Ph.D. Thesis

Summarization and Visualization of Movement Trajectories

Yuri Miyagi

Advanced Science

Graduate School of Humanities and Sciences

Ochanomizu University

March, 2020

Abstract

Various methods have been proposed to record and analyze trajectory dataset. Such a dataset brings us beneficial knowledge in various fields, for example, prevention of disasters, improvement of marketing. However, the dataset may be enormous as we got to easily accumulate them, we need to improve techniques to analyze and display the results. This thesis proposes visualization techniques to find features of several trajectory datasets, specifically walking routes of people and eye-tracking scan-paths. Results of case studies indicate several characteristic behaviors found from the trajectories.

Keywords : Information Visualization, Trajectory, Path, Symbolic, String

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The utilization of trajectory data has been active in our society. Trajectory datasets are sets of records that mainly includes when and where creatures or moving objects passed, such as followings[28]:

- Walking of people
- Migration of animals
- Traffic of vehicles, ships, and airplanes
- Movement of typhoons[29]

These trajectories take various shapes and distributions, for example, "Crossings of independent individual paths," "Gathering of several similar paths like one thick bund," according to environmental elements. Furthermore, spaces which include the trajectories get various scales, from a small room in a couple of meters square, to several nations and continents[30].

Figure 1.1 shows a brief processing flow of the analysis of such trajectories. The development of devices for recording trajectories enabled us to collect such various datasets[31]. A simple small-case is, for example, setting laser sensors in a room or ask visitors to carry small devices to record positions periodically. Another popular

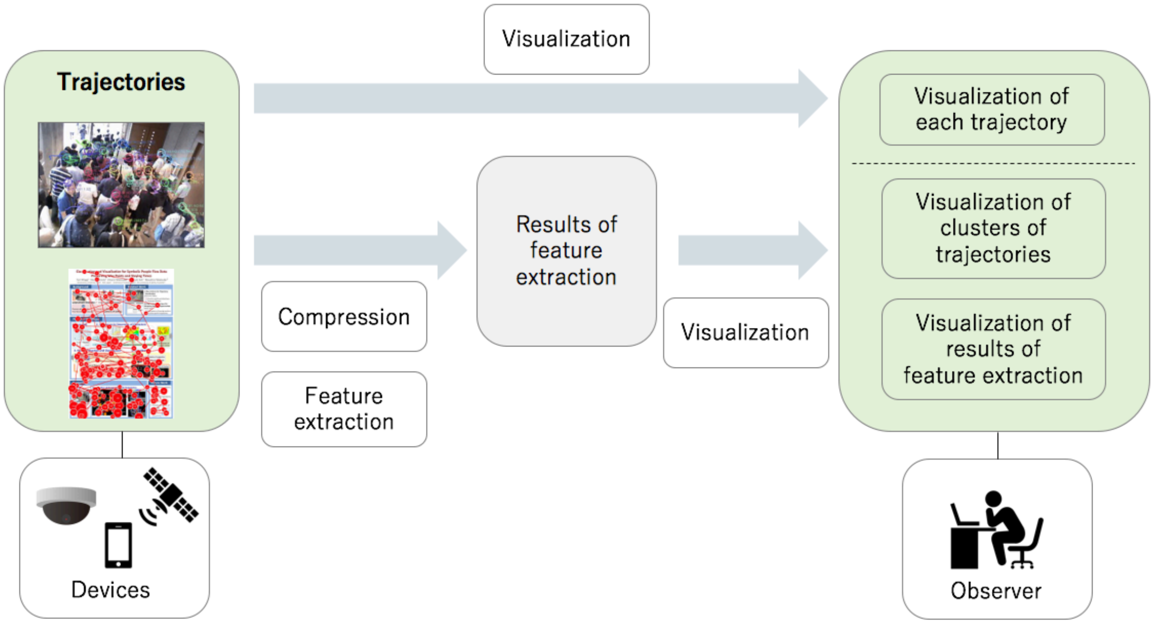


Figure 1.1: Flow of trajectory analysis.

approach is movie processing that extracts pedestrians or moving things from a recorded movie and estimates their positions. In recent years, GPS is also a major technology to collect position datasets in wide areas. GPS is effective to observe movements in relatively wide areas than the above mentioned technologies. Especially, smartphones help to get the movement of not only particularly limited participants of experiments but also general people in daily life. As above, we can collect diverse types of trajectory data in different environments. The scale and accuracy of the datasets have been increasing.

We can get useful knowledge from such trajectory datasets. For example, the followings are major fields related to the collection and analysis of trajectories.

- Prevention of disasters
- Improvement of the traffic system
- Support of marketing

In disaster prevention, analyzing movements of people and vehicles in abnormal circumstances are major tasks. Insights from such analysis are utilized to design better evacuation routes, and estimate times to complete the evacuation. Not only an analysis of existing data but also a simulation of behaviors is also an important task. In the field of traffic, finding crowded places with dense trajectories is a simple but important task. We can discuss how to control flow pedestrians or vehicles to realize smooth traffic, based on the analysis results. Targets of analysis include stations, plaza, intersection, and any event grounds. For marketing, we can utilize records of walking routes of customers but also eye-tracking scan-paths. Accesses to products by such the trajectories indicate which products attracted the customers. Especially, finding major paths that connect different products is helpful to find an attractive combination of products to sell at once and increase sales. As above, the mentioned trajectories represent popular phenomena in our daily life, and we can use them to understand environments and carry out various tasks such as behavior prediction.

Here, we need to establish how to analyze trajectories and display the result of enormous input datasets. Especially, we have to cope with the following problems to display features and trends of trajectory dataset.

- Simplification of the dataset
- Finding characteristic trajectories
- Displaying results relevant to each user

This process is effective to reduce time resources to analyze large or too detailed trajectory datasets, even if this is not always necessary for completing the analysis of trajectories as computers have a large memory space enough to preserve a large trajectory dataset. The second problem, extraction of characteristic trajectories, is essential when we deal with hundreds or thousands of trajectories. This task may be partially different according to types of data and goals of users, such as "Finding represent trajectory that indicates trends of behavior," and "Extracting irregular suspicious paths." Finally, we have to devise how to show the results of the analysis.

There have been various users who observe trajectories as there are many types of trajectories. We have to develop techniques to represent useful information from large datasets interactively and without too complicated representations or interactions.

Here, we focus on the third problem especially. One of the popular solutions for the third problem is visualization of trajectories. We can classify the visualization methods as follows based on which aspects of trajectories are shown.

Type 1 Drawing detailed individual trajectories

Type 2 Displaying summarized trajectories or groups of paths

Type 3 Showing feature values of trajectories without shapes of trajectories

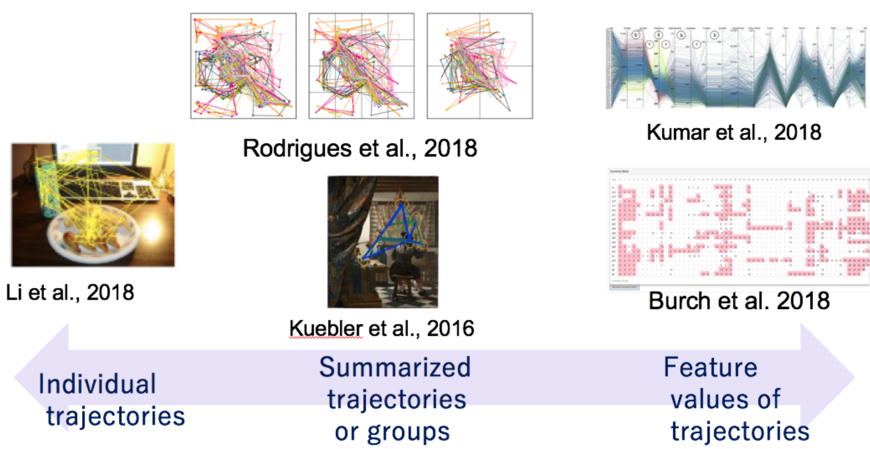


Figure 1.2: A simple classification of visualization methods for trajectory data.

Figure 1.2 shows some examples of visualization techniques. The first type is the illustration of when and where each individual trajectory passed on maps or movies. This is the simplest representation effective to understand real movement on the trajectories, for a wide range of users including beginners of visualization. The users can efficiently compare the drawn trajectories on a single view if the trajectories are few and relatively short. On the other hand, this method often provides declining visibility when users input a large number of trajectories.

The second type shows simplified shapes of trajectories. Users have to remove excess parts of trajectories or classify them[32] as the preprocessing of the visualization process. Such methods can keep high visibility by preventing to show too many data elements while using simple expressions as same as the first type of visualization. However, inappropriate preprocessing may affects the results of visualization and indicates incorrect or misunderstanding information.

The third type does not draw shapes of trajectories differently from the above two types. Instead, users have to observe visualization methods that are suitable for large datasets, such as PCP (Parallel Coordinate Plots) or matrices. Either, Origin-Destination network or radial graph are used to highlight the start- and end-points of each trajectory. Such methods are effective to show large datasets in a sight or subdivided characteristics, but sometimes difficult to understand and need the experiences to use.

The first and second types of visualization methods are standard and simple approaches and do not look to have issues to be improved. Furthermore, as mentioned above, the direct application of such a method is not effective for the current large trajectory datasets. Meanwhile, we suppose the increasing variety of trajectories and observers. Non-expert visualization uses such as the owners of shops and constructers of events are typical new analysts of such datasets we suppose. Thus, showing useful information and removing excess elements while using basic simple drawings of trajectories seem still useful. Such types of visualization methods still need to be improved to deal with large amounts of trajectory datasets although there have been various proposed methods. [28] The following issues mainly remain.

- Using both drawing individual trajectory and expression of a group of them properly

- Suggestion on the proper classification of behaviors and division of regions in space

- Comparison of multiple datasets

The first issue is about the expression of trajectories. Proper expressions of trajectories depend on what users want to visualize and flexible switching is required. For example, “piling many trajectories” and “drawing a particular shape which means a group of trajectories” may give us different impressions even if they represent the same collection of trajectories. Flexible switching of representation can lead to obtaining various information on both rough and detailed characteristics of datasets. The second issue is about the condition of the classification of a dataset. Observers often want to classify recorded trajectories according to their shapes or positions. The condition and difficulty of the classification depends on what information the observers want to find. The process may be difficult in some cases without supposed routes, for example walking patterns in a vacant square. Therefore, the visualization system should flexibly show various classification results. The third issue is the comparison of different datasets. There are many tasks on the comparison, such as two particular trajectories or differences in each time period. The development of comparative visualization methods is still an open problem. It is possible to develop a visualization arranging multiple views as many as the datasets; on the other hand, using a small number of views and only show common or different parts between the datasets can be effective to directory indicate features.

1.2 Contribution of This Research

This thesis proposes the following two visualization techniques based on the above mentioned backgrounds and tasks. Briefly, we convert trajectories to simple strings and extract their features, and finally visualize the extracted features, which to solve the above three problems.

Visualization of people flow

The first technique is to analyze the dataset which indicates the movements of pedestrians. Specifically, the trajectories are recorded by cameras set on the upper walls of some rooms. The technique finds various features like "How kind of behaviors the recorded people had?," "How different movements occurred in different time periods?" as a preprocessing. The visualization view contains drawing specific trajectories.

Visualization of eye tracking data

The other visualization is for visualizing eye-tracking data. We visualize tens of eye-tracking scan-paths and display common behaviors such as transitions and gazing, and similarities of the paths. Differently from the people flow visualization, we combine multiple visualization views on a single window.

These techniques deal with the above issues as following Table 1.1. Detail of each function are mentioned in Chapter 3 and 4.

Table 1.1: Our proposed techniques and tasks on trajectory analysis.

	Technique 1 Visualization of people flow	Technique 2 Visualization of eye-tracking data
Compression of dataset	Conversion by UniversalSAX	-
Pattern extraction	Clustering based on Levenshtein Distances	Pattern extraction by N-gram
Showing individual trajectory and groups	Selection of drawing a trajectory or clustering results	Pattern visualization selected by users
Classification of datasets and spaces	Region division by UniversalSAX	Interactive hierarchical AOIs
Comparison of dataset	-	Comparison of patterns in each trajectory

1.3 Usage Scenario of the Presented Technique

Scenario 1: Finding attractive objects

The density of trajectories is often useful information to understand the features of the data. For example, we would like to find when and where many trajectories appeared. Crowded places in a shop is a typical example as products may attract many customers if they passed around the products. In particular, people flow and eye-tracking data can reflect people's interests strongly because such data record how pedestrians and participants behaved where they can freely act. The techniques can apply to find places with dense trajectories, and common patterns in multiple trajectories focusing on shapes and directions.

Scenario 2: Suggestion for improving layouts

Observers may have particular assumptions of the ideal movements of trajectories in some cases of analyses. For example, visitors are supposed ideal in a museum if they follow the fixed routes and look at many exhibits. In other words, objects with a small number of accesses found from the trajectories sometimes have positional problems. Furthermore, the shapes and lengths of trajectories between departure and destination may be important. Simple trajectories are suitable to reduce the effort of movement. Such an evaluation of existing trajectories indicates how to improve the layout of spaces and control flow of the trajectories.

1.4 Contents of This Thesis

The remainder of this thesis is organized as follows. Chapter 2 introduces related studies on the visualization of trajectories. Chapter 3 presents the technique to visualize people flow trajectory data, and Chapter 4 presents another technique to analyze eye tracking scan-paths. Chapter 5 conclude this thesis.

Chapter 2

Related Work

This chapter introduces a method for extracting patterns from trajectory data. Section 2.1 introduces a method of symbolizing trajectories and methods for analyzing behaviors. Section 2.2 describes techniques on trajectory visualization classifying into three categories according to more detailed tasks. Section 2.3 summarizes the contents of these sections.

2.1 Pattern Extraction for Understanding Movement

This section introduces techniques to extract features from trajectory datasets.

2.1.1 Conversion of Trajectories and Pattern Extraction

The method of symbolizing the trajectory dataset has already been popular. There are roughly two targets of converting a trajectory into a string: "reducing the amount of data" and "conversion to a form that facilitates feature extraction." The following method mainly aims at reducing the amount of data. Konomi et al. [33] improved index system named I-TREE for large trajectory datasets. The system can generate indices for chronological datasets and reduce computation times required to search particular elements. They had the experiments using simulated people flow datasets and search for specified data elements. Studies on symbolization-based trajectory data compression have been inactive due to the evolution of data storage technology in recent

Customer	Time Stamp	X	Y	State	Tag No.
Nancy	15523509	25	46	Stationary	1021
Nancy	15523512	10	14	Stationary	1021
Nancy	15523525	150	204	Move	1032
Nancy	15523537	25	30	Stationary	1032
Nancy	15523544	156	42	Move	1032
Nancy	15523561	58	20	Stationary	1042
Nancy	15523572	201	70	Move	1062
Nancy	15523579	43	32	Stationary	1062
Nancy	15523585	150	66	Move	1062

a) Raw data about customer movement obtained by using RFID.

Tag No.	Sales area	Floor ID
1021	Vegetable	A
1025	Egg	B
1032	Daily	C
1042	Fish	F
1062	Meat	M

b) Layout Mapping Table.

Customer	Time Stamp	State	Tag No.	Floor ID
Nancy	15523509	Stationary	1021	A
Nancy	15523512	Stationary	1021	A
Nancy	15523537	Stationary	1032	C
Nancy	15523561	Stationary	1042	F
Nancy	15523579	Stationary	1062	M

c) Pre-processed data.

Customer	Visiting area String
Nancy	AACFM

d) Transformed String Pattern.

Figure 2.1: Flow of generating strings which presents way points of customers from RFID dataset (refer from [36]).

years. Instead, there have been studies on the following approaches supporting extract features from the trajectory dataset. Oates et al. [34] successfully extracted motifs of trajectories from noisy people flow datasets by applying a context-free grammar technique. These studies adopted SAX or TraSAX (an extension of SAX) during compressing and visualizing datasets, as we also adopt. Ohata et al. [35] categorized the migratory behavior of the customer's trajectories in the store by converting them into strings based on the passing areas. Yada [36] analyzed movements of customers in a supermarket, and searched for popular sections. Figure 2.1 is an example of the generation of strings. This technique converted original datasets to sequences of characters that indicate sections where customers moved to. However, important information including times of staying at each section is not preserved after converting. The main objective of the research is to convert the trajectories to the symbol strings in the above studies. Visualizations of the trajectories applying the symbol strings are quite simple but still an open problem.

The followings are methods to further visualize the trajectory data that has been converted into strings. Teknomo et al. [37] analyzed moving patterns of customers at a supermarket. They allocated letters to each intersection in a hypermarket and expressed customers walking routes as strings. Also, they researched populations and length of walking times. Burch et al. [38] proposed using a timeline to visualize eye-tracking scan-path transitions between AOIs (Areas Of Interests). This method is effective to understand in which parts of transitions the differences occurred. In addition to the displaying shapes of the paths, highlighting transitions between AOIs is also popular. Yang et al. [39] proposed Alpscarf that visualizes the order of access among AOIs for scan-paths. This method does not include showing positions of each AOI, therefore users may feel difficult to understand the specific movement of the scan-paths. These methods have the common feature that trajectories get symbolized and the distribution and appearance rate of the symbols that make up those strings are visualized using a matrix. On the other hand, since the visualization of the shapes of trajectories is not performed, it may be difficult to understand the actual behaviors for beginners of visualization. In our proposed technique, the shapes of trajectories and the appearance rate of the symbol are visualized simultaneously by coloring the trajectories or linking multiple visualization screens.

2.1.2 Other Methods to Analyze Behaviors

Next, we introduce other behavior analysis methods using trajectory datasets. Johnson [40] et al. proposed a method for detecting pedestrian movement information from image sequences. Porikli et al. [41] proposed a method of classifying actions by speed and position by applying hidden Markov models and spectral networking to trajectory data. Suzuki et al. [42] proposed a method for detecting pedestrian departure in real-time using the hidden Markov model. Asahara et al. [43] proposed a method to assign labels to pedestrians and classify positioning data. With the proposed method, the positioning data acquired in a low-accuracy positioning environment could be appropriately classified according to pedestrian behavior. Butenuth et al. [44] superimposed pedestrian simulation data on satellite images and classified their actions using a hidden Markov model.

Differently from the above methods, the proposed technique applies only simple methods such as distance calculation and clustering or N-grams to classify actions. These have the advantages that on simpler parameters and ease for users to adjust the parameters according to what criteria the user wants to classify the behavior.

2.2 Visualization of Trajectories

This section introduces various visualization methods for trajectory datasets. We classify and introduce along with the three issues related to the visualization method presented in Chapter 1.1.

2.2.1 Representation of Trajectories

The existing visualization techniques feature variously different representations to draw the flow lines themselves. This section first introduces unique methods for expressing trajectories. Specifically, the introduced methods have features for (1) drawing the shapes of the measured trajectories in detail, (2) representing trajectories grouped together, and (3) displaying a list of features instead of the shape of the trajectories.

First, we introduce visualization methods for traffic trajectories. Approximately, we can classify the methods according to observing whether people or other moving things like cars. Andrienko et al. [45] collected datasets from a wide range using GPS, and analyzed properties of various moving objects, using both drawing specific trajectories and bar charts on maps. The following two studies are observation of moving people using cameras to obtain the datasets. Yabushita et al. [46] proposed a technique that summarizes pedestrians' trajectories recorded at open spaces where definite routes are not constructed. This technique effectively represents major routes of pedestrians; however, it misses some types of important information including temporal tendency and walking speeds. Fukute et al. [47] applied a spectral clustering algorithm to pedestrians' trajectories to classify them to meaningful sets of walking patterns and visualized temporal transition of populations for each cluster by applying a piled polyline chart. Guo et al. [48] developed a composite visualization tool named "TripVista" to analyze patterns of various objects such as bicycles, and cars (See Figure 2.2). They adopted not only specific drawing of trajectories on maps, but also other visualization methods including piled polyline charts, scatterplots, and parallel coordinates plots. Wang et al. [49] improved TripVista and had experiments to analyze

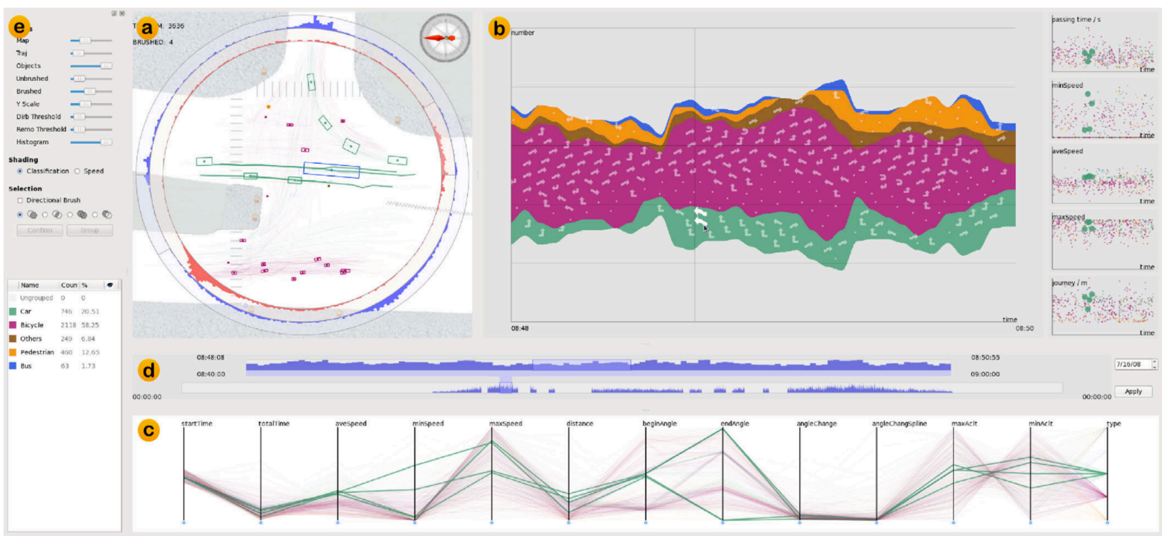


Figure 2.2: View of Trip Vista (refer from [48]).

moving patterns of taxis in Nanjing. They succeeded to find differences of moving patterns between weekdays and weekends. Guo et al. [50] classified walkers' trajectories according to their speed and direction. They also developed a system to visualize important trajectories using meaningful colors based on the HSV model. However, the tool does not support interactive trajectory selection for detail-on-demand visualization. Remark that many studies on the analysis of pedestrians focus on small areas that can be observed with a small number of cameras. Meanwhile, other types of trajectories are also popular targets to analyze. Wang et al. [51] extracted and visualized features of automobiles passing at particular positions, applying datasets collected using many sensors on roads. Al-Dohuki et al. [52] developed a visualization system for trajectories

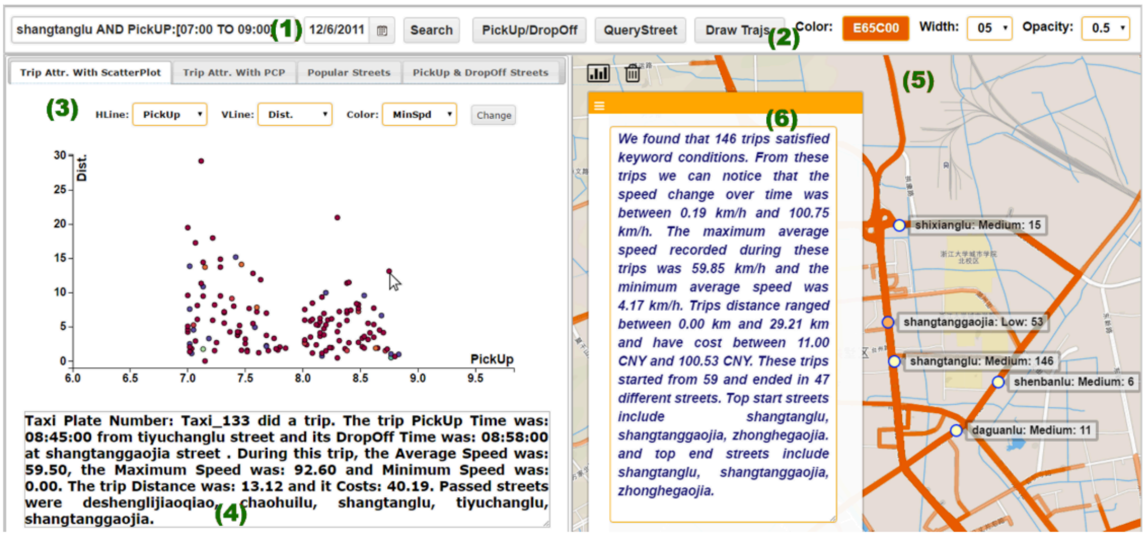


Figure 2.3: GUI of semanticTraj (refer from [52]).

of taxis (See Figure 2.3). The system can visualize not only statistical information about traffic but also messages which answer to questions input by users. For example, uses can select particular streets or time periods, and visualize related datasets.

