

東京大学大学院新領域創成科学研究科  
社会文化環境学専攻

2021 年度  
修 士 論 文

Key Time Extraction in City Mobility  
based on Video Summarization  
映像要約技術に基づく都市移動における重要時点の抽出

2021 年 7 月 9 日提出

指導教員 宋 軒 准教授

副指導教員 山田 育穂 教授

尹 航

Yin, Hang



# *Abstract*

Human mobility mode in range of a city provides sufficient information for urban planning and transportation management, which can help improving traffic efficiency and managing public transportation, thus benefiting life of residents. Nowadays, management methods like timeshare fare, speed limit of road, number limit of license plate, etc. have been widely used in transportation management. However, these methods share a similarity that the fare or the management plan are decided beforehand, which means it could not respond to changes in a timely manner. To address this problem, in this paper, the key frame extraction algorithm from video summarization field is adapted into framework of key time extraction in City mobility to grasp city mobility mode in different scales, based on which, transportation management can make dynamic changes to better adapt to the urban rhythm.

**Key words:** mobility mode mining, key time extraction, submodular optimization

# *Acknowledgements*

First and foremost, I would like to express my sincere gratitude to my adviser Prof. Ryousuke Shibasaki and Prof. Xuan Song, who supported us to do the field research in key time extraction of human mobility and listened to my report every week with full patience. He guided me how to build the structure of the research and gave me lots of useful advices which inspire me to keep going on the research. I am also grateful to my co-advisor, Prof. Ikuho Yamada, who also gave me many practical suggestions as research directions.

Thanks to Researcher Zipei Fan, Haoran Zhang, Renhe Jiang, Kanasugi and Apichon, who taught me lots of useful programming skills and ideas to do research.

In addition, I thank all my friends in Shibasaki Lab: Thanks to Zekun Cai, Jinyu Chen, Liqiang Xu, Haoyu Wang, Lifeng Lin, Yinghao Liu, Xinlei Lian, Pochia Chen, Wenjing Li, with whom I enjoy the happy study life in the University of Tokyo. Thanks to senior students Zhaonan Wang, Zhiwen Zhang and Zhiling Guo who helps me with my daily life in Japan.

Thanks to my parents who give me birth and everyone supporting me.

# Contents

<b>Chapter 1. Introduction .....</b>	<b>1</b>
<b>1.1 Background .....</b>	<b>1</b>
<b>1.2 Related Works.....</b>	<b>4</b>
<b>1.3 Dataset.....</b>	<b>6</b>
<b>Chapter 2. Data Filtering .....</b>	<b>7</b>
<b>2.1 Trajectory clustering .....</b>	<b>8</b>
<b>2.1.1 Several measurements of trajectory distance .....</b>	<b>8</b>
<b>2.1.2 Applicability Analysis.....</b>	<b>10</b>
<b>2.1.3 <i>SSPD</i> distance.....</b>	<b>10</b>
<b>2.2 Reconstruction of low-rank matrix.....</b>	<b>13</b>
<b>2.2.1 Background .....</b>	<b>13</b>
<b>2.2.2 Problem and method .....</b>	<b>15</b>
<b>2.2.3 Missing value recovery .....</b>	<b>16</b>
<b>2.3 Map-matching .....</b>	<b>16</b>
<b>Chapter 3. Video Summarization.....</b>	<b>20</b>
<b>3.1 Problem and goal .....</b>	<b>20</b>
<b>3.2 Approaches .....</b>	<b>20</b>

3.2.1	Supervised video summarization .....	20
3.2.2	Unsupervised video summarization .....	21
3.2.3	Weakly-supervised video summarization.....	22
Chapter 4.	Transformation and Extraction .....	24
4.1	Density dissimilarity matrix.....	24
4.2	Methods.....	25
4.2.1	<i>K-Medoids</i> clustering .....	25
4.2.2	Two Approaches to extract key time .....	25
4.3	Simulation and Comparison .....	27
4.3.1	Key Time 1.....	27
4.3.2	Key Time 2.....	29
4.3.3	Comparison .....	30
4.4	Weakness .....	32
Chapter 5.	Submodular Optimization .....	34
5.1	Definition .....	34
5.2	Examples.....	35
5.3	Consequence analysis .....	37
Chapter 6.	Conclusions and Future Work .....	39
Bibliography	.....	40

# List of Figures

Figure 1. Illustration of relationship between key time extraction and video summarization .....	4
Figure 2. Illustration of <i>Frechet</i> distance.....	9
Figure 3. Trajectory clustering based on OWD distance.....	13
Figure 4. Trajectory clustering based on OWD distance.....	13
Figure 5. Video grading system.....	14
Figure 6. Map-matching appearance.....	19
Figure 7. Flow of supervised video summarization.....	21
Figure 8. Flow of unsupervised video summarization.....	22
Figure 9. Polyline of dissimilarity based on Key Time 1.....	27
Figure 10. Human distribution of extracted time.....	28
Figure 11. Dissimilarity matrix based on Key Time 2.....	29
Figure 12. Situation of ignorance of human mobility.....	32
Figure 13. Drinking water distribution network simulation.....	36

# List of Tables

Table 1. Submodularity and OD matrix based standard deviation.....	37
Table 2. Key Time 2 based standard deviation.....	38



# Chapter 1. Introduction

## 1.1 Background

Human mobility mode in range of a city provides sufficient information for urban planning and transportation management, which can help improving traffic efficiency and managing public transportation, thus benefiting life of residents [1][2]. Nowadays, management methods like timeshare fare, speed limit of road, number limit of license plate, etc. have been widely used in transportation management. However, these methods share a similarity that the fare or the management plan are decided beforehand [3], which means it could not respond to changes in a timely manner. Maybe such methods can deal with transportation problems and be treated as prior experience to help posteriori decision, but may face with some situation with conflict, for example:

There are 2 high ways between location A and location B, based on the reason that average speed of high way m is faster than speed of high way n, management department set the highway toll of m higher than n so that to keep the user balance between 2 high ways and ensure the speed and safety.

However, during a special festival, people living in city A and B want to drop around each other, which doesn't need to be hurry on the road like working days. Under this situation, the usual management method may not deal with the change well. Due to the lower toll of  $n$ , users all concentrate on high way  $n$ , which becomes dangerous and inefficient. It is no need to keep high toll of high way  $m$  any more in order to keep the user balance.

If we can grasp the dynamic change of human mobility macroscopically, we can find out when the change of management should be adjusted. Furthermore, with the tendency of human mobility change and the detailed degree, specific measures like change of toll, road regulation time zone designation, etc. can be carried out. [4]

In another field, video summarization technologies aim to create a concise and complete synopsis by selecting the most informative parts of the video content. Nowadays, when we surf the Internet and watch videos on modern video sites such as YouTube, we can find the extracted key frames are automatically played to tell users their general contents. In this way, users can get a glimpse of certain video summary and quickly decide if it suits their interests simply by looking at the extracted summary video. [5]

The application domain of automatic video summarization is wide and includes (but is not limited to) the use of such technologies by media organizations, to allow effective indexing, browsing, retrieval and promotion. In addition, video summarization that is tailored to the requirements of particular content presentation scenarios can be used for e.g., generating trailers or teasers of movies and episodes of a TV series; presenting the highlights of an event; and creating a video synopsis with the main activities. [6]

Frames are arranged one by one in order of time to compose a video. From a dynamic point of view, traffic data is also organized in this way, which is keeps changing where people are over time [7]. If looking at a dynamic map, it behaves like a video. If making comparison between video summarization and time point extraction, we can find their similarity and dissimilarity:

Similarity: Both of them contain ‘time’ dimension in feature vectors, which means change occur over time.

Dissimilarity: Mobility data vectors contain features such as longitude, latitude, ID, and density, while frame vectors contain two-dimensional coordinates and RGB matrices.

In the case of key time extraction, if mobility data set is treated as one video, the map of the location and movement of people at each time point is treated as one frame of the video, and the transformation matrix is used to convert the features except time in the feature vector of mobility data, the algorithm of video algorithm will be applicable.

