsilon,\delta}(Head(A), Head(B)), \\ \quad if\ ||a_N - b_M|| < \epsilon\ and\ |N-M| < \delta \\ max(LCSS_{\epsilon,\delta}(Head(A), B), \\ \quad LCSS_{\epsilon,\delta}(A, Head(B))),\ otherwise, \end{cases} \tag{2.7}$$

where $Head(A)$ indicates first $N-1$ points belonging to A and $Head(B)$ denotes first $M-1$ points of B. Finally, $D(A,B) = 1 - \frac{LCSS_{\epsilon,\delta}(A,B)}{\max(N,M)}$.

Other distance types: In [75] [76] [82], more other distance types are proposed to consider more properties such as angle distance, center distance and parallel distance. Angle distance is defined as

$$d_{angle}(L_i, L_j) = \begin{cases} ||L_j|| \times \sin(\theta),\ 0^o \leq \theta \leq 90^o \\ ||L_j||,\ 90^o \leq \theta \leq 180^o, \end{cases} \tag{2.8}$$

where θ is the smaller intersecting angle between L_i and L_j. For center distance,

$$d_{center}(L_i, L_j) = ||center_i - center_j||, \tag{2.9}$$

where $d_{center}(L_i, L_j)$ is the Euclidean distance between center points of L_i and L_j. And parallel distance is

$$d_{parallel}(L_i, L_j) = \min(l_1, l_2), \tag{2.10}$$

where l_1 is the Euclidean distances of p_s to s_i and l_2 is that of p_e to e_i. p_s and p_e are the projection points of s_j and e_j onto L_i respectively.

Distance metrics are used in much more fields relating to trajectories clustering, e.g., density clustering [4] [17] [75] [76] [102]. It is critical to choose an optimal distance according to the scene. For instance, LCSS distance is proved to provide outperforming performance without concerning trajectories length [94]. Hausdorff distance aims at finding the minimum distance between two trajectories and ignore time-order in data. A comparison of distance is listed in Table I,

2.3.2 Unsupervised Algorithms of Trajectory Clustering

Unsupervised algorithms infer a function to describe internal relationships between unlabeled data. Clustering is the method to draw this hidden structure, and some models relating to trajectory clustering are reviewed such as Densely Clustering models, Hierarchical Clustering models and Spectral Clustering models.

Table 2.2: Summary of common distance measurements

Measurement types	Unifying lengths	Computational complexity
Euclidean distance	Yes	$O(n)$
Hausdorff distance	No	$O(mn)$
Bhattacharyya distance	Yes	$O(n)$
Frechet distance	No	$O(mn)$
LCSS distance	No	$O(mn)$
DTW distance	No	$O(mn)$
other distance types	No	$O(1)$

2.3.2.1 Densely Clustering Models

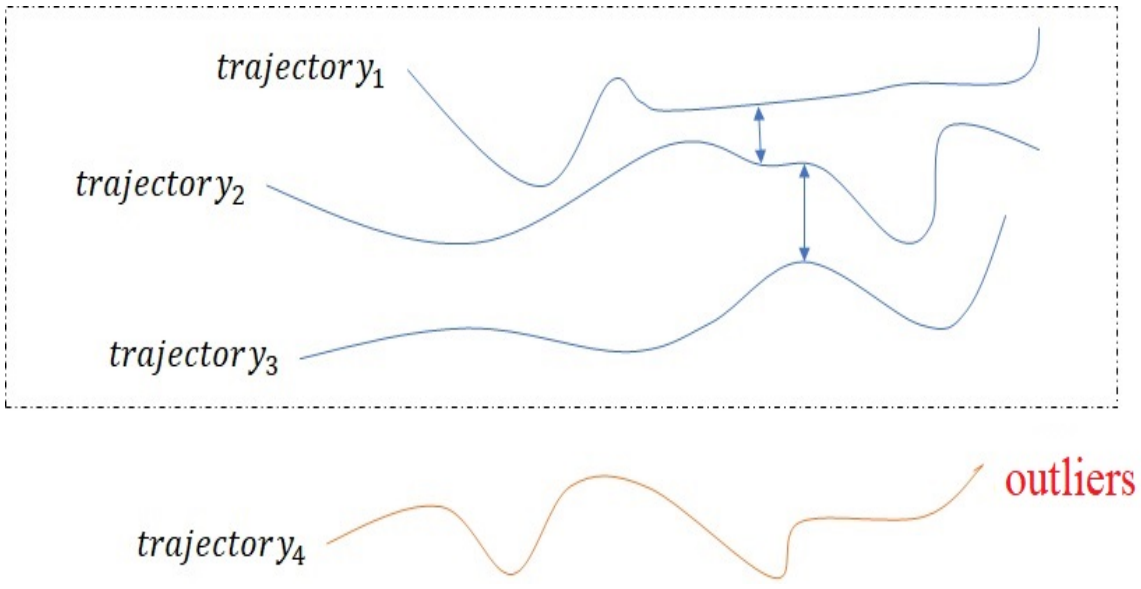

Figure 2.8: DBSCAN for trajectory clustering

Given the centroids, the closely points are packed together and this procedure is called densely clustering. Inspired by this idea, DBSCAN is proposed in [35] and shown in Fig.2.8. A simple presentation is shown in Fig.2.9. In DBCSAN, point p is chosen as the core point and distance threshold ϵ is given in advance.

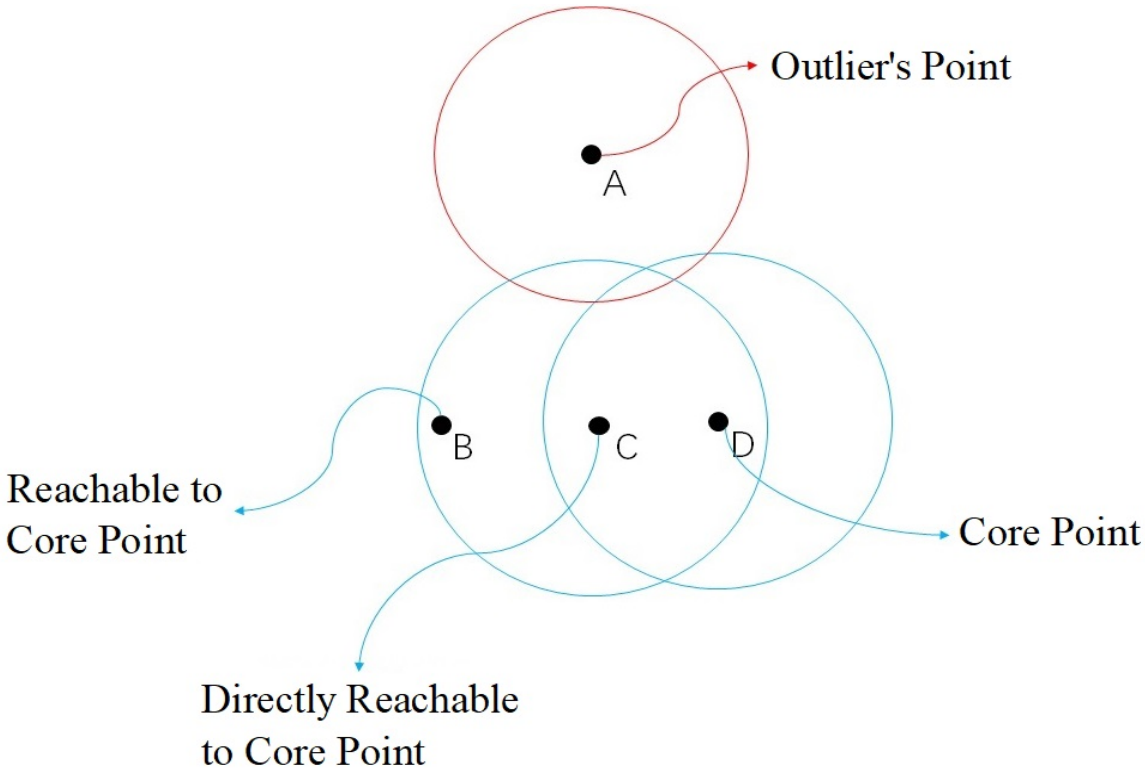

Figure 2.9: DBSCAN

The points inside circle of which the radius is ϵ and the center is p are called *directly reachable* to p. Furthermore, points $\{q_1, q_2, ..., q_n\}$ are *reachable* to p if there is a path that q_1 is *directly reachable* to p and each q_{i+1} is *directly reachable* q_i [59] [139]. Other points are the outliers. Thus, the distance metric and the core parts selection are important. For solving the problem that DBSCAN cannot cluster the trajectories with large differences in densities [64] [75] [76], all trajectories are partitioned and substituted by sub-trajectories, then sub-trajectories are clustered and all clusters are grouped at the last step. However, different from measuring distance by Euclidean distance in [76], the distance is measured by a combination of angle distance, center distance, parallel distance with equal Iight in [64] and [75]. The core trajectories are computed from the clusters and used for classifying new coming trajectory in [30], [75], [166] and [167], e.g., all trajectories points belonging to same cluster are averaged as a new point at each time, and all averaged points form the representations of clusters [75]. In an adaptive

multi-kernel-based method, shrunk clusters represent all groups by considering the attributes including positions, speeds and points, which retains much more discriminative messages in [149].

Besides DBSCAN, there are some other models belonging to Densely Clustering models cluster trajectory data. K-means clusters trajectories by searching centroids of clusters repeatedly [38] [42] [57] [92] [96] [126]. For improving the performance of K-means, EM algorithm is implemented to solve optimization problem iteratively [170], because EM is an iterative method to find maximum likelihood or posteriori estimates. Due to the issues such as data imprecision and complexity of large data sets, a trajectory may belong to multiple clusters so EM is used to classify them [65]. FCM algorithm, which is a fuzzy clustering algorithm, employs parameters to measure the level of cluster fuzziness for each trajectory which called fuzzifier. The algorithm searches correct direction in each iteration for cluster trajectories [104] [105] [121].

2.3.2.2 Hierarchical Clustering Models

Hierarchical Clustering models help to understand trajectory by multiple features, so this tree-type construction is proper to implement. Hierarchical Clustering models generally fall into two clustering types, Agglomerative and Divisive. As shown in Fig.2.10 and Fig.2.11, two hierarchical types are also known as "bottom-up" and "top-down" approaches.

In Agglomerative frameworks, trajectories are grouped and the similar clusters are merged by searching their common properties. Optimal classifications are obtained by repeating representation computation and clusters merging until meeting the requirements. Inspired by this idea, Agglomerative clustering models Ire explored in [168] to mine the locations that users are interested, HITS model is proposed to achieve this goal and movement tracks of users are recorded as trajectories. Top n interesting trajectory clusters are obtained iteratively and the most popular locations are generated.

Different from Agglomerative, Divisive frameworks cluster trajectory data into groups and split them recursively to reach the requirements. Following this frame-

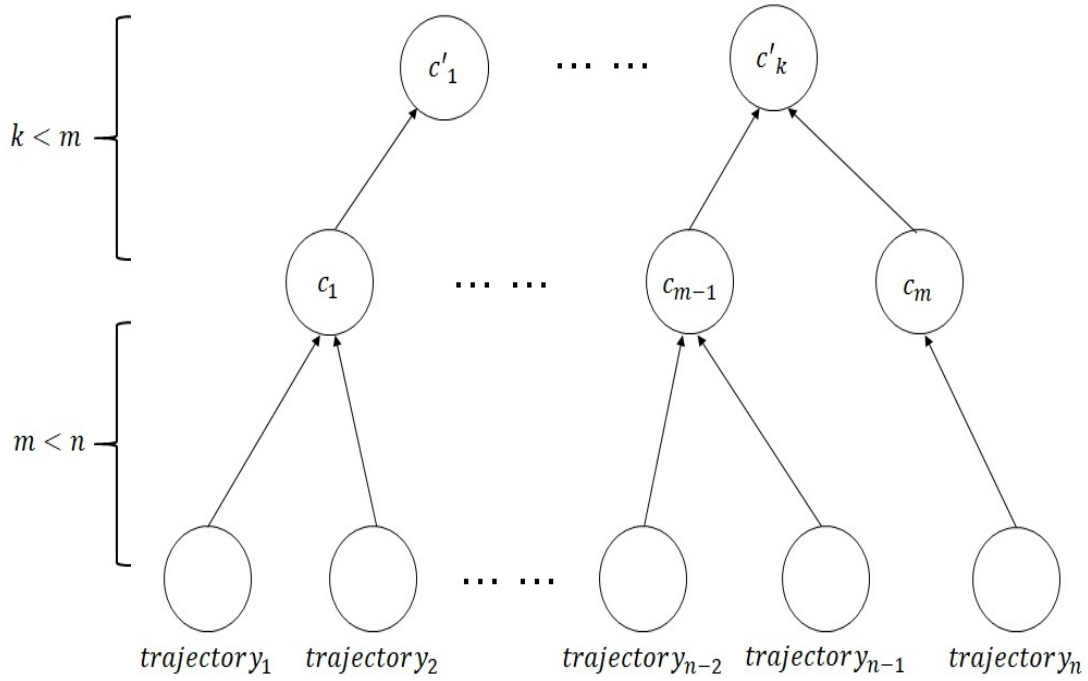

Figure 2.10: Hierarchical clustering models

work, trajectory data are characterized by direction feature and clustered by Dominant-set embedded Bhattacharyya distance in initial clustering stage [79]. In each cluster, trajectories are split further except for the ones holding similar positions. Because of the good performance of iterative models, TAD model is proposed [165] which is a Divisive framework detecting all the closed trajectories firstly and splitting them recursively. More attributes of trajectory points are considered to improve the performance in [143]. For instance, trajectory $A = \{a_1, a_2, \cdots, a_n\}$ where $a_i = \langle x_i, y_i, \beta_i \rangle$. It is comprised of 2-dimensional position and an additional attribute β such as velocity or object size. In the coarse clustering step, the distance measurement between trajectory A and its nearest observation trajectory B are shown as follows,

$$f(A,B) = \frac{1}{N_A} \sum_{a_i \in A} ||(x_i^a - x_{\psi(i)}^b, y_i^a - y_{\psi(i)}^b + \gamma d(\beta_i^a, \beta_{\psi(i)}^b))||, \tag{2.11}$$

where $\psi(i) = \arg\min_{j \in B} ||(x_i^a - x_j^b, y_i^a - y_j^b)||$ and the minimum distance value is counted as the distance between A and B. N_A is the total number of points

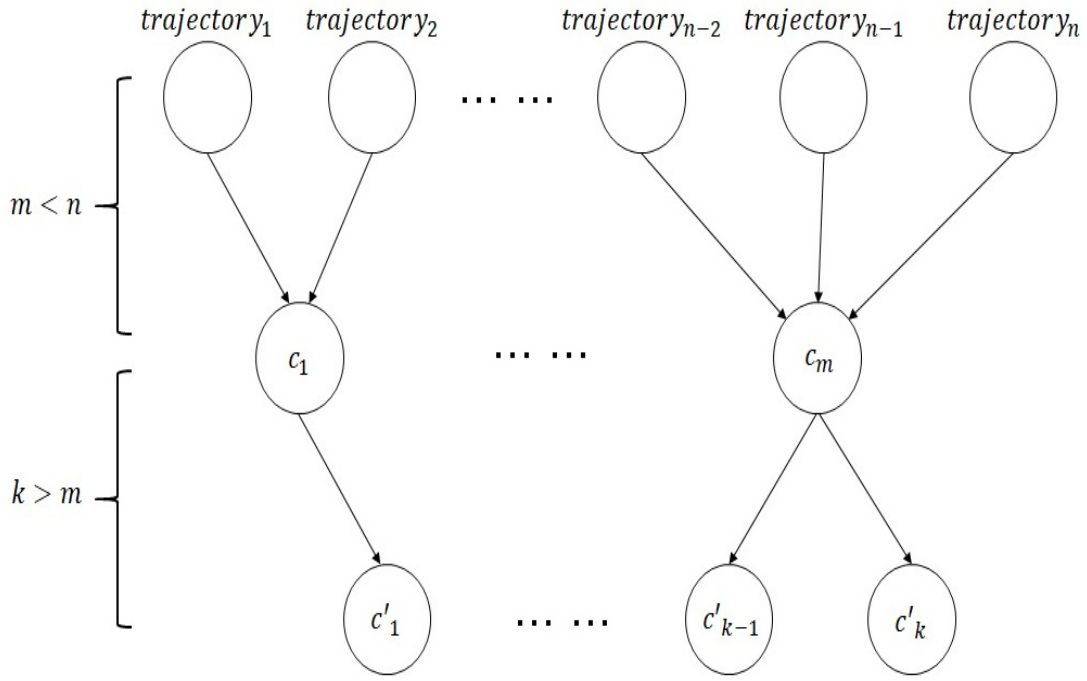

Figure 2.11: Hierarchical clustering models

belonging to A, $d(\beta_i^a, \beta_{\psi(i)}^b)$ indicates the dissimilarity of A and B, and γ is Iight parameter. In the fine-clustering stage, the model aims at distinguishing distortions by considering directed similarity $S_{A\to B}$ and confidence $C_{A\to B}$

$$S_{A\to B} = \frac{\sum_{a_i\in A} c(a_i, b_{\psi(i)}) s(a_i, b_{\psi(i)})}{\sum_{a_i\in A} c(a_i, b_{\psi(i)})}, \tag{2.12}$$

$$C_{A\to B} = \frac{\sum_{a_i\in A} c(a_i, b_{\psi(i)})^2}{\sum_{a_i\in A} c(a_i, b_{\psi(i)})}, \tag{2.13}$$

where $c(a_i, b_{\psi(i)}) = \exp(\frac{-||(x_i^a - x_{\psi(i)}^b, y_i^a - y_{\psi(i)}^b)||}{\sigma_1})$ and $s(a_i, b_{\psi(i)}) = \frac{\exp(-d(\beta_i^a, \beta_{\psi(i)}^b))}{\sigma_2}$.

Furthermore, a similar hierarchical framework is explored to group videos by constructing the trajectories of video [41] as an unordered tree, and a kernel method recognizes videos by clustering the trees. In addition, Hierarchical Clustering models also recognize actions from video in [112] and [127]. For two trajectories in video, $a = \{x_1^a, x_2^a, \cdots, x_{t_a}^a\}$ and $b = \{x_1^b, x_2^b, \cdots, x_{t_b}^b\}$, the distance is computed as follows,

$$d(a,b) = \max_{t\in[\tau_1,\tau_2]} d_{spatial}[t] \cdot \frac{1}{\tau_2 - \tau_1} \sum_{t=\tau_1}^{\tau_2} d_{velocity}[t], \tag{2.14}$$

where $d_{spatial}[t]$ is the positional distance at time stamp t, and $d_{velocity}[t]$ is the similarity measurement relative to velocity. An affinity matrix $w(a,b) = exp(-d(a,b))$ is calculated and trajectories are clustered by greedy agglomerative hierarchical models [112] [127]. The clusters are overlapped because of similar parts, so every trajectory is lighted and optimized to classify in [99]. Since one motion object may generate several trajectories, it is critical to employ as much features as possible to ensure object recognition, and a multi-layer classifier is invented in [5] and [79].

2.3.2.3 Spectral Clustering Models

Trajectory data can be represented as a matrix called affinity matrix, and the relationships between them are extracted as the elements of matrix. The top K eigenvectors form clusters with distinctive gaps between them which can be readily used to separate data into different groups [148]. In addition, affinity matrix characterizes videos [128] and represents the relationships. In [58], affinity matrix A is constructed as follows,

$$A_{ij} = \exp[\frac{-\bar{d_{ij}}}{2\sigma^2}], \tag{2.15}$$

where $\bar{d_{ij}} = \frac{1}{n}\sum_{k=1}^{n} ||x_{i,k} - x_{j,k}||$, and $x_{i,k}$ indicates the kth point of trajectory i. Considering different lengths of trajectories, some novel models are explored to construct affinity matrix [15] [16] and it is constructed as

$$A_{ij} = \begin{cases} e^{(-\frac{1}{\sigma_i\sigma_j}||v_i - v_j||^2)}, \ for\ i \neq j \\ 0, \ otherwise, \end{cases} \tag{2.16}$$

where v_i and v_j are points, σ_i and σ_j indicates scale invariance which computed by the median of the l nearest neighbors. In order to increase the separation of points belonging to different groups, SVD decomposition is used to construct the affinity matrix [72]. In addition, a novel distance method is explored to compute

trajectories P and Q [6] so that spatial distinction can be considered.

$$s(P,Q) = e^{-\frac{1}{2}h_\alpha(P,Q)h_\alpha(Q,P)/(\sigma_p\sigma_Q)}, \tag{2.17}$$

$$h_\alpha(P,Q) = ord^\alpha_{p\in P}\left(\min_{q\in N(C(p))} d(p,q)\right), \tag{2.18}$$

where $h_\alpha(P,Q)$ is the directed Hausdorff distance, $ord^\alpha_{p\in P}f(p)$ indicates the value of $f(p)$ and $N(C(p))$ denotes the subset of points which the ones matching to the point p in trajectory P.

For clustering high dimensional trajectory data by Spectral Clustering models, several novel methods are explored in [21], [53] and [161]. For example, a mixture of affinity subspaces is applied to approximate trajectory in [21], and a new similarity metric captures causal relationships between time series in [53]. Trajectory data are represented by considering covariance features of trajectories in [34], so it avoids considering different lengths of trajectory data. Spectral clustering works with multiple-instance learning frameworks to achieve human action recognition in [152].

Spectral Clustering models are derived from Graph Theory in which an undirected graph represents the relationships and constructs a symmetric adjacency matrix presenting them [14]. By constructing a graph, both explicit and implicit intentions inside trajectory data are mined [23]. The graph is cut into sub-graphs to classify trajectories, and each sub-graph represents its own cluster [83] [158]. Hierarchical layers search sub-clusters in each cluster by treating trajectories points as graph nodes and this procedure is called *Hierarchical graph partitioning* [47]. For considering more variables, a novel measurement function comprised of the entropy rate of a random walk on a graph is presented in [85]. From the idea that an undirected graph can be represented as an adjacent matrix, a directed graph also can be involved [81]. Trajectory Binary Partition Tree (BPT) represents video in [103] by representing trajectories as nodes so the edges indicate relationships between a pair of trajectories, and graph cut method groups trajectory data. Because of the robustness of composite feature descriptors, the descriptors including SURF and Maximally Stable Extremal Regions (MSER) are employed in [83]. An object creates several trajectories if different parts of

the object are tracked, so a model is invented to describe trajectories by feature patches [86]. The edges are computed by geometric distance and appearance distance. Hausdorff distance is utilized to measure the similarities and set as Iights of edges in [62]. Since the great performance of PageRank, it is used to score the edges in [27], too.

2.3.2.4 Discussion

Densely Clustering models classify trajectories by distance metrics mostly, which may result in classifying trajectory data by spatial information. Hierarchical Clustering models fix this problem by considering more attributes in each level. However, this operation cost much more time in computation. Spectral Clustering models compute internal relationships by analyzing the affinity matrix, and it saves much more computational resource by processing all trajectory data together. However, [63] mentions that Spectral Clustering models have their own limitation that they are pre-defined only for the non-negative affinities between trajectories. Furthermore, that trajectory lengths are required to be unified is another issue of applying Spectral Clustering models.

2.3.3 Supervised Algorithms of Trajectory Clustering

Supervised algorithms aims at learning a function which determines the labels of testing data after analyzing labeled training data. Therefore, supervised algorithms outperform others and the supervised ones could save much more computation resource. In some supervised algorithms, trajectory data are classified by unsupervised algorithms and the representations of clusters are obtained to classify new inquiry trajectories. For example, in Densely Clustering models, the representations can be computed from the grouped training trajectory data and new coming trajectories are clustered quickly in [8] and [104]. Trajectory data are classified and organized in a tree-construction and new coming trajectories are clustered by searching the tree in [48] and [106].