Next, we introduce visualization techniques that focus on eye-tracking data. Krueger et al. [53] presented an improved visualization system for chronologically GPS datasets. Users can move a circle looked like a lens and focus on particular areas to get detailed information such as speeds and directions as shown in Figure 2.4. These above mentioned studies introduced various visualization techniques; however, such studies did not apply data compression techniques for the trajectory datasets. Krueger et al. [54] proposed using a lens-shaped window to select scan-paths to show. Users can focus on a particular region and search for scan-paths that passed the selected area. Kuebler et al. [55] visualized scan-paths of when participants observed a painting, by two visualization methods, a heat map, and drawing trajectories (See Figure 2.5). They classified scan-paths according to positions and directions by hierarchical clustering. Ramin et al. [56] proposed the Streakline framework. In this method, trajectory dataset

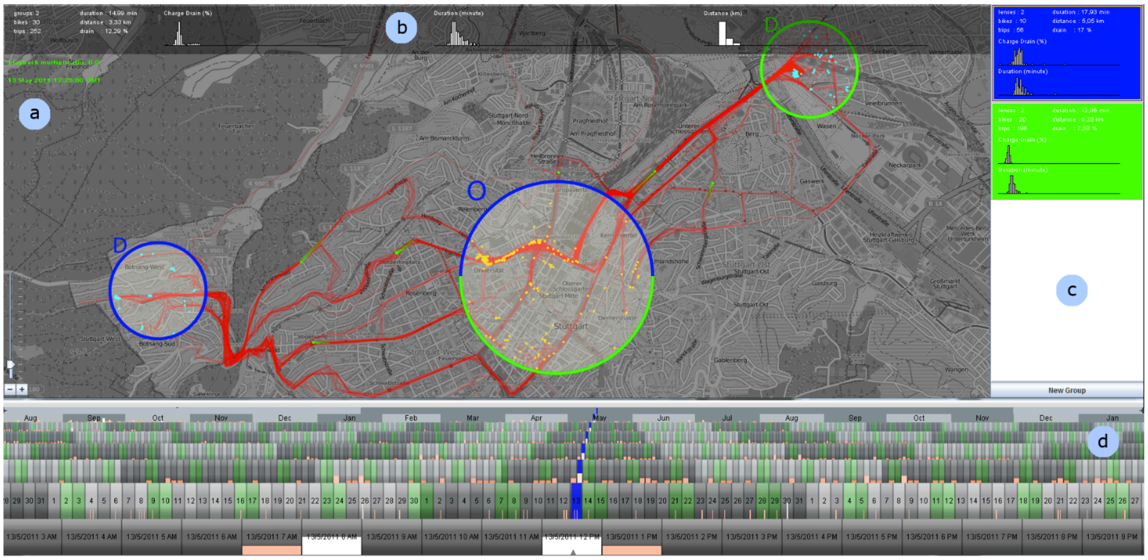
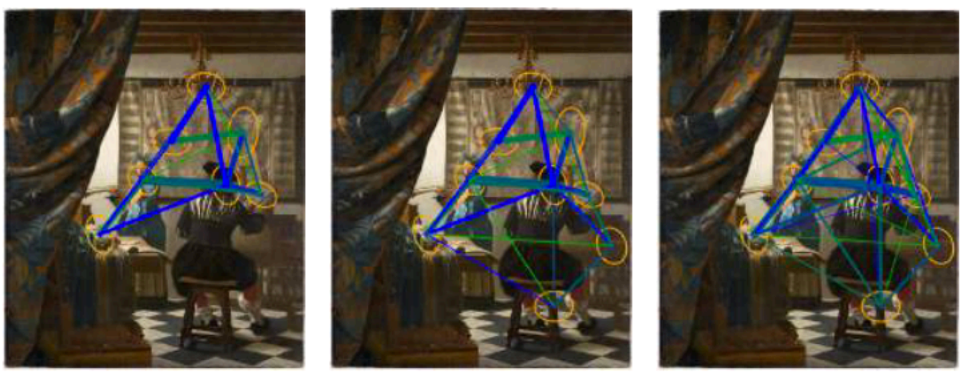


Figure 2.4: Using TrajectoryLenses (refer from [53]).



(a) Direct (b) Low threshold (c) Indirect

Figure 2.5: Eye-tracking scan-paths on the paintings. (refer from [55]).

is treated as streamline data, and flow detection and visualization from moving images are performed. Rodrigues et al. [57] proposed a method to gradually summarize scan-paths. Rudi et al. [58] proposed a combination of drawing trajectories and showing timeline visualization. In such the composite visualization methods, too detailed scan-paths often get visualized and reduces visibility. Users can arrange the selection and summarization of scan-paths flexibly with such methods. On the other hand, avoiding overlapping and pattern extraction are not main focuses of these studies. They succeeded to extract popular behavior; however, they did not focus on finding exceptional movements.

The method proposed in this thesis draws trajectories basically one by one and represents groups of the trajectories by overwriting them. Users can flexibly switch between drawing individual trajectories and visualizing groups of trajectories depending on their selections.

2.2.2 Classification of Data Elements

Next, we introduce methods for drawing pre-classified trajectory data. Methods to classify and visualize trajectories have been studied for a long time as a part of behavior analysis. Okazaki et al. [59] proposed a simulation method of crowd movement in an office building. The feature of this method is that it uses a magnetic model. The classification focused on for example, when the vehicle passed a prescribed route, or whether pedestrians are getting lost the way. Chittaro et al. [60] proposed a method to detect characteristic behavior from human flow data in a virtual environment. Krueger et al. [61] improved another visualization system named TravelDiff. The system applies messages in Twitter instead of datasets of specific trajectories and visualizes the movement patterns and crowded places. They aimed to visualize three types of datasets, for pedestrians, taxis, and airplanes using graphs and heat maps. Gupta et al. [62] worked on the visualization of relationships among a small number of pedestrians. Figure 2.6 shows the main view developed as a part of this technique. They did not visualize particular shapes of trajectories, but used a Gantt chart and visualized places where people stayed. Especially, users can find places where plural people stayed at the same time. Thach et al. [63] converted the spatiotemporal trajectories to strings as shown in Figure 2.7, preserving the distances in the original space, and then divided the trajectories. They succeeded to distinguish four types of trajectories collected under different conditions, by taking into account both positions and movement patterns. Alameda-Pineda et al. [64] analyzed two datasets of movements: participants during a poster session, and actions and communications at a stationary. One of the uniqueness

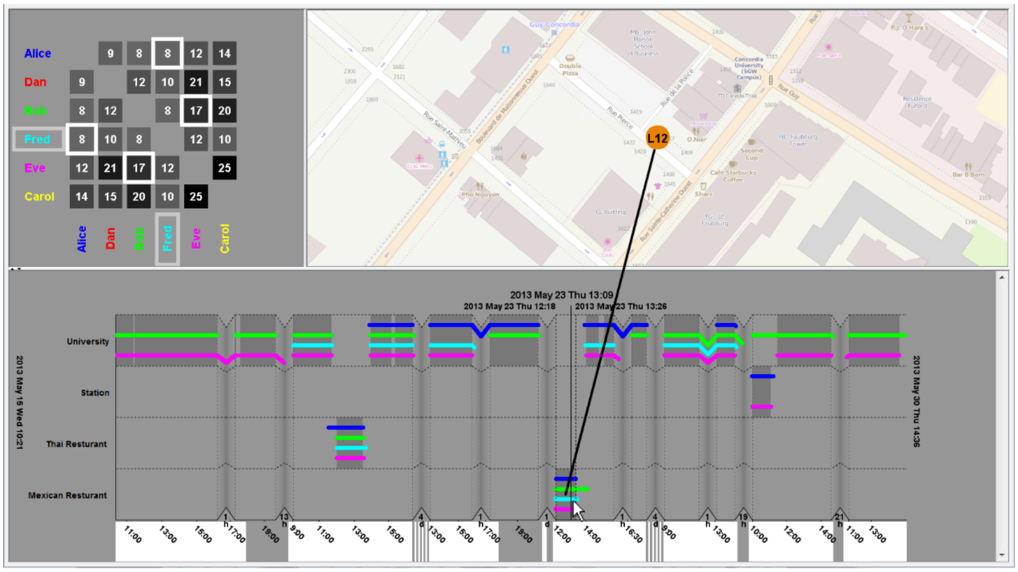


Figure 2.6: Main view of movementSlicer (refer from [62]).

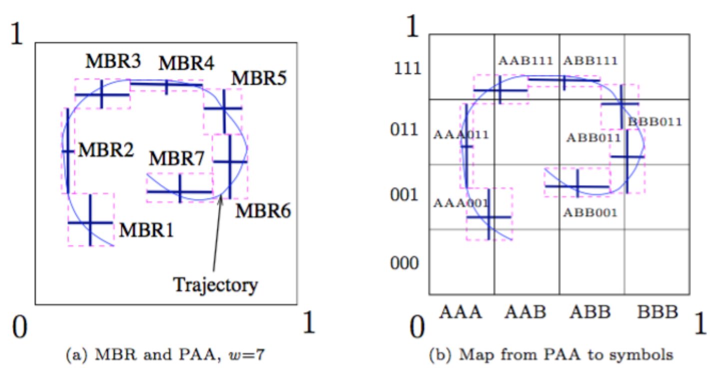


Figure 2.7: Maps for convert a trajectory to strings (refer from [63]).

of this study is that the method realized the comprehensive analysis by applying voice data in addition to the positioning data. The following studies focused on gaze trajectory classification and drawing. Privitera et al. [65] segmented regions such as paintings (Mona Lisa in this case) based on gaze trajectories. Muthumanickam et al. [66] proposed an AOI visualization method that represents how people can categorize places they are looking at in web pages. This study applied a 3D visualization to show represent in AOI over time.

Similar to these methods, our proposed technique applies clustering, etc., and then color the trajectories to represent the classification results. In addition, as described in the previous section, users can flexibly change the contents such as switching to individual trajectory visualization.

2.2.3 Comparison of Trajectories

Finally, we introduce visualization methods whose main target is to compare multiple trajectories. Blascheck et al. [67] used multiple radial transition graphs to show movement between AOIs. Users have to prepare wide display and compare the graphs since one graph indicates one scan-path. Burch et al. [68] analyzed the differences between left and right eye movements using a time-series visualization method named color bands. This method is suitable for analyzing detailed eye movements of individuals; however, it is necessary to compare the same number of graphs to compare data of multiple people. Burch et al. [69] converted scan-paths to strings and visualized their features by a matrix. Gu et al. [70] developed the composite visualization method ETGraph for comparison of scan-paths (See Figure 2.8). Users can display several contents such as trajectories on a stimulus and comparison of gaze times. Peysakhovich et al. [71] visualized differences of scan-paths on a painting using a heat map.

Users cannot extract characteristic parts such as common or different parts in the scan-paths using the above methods. Our proposed technique can narrow down and display the differences or common parts according to the user's operation.

2.3 Conclusion

We have introduced existing research on trajectory analysis in this section. Firstly, we introduced preprocessing of visualization, specifically methods of symbolizing trajectories and that of classifying behaviors in Section 2.1. We also described a visualization method that specifically targets the trajectories that have been symbolized. In Section 2.2, we introduced the visualization methods for trajectories dividing into three categories.

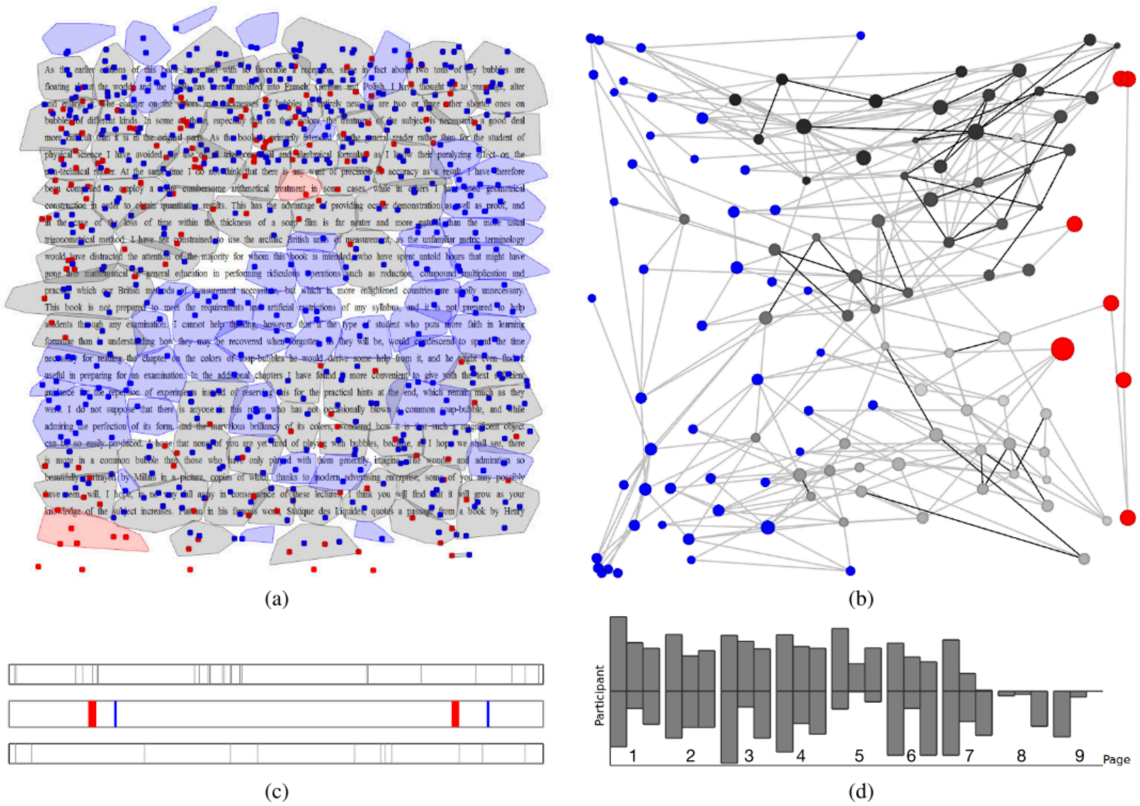


Figure 2.8: A comparison of two participants by ETGraph (refer from [70]).

As a whole, we mainly introduced existing studies that mainly focused on human flow data and gaze trajectory data targeted by our proposed method. General-purpose methods applicable to a wide range of flow line data will also be required while it is necessary to develop a method specialized for each type of data.

Chapter 3

People Flow Visualization

3.1 Introduction

This chapter introduces a method to visualize people flow data. We first compress large trajectory data set by symbolizing and extract features to visualize. Section 3.2 mentions overview of the proposed technique. Section 3.3 contains details of the processing flow and we introduce the results of experiments in Section 3.4. Section 3.5 discusses results of the experiments, and finally, Section 3.6 summarize this chapter.

3.2 Overview

In recent years, security cameras improved and set at various places, such as stations, shops, residential area. These cameras record many pictures and movies of pedestrians every day. Using these datasets, we can understand features of movings, such as "Where pedestrians tend to walk?," "Which place is most crowded?."

Such information is very useful to solve various social problems, and develop our life. For example, most popular problem is traffic control. Finding crowded areas and suggesting solutions to prevent unwanted congestions are simple but significant problem in our society. In the field of disaster prevention, analysis or simulation of evacuation routes have been getting popular research theme since outbreak of the Great East Japan Earthquake in 2011. For exmaple, Onishi [72] observed movings of audiences in evacuation drill at a concert hall. They found that several people selected wrong routes since the people just followed leading people without any idea to try to pass correct evacuation routes. Furthermore, analyzing movings of customers in various shops is also important problem. Mainly, salesclerks can know which products attract many people by knowing actions of customers in the shops, and suggest better product display. Like this, recording moving routes of people using cameras, and analysis of moving patterns should be useful.

However, such cameras work for night and day, and exist at many places. Following two problems are still open problem in analysis of people flow data.

- People flow datasets easily accumulate and they are difficult to manage.
- Observers can not find characteristic data elements or trends from the datasets efficiently.

Nowadays we can record huge datasets since hard disks improved and inexpensive. However, solely saved datasets are not suitable to efficient management and analysis, therefore we should improve methods to reduce sizes of the dataset. Furthermore, finding important data elements is not an easy task. We have to develop methods to find features and trends of people data, then present the information with simple expression which is easy enough for general people.

Then, we focused on the two problems and developed visualization system to analyze large-scaled datasets. We compress people flow datasets recorded by cameras into sequences of characters, then visualize the features of the datasets. Concerning compression, the process reduces sizes of dataset preserving essential way points and staying times. One of characteristic points is that the process discards too fine information about positions of persons. For example, if users want to understand about "Which booth is most crowded in an exhibition?," distinction of differences of only one step is not essential. The system reduces such excess information about positions of persons to manage the datasets efficiently.

Furthermore, the system generates pictures that allow users to understand trends of datasets recorded for several hours in a short time. Users can know particular walking routes by looking graphs with colored nodes and edges that reflect map of locations. Such visualization is useful to understand overview of people flow datasets. Also, users can find several areas which is worth analyzing like especially crowded places or main walking routes. Then users can check detailed information on the area. In other words, users do not have to compare too much information on various places.

This chapter presents a technique to analyze large-scale people flow datasets efficiently. First, record trajectories of waking people using RGB-D camera Xtion. The datasets include ID, positions in 2D space, and times. Then we convert the position data as real values to simple strings applying UniversalSAX[73]. UniversalSAX is an expansion of SAX (Symbolic Aggregate approXimation) which can convert chronologically datasets to sequence of simple characters. This operation means compression of the datasets and transforming so as to make extraction of features easy. Furthermore, we convert the strings to Run-Length codes which can reduce length of the strings preserving original information. Run-Length code can directory express way points and staying times at each region. Using the Run-Length codes, we classify trajectories by applying clustering. Finally, we visualize walking patterns of people using a graph. This chapter also shows a case study with a real-world people flow dataset in an exhibition, then discusses effectiveness of the presented tool.

3.3 Implementation

This section presents the processing flow shown as figure 3.1 and detailed description of technical components of the proposed technique. Section 3.3.1 defines the people flow datasets. Section 3.3.2 describes a technique to convert the trajectories into sets of characters. Then section 3.3.3 introduces preparation before extraction of features, following by the discovery of typical movement features described in Section 3.3.4. Section 3.3.5 shows a method to classify walking routes. Lastly, we describe a visualization technique which emphatically displays the features in Section 3.3.6.

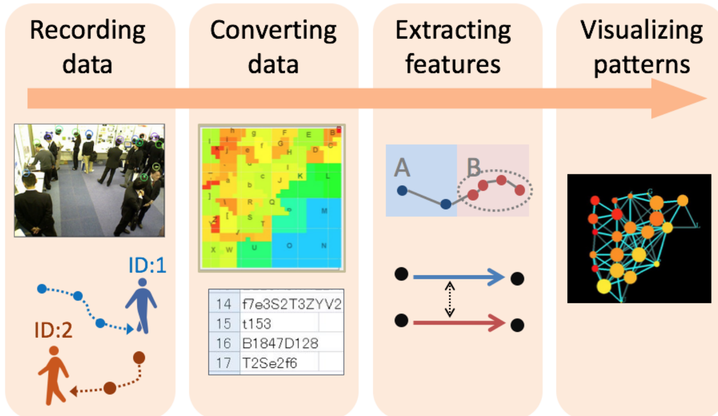


Figure 3.1: Processing flow of the proposed technique.

3.3.1 Recording of People Flow Data

We define that a record of people flow data includes the following information:

- Time that the position of a walker is measured
- ID of the walker
- Position of the walker in a 2D space (x, y)

We can construct a trajectory of this walker by collecting the records which have the particular ID corresponding to the walker, and then chronologically ordering the collected records.

Our current implementation uses a RGB-D camera Xtion to record the people flow data applying a technique in [74]. Xtion can assign a particular ID to each walker, and measure positions of heads of pedestrians every dozens of milliseconds. In this time, Xtion can measure positions of pedestrians in a real three dimensional space; however, we adopted only two dimensional coordinates on floors and regard walkers' heights as constant. One camera can capture an area of sabout 5 meters square, and also can to record a wider range of pedestrian traffic lines by combining the recorded contents of multiple cameras [75].