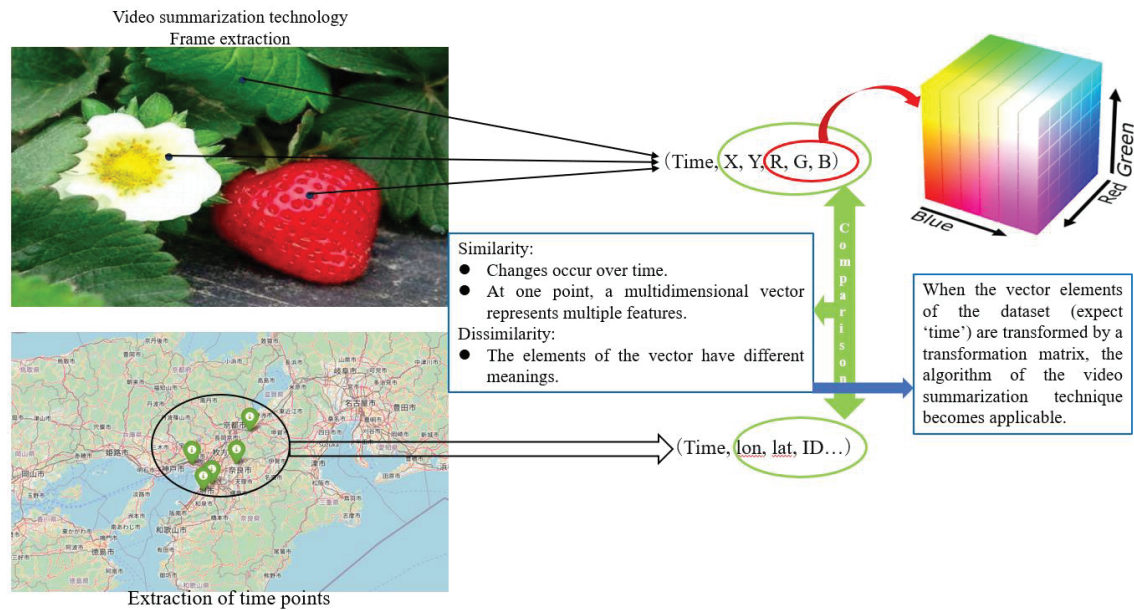


Figure.1

Illustration of relationship between key time extraction and video summarization

## 1.2 Related Works

In video summarization field, several approaches have been developed over the last couple of decades and the current state of the art is represented by methods that rely on modern deep neural network architectures.

However, even after the process of matrix transformation, not all the approaches of video summarization are applicable in key time extraction, which will be illustrated.

Before carrying out matrix transformation, due to the mobility data is different from a clear video that it often has missing values, and could not describe location of all residents at all time points. Therefore, data flittering should be done, which includes missing value complement, data interpolation, etc. Furthermore, the consequence of interpolation sometimes appears to have flash phenomenon [8] such as going straight across the building, high way, sea, etc., which needs to match to their location close to reality.

When carrying out matrix transformation, different distance metrics lead to different penalty function, and then influence the process of key time extraction. Several of trajectory distance computation methods will be compared in order to select the suitable method under the situation of computing the distance between the location of same group of residents at different times.

After finishing matrix transformation, based on the transformed mobility data in time sequence, submodular optimization will be adapted to the extracted process, which means selecting certain (artificial set before) most representative time points to generalize the mode and mine the abnormal points.

### **1.3 Dataset**

In this research, 2 datasets are utilized:

- Data of Osaka and its surrounding areas (kansai reigon) with time, ID, longitude and latitude to compare the difference between consequence of 2 time extraction methods.
- Data of Tokyo and its surrounding areas (kanto reigon) with time, ID, longitude and latitude to test the appearance of improved time extraction method.

## Chapter 2. Data Filtering

### 2.1 Trajectory clustering

A continuous trajectory is a function which gives the location of a moving object as a continuous function of time. In the case of human mobility trajectory, we only consider discrete trajectories defined here after.

**Definition 1.** A trajectory  $T$  is defined as:

$$T: ((p_1, t_1), \dots, (p_n, t_n)),$$

where  $p_k \in \mathbb{R}^2, t_k \in \mathbb{R} \forall k \in [1, \dots, n], \forall n \in \mathbb{N}$  and  $n$  is the length of the trajectory  $T$ . The exact location between time  $t_i$  and  $t_{i+1}$  are unknown.

#### 2.1.1 Several measurements of trajectory distance

In order to know how close or how similar two objects are far from another, classically we can define dissimilarity as:

**Definition 2.** Let  $\mathcal{T}$  be a set of trajectories containing several independent trajectories. The function  $d: \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{R}$  is called a dissimilarity on  $\mathcal{T}$  if for all  $T_1, T_2 \in \mathcal{T}$ :

- $d(T_1, T_2) \geq 0$
- $d(T_1, T_2) = d(T_2, T_1)$
- $d(T_1, T_1) = 0$

If all these conditions are satisfied and  $d(T_1, T_2) = 0 \Rightarrow T_1 = T_2$ ,  $d$  is considered to be a symmetric. If the triangle inequality is also satisfied,  $d$  is called a metric.

Based on similarity of trajectory shape, *Hausdorff*, *Frechet*, *OWD* and *SSPD* distance are computed to measure the distance. [9]

**Definition 3.** The *Hausdorff* distance between two discrete trajectory is defined as:

$$D_{Haus}(T_1, T_2) = \max \left\{ \begin{array}{l} \max_{\substack{i_1 \in [1 \dots n^1] \\ j_2 \in [1 \dots n^2-1]}} \{D_{ps}(p_{i_1}^1, s_{j_2}^2)\} \\ \max_{\substack{j_1 \in [1 \dots n^1-1] \\ i_2 \in [1 \dots n^2]}} \{D_{ps}(p_{i_2}^2, s_{j_1}^1)\} \end{array} \right.$$

The *Hausdorff* distance can be computed in a  $O(n^2)$  computational time.

**Definition 4.** The *Frechet* distance between two discrete trajectory is defined as:

$$D_{Frec}(T_1, T_2) = \min_W \left\{ \max_{k \in [1 \dots |W|]} \|\omega_k\|_2 \right\},$$

in which,  $W$  means a warping path. The *Frechet* distance can be computed in  $O(n^2)$  time. The *Frechet* distance measures similarity between curves.

It is often known as the ‘walking-dog distance’ (as Figure 2 shows).

Imagine a dog and its owner walking on two separate paths without backtracking from one endpoint to one other. The *Frechet* distance is the minimum length of leash required to connect a dog and its owner.



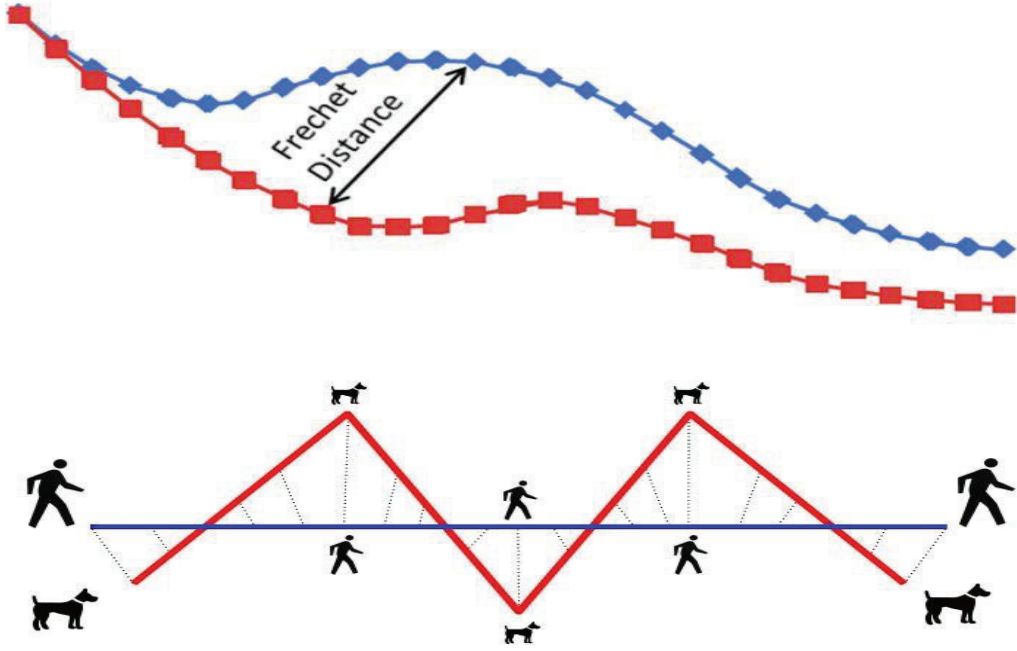


Figure 2. Illustration of Frechet distance

**Definition 5.** The OWD (One-Way) distance from a discrete trajectory  $T_1$  to another discrete trajectory  $T_2$  is defined as the integral of the distance from points of  $T_1^{pl}$  to trajectory  $T_2^{pl}$  divided by the length of  $T_1^{pl}$ .