2.3.3.1 Nearest Neighbor Algorithms

Nearest Neighbor algorithms, such as k-NN algorithm, are finding a voting system to determine the category of a new coming entity and all data are kept in the same feature space. In trajectory clustering, the distances from an inquiry trajectory to all labeled trajectory data are computed, and the label of the inquiry trajectory is voted by its k nearest neighbors. Shown in Fig.2.12, the inquiry trajectory is assigned as blue cluster if $k = 1$ and assigned as red one if $k = 3$.

In the implementation, it is important to choose a suitable distance metric

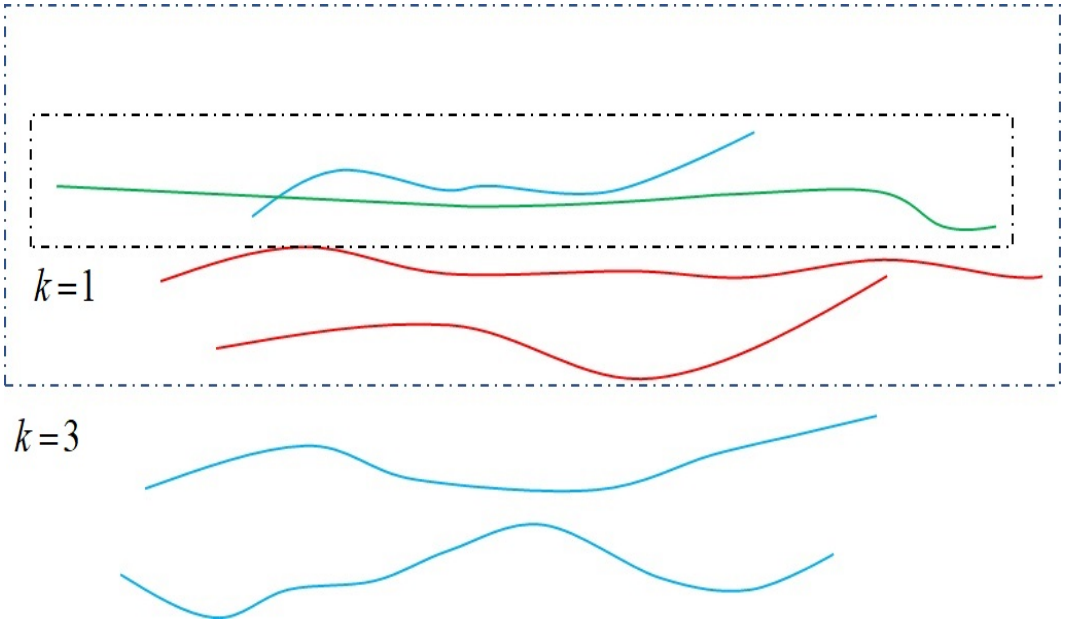

Figure 2.12: k-NN for trajectory clustering. Inquiry trajectory is the green one, the labeled data are the red and the blue ones which means two clusters.

according to the scenario, occlusion, trajectory data source and feature types. Therefore, trajectory data are represented by MBR and classified by k-NN in [44]. It avoids occlusion and increasing inter-object separability. Furthermore, trajectory data are represented in Riemannian manifold [32] so their shapes can be modeled and compared by using an elastic metric. For accessing k-NN faster, fast nearest neighbor (fastNN) algorithm organizes trajectory data in an Octree [110]. With the increasing inquiry trajectories, the trends of trajectory data in a fixed period are required instead of general representation, so a circumstance

that dynamically searches the nearest neighbors in a fixed period or the ones belonging to some specific types is considered in [40]. As a supervised classification method, SVM is trained to generate the hypervolume, and the inquiry trajectory is determined as outliers if it falls outside the hypervolume [108]. Structural SVM is explored to detect social groups in crowds in [124]. Furthermore, SVM works with Graph Theory to cluster trajectories in [122].

2.3.3.2 Statistical Models

Statistical models exploit a set of probability distributions to represent the data generating process such as GMM and Bayesian inference. GMM usually combines with EM algorithm to train each component, and Bayesian inference obtains a set of probability functions which determine the categories of inquiry trajectory data. Bayes' theorem is critical for Bayesian inference and written as $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ where A and B indicate two events in event space.

GMM aims at describing the sample from $\{x_1, x_2, \cdots, x_n\}$ in a component of GMM as

$$P(x_j) = \sum_{i=1}^{K} \pi_i N(x_j; \mu_i, \Sigma_i), \tag{2.19}$$

where $N(x_j; \mu_i, \Sigma_i)$ is the probability density of the ith component belonging to a component with mean μ_i and variance Σ_i. π_i is the Iight with a constraint that $\sum_{i=1}^{K} \pi_i = 1$, and they can be computed according to event frequency. Generally, EM algorithm iteratively optimizes the parameters of GMM, but Maximum Likelihood algorithm is implemented instead of EM if labeled trajectory data are available in training stage. For example, video events are treated as a linear combination of a set of event patterns, and two probabilistic terms are proposed to characterize video events in [156]. Furthermore, the abnormal patterns are scored by summarizing the probabilities of trajectory data of the corresponding video. GMM models the variance caused by the environmental factors and embedded into DTW to recognize gestures [10].

Bayesian inference classifies new coming data, and the classified ones update

the probability functions of Bayesian inference. For samples $\{x_1, x_2, \cdots, x_n\}$, the probability of the corresponding labels $y_{1:n}$ is $p(y_{1:n}|x_{1:n})$. Derived from MCMC algorithm, the distribution of variables can be approximated by a joint distribution, so Gibbs sampling is used to approximate $p(y_{1:n}|x_{1:n})$ by sampling $p(y_i|y_{-i}, x_{1:n})$ iteratively. According to Bayes' theorem, $p(y_i|y_{-i}, x_{1:n})$ is represented as $p(y_i|y_{-i}, x_{1:n}) \propto p(x_i|y_i)p(y_i|y_{-i})$ where $p(x_i|y_i)$ is the likelihood and $p(y_i|y_{-i})$ is the marginal distribution. In DP model which is one of the Bayesian inference frameworks, $p(y_i|y_{-i})$ is formulated as $p(y_i|y_{-i}) \propto \alpha G_0(y_i) + \sum_{j \in -i} \delta(y_i - y_j)$ where α is scale parameter and G_0 is base measure in sample space. The clusters can be parameterized for classifying new inquiry data, e.g., DPMM is used to represent all m clusters with parameterized indexes $\{\Theta_1, \Theta_2, \cdots, \Theta_m\}$ in [56]. Finally, the new inquiry trajectory is classified by a trained DPMM as $p(\Theta_k|R) \propto p(R|\Theta_k)p(\Theta_k)$ where $p(R|\Theta_k)$ is the likelihood and $p(\Theta_k)$ is the prior probability. In order to learn coupled spatial and temporal patterns, HDP algorithm is applied in [135]. Bayesian model is used to segment object by classifying trajectories, so that human motion is also detected [31].

2.3.3.3 Neural Network

Neural network is an artificial system simulating the biological neural network in animal brains. The network is constructed by a number of mutually connected neurons, and each neuron is represented as a real number. Neural networks can represent data such as deep generative model and applied in Computer Version mostly, as shown in Fig.2.13, which is a popular graph presentation. It is trained to represent multivariate time series if trajectory data are generated as a vector [155], and a deep fully-connected Neural Network with Iight decay and sparsity constraint transfers trajectory data from different viewpoints to a fixed viewpoint in compact representation [111].

In most cases, Neural Network is used to classify data. It can be vieId as a mathematical function $f : X \to Y$ where X is the observation and Y indicates the corresponding label. For example, CNN or called ConvNet consists of multiple layers including convolutional, pooling and fully connected layers.

That layout tolerates the variations of the input data, avoids overfitting problem and distinguishes data as similar as MLP. CNN has been proved efficient in clustering issue of computer vision. As Fig.2.13 shown, CNN is comprised by two convolutional layers, two pooling layers, two fully connected layers and one output layer which acts as an image classifier. CNN is employed for trajectory clustering in [26] and [146]. Furthermore, CNN also ranks the trajectory clustering results in [39]. A flexible deep CNN called Deep Event Network is trained by ImageNet data set, and the trained Deep Event Network is tuned to extract generic image-level features of trajectory data in [43]. In order to figure out the differences between image classification and multimedia event detection, DevNet fine tunes parameters by a specific data set, and backward passing is employed to identify pixels in consecutive frames to recount events. DNN is another Neural Network which learns a more compact and powerful representation of trajectories [49]. Furthermore, DNN keeps the structural relationships between trajectories in [119], and mines the relationship between multiple features including spatiotemporal features, audio features and inter-class relationship to classify videos in [61] and [147]. Self-Organizing Map learns the similarities between trajectories in a 2-dimensional grid and each element of the grid indicates a specific prototype in [97] and [117]. In training steps, each training trajectory is trying to find the most suitable prototype in network, and adjust the neighbors of the matched one accordingly.

2.3.3.4 Discussion

Nearest Neighbor algorithm only considers the spatial relationships between a pair of trajectory data but ignores local characters. Statistical model makes up for this imperfection by combining them in a mixture model or inferring the relationships in Bayesian models. Neural Network considers the differences of trajectory data and requires a huge number of data to train it. Though the supervised methods obtain the classifiers by observing a number of training data, overfitting problem may happen when the model overreacts training data.

2.3.4 Semi-supervised Algorithms of Trajectory Clustering

Semi-supervised algorithms fall between unsupervised algorithms and supervised algorithms. The algorithms make use of a small number of labeled data and continuous inquiry data to complete tasks. The model is trained by labeled data firstly, then inquiry data are kept sending to the trained model to make sure that it can be updated to outperform the previous model. Semi-supervised procedure needs only a small cost in terms of human classification efforts. This procedure not only avoids overfitting problem, but also is more accurate than the unsupervised ones.

Therefore, some semi-supervised algorithms are invented from unsupervised or supervised algorithms. For example, trajectory data are classified firstly and the new inquiry ones are clustered to update the classifier automatically [48] [73] [141] [156]. Detected anomaly trajectory data are used to recalculate the representation of anomaly trajectory clusters in [73]. Trajectory data of video are modeled as the combination of normal and abnormal patterns, and probabilistic terms characterize the patterns in [156]. From this modeling, the terms can be updated by the detected inquiry trajectory. In order to detect abnormal trajectories faster in complex scene, low-rank approximation is employed to describe trajectory data and the new detected abnormal ones update the threshold in [141].

Inspired by Hierarchical Frameworks, trajectories and the clusters are represented as a tree where children nodes indicate trajectories and roots denote the representations of the clusters in [69], [80] and [107]. A new cluster is created if no clusters close to the inquiry trajectory. Trajectory T is constructed as a vector of 2 dimensional coordinates $T = \{t_1, \cdots, t_n\}$ where $t_j = \{x_j, y_j\}$. A representation of cluster is computed as $C_i = \{c_{i1}, \cdots, c_{im}\}$, where $c_{ij} = \{x_{ij}, y_{ij}, \sigma_{ij}^2\}$ and σ_{ij}^2 is an approximation of the local variance of the cluster i at time j. The inquiry trajectory is assigned to the nearest cluster and the corresponding cluster should be updated by the new one. For the nearest cluster point $c = \{x, y, \sigma^2\}$ to the point of trajectory $t = \{\hat{x}, \hat{y}\}$, c is updated as following

$$\begin{cases} x = (1-\alpha)x + \alpha\hat{x} \\ y = (1-\alpha)y + \alpha\hat{y} \\ \sigma^2 = (1-\alpha)\sigma^2 + \alpha[dist(t_i, c_j)]^2, \end{cases} \tag{2.20}$$

where α is the update rate between 0 and 1.

Considering the fact that Bayesian model is derived from Bayes' theorem, the parameters are optimized by sampling training data, and it is feasible to update the model by classified new inquiry data [56]. Furthermore, in order to add new trajectory data, the previous samples and the new ones are sampled by Gibbs Sampling as

$$p(\eta_i|\eta_{-i}, y_{1:N+\phi}) = p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new}, y_{1:N+\phi}), \tag{2.21}$$

where y is trajectory data, $\eta_{1:N}$ indicate the known states of the previous samples, and $N+1 < i < N+\phi$. η_{-i}^{new} denote the states of new inquiry trajectory data except for the ith one. From Bayes' theorem, the cluster process is rewritten as $p(y_i|\eta_i)p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new})$. $p(y_i|\eta_i)$ is estimated by the previous samples and it is assumed to be Gaussian distribution. The only issue need to be fixed is carrying out Gibbs Sampling on $\eta_{N+1:N+\phi}$ to compute $p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new})$.

2.4 Conclusion

In this chapter, we reviewed the methods of trajectory classification. According to the fact that trajectory data have a variety of characterizations and data forms, training data have been involved or motion information such as speed value are recorded. Then, different algorithms are required to reach the goal, so they are classified into three categories: unsupervised, supervised and semi-supervised algorithms. Unsupervised algorithms are called as clustering methods, and it can be grouped into three sub-categories: Densely Clustering models, Hierarchical Clustering models and Spectral Clustering models, and Spectral algorithms take better performance in this category. By means of a comprehensive analysis, we found that unsupervised algorithms have the disadvantages of high computa-

tion cost and heavy memory load, though there is no training data requirement or human experts' supervision. Supervised algorithms are divided into Nearest Neighbor algorithms, Statistical models and Neural Network. Furthermore, Neural Network is applied to solving a number of issues including trajectories classification. Although a huge number of training data and plenty of time are needed to understand the construction inside trajectory data by training Neural Network, it is fast on classifying the new coming trajectory and shows robustness, accuracy in the real-time application. Semi-supervised algorithms combine the advantages of both previous algorithms, but validation step and correction step are required to iterate several times. Therefore, it reduces computation time and it is suitable for the scenario that a small number of labeled data is involved.

From the above discussion, some novel models employ more feature and most of them consider spatial information. However, only spatial information presents location, distance from others and traveled distance, the motion trajectory relative to world, direction, motion streaming and some other information are discarded. Therefore, we need to find a way to employ trajectory data directly or transfer them into another representational space. Furthermore, the motion relative to recording device should also be recorded, because a lot of trajectories generating by static objects have similar shape with recording device. According to the above discussion, a novel model is needed to combine two different semantic features, which are continue feature and discrete feature. In the case that big size of data set, small number of categorizes and the independence between arbitrary two trajectories involves here, topic model is employed to implement trajectory clustering and find the hidden "topics". My algorithm flow is shown in Fig.2.14.

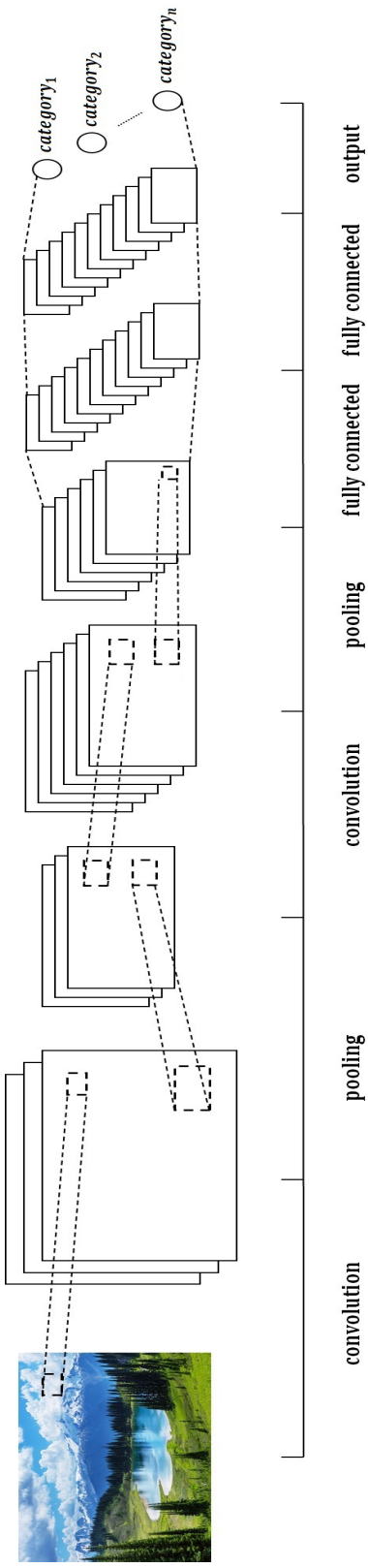

Figure 2.13: CNN is one of the classical models of Neural Network and widely used in images classification

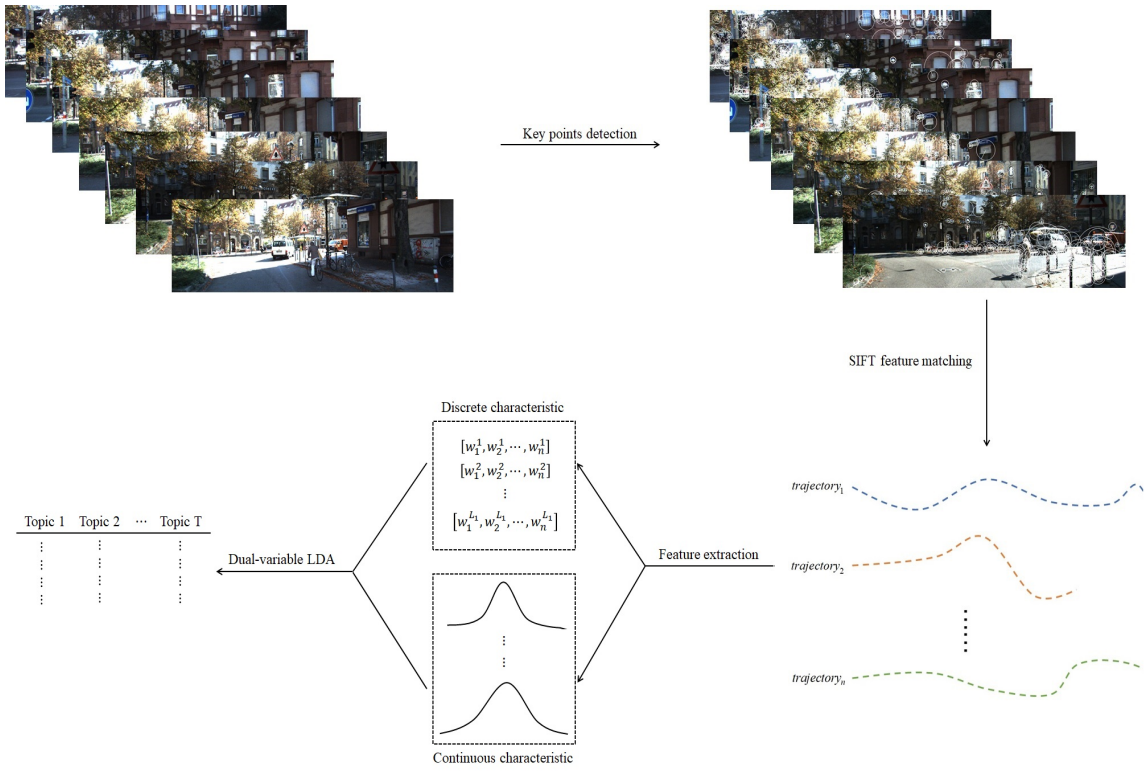

Figure 2.14: Our proposed algorithm clustering trajectories

Chapter 3

Trajectory generation with SIFT

Generating trajectory from video data set impedes the efficiency of clustering approaches. However, most methods record the object tracks by manually tagging. Although the objects are tracked by most advanced methods [37] [90] [91], the the size is different in different frames, and it also influences the accuracy of the clustering. In order to fix the above issues, multiple points are extracted and characterized to represent one single object. Furthermore, the feature extraction and description methods are supposed to be robust in the our algorithm. Therefore, based on the discussion in Chapter 2, we use SURF and SIFT algorithm to locate and track the objects. In doing so, multiple fixed features are extracted and represent single object in the video. In doing so, the points are located and described by SURF in fast and robust characterization, and SIFT tracks the features in consecutive frames. However, 2D image coordinates contain less information than 3D real world coordinates, so we rectify 3D coordinate into 2D image coordinates and compare them with the extracted feature points. Experiments show the our generation method can extract trajectories fast and efficiently.

3.1 Methodology

In this section, we introduce the method that detect and track the characterize points tasks. Furthermore, characterize points are the feature points of objects

and described in feature descriptor, so that is a critical task in our implementation. Firstly, we introduce the SURF algorithm detecting the points. Then, we explain how SIFT algorithm tracks the points and the corresponding 3D coordinates are chosen to represent the trajectories. Finally, we show the experiment results improving the efficient of trajectory generation.

3.1.1 Characterize points detection

Similar but different to SIFT algorithm, some square-shaped filters are employed as Gaussian smoothing. In this way, it faster than SIFT to locate the characterize points. In order to improve the performance of computation accuracy, Hessian matrix $\mathcal{H}(\mathbf{x}, \sigma)$ is used to measure the location and its scale [11]. Given pixel $\mathbf{x} = (x, y)$ of image I, $\mathcal{H}(\mathbf{x}, \sigma)$ of pixel $\mathbf{x}$ is

$$\mathcal{H} = \begin{pmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{pmatrix} \tag{3.1}$$

where $L_{xx}(p, \sigma)$ is the convolution of second-order derivative of Gaussian with image at $\mathbf{x}$, and it is similar to the other elements in equation.(3.1). σ dedicates the scale in the above. In general, we use a box filter of size 9×9, which approximate to a Gaussian second order derivatives with $\sigma = 1.2$ and Fig.3.1 shows how box filter works on an image.

In some methods, Gaussian filters are applied to generate image pyramids, which means repeatedly smoothed with a Gaussian and sub-sampled to achieve the goal. In such a method, Hessian matrix, there is no need to apply Gaussian filters iteratively. Furthermore, the other layers of image are resulted from the filters of size 15, 21 and so on accordingly.