3.3.2 Conversion of People Flow Data to Sequences of Characters

In the next process, we generate sequences of characters from position values in the people flow datasets, in order to reduce the data sizes and make it easier to extract movement features.

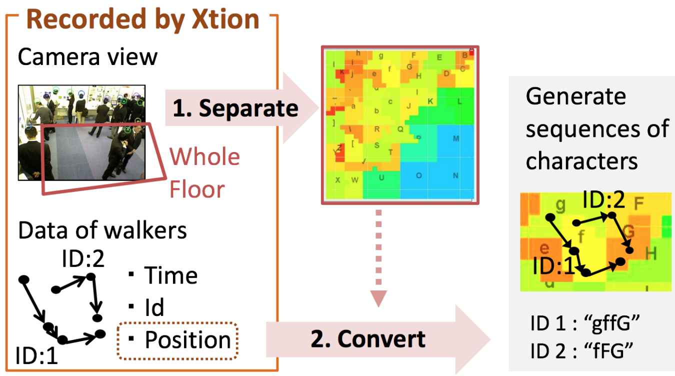


Figure 3.2: Flow of converting position values.

Figure 3.2 illustrates this process. We apply UniversalSAX [73], an extended implementation of SAX (Symbolic Aggregate approXimation) which converts time series data recorded as real values to sequences of characters. There have been several other techniques on extended implementation of SAX to deal with multidimensional real values; UniversalSAX has advantages against other techniques on preservation of numeric features of all dimensions and distances among data items.

The following briefly describes the processing flow of the setup phase of UniversalSAX:

1. Divide multidimensional space to multiple regions, and generate a distance table among the regions.

2. Allocate a particular character to each of regions so that we can convert positions described as real values to characters.

Users can adjust the resolution of space division with the following four parameters:

d : dimension of data

2^q : number of partitions in each axis

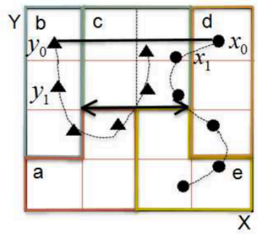
z : number of characters

2^b : threshold value to divide large regions

UniversalSAX assigns characters to the regions by applying a Hilbert curve, a kind of space-filling curves. In this process, we firstly separate a d-dimensional space (d is 2 in this study) to lattices where each axis is divided into 2^q segments. The shapes of the regions generated thereafter are determined along the Hilbert curve passing through this grid. Therefore, increasing the resolution tends to complicate the shapes of the generated areas. This process can lead to producing the uniform distribution or finely reproducing the distribution of the input data. However, the comprehensibility of the divided regions may be worse because of their complex shapes. Therefore, users have to take care not to divide too finely. Then, we generate a Hilbert curve that passes every block once. A sequential number that indicates the order in which the Hilbert curve passes is assigned to each block. Unlike other space-filling curves such as Z-ordering and Peano curve, the Hilbert curve always passes an adjoining block right after passed one block. Pairs of neighbor blocks are assigned closer numbers, while apart blocks are assigned entirely different ones. Therefore, the one dimensional block numbers represent original coordinates in a multi-dimensional space, preserving correlations of distances among data items. Next, we generate z regions by grouping these blocks and assign characters alphabetically to each region in the order of the sequential numbers of the blocks. The number of blocks belonging to one region depends on the distribution of data items. A smaller number of blocks are assigned to regions that many data items belong. On the contrary, a larger number of blocks are assigned to a region if few data items belong to. The following are two reasons to adjust areas of regions according to the density of data items, not regularly separating the space in a grid pattern. One is to keep a fine resolution to preserve more accurate spatial information in high density regions such as crowded places. The crowded place is usually worth paying attention to understand the walking patterns of people. The other is to avoid a lack of characters to assign as names of regions by too fine separation at sparse regions. The number of regions z needs to be adjusted according to the environment in which the multidimensional data is acquired and what behaviors the user needs to analyze. For example, in the case of pedestrian flow line data, it is conceivable to first set a numerical value that can reproduce the area division that can be assumed in advance (such as the division of the store in the store). If users do not have enough knowledge to determine these or apply the datasets of the places where the degree of freedom is high, the users have to determine how finely pedestrian movements need to be analyzed. When z is small and the area of one area is large, finding the characteristic of movements of the pedestrian groups is possible. On the contrary, if z is large and the area of the area is small, a comparison of individual pedestrians gets easy. Furthermore, users can set a threshold of 2^b and divide a large area. This process prevents coordinates that are too far away from being included in the same area and being considered as the same point. Similar to the number of regions, the threshold 2^b needs to be determined according to the data acquisition environment and what kind of behavior needs to be analyzed.

At the same time, this process saves a table of distances among all regions conveniently. The distance between the regions is the same as the shortest route connecting the regions, as in the processing in SAX. For example, when calculating the distance between the area A and the area B, the Euclidean distance between the squares is calculated for each of "all the cells that constitute the area A" and "all the cells that constitute the area B," and these distances are calculated. Then the minimum value of these distances becomes the final distance between regions A and B. As described above, since the shape of these regions is determined by the Hilbert curve, the cells that form one region are densely gathered. Therefore, the distance between regions strongly maintains the distance relationship before the transformation. When a space-filling curve other than the Hilbert curve is used, it is possible that "an elongated area is generated" or "the masses that form one area are discrete."

Therefore, the distance between regions may not be intuitive. The method of adopting



	a	b	c	d	e
a	0	0	0	1	0
b	0	0	0	2	1
c	0	0	0	0	0
d	1	2	0	0	0
e	0	1	0	0	0

Figure 3.3: Illustration of the distance between original data and distance after character string conversion(refer from [73]).

the shortest distance between masses has already been adopted in SAX as the base of UniversalSAX. Figure 3.3 illustrates an example of distance definition. While applying to the distance calculation of the trajectory data, it means that "ignoring the size of the region itself and the moving distance inside the region and considering only the distance outside the region." When the segmentation by UniversalSAX is appropriate, "Moving within a single area" means an action that is judged by the user to be less important than "moving between different areas" or need not be considered. Such processing is considered to be effective for calculating the distance according to the target of the users' analysis. As a specific example, let us suppose a case where each poster is divided into one area for "analyzing how far a poster session participant walked and how many posters the participants passed." In this case, between which region (poster) the movement has occurred is particularly important. On the other hand, positions and movements within a single area (e.g. whether you are directly in front of or present beside the presenter) are determined to be of low importance. Here, only the moving distance corresponding to the target "moving to a different poster" can be calculated by calculating the distance using the shortest distance between the cells. If another distance (e.g. between the center of the region or between distant cells) is used instead of the shortest distance, the obtained distance contains as both "outside the region" and "inside the region." In particular, while calculating the distance between regions with large areas, it is necessary to take into account that the influence of the latter will increase. As described above, redundant movement information can be reduced by using the shortest distance. On the other hand, if parameter settings of the Universal SAX are not appropriate and points that should be in different areas are included in the same area, the movement information to be taken into account will be reduced.

We can refer the table while calculating distances among walking routes. This lookup-table-based implementation reduces the computation time of this process.

After completing the setup process described above, we can convert people flow datasets to sequences of characters which form much smaller datasets. In this process, we firstly apply an Affin transformation to positions at each time step of the trajectories to complete the calibration. Then, we generate sequences of characters from pedestrians' walking routes with the regions divided by the setup process. We define three kinds of datasets, G consisting names of regions, S_k which represents single walking route, and P that is a collections of all of walking routes as:

$$G = \{g_1 \; g_2 \; ... \; g_z\} \tag{3.1}$$

$$S_k = \{s_1, \; s_2, \; ... \; , \; s_l \; | \; s_j \; \in \; G\} \tag{3.2}$$

$$P = \{S_1 \; S_2, \; ... \; S_p\} \tag{3.3}$$

g_i ($i \; = 1, \; 2, \; ... \; , z$) is characters as names of regions, selected from 'A'-'Z', 'a'-'z', and some symbols such as '\', '[', and ']' in this order. The process chooses l letters as s_j from the names of regions permitting duplication, and generates S_k ($k \; = 1, \; 2, \; ... \; , p$) using s_j which are chronologically ordered.

3.3.3 Preparation of Calculation of Distances

The following processing is added to the strings generated by UniversalSAX in the previous section. This process can convert the strings into a form that more clearly shows the characteristics of the trajectories.

Run-Length encoding

The sequence of characters usually construct much smaller datasets comparing with the original people flow datasets.

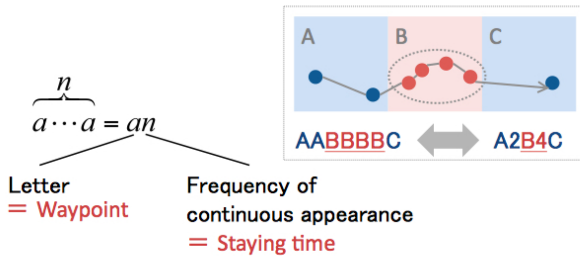


Figure 3.4: Applying run length encoding to a walking route.

To compress the datasets further, we apply the Run-Length encoding to the sequences of characters, as shown in Figure 3.4. From each S_k, we generate p Run-Length codes defined as S'_k. Run-length encoding is a reversible compression method which is especially effective for sequences that the same letter continuously appears; it corresponds to a situation that a walker stays for a while in one region. Contribution of run-length encoding is not only the compression of the sequences of characters; it also assists the discovery of places which walkers stays for a while, by searching for continuously appearing characters.

We extract features in the sequences of characters to understand property of people flow datasets. One of the features is where people stop walking. We can discover congestions and their average staying time, by searching for the g_i which continuously appear in S_k. Furthermore, we calculate distances among the sequences of characters to summarize or search for particular walking routes.

Correction of Abnormally Long Strings

Before calculating distances, we correct abnormally long strings generated by the people flow conversion process described in Section 3.3.2 to calculate appropriate distances. (See Figure 3.5.) Usually, Run-Length codes generated from movements covering wide

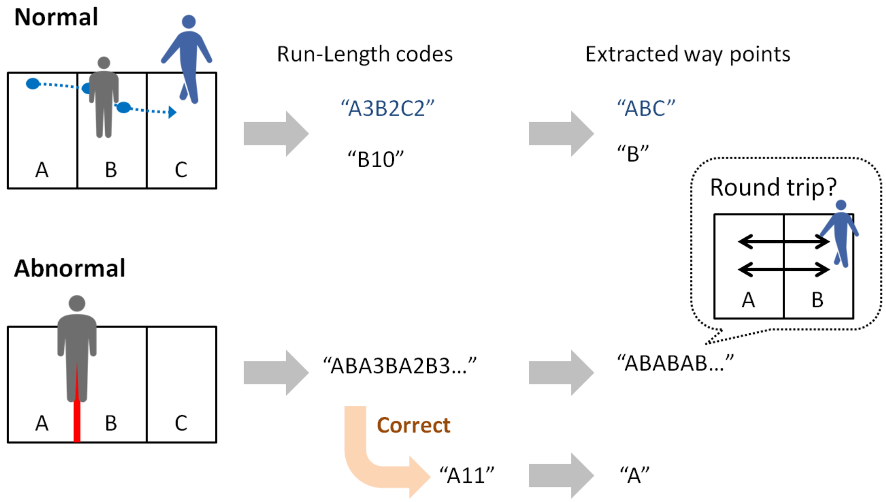


Figure 3.5: Correction of abnormal strings.

places get longer than other ones from movements staying at limited regions. However, if a walker stopped on a border of two regions 'A' and 'B', positions recorded at regular time intervals may be occasionally judged as 'A' and converted to 'B' at other times, affected by noises or other factors. A generated Run-length code have alternately appearing 'A' and 'B' with small numbers which means short staying times while converting such positions. In other words, the code means round trips between two regions 'A' and 'B', even though the walker actually stopped. Focusing on camera view in this study, whole view is about five square meters, and approximate area of each region is from tens of square centimeters to one square meter, so actually it does not often happen that walkers have such round trips many times. To solve this problem, we arrange such Run-length codes. Specifically, we convert reputations of two letters and staying times at each region, to one of the letters and staying time there. We select one letter which has longer staying time than the other, and set new staying times by accumulating staying times at each region. In this implement, we correct parts which have reputations of patterns "ABA" of way points (like "ABABA"). We do not adjust only one appearing (like "ABA"), since it is likely a real turning.

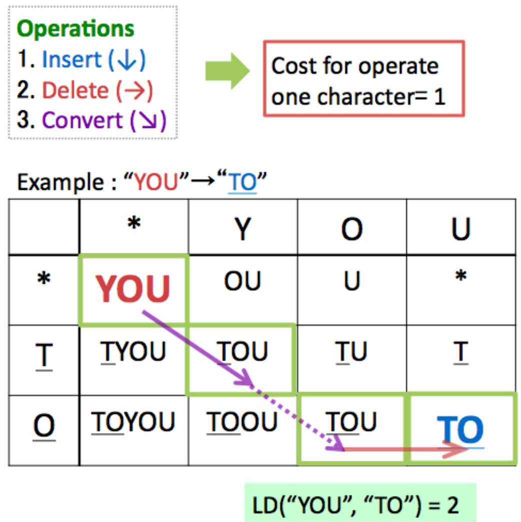


Figure 3.6: An example of calculating original Levenshtein Distance.

3.3.4 Calculation of Distances among Walking Routes

After this process, we calculate distances among S'_k. We adopt Levenshtein Distance (LD), one of popular definitions of distances between sequences of characters. We can calculate LD even S'_1 and S'_2 have different lengths. If length of S'_1 is m_1, and that of S'_2 is m_2, time complexity to calculate LD between them is $O(m_1m_2)$. This method determines how many operations (insert, delete or replace one character) are required to convert S'_1 to S'_2. The number of operations corresponds to the distance between sequences S'_1 and S'_2. We regard S'_1 as closer to S'_2, if the less number of operation is required to convert S'_1 to S'_2. Specifically, this method calculates sums of conversion costs, while supposing costs of any operations as 1. We can determine the smallest cost to convert S'_1 to S'_2 applying dynamic programming. (See Figure 3.6.)

Here, original definition of LD causes a problem in our study. Degrees of a difference between two characters are ignored in the original definition since all costs are uniformly defined as 1. For this reason, we cannot calculate degrees of differences between any two letters, and express differences among walking routes accurately. Suppose three region 'A', 'B', and 'C', and a walking route"ABC", and compare this walking route "ABC" to other walking routes "ADC" and "AZC", by calculating differences using LD. If region

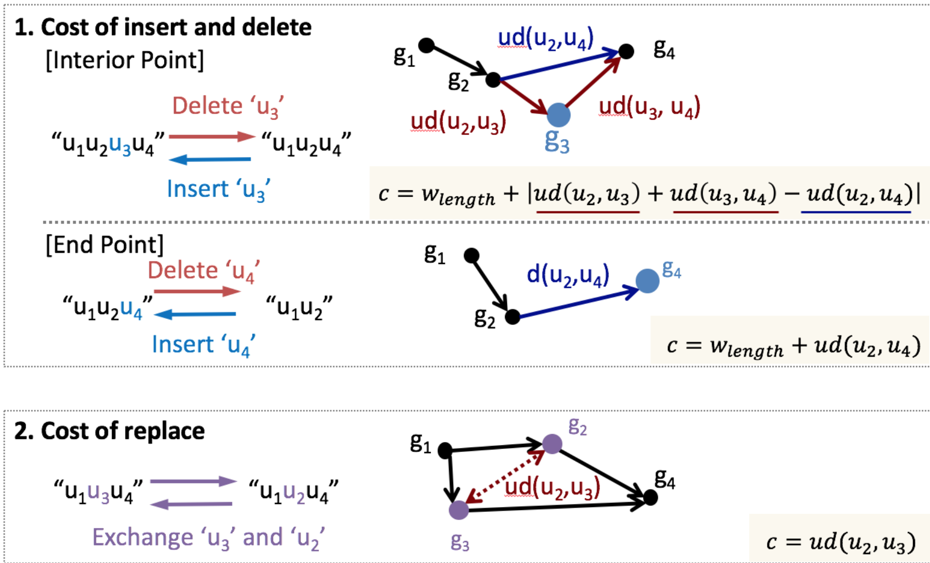


Figure 3.7: Costs of each operation.

‘D’ is close to ‘B’, and region ‘Z’ is far from ‘B’, we should set a distance between “ABC” and “AZC” larger than that of “ABC” and “ADC”. However, in practice LD(“ABC”, “ADC”) and LD(“ABC”, “AZC”) get the same value 1, and no differences between the distances are produced. To solve this problem, we apply a weighted LD (WLD) [76] which can reflect differences between arbitrary pairs of characters by flexibly adjusting costs of insert, delete and replace operations. Furthermore, we regard a pair of one name of region and staying time there as one unit $u_{j'}$ ($j' = 1, 2, \dots, l'$) and calculate sum of cost of operating one unit. We define $S'_k = \{u_1, u_2, \dots, u_{l'}\}$, where $t_{j'}$ denotes how many times $g_{j'}$ continuously appeared:

$$u'_{j'} = \begin{cases} g_{j'} \ (t_{j'} = 1) \\ g_{j'} t_{j'} \ (otherwise) \end{cases} \tag{3.4}$$

We define costs of each operations as shown in Figure 3.7 using these definitions. $ud(u_{j'_1}, u_{j'_2})$ denotes distances between two units described as:

$$ud(u_{j'_1}, u_{j'_2}) = d(g_{j'_1}, g_{j'_2}) + w_{time}\sqrt{|t_{j'_1} - t_{j'_2}|} \tag{3.5}$$

The first term denotes distances between two regions, and the second one means staying times at each region. We simply refer a lookup-table generated by the setup process described in Section 3.3.2, to identify a distance between two regions $g_{j'_1}$ and $g_{j'_2}$ as $d(g_{j'_1}, g_{j'_2})$. w_{time} is weight of influence of staying times. If w_{time} is zero, the system distinguish walking routes according to only shapes or way points. We define same formulas for inserting and deleting operations since they are just opposite each other. When applying inserting or deleting, selection of a formula depends on whether the modified unit is at the top or the last of S'_k, or not.

We can change values of costs by not only which type of operation is applied but also to which letter we apply operations, by using $d(g_{j'_1}, g_{j'_2})$. The more distance between $g_{j'_1}$ and $g_{j'_2}$, the more $c(g_{j'_1}, g_{j'_2})$. If two S'_k include different letters, we can express not only information that way points are not same, but also degrees of differences between $g_{j'_1}$ and $g_{j'_2}$. At this moment, we add a positive real number w_{length} to costs of inserting and deleting, so as to make these costs larger than cost of converting. When we calculate LD between two S'_k that have different lengths, insert and delete tend to be required operations. Hence, adding w_{length} causes that LD between S'_k that have different lengths get larger. Length of S'_k denotes how many regions does the walker pass, and get a different value whether the walker moved wide area or not. We can classify these movements easily using u. It is possible to define an operation to swap the order of a pair of characters as one operation. We did not implement it as an individual operation because a sequence and its reverse cannot be treated as the same meaning; such sequences correspond to opposite directions of walking routes in our study.

In summary, LD calculation in our study takes into account the following features:

- Places where walkers passed
- Wideness of the walking routes
- Directions
- Staying times

Our method includes the above distance calculation method. On the other hand, there is room to use other distance calculation depending on the analysis target. For example, Jaro-Winkler distance is also a representative distance for strings similar to edit distance. The Jaro-Winkler distance is a method of adding the number of symbols that co-occur in two strings as a score, and the high score means high similarity. The main difference from the edit distance is that the more characters that match in the front part of the string, the higher the score. Therefore, it can be used effectively in the following cases.

- When users want to focus on the movement at the initial stage and analyze it (for example, finding the product that shoppers want to buy first)
- If users want to check that the prescribed behavior is being followed at the initial stage of the behavior
- When the starting point of the trajectories tends to be aligned to some extent

Conversely, in such cases such as "there is no prescribed route and the degree of freedom of movement is high," "the starting point of the trajectory is dispersed," and "the order in which the traffic lines pass through each point is not important"there may be little reason to introduce it [2].

Using results of the calculations, we can classify trajectories or search particular walking routes and submit patterns of walking to observers. Users can select a method to display the patterns. For example, if they want to count numbers of persons who passed specified routes, they can input the route as search term and find corresponding trajectories. On the other hand, when using people flow data recorded at places which do not have designated walking routes, users can hardly understand existing patterns of movements. Then we apply clustering to dataset of strings, and divide them to some groups automatically.

3.3.5 Classification of Trajectories

We select K-medoids as method of clustering. This is one of non-hierarchical clustering methods which is suitable for our large people flow datasets. The process is similar to the major non-hierarchical clustering algorithm K-means. But K-medoids applies not centroids but medoids selected from existing elements. That means we have to calculate only distances among elements. Instead of describing walking routes by vectors that have same length, we calculate distance matrix among elements as mentioned above, and apply K-medoids clustering.

In K-medoids, users can set the number of clusters freely, however it is sometimes difficult to select the optimal value. In that case, we apply the following way to determine the number of clusters.

1. Set the number of clusters n to 2.

2. Run K-medoids and generate n clusters.

3. Finish the process when the clustering result satisfies the following formula 3.4.2. If the following expression is not satisfied, increase n by 1 and return to step 2.

$$0.3 < \frac{c}{n} \tag{3.6}$$

c is the number of clusters whose number of elements is equal to or less than the threshold value m. By setting such a criterion, both of a cluster containing many elements and a cluster containing only few elements are generated. Clusters with many elements are considered to contain typical behaviors, and clusters with only few elements are considered to include exceptional behaviors and characteristic actions. These are useful information for analysis regardless of what kind of trajectory dataset is used.

3.3.6 Visualization of Walking Routes

Finally, we visualize people flow data as sequences of characters emphasizing its features. We generate nodes at the centers of the regions divided by the setup process described in Section 3.3.2. This process displays the connections of the nodes based on the sequences of characters to represent the walking routes. We suppose two types of operations with this representation. One is to represent abstract information of the data with the overview, and the other is to select particular regions interactively and then display detailed information at the selected regions. We suppose that users can find multiple regions by selectively displaying the following:

- Regions which many walkers densely stopped
- Number of walkers moving across a particular pair of regions

We visualize the congestion of walkers at each region by drawing circles at the nodes. Their radii depict the numbers of walkers that passed the corresponding regions, while their colors depict the average staying times of the walkers at each region. Warmer colors are assigned to the circles when the average staying times are longer. We also represent the populations of walkers moving across the pairs of regions by widths of segments connecting the corresponding pairs or nodes.