$$D_{OWD}(T_1, T_2) = \frac{1}{n_{pl}^1} \int_{p^1 \in T_{pl}^1} d_{point}(p^1 T_2) dp^1$$

$d_{point}(p, T)$  is the distance from the point  $p$  to the trajectory  $T$ , so it can be computed as:

$$d_{point}(p, T) = \min_{q \in T_{pl}} \|pq\|_2$$

The OWD distance can be computed in a  $O(n^2 \log(n))$  computational time.

### 2.1.2 Applicability Analysis

- *Hausdroff* distance can deal with space more complicated than point set, but complicated and resource intensive to calculated when applied to most existing trajectory sets.
- Similarity measurement of *Frechet* distance between trajectory segments is excellent, but is not suitable for comparison of similarity between overall trajectories (easy to be interfered by a certain extreme maximum distance).
- The *OWD* distance considers the trajectory as a whole, including the shape and physical distance of the trajectory. The complexity of the algorithm is relatively high.

### 2.1.3 *SSPD* distance

In order to deal with whole trajectory sets and improve the efficiency, we adapt *SSPD* distance into distance matrix computation of clustering process.

The distance  $D_{pt}$  from a point  $p$  to a trajectory  $T$  is the minimum of distances between this point and all segments  $s$  that compose  $T$ . The *Segment-Path Distance (SPD)* from trajectory  $T_1$  to trajectory  $T_2$  is the mean of all distances from points composing  $T_1$  to the trajectory  $T_2$ .

$$D_{SPD}(T_1, T_2) = \frac{1}{n_1} \sum_{i_1=1}^{n_1} d_{pt}(p_{i_1}^1, T_2)$$

$$d_{pt}(p_{i_1}^1, T_2) = \min_{i_2 \in [0, \dots, n_2-1]} d_{ps}(p_{i_1}^1, s_{i_2}^2)$$

In order to make *SPD* have symmetrical characteristic, *SSPD* is defined as:

$$D_{SSPD}(T_1, T_2) = \frac{D_{SPD}(T_1, T_2) + D_{SPD}(T_2, T_1)}{2}$$

Comparing with *OWD* distance:

- *SSPD* distance pays more attention to the original actual data information, because *OWD* distance uses interpolation to complete the trajectory. Inheriting the advantage of *OWD* distance at the same time (not easily affected whole similarity by local trajectory), because *SSPD* distance uses mean instead of the maximum to express similarity.

- From the map as Figure 3 and Figure 4, there are many trajectories which shares similar trajectory (origin, destination and route) clustered into different color (categories), which is unreasonable.

In this case, *SSPD* distance will be adapted into trajectory clustering process as the distance measurement.

## 2.2 Reconstruction of low-rank matrix

From trajectory clustering, we can get to know the group of trajectories in mobility data set, but the missing values also remain. For certain missing part of value of one ID, we can recovery the possible value through low-rank matrix reconstruction technology.

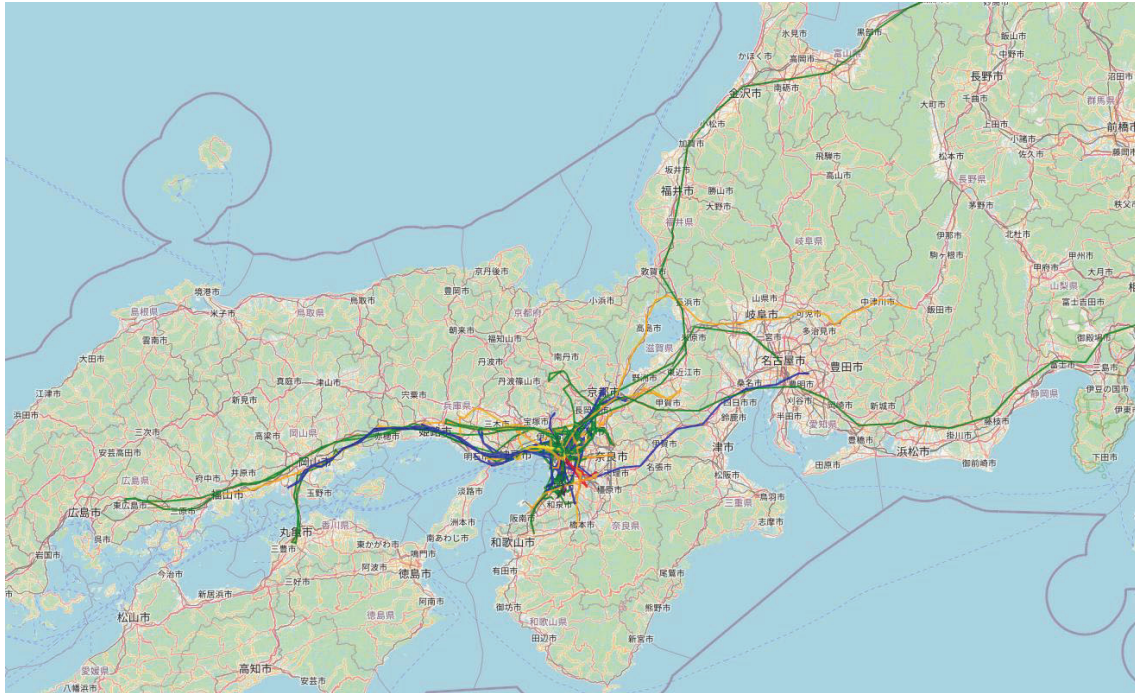


Figure 3. Trajectory clustering based on OWD distance

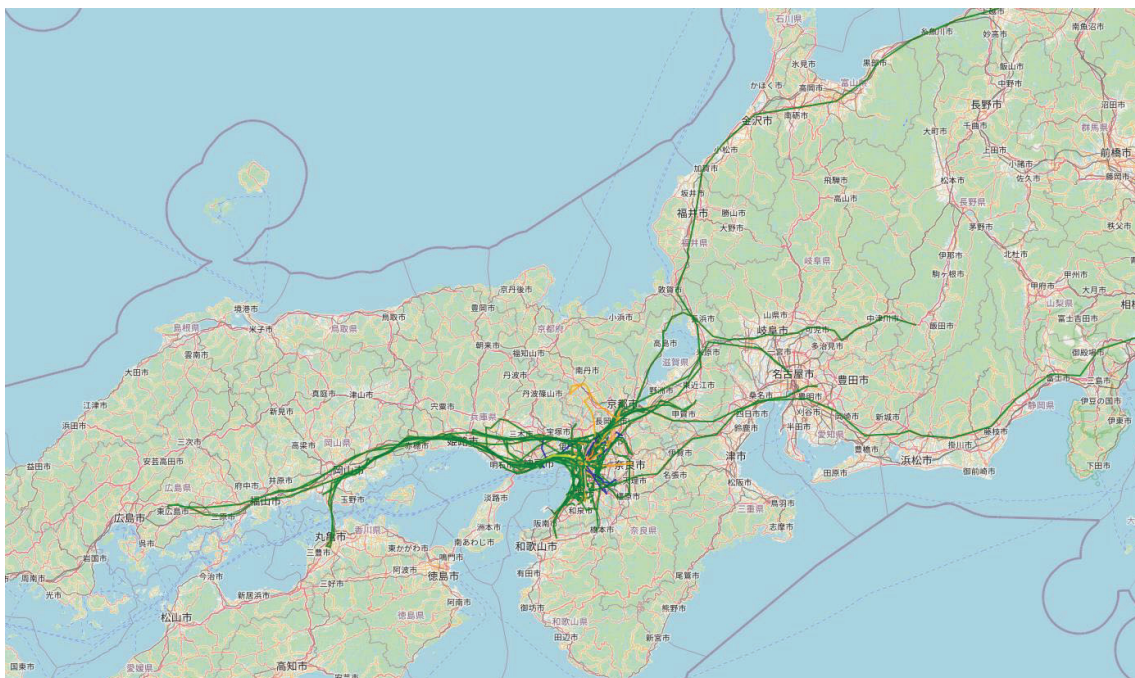


Figure 4. Trajectory clustering based on SSPD distance

### 2.2.1 Background

To make it easy to understand, we use the example of film recommendation system to explain.