3.1.2 Characterize points tracking

After detecting the characterize points, the next step is tracking the points and consist them into trajectories. A proper descriptor is critical to match the two points belonging to different frames. SIFT algorithm transforms an image into a large set of local feature vectors, and each feature vector is invariant to image

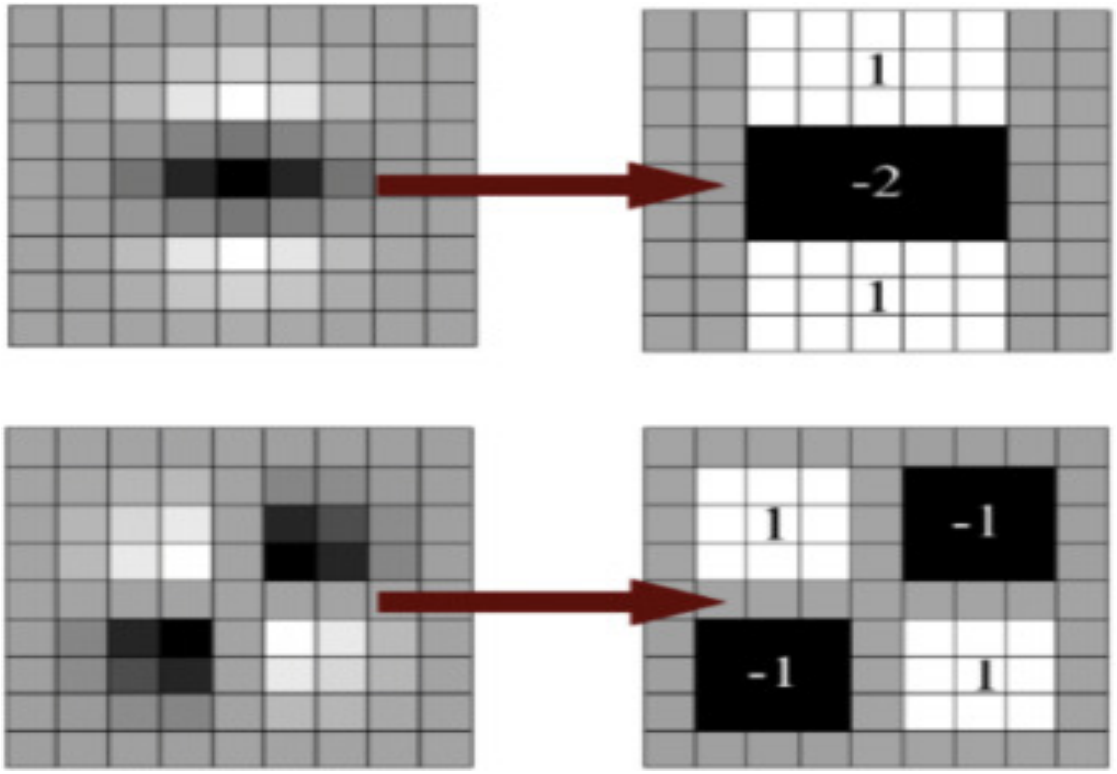

Figure 3.1: Box filters approximates to Gaussian second order in y- and xy-direction [11].

translation, scaling, rotation and illumination. However, different SIFT algorithm, we use SURF feature detector to locate the characterize points. For the descriptor and matching step, SIFT algorithm is employed because it outperforms other contemporary local descriptor on a lot of complicated scenes. From SIFT algorithm, it gives each point two types of features as the representation. The first one is orientation, it achieves that the descriptor is invariance to rotation, location and scale by computing the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ as follow,

$$m(x,y) = \sqrt{\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2} \quad (3.2)$$

$$\theta(x,y) = atan2((L(x+1,y) - L(x-1,y)), (L(x,y+1) - L(x,y-1))) \quad (3.3)$$

where $L(x, y)$ indicates an image sample, the magnitude and direction calculations are done for each pixel in a neighboring region around characterize point which we located by SURF feature. Furthermore, an orientation histogram with 8 bins is formed and each bin covers 45 degrees, and example is shown in Fig.3.2. For each sample, it added to a histogram bin is weighted by its gradient magnitude.

Given the computation of each image pixel, a description is presented as follow. SIFT algorithm segments 16×16 region with characterize point as the centering one, and the region contains 4 sub-regions which is 4×4 pixels size. In each sub-region, every pixel is computed and obtain an orientation histogram with 8 bins. The histogram is computed by using a Gaussian-weighted window with σ that is 1.5 times that of the scale of the characterize points. In doing so, a descriptor is generated and presented as a vector of all the values of these histogram, and it has 128 elements. In order to enhance invariance to affine changes in illumination, the description vector is also need to be normalized in unit length.

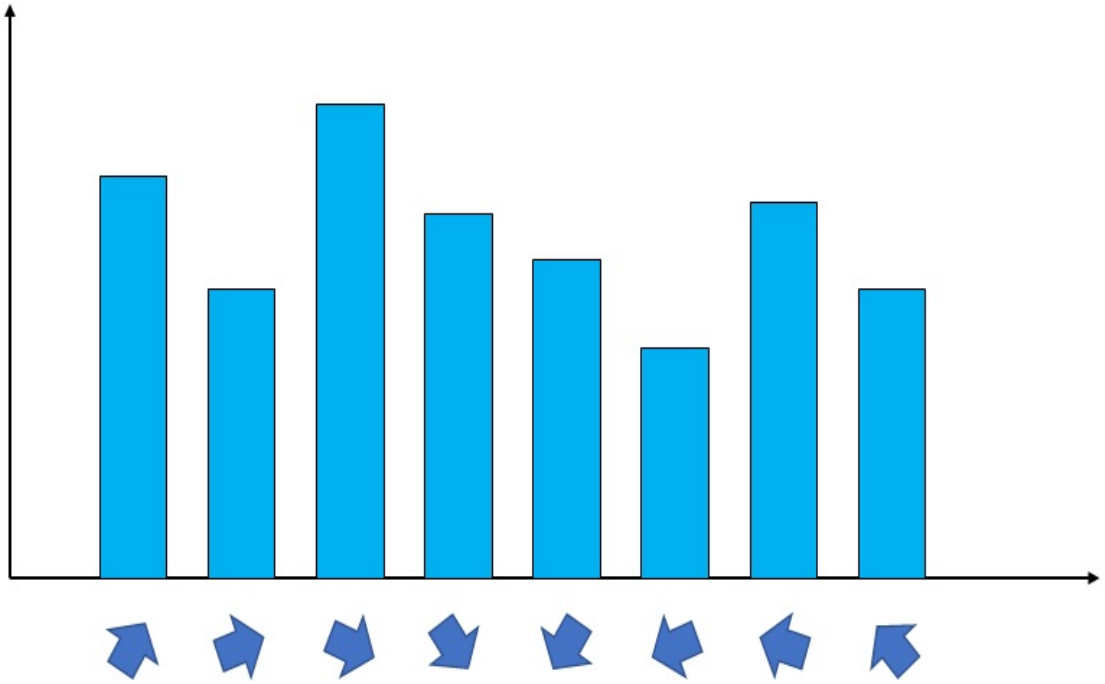

Figure 3.2: Orientation histogram

3.2 Experiments

The aim of the implement is two-fold: (1) to test if the characterize points of meaningful object can be extracted; (2) to test if the trajectories can be generated. From the review of the above, we compare SURF and SIFT with other algorithms, and decide to apply them to generate trajectory data from video dataset.

We conduct experiments on tracking benchmarks of the KITTI Vision Benchmark Suite [45], which are collected by drive a standard station wagon with two high -resolution camera and grayscale video cameras around mid-size city. For tracking benchmarks, it developed by its specific purpose, tracking, and it has 21 training sequence including 1756 images. We apply label information on each trajectory by identifying its first point of the corresponding trajectory which label belong to.

For global setting, 400 is the minimum Hessian threshold throughout our experiments. We use the default setting of [87]'s SIFT algorithm parameter setting,

and the parameters settings are given as follow,

Table 3.1: SURF parameters setting

Hessian threshold	400
Octaves number	2
feature dimensions	64

Table 3.2: SIFT parameters setting

feature numbers	100
Octaves number	2
contrast threshold	0.04
Gaussian function σ	1.6

3.2.1 SURF algorithm

We choose first 100-150 images from all sequences, the minimum Hessian threshold was chosen as 400, and the results are shown in Fig.3.3.

3.2.2 SIFT algorithm

As Fig.3.3 shown, a bunch of characterize points are located. However, only a small set has meaning, and most of them belong to miscellaneous. Therefore, we apply label information on that and choose the points meaningful, few other points are chosen as miscellaneous points. Furthermore, after computing the distance between arbitrary two feature descriptor, we set a distance threshold to choose some good matches. We display some matching results in Fig.3.4 and Fig.3.5. From the results, we found the matched points could coming from different objects. Therefore, we compute the distance from the current point to original point and make a comparison between them to measure if one object

Figure 3.3: Results of experiments on the first image of first three sequences in KITTI data set.

them came from. In the case that the ego-vehicle platform is moving, so the still object is moving relative to the ego-platform, but the speed and relative motion of all still objects are same. Therefore, in the feature extraction step, we have to take into account all these factors and classify the trajectories by using them.

Based on our opinion, we generate 350,609 trajectories and most of them are meaningful such as pedestrian, van, car, tram and cyclist. After that, we match the characterize point to 3D real world coordinate. Camera calibration technol-

ogy is used here to calibrate the point from 3D real world coordinate to 2D image coordinate. Furthermore, 3D coordinate and image data are recorded by Lidar device and camera, respectively, so the calibration between two different devices is also needed.

3.3 Conclusion

In this chapter, we proposed a method to locate, extract and connect points as trajectory, and find their label for further use as well. The method is specifically designed for generating trajectory in 3D real world coordinate that have camera calibration, characterize point location and characterize point connection. We illustrate how to find the points in 3D environment. To increase the accuracy of generation, we propose to compute the distance between two points belonging to the consecutive frames. respectively. Experimentally, our method generates a lot of trajectory data and most of them have meaning. In future work, we will focus on image segmentation and choose one point to represent object and improve the performance of trajectory generation.

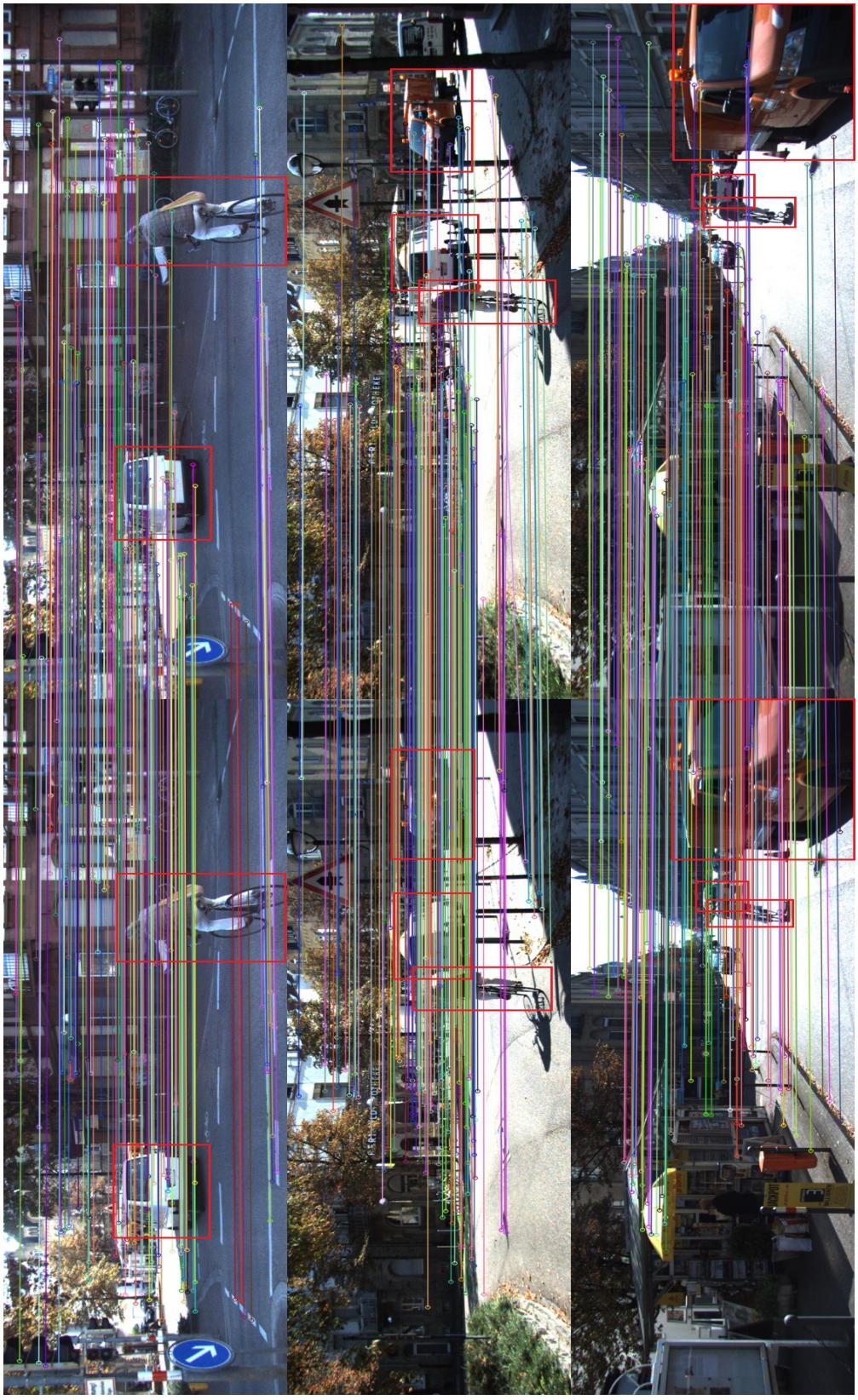

Figure 3.4: Results of experiments on arbitrary three images of the first sequence of KITTI data set.

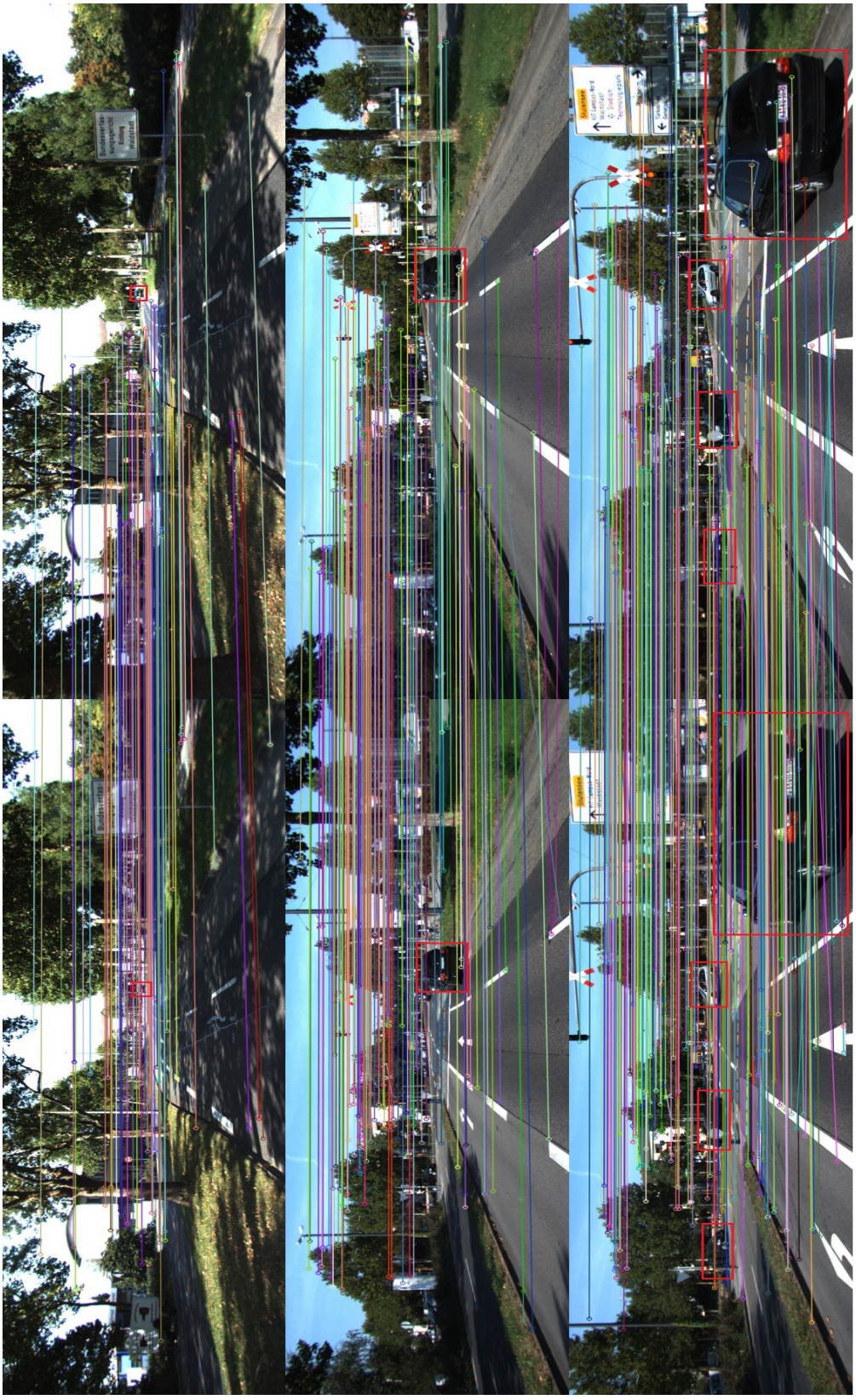

Figure 3.5: Results of experiments on arbitrary three images of the second sequence of KITTI data set.

Chapter 4

Trajectory Feature Extraction

In Chapter 3, trajectory data are generated by connecting characterize points in consecutive frames. If the spatiotemporal information are treated as main features or factors to cluster, classification performance could be improved. With spatiotemporal information, a feature extraction method can be implemented here.

Trajectory data contains a lot of information about objects, which even can help computer to recognize what the object is. Therefore, a proper method is needed to extract the features including all information. Most existing methods focus on single type feature such as motion and relative location, but it ignores other information. For example, vehicle can be recognized by its motion but it could be confused with cyclist, so it only be separated if speed information is involved in clustering step. In this chapter, we extract two types features including continues feature and discrete feature.

In this chapter, we propose to extract two different feature types of trajectory data. First, the coefficients of DFT are extracted to represent the spatiotemporal information in unique frequency domain, and it is important that all trajectory data are transformed into a fixed number of parameters. Then, we compare the initial and end states of object trajectory to obtain the relative motion of the corresponding object.

4.1 Methodology

Trajectory data contain a lot of information that is useful for trajectory clustering. However, limited methods are proposed to extract them and the extracted information are finite. Therefore, DFT and relative motion are employed to represent trajectory data.

Furthermore, another issue need to be fixed, the trajectory length. Trajectory data are recorded as different length. For example, suppose we have two trajectories, $Trajectory_1 = \{tr_1, tr_2, \cdots, tr_{m_1}\}$ and $Trajectory_1 = \{tr_1, tr_2, \cdots, tr_{m_2}\}$, which have different lengths, so we need to find a proper method to measure the difference between $Trajectory_1$ and $Trajectory_2$ which difference from some popular methods such as Distance measuring methods. Therefore, that is a issue if $m_1 \neq m_2$ and they hold much information including trajectory lengths.

4.1.1 Continuous Features

DFT is an algorithm converting digital signals in time domain to the samples in frequency domain. The coefficient of DFT are the parameters of sine and cosine function, so we can use same number of parameters represent different trajectories. That operation makes data could been compared under same circumstance.

For DFT, it is defined as follow,

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \cdot \exp\left(-\frac{2\pi i}{N} kn\right) \\ &= \sum_{n=0}^{N-1} x_n \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)] \end{aligned} \tag{4.1}$$

where $\{x_n\}$ have N elements. All $\{x_n\}$ can be transformed into $\{X_k\}$ and each X_k is represented by the combination of $\sum_{n=0}^{N-1} x_n \cos(2\pi kn/N)$ and $\sum_{n=0}^{N-1} x_n \sin(2\pi kn/N)$.

We collect trajectory data as 3 dimensional format as $tr_i = (x_i, y_i, z_i)$, so we

can get 3 set of DFT coefficients as X_k, Y_k and Z_k.

$$\begin{cases} X_k = \sum_{n=0}^{N-1} x_n \cdot \exp\left(-\frac{2\pi i}{N} kn\right) \\ Y_k = \sum_{n=0}^{N-1} y_n \cdot \exp\left(-\frac{2\pi i}{N} kn\right) \\ Z_k = \sum_{n=0}^{N-1} z_n \cdot \exp\left(-\frac{2\pi i}{N} kn\right) \end{cases} \tag{4.2}$$

Furthermore, we can set the number of coefficients are fixed, even though the number of points in each trajectory may vary. The coefficients of DFT follow continues distribution, so we call the coefficients of DFT as continues feature.

4.1.2 Discrete Features

Although we have extracted continues features, the object status is still unknown because the still objects may have similar or even same trajectory. For example, two objects on the roadside may have motion when the recording device is moving on the road, even one of them is on the road such as stopping vehicle. Therefore, the object motion related to recording device is critical to determine whether it is still, it improves the performance of feature extraction and clustering accuracy.

Based on the above discussion, we need to compute the distance that object moved. However, it could be large in most circumstance, for example, a parking vehicle on the roadside could be recorded for a brunch of frames and generate a large of trajectory points, so the distance between the first point and the last one would be a big distance. Therefore, in the real world, it is unexpectable to determine the motion with regard to the ego-platform when 3D positions of trajectory apply in our model.

We prefer to use the data that record in ego-viewpoint, for instance, camera device [162]. In each frame, objects are observed and recorded in ego-device's viewpoint, i.e. what you see is what you get relative to your position in Fig.4.1 and Fig.4.2.

Figure 4.1: Top to bottom: 0th, 40th, 80th and 120th frame in 1st sequence of KITTI benchmark. A white van and a cyclist keep staying in the center area of camera image. According to that ego-platform is moving, we have the information that the van and the cyclist is moving.

Figure 4.2: Top to bottom: 120th, 130th and 140th frame in 1st sequence of KITTI benchmark. The vehicles parking on the roadside moving a big distance in camera image, such as the silver one moving from center area to border area.

The relative motion only used to determine the object motion status, so we call this type of feature as Discrete feature.

4.2 Experiments

Our experiments are still conducted on tracking data set of KITTI benchmarks. For continues feature, we follow equation.(5.1) to generate the features. For discrete features, we propose to split the image frame into 3×3 patches and compute the object moving distance from the start to the end, shown in Fig.4.3 and Fig.4.4. By computing the distance of object traveled, we can determine the corresponding object is moving or not, even it slower or faster than the speed of camera device.

Figure 4.3: Results of split camera image into 3×3 patches. The white van and cyclist are keeping in the center area.

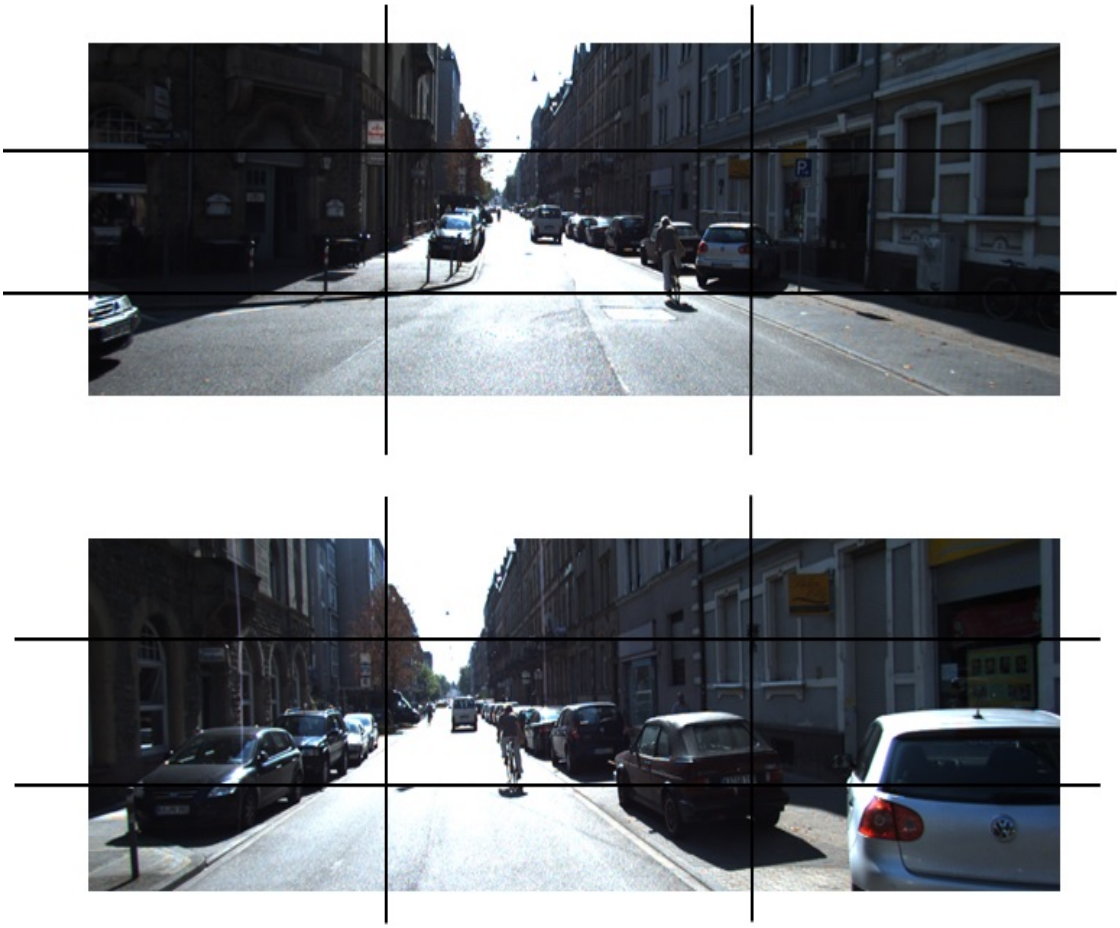

Figure 4.4: Results of split camera image into 3×3 patches. The white van and cyclist are keeping in the center area.