Using the functions mentioned above, users can narrow down several regions that worth exploring finely down. They can select particular regions and observe detailed information related to the regions using the following:

- Animation of walking routes
- Search function for walkers who passed a particular route

Users can observe animations that show particular walking routes. Our implementation draws segments connecting pairs of regions increasingly in the temporal order. Users can select segments to be drawn by selecting a node and a radio button. Specifically, our implementation allows to interactively select a particular region to draw segments representing walkers leaving or coming to. We adjust densities in segments reflecting the number of the segments. Our implementation separately draws each segment to directly compare them, as shown in Figure 3.8(left), if there are small number of segments to display. Otherwise, our implementation bundles similar segments as they look like a single thick segment, as shown in Figure 3.8(right). This representation avoids too complex visualization results caused by huge number of visible segments.

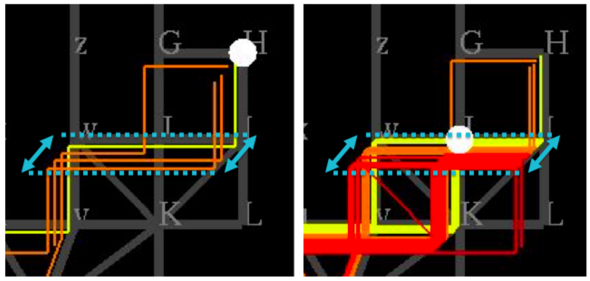


Figure 3.8: Visualizations of small number of segments (left) and bundles of similar segments(right).

Colors of the segments indicate the time when walkers passed the selected region: earlier walking routes are drawn in warmer colors, while later routes are in cooler colors. Users can understand various features of the people flow, such as where walkers passed continuously, and which time the regions are most crowded, by observing the visualization results.

Our implementation also provides a user interface for search operations to specify the segments to be drawn. When a user enters a keyword associated to a particular walking route, our implementation calculates distances between the specified walking routes and others. Then, only walking routes similar to the specified walking route are displayed. The presented technique can assist the understanding of the peculiarity of walkers' actions sufficiently and quickly, by visualizing people flow data emphasizing the tendency of the walkers.

The following two methods are color schemes for trajectory visualization.

Color setting 1 Expressing clustering results in detail

Color setting 2 Expressing the appearance rate of each behavior

The first one is for expressing the clustering results in detail. Our implementation also features visualization of clustering results. Users can choose all clusters or one of them to be displayed. Our implementation specifies colors of clusters based on the following rule using the HSB model. (See Figure 3.9.)

$$Hue = \frac{300\ v}{(V-1)} \tag{3.7}$$

$$Saturation = s_{min}\ +\ w_{sat}\frac{t_{gv}}{p_v} \tag{3.8}$$

$$Brightness = b_{min}\ + w_{bright}\ (255\ -\ b_{min})\ \frac{p_v}{p} \tag{3.9}$$

V is the total number of clusters, and v is the number of selected clusters to be visualized. Particular hues are assigned to each of the clusters. Clusters which have smaller numbers get redder, while clusters which are assigned larger numbers get bluer.

Saturation is also specified for each region. s_{min} is a natural number to avoid S gets too low and therefore it gets difficult to visually distinguish hues of clusters.w_{sat} arranges ranges of saturation. t_{gv}/p_v is the average staying time of people who passed the region g and belonged to the cluster v. Vivid colors are assigned to places that many people tend to stop.

Brightness denotes populations of each cluster in our implementation. Here, we would like to discover typical moving patterns than exceptional ones in this study. Thus, we color clusters that have many trajectories lighter in order to make the clusters conspicuous. If the clusters have larger brightness, it is easier to visually recognize changes of colors while increasing or decreasing their saturation. Therefore, users can recognize differences of average staying times at each region. b_{min} is a natural number to avoid assimilation of trajectories and black background. w_{bright} sets amount of changes when p_v changes.

We also draw yellow hexagons on departure points of trajectories, and white rhombuses on their arrival points to depict moving directions. Sizes of the figures reflect numbers of passed trajectories there.

Next, we introduce the second color setting. This is the use of colors to express the appearance of actions included in each trajectory. The appearance rate of each cluster

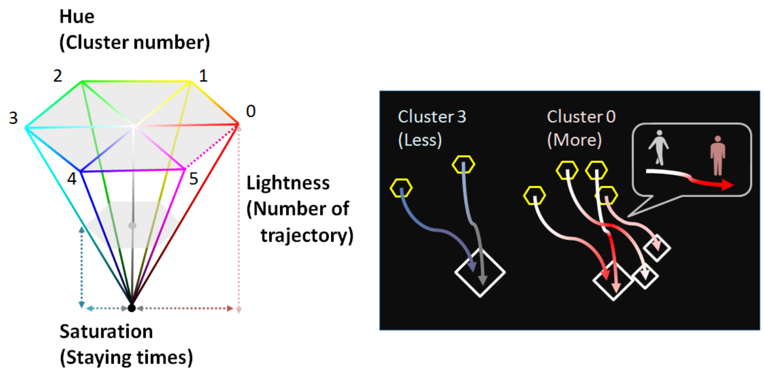


Figure 3.9: Selecting colors of trajectories.

p is represented by changing only the saturation of the flow line. Hue and lightness are set to constant values.

$$Saturation = 255 - (1 - p) \tag{3.10}$$

The appearance rate p of the cluster c_i is the product of the following values.

- Number of words belonging to c_i / Total number of words
- Number of trajectories containing words belonging to c_i / Total number of trajectories

A word representing a certain action is calculated based on "how many times the word appears in the entire trajectory data" and "in how many trajectories do the word appear." We can visualize which part of a certain trajectory is a typical behavior and which part is an exceptional action by this color scheme.

3.4 Experiments

This section presents experiments of classification and visualization using people flow dataset recorded at an exhibition. Broadly, we introduce the results of the following two data sets.

- Visitors of an exhibition.
- Attendees of a poster session.

3.4.1 Case of an Exhibition

Figure 3.10 shows a layout of the ground. We set three cameras at the entrance, a corridor, and an exhibition room, then recorded trajectories of visitors.

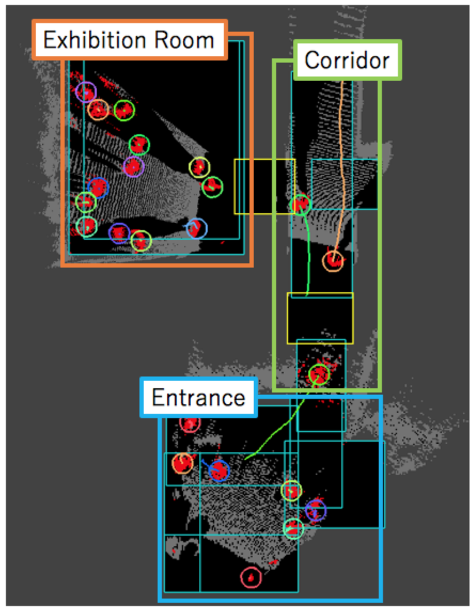


Figure 3.10: Maps of the exhibition.

Exhibition Room

We visualized people flow data recorded in an exhibition room. There were two doorways in the room. One lied at the lower edge in the recorded picture, while the other was at the upper right corner. There were exhibits in the left and upper sides, as shown in Figure 3.11. The dataset contained 5531 trajectories recorded for 8 hours (9:00 to 17:00). We divided the whole floor in the recorded view to 44 regions by UniversalSAX, and assigned characters ‘A’-‘Z’, ‘\’, ‘]’, ‘^’, ‘_’, ‘`’, ‘a’-‘l’ to the regions. We firstly visualized congestions to briefly understand the tendency of the people flow as shown in Figure 3.12, and then searched for where many people passed.

There were many large circles painted in yellow or orange, from lower-left to upper-right regions, in the visualization result. It depicted the regions that were on a walking route where many people passed smoothly. On the other hand, circles at upper-left

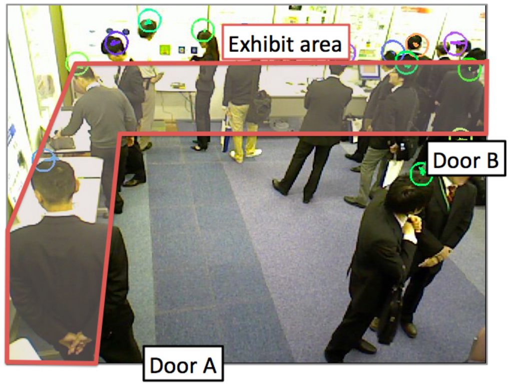


Figure 3.11: Camera view in the exhibition room.

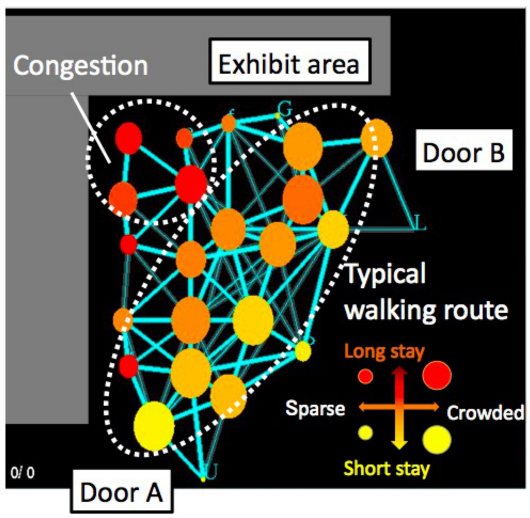


Figure 3.12: Visualization of congestions.

regions got red, which suggested certain number of walkers stopped in front of exhibit there to look carefully. Especially, circles at the upper-left corner were larger than others, which depicted exhibits on the circles got attention of the participants.

Next, we visualized particular walking routes and compared populations in each time periods, as shown in Figure 3.13. We colored trajectories based on HSB model, where larger hue values mean that walkers appeared in a later time period. Each of colors correspond to particular hour; for example, segments drawn in red depict walkers' movement during 9:00 to 10:00. We could find various colors from red to purple near the exhibits, which illustrates the exhibit attracted people constantly. In particular,

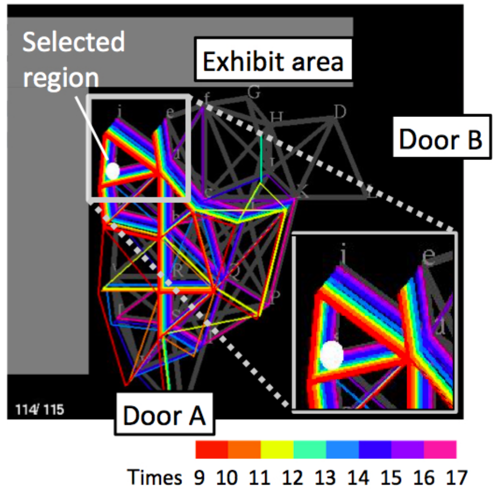


Figure 3.13: Visualizing walking routes passed the selected region.

there were large number of dark blue segments corresponding to participants visited during 14:00 to 15:00, which illustrates larger number of participants visited the exhibits around midday compared to other times.

Then, we selected more specific walking routes as sequential characters, and counted numbers of people who passed the routes by the search function. Figure 3.14 shows eight types of routes that we searched for. We focused on two doors and the exhibition at the upper left corner, and specified these routes based on their directions and way points. There were larger number of walkers (corresponding to routes 1, 3, and 5 in Figure 3.14) came from door A, while smaller number of other walkers (corresponding to routes 2, 4, and 6 in Figure 3.14) came from the opposite direction. Totally, many people entered the room from door A. As above, exhibits constantly had walkers' attention enough to make them stop walking. On the other hand, several walkers (corresponding to routes 5 and 6 in Figure 3.14) straightly moved from a door to the other, not passing in front of the exhibits. Moreover, small number of walkers (corresponding to routes 7 and 8 in Figure 3.14) turned buck after moving to the upper left exhibition. These indicate that we ought to devise arrangement of the exhibition to attract more visitors.

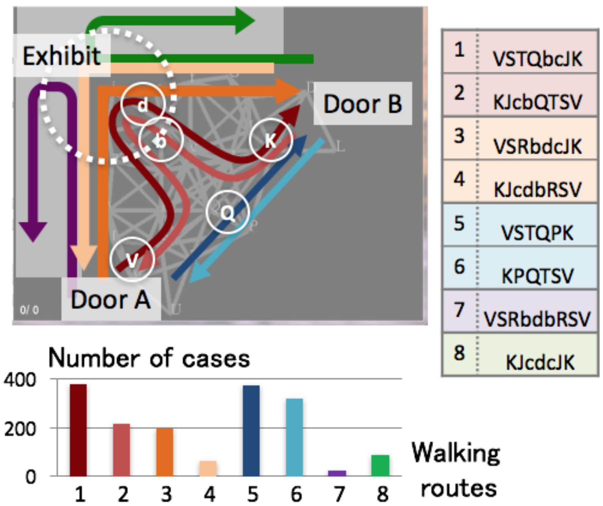


Figure 3.14: Result of counting numbers of people.

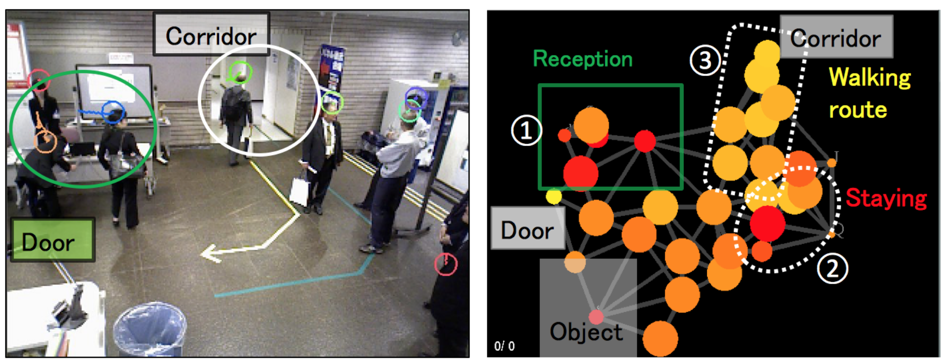


Figure 3.15: Camera view at the entrance(left) and visualization of congestions there(right).

Entrance

This subsection introduces cases of using datasets recorded near reception of the exhibition. First dataset is 8160 trajectories recorded near a reception for 8 hours (Figure 3.15(left)). The data decreased from 138MB to 151KB (compression rate is 99.89%) by UniversalSAX.

Figure 3.15(right) shows congestions at the entrance made from the data. There are red circles which depict many people stopped for a long time at (1) and (2), which means congestions occurred near the reception a whiteboard. From the camera view (Figure 3.15(left)), the congestions are due to visitors' procedures at the reception and

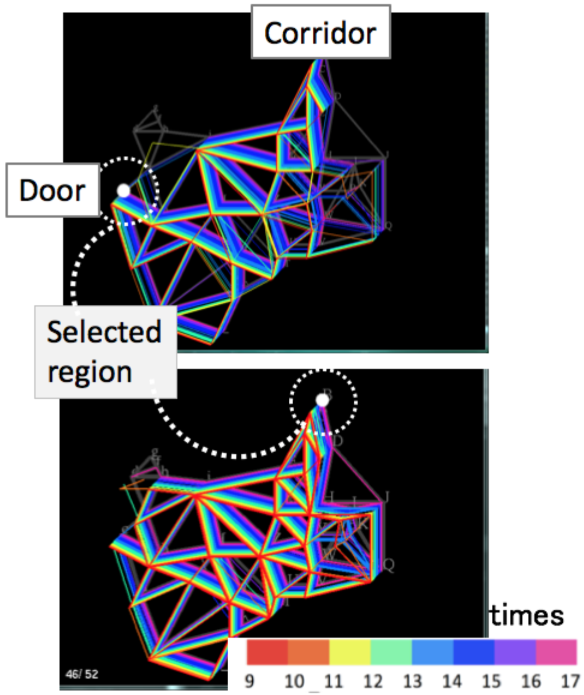


Figure 3.16: Animations of moving around the reception.

chatting in front of the whiteboard. In contrast, we can know that walkers moved smoothly at (3), since several yellow circles appeared there.

Figure 3.16 are examples of visualizing walking route in each time period. Trajectories in the upper picture correspond to walkers who passed a region near the door. In the lower figure, we selected a region near the upper corridor. One of the different points is that in the upper picture there are less red lines than in the lower picture. Red trajectories depict visitors who came in the morning from 9:00 to 10:00. That ' s because there were not many events in the morning so visitors who entered from left door were rare.

Corridor

This subsection introduces results of another experiment using datasets recorded at a corridor next to the exhibition room, as shown in Figure 3.17. We applied clustering to visualize the datasets, changing numbers of clusters and weights of features while

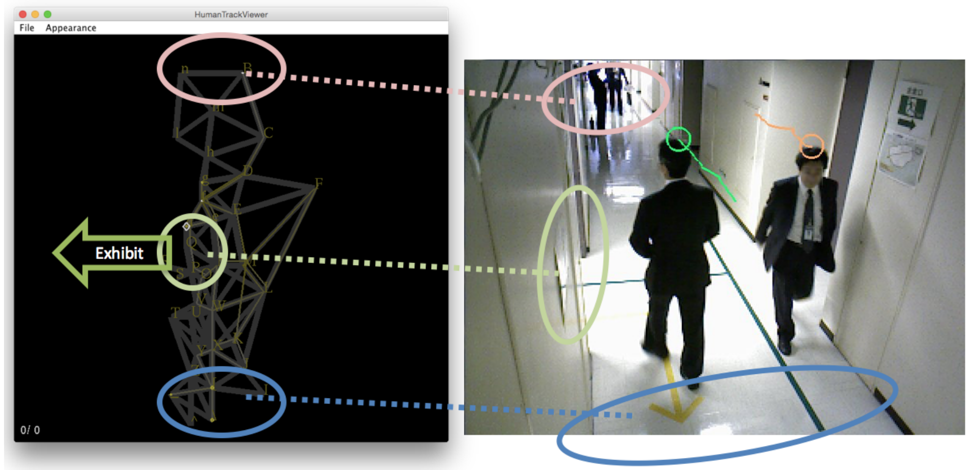


Figure 3.17: Camera view at a corridor and a result of visualizing passable routes.

calculating distances. We selected numbers of clusters as 5, 10, and 15. We can grab representative patterns of movement, meanwhile finding exception or persons walked at narrow spaces is difficult, while selecting the number as 5. On the other hand, unnoticeable trajectories formed a single cluster when the number is 10 or more. We changed in the weights as follows.

- $w_{length} = 0.1$ or 1.0
- $w_{times} = 0.0$ or 5.0

Major differences among clusters are not observed while changing in weights. Comparing constructions where w_{times} is 0.0 or 5.0, we can find correspondence among clusters and how trajectories belonged to different clusters.

We used 302 trajectories recorded from 9:00 to 10:00 and drew clusters one by one. Figure 3.18 shows the distribution of trajectories while staying time information is not taken into account. Figure 3.19 shows a result with applying staying time information.

Figure 3.20 is an example of visualizing results of clustering. Trajectories in these clusters did not disperse to many types of clusters. Clusters in the left rank are results where $w_{times} = 0.0$, and the right ones are generated where $w_{times} = 5.0$. While visualizing each cluster, we can find that trajectories in one cluster have a couple of

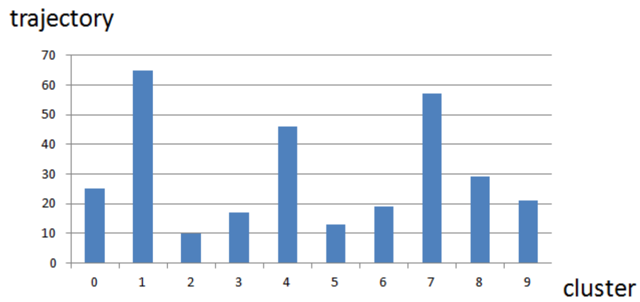


Figure 3.18: Generated clusters excluding staying time information.

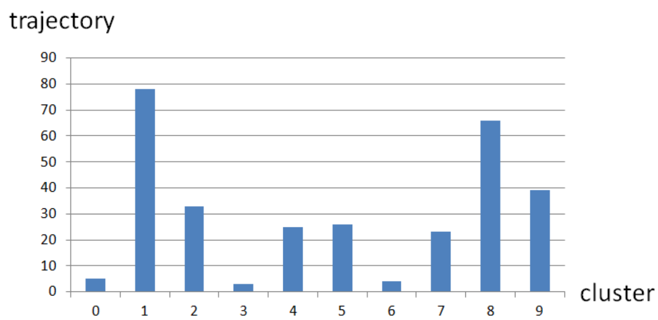


Figure 3.19: Generated clusters including staying time information.

representative departure and arrival points. On the other hand, when $w_{times} = 5.0$, many trajectories are belonged to clusters in the right rank. Arrows depict how many trajectories belonged to clusters in the left rank moved to clusters in the right rank. Observing the results of visualizing cluster 1 and 9 in the right rank, shapes of clusters are similar, though they have different saturations. Whole saturations of cluster 1 is higher than those of cluster 9. This means cluster 1 includes trajectories of walkers who stopped or walked slowly in the way, and trajectories belonged to cluster 9 represent moving people. However, while increasing weights of times, representative departure points are scattered and classifying based on way points are ambiguous. Specifically, trajectories started from the upper border and went to the lower one and trajectories from door of the left room are mixed. Users should change in weights based on whether they want to focus on way points or times to move.