Figure 5. Video grading system

Many video sites have the video grading system to know the tendency of users and give personalized recommendations to users depending on their interest. However, users may not give grade every time after finishing watching every video like Figure 5 shows. Based on the basic assumption that every user belongs to certain group, which contains users sharing similar interest with him, we can reconstruction the grade matrix using the limitation of low-rank.



### 2.2.2 Problem and method

Assume that  $D \in R^{m \times n}$  is a low-rank matrix with missing elements, in order to reconstruct the matrix, we can divide matrix  $D$  into the sum of two matrices  $D = A + E$ . [10] We don't know the detailed elements of  $A$  and  $E$ , but the limitation is that  $A$  is a low-rank matrix. We can reconstruct matrix  $D$  through solving optimization problem:

$$\begin{aligned} \min_{A,E} \|E\|_F \quad s.t. \quad & rank(A) \leq rank(E), \\ & rank(E) \ll \min(m, n), D = A + E \end{aligned}$$

We can use *SVD* (singular value decomposition) to solve the optimization problem above. For matrix  $D \in R^{m \times n}$ , the process of SVD is defined as:

$$D = USV^T = U \begin{pmatrix} \sum_0^r & 0 \\ 0 & 0 \end{pmatrix} V^T, \quad rank(D) = r$$

$U \in R^{m \times n}$  and  $V \in R^{n \times n}$  are orthogonal matrix, diagonal matrix  $\sum r = diag(\sigma_1, \sigma_2, \dots, \sigma_r) \in R^{r \times r}$ , and diagonal elements satisfy  $\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > 0$ . We note  $U = (u_1, u_2, \dots, u_m)$ ,  $V = (v_1, v_2, \dots, v_n)$ , and the *SVD* process can be transformed into:

$$D = \sum_{i=1}^r \sigma_i u_i v_i^T,$$

$u_i$  and  $v_i$  is left singular vector and right singular vector of singular value  $\sigma$ .

*Frobenius* norm of matrix  $D$  is  $\|D\|_F = \sqrt{\sum_{i=1}^r \sigma_i^2}$ , the optimized rank  $r_0$  is close to  $D_{r_0} = \sum_{i=1}^{r_0} \sigma_i u_i v_i^T$ , and  $\|D - D_{r_0}\|_F^2 = \sum_{i=r_0+1}^r \sigma_i^2, r_0 < r$ . Then apply this equation to *SVD* transformed problem, and apply solution of *SVD* into reconstruction of low-rank matrix.

### 2.2.3 Missing value recovery

Depending on the consequence of 2.2, we have the group list of overall ID in mobility data set, the number or the group can be used as rank of the matrix we want to reconstruct. Adding this limitation into process of low-rank matrix reconstruction, the problem will be simplified. Then we can get the mobility data set whose ID and time shares the same dimension without missing value of latitude and longitude.

## 2.3 Map-matching

So far, we have got the relatively completed mobility data set, but some abnormal values appear like flash phenomenon, pendulum phenomenon, etc., which are unreasonable due to the difference between real human mobility. Therefore, map-matching should be carried out to transform these abnormal values to most possible trajectory behaving like human.



The existing map-matching algorithms could be divided into four categories: **geometry**, **topology**, **probability** and **advanced**. [11]

- **Geometry**: The geometric information of known points, such as the moving distance and angle of turn of trajectory, is considered in the algorithm.
- **Topology**: Topology based algorithm utilizes road topology information to control the result.
- **Probability**: The probability method matches the points based on the probability of the case.
- **Advanced**: Advanced algorithms often consider the utilize of comprehensive information, such as Hidden Markov Model.

In this paper, the flow of map-matching will be noted step by step as here below and Figure 6 is the illustration of appearance.

**step.1** Form candidate road list using range query.

**step.2** Calculate travel likelihood for all roads in candidate road list.

**step.3** Is Vehicle or human off-road? Yes→**step.2**; No→**step.4**.

**step.4** Map GPS-estimated position to identified road of travel.

**step.5** Generate successor roads? Yes→**step.6**; No→**step.2**.

**step.6** Add successor roads to candidate road list and delete roads with low travel likelihood.

In order to minimize the amount of computation required, we do not attempt to compare the trajectory of GPS-estimated positions to each road in the network. Instead, we limit the computations to a small number of roads that are maintained in a candidate road list. As the vehicle or human moves along the road network, the contents of the candidate road list are updated if one of the following two situations occurs.

- When the vehicle or human is off-road. A vehicle or human may occasionally leave the road network to enter its destination such as car park or working place, etc. An off-road condition can also occur when the vehicle enters a road that is not represented in the digital map. In such cases, the candidate road list is continuously updated by assigning to it all the roads that lie within a certain distance of the current GPS-estimated position. This range of query was chosen to be twice that of the error range of the GPS receiver.
- When the vehicle has travelled a significant portion of the road that is identified as the current road of travel. In that case, the successor roads that are topologically connected to the upcoming intersection are added to the candidate road list. The older roads in the candidate road list, which are less likely to be the road of travel, are removed from the list.



Figure 6. Map-matching appearance

## **Chapter 3. Video Summarization**

### **3.1 Problem and goal**

With the filtered mobility data, we can now show the dynamic flow of human mobility in map like the video frames. However, due to the differences between characteristic of mobility data and video data, before adapting the similar method from video summarization, applicability analysis will be done to select the most suitable method.

### **3.2 Approaches**

Concerning the adopted training strategy, the existing deep-learning-based video summarization algorithms can be coarsely categorized in the following categories [12][13]:

#### **3.2.1 Supervised video summarization**

Supervised approaches that rely on datasets with human-labeled ground-truth annotations, based on which they try to discover the underlying criterion for video frame selection and video summarization. It is detailly divided as follow:

- Learn frame importance by modeling the temporal dependency among frames.
- Learn frame importance by modeling the spatiotemporal structure of the video.
- Learn summarization by fooling a discriminator when trying to discriminate a machine-generated from a human-generated summary.

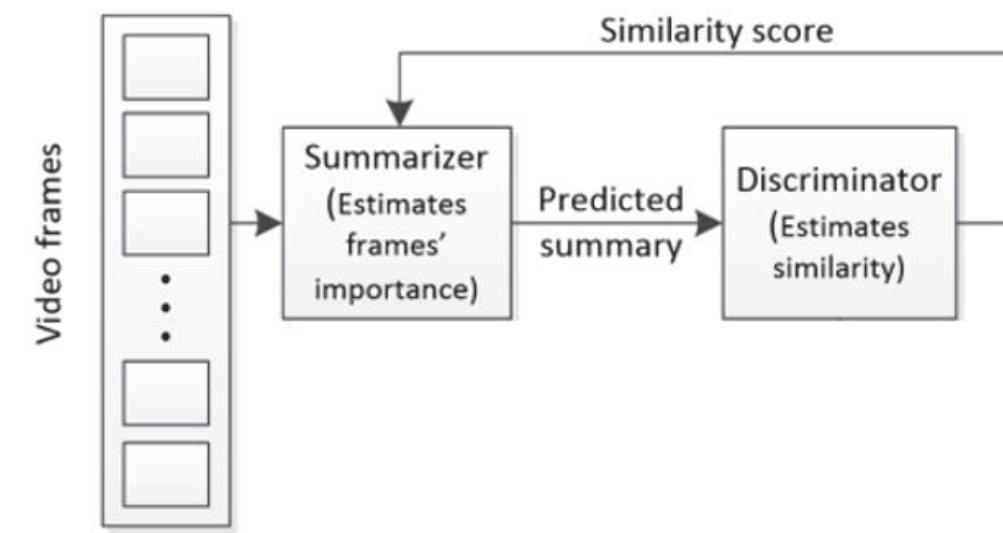


Figure 7. Flow of supervised video summarization

### 3.2.2 Unsupervised video summarization

Unsupervised approaches that overcome the need for ground-truth data, based on learning mechanisms that require only an adequately large collection of original videos for their training [14]. It is detailly divided as follow:

- Learn summarization by fooling a discriminator when trying to discriminate the original video from a summary-based reconstruction of it.
- Learn summarization by targeting specific desired properties for the summary.
- Build object-oriented summaries by modeling the key-motion of important visual objects.