4.3 Conclusion

This chapter proposes to extract two different features which is continue feature and discrete feature to represent trajectory data. Experiments on KITTI benchmark show it efficient on describing object motion status. However, it still have some issues need to be noticed and improved that, one object may have multiple trajectories including long trajectories and short trajectories. The short ones may not properly describe the object motion status, even few information inside the coefficients of DFT. Therefore, a threshold is needed to determine which trajectories are required and suitable for feature extraction, which means the trajectories we need are the ones have long tracks.

In the following chapter, we will introduce dual-variable LDA model to process all these features to cluster trajectory data. According to the fact that LDA algorithm has state-of-art performance in clustering algorithm, we proposes a novel algorithm to analysis two different features and give a cluster result. Hence, the proposed dual-variable LDA model is expected to boost the clustering accuracy.

Chapter 5

Dual-variable LDA Model for Trajectory Clustering

As a type of statistical model for discovering the abstract "topics" that occur in vocabulary corpus, topic model is an efficient and fast method to classify query data, but the input data is only vocabulary words or bag of words. Due to its great performance and highly accuracy, a lot of models use topic model to implement clustering works [68] [74] [114] [142].

Furthermore, according to the application in real world, we prefer to employ unsupervised classification method, or called clustering method, to implement our trajectories clustering. As a model employing a lot of features and categorizing into small number of topics, topic model is a proper method to implement the goal. However, only one type of semantic feature is considered. Thus, a topic model considering two more semantic features is needed here. Motivated by [67] and [77], we propose a model to discover semantic content from the obtained evidences. Furthermore, our model is derived from topic model [78], which is derived from pLSA [51] [52] and LDA [12] [133]. Specifically, LDA is a generative probabilistic model introduced, and it is a three-level hierarchical Bayesian probabilistic model that a mixture of a latent set of distributions of discrete semantic data to set topics. Our multi-modal LDA model is derived from LDA model. It is a generative model that allows a set of few unobserved words explaining a large set of observed words. In simple words, the observed data can be clustered into

categories which could be described by few topic words. In the previous chapter, we obtain the features of trajectory data, but they are different types and have different data type. Hence, we need to propose a novel method to process these features and classify them into corresponding categories, because words, one of the discrete features, is only input data that LDA model can process.

In this chapter, we present dual-variable LDA model by combining discrete and continue features, which discrete feature are traditional input of LDA model. In doing so, spatiotemporal feature which act as continues feature and motion feature which act as discrete feature are both applied in clustering model and improve the performance. However, traditional LDA model only considers discrete feature, *vocabulary words*, so we need to derive a novel distribution function to estimate the probability of continues feature allocating to the topics. Then, Gibbs sampler has been approved the accuracy and fast in sampling proper words and assigned into topics, and it is a MCMC algorithm for obtaining a sequence of observations which are approximated from a specified multivariate probability distribution. Therefore, we need to find a sampling method for the coefficients of DFT and it should be suitable for sampling in our model. Finally, we combined two probability distribution together to form the assignment results and obtain the categories for each trajectory.

LDA algorithm are efficient to process semantic data and it was presented as a graphical model [12]. More than semantic data, a lot of other features are involved as well [1] [68] [142]. It still remaining challenging to improve the performance and operating speed for clustering images or video data. A straightforward method to cluster *visual words* which indicate image patch to generate a few set of topics, and all visual words are consist as documents [123] [142]. The method is useful to segment the objects from image and classify them into categories. Furthermore, LDA algorithm is used for recognize human actions [100]. However, they all employ low-level visual words, such as image patches, spatial interesting points and pixels, into topics with semantic meaning.

In this chapter, we propose a dual-variable LDA algorithm to process two variables extracted from single data. Firstly, a dual-variable LDA model for two discrete features is proposed and apply on lecture videos, which represent as two semantic words sets extracted from the speech of speaker and content of slides,

we call this model as multimodel-LDA model. After that, another dual-variable LDA model is further derived to process trajectory data, represented by continues feature and discrete feature, and it will have an experiment on KITTI benchmark.

Lecture videos are knowledge sources, intellectual properties of university and material for multimedia course ware and teaching evaluation. Compared with text in the book, lecture video has many unique advantages: it is more salient and attractive, so it can grab users' attention instantly; it carries more visual information that can be comprehended more quickly. Currently, data, speech and digital TV broadcast are regarded as the most considerable contents of online learning or e-learning is rapidly emerging in the world, and separate education to students distributed around the world. According to the booming of Internet and digital technology, Internet based distance learning has many advantages such as high degree of interactivity, a variety of courses is available at any time, uses less bandwidth. To give background, Web Based Training is a computer-based educational service that uses the Internet to support distance learning. And this technology is becoming popular for providing university courses and business training as it allows students to learn wherever they are situated.

Therefore, how to mine the related knowledge and corresponding multimedia from Internet is a key for online-learning. For this purpose, we propose a system that users can efficiently find their interesting multimedia from the Internet. Furthermore, users can publish their note or comments for some lectures, and share with other users through the network. This makes a community for online-learning, which is interactive and efficiently. Therefore, the content-based multimedia retrieval is an important problem for our system.

Different from general multimedia on the Internet, lecture videos contain many information sources. Specially, a lecture video also contains speech of speaker, and the video usually contains the slides for the lecture, as shown in Fig.5.1. Furthermore, there are some scripts or PowerPoint for the lecture on the Internet. The fusion of this video is important for content-based multimedia retrieval.

As shown in Fig.5.1, our system first segments each video into shots, and obtains two kinds of information from each shot: speech of speaker and content of slides. Regards to speech of speaker, we extract what the speaker has said by ASR which is an algorithm extracting semantic text from spoken language. Re-

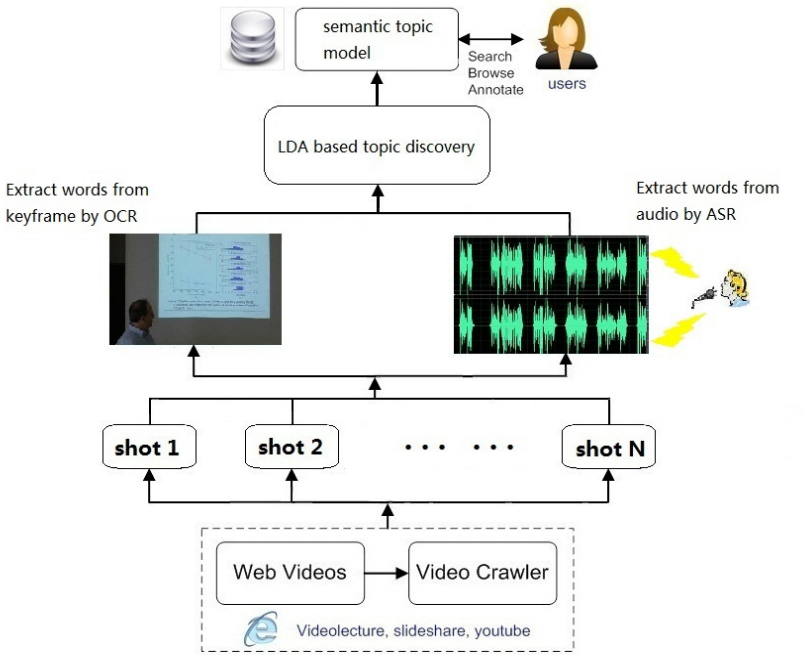

Figure 5.1: Multimodel-LDA based topic discovery for lecture videos.

gards to content of slides, we first extract keyframes of the shot, which is defined as the frame that contains the slides of the lecture. Since the video of lecture usually contains two kinds of frames: one is an image of speaker, the other is an image of slides. It is obvious the frames contains slides contains more useful clue for content analysis. After keyframe extraction, we extract what the slide shows by OCR which converts text from images to machine-encoded text. Therefore, we obtain two sets of texts from the shot, which are treated as evidence for semantic content analysis. In this section, we propose a model for discovery of semantic content from the obtained evidence, as shown in Fig.5.3(b).

After multimodel-LDA model for two discrete variables, we furthermore derive a dual-variable LDA model for two different variables in the following section. This is a more proper method to solving trajectory data generated from video data. Trajectory are represented as vector and one variable indicating object motion status. All variables are set up as word-document assignment and a generative procedure to assign words to documents.

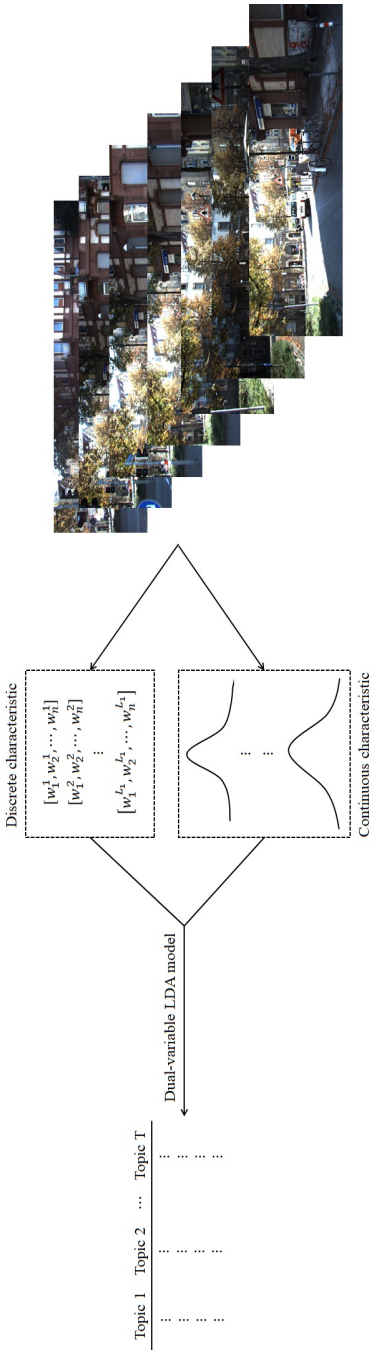

Figure 5.2: Dual-variable LDA model for trajectory data.

5.1 Multimodel-LDA Model for Semantic Topic Discovery

Topic model, such as pLSA [51] [52] and LDA [12], is originally proposed for text processing, where the topics are described by a distribution of words. For example, LDA is a generative probabilistic model introduced. It is a three-level hierarchical Bayesian probabilistic model that a mixture of a latent set of distributions of discrete semantic data to set topics, as shown in Fig.5.3(a).

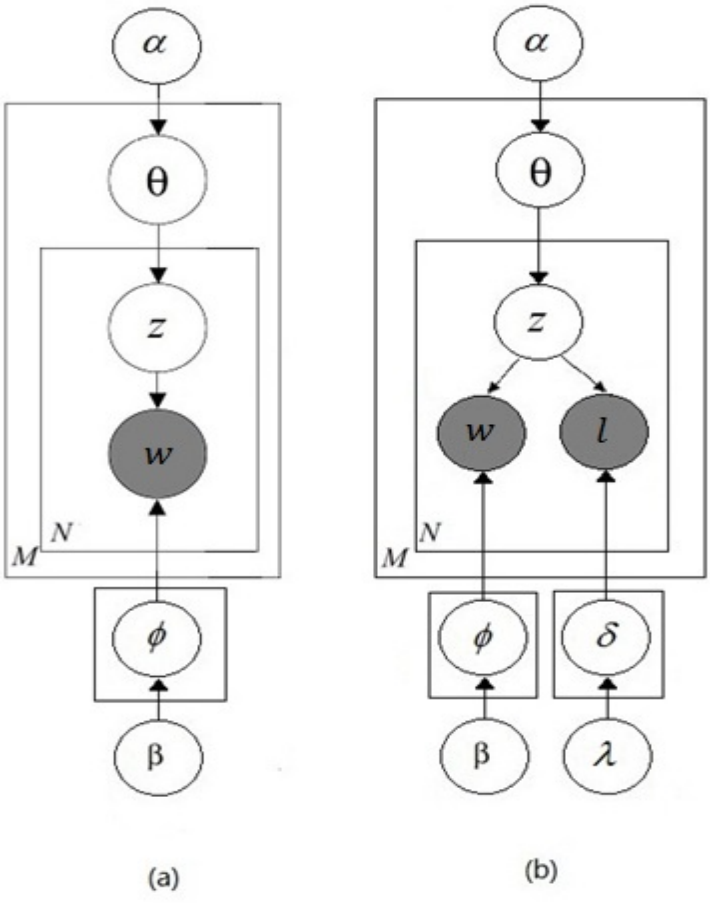

Figure 5.3: (a). standard LDA model. (b). multi-modal LDA model

5.1.1 Evidence Extraction

Before we process semantic information from video, we need to extract that by some methods. In lecture video, speakers teach knowledge to students or Internet users through speaking and showing slides, so semantic information can be extracted from lecture's presentation and slides on projector.

ASR can extract information from speech through capturing spoken words and then classify words groupings to form a sentence. It is a computer-driven and transcribes spoken language into text that can be read by using real time. An ASR system to function, it must follow three steps. The system will capture words that are recording with any storage device, and then converting the digital signals of the speech into syllables (phonemes). This is referred to as feature analysis. Next, the system will match the spoken syllables to a phoneme sequence that is kept in an acoustic model database. This is called pattern classification. The system will then try to make sense of what is being said by comparing the word phonemes from the previous step, this time with a language model database. Based on ASR, speech information can be extracted.

OCR extracts semantic information on slides through scanning and translating image of handwritten, typewritten or printed text into machine-encoded text. An OCR system recognizes the fixed static shape of the character and using a smaller dictionary to match recognized words, so that it can increase recognition rates. Through this technology, the semantic information on slides can be extracted.

5.1.2 Multimodal-LDA Model

The Markov chain Monte Carlo is constructed to converge to the target distribution [46]. Furthermore, [51] uses Gibbs Sampler, the heat bath algorithm in statistical physics, where the next state is reached by sequentially sampling all variables from their distribution when conditioned on the current values of all other variables and the data. So based on this thinking, Thomas proposed the equation

$$P\left(z_i = j \,|\mathbf{z}_{-i}, \mathbf{w}\right) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(w_i)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,}^{(d_i)} + T\alpha} \tag{5.1}$$

where $n_i^{(\cdot)}$ is a count that does not include the current assignment of z_i . This result is quite intuitive; the first ratio expresses the probability of w_j under topic j, and the second ratio expresses the probability of topic j in document d_i. Having obtained the full conditional distribution, the Monte Carlo algorithm is then straightforward. $\{z_i\}$ are initialized to values in $\{1, 2, \cdots, T\}$, determining the initial state of the Markov chain. We do this with an online version of the Gibbs sampler.

Table 5.1: Multimodal-LDA variable list

α	Dirichlet prior parameter on the per-document topic distributions
β	Dirichlet prior parameter on the per-topic word distribution based on ASR
λ	Dirichlet prior parameter on the per-topic word distribution based on OCR
θ_i	the topic distribution for document i, and **theta**$\sim Dir(\alpha)$
α	the word distribution for topic i based on ASR, and $\Phi \sim Dir(\beta)$
α	the word distribution for topic i based on OCR, and $\delta \sim Dir(\lambda)$

1. Choose $\theta_i \sim Dir\left(\alpha\right)$, where $i \in \{1, \cdots, M\}$ and $Dir\left(\alpha\right)$ is Dirichlet Distribution;

2. For each ASR words vector w_i of W_{ASR}, choose $\phi_i \sim Dir\left(\beta\right)$, where $w_i \in [1, \cdots, w_{W_1}]$;

3. For each OCR words vector l_j of W_{OCR}, choose $\delta_j \sim Dir\left(\lambda\right)$, where $l_j \in [1, \cdots, l_{L_1}]$;

4. For each word w_i in the ASR word set:

- Choose $z_i \sim Multinomial\left(\theta_i\right)$.
- Choose w_i from $p\left(w_i|z_i, \beta\right)$, a multinomial probability conditional on the topic z_i

5. For each word l_j in the OCR word set:

- Choose $z_j \sim Multinomial\,(\theta_j)$.
- Choose l_j from $p\,(l_j|z_j, \beta)$, a multinomial probability conditional on the topic z_i

Now, we explain these variables in Table.5.1. According to Thomas's thinking, we advised LDA model based on Gibbs Sampler so that they can input two available and cluster them based on semantic information. And we get words from videos by ASR, get words from videos by OCR, and given documents containing topics expressed over words. Based on [46], we need to derive equation include two variables.

Φ is $T \times V$ Markov matrix, where V is the dimension of the vocabulary and T is the dimension of the topics, and each row of which denotes the word distribution of a topic. Our strategy for discovering topics differs from previous approaches in not explicitly representing θ and δ as parameters to be estimated, but instead considering the posterior distribution over the assignments of words to topics $p\,(\mathbf{L}|\mathbf{Z})$. We then obtain estimates of θ and δ by examining this posterior distribution.

Before we derive multimodal-LDA algorithm, we have to derive Gibbs sampling based LDA model firstly. we should notice that α is a value. Although these hyper-parameters could be vector-valued, we assume symmetric Dirichlet priors because of the purposes of Gibbs sampling based LDA model, with and each having a single value. However it is a k dimensional vector in original LDA model. With generative model, the formula can be written as

$$P(\mathbf{Z}, \mathbf{W}|\alpha, \beta) = \mathbf{P}(\mathbf{Z}|\alpha) \cdot \mathbf{P}(\mathbf{W}|\mathbf{Z}, \beta), \tag{5.2}$$

where $P(\mathbf{Z}|\alpha)$ and $P(\mathbf{W}|\mathbf{Z},\beta)$ are derived in the following.

$$\begin{aligned} P(\mathbf{Z}|\alpha) &= \int_{\theta} P(\mathbf{Z},\theta|\alpha)\mathbf{d}\theta \\ &= \int_{\theta} \prod_{d} P(\mathbf{Z_d}|\theta_{\mathbf{d}})\mathbf{P}(\theta_{\mathbf{d}}|\alpha)\mathbf{d}\theta \\ &= \int_{\theta} \prod_{d} P(\theta_{\mathbf{d}}|\alpha) \prod_{\mathbf{n}} \mathbf{P}(\mathbf{Z_{d,n}}|\theta_{\mathbf{d}})\mathbf{d}\theta \\ &= \prod_{d} \int_{\theta_d} P(\theta_{\mathbf{d}}|\alpha) \prod_{\mathbf{n}} \mathbf{P}(\mathbf{Z_{d,n}}|\theta_{\mathbf{d}})\mathbf{d}\theta_{\mathbf{d}}, \end{aligned} \tag{5.3}$$

where d indicates document index and n denotes word index. Furthermore, d, n are integer values, and $d \in [1, D]$, $n \in [1, N]$. $P(\theta_{\mathbf{d}}|\alpha)$ is chosen from a Dirichlet Distribution, and $P(Z_{d,n}|\theta_{\mathbf{d}})$ is equal to $\prod_t \theta_{d,t}^{n_{n,t}^d}$ where t indicates topic index, $n_{n,t}^d$ is the total number of the nth word belonging to the dth document that assigning to the tth topic. Therefore, we can obtain

$$\begin{aligned} P(\mathbf{Z}|\alpha) &= \prod_{d} \int_{\theta_d} P(\theta_{\mathbf{d}}|\alpha) \prod_{\mathbf{n}} \mathbf{P}(\mathbf{Z_{d,n}}|\theta_{\mathbf{d}})\mathbf{d}\theta_{\mathbf{d}} \\ &= \prod_{d} \int_{\theta_{\mathbf{d}}} \frac{1}{B(\alpha)} \prod_{t} \theta_{d,t}^{\alpha-1} \cdot \prod_{n} \prod_{t} \theta_{d,t}^{n_{n,t}^d} d\theta_{\mathbf{d}} \\ &= \prod_{d} \int_{\theta_{\mathbf{d}}} \frac{1}{B(\alpha)} \prod_{t} \theta_{d,t}^{\alpha-1} \cdot \prod_{t} \theta_{d,t}^{\sum_n n_{n,t}^d} d\theta_{\mathbf{d}} \\ &= \prod_{d} \int_{\theta_{\mathbf{d}}} \frac{1}{B(\alpha)} \prod_{t} \theta_{d,t}^{\alpha-1+\sum_n n_{n,t}^d} d\theta_{\mathbf{d}} \\ &= \prod_{d} \frac{1}{B(\alpha)} \int_{\theta_{\mathbf{d}}} \prod_{t} \theta_{d,t}^{\alpha-1+\sum_n n_{n,t}^d} d\theta_{\mathbf{d}}. \end{aligned} \tag{5.4}$$

According to the properties of Dirichlet Distribution, $\int Dir(\alpha)dx = \int \frac{1}{B(\alpha)} x^{\alpha-1}dx = 1$, then we can get $\int x^{\alpha-1}dx = B(\alpha)$.

$$
\begin{aligned}
P(\mathbf{Z}|\alpha) &= \prod_d \frac{1}{B(\alpha)} \int_{\theta_\mathbf{d}} \prod_t \theta_{d,t}^{\alpha-1+\sum_n n_{n,t}^d} d\theta_\mathbf{d} \\
&= \prod_d \frac{1}{B(\alpha)} \cdot B\left(\alpha + \sum_n n_{n,t}^d\right) \\
&= \prod_d \frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T} \cdot \frac{\prod_t \Gamma\left(\alpha + \sum_n n_{n,t}^d\right)}{\Gamma\left[\sum_t \left(\alpha + \sum_n n_{n,t}^d\right)\right]} \\
&= \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \cdot \prod_{i=1}^D \frac{\prod_{t=1}^T \Gamma\left(\alpha + n_t^d\right)}{\Gamma\left(T\alpha + n_{(\cdot)}^d\right)}.
\end{aligned}
\tag{5.5}
$$

From the equation.(5.5), $P(\mathbf{Z}|\alpha)$ is only relating to documents and topics. For the second term $P(\mathbf{W}|\mathbf{Z},\beta)$ of equation.(5.2), the derivation is following,

$$
\begin{aligned}
P(\mathbf{W}|\mathbf{Z},\beta) &= \int_\Phi P(\mathbf{W}|\mathbf{\Phi},\mathbf{Z},\beta)\, P(\Phi|\mathbf{Z},\beta)\, d\Phi \\
&= \int_\Phi P(\mathbf{W}|\mathbf{\Phi},\mathbf{Z})\, P(\Phi|\beta)\, d\Phi \\
&= \prod_t \int_{\phi_\mathbf{t}} P(\mathbf{W}|\phi_\mathbf{t},\mathbf{Z})\, P(\phi_\mathbf{t}|\beta)\, d\phi_\mathbf{t} \\
&= \prod_t \int_{\phi_\mathbf{t}} P(\phi_\mathbf{t}|\beta) \prod_n \prod_d P(w_{d,n}|\mathbf{Z}_{\mathbf{d},\mathbf{n}} = \mathbf{t}, \phi_\mathbf{t})\, d\phi_\mathbf{t}.
\end{aligned}
$$