Figure 3.21 is a part of a clustering result where the number of clusters is 15. We focused on cluster 14 generated where $w_{times} = 0.0$, and traversed trajectories in the

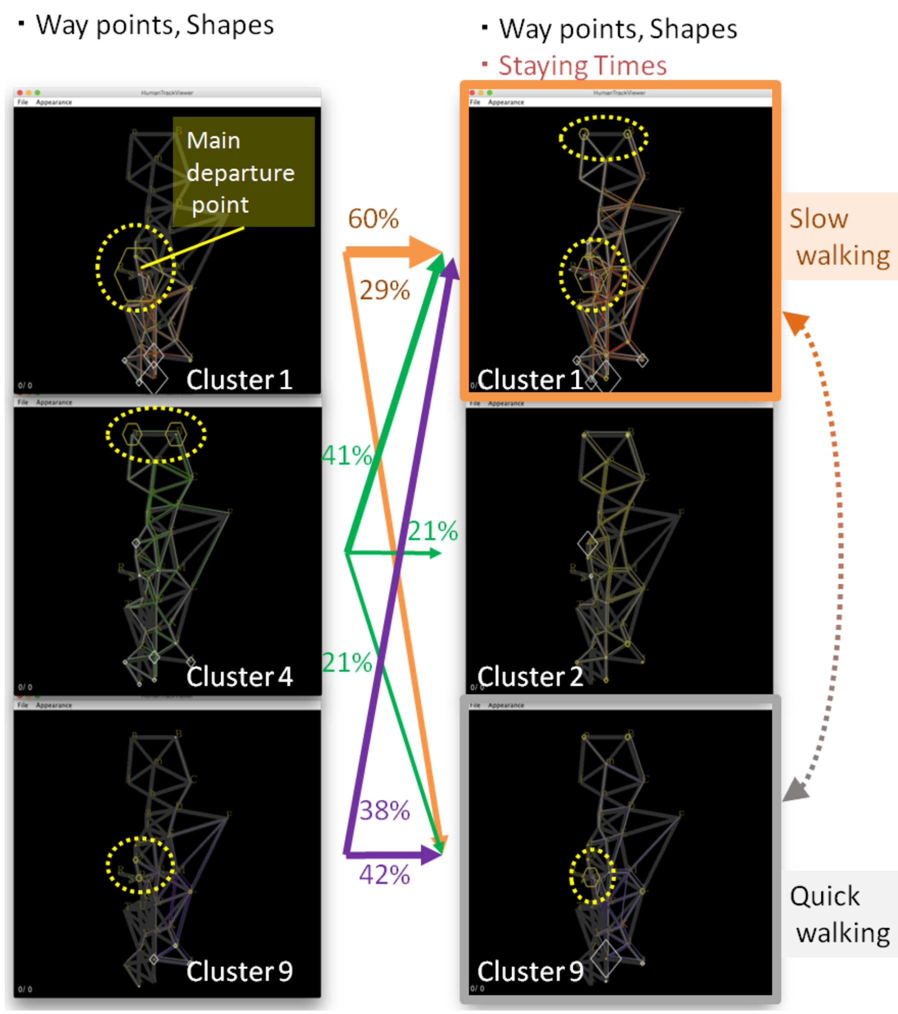


Figure 3.20: Comparing affiliations of trajectories in particular clusters.

cluster. While setting $w_{times} = 5.0$, almost 80 percent of trajectories went to five clusters in Figure 3.21, and approximately half of trajectories moved into cluster 1 or cluster 14. Cluster 1 includes trajectories that passed to the left side or the corridor without stopping for a long time. On the other hand, cluster 14 includes trajectories at the right side and high saturations which depict longer staying time. Trajectories that passed the left side denote people who came from the room and turned right soon to move. Meanwhile, people at the right side of the corridor are not always moving, but stopped near the wall so as not to distribute walkers. Thus, we can separate these different types of movement not from way points but staying times.

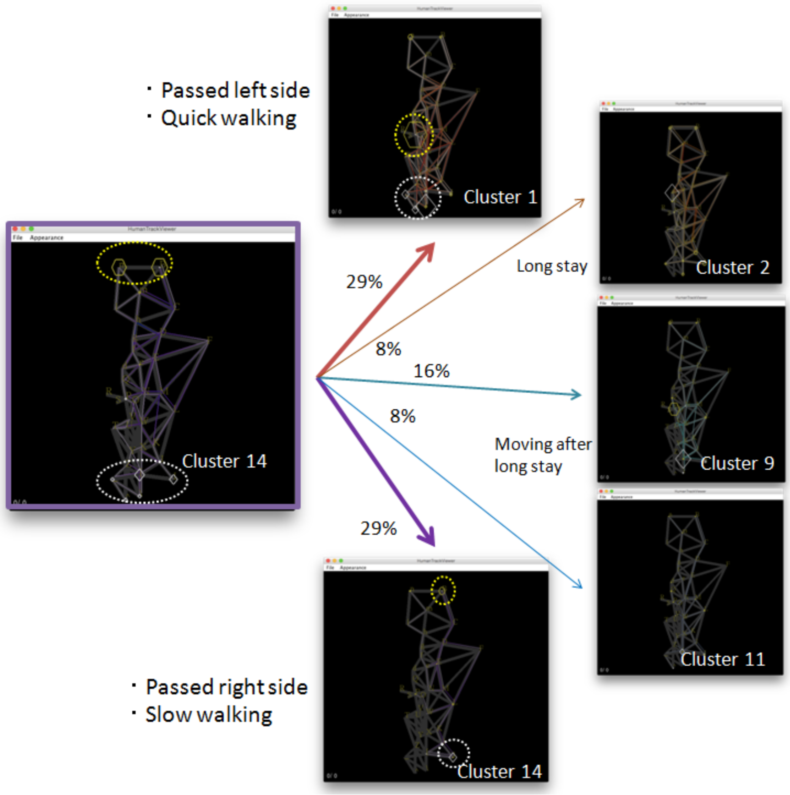


Figure 3.21: Dividing trajectories which passed almost same places by features of staying times.

3.4.2 Case of Poster Session

This section introduces the results of using poster session participants' trajectory data recorded in June 2017. We set eight Xtions to record trajectory datasets and visualized the dataset. Figure 3.22 (left) is an example of the field of view of Xtion. Figure 3.22(right) is the result of dividing the room into regions by UniversalSAX. As a rough layout, poster panels were placed on the wall approximately every 1m, and tens of participants could move between two rows of posters. The number of trajectories was 403 in 10 minutes measured by Xtion, and the total number of words generated by dividing the strings was 968. For the region segmentation, we used an almost grid-like segmentation result.

In the initial stage of the experiment, we tried to generate a result that strongly reflected the trajectory distribution as shown in Figure 3.23(right). These were the

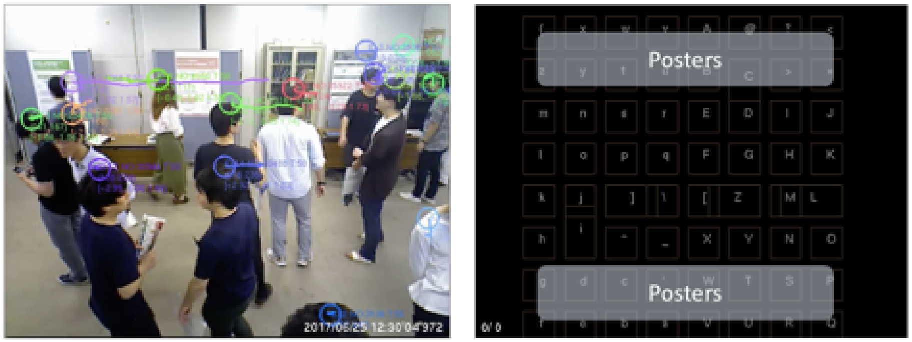


Figure 3.22: Venue for poster sessions. (left)An example of camera view. (right)Result of region division.

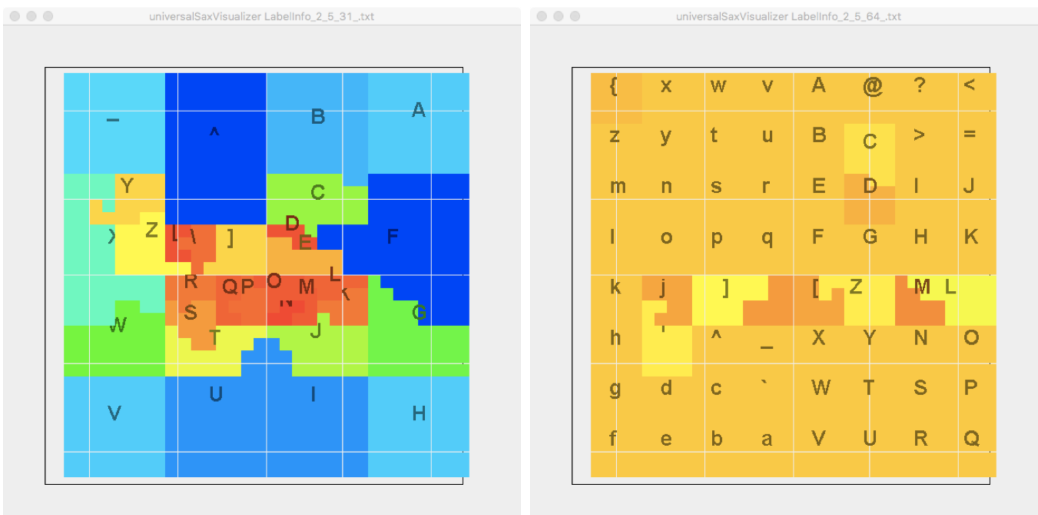


Figure 3.23: Comparison of results of UniversalSAX. (left)strongly reflecting distribution of trajectories. (right)division to like lattices.

results of segmentation using parameters similar to the case of the exhibition hall. The area near the center of the room with heavy traffic was divided into extremely small areas in this result. Since the space divided by UniversalSAX is based on a square, applying to data in a rectangular room seems difficult. The comparison of the distribution of the trajectory data including the coordinates around the walls that the participants did not able to pass affected the results. It seems dividing one poster into one region is preferable in the case of poster presentation venue. However, it is difficult to estimate the poster layout from this segmentation results. Therefore, we arranged the parameters and used the division result 3.23 (right) which similar to a substantial

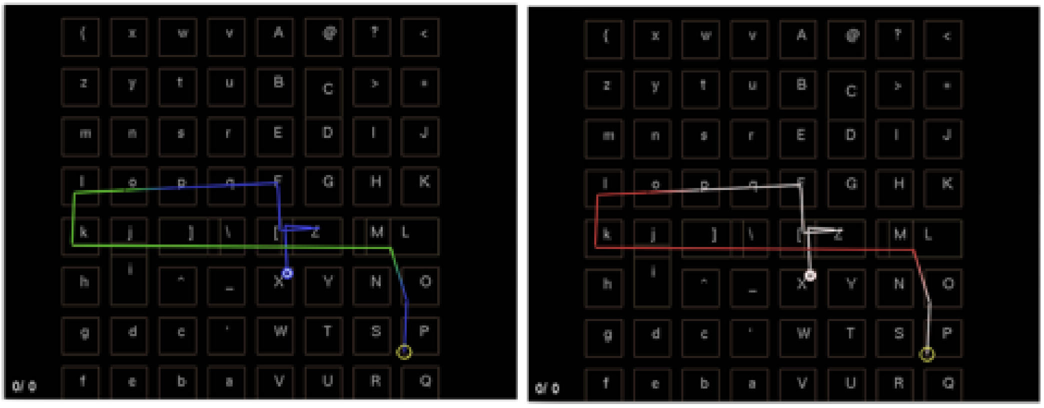


Figure 3.24: A simple trajectory of one person (left)Visualization of clusters to which each action belongs by hue.(right)Visualization of action appearance rate p using saturation.

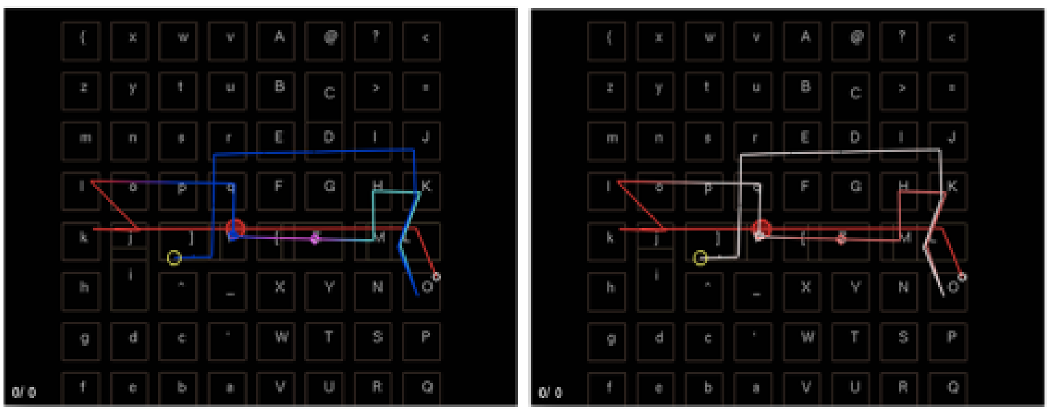


Figure 3.25: A complex trajectory of one person(left)Visualization of clusters to which each action belongs by hue(right)Visualization of action appearance rate p using saturation.

grid pattern.

Next, we introduce the results on the classification of the trajectories. First, we applied clustering to 968 words using the dissimilarity weighting the difference between the trajectory length and the passing point. The number of clusters was automatically determined by using the formula . As a result, 6 clusters were generated. Based on these results, we compared the visualization results of the trajectories representing individual pedestrians.

Figure 3.24 is the visualization result of the trajectory of the pedestrian who has moved from area 'P' to area 'X'. Since no circle was drawn in addition to the start and endpoints, this means the participant did not stop for a long time. Figure 3.24(left) shows the cluster to which the words that make up the flow line belong, expressed in hue. The behaviors of the pedestrian were classified as follows.

1. Movement from the end of the venue to the central passage (blue)
2. Movement from the right end to the left end of the venue (green)
3. Movement from the left edge of the venue to near the center and turning right (blue)

Here, Figure 3.24(right) which indicates the appearance rate of the cluster shows that the blue part of the trajectory is an action with a high appearance frequency, while the green part is an action with a low appearance frequency.

Figure 3.25 is the visualized trajectory of another pedestrian who has moved from the area ']' to the area 'O'. As in the case of Figure 3.24, the behavior of pedestrians was classified according to hue as follows.

1. Movement with a little detour to the right side of the venue (blue)
2. Movement slightly away from the wall (light blue)
3. Movement from near the center of the venue to the left (purple)
4. Movement slightly toward the opposite wall and movement to the left (blue)
5. Movement toward the right side of the venue (red)

From Figure 3.25(right), we can see that the light blue and purple part in the Figure 3.25(left) correspond to slightly less frequent actions, and the red trajectory in the left figure is a more exceptional action.

Next, we applied clustering on 968 words, taking into account both passing points and staying time. Thirteen clusters were generated in this process. Figure 3.26 is

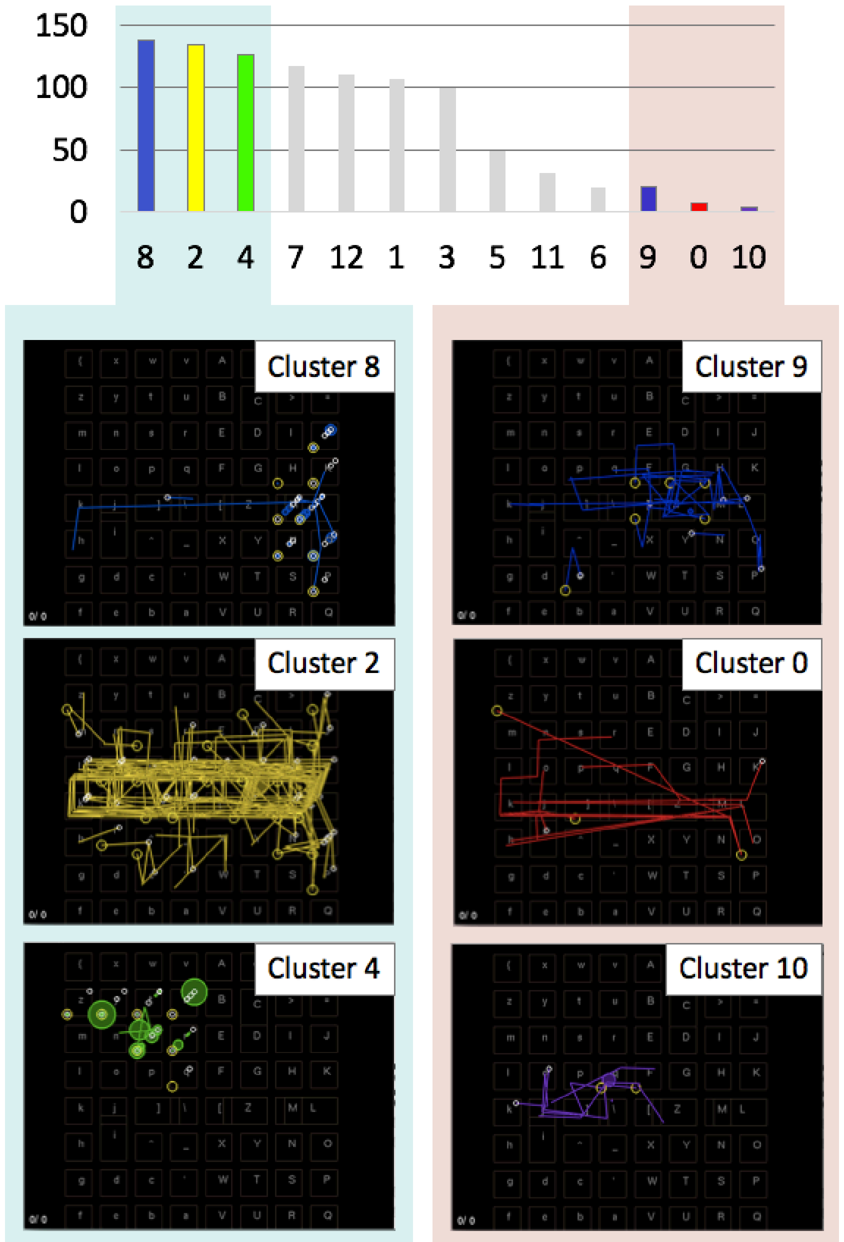


Figure 3.26: Clustering results of the poster session trajectory dataset.

a visualization of three clusters that have many words belonging to them (light blue group) and three clusters that have few words (pink group). The numbers on the horizontal axis of the bar graph are cluster IDs, and the numbers on the vertical axis indicate the number of words included in each cluster. Cluster 2, which contains many long trajectories at the center of the venue, is considered to represent the typical walking pattern involving occasional stopping and looking around at multiple posters. Cluster 4 represents behavior that stays in a narrow area for a long time. There was a small number of posters in this place, and many participants seem to had listened to the explanation for a relatively long time. Cluster 8 represents a group that stops at the venue. Not a few participants seem to had stayed away from the poster. On the other hand, clusters in the pink group in Figure 3.26 represent exceptional behavior. All clusters of 9, 0, and 10 represent a movement that moves around the center of the venue without stopping. This means that there was a small number of behaviors passing through the poster and crossing the venue as the dataset was the record at the crowded poster presentation venue.

3.5 Discussion

This section discusses the execution results. First, we review the results for each of the two types of data and then describe the achievement rates of the tasks described in 3.1.

First, we describe the results of the data at the exhibition introduced in 3.4.1. The presented technique can deal with huge people flow datasets and simply visualize movement patterns discovered from the datasets, as solutions of the two problems which we mentioned in Section 3.2. Compression by UniversalSAX and Run-Length encoding can archive close to 100% of compression rates. The rate is high enough to increase capacity for large datasets and process the data in a short time. Our system can reduce the sizes of processing data while preserving information of way points and staying times. These properties lead the results of the visualizations. This paper introduced that the presented analysis and visualization techniques generated meaningful results from the compressed data. We found various information; such as crowded places and popular walking routes. Finding such information is useful in a variety of locations including markets and stations, as well as exhibitions. Especially, the system distinguished way points and staying times by the formulas defined in Section 3.3.4. By arranging the values of weights in the calculation of distances, we found different types of patterns. For example, we found particular walking patterns including staying next to the wall in the corridor. Next, we look back at the analysis results at the poster venue described in 3.4.2. In this case, we could not verify the compression ratio sufficiently because the data used was small. However, we could detect some characteristic behaviors even from small data and classify them into clusters. In particular, both of typical movements and exceptional actions were detected, different from the results in 3.4.1.

In each case, we have succeeded in detecting characteristic behaviors and extracting them as clusters. In addition to being able to identify the location of the exhibits and posters where many people gathered, we found places that could be called layout problems, such as congestion and gathering of people who did not participate in the session, at the poster venue. This is close to the scenario using the analysis results described in 1.3. Therefore, we can see that the string keeping the passing point and the stay time was effective for feature extraction. On the other hand, there is a difference in the result of region segmentation in UniversalSAX. In 3.4.1, the region segmentation results could be used according to the flow line data distribution, but in 3.4.2, good results could not be obtained, and the segmentation results in a grid were used. Based on the above results, the cases where UniversalSAX can be applied effectively at this time are limited to “acquisition on a square floor” and “places with few prescribed routes and obstacles and high degree of freedom of action.” When applied to more diverse cases, users have to specify the obstacles and immovable coordinates in advance and divide the remaining free-traffic range based on the flow line distribution.

As for the issues listed in 1.1, the effectiveness of the proposed method for data compression and feature extraction was confirmed. Furthermore, the problem of how to present the results of the visualization of the flow line was largely achieved by visualizing both individual and group movements and visualization of clustering results. However, there is still room for improvement in the visualization of trajectories. In particular, since a function for comparing multiple trajectories was not introduced, conventional methods such as switching the cluster to be visualized and comparing two visualization result screens side by side were used.

Future issues of this study are follows.

- Improvement of formulas and conditions of clustering.
- More functionality for searching for walking routes.
- Manual specification of region division in the setup process.
- Inferring the semantics of walking actions.

In these experiments, we demonstrated shapes of trajectories and staying times are important to distinguish movements of people. On the other hand, we should discuss further goals and methodologies, such as how to automatically specify appropriate number of clusters for proper classification, or how to determine balance of weights for calculating distances among sequences of characters.

We would like to develop additional user-specified conditions to the search user interface. We would like to adjust the costs for distance calculation according to the user-specified conditions, so that we can search for various walking routes from various viewpoints. It is also useful to develop a sketch user interface to query the walking routes with arbitrarily shaped trajectories, instead of specifying sequences of characters.

In current processing of region division, we cannot specify positions and shapes of borders between the regions. We would like to develop a user interface to specify the attributes, positions, and shapes of the regions for the cases that we know the objects in the scene which should divide the regions.

In addition to the above development, we would like to challenge how we can visualize the semantics of walking actions from the results of summarization and grouping of walking routes. Adopting a method of analyzing actions of people, we would like to visualize its result with trajectories.

3.6 Conclusion

In this chapter, we proposed a technique to visualize features of people flow data by converting real values of walking routes into sequences of characters. Section 3.2 presented importance of understanding features of people flow and problems on analysis. Section 3.3 presented the technique for compression and visualization of people flow dataset. Section 3.4 stated results of experiment of applying the visualization technique to datasets recorded in some events. Section 3.5 discussed results of the experiments and future work. This technique allows users finding nature of people flow and important factors quickly. Especially, we focused on two kinds of features, way points and staying times, then showed they are effective to classify trajectories.

Chapter 4

Eye Tracking Data Visualization

4.1 Introduction

This chapter introduces the second proposed technique for visualization of eye-tracking data. Section 4.2 includes overview of the proposed technique. Section 4.3 introduces the processing flow. Section 4.4 presents two case studies applying the technique. Section 4.5 discusses the result, and Section 4.6 is summarization.