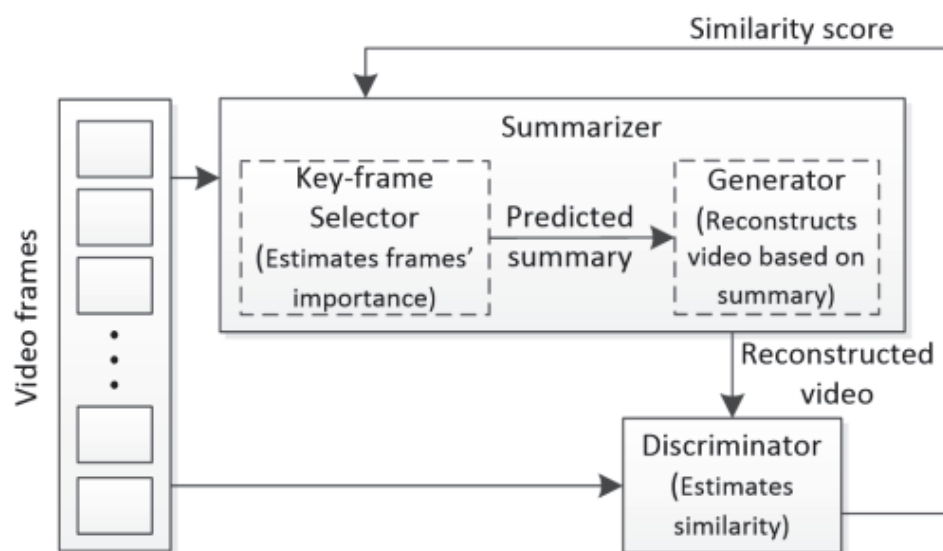


Figure 8. Flow of unsupervised video summarization

### **3.2.3 Weakly-supervised video summarization**

Weakly-supervised approaches that, similarly to unsupervised approaches, aim to alleviate the need for large sets of hand-labeled data. Less-expensive weak labels are utilized with the understanding that they are imperfect compared to a full set of human annotations, but can nonetheless be used to create strong predictive models.

## Chapter 4. Transformation and Extraction

### 4.1 Density dissimilarity matrix

From the mobility data that has been filtered and interpolated, based on the common sense of population density formula: **Population Density = Number of People in Range/ Area of Range**, we can artificially divide the map into several blocks, whose range could be determined by parameters. Then we count the population in every range, and person who is coincidentally drop on the boundary of range will be treat as 0.5 and count into both ranges.

Finishing computing the population density of every time generated in interpolation process, we can get population density matrix set:

$$\wp = \{P_{t_1}^{m \times n}, P_{t_2}^{m \times n}, \dots, P_{t_s}^{m \times n}\}, \quad t_{i+1} - t_i = \text{interpolation time unit},$$

$$m = \frac{\text{latitude}}{\Delta \text{latitude}}, n = \frac{\text{longitude}}{\Delta \text{longitude}}, \quad s = \text{number of interpolation}$$

According to *Jaccard* distance, which is also known as *Jaccard* dissimilarity coefficient and used to measure the dissimilarity between sample sets [15]. If A and B are two sample sets, *Jaccard* distance is defined as:



$$d_J(A, B) = 1 - J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

We adapt *Jaccard* distance to compute dissimilarity between each population density matrix  $P_{t_a}^{m \times n}$  and  $P_{t_b}^{m \times n}$  ( $a, b \in [1 \dots s]$ ), and formula will be transformed as:

$$d_J(a, b) = d_J(P_{t_a}^{m \times n}, P_{t_b}^{m \times n}) = 1 - \frac{\sum_n \sum_m \min(P_{t_a}^{i,j}, P_{t_b}^{i,j})}{\sum_n \sum_m \max(P_{t_a}^{i,j}, P_{t_b}^{i,j})}$$

## 4.2 Methods

### 4.2.1 *K-Medoids* clustering

*K-Medoids* clustering is similar to *K-means* as they need to choose initial centers. The center point selected by *K-means* is the center of gravity of all points in the current class, while the center point selected by *K-Medoids* method is a point existing in the current cluster. The criterion function of *K-Medoids* is the minimum sum of the distances between all other points in the current cluster and the center point, which weakens the influence of outliers to some extent. Although it needs more computational time, since the consequence we want to get is the existing time of mobility data set to represent the reality time, *K-Medoids* is suitable here.

The Flow to solve *K-Medoids* clustering problem is [16]:

**step.1** Greedily select  $k$  of the  $n$  data points as the medoids to minimize the cost.

**step.2** Associate each data point to the closest medoid.

**step.3** While the cost of the configuration decreases:

**step3.1** For each medoid  $m$ , and for each non-medoid data point  $o$ :

- (1) Consider the swap of  $m$  and  $o$ , and compute the cost change.
- (2) If the cost change is the current best, remember this  $m$  and  $o$  combination.

**step3.2** Perform the best swap of  $m_{best}$  and  $o_{best}$ , if it decreases the cost function. Otherwise, the algorithm terminates.

#### 4.2.2 Two Approaches to extract key time

With population density dissimilarity  $d_J(a, b)$  between time point  $a$  and  $b(a, b \in [1 \dots s])$ , basically we have two approaches to extract key time:

- **Key Time 1:** Compute *Jaccard* distance only between neighbor time points  $a$  and  $a+1$  ( $a, a+1 \in [0 \dots s]$ ) to get the vector  $d_{J-neig} = (d_J(1,2), d_J(2,3), \dots, d_J(s-1, s))$ , and then artificially set the extraction degree  $p\%$  to extract time points with dissimilarity higher than  $p\%$  time in  $d_{J-neig}$ .

- **Key Time 2:** Construct the density dissimilarity matrix  $D_J^{s \times s} = (d_J(a, b))_{s \times s}$  based on population density dissimilarity between each time point combination. Then we adapt *K-Medoids* clustering algorithm, (which is introduced above) and artificially set the clustering grid  $k$  to select the center of every group of clustering consequence. Clustering centers represent the extracted time points.

### 4.3 Simulation and Comparison

#### 4.3.1 Key Time 1

With the dissimilarity vector  $d_{J-neig}$  in Key Time 1, based on one day of Osaka mobility data, we can draw line chart Figure 9.