It should be concern that Φ are the words distributions over K topics and they are unrelated with document, so we assume that $w_{n,d}$ is corresponding to the vth word in the vocabulary where $v \in [1, V]$.

$$
\begin{aligned}
P(\mathbf{W}|\mathbf{Z},\beta) &= \prod_t \int_{\phi_\mathbf{t}} P(\phi_\mathbf{t}|\beta) \prod_n \prod_d P(w_{d,n}|\mathbf{Z}_{\mathbf{d},\mathbf{n}}, \phi_\mathbf{t})\, d\phi_\mathbf{t} \\
&= \prod_t \int_{\phi_\mathbf{t}} \frac{1}{B(\beta)} \prod_v \phi_{t,v}^{\beta-1} \cdot \prod_v \phi_{t,v}^{n_{t,v}} d\phi_\mathbf{t} \\
&= \prod_t \frac{1}{B(\beta)} \int_{\phi_\mathbf{t}} \prod_v \phi_{t,v}^{\beta-1+n_{t,v}} d\phi_\mathbf{t} \\
&= \prod_t \frac{1}{B(\beta)} B(\beta + n_{t,v})
\end{aligned}
$$

$$
\begin{aligned}
&= \prod_t \frac{\Gamma(V\beta)}{\Gamma(\beta)^V} \cdot \frac{\prod_v \Gamma(\beta + n_{t,v})}{\Gamma\left[\sum_v (\beta + n_{t,v})\right]} \\
&= \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^T \cdot \prod_t^T \frac{\prod_v \Gamma(\beta + n_{t,v})}{\Gamma\left(N\beta + n_{t,(\cdot)}\right)}.
\end{aligned} \tag{5.6}
$$

Therefore, from equation.(5.5) and equation.(5.6), the generative model are rewritten as

$$
\begin{aligned}
P(\mathbf{Z}, \mathbf{W}|\alpha, \beta) = &\left[\left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \cdot \prod_{i=1}^D \frac{\prod_{t=1}^T \Gamma\left(\alpha + n_t^d\right)}{\Gamma\left(T\alpha + n_{(\cdot)}^d\right)}\right] \\
&\times \left[\left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^T \cdot \prod_t^T \frac{\prod_v \Gamma(\beta + n_{t,v})}{\Gamma\left(N\beta + n_{t,(\cdot)}\right)}\right].
\end{aligned} \tag{5.7}
$$

Posterior probability can be obtained by Bayes' theorem,

$$
P(\mathbf{Z}|\mathbf{W}) = \frac{P(\mathbf{Z}, \mathbf{W})}{\sum_{\mathbf{Z}} P(\mathbf{Z}, \mathbf{W})}. \tag{5.8}
$$

It is difficult to compute posterior probability $P(\mathbf{Z}|\mathbf{W})$ directly, because the denominator $\sum_{\mathbf{Z}} P(\mathbf{Z}, \mathbf{W}) = \sum_{d=1}^D \sum_{n=1}^N P(Z_{d,n}, \mathbf{W})$ cannot be compute. Thus, Gibbs sampler is employed to solve the equation. Gibbs sampler is derived from Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of samples by applying sampling method iterative, and it useful when direct sampling is difficult. The probability of nth word belonging to dth document is assigned to tth topic is proposed as follow,

$$
\begin{aligned}
P\left(Z_{d,n} = t|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{W}\right) &= P\left(Z_{d,n} = t|W_{d,n}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{W}_{-(\mathbf{d},\mathbf{n})}\right) \\
&= \frac{P\left(W_{d,n}, Z_{d,n} = t|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)}{P\left(W_{d,n}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)},
\end{aligned} \tag{5.9}
$$

By Bayesian rule (posterior is in propotion to prior multiple with likelihood), the conditional prior of $Z_{d,n}$ is $P\left(Z_{d,n} = t|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)$ and the likelihood is $P\left(W_{d,n}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}} = \mathbf{t}\right)$,

$$P\left(Z_{d,n}=t|W_{d,n},\mathbf{Z}_{-(\mathbf{d},\mathbf{n})},\mathbf{W}_{-(\mathbf{d},\mathbf{n})}\right)\propto P\left(Z_{d,n}=t|\mathbf{W}_{-(\mathbf{d},\mathbf{n})},\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)\cdot P\left(W_{d,n}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})},\mathbf{Z}_{-(\mathbf{d},\mathbf{n})},\mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right), \tag{5.10}$$

For the first term, $P\left(Z_{d,n}=t|\mathbf{W}_{-(\mathbf{d},\mathbf{n})},\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right), Z_{d,n}$ is sampled by the fixed $\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}$ with Gibbs sampling, so $P\left(Z_{d,n}=t|\mathbf{W}_{-(\mathbf{d},\mathbf{n})},\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)=P\left(Z_{d,n}=t|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)$.

$$\begin{aligned}
P\left(Z_{d,n}=t|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)&=\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}P\left(Z_{d,n}=t|\theta_\mathbf{i},\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)P\left(\theta_\mathbf{i}|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)d\theta_\mathbf{i}\\
&=\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}P\left(Z_{d,n}=t|\theta_\mathbf{i}\right)P\left(\theta_\mathbf{i}|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)d\theta_\mathbf{i}\\
&\propto\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}P\left(Z_{d,n}=t|\theta_\mathbf{d}\right)\cdot P\left(\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}|\theta_\mathbf{i}\right)P\left(\theta_\mathbf{i}\right)d\theta_\mathbf{i}\\
&=\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}\theta_{d,t}^{n_{n,t}^{d}}\cdot P\left(\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}|\theta_\mathbf{i}\right)P\left(\theta_\mathbf{i}\right)d\theta_\mathbf{i}\\
&=\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}\theta_{d,t}^{n_{n,t}^{d}}\cdot\prod_{k=1}^{T}\theta_{i,k}^{n_{-n,k}^{i}}\cdot\frac{1}{B\left(\alpha\right)}\prod_{k=1}^{T}\theta_{i,k}^{\alpha-1}d\theta_\mathbf{i}\\
&=\frac{1}{B\left(\alpha\right)}\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}\prod_{k=1}^{T}\theta_{i,k}^{\alpha-1+\sum_{j=1}^{N}n_{j,k}^{i}}d\theta_\mathbf{i}\\
&\propto\prod_{i=1}^{D}\int_{\theta_\mathbf{i}}\prod_{k=1}^{T}\theta_{i,k}^{\alpha-1+\sum_{j=1}^{N}n_{j,k}^{i}}d\theta_\mathbf{i}\\
=&\prod_{i=1}^{D}\frac{\prod_{k}\Gamma\left(\alpha+\sum_{j=1}^{N}n_{j,k}^{i}\right)}{\Gamma\left[\sum_{k}\left(\alpha+\sum_{j=1}^{N}n_{j,k}^{i}\right)\right]}\\
=&\prod_{i=1,i\neq d}^{D}\frac{\prod_{k}\Gamma\left(\alpha+\sum_{j=1}^{N_i}n_{j,k}^{i}\right)}{\Gamma\left(T\alpha+\sum_{j=1}^{N_i}n_{j,(\cdot)}^{i}\right)}\cdot\frac{\prod_{k}\Gamma\left(\alpha+\sum_{j=1}^{N_d}n_{j,k}^{d}\right)}{\Gamma\left(T\alpha+\sum_{j=1}^{N_d}n_{j,(\cdot)}^{d}\right)}.
\end{aligned}$$

All parameters are fixed in first term of the above equation, and the denominator of second term is fixed as well, so it can be dropped.

$$
\begin{aligned}
P\left(Z_{d,n}=t|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right) &= \prod_{i=1,i\neq d}^{D}\frac{\prod_k \Gamma\left(\alpha+\sum_{j=1}^{N_i} n_{j,k}^{i}\right)}{\Gamma\left(T\alpha+\sum_{j=1}^{N_i} n_{j,(\cdot)}^{i}\right)}\cdot\frac{\prod_k \Gamma\left(\alpha+\sum_{j=1}^{N_d} n_{j,k}^{d}\right)}{\Gamma\left(T\alpha+\sum_{j=1}^{N_d} n_{j,(\cdot)}^{d}\right)} \\
&\propto \prod_k \Gamma\left(\alpha+\sum_{j=1}^{N_d} n_{j,k}^{d}\right) \\
&= \prod_{k\neq t} \Gamma\left(\alpha+\sum_{j=1}^{N_d} n_{j,k}^{d}\right)\cdot\Gamma\left(\alpha+\sum_{j=1}^{N_d} n_{j,t}^{d}\right) \\
&= \prod_{k\neq t} \Gamma\left(\alpha+n_{-n,k}^{d}\right)\cdot\Gamma\left(\alpha+n_{-n,t}^{d}+n_{n,t}^{d}\right).
\end{aligned}
$$

The first term $n_{j,k}^{d}=0 when k\neq t, j=n$ in dth document. The property of Gamma function is used in the following derivation, $\Gamma(x+1)=x\Gamma(x)$.

$$
\begin{aligned}
P\left(Z_{d,n}=t|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right) &= \prod_{k\neq t} \Gamma\left(\alpha+n_{-n,k}^{d}\right)\cdot\Gamma\left(\alpha+n_{-n,t}^{d}+n_{n,t}^{d}\right) \\
&= \prod_{k\neq t} \Gamma\left(\alpha+n_{-n,k}^{d}\right)\cdot\Gamma\left(\alpha+n_{-n,t}^{d}+1\right) \\
&= \prod_{k\neq t} \Gamma\left(\alpha+n_{-n,k}^{d}\right)\cdot\Gamma\left(\alpha+n_{-n,t}^{d}\right)\cdot\left(\alpha+n_{-n,t}^{d}\right) \\
&= \prod_{k} \Gamma\left(\alpha+n_{-n,k}^{d}\right)\cdot\left(\alpha+n_{-n,t}^{d}\right),
\end{aligned}
$$

That the words assigned to tth topic influences the results and other words are fixed is the condition should be taken into account, so

$$
\begin{aligned}
\prod_k \Gamma\left(\alpha+n_{-n,k}^{d}\right)\cdot\left(\alpha+n_{-n,t}^{d}\right) &\propto \Gamma\left(\alpha+n_{-n,t}^{d}\right)\cdot\left(\alpha+n_{-n,t}^{d}\right) \\
&= \Gamma\left(\alpha+n_{-n,t}^{d}+1\right),
\end{aligned}
$$

It should be clarified that $\Gamma(x)$ is an increasing function except for the circumstance that $x\leq 0$,

$$
P\left(Z_{d,n}=t|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}\right)=\Gamma\left(\alpha+n_{-n,t}^{d}+1\right)\propto\alpha+n_{-n,t}^{d}. \tag{5.11}
$$

For the second term of equation.(5.10), $\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}$ and $Z_{d,n}$ are fixed

$$
\begin{aligned}
& P\left(W_{d,n}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) \\
=& \prod_k \int_{\phi_{\mathbf{k}}} P\left(W_{d,n}|\phi_{\mathbf{k}}, \mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) \cdot P\left(\phi_{\mathbf{k}}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) d\phi_{\mathbf{k}} \\
=& \prod_k \int_{\phi_{\mathbf{k}}} P\left(W_{d,n}|\phi_{\mathbf{t}}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) P\left(\phi_{\mathbf{k}}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) d\phi_{\mathbf{k}} \\
\propto& \prod_k \int_{\phi_{\mathbf{k}}} P\left(W_{d,n}|\phi_{\mathbf{t}}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) \cdot P\left(\mathbf{W}_{-(\mathbf{d},\mathbf{n})}|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}, \phi_{\mathbf{k}}\right) P\left(\phi_{\mathbf{k}}\right) d\phi_{\mathbf{k}},
\end{aligned}
$$

where $W_{d,n}$ is equal to the vth word in vocabulary, and r is involved to indicate arbitrary word index in vocabulary.

$$
\begin{aligned}
& P\left(W_{d,n}|\mathbf{W}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) \\
\propto& \prod_k \int_{\phi_{\mathbf{k}}} P\left(W_{d,n}|\phi_{\mathbf{t}}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}\right) \cdot P\left(\mathbf{W}_{-(\mathbf{d},\mathbf{n})}|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}, \phi_{\mathbf{k}}\right) P\left(\phi_{\mathbf{k}}\right) d\phi_{\mathbf{k}} \\
=& \prod_k \int_{\phi_{\mathbf{k}}} \phi_{t,v}^{n_{t,v}^{(d,n)}} \cdot P\left(\mathbf{W}_{-(\mathbf{d},\mathbf{n})}|\mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{Z}_{\mathbf{d},\mathbf{n}}=\mathbf{t}, \phi_{\mathbf{k}}\right) P\left(\phi_{\mathbf{k}}\right) d\phi_{\mathbf{k}} \\
=& \prod_k \int_{\phi_{\mathbf{k}}} \phi_{t,v}^{n_{t,v}^{(d,n)}} \cdot \prod_r \phi_{k,r}^{n_{k,r}-n_{t,v}^{(d,n)}} \cdot \frac{1}{B\left(\beta\right)} \prod_r \phi_{k,r}^{\beta-1} d\phi_{\mathbf{k}} \text{ (Note: } n_{t,v}^{(d,n)}=1 \text{)} \\
=& \frac{1}{B\left(\beta\right)} \cdot \prod_k \int_{\phi_{\mathbf{k}}} \prod_r \phi_{k,r}^{n_{k,r}+\beta-1} d\phi_{\mathbf{k}} \\
\propto& \prod_k \frac{1}{B\left(\beta+n_{k,r}\right)} \\
=& \prod_k \frac{\prod_r \Gamma\left(\beta+n_{k,r}\right)}{\Gamma\left[\sum_r \left(\beta+n_{k,r}\right)\right]} \\
=& \prod_k \frac{\prod_r \Gamma\left(\beta+n_{k,r}\right)}{\Gamma\left(V\beta+n_{k,(\cdot)}\right)} \\
=& \prod_k \left[\prod_{r\neq v} \Gamma\left(\beta+n_{k,r}\right) \cdot \frac{\Gamma\left(\beta+n_{k,v}\right)}{\Gamma\left(V\beta+n_{k,(\cdot)}\right)}\right] \\
\propto& \prod_k \frac{\Gamma\left(\beta+n_{k,v}\right)}{\Gamma\left(V\beta+n_{k,(\cdot)}\right)} \\
=& \prod_{k\neq t} \frac{\Gamma\left(\beta+n_{k,v}\right)}{\Gamma\left(V\beta+n_{k,(\cdot)}\right)} \cdot \frac{\Gamma\left(\beta+n_{t,v}\right)}{\Gamma\left(V\beta+n_{t,(\cdot)}\right)}
\end{aligned}
$$

$$
\begin{aligned}
&= \prod_{k \neq t} \frac{\Gamma\left(\beta + n_{k,v}^{-(d,n)}\right)}{\Gamma\left(V\beta + n_{k,(\cdot)}^{-(d,n)}\right)} \cdot \frac{\Gamma\left(\beta + n_{t,v}^{-(d,n)} + 1\right)}{\Gamma\left(V\beta + n_{t,(\cdot)}^{-(d,n)} + 1\right)} \\
&= \prod_{k \neq t} \frac{\Gamma\left(\beta + n_{k,v}^{-(d,n)}\right)}{\Gamma\left(V\beta + n_{k,(\cdot)}^{-(d,n)}\right)} \cdot \frac{\Gamma\left(\beta + n_{t,v}^{-(d,n)}\right)}{\Gamma\left(V\beta + n_{t,(\cdot)}^{-(d,n)}\right)} \cdot \frac{\beta + n_{t,v}^{-(d,n)}}{V\beta + n_{t,(\cdot)}^{-(d,n)}} \\
&= \prod_{k} \frac{\Gamma\left(\beta + n_{k,v}^{-(d,n)}\right)}{\Gamma\left(V\beta + n_{k,(\cdot)}^{-(d,n)}\right)} \cdot \frac{\beta + n_{t,v}^{-(d,n)}}{V\beta + n_{t,(\cdot)}^{-(d,n)}} \\
&\propto \frac{\Gamma\left(\beta + n_{t,v}^{-(d,n)}\right)}{\Gamma\left(V\beta + n_{t,(\cdot)}^{-(d,n)}\right)} \cdot \frac{\beta + n_{t,v}^{-(d,n)}}{V\beta + n_{t,(\cdot)}^{-(d,n)}} \\
&\propto \frac{\beta + n_{t,v}^{-(d,n)}}{V\beta + n_{t,(\cdot)}^{-(d,n)}}
\end{aligned}
\tag{5.12}
$$

Therefore, based on equation.(5.11) and equation.(5.12), we can obtain

$$
P\left(Z_{d,n} = t | \mathbf{Z}_{-(\mathbf{d},\mathbf{n})}, \mathbf{W}\right) = \left(\alpha + n_{-n,t}^{d}\right) \cdot \frac{\beta + n_{t,v}^{-(d,n)}}{V\beta + n_{t,(\cdot)}^{-(d,n)}}. \tag{5.13}
$$

Now we need to add one more discrete variable in this model and form LDA based on Gibbs Sampler had two variables. And we get equation $P(\mathbf{Z}|\mathbf{W}, \mathbf{L})$

$$
P(\mathbf{Z}|\mathbf{W}, \mathbf{L}) = P\left(\mathbf{W}|\mathbf{Z}\right) \times P\left(\mathbf{Z}\right) \times P\left(\mathbf{L}|\mathbf{W}, \mathbf{Z}\right). \tag{5.14}
$$

Through clustering words by $P\left(\mathbf{W}|\mathbf{Z}\right)$ and $P\left(\mathbf{L}|\mathbf{Z}\right)$, and getting that $P\left(\mathbf{L}|\mathbf{W}, \mathbf{Z}\right) = P\left(\mathbf{L}|\mathbf{Z}\right)$. Therefore, equation.(5.13) can be express as follow

$$
P\left(\mathbf{W}|\mathbf{Z}\right) \times P\left(\mathbf{Z}\right) \times P\left(\mathbf{L}|\mathbf{W}, \mathbf{Z}\right) = P\left(\mathbf{W}|\mathbf{Z}\right) \times P\left(\mathbf{Z}\right) \times P\left(\mathbf{L}|\mathbf{Z}\right) \tag{5.15}
$$

Through analyzing this formula, the equation with two available based on Gibbs Sampler is

$$
P(Z_k|\mathbf{Z}_{-k}, \mathbf{W}, \mathbf{L}) = \frac{n_{(\cdot),v}^{k} + \beta}{\sum_{r=1}^{V_{ASR}} n_{(\cdot),r}^{k} + \beta} \frac{n_{(\cdot),v}^{k} + \lambda}{\sum_{r=1}^{V_{\mathrm{OCR}}} n_{(\cdot),r}^{k} + \lambda} n_{u,(\cdot)}^{k} + \alpha \tag{5.16}
$$

5.2 Dual-variable LDA model for Trajectory Clustering

After multi-modal LDA model derivation, we come to derive dual-variable LDA model which contains continues features and discrete features. From the above section, we can derive a novel Gibbs sampling based LDA model, so we further derive a LDA model to process two different features in this section.

As a generative model, topics generate words in LDA model, and words can be allocated to categories by posterior distribution. Therefore, as the continues features, the coefficients of DFT are real vectors, so we need to find a proper distribution to character the vectors. In our opinion, we use Multivariate distribution to describe real vector, and Normal-inverse-Wischart distribution is used as posterior distribution to inference the categories of words. The parameters of Normal-inverse-Wischart distribution includes λ, ν and p, they corresponds to $L+m$ where m is the counting of samples and it also presented as ν, p indicates L dimensions. In the following content, we'll discuss and derive our model.

$P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right)$ is the only term we need to concern which indicates the generation of word n belonging to document d, and the probability of the other words $P\left(W_{-(d,n)}|Z_{-(d,n)}, Z_{d,n}=t, \mu_k, \Sigma_k\right)$ follows Multivariate distribution. Furthermore, the parameters μ_k and Σ_k are generated by the vocabularies belonging the corresponding topics. Therefore, the derivation we can get is as follows

$$
\begin{aligned}
& P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right) \\
=& \int_{\mu}\int_{\Sigma} P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t, \mu, \Sigma\right) \cdot P\left(\mu, \Sigma|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right) d\Sigma d\mu \\
=& \prod_{k}\prod_{k}\int_{\mu_k}\int_{\Sigma_k} P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t, \mu_k, \Sigma_k\right) \\
& \cdot P\left(\mu_k, \Sigma_k|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right) d\Sigma_k d\mu_k \\
=& \prod_{k}\prod_{k}\int_{\mu_k}\int_{\Sigma_k} P\left(W_r|Z_r=t, \mu_t, \Sigma_t\right) \cdot P\left(\mu_k, \Sigma_k|W_{-r}, Z_{-r}, Z_r=t\right) d\Sigma_k d\mu_k \\
\propto& \prod_{k}\prod_{k}\int_{\mu_k}\int_{\Sigma_k} \mathcal{N}\left(\mu_t, \Sigma_t\right) \cdot \left[P\left(W_{-r}|Z_{-r}, Z_r=t, \mu_k, \Sigma_k\right) \cdot P\left(\mu_k, \Sigma_k\right)\right] d\Sigma_k d\mu_k
\end{aligned}
$$

$$
\begin{aligned}
&= \prod_{Z_v} \prod_{Z_v} \int_{\mu_{Z_v}} \int_{\Sigma_{Z_v}} \mathcal{N}\left(\mu_{Z_r=t}, \Sigma_{Z_r=t}\right) \cdot \left[\prod_{v=1, v\neq r}^{V} \mathcal{N}\left(\mu_{Z_v}, \Sigma_{Z_v}\right) \cdot P\left(\mu_{Z_v} | \Sigma_{Z_v}\right) \cdot P\left(\Sigma_{Z_v}\right) \right] d\Sigma_{Z_v} d\mu_{Z_v} \\
&= \prod_{Z_v} \prod_{Z_v} \int_{\mu_{Z_v}} \int_{\Sigma_{Z_v}} \prod_{v=1}^{V} \mathcal{N}\left(\mu_{Z_v}, \Sigma_{Z_v}\right) \cdot P\left(\mu_{Z_v} | \Sigma_{Z_v}\right) \cdot P\left(\Sigma_{Z_v}\right) d\Sigma_{Z_v} d\mu_{Z_v} \\
&= \prod_{Z_v} \prod_{Z_v} \int_{\mu_{Z_v}} \int_{\Sigma_{Z_v}} \prod_{v=1}^{V} \mathcal{N}\left(\mu_{Z_v}, \Sigma_{Z_v}\right) \cdot \mathcal{N}\left(\mu_{Z_v}, \frac{1}{L + m_{Z_v}} \Sigma_{Z_v}\right) \cdot \mathcal{W}^{-1}\left(\Psi_{Z_v}, \nu + m_{Z_v}\right) d\Sigma_{Z_v} d\mu_{Z_v}.
\end{aligned}
\tag{5.17}
$$

It should be noted that, in generative model, Z_r generates W_r only and only vth vocabulary is considering. Furthermore, Z_v are all fixed here, and $k = t$ only when $r = v$. In order to simplified the equation, so we assume that $Z_v = k$ which indicates kth topic. Furthermore, μ_k is L dimensional mean vector and Σ_k is $L \times L$ covariance matrix, m_k indicates the counting of vocabularies classified into kth topic. Therefore,

$$
\begin{cases}
\mathcal{N}\left(\mu_k, \Sigma_k\right) = \det\left(2\pi\Sigma_k\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2} \left(W_{r:Z_r=k} - \mu_k\right)^{\top} \Sigma_k^{-1} \left(W_{r:Z_r=k} - \mu_k\right) \right] \\
\mathcal{W}^{-1}\left(\Psi_k, \nu + m_k\right) = \dfrac{\left|\Psi_k\right|^{\frac{\nu+m_k}{2}}}{2^{\frac{L(\nu+m_k)}{2}} \cdot \Gamma\left(\frac{\nu+m_k}{2}\right)} \left|\Sigma_k\right|^{-\frac{L+\nu+m_k+1}{2}} \exp\left(-\frac{1}{2} tr\left(\Psi_k \Sigma_k^{-1}\right) \right)
\end{cases}
$$

Based on Bayes' Theorem, $P\left(B|A\right) \propto P\left(B\right) \cdot P\left(A|B\right)$, so we can obtain the equation as follows,

$$
\begin{aligned}
&P\left(W_{d,n} | W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t, \mu, \Sigma\right) = \\
&\frac{P\left(W_{d,n} | W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t\right) \cdot P\left(\mu, \Sigma | W_{d,n}, W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t\right)}{P\left(\mu, \Sigma | W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t\right)}
\end{aligned}
\tag{5.18}
$$

where $W_{d,n}$ is the L-dimensional vector. $P\left(\mu, \Sigma | W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t\right)$ is characterized by Normal-inverse-wishart distribution, and μ, Σ are determined by words $W_{d,n}$. However, in the denominator of equation.5.18, $W_{-(d,n)}$ and $Z_{-(d,n)}$ are fixed, $W_{d,n}$ is not involved in, so the denominator is fixed value. From equation.(5.18), it can be rewritten as

$$P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right) \propto \frac{P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t, \mu, \Sigma\right)}{P\left(\mu, \Sigma|W_{d,n}, W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right)}. \tag{5.19}$$

Then, we need to derive two terms in the right hand of the equation. For the numerator, $P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t, \mu, \Sigma\right)$, it characterizes as Multivariate distribution with mean μ and variance Σ as

$$\begin{aligned} &P\left(W_{d,n}|W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t, \mu, \Sigma\right) \\ &= \prod_r \prod_k \det\left(2\pi\Sigma_k^{liter}\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(W_r-\mu_k^{liter}\right)^\top \cdot \Sigma_k^{-1} \cdot \left(W_r-\mu_k^{liter}\right)\right] \end{aligned} \tag{5.20}$$

where $\mu_k = \mu_t$ only when $r = v$, W_r indicates rth vocabulary in the dictionary, k is topic index and v is the index of the current vocabulary.