4.2 Overview

The development of eye-tracking devices brought the spread of the study of eye-tracking data. Analysis of eye-tracking on a stimulus such as web pages or advertisements [77] indicates what people are interested in. Furthermore, eye-tracking is effective to measure proficiency in learning[78]. Such information is effective to improve the design or layouts of the stimulus. There have already developed various methods of eye-tracking data visualization[79][80]. Especially, visualization of trajectories is an intuitive expression even for non-expert of visualization. However, the following two problems are still left. One is the lack of visualization method to show transitions among multiple areas of interest (AOIs). AOI means region that surrounds an object such as a text paragraph and picture, or part of it, and frequently used in the analysis [66][81][82]. Figure 4.1 is a simple example of AOIs on a website. The order of scan-path transitions between AOIs indicates relationships in contents or products such as

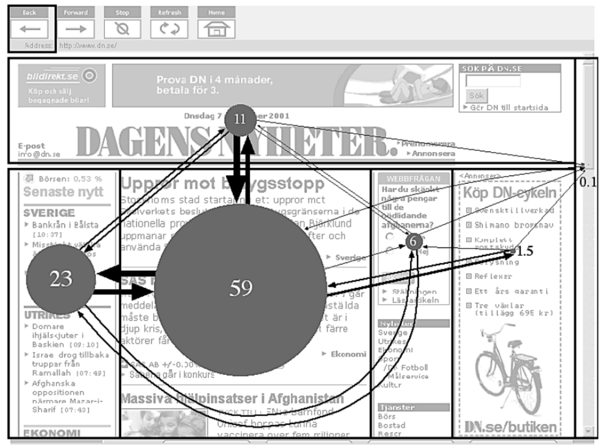


Figure 4.1: An example of an analysis of gazing times in each AOIs. (refer from [81])

"The customer who likes product A also like product B." However, the visualization of multiple scan-paths often involve overlapping of the paths and reduction of visibility, therefore we have to carefully select elements to visualize. The second problem is a shortage of methods to compare multiple scan-paths. Finding common or different parts in the scan-paths leads to understanding popular or exceptional behaviors. In existing comparison methods, basically users have to overlap or line up scan-paths to compare them[68]. Such a way sometimes brings overlooking differences between the paths since too much information is shown. In this chapter, we propose a visualization method of characteristic parts of scan-paths on static stimulus, especially focusing on solving the above two problems. First, we obtain scan-paths when the subject observes a stimulus such as a web page or advertisement. Next, we create hierarchical AOIs according to the content of the stimulus. Then, we convert the scan-paths to strings following the layout of the AOIs. We apply N-gram [83] to the strings and extract patterns which mean orders of access between AOIs. Finally, we visualize the results of pattern extraction and transitions between the AOIs on the stimulus like Figure 4.2. (We will mention details of the panels (A) to (F) in the Section 4.3.3.) These techniques are effective to find frequent behaviors or differences in participants. As case studies, we introduce two case studies using different stimuli, a web page of Wikipedia and a

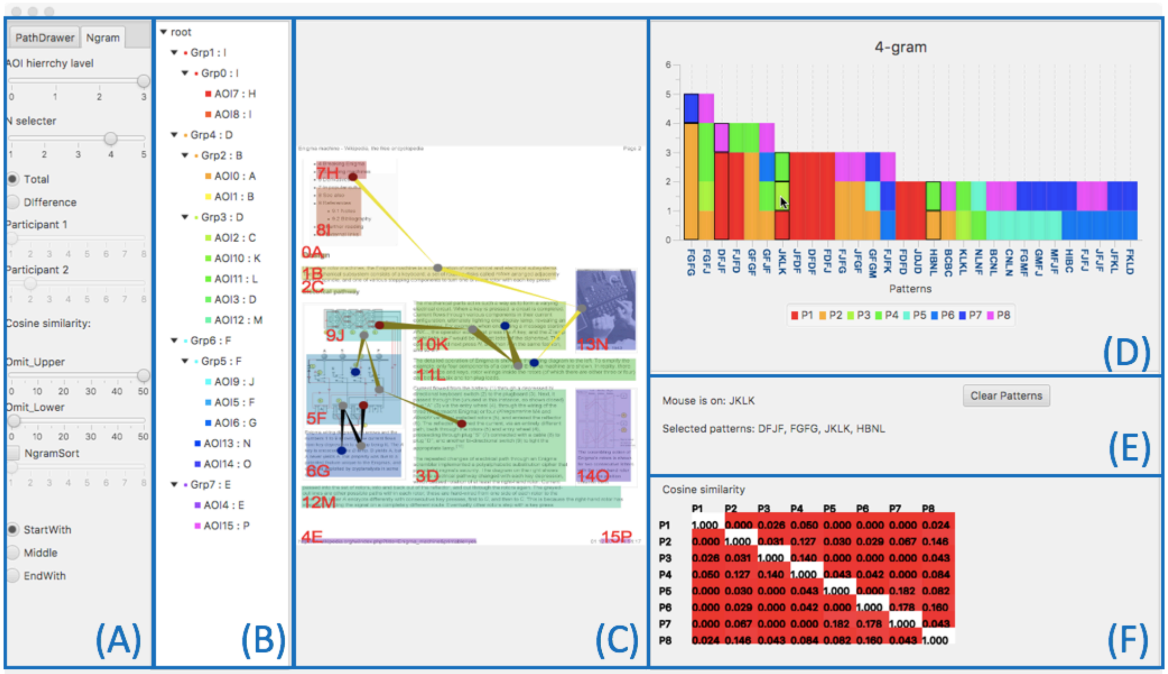


Figure 4.2: Construction of our visualization system.

poster with several illustrations.

4.3 Implementation

This chapter contains the procedure of the gaze trajectory visualization method. Section 4.3.1 explains how to acquire the gaze trajectory and generate a hierarchical AOI based on the data. Section 4.3.2 describes a method of converting gaze trajectories into strings based on the layout of AOI and extracting patterns using N-grams. Section 4.3.3 contains how to visualize the shape of the extracted pattern and the difference between the behavior patterns of each subject on a single window.

4.3.1 Recording of Eye tracking Data and Generation of AOIs

Figure 4.3 shows the processing flow of our visualization system. First, we use record eye-tracking scan-paths $C_i (1 \leq i \leq p)$ from p participants using eye tracking device.

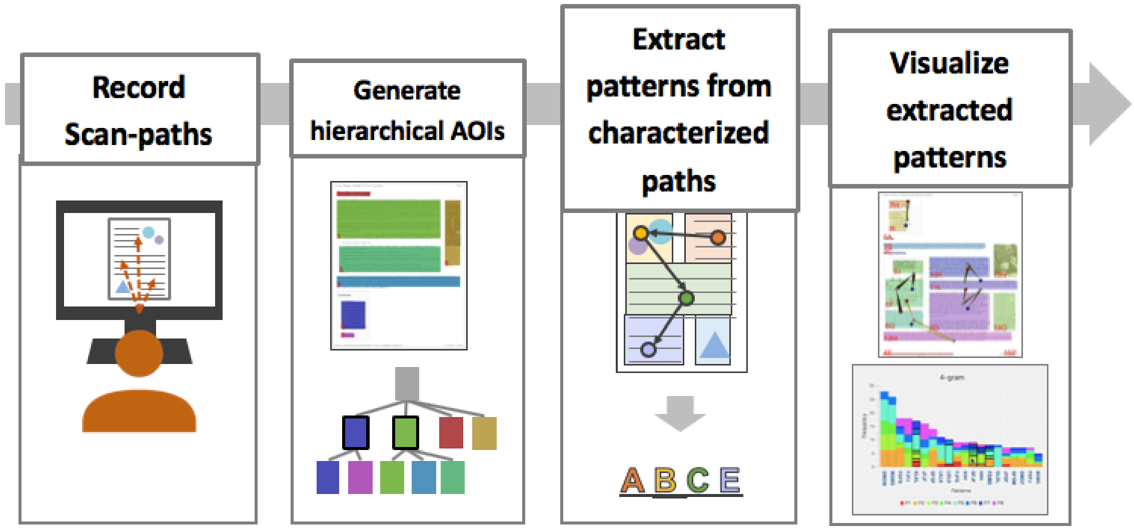


Figure 4.3: The processing flow of our proposed technique.

We define the scan-paths using position c_q and the number of points n_i.

$$C_i = \{c_1, ..., c_{n_i}\} \tag{4.1}$$

Next, we define hierarchical AOIs on the stimulus. Hierarchical AOIs are a collection of AOI layouts on a single stimulus that is saved as hierarchical data according to a degree of fineness [84]. The layout of AOIs effects results of pattern extraction, thus users can change behavior to analyze by switching layouts of AOIs. In this time, users can efficiently generate various AOIs by merging or dividing existed AOIs as parents or children of the old AOIs. Figure 4.4 shows the processing flow of the AOI definition consists of 2 steps. In the first step, users can generate a detailed layout of AOIs that one object gets single AOI. Secondly, the users can set rough AOIs and hierarchy by merging parts of the existing AOIs. The detail of each procedure is as follows. First, users can manually or automatically generate the most detailed AOIs. One AOI gets the shape of a rectangle to keep visibility when the user visualizes scan-paths and AOIs on the stimulus as mentioned in Section 4.3.3. In the automatic AOI definition, color information of the stimulus affects the AOI layout as shown in Figure 4.4 (1) to (4). The detail of each step is as follows.

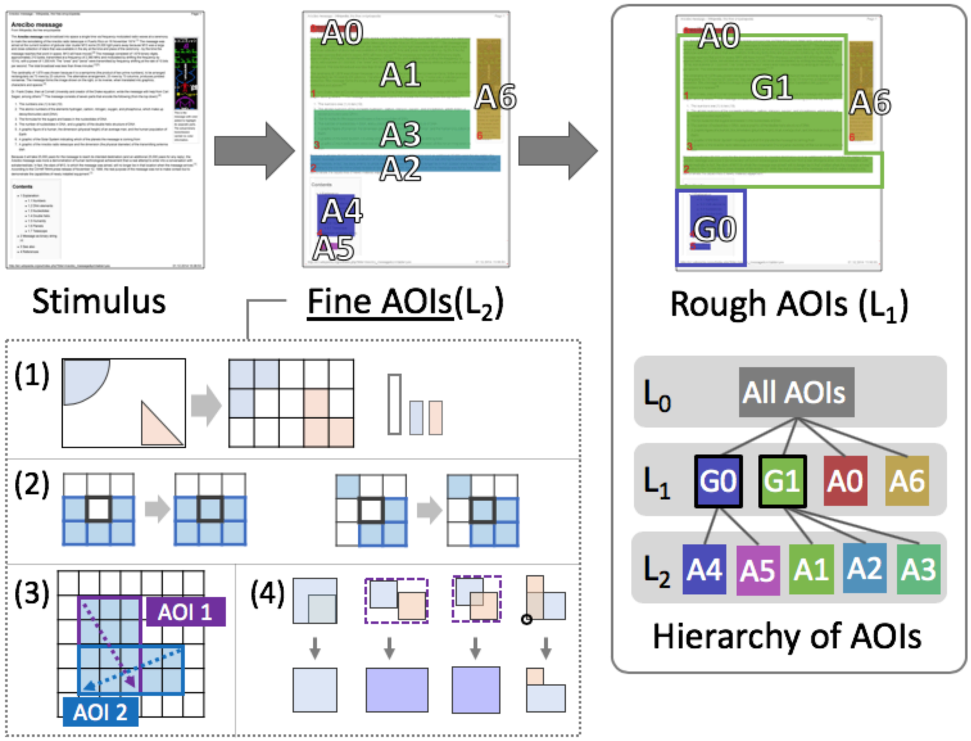


Figure 4.4: Flow of the definition of AOIs.

1. Apply mosaic processing and posterization, and select the most frequent color as blank color. The other colors are regarded as objects.
2. Paint blank area which is the dent of an object by another color.
3. Surround the objects by rectangles and set temporary AOIs.
4. Arrange shapes of overlapping AOIs by merging or transforming into two AOIs.

On the other hand, users can manually set AOIs to do fine regulation. The users can surround objects one by one and set each AOI. In this time, if the newest AOI and the existing AOI are close, the newest AOI adjoin to the existing AOI to avoid overlapping or leaving a minute opening between the AOIs. The next step is setting rough AOIs and hierarchy based on the detailed AOIs. Users can merge multiple AOIs to one large AOI and also subjectively set various AOI layout, for example, “Gathering close AOIs” or “Merging figure and tables.” This is an efficient way to increase AOI layouts since

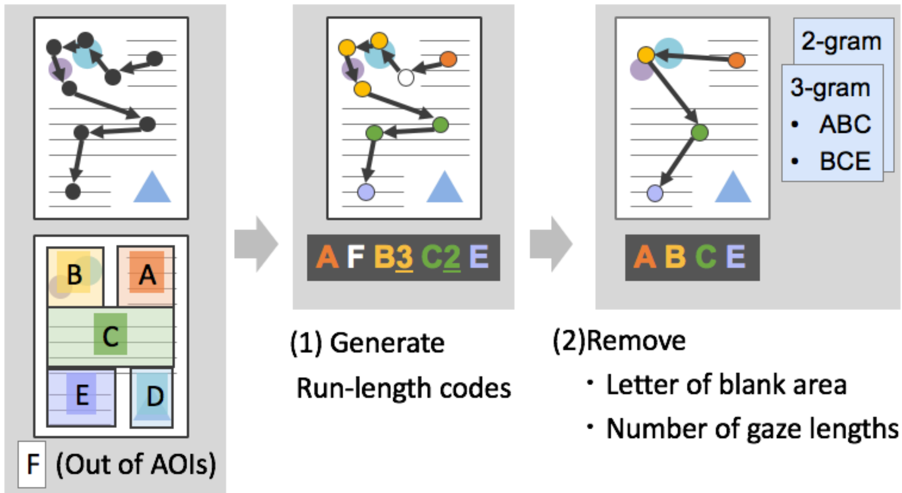


Figure 4.5: Conversion of a scan-paths to a string.

the users can generating rough AOI from existing AOIs. Figure 4.4 includes an example of hierarchical AOIs. We define the level of fineness as k, and corresponding AOI layout as L_k.

4.3.2 Symbolization of Scan-paths and Extraction of Patterns by N-gram

Next, we convert the scan-paths to strings and and divide to N letters parts using N-gram. These process can show the variety and order of AOIs which the scan-paths passed, and combinations of AOIs which the participants focused. Figure 4.5 shows the flow of the conversion to strings. First, we use the finest AOIs L_h and allocate letters to each AOI and blank area. Using this result, we convert the scan-paths C_i generated in Section 4.3.2 to strings C_i' as shown in Figure 4.5 (1). Next, we apply run-length encoding to C_i' and generate R_i. R_i consists of the repetition of m_j the AOI or blank area which the scan-path passed, and t_j which means the times of gazing m_j. Now the run-length codes reflect the finest AOI layout and means the detailed behaviors. Then we convert R_i to additional codes which reflects rough AOIs and shows summarized behaviors. A letter that corresponds to a group are the same as the largest AOI in the group. This process prevents too much gathering of scan-paths at a small AOI when the trajectory visualization as mentioned in Section 4.3.3. We make strings for all L_k, in other words, we generate h strings from one scan-path. Next, we extract transition patterns between AOIs using the run-length codes. As a preparation, we delete (1)the letter which means a blank area (2)the length of gazing t_j and (3)m_j with small t_j from the string. As a result the strings only preserve letters that indicate AOIs that the participant noticed and the string shows the transition between the AOIs. We apply N-gram to find AOIs in which the scan-paths successively passed. N-gram is a popular simple method for text analysis and has the advantage that it is practicable for small-scale datasets. Finally, we compare the differences between extracted patterns from each scan-path. We calculate the variety and frequency of each pattern and get the cosine similarity in p strings to compare the whole scan-path. Here, we introduce a supplementary method of calculating the similarity that can be applied to eye-tracking data in addition to the cosine similarity. First, the similarity can be calculated using the correlation coefficient of a random variable, such as Pearson's correlation coefficient. This coefficient is basically similar to cosine similarity. The difference is that the data is assumed to follow a normal distribution, and the average value of the vector is used. We can also apply the pattern similarity and the deviation pattern similarity. The difference from Pearson's correlation coefficient is whether the used average is of each vector or the entire vector. The average value of these vectors, in other words, whether the average trajectory is effective at the time of analysis depends on the data to be analyzed. For example, the average trajectory may be helpful when a route is specified and users want to detect the behavior deviating from it. On the other hand, when there is a high degree of freedom and various flow lines are included, we have little reason to actively apply them because the average trajectory itself has no significant semantics.

The gaze trajectory data used in this study recorded the movements of the subject when he/she freely observed a specific web page or poster, and no specific behavior was assumed and there were many variations in different participants. In this case, defining a proper average trajectory is difficult, and therefore, we have little reason to use the correlation coefficient using the average value. On the other hand, a specific task (searching for specific information, tracing numbers in order, etc.) is often imposed on the subject in gaze measurement. Along with the tasks, we need to consider the cases where the subject has "the action that becomes the correct answer", and the behavior of the subject is strongly induced. We also need to consider the cases where the gaze movements of a skilled person and a beginner are compared using a specific task. In that case, the correlation coefficient may be applied to detect how the behavior deviated from those assumed behavior. In addition, we can apply an index that compares the similarity between sets by treating a string as a set of words. Jaccard index, Dice index, and Simpson's diversity index are representative sets of similarities. Each coefficient calculates the proportion of elements common to the two sets, and the weight increases in the order of Jaccard index, Dice index, and Simpson's diversity index. Since the lengths of symbolized trajectory data often differ greatly, the Simpson's diversity index is considered to be effective among them. However, the number of occurrences of each word is not considered in any coefficient. It is therefore basically better to use the above comparison as a string or comparison as a vector. On the other hand, the comparison as a set may be effective in cases where the same behavior is unlikely to be repeated, or where special emphasis is placed on "whether or not a particular behavior has occurred at least once".

4.3.3 Visualization of Scan-paths and Extracted Patterns

Finally, we visualize the result of pattern extraction by N-gram. The visualization system includes panels (A) to (F) as shown in Figure 4.2, and each panels have following roles.

(A) Selection of datasets and parameters

(B) Showing hierarchy of AOIs

(C) Drawing a stimulus, a layout of AOIs, and scan-paths

(D) Visualization of pattern extraction by N-gram

(E) Showing the list of visualized patterns in (C)

(F) Display of cosine similarity in the strings

(C) and (D) are the main visualization view in the system. In the panel (D), users can set fineness of AOIs k and length of extracted patterns N, and select contents to visualize from "total" and "difference." "total" is the display of frequency of all extracted patterns, and "difference" shows the comparison of included patterns in two participants that users specify in panel (A). X-axis means each string as a pattern, and Y-axis shows the frequency of each string. Users can sort strings for example by "Frequency in all scan-paths" and "Search for patterns that a specified scan-path include much." In panel (C), we display a little bright stimulus and translucent AOIs. This is to keep visibility when users draw scan-paths on the stimulus and the AOIs later. The number and letter in the lower-left corner are for the identification of the AOI. Colors of AOI follow the rule below and reflect the hierarchy of AOIs. First, we set the cost of leaves in the hierarchy (each AOI) as 1, and calculate the other parent nodes (groups of AOIs) as a total of the cost of their children. Next, we collect colors in different hues as many as the AOIs and save them in a list following the order of the hues. Starting from the root node, we distribute each color to the nodes. We allocate the color at the top of the list to each node and pass colors as many as the cost to the child nodes. If the node have multiple child nodes, we divide the list and pass different colors to each of them. As a result, close AOIs in the hierarchy have similar colors. Furthermore, little changes occur when users change L_k and can keep visibility. We draw a directed graph that means extracted patterns, on the stimulus and the AOIs. Users can select patterns to visualize by clicking bars in panel (D) or AOI in panel (C). When selecting AOI, the users can search for patterns that passed the AOI as a starting point or end or relay point. Figure 4.6 is an example of the graph. We put nodes at AOIs that the selected patterns passed, and connect the nodes by edges to