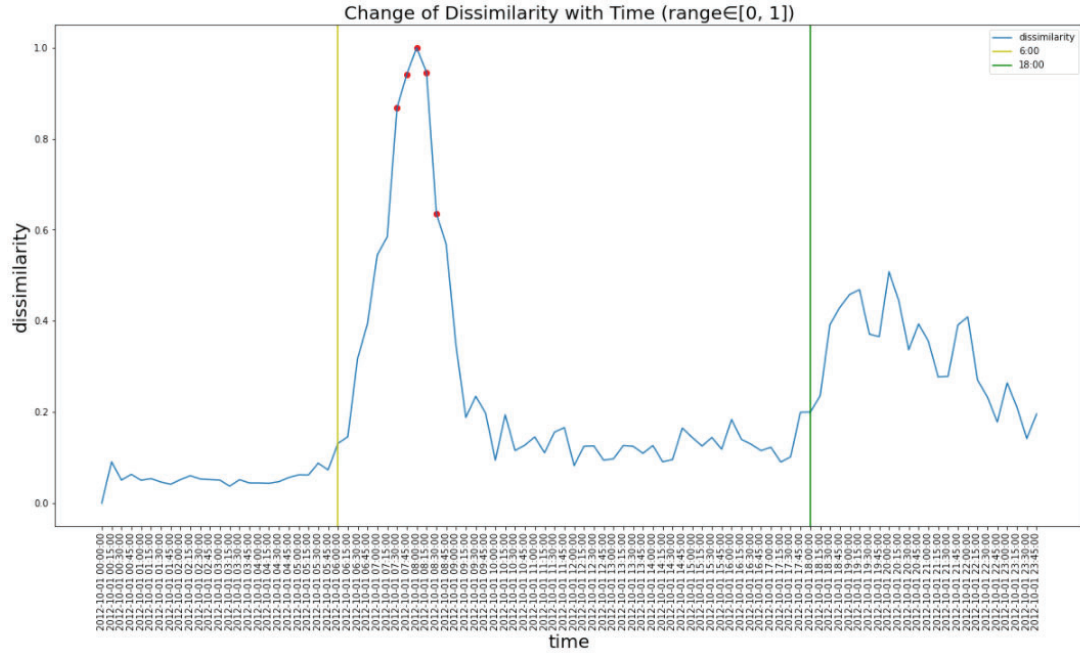


Figure 9. Polyline of dissimilarity based on Key Time 1

The blue polyline shows the change of dissimilarity during this day from 0 am to 23:45 pm with 15 minutes of time unit, the yellow line is 6 am and the green line is 6 pm. In the case that we set the  $p\%$  as 5% to extract key time points, the red points represent extracted time.

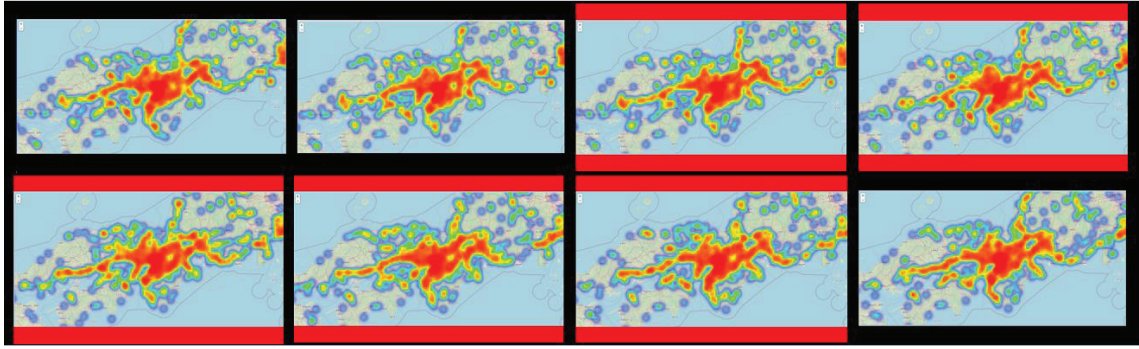


Figure 10. Human distribution of extracted time

Figure 10 shows the system simulation of the distribution change of population in mobility data set. From top left to bottom right is 7 am, 7.15 am, 7.30 am, 7.45 am, 8 am, 8.15 am, 8.30 am, 8.45 am, and the time points which are highlighted by red lines represent red points in Figure 9.

### 4.3.2 Key Time 2

With the density dissimilarity matrix  $D_J^{s \times s} = (d_J(a, b))_{s \times s}$  in Key Time 2, based on the same part of same data set, we can get Figure 11 shows the tendency of mobility situation change through dissimilarity matrix.

In Figure 11, horizontal axis and vertical axis are all time sequence from 0 am to 23.45 pm, and the color represents dissimilarity (red means high and blue means low). With 6 am and 6 pm as the boundary, it is surrounded by a purple frame at night and a green frame during the day. The area surrounded by the same color frame is mainly blue, and the area surrounded by different frames is mainly red. The meaning is that the day and night human mobility modes are very different from each other, which meets our common sense.

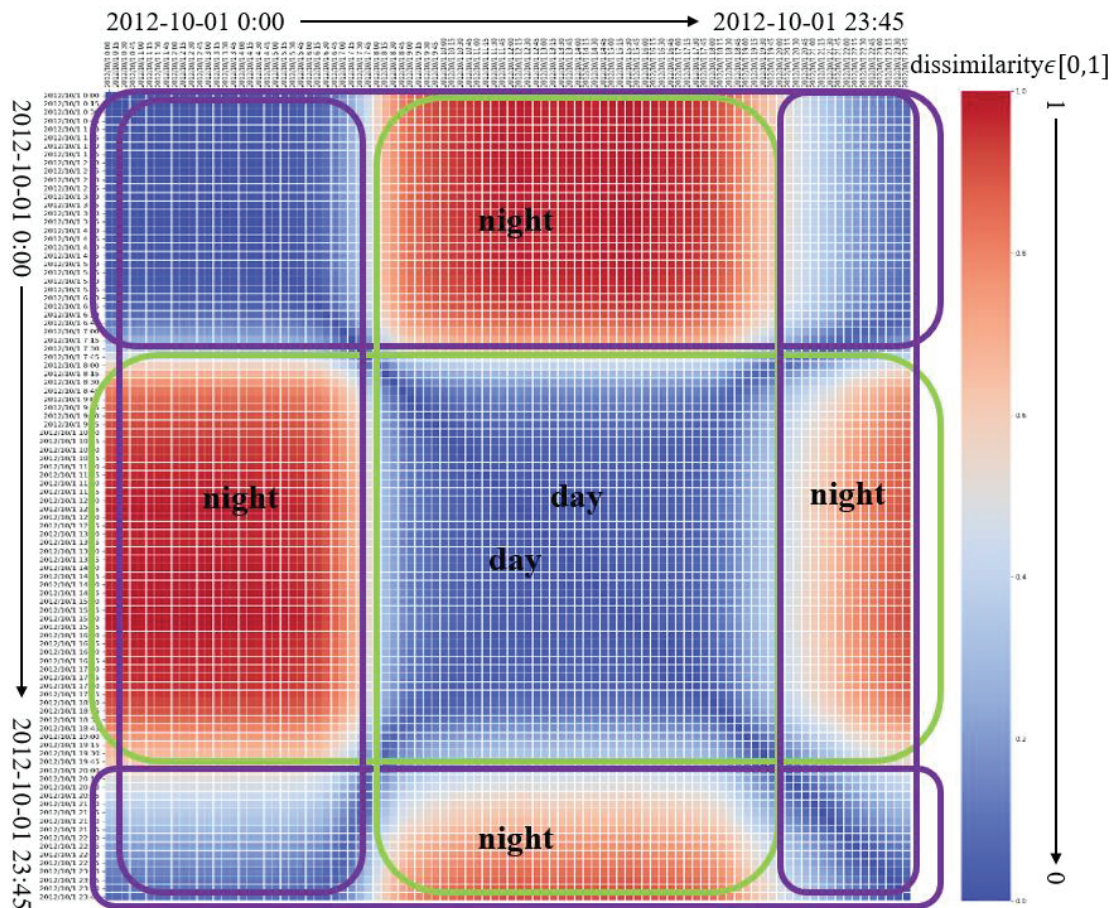


Figure 11. Dissimilarity matrix based on Key Time 2

With density dissimilarly matrix, based on *K-Medoids* clustering, we can extract key time points as Key Time 1 did. The execution consequence is 3.30 am, 8.30 am, 14.15 pm, 19.45 pm, 22.30 pm.

### 4.3.3 Comparison

Based on consequence of Key time 1 and Key time 2, we can make comparison that:

- **Computational complexity**

- Key Time 1 is based on density of adjacent, while Key Time 2 is based on density of overall time.
- Computational complexity of Key Time 1 is squared size simpler than Key Time 2.

- **Overall perspective**

- Key Time 1 pays more attention to continuous time, while Key time 2 grasps overall change and adjacent change at the same time.
- Key Time 1 extracts time points all fall in the range from 7 am to 9 am, which is the morning peak. During morning peak, it is certainly that mobility situation will vary fast, but it couldn't summarize the situation of mobility mode change.



- Extraction consequence of Key Time 2 is separated overall day, which can roughly represent 5 situation: sleeping time, morning peak, working time, dinner time, night peak.