For the denominator, $P\left(\mu, \Sigma|W_{d,n}, W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right)$, it follows Normal-inverse-wishart distribution. In details, Σ is sampled from inverse Wishart distribution with parameters Ψ and ν, and μ is sampled from Multivariate distribution with mean μ^0 and variance $\frac{1}{\lambda}\Sigma$

$$\begin{aligned} &P\left(\mu, \Sigma|W_{d,n}, W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n}=t\right) \\ &= \prod_r \prod_k h\left(\nu_k^{liter}, L\right) |\Sigma_k^{liter}|^{-\frac{\nu_k^{liter}+L+1}{2}} \exp\left[-\frac{1}{2} tr\left(\Psi_k^{liter} \cdot \left(\Sigma_k^{liter}\right)^{-1}\right)\right] \cdot \\ &\det\left(2\pi \cdot \frac{1}{\lambda_k^{liter}}\Sigma_k^{liter}\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\mu_k^{liter}-\mu_k^0\right)^\top \cdot \frac{1}{\lambda_k^{liter}}\left(\Sigma_k^{liter}\right)^{-1} \cdot \left(\mu_k^{liter}-\mu_k^0\right)\right] \end{aligned} \tag{5.21}$$

where $\lambda_k^{liter} = \lambda + m_k$, $\nu_k^{liter} = \nu + m_k$ and m_k indicates the counting of samples belonging to kth topic. $h\left(\nu, L\right) = \frac{|\Psi|^{\frac{\nu}{2}}}{2^{\frac{\nu L}{2}}\Gamma_L\left(\frac{\nu}{2}\right)}$ where $\Gamma\left(\cdot\right)$ is the multivariate gamma function. Furthermore, in equation.(5.20) and equation.(5.21), μ and Σ are updated as

$$\begin{cases} \mu_k^{liter} & = \dfrac{\lambda_k \mu + m_k \overline{W}_{r:k}}{\lambda_k + m_k}, \\ \Sigma_k^{liter} & = \dfrac{\Psi_k^{liter}}{(\nu_k - L + 1)}. \end{cases}$$

where $\Psi_k^{liter} = \Psi + S_k + \frac{\lambda_k m_k}{\lambda_k + m_k} \left(\overline{W}_{r:Z_r=k} - \mu_k^0\right) \left(\overline{W}_{r:Z_r=k} - \mu_k^0\right)^\top$, and $S_k = \sum_{r=1}^{V} \left(W_{r:Z_{r=k}} - \overline{W}_{r:Z_r=k}\right)^\top \left(W_{r:Z_{r=k}} - \overline{W}_{r:Z_r=k}\right)$

From equation.(5.20) and equation.(5.21),

$$\begin{aligned} & P\left(W_{d,n} | W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t\right) \\ & = \prod_r \prod_k \det\left(2\pi \Sigma_k^{liter}\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(W_r - \mu_k^{liter}\right)^\top \cdot \left(\Sigma_k^{liter}\right)^{-1} \cdot \left(W_r - \mu_k^{liter}\right)\right] \cdot \\ & \quad \left\{\prod_r \prod_k h\left(\nu_k^{liter}, L\right) |\Sigma_k^{liter}|^{-\frac{\nu_k^{liter}+L+1}{2}} \exp\left[-\frac{1}{2} tr\left(\Psi_k^{liter} \cdot \left(\Sigma_k^{liter}\right)^{-1}\right)\right]\right\}^{-1} \cdot \\ & \quad \left\{\prod_r \prod_k \det\left(2\pi \cdot \frac{1}{\lambda_k^{liter}} \Sigma_k^{liter}\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\mu_k^{liter} - \mu_k^0\right)^\top \cdot \frac{1}{\lambda_k^{liter}} \left(\Sigma_k^{liter}\right)^{-1} \cdot \left(\mu_k^{liter} - \mu_k^0\right)\right]\right\}^{-1} \\ & = \prod_r \left(\frac{1}{\lambda_t^{liter}}\right)^{\frac{L}{2}} \cdot f_t^{-1} \cdot \exp \\ & \left[-\frac{2\lambda_t^{liter}}{\Psi_t^{liter}} tr^{-1}\left(\Sigma_t^{liter}\right)\left(W_{r:Z_r=t} - \mu_t^{liter}\right)^\top \left(W_{r:Z_r=t} - \mu_t^{liter}\right) \cdot \left(\left(\mu_t^{liter} - \mu_t^0\right)^{-1}\right)^\top \left(\mu_t^{liter} - \mu_t^0\right)^{-1}\right] \end{aligned}$$

It should be noticed that only the vocabularies classified to tth topic are concerned in equation, so the others can be dropped. In order to combine two factors together, we need to derive the joint probability distribution. Based on Gibbs sampling,

$$\begin{aligned} P\left(Z_{d,n} = t | W_{d,n}, Z_{-(d,n)}, Z_{d,n}\right) \propto & P\left(Z_{d,n} = t | W_{-(d,n)}, Z_{-(d,n)}\right) \\ & \cdot P\left(W_{d,n} | W_{-(d,n)}, Z_{-(d,n)}, Z_{d,n} = t\right). \end{aligned}$$

Two factors are substituted into equation,

$$\begin{aligned}
&P\left(Z_{d,n}=t|W_{d,n},Z_{-(d,n)},Z_{d,n}\right)\\
\propto&P\left(Z_{d,n}=t|W_{-(d,n)},Z_{-(d,n)}\right)\cdot P\left(W_{d,n}|W_{-(d,n)},Z_{-(d,n)},Z_{d,n}=t\right)\\
=&P\left(Z_{d,n}=t|Z_{-(d,n)}\right)\cdot P\left(W_{d,n}^{1},W_{d,n}^{2}|W_{-(d,n)}^{1},W_{-(d,n)}^{2},Z_{-(d,n)},Z_{d,n}=t\right) \quad (5.22)\\
=&P\left(Z_{d,n}=t|Z_{-(d,n)}\right)\cdot P\left(W_{d,n}^{1}|W_{-(d,n)}^{1},W_{-(d,n)}^{2},Z_{-(d,n)},Z_{d,n}=t\right)\\
&\cdot P\left(W_{d,n}^{2}|W_{-(d,n)}^{1},W_{-(d,n)}^{2},Z_{-(d,n)},Z_{d,n}=t\right)
\end{aligned}$$

We need to apply the properties of conditional probability distribution here. Assume that two event sets and B are independent,

$$P\left(A_i,B_{-i}|A_{-i}\right)=P\left(A_i|A_{-i}\right)\cdot P\left(B_{-i}|A_{-i}\right).$$

For the formula of conditional probability distribution,

$$P\left(A_i,B_{-i}|A_{-i}\right)=P\left(A_i|A_{-i},B_{-i}\right)\cdot P\left(B_{-i}|A_{-i}\right).$$

From the above two equation, we can obtain that $P\left(A_i|A_{-i}\right)=P\left(A_i|A_{-i},B_{-i}\right)$, thus, equation.(5.22) can be rewritten as

$$P\left(Z_{d,n}=t|Z_{-(d,n)}\right)\cdot P\left(W_{d,n}^{1}|W_{-(d,n)}^{1},Z_{-(d,n)},Z_{d,n}=t\right)\cdot P\left(W_{d,n}^{2}|W_{-(d,n)}^{2},Z_{-(d,n)},Z_{d,n}=t\right)$$

After Gibbs sampling based LDA model for continues features are obtained, we discuss dual-variable LDA model for trajectory clustering as follows. Trajectory is generated by recording the positions of moving object in fixed time period. However, the length of trajectory may be different from the others. Thus, trajectories can be transformed into frequency domains and represented by a fixed number of coefficients. In computer version, DFT aims to describe the degree of gray scale change of image, and considers the relationship between arbitrary pair of trajectory points. Therefore, DFT is used to characterize the positional variation of trajectories and its coefficients represent trajectory by denoting as $w_{l,j}$, where l indicates frequency domain index and j donates trajectory index. DFT coefficients belonging to same frequency domain comprise a independent vocabulary. The original vocabulary which comprised by N words instance is substituted

by new vocabularies in our model, so that L_1 dimensional vector represents trajectory in our model. (Note: other features should be added and discussed here. Speed and direction can be extracted and added. I intend to employ object movement condition as another feature. In details, the scene is split into 3×3 patches and the trajectory is determined as moving relative to the camera if the first point and the last point are belonging to different patches.) However, the dimension of L_1 is increased if more continuous features are added. $L = L_1 + L_2$ where L_2 is the dimensions of discrete features. One of the vocabularies are constructed as follows

$$\mathbf{w_j} = [\mathbf{w_{1,j}}, \mathbf{w_{2,j}}, \cdots, \mathbf{w_{L,j}}]$$

where j is word index. In $\mathbf{w_j}$, DFT coefficients are continuous and the relative motion is discrete feature. Therefore, DFT coefficients are represented by Multivariate distribution because Gaussian distribution defines continuous feature conditioned on the topic Z_j as $P(w_{l_1,j}|Z_j = t) = \mathcal{N}(\mu, \Sigma)$. Multinomial distribution describes the relative motion in dual-variable LDA model. The graphical representation is shown in Fig.5.4. φ denotes multinomial distribution representing each topic is characterized by discrete *words* and the size of φ is $T \times N \times L_1$ where N is the amount of trajectories and T indicates the total number of topics. Each continuous feature follows Multivariate distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$. It should be noticed that the parameters are unknown in generative process, so the prior are $\mu \sim \mathbf{Multivariate}\left(\mu_\mathbf{0}, \kappa_\mathbf{0}^{-1}\mathbf{\Sigma}\right)$ and $\Sigma \sim Wishart^{-1}(\mathbf{\Psi}, \nu_\mathbf{0})$ where $\mu_\mathbf{0}$ is the mean of κ_0 observations, and Ψ indicates the sum of pairwise deviation products of ν_0 observations, $\Psi = \nu_0\Sigma_0$. The documents are represented by the mixtures of latent topics and they still be characterized by multinomial distribution. All features are separated as discrete features W_1 and continuous features W_2, and the generative process of dual-variable LDA model are presented as follows: (Note: covariance matrix is critical if multi-dimensional continuous features are involved, and the same situation should be also considered when it happened in discrete circumstances.)

1. Choose $\theta_i \sim Dir(\alpha)$, where $i \in \{1, \cdots, M\}$ and $Dir(\alpha)$ is Dirichlet Distribution.

2. For each discrete feature $vectorl_1$ of W_1, choose $\varphi_{\mathbf{l_1,t}} \sim \mathbf{Dir}(\beta_{\mathbf{l_1}})$, where $t \in$

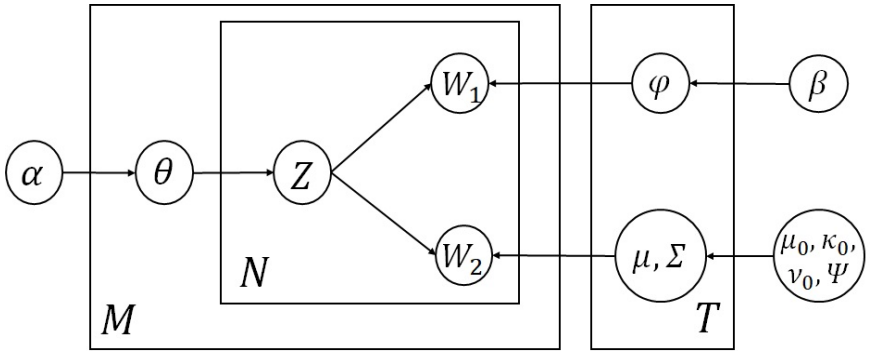

Figure 5.4: Graphical representation of dual-variable LDA model.

$\{1, \cdots, T\}$ and $l_1 \in [1, \cdots, L_1]$.

3. For each continuous feature vector l_2 of W_2, choose $\mu_{\mathbf{l_2}} \sim \mathbf{Multivariate}\left(\mu_{\mathbf{0}}^{\mathbf{l_2}}, \left(\kappa_{\mathbf{0}}^{\mathbf{l_2}}\right)^{-1} \mathbf{\Sigma_{l_2}}\right)$, $\Sigma_{l_2} \sim Wishart^{-1}\left(\mathbf{\Psi_{l_2}}, \nu_{\mathbf{0}}^{\mathbf{l_2}}\right)$ and $l_2 \in [L_1 + 1, L_1 + L_2]$.

4. For each feature of each of the N words $w_{l,j}$:

4.1. Choose $Z_j \sim Multinomial\left(\boldsymbol{\theta_i}\right)$.

4.2. Choose a discrete feature of word $w_{l_1,j}$, it draws from $Multinomial\left(\boldsymbol{\varphi_{l_1,Z_j}}\right)$.

4.3. Choose a continuous feature of word $w_{l_2,j}$, it draws from Gaussian distribution $\mathcal{N}\left(\mu_{l_2,Z_j}, \Sigma_{l_2,Z_jZ_j}\right)$.

5.3 Experiments

5.3.1 Multi-modal LDA model

In our experiment, we choose 100 lecture videos from *http://videolectures.net* to have an experiment, and these 100 videos have different titles which focus on Beyesian Process or Dirichlet Process in Machine Learning, which means the words relating to probability knowledge, characterizing method and describing models are involved. Then, we segment them into many shots based on time, for example, we segment a 20-mins video into 20 shots that each shot has 1 minute. Therefore, we totally get 1640 shots and 11000 key frames from these 100 videos. After that, we use ASR and OCR to extract semantic information from these

key frames. The words extracted by ASR and OCR are treated as two kinds of evidences.

Since most existed OCR tools focus on recognize characters instead of words in OCR, a word will be recognized incorrectly even if there is a character in the word is recognized incorrectly. Thus, we need do some spelling check and correction operation after OCR extraction.

In ASR, some speakers teach their lectures with their assent, relating to different countries or different regions, for example, someone speak "dream" as "doraemon"' and "very" as "vely". Besides that, speakers say some prepositions such as "is", "and", "or", so we also need to delete these words and the similar ones after extraction step.

To address these problems, we build up a stop-word dictionary as follows: firstly, we find a stop-word list on the network as initial dictionary; and then through analyzing the extracting information, the words such as "enough" and "local", can be added to dictionary. Repeating this program one by one, we can get dictionary and use it as filter to shield words that have no meaning for discovery knowledge to classify shots.

Now we take experiment with extracting semantic information to discovery knowledge and classify these shots into different topics. The dataset of clustering is too large, so we show a part of them.

In Table.5.2 and Table.5.3, we can find some words like "cftp" is not a word

Table 5.2: Extracting words from the first lecture video.

	ASR	OCR
Extracting words	acknowledged	approach
	act	approaches
	action	bayesian
	active	beliefs
	actual	borrow
	added	borrow
	additional	cftp

that has some mean. Therefore, we state that "cftp" means "coupling from the

Table 5.3: Extracting words from the second lecture video.

	ASR	OCR
Extracting words	act	applicable
	additional	bayesian
	africa	bound
	against	bounds
	ago	cholesky
	ahead	concave
	analyst	concavity

past" and it occurs in the sentence that, *We borrow methods for inference UGMs: CFTP in Ising and Potts models.* Thus, we need to note that this type of word means Abbreviation.

Because they have different titles mainly focus on 3 fields, so we set $T = 3$, which means we should classify 100 videos include 11000 key frames into 3 topics. Then, we adjust the dimension of vocabulary and topic matrix, so we can index vocabulary and word number clearly. α, λ are the assuming symmetric Dirichlet priors, we set $\alpha = 5/T$, $\beta = 0.01$ and $\lambda = 2$.

In case that each topic contains a lot of words, so we only list first 5 words

Table 5.4: Result of Multi-modal LDA with Gibbs Sampler

Topic 1	Topic 2	Topic 3
reg	features	gp-lvm
prior	parameter	portfolios
changes	algorithm	performance
extend	rsthq	modular
vary	bars	sensitive

of each topic. Furthermore, we compare the classification and the corresponding probabilities with the category of lecture videos. In Table.5.4, we can find that the words in Topic 1 discuss probability distribution, the words in Topic 2 focus on the feature of model or characterizing the objects, and Topic 3 involves in

analyzing experiment results and performance of models in science field. It fits the categories of lecture video on website. Then, we can prove multi-modal LDA model, processing with two variables with Gibbs Sampler, is useful.

Based on these topics and through analyzing the meaning of every topic, we can define topics and may be one topic can be given two or three words for indexing. Furthermore, this method can be used in some other applications involving different features and semantic topics are needed. Users can browse videos through searching these words, and scanning lecture videos directly.

5.3.2 Dual-variable LDA model

In our experiment, we apply the trajectory data generated from KITTI benchmark in the previous sections. All trajectory data are generated by "van", "cyclist", "pedestrian", "car" and "misc" which means miscellaneous objects in data set such as traffic lights, road signs and trees on roadside. The aim of following experiment is evaluated with traditional LDA model [12], the derived LDA model and dual-variable LDA model. We set parameters in same setting, where $\alpha = 5/T$, $\beta = 0.1$, $\mu_0 = 1$, $\kappa_0 = 5$, $\nu_0 = 5$ and iterate sampling step for 50 times. Furthermore, there are a lot of trajectory data have same labels, so we integrate them as the unique one.

We take an experiment on Sequence.00 of training data set of KITTI benchmark, and the results are displayed in the following table,

Table 5.5: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.00 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.275182		0.283880		0.244036		0.196902
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.797675	Misc	0.744048	Misc	0.814285	Misc	0.679984
	Van	0.153095	Cyclist	0.155845	Van	0.114583	Van	0.242373
	Cyclist	0.049227	Van	0.096456	Cyclist	0.071128	Cyclist	0.077638
	Pedestrian	0.000003	Pedestrian	0.003651	Pedestrian	0.000003	Pedestrian	0.000004
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.258491		0.247391		0.246148		0.247971
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Cyclist	0.318233	Cyclist	0.310658	Misc	0.377422	Misc	0.293645
	Misc	0.313821	Misc	0.296433	Cyclist	0.343025	Van	0.28901
	Pedestrian	0.316311	Van	0.256301	Van	0.265028	Cyclist	0.26392
	Van	0.051735	Pedestrian	0.136508	Pedestrian	0.014427	Pedestrian	0.152535
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.267853		0.273746		0.254126		0.236712
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.451273	Misc	0.662023	Misc	0.475956	Misc	0.334036
	Cyclist	0.419016	Cyclist	0.22273	Van	0.348032	Van	0.32153
	Van	0.109403	Van	0.078147	Cyclist	0.15881	Cyclist	0.184327
	Pedestrian	0.020304	Pedestrian	0.0371	Pedestrian	0.017302	Pedestrian	0.160303

In Table.5.5, the probabilities of the assigned words which belonging to same group are summed together to check the performance of clustering. For example, the trajectories of "Misc" in Topic 1 are summed up as 79.76% by applying traditional LDA model. θ gives a probability of topic in the document which denotes as vocabulary in our experiment, because all the *words* are set in one document. Therefore, θ indicates the confidence of the corresponding topic determination. Traditional LDA model considers discrete feature as visual words, and derived LDA model could employ continues feature which is motion information. We can see that "misc" is set as priority by traditional LDA model and "cyclist" is ranked as the first one by derived LDA model, which means continue feature can get better performance than discrete feature for trajectory clustering. It proves that continues feature presents more characterized content, and traditional LDA only focus on the limited feature. However, derived LDA model focuses on continues features and proves the performance of classification, because continues feature contains more robust and useful characterization than discrete feature. From Table.5.5, "misc" generating too much trajectory data so they are the noisy of trajectory data in sequence 00. Therefore, after accumulating the probabilities belonging to the corresponding objects, the trajectories labeled as "misc" are set as the priority in some topics. By derived LDA model, "Cyclist" ranks as the first one in two topics, and the differences between the probability of "Misc" and others are smaller than the ones in traditional LDA model. By dual-variable LDA model, although "misc" still ranks as first classification in each topic, their probabilities are lower relative to the others by comparing the experiment results in traditional LDA. Thus, it also proves that continues feature is critical to trajectory clustering. For the trajectories labeled as "pedestrian", only 25 trajectories and 24140 trajectories are generated totally, so they are much less than the amount of the arbitrary ones. Therefore, the probabilities of trajectory in pedestrian topic are computed as the lowest *words* in each topic. For other 20 benchmark of KITTI, their experiment results can be found in Appendix.A.

5.4 Conclusion

In this chapter, we propose a novel LDA model to process two different features and classify the objects into categories, we call the method as dual-variable LDA model. This model is specifically designed for process continue feature and discrete feature, respectively, and combine two probabilities of the corresponding same object as the result. In doing so, the spatiotemporal information of trajectory data are used up to analyze the properties of objects. To illustrate how performance our model has, we firstly derive multi-modal LDA model and take an experiment on lecture video and prove its accuracy. Then, dual-variable LDA model is derived and it proved that the model is superior than single variable LDA model on trajectory data analysis. However, there is a issue need to be fixed that it is not too robust to tolerate much noise. In future work, we will focus on optimizing trajectory data construction, repeating and redundant trajectories are dropped out. Furthermore, a supervised LDA model is needed to be derived to improve the performance.

Chapter 6

Conclusions

In this thesis, we exploit a novel method to cluster trajectory data and lecture videos.