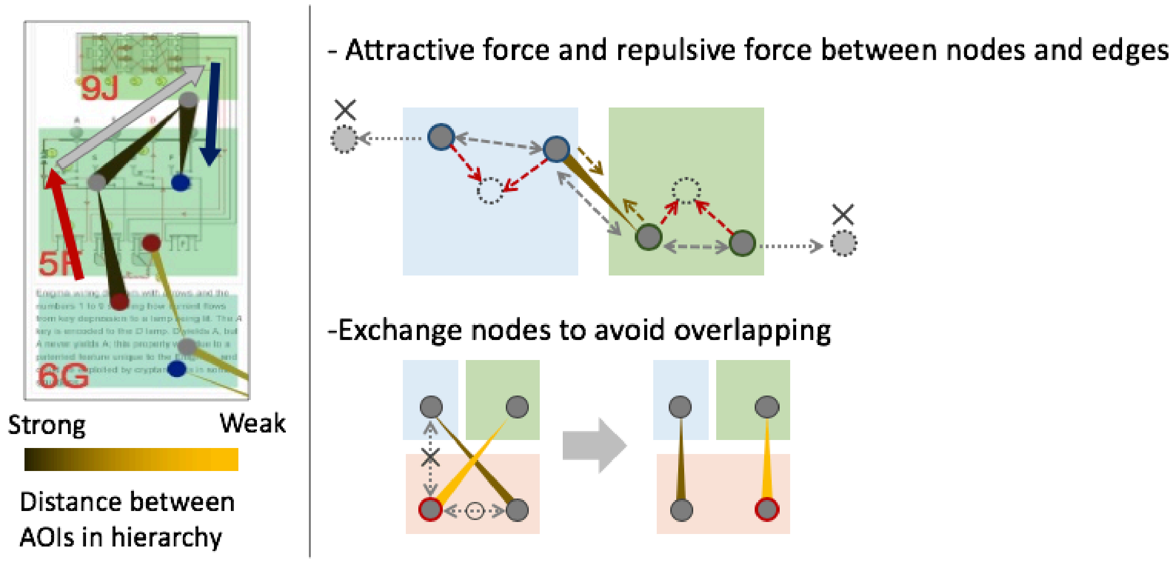
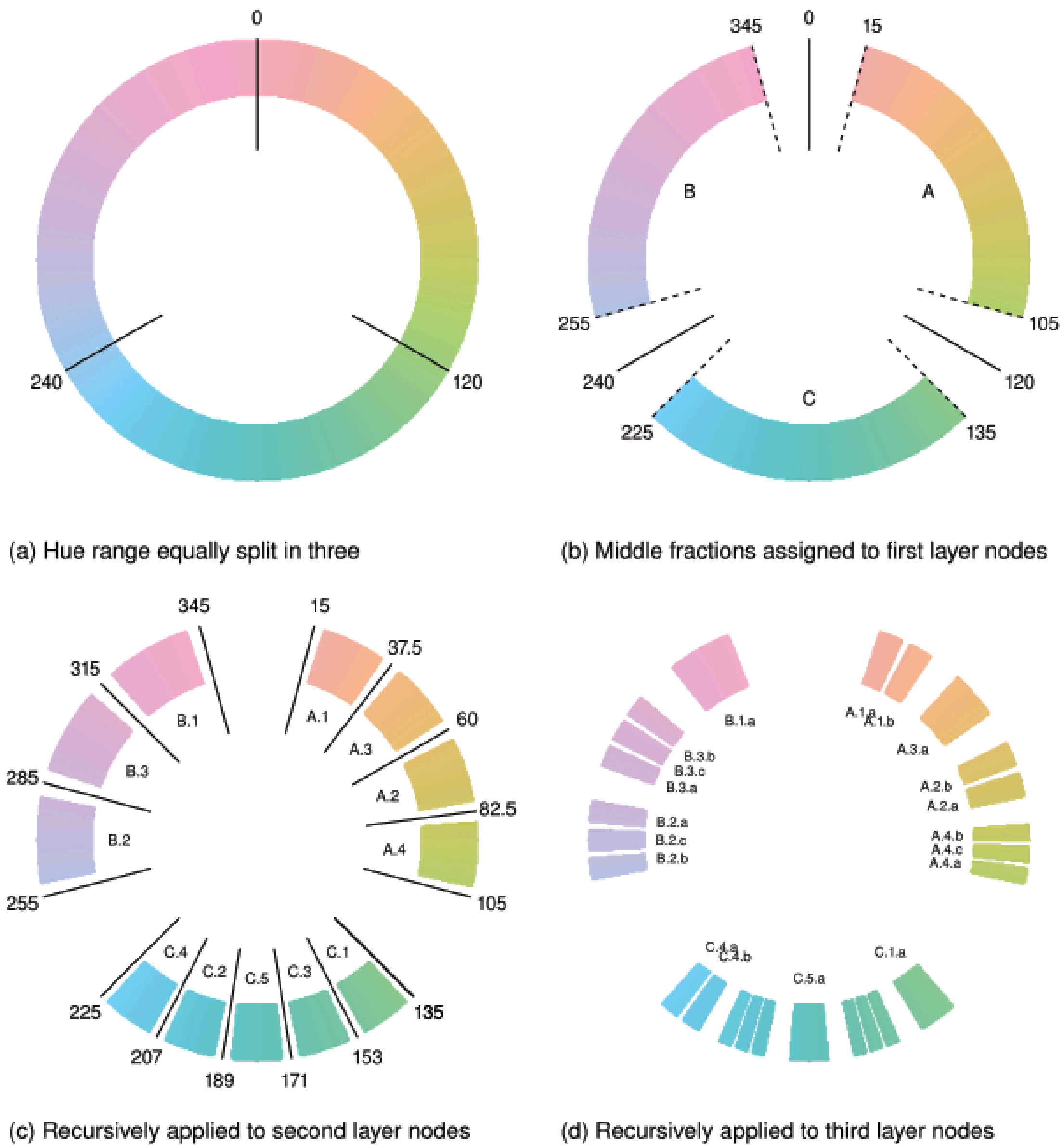


Figure 4.6: Force-directed graph to display patterns.

display transitions. Red nodes are the starting point of the patterns and blue ones mean endpoints. The other nodes get gray. The color of edges means relations between the connected AOIs. Edges get yellow if the corresponding string connects AOIs in rather different groups. These edges can show connections between the AOIs that the user who set the hierarchy did not expect. We put these nodes and edges following the rules which are based on force-directed graph layout [85] and the conditions below. First, we added attractive forces from centers of AOIs to collect the nodes in the same AOI to avoid the nodes overlap the borders of AOIs. Furthermore, we apply the exchange of nodes if the crossing edges have nodes in common AOI to reduce the overlapping of edges. These processes can display positions and orders of AOIs that the scan-paths passed, and also avoid overlapping and gathering of nodes and edges espeially in a common AOI. Each AOI uses a different hue color to make it easier for the user to distinguish areas. We use the method of Tree Colors [86], then the colors of large AOI groups including many AOIs are greatly different, and individual AOIs belonging to similar groups can be given similar colors become. Figure 4.7 describes an example of hue assignment.



(a) Hue range equally split in three

(b) Middle fractions assigned to first layer nodes

(c) Recursively applied to second layer nodes

(d) Recursively applied to third layer nodes

Figure 4.7: Assignment of hue values. (refer from [86])

4.4 Experiments

In this section, we introduce the results of visualization using two types of static stimuli. We used Tobii Pro T60 XL to record scan-paths from Eight participants P_i. The participants observed each stimulus for 90 seconds, and we recorded 16 scan-paths. When applying N-gram to analyze the scan-paths, we referred [87] and set N relatively small values, from 2 to 5.

4.4.1 Case A: Page of Wikipedia

First, we applied the automatic AOI definition to the stimulus and set a hierarchy of which depth h is 3. Then we generated 8 strings from the scan-paths. Figure 4.8 is the visualization of scan-paths in the finest AOIs L_3. The left result is the visualization of patterns that starts from the text paragraph surrounded by the circle. The value of N is 3. In this result, many red nodes which mean starting points of patterns gather in the left picture and introduction. Furthermore, there are many gray nodes on just above text paragraph of the circled AOI. This means the participants read the text paragraphs following the supposed order. On the other hand, there are a small number of yellow edges that detour first text paragraph and passed the upper right figure. The right picture of Figure 4.9 is the visualization of patterns extracted by 4-gram and passed the circled text paragraph. Many red nodes are in the AOI in the right area, and blue nodes are on the left figure. Furthermore, no pattern directly moved to the next text paragraph from the circled paragraph. These mean many participants moved the gazing points from the right area to the left area looking the figures and text paragraphs. Like this, we visualized frequent patterns and relations of AOIs on the stimulus. Next, we introduce the visualization of differences between participants. We set k as 1 and AOI layouts L_1 which contains (1)introduction(red) (2)main text(brown) (3)figures(blue) (4)footer(purple). Then we extracted patterns by 3-gram as shown in Figure 4.8. We visualized cosine similarities by a matrix and found that raws of P_1, P_3, P_5 were strong red which means low similarity. Furthermore, there are three yellow green bars on the right area of the bar chart. These elements show a unique behavior of P_3, specifically transitions among the table of contents, main texts, and figures. Like this, we visualized that participants P_1, P_3 and P_5 had different behaviors from the other participants.

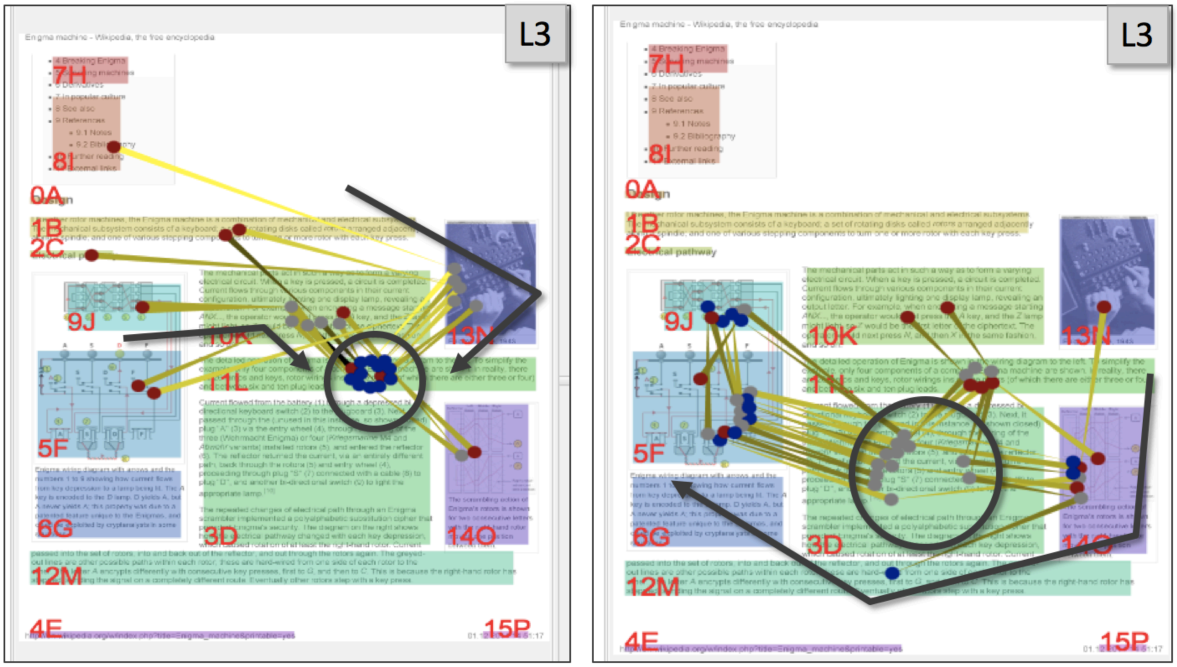


Figure 4.8: Visualization result of transitions between paragraphs on the detailed AOI (L_3)

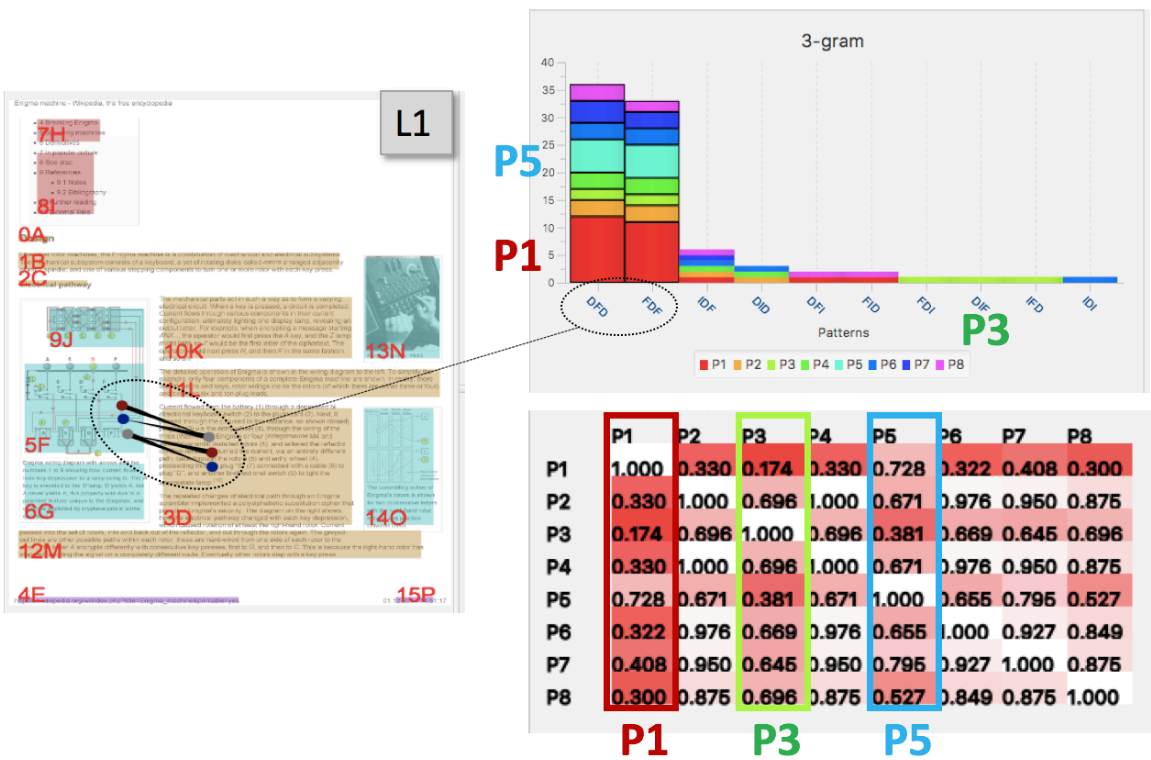


Figure 4.9: Comparison of differences in scan-paths on the rough AOI (L_1).

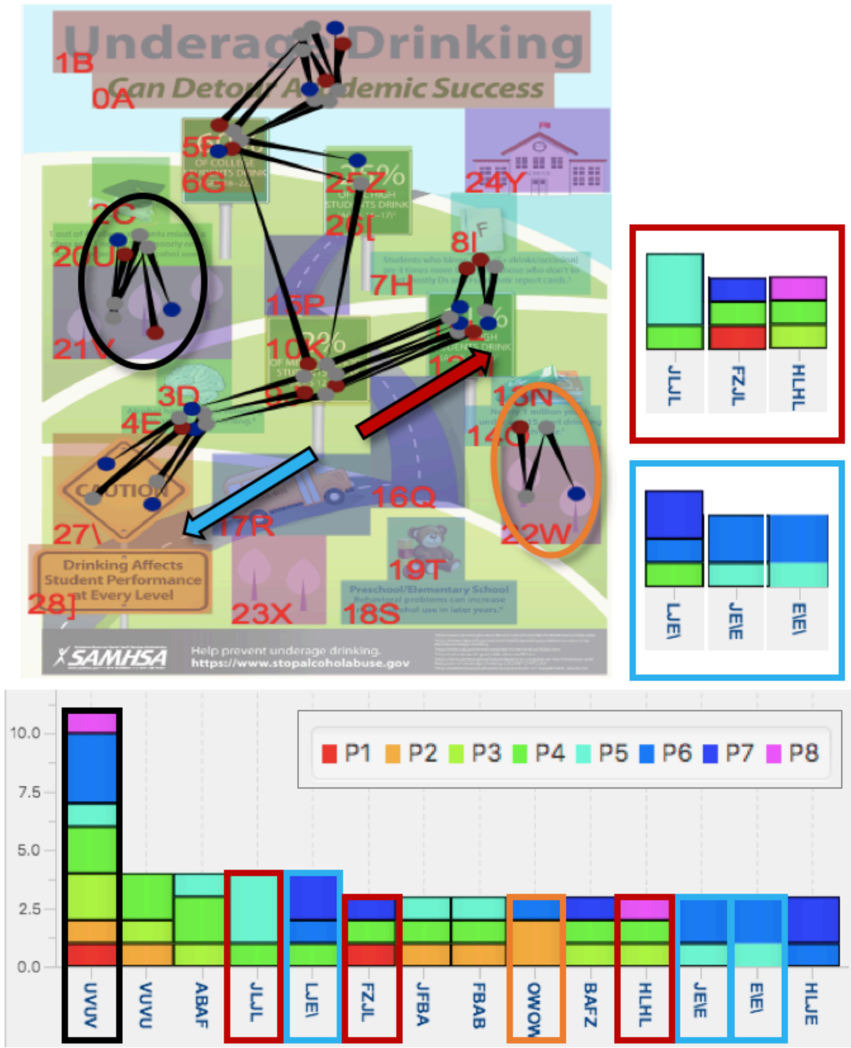


Figure 4.10: Visualization result of frequent patterns in the poster.

4.4.2 Case B: Poster to Alert Influence of Drinking

Next, we introduce another case that we used a poster with a more complex layout as a stimulus. We specified the layout of AOIs as this stimulus was difficult to estimate the color of the blank area automatically. Figure 4.10 is the visualization result of frequent patterns extracted by 4-gram. The most frequent behavior is the transition between the circled texts and illustrations. 7 participants except for P_7 had this action. Briefly, there were two representative behaviors. One is the red pattern, starting from the title and reached the right board (L) or texts (H) through the centerboard (J). The other is

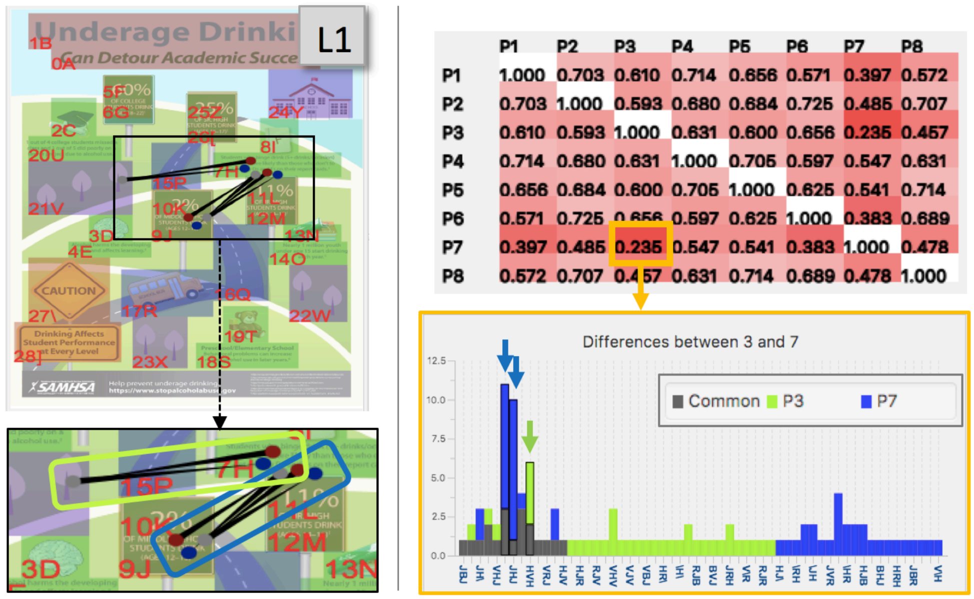


Figure 4.11: Comparison of the two participants P_3 and P_7.

the blue pattern separated on (J) and went to a light yellow board (\). We checked the color of the corresponding bars in the bar chart and found differences in the participants. P_4, P_5, P_7 included the both patterns. P_1, P_8 only included the red pattern and P_6 had the blue one. Furthermore, we can find P_2 and P_6 paid attention to the lower right texts in the orange circle. Figure 4.11 is the visualization result of differences between participants. We used AOI layout that includes (1)title (2)numbers above the four square boards (3)texts under the square boards (4)illustrations on the grass (5)texts under (4) (6)illustrations of trees. First, we selected P_3 and P_7 and visualized patterns that they include. Common patterns (bars including gray) are about one-third and frequencies of each pattern are different. We found differences that P_3 had many transitions between (5) and (6), and P_7 moved much between (5) and (3).

4.5 Discussion

In both cases A and B, we succeeded to visualize characteristic patterns by combining drawing scan-paths and showing the result of pattern extraction. We look back to the two problems that we mentioned in Section 4.1. The first one is lack of the visualization of patterns that passed multiple AOIs. We succeeded to visualize relatively long patterns keeping visibility since we selected only remarkable patterns from the bar chart. In this time, we found unique behavior in particular participants since we visualized not only representative patterns but also exceptional behavior. Furthermore, the force-directed graph was effective to avoid overlapping in AOIs and classify edges with common starting points but different directions. The second problem is comparison of multiple scan-paths. We displayed common and exceptional behaviors on the "different" view and selected patterns that users were interested in. As a result, we visualized only common or different part in the scan-paths as trajectories. Showing such a characteristic paths are not major in the existing methods.

Regarding the tasks mentioned in section 1.1, we believe that almost all tasks except for the compression of the flow line data have been achieved. First, N-gram was able to detect common patterns between trajectories to some extent and we were able to grasp the characteristics of actions. Regarding the contents of the visualization, a wide range of actions were visualized as a result of increasing the items that can be adjusted by the user, such as interactive pattern selection and selection of the AOI layout to be used. Furthermore, the behavior between subjects was compared in detail and the results were clearly visualized, unlike the results of people flow visualization in 3.4.

In this case, we used the data obtained when observing a stimulus freely without specifying a task. As a result, on both the Wikipedia page and the poster, we could visualize that many participants had looked at noticeable figures. This result is considered to close to scenario 1 indicated by 1.3. On the other hand, we did not set "correct action", so we could not obtain a direct result of the design layout improvement as in scenario 2. By applying data when a task such as searching for specific information is specified, or by evaluating numerical values such as the length of the gaze trajectory or the distribution of the AOI through, we will be able to obtain results which lead to getting an evaluation of layouts in the stimulus. We think that scenario 2 can be clearly realized in this way.

On the other hand, some problems are still left. We could not find common patterns when we set 4 and 5 as N. This means our method should be developed to find longer patterns. We would like to regard partially similar patterns as same if necessary.

4.6 Conclusion

In this Chapter, we proposed a method to extract patterns from multiple eye tracking scan-paths. Section 4.3 presented details of the proposed technique. Section 4.4 introduced two case studies using the technique, and Section 4.5 discussed the results. The combination of showing results of N-gram and a force directed graph to show the specific path was effective to find characteristic patterns and directory display the corresponding behaviors.

Chapter 5

Conclusion

5.1 Summary

We proposed two visualization techniques for the trajectories of people flow data and eye-tracking scan-paths in this thesis. Chapter 1 discussed the usefulness of the trajectory data in society and explained the outline of the recording and analysis methods. Based on these backgrounds, we organized the issues related to flow line analysis, especially issues related to flow line visualization. Chapter 2 introduced existing research on trajectory analysis. We categorized existing studies according to the issues in the flow analysis described in Chapter 1 and compared them with our proposed techniques. In the next two chapters, we presented the details of the proposed techniques. Chapter 3 described the method to visualize people flow trajectories after classifying characteristic behaviors. We conducted experiments on human flow data acquired at the exhibition and the poster session and demonstrated that the proposed method can classify behaviors based on the passage points of traffic lines and the duration of stay. In Chapter 4, we proposed a method to visualize gaze trajectory data. We analyzed the characteristic of trajectories passing the AOIs with the strings symbolized based on the hierarchical AOIs. Furthermore, we also visualized differences in behavior among the participants by using a bar graph or selective visualization of the flow line shape.

We mentioned three problems with the analysis of the trajectory dataset in Chapter 1.

As compression of data, we adopted conversion from trajectories to strings. Especially, the compression rate got high on the people flow data. This process succeeded to preserve characteristic parts of trajectories. As pattern extraction, we applied different methods such as weighted Levenshtein distances, clustering, and N-gram. We found various useful features like common behaviors, long-staying or gazing at particular places. On the third problem, we used a direct drawing of trajectories first, then added supporting visualization to display characteristic data elements. These methods were effective to visualize both particular shapes and positions of trajectories and detailed useful features in a single view.

As future work, we would like to extend the proposed techniques so that we can apply other types of trajectories and work with more various fields.