- **Characteristic**

- Obviously, Key Time 1 couldn't complete the summary work of mobility mode, but it can deal with the situation when we need to mine out the irregularity or the change tendency of dataset, with low time cost.
- Key Time 2 can summarize the mobility mode by representative time points, but population density dissimilarity is not enough for it appearance, which will be illustrated after.

#### **4.4 Weakness**

Assume that we extract time points to summarize the mode and change tendency by method of Key Time 2, when facing with situation like Figure 12 shows, Key Time 2 will ignore the existing change of human location.

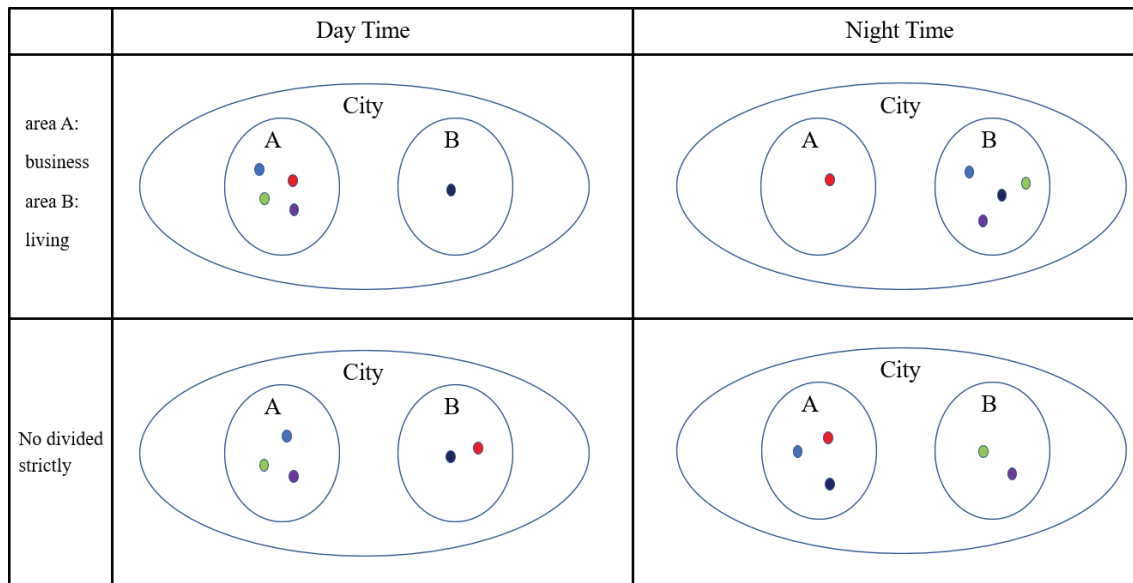


Figure 12. Situation of ignorance of human mobility

In a certain city, there are area A and area B, and different color points represent individual person.

- If A is strictly business area while B is strictly living area, during day time, population density of A will be high and B will be low, while during night time, it will be in contrast. Key time 2 can easily deal with this situation.
- Contrast to first situation, if A and B don't have obvious distinction of living and business, people may move without change of population density as below. we can't just conclude the pattern from density under this situation, because information of human movement is ignored.



In consideration of the second situation, we apply OD (Origin-Destination) similarity into extraction process. For every individual data from time point  $a$  and time point  $b$ , we compute the Euclidean distance of the location from different time segments, and then normalize it into a degree representing the similarity between  $a$  and  $b$  to construct the OD similarity-based matrix

$$D_{OD}^{s \times s} = (d_{OD}(a, b))_{s \times s}, s = \text{number of interpolation},$$

With the OD similarity joining into the penalty function, even human mobility without change of population density will be computed into extraction.

## Chapter 5. Submodular Optimization

### 5.1 Definition

In mathematics, a submodular function is a set function whose value, informally, has the property that the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases. Submodular functions have a natural diminishing returns property which makes them suitable for many applications.

Submodularity is a property of set functions and is defined relying on a notion of discrete function, often also called the marginal gain.

#### Definition 1. Discrete derivative

For a set function  $f: 2^V \rightarrow \mathbb{R}$ ,  $S \subseteq V$ , and  $e \in V$ , let  $\Delta_f(e | S) = f(S \cup \{e\}) - f(S)$  be the discrete derivation of  $f$  at  $S$  with respect to  $e$ .

Where the function  $f$  is clear from the context, we drop the subscript and simply write  $\Delta(e | S)$

**Definition 2. Submodularity**

A function  $f: 2^V \rightarrow \mathbb{R}$  is submodular if for every  $A \subseteq B \subseteq V$  and  $e \in V \setminus B$  it holds that

$$\Delta(e | A) \geq \Delta(e | B).$$

Equivalently, a function  $f: 2^V \rightarrow \mathbb{R}$  is submodular if for every  $A, B \subseteq V$ ,

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$$

Suppose we interpret  $S \subset V$  as a set of actions which provide some benefit  $f(S)$ . Then as the definition says, for a submodular function  $f$ , after performing a set  $A$  of actions, the marginal benefit of any action  $e$  does not increase as we perform the actions in  $B \setminus A$ . Therefore, submodular set functions exhibit a natural diminishing returns property.

**5.2 Examples**

To easy understand, we will consider the problem of deploying sensors in a drinking water distribution network (Figure 13) in order to detect contamination. In this domain, we may have a model of how contaminants, accidentally or maliciously introduced into the network, spread over time. Such a model then allows to quantify the benefit  $f(A)$  of locations in terms of the detection performance (such as average time to detection). Based on

this notion of utility, we then wish to find an optimal subset  $A \subseteq V$  of location maximizing the utility,  $\max_A f(A)$ , subject to some constraints (such as bounded cost) [17]. As the illustration of the diminishing returns effect in context of placing sensors in a water distribution network to detect contaminations. The blue regions indicate nodes where contamination is detected quickly using the existing sensors  $S$ . The red region indicates the additional coverage by adding a new sensor  $s'$ . If more sensors are already placed (b), there is more overlap, hence less gain in utility:  $\Delta(s' | \{s_1, s_2\}) \geq \Delta(s' | \{s_1, \dots, s_4\})$  [18].

This application requires solving a difficult real-world optimization problem, that can be handled with submodularity optimization theory.

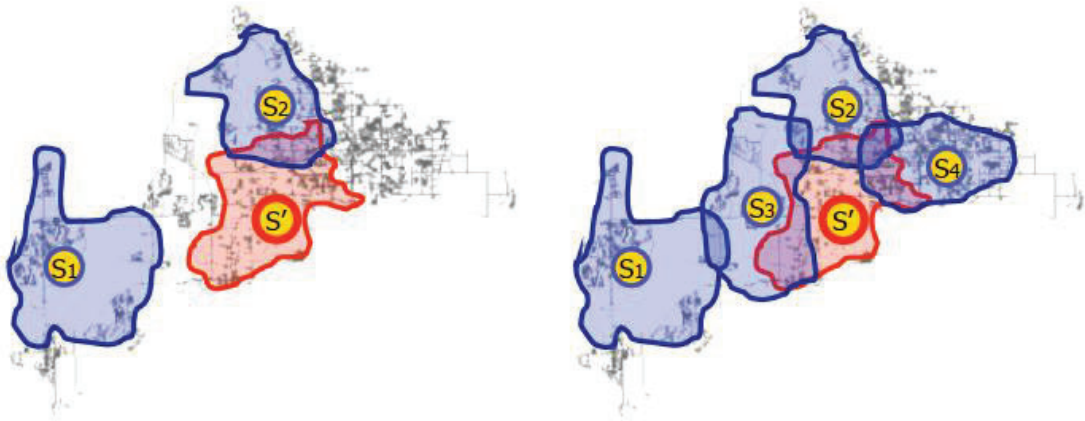


Figure 13. Drinking water distribution network simulation

### 5.3 Consequence analysis

Based on Submodularity Optimization and its greedy methodology, we can extract key time points from mobility data set. For comparison, here we use Tokyo mobility data, which is filtered through the same process like Osaka mobility data, containing 2 weeks of mobility data. In order to show the stability of dealing with several situation, if time point  $t$  is in the format as ' $y$  year- $m$  month- $d$  day  $h$  hour:  $m$  minute:  $s$  second', we transform the extracted time point  $t$  by the formula:

$$transform(t(y, m, d, h, m, s)) = \frac{3600h + 60m + s}{86400},$$

and then use standard deviation to measure the stability.