We saw that how to measure trajectories with different lengths is important. Thus, a representation method or feature descriptor is essential for trajectory clustering. In recent years, transforming trajectory data into other space is paid more attentions, such as DFT which keeping data information and unifying lengths of trajectory data [56]. For other preparation works, re-sampling is efficient for sparse scene [140], but it limits the robustness of model. Curve approximation fits the movement of trajectory [120] [158]. Hence, trajectory data preparation may be a promising and helpful direction.

Recently, Densely Clustering models have achieved great progress in trajectory clustering. In particular, novel distance metrics have been proposed to measure trajectory data according to different properties. Furthermore, for the trajectory data with large difference in density, grid construction is employed to improve the performance [131]. Besides grid-based DBCSAN, sub-trajectories are acted as the substitutes for trajectory in [64], [75] and [76].

Though Spectral Clustering models and Graph method share a similar idea, they are intrinsically different. Spectral Clustering models are easy to implement and have no restriction on data dimensions, but the models require non-negative affinities and this limitation restricts the performance and the application. Therefore, a suitable affinity matrix construction method is needed. Furthermore, it is critical to determine scale value when the affinity matrix is being computed,

because it determines the clustering results are efficient or not. Thus, Spectral Clustering models need to handle the problem of constructing affinity matrix.

In supervised algorithms, a large number of training data are required to obtain an efficient model. However, such as in Neural Network, there may have overfitting problem and some special steps are needed like pooling layers in CNN. In addition, it should be noticed that a meaningful distance metric is essential for Nearest Neighbor algorithms.

They are the possible direction on improving trajectory clustering by other models. In the thesis, we first reviewed the current popular cluster models including the unsupervised ones, the supervised ones and semi-supervised ones. Furthermore, different trajectory representations are discussed, either. After comparing different clustering models, unsupervised model is proper in our application, because no training data is involved here, and the representation by transforming to another space is chosen in the case that much more information are useful and powerful to recognize the inner connection between different trajectories.

For trajectory clustering, the first step of the method is the trajectory generation step. Here, a SURF detector extracts multiple characteristic points for each object. Then, a SIFT descriptor tracks these characteristic points by matching them across several consecutive frames. Experiments demonstrate good performance in both representing and tracking objects.

Next, a DFT algorithm is used to reveal spatiotemporal information about the trajectories. Each trajectory is transformed into the same length and its original information is preserved. The process is fast and efficient. Further, relative motion is incorporated into the model to distinguish still objects from moving objects – i.e., still objects have the same trajectory as a moving ego-platform.

Lastly, two novel LDA models that derived LDA model process continue feature to cluster lecture video data, and dual-variable LDA model process two different feature types to classify trajectory data. In details, Gibbs sampling is used to improve the clustering performance because of good clustering performance of MCMC algorithm. These methods not only quickly and efficiently clusters the trajectories, they also consider the properties of the trajectory data and outperform traditional LDA models.

The implementation is inspired by the previous works, and derives a more

advanced model to improve the performance and fit to our application. From the experimental results, the algorithms presented in this thesis are effective and robust for clustering lecture videos and video trajectory data. They can be used to cluster data depending on continue feature or two different kinds of features, such like videos. Hence, they inspire several possible directions for further improving the performance. For example, different parameters may not follow one Dirichlet Distribution, so more complicated distribution can be employed, and the trajectory generation step could be improved to solve noise issues better. Furthermore, deep learning methods are state-of-art models on extracting features and classification, so that is another future work to be done.

Appendix A

Results of Video Trajectory Clustering

Table A.1: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.01 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.272991		0.255502		0.196093		0.275413
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.719238	Misc	0.671156	Misc	0.632166	Misc	0.657999
	Car	0.280758	Car	0.320202	Car	0.367828	Car	0.341997
	Van	0.000004	Van	0.008641	Van	0.000005	Van	0.000004
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.258032		0.247161		0.247125		0.247592
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.433622	Misc	0.453977	Misc	0.608528	Misc	0.50927
	Car	0.407753	Car	0.437517	Car	0.31602	Van	0.296012
	Van	0.158625	Van	0.108506	Van	0.075452	Car	0.194708
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.271011		0.256312		0.200311		0.267832
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.496827	Misc	0.37631	Misc	0.655177	Misc	0.6215
	Car	0.334628	Car	0.349422	Car	0.178821	Car	0.190477
	Van	0.168645	Van	0.274278	Van	0.166011	Van	0.188023

Table A.2: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.02 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.279467		0.204508		0.299836		0.216189
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.952085	Misc	0.972702	Misc	0.884067	Misc	0.907368
	Car	0.047902	Pedestrian	0.017519	Car	0.097907	Car	0.092616
	Pedestrian	0.000004	Car	0.009767	Cyclist	0.016647	Pedestrian	0.000005
	Van	0.000004	Van	0.000006	Van	0.001375	Van	0.000005
	Cyclist	0.000004	Cyclist	0.000006	Pedestrian	0.000004	Cyclist	0.000005
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.246772		0.251291		0.248004		0.253933
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.366717	Misc	0.329321	Misc	0.296331	Misc	0.411528
	Car	0.321453	Pedestrian	0.244227	Car	0.244637	Car	0.310019
	Van	0.13811	Car	0.187633	Van	0.189122	Pedestrian	0.207733
	Pedestrian	0.09611	Van	0.13941	Cyclist	0.177605	Van	0.04141
	Cyclist	0.07761	Cyclist	0.099509	Pedestrian	0.092305	Cyclist	0.02951
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.265431		0.234621		0.28993		0.223762
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.317326	Misc	0.447437	Misc	0.32181	Misc	0.284929
	Car	0.264857	Pedestrian	0.264529	Car	0.282634	Car	0.262544
	Van	0.19331	Car	0.189919	Van	0.166451	Pedestrian	0.24912
	Pedestrian	0.180404	Van	0.091412	Cyclist	0.140602	Van	0.171601
	Cyclist	0.044103	Pedestrian	0.006703	Pedestrian	0.088603	Cyclist	0.031902

Table A.3: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.03 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.30256		0.254008		0.24417		0.199262
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.831675	Misc	0.840026	Misc	0.955034	Misc	0.812297
	Car	0.168325	Car	0.159974	Car	0.044966	Car	0.187703
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.248042		0.24836		0.250273		0.253325
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.799459	Car	0.844197	Car	0.649135	Misc	0.849922
	Misc	0.200541	Misc	0.155803	Misc	0.350864	Car	0.150177
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.257732		0.251101		0.239981		0.207364
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.900133	Misc	0.994037	Misc	0.51516	Misc	0.825536
	Car	0.099867	Car	0.005962	Car	0.484839	Car	0.174464

Table A.4: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.04 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.244526		0.19254		0.272554		0.290381
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.784396	Car	0.601438	Car	0.805815	Van	0.681272
	Car	0.215596	Van	0.323607	Misc	0.187182	Misc	0.229459
	Van	0.000004	Misc	0.057552	Van	0.006999	Car	0.089266
	Tram	0.000004	Tram	0.017403	Tram	0.000004	Tram	0.000004
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.252512		0.246896		0.248957		0.251636
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.379053	Car	0.362606	Misc	0.477732	Misc	0.468353
	Misc	0.333126	Misc	0.250053	Car	0.285307	Van	0.269431
	Van	0.179609	Van	0.20014	Van	0.146653	Car	0.206512
	Tram	0.108213	Tram	0.187101	Tram	0.090308	Tram	0.055704
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.243361		0.200335		0.267732		0.273365
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.334039	Car	0.523718	Car	0.429738	Van	0.449523
	Car	0.319255	Van	0.21493	Misc	0.214747	Misc	0.443065
	Van	0.173604	Misc	0.196642	Van	0.222513	Car	0.062111
	Tram	0.173004	Tram	0.06471	Tram	0.133102	Tram	0.045401

Table A.5: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.05 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.23775		0.291679		0.254033		0.216537
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.94122	Misc	0.941333	Misc	0.998224	Car	0.886807
	Car	0.058773	Car	0.058662	Car	0.001771	Misc	0.102834
	Cyclist	0.000006	Cyclist	0.000005	Cyclist	0.000006	Cyclist	0.010359
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.250224		0.252689		0.247759		0.248328
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.661223	Misc	0.690108	Car	0.753733	Misc	0.399708
	Car	0.269909	Car	0.296863	Misc	0.196246	Car	0.353967
	Cyclist	0.068767	Cyclist	0.012929	Cyclist	0.050021	Cyclist	0.246425
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.247726		0.273312		0.250322		0.230153
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.468834	Misc	0.648326	Misc	0.439609	Car	0.520234
	Car	0.344507	Car	0.178455	Car	0.352681	Misc	0.434327
	Cyclist	0.186659	Cyclist	0.173219	Cyclist	0.20771	Cyclist	0.045439

Table A.6: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.06 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.245301		0.201811		0.301002		0.251886
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.606034	Car	0.519720	Car	0.452801	Car	0.599446
	Car	0.345831	Misc	0.404759	Misc	0.41541	Misc	0.314987
	Truck	0.048135	Truck	0.075521	Truck	0.131788	Truck	0.085567
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.255453		0.247771		0.244204		0.252572
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.355434	Car	0.385273	Car	0.362253	Car	0.470931
	Misc	0.326357	Misc	0.383612	Truck	0.352242	Misc	0.309228
	Truck	0.318309	Truck	0.231115	Misc	0.285505	Truck	0.219741
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.255316		0.245713		0.242832		0.256139
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.433577	Car	0.353682	Car	0.480533	Car	0.664927
	Misc	0.404521	Misc	0.333607	Misc	0.307452	Misc	0.181556
	Truck	0.161902	Truck	0.312611	Truck	0.212015	Truck	0.153417

Table A.7: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.07 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.264756		0.231013		0.240138		0.264093
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.803997	Misc	0.761026	Car	0.641834	Misc	0.807407
	Car	0.152628	Car	0.158357	Misc	0.285823	Car	0.14256
	Truck	0.043372	Truck	0.073604	Truck	0.072339	Truck	0.030242
	Van	0.000003	Van	0.007013	Van	0.000003	Van	0.019791
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.245621		0.252134		0.249926		0.252318
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.410977	Car	0.499223	Misc	0.402352	Misc	0.390353
	Car	0.214813	Misc	0.232641	Car	0.371707	Car	0.268211
	Truck	0.2131	Truck	0.168212	Truck	0.180317	Truck	0.172136
	Van	0.16101	Van	0.099934	Van	0.045603	Van	0.16931
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.244812		0.250662		0.251619		0.252907
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.430137	Misc	0.328153	Car	0.679739	Misc	0.340634
	Car	0.364606	Car	0.274105	Misc	0.222841	Car	0.267951
	Truck	0.134132	Truck	0.20603	Truck	0.072603	Truck	0.212212
	Van	0.071036	Van	0.191715	Van	0.024816	Van	0.179202

Table A.8: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.08 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.23521		0.30105		0.268471		0.19527
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.77535	Misc	0.802073	Misc	0.831121	Misc	0.771983
	Van	0.122322	Van	0.187954	Van	0.157696	Van	0.167949
	Truck	0.066381	Car	0.009968	Car	0.011178	Car	0.060061
	Car	0.035947	Truck	0.000005	Truck	0.000005	Truck	0.000007
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.250545		0.244752		0.253749		0.250954
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.436553	Misc	0.334423	Misc	0.406527	Misc	0.346852
	Van	0.247928	Van	0.279357	Van	0.279533	Van	0.336334
	Truck	0.208007	Car	0.262012	Car	0.162821	Car	0.302509
	Car	0.107612	Truck	0.124107	Truck	0.151219	Truck	0.014307
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.251977		0.24557		0.251363		0.251091
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.494065	Misc	0.445681	Misc	0.831121	Misc	0.312491
	Van	0.243827	Van	0.404008	Van	0.157696	Van	0.269407
	Truck	0.1904	Car	0.084401	Car	0.011178	Car	0.247201
	Car	0.071808	Truck	0.06601	Truck	0.000005	Truck	0.170901

Table A.9: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.09 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.363018		0.245167		0.188652		0.203164
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.692574	Misc	0.819827	Misc	0.68363	Misc	0.718214
	Misc	0.307426	Car	0.180183	Car	0.31637	Car	0.281786
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.246203		0.252197		0.249718		0.251882
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.597272	Misc	0.581723	Misc	0.582743	Misc	0.567337
	Car	0.402728	Car	0.418276	Car	0.417256	Car	0.432663
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.246834		0.252918		0.249358		0.25089
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.727153	Misc	0.779309	Misc	0.960527	Misc	0.510233
	Misc	0.272847	Car	0.220691	Car	0.039473	Car	0.489766

Table A.10: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.10 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.209578		0.225042		0.304546		0.260834
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.881533	Misc	0.841173	Misc	0.947767	Misc	0.943794
	Car	0.114587	Car	0.158816	Car	0.048949	Car	0.056197
	Van	0.003875	Van	0.000006	Truck	0.003281	Van	0.000005
	Truck	0.000006	Truck	0.000006	Van	0.000004	Truck	0.000005
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.248613		0.24774		0.251917		0.25173
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.563527	Misc	0.367423	Misc	0.797152	Misc	0.362957
	Car	0.311433	Car	0.267637	Car	0.083533	Car	0.362723
	Truck	0.106819	Truck	0.187952	Truck	0.06971	Van	0.231412
	Van	0.018221	Van	0.176908	Van	0.049605	Truck	0.042908
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.248862		0.247927		0.251668		0.251543
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.352228	Misc	0.338223	Misc	0.559701	Misc	0.378009
	Car	0.321733	Car	0.278364	Car	0.293377	Car	0.365328
	Van	0.252927	Van	0.225702	Truck	0.090321	Van	0.177852
	Truck	0.073112	Truck	0.157811	Van	0.056603	Truck	0.077811

Table A.11: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.11 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.263082		0.303294		0.192061		0.241562
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.710455	Misc	0.714034	Misc	0.768918	Misc	0.714277
	Car	0.239965	Car	0.276636	Car	0.1232	Van	0.149309
	Van	0.042776	Van	0.009326	Pedestrian	0.105172	Car	0.135215
	Pedestrian	0.006804	Pedestrian	0.000004	Van	0.00271	Pedestrian	0.001199
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.251255		0.249639		0.253159		0.245947
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.375326	Misc	0.326927	Misc	0.773037	Misc	0.349753
	Car	0.297038	Car	0.312763	Car	0.131322	Van	0.266127
	Van	0.284522	Van	0.214909	Pedestrian	0.083431	Car	0.242507
	Pedestrian	0.043112	Pedestrian	0.1454	Van	0.01221	Pedestrian	0.141502
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.251601		0.247332		0.253505		0.247562
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.431877	Misc	0.408933	Misc	0.676632	Misc	0.411131
	Car	0.251016	Car	0.291726	Car	0.120748	Van	0.312047
	Van	0.193705	Van	0.173831	Pedestrian	0.115903	Car	0.261713
	Pedestrian	0.123301	Pedestrian	0.12541	Van	0.086706	Pedestrian	0.015209

Table A.12: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.12 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.172181		0.305723		0.190292		0.331804
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.870717	Misc	0.65531	Misc	0.838568	Misc	0.773458
	Cyclist	0.129227	Cyclist	0.33043	Car	0.137241	Car	0.152227
	Car	0.000028	Car	0.024165	Cyclist	0.024165	Cyclist	0.052451
	Pedestrian	0.000028	Pedestrian	0.000016	Pedestrian	0.000025	Pedestrian	0.021863
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.25815		0.26129		0.235209		0.245351
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Cyclist	0.490323	Cyclist	0.420763	Misc	0.394427	Misc	0.330523
	Misc	0.230247	Misc	0.317221	Cyclist	0.350533	Car	0.326941
	Car	0.184311	Car	0.185112	Car	0.156321	Cyclist	0.215723
	Pedestrian	0.095119	Pedestrian	0.076906	Pedestrian	0.098718	Pedestrian	0.126813
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.258875		0.261773		0.231828		0.247525
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.433178	Misc	0.341019	Misc	0.365757	Misc	0.482323
	Cyclist	0.203901	Cyclist	0.286122	Car	0.355022	Car	0.350642
	Car	0.186212	Car	0.253557	Cyclist	0.235712	Cyclist	0.153831
	Pedestrian	0.176809	Pedestrian	0.119402	Pedestrian	0.043509	Pedestrian	0.013304

Table A.13: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.13 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.193596		0.301518		0.22851		0.276376
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.927429	Misc	0.818556	Misc	0.775525	Misc	0.87638
	Pedestrian	0.070836	Pedestrian	0.080725	Pedestrian	0.090949	Cyclist	0.061805
	Cyclist	0.001721	Van	0.060427	Cyclist	0.061049	Person	0.026784
	Person	0.000005	Car	0.027383	Car	0.027827	Van	0.017514
	Car	0.000005	Cyclist	0.012906	Van	0.023467	Pedestrian	0.011162
	Van	0.000005	Person	0.000003	Person	0.021183	Car	0.006355
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.247106		0.249526		0.251328		0.25204
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.218727	Misc	0.324727	Misc	0.279111	Misc	0.335027
	Pedestrian	0.217607	Van	0.263533	Cyclist	0.236437	Cyclist	0.262952
	Cyclist	0.199751	Pedestrian	0.20231	Pedestrian	0.207132	Person	0.182211
	Person	0.187212	Car	0.158612	Car	0.153209	Van	0.114701
	Car	0.147301	Cyclist	0.034108	Van	0.093713	Pedestrian	0.066109
	Van	0.029502	Person	0.01671	Person	0.030407	Car	0.0391
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.247059		0.251376		0.249431		0.252135
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.282438	Misc	0.255617	Misc	0.269227	Misc	0.259431
	Pedestrian	0.170137	Pedestrian	0.252022	Pedestrian	0.260233	Cyclist	0.239922
	Cyclist	0.167212	Van	0.180517	Cyclist	0.258718	Person	0.207607
	Person	0.147607	Car	0.17032	Car	0.098701	Van	0.185121
	Car	0.142005	Cyclist	0.077201	Van	0.086807	Pedestrian	0.105909
	Van	0.0907	Person	0.064321	Person	0.026314	Car	0.00211

Table A.14: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.14 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.184218		0.291856		0.248325		0.275782
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.933885	Misc	0.926538	Misc	0.803138	Car	0.592815
	Van	0.058909	Car	0.071293	Car	0.175516	Misc	0.299039
	Car	0.007198	Van	0.002165	Pedestrian	0.018547	Van	0.071105
	Pedestrian	0.000007	Pedestrian	0.000004	Van	0.002799	Pedestrian	0.037041
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.248613		0.252774		0.248802		0.249811
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.518531	Misc	0.389321	Misc	0.534623	Car	0.427231
	Van	0.261828	Car	0.233046	Car	0.321133	Misc	0.280346
	Car	0.291021	Van	0.214521	Pedestrian	0.119821	Van	0.202021
	Pedestrian	0.00072	Pedestrian	0.163011	Van	0.024423	Pedestrian	0.090402
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.249685		0.253278		0.250946		0.246092
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.414701	Misc	0.428508	Misc	0.417018	Car	0.285971
	Van	0.341228	Car	0.357611	Car	0.351723	Misc	0.284601
	Car	0.136441	Van	0.125737	Pedestrian	0.175531	Van	0.241418
	Pedestrian	0.10773	Pedestrian	0.088143	Van	0.055608	Pedestrian	0.188004

Table A.15: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.15 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.232468		0.221238		0.291802		0.254492
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.699319	Misc	0.707465	Car	0.633655	Misc	0.860704
	Car	0.162418	Pedestrian	0.179127	Misc	0.159753	Cyclist	0.087301
	Pedestrian	0.104433	Car	0.089707	Cyclist	0.145853	Pedestrian	0.045369
	Cyclist	0.03883	Cyclist	0.0237	Pedestrian	0.060739	Car	0.006626
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.252433		0.253057		0.247816		0.246693
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.382709	Misc	0.424121	Misc	0.328912	Misc	0.370737
	Car	0.275033	Pedestrian	0.363257	Car	0.282861	Cyclist	0.291608
	Pedestrian	0.257517	Car	0.141309	Cyclist	0.195908	Pedestrian	0.183337
	Cyclist	0.08464	Cyclist	0.071412	Pedestrian	0.192319	Car	0.154318
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.251061		0.251747		0.24975		0.247442
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.364452	Misc	0.436317	Car	0.430238	Misc	0.272431
	Car	0.268521	Pedestrian	0.271236	Misc	0.295607	Cyclist	0.269209
	Pedestrian	0.254812	Car	0.159628	Cyclist	0.219523	Pedestrian	0.229353
	Cyclist	0.112214	Cyclist	0.132828	Pedestrian	0.054632	Car	0.229007

Table A.16: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.16 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.276766		0.236292		0.239568		0.247374
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.514796	Pedestrian	0.551602	Pedestrian	0.444259	Pedestrian	0.616123
	Pedestrian	0.300242	Misc	0.39412	Misc	0.420115	Misc	0.34762
	Cyclist	0.097531	Car	0.05019	Cyclist	0.109461	Car	0.019882
	Car	0.087431	Cyclist	0.004088	Car	0.026164	Cyclist	0.016375
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.249205		0.251036		0.246314		0.253445
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.385127	Pedestrian	0.294203	Pedestrian	0.621049	Pedestrian	0.358402
	Pedestrian	0.226233	Misc	0.282233	Misc	0.17132	Misc	0.220857
	Cyclist	0.204112	Car	0.233526	Cyclist	0.123927	Car	0.218841
	Car	0.184528	Cyclist	0.190038	Car	0.083704	Cyclist	0.2019
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.247759		0.252674		0.247663		0.251903
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.318673	Pedestrian	0.435337	Pedestrian	0.508404	Pedestrian	0.338039
	Pedestrian	0.288604	Misc	0.258324	Misc	0.298027	Misc	0.315841
	Cyclist	0.233611	Car	0.194833	Cyclist	0.16354	Car	0.181207
	Car	0.169112	Cyclist	0.111406	Car	0.029929	Cyclist	0.164913

Table A.17: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.17 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.210186		0.218035		0.221632		0.350147
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.671198	Misc	0.530003	Pedestrian	0.511435	Misc	0.549273
	Pedestrian	0.325681	Pedestrian	0.463613	Misc	0.419924	Pedestrian	0.417327
	Cyclist	0.003121	Cyclist	0.006384	Cyclist	0.068641	Cyclist	0.0344
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.249182		0.251717		0.249591		0.249509
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.384733	Misc	0.547627	Pedestrian	0.637481	Misc	0.519451
	Pedestrian	0.373547	Pedestrian	0.427166	Misc	0.209703	Pedestrian	0.251622
	Cyclist	0.24172	Cyclist	0.025306	Cyclist	0.152915	Cyclist	0.228927
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.247139		0.254905		0.250082		0.247874
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.409477	Misc	0.559828	Pedestrian	0.459161	Misc	0.428831
	Pedestrian	0.389521	Pedestrian	0.279931	Misc	0.382327	Pedestrian	0.345427
	Cyclist	0.201002	Cyclist	0.160241	Cyclist	0.158512	Cyclist	0.225741

Table A.18: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.18 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.324637		0.152514		0.250674		0.272175
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.902131	Misc	0.965762	Misc	0.959268	Misc	0.86135
	Van	0.067129	Car	0.03423	Car	0.039583	Car	0.138646
	Car	0.03074	Van	0.000008	Van	0.001148	Van	0.000004
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.249412		0.250043		0.246488		0.254057
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.479339	Misc	0.414328	Misc	0.623739	Misc	0.416671
	Van	0.427657	Car	0.370261	Car	0.212021	Car	0.377911
	Car	0.093004	Van	0.215411	Van	0.16424	Van	0.205418
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.249814		0.249871		0.245456		0.254859
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.480871	Misc	0.403287	Misc	0.497665	Misc	0.709559
	Van	0.325113	Car	0.330502	Car	0.363221	Car	0.211617
	Car	0.193916	Van	0.266212	Van	0.139114	Van	0.078824

Table A.19: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.19 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.209211		0.269387		0.214022		0.307381
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.831119	Misc	0.706127	Misc	0.69004	Misc	0.691732
	Van	0.070488	Car	0.142716	Car	0.101376	Van	0.166957
	Cyclist	0.043232	Cyclist	0.088806	Cyclist	0.080769	Pedestrian	0.076524
	Car	0.034501	Van	0.061686	Pedestrian	0.073899	Cyclist	0.047394
	Pedestrian	0.02066	Pedestrian	0.000665	Van	0.053916	Car	0.017394
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.250933		0.248209		0.250076		0.250782
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.256731	Misc	0.278531	Misc	0.290131	Misc	0.295621
	Van	0.235027	Car	0.247726	Car	0.234922	Van	0.277454
	Cyclist	0.192416	Cyclist	0.243811	Cyclist	0.179617	Pedestrian	0.188601
	Car	0.161021	Van	0.229721	Pedestrian	0.154721	Cyclist	0.163512
	Pedestrian	0.154805	Pedestrian	0.00011	Van	0.140608	Car	0.074813
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.251993		0.248663		0.249117		0.250227
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Misc	0.318161	Misc	0.252431	Misc	0.322739	Misc	0.342131
	Van	0.287124	Car	0.229044	Car	0.267421	Van	0.335022
	Cyclist	0.252407	Cyclist	0.187613	Cyclist	0.202217	Pedestrian	0.192037
	Car	0.106302	Van	0.16991	Pedestrian	0.201702	Cyclist	0.0830
	Pedestrian	0.036005	Pedestrian	0.160902	Van	0.005921	Car	0.047909

Table A.20: Results of traditional LDA model [12], derived LDA model with continue feature only and dual-variable LDA model on sequence.20 of training data set of KITTI benchmark.