5.2 Future Outlook on Trajectory Analysis

Finally, we discuss the future prospects for research and practical applications of trajectory analysis. We suppose the following trends will be more important and we will need to address them.

- Increasing the size and complexity of trajectory data to be analyzed.
- Promotion of a detailed analysis combining trajectory data with other data.
- Increasing the use of simulated trajectory data.

First, the scale of the data is expected to diversify larger in the future. For example, in the target traffic data, the number of moving objects and the lengths of measured periods will be expanded due to the active traffic network, logistics, and the flow of tourists in urban areas. Therefore, techniques on a more efficient trajectory analysis will be necessary. It will be particularly desirable to develop techniques that can not only select important trajectories from a large number of flow line groups but also directly extract knowledge about the meaning of the movement extracted from the trajectory datasets. In addition, user interfaces for visualization will be important. More flexible selection and adjustment of visualization contents should be possible by applying the latest technologies for interactive systems such as Virtual Reality (VR) and Powerwall. We need to develop visualization methods using large devices, apart from existing visualization techniques available on relatively small devices such as personal computers and tablet terminals. Second, the analysis integrating other types of data in addition to trajectory data may be more effective. Other relevant data include location information and comments transmitted on Social Network Service (SNS), image contents, face images captured by cameras, and voice recordings. Several past studies have addressed analysis and visualization that focuses on the movement information itself, such as using only trajectory data or performing classification based on the position and speed of traffic lines. On the other hand, in recent years, there have been studies that presuppose the use of additional data in addition to movement information [64]. Moreover, opportunities for individuals to contribute to the accumulation of data is also increasing through the daily use of location information and the spread of SNS. Such a composite analysis will be more practical for a more detailed analysis. One of the major issues is the development of simultaneously visualizing the state of a moving object and the shape of a flow line. For example, conceivable targets for visualization are "Where the pedestrian is aiming" and "While he is stopped, what he is paying attention to" along with the pedestrian's trajectory. Visualizing such the inner surfaces of such pedestrians lead to understand what actions have what kinds of meanings and to apply it for prediction and guidance of actions. Third, the analysis of simulated flow line data will be more important. As mentioned in Chapter 1.1, the analysis of past trajectory data is often intended to predict and control future movements or behaviors. Advances in research on visualization of flow line data and behavior analysis have enabled the simulation of trajectory data in complex environments. The collection of a large-scale measurement dataset is often difficult because the recording trajectory data takes time and often needs careful privacy management. Simulation is more effective for rapid flow analysis and prediction of the movement in various environments. It is therefore essential to develop a visualization method that can efficiently compare the trajectory simulation datasets under different conditions.

Acknowledgements

I really appreciate for my supervisor, Professor Takayuki Itoh's polite supervising. He is my first and great role model as a researcher, and I learned so many things about working on research. I also would like to thank Professor Ichiro Kobayashi, Professor Masato Oguchi, Professor Itiro Siio, and Professor Mariko Hagita for giving me precious many-sided advice as vice-chairs.

Doctor Masaki Onishi at National Institute of Advanced Industrial Science and Technology kindly provided me people flow dataset, and coached studiously thorough meetings. I also would like to thank him for these matters. Doctor Chiemi Watanabe at University of Tsukuba provided program of UniversalSAX which is essential in this study. I would like to thank her so much. Also, I am thankful to Doctor Masahiro Takatsuka at The University of Sydney for supervising me during my studying abroad for two months.

I gratitude Professor Daniel Weiskopf and Mr. Nils Rodrigues at University of Stuttgart, as kindly accepted me for two months and gave me useful advice on analysis of eye tracking data.

Finally, I would like to thank all members of Itoh laboratory, and my family who warmly supported me.

Bibliography

Publication

Journal Papers

[1] Y. Miyagi, M. Onishi, C. Watanabe, T. Itoh, M. Takatsuka, Classification and Visualization for Symbolic People Flow Data, Journal of Visual Languages and Computing, Vol. 43, pp. 91-102, December 2017.

[2] 宮城, 大西, 渡辺, 伊藤, 高塚, 記号化された歩行経路群のパターン抽出と可視化, 可視化情報学会論文集, Vol. 38, No. 3, pp. 7-16, 2018 年 3 月.

[3] 宮城, Rodrigues, Weiskopf, 伊藤, N-gram による視線軌跡からの AOI 推移パターン抽出と軌跡形状の可視化, 芸術科学会論文誌, Vol. 18, No. 4, pp. 125-133, 2019 年 11 月 (NICOGRAPH2019 最優秀論文賞).

Conference Talks in English

[4] Y. Miyagi, M. Onishi, C. Watanabe, T. Itoh, M. Takatsuka: Feature Extraction and Visualization for Symbolic People Flow Data, IEEE Pacific Visualization Symposium, Poster Session, April 2016.

[5] Y. Miyagi, M. Onishi, C. Watanabe, T. Itoh, M. Takatsuka: Feature Extraction and Visualization for Symbolic People Flow Data, 20th International Conference on Information Visualisation (IV2016), July 2016 (The Best Paper Award).

[6] Y. Miyagi, M. Onishi, C. Watanabe, T. Itoh, M. Takatsuka: Classification and Visualization for Symbolic People Flow Data Preserving Way Points and Staying Times, IEEE Pacific Visualization Symposium, Poster Session, April 2017.

[7] Y. Miyagi, N. Rodrigues, D. Weiskopf, T. Itoh: Visualization of Eye Tracking Data Using a Directed Graph with Edge Bundling, IEEE Pacific Visualization Symposium, Poster Session, April 2018.

[8] Y. Miyagi, N. Rodrigues, D. Weiskopf, T. Itoh: Analysis and Graph Visualization of Eye Tracking Data with a Static Stimulus, IEEE VIS, Poster Session, October 2018.

[9] Y. Miyagi, N. Rodrigues, D. Weiskopf, T. Itoh: Pattern Extraction and Visualization of Eye Tracking Scan Paths on Hierarchical AOIs, IEEE Pacific Visualization Symposium, Poster Session, April 2019.

Conference Talks in Japanese

[10] 宮城, UniversalSAX を用いた人流可視化, 第 14 回 ビジュアル情報処理研究合宿, 2014 年 9 月.

[11] 宮城, 大西, 渡辺, 伊藤 UniversalSAX を用いた人流可視化, 第 5 回情報可視化研究合宿, 2014 年 9 月.

[12] 宮城, 大西, 渡辺, 伊藤, 文字列化による人流データの圧縮と可視化, 第 7 回データ工学と情報マネジメントに関するフォーラム (DEIM 2015), F8-5, 2015 年 3 月 (学生プレゼンテーション賞).

[13] 宮城, 大西, 渡辺, 伊藤, UniversalSAX を用いた人流可視化, 情報処理学会第 77 回全国大会, 4N-08, 2015 年 3 月 (学生奨励賞, 大会奨励賞).

[14] 宮城, 大西, 渡辺, 伊藤, 記号化による人流データの圧縮と可視化, 第 18 回画像の認識・理解シンポジウム, SS5-9, 2015 年 7 月.

[15] 宮城, 大西, 渡辺, 伊藤記号化による人流データの圧縮と可視化, 第 15 回 ビジュアル情報処理研究合宿, 2015 年 9 月.

[16] 宮城, 大西, 渡辺, 伊藤, 高塚記号化による人流データの圧縮と可視化, 第 6 回情報可視化研究合宿, 2015 年 9 月.

[17] 宮城, 大西, 渡辺, 高塚, 伊藤, 記号列化した人流データからの特徴抽出と可視化, 第 8 回データ工学と情報マネジメントに関するフォーラム (DEIM 2016), F3-5, 2016 年 3 月 (学生プレゼンテーション賞).

[18] 宮城, 大西, 渡辺, 伊藤, 高塚, 記号列化した人流データからの特徴抽出と可視化, 情報処理学会第 78 回全国大会, 4N-05, 2016 年 3 月 (学生奨励賞, 大会優秀賞).

[19] 宮城, 大西, 渡辺, 伊藤, 高塚記号化された歩行経路群のパターン抽出と可視化, 第 7 回情報可視化研究合宿, 2016 年 9 月.

[20] 宮城, 大西, 渡辺, 伊藤, 高塚, 通過点と滞在時間情報を保持し記号列化した歩行者動線の分類と可視化, 情報処理学会第 79 回全国大会, 4P-02, 2017 年 3 月.

[21] 宮城, 大西, 渡辺, 伊藤, 高塚, 通過点と滞在時間情報を保持し記号列化した歩行者動線の分類と可視化, 第 45 回可視化情報シンポジウム, B205, 2017 年 7 月.

[22] 宮城, 大西, 渡辺, 伊藤, 高塚通過点と滞在時間情報を保持し記号列化した歩行者動線の分類と可視化, 第 8 回情報可視化研究合宿, 2017 年 9 月.

[23] 宮城, 大西, 渡辺, 伊藤, 高塚, 室内で撮影した歩行者軌跡を対象としたパターン分析と可視化, 第 1 回ビジュアリゼーションワークショップ, ポスターセッション, 2018 年 3 月.

[24] 宮城, 大西, 渡辺, 伊藤, 高塚, RGB-D カメラで記録した歩行者軌跡のパター ン分析と可視化, 第 46 回可視化情報シンポジウム, 2018 年 7 月 (ベストプレゼンテーション賞).

[25] 宮城, Weiskopf, 伊藤, 階層的な AOI を用いた視線軌跡のパターン分析と可視化, 第 2 回ビジュアリゼーションワークショップ, ポスターセッション, 2019 年 3 月.

[26] 宮城, Rodrigues, Weiskopf, 伊藤, 力指向グラフによる視線推移パターンの可視化と比較, 第 47 回可視化情報シンポジウム, 2019 年 7 月.

[27] 宮城, Rodrigues, Weiskopf, 伊藤 N-gram による視線軌跡からの AOI 推移パターン抽出と形状可視化, 第 10 回情報可視化研究合宿, 2019 年 9 月.

Bibliography

[28] Y. Zheng, Trajectory Data Mining: An Overview, ACM Transactions on Intelligent Systems and Technology, 6(3), 41, 2015.

[29] 気象庁, 台風情報の種類と表現方法, https://www.jma.go.jp/jma/kishou/know/typhoon/7-1.html

[30] N. Andrienko, G. Andrienko, Visual analytics of movement: An overview of methods, tools and procedures, Information Visualization, 12(1), 3-24, 2013.

[31] T. Yamashita, S. Soeda, M. Onishi, I. Yoda, I. Noda, Approach to Safety and Secure Evacuation with Sensor Data Mining, The journal of the Institute of Electronics, Information and Communication Engineerss, 94(4), 294-298, 2011.

[32] A. Li, Z. Chen, Representative Scanpath Identification for Group Viewing Pattern Analysis, Journal of Eye Movement Research, 11(6), 179-194, 2018.

[33] S. Konomi, H. Ishizuka, M. Iwai, J. Miyazaki, K. Sezaki, Y. Tobe, I-Tree: A Spatial Time-series Indexing Mechanism for Supporting Integrated Retrieval of Sensing Data, IPSJ Transactions on Databases, 4(1), 26-39, 2011.

[34] T. Oates, A. P. Boedihardjo, J. Lin, C. Chen, S. Frankenstein, S. Gandhi, Motif Discovery in Spatial Trajectories using Grammar Inference, ACM International

Conference on Information and Knowledge Management (CIKM 2013), 1465-1468, 2013.

[35] Y. Ohata, A. Ohno, T. Yamasaki, K. Tokiwa, An analysis of shopping behavior pattern and purchase amount in the inner areas of the sales floor in a retail store, Proceedings of the Forum on Information Technology 2015, 14(2) 297-302, 2015.

[36] K. Yada, String analysis technique for shopping path in a supermarket. Journal of Intelligent Information Systems, 36(3), 385-402, 2011.

[37] K. Teknomo, G. P. Gerilla, Pedestrian Static Trajectory Analysis of a Hypermarket, Proceedings of the Eastern Asia Society for Transportation Studies, 7, 2009.

[38] M. Burch, A. Kumar, K. Mueller, The Hierarchical Flow of Eye Movements, Proceedings of the 3rd Workshop on Eye Tracking and Visualization, 1-5, 2018.

[39] C. K. Yang, C. Wacharamanotham Alpscarf: Augmenting Scarf Plots for Exploring Temporal Gaze Patterns, Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, 1-6, 2018.

[40] N. Johnson, D. Hogg, Learning the Distribution of Object Trajectories for Event Recognition, Image and Vision Computing, 14(8), 609-615, 1996.

[41] F. Porikli, T. Haga, Event Detection by Eigenvector Decomposition Using Object and Frame Features, Computer Vision and Pattern Recognition Workshop, 114-114, 2004.

[42] N. Suzuki, K. Hirasawa, K. Tanaka, T. Kobayashi, Y. Sato, Y. Fujino, Detection of Abnormal Behavior and Patterns by Human Trajectories Analysis, The IEICE transactions on information and systems (Japanese edition), 91(6), 1550-1560, 2008.

[43] A. Asahara, A. Sato, K. Maruyama, Indoor and Outdoor Pedestrian Trajectory Analysis and the Experiment, IPSJ SIG technical reports, 135, V1-V8, 2009.

[44] M. Butenuth, F. Burkert, F. Schmidt, S. Hinz, D. Hartmann, A. Kneidl, A. Borrmann, B. Sirmacek, Integrating pedestrian simulation, tracking and event detection for crowd analysis , 2011 IEEE International Conference on Computer Vision Workshops, 150-157, 2011.

[45] G. Andrienko, N. Andrienko, S. Wrobel, Visual Analytics Tools for Analysis of Movement Data, ACM SIGKDD Explorations Newsletter, 9(2), 38-46, 2007.

[46] H. Yabushita, T. Itoh, Summarization and Visualization of Pedestrian Tracking Data, 15th International Conference on Information Visualisation (IV2011), 537-542, 2011.

[47] A. Fukute, M. Onishi, T. Itoh, A Linked Visualization of Trajectory and Flow Quantity to Support Analysis of People Flow, 17th International Conference on Information Visualisation (IV2013), 561-567, 2013.

[48] H. Guo, Z. Wang, B. Yu, H. Zhao, X. Yuan, TripVista: Triple Perspective Visual Trajectory Analytics and its application on microscopic traffic data at a road intersection, IEEE Pacific Visualization Symposium, 163-170, 2011.

[49] Z. Wang, H. Guo, X. Yuan, H. Liu, H. Zhang, Discovery Exhibition: Visual Analysis on Traffic Trajectory Data, Poster Proceedings of IEEE Visualization Discovery Exhibition, 2011.

[50] Y. Guo, Q. Xu, X. Li, X. Luo, M. Sbert, A New Scheme for Trajectory Visualization, 18th International Conference on Information Visualisation (IV2014), 40-45, 2014.

[51] Z. Wang, T. Ye, M. Lu, X. Yuan, H. Qu, J. Yuan, Q. Wu, Visual Exploration of Sparse Traffic Trajectory Data, IEEE Transactions on Visualization and Computer Graphics, 20(12), 1813-1822, 2014.

[52] S. Al-Dohuki, F. Kamw, Y. Zhao, C. Ma, Y. Wu, J. Yang, X. Ye, F. Wang, X. Li, W. Chen, SemanticTraj: A New Approach to Interacting with Massive Taxi Trajectories. IEEE Transactions On Visualization and Computer Graphics, 23(1), 11-20, 2017.

[53] R. Krueger, D. Thom, M. Woerner, H. Bosch, and T. Ertl, TrajectoryLenses - A Set-based Filtering and Exploration Technique for Long-term Trajectory Data, Computer Graphics Forum (proceedings of The Eurographics Conference on Visualization), 32 (3), 451-460, 2013.

[54] R. Krueger, S. Koch, T. Ertl, Saccadelenses: interactive exploratory filtering of eye tracking trajectories, 2016 IEEE Second Workshop on Eye Tracking and Visualization, 31-34, 2016.

[55] T. Kuebler W. Fuhl, R. Rosenberg, E. Kasneci, Novel Methods for Analysis and Visualization of Saccade Trajectories, Computer Vision – ECCV 2016 Workshops, 783-797, 2016.

[56] M. Ramin, A Streakline Representation of Flow in Crowded Scenes, Computer Vision – ECCV 2010, 439-452, 2010.

[57] N. Rodrigues, R. Netzel, J. Spalink, D. Weiskopf, Multiscale scanpath visualization and filtering, Proceedings of the 3rd Workshop on Eye Tracking and Visualization, 1-5, 2018.

[58] D. Rudi, P. Kiefer, M. Raubal Visualizing pilot eye movements for flight instructors, Proceedings of the 3rd Workshop on Eye Tracking and Visualization, 1-5, 2018

[59] S. Okazaki, S. Matsushita, A study of simulation model for pedestrian movement with evacuation and queuing, International Conference on Engineering for Crowd Safety, 271, 1993.

[60] L. Chittaro, L. Ieronutti, A Visual Tool for Tracing Users' Behavior in Virtual Environments, Proceedings of the working conference on Advanced visual interfaces, 40-47, 2004.

[61] R. Krueger, G. Sun, F. Beck, R. Liang, T. Ertl, TravelDiff: Visual comparison analytics for massive movement patterns derived from Twitter, IEEE Pacific Visualization Symposium, 176-183, 2016.

[62] S. Gupta, M. Dumas, M. J. McGuffin,m T. Kapler, MovementSlicer: Better Gantt charts for visualizing behaviors and meetings in movement data, IEEE Pacific Visualization Symposium, 168-175, 2016.

[63] N. H. Thach, E. Suzuki, A Symbolic Representation for Trajectory Data, The Japanese Society Artificial Intelligence, 1A2-2, 2010.

[64] X. Alameda-Pineda, Y. Yan, E. Ricci, O. Lanx, N. Sebe, Analyzing Free-standing Conversational Groups: A Multimodal Approach, the 23rd ACM international conference, 5-14, 2015.

[65] C. M. Privitera, L. W. Stark, Algorithms for defining visual regions-of-interest: comparison with eye fixations, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(9), 970-982, 2000.

[66] P. Muthumanickam, K. Vrotsou, A. Nordman, J. Johansson, M. Cooper, Identification of Temporally Varying Areas of Interest in Long Duration Eye Tracking Data Sets, IEEE Transactions on Visualization and Computer Graphics, 25, 87-97, 2018.

[67] T. Blascheck, M. Schweizer , F. Beck, T. Ertl, Visual Comparison of Eye Movement Patterns, Comput. Graph. Forum, 36(3), 87-97, 2017.

[68] M. Burch, A. Kumar, K. Mueller, D. Weiskopf, Color bands: visualizing dynamic eye movement patterns, Proceedings of the 2nd Workshop on Eye Tracking and Visualization, 40-44, 2016.

[69] M. Burch, K. Kurzhals, N. Kleinhans, D. Weiskopf EyeMSA: Exploring Eye Movement Data with Pairwise and Multiple Sequence Alignment, Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, 1-5, 2018.

[70] Y. Gu, C. Wang, R. Bixler, S. D'Mello, ETGraph: A Graph-Based Approach for Visual Analytics of Eye-Tracking Data, Computers & Graphics, 62, 1–14, 2017.

[71] V. Peysakhovich, C. Hurter, Scanpath visualization and comparison using visual aggregation techniques, Journal of Eye Movement Research, 10(5), 1-14, 2018.

[72] M. Onishi T. Yamashita T. Hoshikawa, K. Sato, Transmission of Knowledge for Evacuation Drill using Pedestrian Tracking and Simulation - Example of Opera Concert with Evacuation Drill in New National Theatre, The Japanese Society for Artificial Intelligence Special Interest Group on Knowledge and Skills Transfer, SIG-KST-026-06, 2015.

[73] A. Onishi, C. Watanabe, Universal SAX: Applied SAX to the Multidimensional Time Series Data Using the Space Filling Curve, DBSJ Journal, 11(1), 43-48, 2012.

[74] M. Onishi, I. Yoda, Dynamic Trajectory Extraction from Stereo Vision Using Fuzzy Clustering, The transactions of the Institute of Electrical Engineers of Japan. C, A publication of Electronics, Information and System Society 128, 9, 1438-1446, 2008.

[75] H. Hikita, M. Onishi, A fault tolerant autonomous distributed system for pedestrian measurement using multiple cameras, CVIM, 2017-CVIM-205(26), 1-6, 2017.

[76] K. Yarimizu, "Weighted Levenshtein Distance" for Calculating the Distance between Words, Meikai Japanese language journal (18), 179-194, 2013.

[77] S. Eraslan, Y. Yesilada, S. Harper, Scanpath Trend Analysis on Web Pages: Clustering Eye Tracking Scanpaths, ACM Trans. Web, 10(4), 1-35, 2016.

[78] U. Obaidellah, M. Al Haek, P. C-H. Cheng, A Survey on the Usage of Eye-Tracking in Computer Programming, ACM Computing Surveys, 51(1), 58, 2018.

[79] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, T. Ertl, Visualization of Eye Tracking Data: A Taxonomy and Survey, Computer Graphics Forum, 36(8), 260-284, 2017.

[80] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, T. Ertl, State-of-the-Art of Visualization for Eye Tracking Data, Eurographics Conference on Visualization, 2014.

[81] K. Holmqvist, J. Holsanova, M. Barthelson, D. Lundqvist, Reading or scanning? A study of newspaper and net paper reading, The mind's eye: cognitive and applied aspects of eye movement research, 657-670, 2003.

[82] K. Kurzhals, M. Burch, T. Blascheck, G. Andrienko, N. Andrienko, D. Weiskopf A task-based view on the visual analysis of eye-tracking data, Mathematics and Visualization, 3-22, 2017.

[83] C.E. Shannon, Prediction and entropy of printed English, Bell Syst. Tech. J., 30, 50-64, 1951.

[84] T. Blascheck, K. Kurzhals, M. Raschke, S. Strohmaier, D. Weiskopf, T. Ertl AOI hierarchies for visual exploration of fixation sequences, Proceedings of the Ninth

Biennial ACM Symposium on Eye Tracking Research & Applications, 111-118, 2016.

[85] D. Holten, J. J. van Wijk Force-directed edge bundling for graph visualization, Eurographics/ IEEE-VGTC Symposium on Visualization 2009, 983-998, 2009.

[86] M. Tennekes, E. de Jonge, Tree Colors: Color Schemes for Tree-Structured Data, Ieee Transactions on Visualization and Computer Graphics, 20(12) 2072-2081, 2014.

[87] M. Reani, N. Peek, C. Jay An Investigation of the Effects of N-gram Length in Scanpath Analysis for Eye-tracking Research, Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, 1-8, 2018.

Advanced Science,
Graduate School of Humanities and Sciences,
Ochanomizu University,
Yuri Miyagi