Date	Day of Week	Extracted time1	Extracted time2	Extracted time3	Extracted time4	Extracted time5	standard deviation
2012/1/1	Sun	0.14583	0.35417	0.59375	0.82292	0.9375	5.48%
2012/1/2	Mon	0.11458	0.3125	0.5625	0.80208	0.92708	9.35%
2012/1/3	Tue	0.13542	0.33333	0.58333	0.80208	0.92708	8.76%
2012/1/4	Wed	0.13542	0.34375	0.57292	0.80208	0.92708	8.23%
2012/1/5	Thu	0.13542	0.33333	0.55208	0.79167	0.91667	7.46%
2012/1/6	Fri	0.13542	0.34375	0.57292	0.79167	0.91667	7.83%
2012/1/7	Sat	0.14583	0.35417	0.54167	0.76042	0.90625	4.86%
2012/1/8	Sun	0.125	0.3125	0.57292	0.83333	0.9375	6.42%
2012/1/9	Mon	0.17708	0.4375	0.57292	0.70833	0.84375	13.48%
2012/1/10	Tue	0.13542	0.38542	0.52083	0.73958	0.91667	7.54%
2012/1/11	Wed	0.14583	0.39583	0.51042	0.72917	0.89583	8.36%
2012/1/12	Thu	0.13542	0.39583	0.52083	0.75	0.91667	7.94%
2012/1/13	Fri	0.17708	0.34375	0.5625	0.73958	0.91667	7.68%
2012/1/14	Sat	0.14583	0.34375	0.5625	0.8125	0.92708	4.98%
							7.74%

Table 1. Submodularity and OD matrix based standard deviation

Date	Day of Week	Extracted time1	Extracted time2	Extracted time3	Extracted time4	Extracted time5	standard deviation
2012/1/1	Sun	0.18749	0.39583	0.59375	0.83468	0.9375	9.39%
2012/1/2	Mon	0.09783	0.3125	0.5625	0.72917	0.92708	11.25%
2012/1/3	Tue	0.13542	0.30425	0.56353	0.80208	0.91667	10.74%
2012/1/4	Wed	0.9375	0.35417	0.51042	0.80208	0.92708	10.86%
2012/1/5	Thu	0.13542	0.33333	0.55208	0.82475	0.92036	7.58%
2012/1/6	Fri	0.1042	0.34375	0.57292	0.79167	0.91667	9.86%
2012/1/7	Sat	0.16667	0.35745	0.54167	0.76042	0.90625	6.35%
2012/1/8	Sun	0.13542	0.39583	0.57292	0.82475	0.9375	7.46%
2012/1/9	Mon	0.13542	0.4375	0.56353	0.70833	0.86498	10.89%
2012/1/10	Tue	0.1042	0.38542	0.52083	0.73958	0.90994	9.44%
2012/1/11	Wed	0.13542	0.35417	0.51437	0.72917	0.92708	10.38%
2012/1/12	Thu	0.9479	0.39583	0.52083	0.78645	0.92036	8.84%
2012/1/13	Fri	0.17708	0.33333	0.5625	0.73958	0.91667	9.73%
2012/1/14	Sat	0.14583	0.3275	0.5625	0.84275	0.92708	7.26%
							9.29%

Table 2. Key Time 2 based standard deviation

According to the average standard deviation of Table 1 and Table 2, 7.74% is lower than 9.29%, which means Submodularity and OD matrix based standard deviation shows better appearance on grasping the mobility mode and it change tendency. The standard deviation of 2012/1/9 is highlighted by red color due to its relatively higher than other standard deviation. According to the calendar, 2012/1/9 is coincidentally Seijin No Hi, which can explain the abnormal mobility mode to some extent.

## Chapter 6. Conclusions and Future Work

In this research, 3 methods is proposed to address the problems of key time extraction for human mobility change detection. We analyze the approaches based on their appearance and computational complexity.

Comparing to other research, the main contribution of this research is as follows:

- Low-rank matrix based missing value reconstruction is adapted to mobility data filtering process with the stable appearance.
- Based on *Jaccard* distance, dissimilarity between time units is computed into the format of matrix, with which two approaches was proposed to extract key time.
- Based on the appearance and analysis, we apply new parameter OD similarity into the origin penalty function. And then the appearance of time extraction is better than the one without taking OD similarity into thought.

## Bibliography

- [1] Yue, Y., Lan, T., Yeh, A. G. O., & Li, Q.-Q. (2014). Zooming into individuals to understand the collective: A review of trajectory-based travel behavior studies. *Travel Behavior and Society*, 1, 69-78.
- [2] Ahmed M, Wenk C. (2012). Probabilistic street-intersection reconstruction from GPS trajectories: approaches and challenges. *Proceedings of the Third ACM SIGSPATIAL International Workshop on querying and mining uncertain spatio-temporal data, 2012*, 34-37.
- [3] Liu, K., Gao, S., & Lu, F. (2019). Identifying spatial interaction patterns of vehicle movements on urban road networks by topic modelling. *Computers, Environment and Urban Systems*, 74, 50-61.
- [4] Y. Zhao, “Vehicle Location and Navigation Systems”, Artech House, 1997.
- [5] M. Barbieri, L. Agnihotri, and N. Dimitrova, “Video summarization: methods and landscape,” in *Internet Multimedia Management Systems IV*, J. R. Smith, S. Panchanathan, and T. Zhang, Eds., vol. 5242, International Society for Optics and Photonics. SPIE, 2003, pp. 1-13
- [6] Ying Li, Shih-Hung Lee, Chia-Hung Yeh, and C.-J. Kuo, “Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques,” *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 79–89, 2006.



- 
- [7] B. T. Truong and S. Venkatesh, "Video Abstraction: A Systematic Review and Classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, p. 3–es, Feb. 2007.
  - [8] Spatial Interpolation Methods: A Review, Nina Siu-Ngan Lam, Page 129-150, 14 Mar 2013.
  - [9] Review & Perspective for Distance Based Trajectory Clustering, Philippe Besse, Universite de Toulouse INSA, Institut de Math ´ ematiques UMR CNRS 5219.
  - [10] An Overview of Low-Rank Matrix Recovery From Incomplete Observations, *IEEE Journal of Selected Topics in Signal Processing*, June 2016.
  - [11] A Map Matching Method for GPS Based Real-Time Vehicle Location, G.R. Jagadeesh, *Journal of Navigation*, Sept 2004.
  - [12] L. Lebron Casas and E. Koblents, "Video Summarization with LSTM and Deep Attention Models," in *MultiMedia Modeling*, I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, Eds. Cham: Springer International Publishing, 2019, pp. 67–79.
  - [13] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing Videos with Attention," in *Asian Conf. on Computer Vision (ACCV) 2018 Workshops*, G. Carneiro and S. You, Eds. Cham: Springer International Publishing, 2019, pp. 39–54.
  - [14] A. G. del Molino, C. Tan, J. Lim, and A. Tan, "Summarization of Egocentric Videos: A Comprehensive Survey," *IEEE Trans. on Human Machine Systems*, vol. 47, no. 1, pp. 65–76, Feb 2017.

- [15] Using of Jaccard Coefficient for Keywords Similarity, Supakit Niwattanakul\*, Jatsada Singthongchai, Ekkachai Naenudorn and Supachanun Wanapu, Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013, Hong Kong.
- [16] A simple and fast algorithm for K-medoids clustering, HS Park, CH Jun - Expert systems with applications, 2009 - Elsevier.
- [17] Krause, Andreas, Leskovec, Jure, Guestrin, Carlos, VanBriesen, Jeanne, and Faloutsos, Christos. 2008b. Efficient Sensor Placement Optimization for Securing Large Water Distribution Networks. Journal of Water Resources Planning and Management, 134(6), 516{526.
- [18] Krause, Andreas, McMahan, Brendan, Guestrin, Carlos, and Gupta, Anupam. 2008c. Robust Submodular Observation Selection. Journal of Machine Learning Research (JMLR), 9(December), 2761{2801.