	Topic 1		Topic 2		Topic 3		Topic 4	
traditional LDA model [12]	θ	probability	θ	probability	θ	probability	θ	probability
		0.250442		0.29207		0.235696		0.221792
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.7526	Car	0.640687	Car	0.744626	Car	0.862398
	Misc	0.247393	Misc	0.340548	Misc	0.212091	Misc	0.137595
	Van	0.000007	Van	0.018764	Van	0.043282	Van	0.000008
derived LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.246903		0.260133		0.250695		0.242268
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.518731	Car	0.598771	Car	0.607911	Car	0.379731
	Misc	0.454947	Misc	0.353018	Misc	0.239147	Misc	0.349327
	Van	0.026321	Van	0.048211	Van	0.152942	Van	0.270842
dual-variable LDA model	θ	probability	θ	probability	θ	probability	θ	probability
		0.245639		0.260302		0.250274		0.243785
	word assign	probability	word assign	probability	word assign	probability	word assign	probability
	Car	0.661871	Car	0.401181	Car	0.484872	Car	0.394066
	Misc	0.293717	Misc	0.31081	Misc	0.402916	Misc	0.384736
	Van	0.044412	Van	0.288009	Van	0.112212	Van	0.221208

Bibliography

[1] D. Agarwal and B.-C. Chen, "flda: matrix factorization through latent dirichlet allocation," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 91–100. 58

[2] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1442–1456, 2011. 16

[3] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P. S. Yu, "Global distance-based segmentation of trajectories," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 34–43. 17, 18

[4] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti, "Interactive visual clustering of large collections of trajectories," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, 2009, pp. 3–10. 21

[5] G. Antonini and J.-P. Thiran, "Counting pedestrians in video sequences using trajectory clustering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 8, pp. 1008–1020, 2006. 27

[6] S. Atev, O. Masoud, and N. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," in *IROS*, 2006, pp. 4851–4856. 1, 28

[7] S. Atev, G. Miller, and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 647–657, 2010. 19

[8] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE transactions on Image Processing*, vol. 16, no. 7, pp. 1912–1919, 2007. 1, 13, 14, 17, 29

[9] ——, "Real-time motion trajectory-based indexing and retrieval of video sequences," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 58–65, 2007. 13, 14, 17

[10] M. A. Bautista, A. Hernández-Vela, S. Escalera, L. Igual, O. Pujol, J. Moya, V. Violant, and M. T. Anguera, "A gesture recognition system for detecting behavioral patterns of adhd," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 136–147, 2016. 20, 31

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417. xii, 12, 40, 41

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003. 57, 58, 62, 82, 83, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109

[13] N. Brandle, D. Bauer, and S. Seer, "Track-based finding of stopping pedestrians-a practical approach for analyzing a public infrastructure," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 115–120. 18

[14] A. E. Brouwer and W. H. Haemers, *Spectra of graphs*. Springer Science & Business Media, 2011. 28

[15] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *European conference on computer vision*. Springer, 2010, pp. 282–295. 27

[16] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3491–3496. 27

[17] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, "Efficient anomaly monitoring over moving object trajectory streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 159–168. 21

[18] G. Cai, K. Lee, and I. Lee, "A framework for mining semantic-level tourist movement behaviours from geo-tagged photos," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2016, pp. 519–524. 10

[19] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007. 17

[20] R. Chaker, Z. Al Aghbari, and I. N. Junejo, "Social network model for crowd anomaly detection and localization," *Pattern Recognition*, vol. 61, pp. 266–281, 2017. 1

[21] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009. 28

[22] J. Chen, R. Wang, L. Liu, and J. Song, "Clustering of trajectories based on hausdorff distance," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*. IEEE, 2011, pp. 1940–1944. 19

[23] W. Chen and J. J. Corso, "Action detection by implicit intentional motion clustering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3298–3306. 28

[24] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie, "Searching trajectories by locations: an efficiency study," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 255–266. 1

[25] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3. IEEE, 2002, pp. 545–548. 10

[26] K. Cho and X. Chen, "Classifying and visualizing motion capture sequences using deep neural networks," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2. IEEE, 2014, pp. 122–130. 15, 33

[27] M. Cho and K. MuLee, "Authority-shift clustering: Hierarchical clustering by authority seeking on graphs," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3193–3200. 29

[28] D.-W. Choi, J. Pei, and T. Heinis, "Efficient mining of regional movement patterns in semantic trajectories," *Proceedings of the VLDB Endowment*, vol. 10, no. 13, pp. 2073–2084, 2017. 10

[29] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003. 10

[30] K. Deng, K. Xie, K. Zheng, and X. Zhou, "Trajectory indexing and retrieval," in *Computing with spatial trajectories*. Springer, 2011, pp. 35–60. 23

[31] M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, and A. Del Bimbo, "Motion segment decomposition of rgb-d sequences for human behavior understanding," *Pattern Recognition*, vol. 61, pp. 222–233, 2017. 32

[32] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3-d human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015. 30

[33] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2790–2797. 17

[34] H. Ergezer and K. Leblebicioğlu, "Anomaly detection and activity perception using covariance descriptor for trajectories," in *European Conference on Computer Vision*. Springer, 2016, pp. 728–742. 28

[35] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231. 15, 22

[36] D. R. Faria and J. Dias, "3d hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1284–1289. 17

[37] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation (for robots)," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, pp. 691–700, 1989. 39

[38] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, and C. T. Silva, "Vector field k-means: Clustering trajectories by fitting multiple vector fields," in *Computer Graphics Forum*, vol. 32. Wiley Online Library, 2013, pp. 201–210. 1, 13, 15, 16, 17, 24

[39] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4083–4090. 18, 33

[40] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis, "Algorithms for nearest neighbor search on moving object trajectories," *Geoinformatica*, vol. 11, no. 2, pp. 159–193, 2007. 31

[41] A. Gaidon, Z. Harchaoui, and C. Schmid, "Activity representation with motion hierarchies," *International journal of computer vision*, vol. 107, no. 3, pp. 219–238, 2014. 26

[42] L. Galluccio, O. Michel, P. Comon, and A. O. Hero, "Graph based k-means clustering," *Signal Processing*, vol. 92, no. 9, pp. 1970–1984, 2012. 24

[43] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "Devnet: A deep event network for multimedia event detection and evidence recounting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2568–2577. 33

[44] Y.-J. Gao, C. Li, G.-C. Chen, L. Chen, X.-T. Jiang, and C. Chen, "Efficient k-nearest-neighbor search algorithms for historical moving object trajectories," *Journal of Computer Science and Technology*, vol. 22, no. 2, pp. 232–244, 2007. 15, 30

[45] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 3354–3361. 43

[46] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004. 63, 65

[47] D. Guo, S. Liu, and H. Jin, "A graph-based approach to vehicle trajectory analysis," *Journal of Location Based Services*, vol. 4, no. 3-4, pp. 183–199, 2010. 28

[48] S. Gurung, D. Lin, W. Jiang, A. Hurson, and R. Zhang, "Traffic information publication with privacy preservation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, p. 44, 2014. 1, 15, 29, 34

[49] M. Hasan and A. K. Roy-Chowdhury, "A continuous learning framework for activity recognition using deep hybrid feature models," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1909–1922, 2015. 33

[50] W. He, T. Yamashita, H. Lu, and S. Lao, "Surf tracking," in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 1586–1592. 12

[51] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001. 57, 62, 63

[52] ——, "Probabilistic latent semantic indexing," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM, 2017, pp. 211–218. 57, 62

[53] D. Hong, Q. Gu, and K. Whitehouse, "High-dimensional time series clustering via cross-predictability," in *Artificial Intelligence and Statistics*, 2017, pp. 642–651. 28

[54] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981. 10

[55] H. Hu, J. Feng, and J. Zhou, "Exploiting unsupervised and supervised constraints for subspace clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1542–1557, 2015. 16

[56] W. Hu, X. Li, G. Tian, S. Maybank, and Z. Zhang, "An incremental dpmm-based method for trajectory clustering, modeling, and retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 5, pp. 1051–1065, 2013. 14, 15, 17, 32, 35, 86

[57] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006. 24

[58] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Transactions on image processing*, vol. 16, no. 4, pp. 1168–1181, 2007. 27

[59] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of convoys in trajectory databases," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1068–1080, 2008. 13, 23

[60] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *European Conference on Computer Vision*. Springer, 2012, pp. 425–438. 10, 15, 17, 18

[61] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *arXiv preprint arXiv:1502.07209*, 2015. 33

[62] I. N. Junejo and H. Foroosh, "Euclidean path modeling for video surveillance," *Image and Vision computing*, vol. 26, no. 4, pp. 512–528, 2008. 29

[63] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3271–3279. 29

[64] H. S. Khaing and T. Thein, "An efficient clustering algorithm for moving object trajectories," in *3rd International Conference on Computational techniques and Artificial Intelligence (ICCTAI 2014) February*, 2014, pp. 11–12. 23, 86

[65] Z. Kim, "Real time object tracking based on dynamic feature grouping with background subtraction," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. 24

[66] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association, 2008, pp. 275–1. 18

[67] S.-Y. Kong and L.-S. Lee, "Semantic analysis and organization of spoken documents based on parameters derived from latent topics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1875–1889, 2011. 57

[68] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 61–68. 57, 58

[69] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *The International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008. 34

[70] S. Kumar, Y. Dai, and H. Li, "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion," *Pattern Recognition*, vol. 71, pp. 428–443, 2017. 1

[71] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. 18

[72] F. Lauer and C. Schnörr, "Spectral clustering of linear subspaces for motion segmentation," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 678–685. 27

[73] R. Laxhammar and G. Falkman, "Online learning and sequential anomaly detection in trajectories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1158–1173, 2014. 34

[74] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 2169–2178. 57

[75] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, "Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1081–1094, 2008. 21, 23, 86

[76] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD interna-*

tional conference on Management of data. ACM, 2007, pp. 593–604. 14, 17, 18, 21, 23, 86

[77] L.-S. Lee, S.-C. Chen, Y. Ho, J.-F. Chen, M.-H. Li, and T.-H. Li, "An initial prototype system for chinese spoken document understanding and organization for indexing/browsing and retrieval applications," in *2004 International Symposium on Chinese Spoken Language Processing*. IEEE, 2004, pp. 329–332. 57

[78] W. Li, X. Zhou, and T. Chai, ""bag of visual words" and latent semantic analysis-based burning state recognition for rotary kiln sintering process," in *2011 Chinese Control and Decision Conference (CCDC)*. IEEE, 2011, pp. 377–382. 57

[79] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 591–594. 1, 19, 25, 27

[80] X. Li, V. Ceikute, C. S. Jensen, and K.-L. Tan, "Effective online group discovery in trajectory databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2752–2766, 2013. 34

[81] X. Li, J. Han, J.-G. Lee, and H. Gonzalez, "Traffic density-based discovery of hot routes in road networks," in *International Symposium on Spatial and Temporal Databases*. Springer, 2007, pp. 441–459. 1, 28

[82] Z. Li, J.-G. Lee, X. Li, and J. Han, "Incremental clustering for trajectories," in *International Conference on Database Systems for Advanced Applications*. Springer, 2010, pp. 32–46. 21

[83] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4844–4857, 2012. 28

[84] W.-G. Liou, C.-Y. Hsieh, and W.-Y. Lin, "Trajectory-based sign language recognition using discriminant analysis in higher-dimensional feature

space," in *2011 IEEE International Conference on Multimedia and Expo*. IEEE, 2011, pp. 1–4. 17

[85] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 99–112, 2014. 19, 28

[86] X. Liu, L. Lin, S.-C. Zhu, and H. Jin, "Trajectory parsing by cluster sampling in spatio-temporal graph," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 739–746. 29

[87] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157. 11, 43

[88] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 514–521. 18

[89] ——, "Representing pairwise spatial and temporal relations for action recognition," in *European Conference on Computer Vision*. Springer, 2010, pp. 508–521. 18

[90] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525. 39

[91] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012. 39

[92] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Detection and classification of highway lanes using vehicle motion trajectories," *IEEE*

Transactions on intelligent transportation systems, vol. 7, no. 2, pp. 188–200, 2006. 24

[93] Y. Mo, D. Wu, and Y. Du, "Application of trajectory clustering and regionalization to ocean eddies in the south china sea," in *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference on*. IEEE, 2015, pp. 45–48. 1

[94] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 312–319. 19, 21

[95] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 8, pp. 1114–1127, 2008. 18

[96] ——, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 11, pp. 2287–2301, 2011. 18, 24

[97] A. Naftel and S. Khalid, "Motion trajectory learning in the dft-coefficient feature space," in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*. IEEE, 2006, pp. 47–47. 14, 15, 17, 33

[98] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 267–289, 2006. 1, 13, 19

[99] B. Ni, P. Moulin, X. Yang, and S. Yan, "Motion part regularization: Improving action recognition via trajectory selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3698–3706. 27

[100] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International journal of computer vision*, vol. 79, no. 3, pp. 299–318, 2008. 58

[101] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014. 17

[102] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 863–868. 21

[103] G. Palou and P. Salembier, "Hierarchical video representation with trajectory binary partition tree," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2099–2106. 28

[104] N. Pelekis, I. Kopanakis, E. E. Kotsifakos, E. Frentzos, and Y. Theodoridis, "Clustering trajectories of moving objects in an uncertain world." in *ICDM*, vol. 9, 2009, pp. 417–427. 24, 29

[105] ——, "Clustering uncertain trajectories," *Knowledge and Information Systems*, vol. 28, no. 1, pp. 117–147, 2011. 24

[106] N. Pelekis, P. Tampakis, M. Vodas, C. Doulkeridis, and Y. Theodoridis, "On temporal-constrained sub-trajectory cluster analysis," *Data Mining and Knowledge Discovery*, pp. 1–37, 2017. 29

[107] C. Piciarelli and G. L. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1835–1842, 2006. 34

[108] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transactions on Circuits and Systems for video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008. 13, 31

[109] N. Piotto, N. Conci, and F. G. De Natale, "Syntactic matching of trajectories for ambient intelligence applications," *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1266–1275, 2009. 17

[110] S. Poularakis and I. Katsavounidis, "Low-complexity hand gesture recognition system for continuous streams of digits and letters," *IEEE transactions on cybernetics*, vol. 46, no. 9, pp. 2094–2108, 2016. 30

[111] H. Rahmani, A. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 667–681, 2018. 32

[112] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*. IEEE, 2012, pp. 1242–1249. 26, 27

[113] F. Remondino, M. G. Spera, E. Nocerino, F. Menna, and F. Nex, "State of the art in high density image matching," *The Photogrammetric Record*, vol. 29, no. 146, pp. 144–166, 2014. 10

[114] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 487–494. 57

[115] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571. 12

[116] J. Schenk and G. Rigoll, "Novel hybrid nn/hmm modelling techniques for on-line handwriting recognition," in *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006. 17

[117] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer, "Visual cluster analysis of trajectory data with interactive kohonen maps," *Information Visualization*, vol. 8, no. 1, pp. 14–29, 2009. 33

[118] Z. Shao and Y. Li, "On integral invariants for effective 3-d motion trajectory matching and recognition," *IEEE transactions on cybernetics*, vol. 46, no. 2, pp. 511–523, 2016. 20

[119] Y. Shi, W. Zeng, T. Huang, and Y. Wang, "Learning deep trajectory descriptor for action recognition in videos using deep neural networks," in *2015 IEEE International Conference on Multimedia and Expo*. IEEE, 2015, pp. 1–6. 33

[120] R. R. Sillito and R. B. Fisher, "Semi-supervised learning for anomalous trajectory detection." in *BMVC*, vol. 1, 2008, pp. 035–1. 15, 16, 86

[121] D. Simonnet, E. Anquetil, and M. Bouillon, "Multi-criteria handwriting quality analysis with online fuzzy models," *Pattern Recognition*, vol. 69, pp. 310–324, 2017. 24

[122] D. Singh and C. K. Mohan, "Graph formulation of video activities for abnormal activity recognition," *Pattern Recognition*, vol. 65, pp. 265–272, 2017. 31

[123] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," 2005. 58

[124] F. Solera, S. Calderara, and R. Cucchiara, "Socially constrained structural learning for groups detection in crowd," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 995–1008, 2016. 31

[125] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2004–2011. 18

[126] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino, "Learning motion patterns and anomaly detection by human trajectory analysis," in *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2007, pp. 498–503. 24

[127] S. S. Tabatabaei, M. Coates, and M. Rabbat, "Ganc: Greedy agglomerative normalized cut," *arXiv preprint arXiv:1105.0974*, 2011. 26, 27

[128] F. Turchini, L. Seidenari, and A. Del Bimbo, "Understanding sport activities from correspondences of clustered trajectories," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 43–50. 27

[129] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and trends® in computer graphics and vision*, vol. 3, no. 3, pp. 177–280, 2008. 18

[130] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition." in *BMVC*, 2008, pp. 1–10. 10, 18, 19

[131] O. Uncu, W. A. Gruver, D. B. Kotak, D. Sabaz, Z. Alibhai, and C. Ng, "Gridbscan: Grid density-based spatial clustering of applications with noise," in *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, vol. 4. IEEE, 2006, pp. 2976–2981. 86

[132] E. Vig, M. Dorr, and D. Cox, "Space-variant descriptor sampling for action recognition based on saliency and eye movements," in *European conference on computer vision*. Springer, 2012, pp. 84–97. 18

[133] N. Vretos, N. Nikolaidis, and I. Pitas, "A perceptual hashing algorithm using latent dirichlet allocation," in *2009 IEEE International Conference on Multimedia and Expo*. IEEE, 2009, pp. 362–365. 57

[134] F. Wang, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and visual relatedness," in *Proceedings of the 16th ACM international conference on Multimedia*. ACM, 2008, pp. 239–248. 18, 19

[135] H. Wang and C. O'Sullivan, "Globally continuous and non-markovian crowd activity analysis from videos," in *European Conference on Computer Vision*. Springer, 2016, pp. 527–544. 1, 32

[136] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176. 15, 18

[137] ——, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013. 18

[138] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238, 2016. 10, 18

[139] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558. 13, 23

[140] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009-British Machine Vision Conference*. BMVA Press, 2009, pp. 124–1. 17, 86

[141] L. Wang and M. Dong, "Detection of abnormal human behavior using a matrix approximation-based approach," in *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*. IEEE, 2014, pp. 324–329. 1, 15, 34

[142] X. Wang and E. Grimson, "Spatial latent dirichlet allocation," in *Advances in neural information processing systems*, 2008, pp. 1577–1584. 57, 58

[143] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," in *European conference on computer vision*. Springer, 2006, pp. 110–123. 15, 25

[144] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *European conference on computer vision*. Springer, 2008, pp. 650–663. 18

[145] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A comparative study of sift and its variants," *Measurement science review*, vol. 13, no. 3, pp. 122–131, 2013. 11

[146] Z. Wu, Y. Fu, Y.-G. Jiang, and L. Sigal, "Harnessing object and scene semantics for large-scale video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3112–3121. 15, 33

[147] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue, "Exploring inter-feature and inter-class relationships with deep neural networks for video classification," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 167–176. 33

[148] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognition*, vol. 41, no. 3, pp. 1012–1029, 2008. 15, 27

[149] H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4328–4336. 24

[150] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty, "A hybrid model and computing platform for spatio-semantic trajectories," in *Extended Semantic Web Conference*. Springer, 2010, pp. 60–75. 17

[151] T. Yao, Z. Wang, Z. Xie, J. Gao, and D. D. Feng, "Learning universal multiview dictionary for human action recognition," *Pattern Recognition*, vol. 64, pp. 236–244, 2017. 1

[152] Y. Yi and M. Lin, "Human action recognition with graph-based multiple-instance learning," *Pattern Recognition*, vol. 53, pp. 148–162, 2016. 28

[153] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 34–43. 9

[154] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Mining user similarity from semantic trajectories," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*. ACM, 2010, pp. 19–26. 9

[155] C. Yuan and A. Chakraborty, "Deep convolutional factor analyser for multivariate time series modeling," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 1323–1328. 32

[156] Y. Yuan, Y. Feng, and X. Lu, "Statistical hypothesis detector for abnormal event detection in crowded scenes," *IEEE transactions on cybernetics*, vol. 47, no. 11, pp. 3597–3608, 2017. 1, 15, 31, 34

[157] E. Zavarehei and S. Vaseghi, "Speech enhancement in temporal dft trajectories using kalman filters," in *Ninth European Conference on Speech Communication and Technology*, 2005. 14

[158] T. Zhang, H. Lu, and S. Z. Li, "Learning semantic scene models by object classification and trajectory clustering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1940–1947. 13, 15, 16, 28, 86

[159] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 1135–1138. 19

[160] Z. Zhang, K. Huang, T. Tan, and L. Wang, "Trajectory series analysis based event rule induction for visual surveillance," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8. 18

[161] Z. Zhang, K. Huang, T. Tan, P. Yang, and J. Li, "Red-sfa: Relation discovery based slow feature analysis for trajectory clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 752–760. 28

[162] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. 51

[163] W. Zhao, Z. Zhang, and K. Huang, "Gestalt laws based tracklets analysis for human crowd understanding," *Pattern Recognition*, vol. 75, pp. 112–127, 2018. 1

[164] B. Zheng, N. J. Yuan, K. Zheng, X. Xie, S. Sadiq, and X. Zhou, "Approximate keyword search in semantic trajectory database," in *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 2015, pp. 975–986. 10

[165] K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang, "On discovery of gathering patterns from trajectories," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 242–253. 25

[166] Y. Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, p. 29, 2015. 23

[167] Y. Zheng, X. Xie, and W.-Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010. 23

[168] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800. 24

[169] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 112, 2013. 18

[170] Y. Zhou, S. Yan, and T. S. Huang, "Detecting anomaly in videos from trajectory similarity analysis," in *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 2007, pp. 1087–1090. 24