Classification and Visualization for Symbolic People Flow Data

Yuri Miyagi[a], Masaki Onishi[b], Chiemi Watanabe[c], Takayuki Itoh[a], Masahiro Takatsuka[d]

[a] *Ochanomizu University, Japan.*
[b] *National Institute of Advanced Industrial Science and Technology, Japan.*
[c] *University of Tsukuba, Japan.*
[d] *The University of Sydney, Australia.*

Abstract

People flow information brings us useful knowledge in various industrial and social fields including traffic, disaster prevention, and marketing. However, it is still an open problem to develop effective people flow analysis techniques. We considered compression and data mining techniques are especially important for analysis and visualization of large-scale people flow datasets. This paper presents a visualization method for large-scale people flow dataset featuring compression and data mining techniques. This method firstly compresses the people flow datasets using UniversalSAX, an extended method of SAX (Symbolic Aggregate Approximation). Next, we apply algorithms inspired by natural language processing to extract movement patterns and classify walking routes. After this process, users can interactively observe trajectories and extracted features such as congestions and popular walking routes using a visualization tool. We had experiments of classifying and visualizing walking routes using two types of people flow dataset recorded at an exhibition and a corridor applying our method. The results allow us to discover characteristic movements such as stopping in front of particular exhibits, or persons who passed same places but walked at different speeds.

Keywords: People flow, Symbolic trajectory, Movement pattern, Graph visualization

1. Introduction

Security cameras record many pictures of pedestrians every day. It is worth analyzing the pictures to discover movement patterns of the people, since we can get useful information to solve many social problems. For example, we can establish better evacuation routes, find causes of traffic jams, and come up with product displays that attract more customers, by interpreting the discovered movement patterns. However, it may be difficult to understand overall trends and find important knowledge from large amount of people flow datasets in a short time. For example, sizes of our datasets containing people flow during several weeks may be more than 10 GB. We can preserve such datasets; however, it is not easy to observe or edit the datasets shortly (e.g. within a couple of minutes). It is still an open problem to develop compression and data mining techniques and visualize the movement patterns discovered from large-scale people flow datasets.

In this paper, we propose a visualization technique for large-scale people flow datasets. The technique firstly compresses the people flow data applying UniversalSAX [1], an extended implementation of SAX (Symbolic Aggregate Approximation). It converts the numeric position data to sets of smaller sizes of character datasets. It then applies natural language processing algorithms to the character datasets to quickly discover typical movement patterns and classify them. Finally, the technique visualizes the people flow datasets focusing on the movement patterns. Users can observe people flow interactively using three types of views, overview view, detailed view and clustering view.

This paper shows case studies with real-world people flow datasets in an exhibition and a corridor, then discusses effectiveness of the presented tool. In the first case, we used a 669MB people flow dataset and compressed it to only 115KB. Then, we visualized characteristic points such as congestion and walking routes. In the second case, we applied clustering before visualizing trajectories. We compared results of clustering following different conditions.

The remainder of this paper is organized as follows. In Section 2, we introduce related work on analysis of trajectories of pedestrians and other moving things. Section 3 presents the detail of the proposed technique. Section 4 introduces case studies using trajectory datasets recorded at an exhibition and a corridor. Section 5 contains a discussion of visualizations using the technique. Section 6 summarizes this paper.

2. Related Work

This section introduces existing studies to analyze people flow and trajectories of other moving things. Visualizing trajectories using graphs is major technique. Höcker [2] et al. proposed a graph structure to represent paths of walkers and an algorithm which searches for particular trajectories. Çetinkaya [3] et al. compared 4 types of graph visualization techniques using datasets of locations of states. These studies did not focus

Email addresses: miyagi@itolab.is.ocha.ac.jp (Yuri Miyagi), onishi@ni.aist.go.jp (Masaki Onishi), chiemi@cs.tsukuba.ac.jp (Chiemi Watanabe), itot@itolab.is.ocha.ac.jp (Takayuki Itoh), masa.takatsuka@sydney.edu.au (Masahiro Takatsuka)

on summarization of trajectories, or development of interactive visualization tools.

Not only using graphs, there have been many existing methods which classify and visualize spatio-temporal people flow data recorded as real values. Following studies commonly visualized trajectories or their features on maps. Andrienko et al. [4] collected datasets from wide range using GPS, and analyzed properties of various moving objects, using both of drawing specific trajectories and bar charts on maps. On the contrary, our methods supposes that datasets are collected by cameras in small area like one floor in a shop. Andrienko et al. [5] also proposed interactive clustering method for trajectories. Furthermore, they visualized popular walking patterns on maps. Guo et al. [6] developed a composite visualization tool to analyze patterns of various objects such as pedestrians, bicycles, and cars. They adopted not only direct drawing of trajectories on maps, but also other visualization methods including piled polyline charts, scatterplots, and parallel coordinates plots. These techniques do not apply data compression techniques for trajectory datasets. Wang et al. [7] also extended the technique presented in [6], and visualized moving patterns of taxis in wider regions. Krueger et al. [8] presented an improved visualization system for chronologically GPS datasets. The main feature of this system is that users can move a circle looked like a lens, and focus on particular regions to observe detailed information such as speeds and directions. Wang et al. [9] extracted and visualized features of automobiles passing at particular positions, applying datasets collected using many sensors on roads. Lu et al. [10] proposed TrajRank, a visualization system for trajectories for vehicles like taxies. They separated trajectories to segments and calculated their ranks to suggest characteristic travel behaviors.

Other techniques introduced below commonly visualized trajectories on pictures of camera views. Mehran et al. [11] proposed a technique using streamline visualization and abnormality detection methods. They claimed that streamlines are superior to path lines. Yabushita et al. [12] proposed a technique which 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. Ko et al. [13] focused on angles of trajectories of walking people, and extracted irregular movings such as a zigzag. They also tried to transform the trajectories onto pictures of camera views, and visualized them in order for users to understand the angles of the trajectories. Fukute et al. [14] 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. Andrienko et al. [15] proposed "trajectory wall" as an extension of space-time cubes. Users can grasp trajectories in a same cluster which have similarities about routes. Guo et al. [16] classified walkers' trajectories according to their speed and direction. They also developed a system to visualize important trajectories using meaningful colors based on HSV model. However, the tool does not support interactive trajectory selection for detail-on-demand visualization.

Following researches focused on observation of moving patterns, though analysis of trajectories is not exactly a main task. Gupta et al. [17] worked on to visualize relationships among small number of pedestrians. They did not visualize particular shapes of trajectories, but applied a gantt chart and visualized places where people stayed. This representation is especially useful for users because they can find places where multiple persons stayed at the same time. Krueger et al. [18] presented an improvement of another visualization system named TravelDiff. The system consumes messages in Twitter instead of applying datasets of trajectories, and visualizes movement patterns and crowded places. They tried to visualize three types of datasets, for pedestrians, taxies, and airplanes, as graphs and heatmaps.

AI-Dohuki et al. [19] developed a visualization system for trajectories of taxies. The system can visualize not only statistical traffic information but also messages which answer to questions input by users. For example, users can select particular streets or time periods, and visualize related datasets.

Also, there have been many techniques on compression and pattern discovery techniques for people flow datasets. Teknomo et al. [20] 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 analyzed populations and length of walking times. Thack et al. [21] converted the spatiotemporal trajectories 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. Oates et al. [22] 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. However, a common issue in these studies is that they displayed all recorded trajectories as is. Their visualization results are therefore so complex that it may be difficult to find important routes or places. Contrary to these methods, our technique firstly extracts typical movement patterns and then visualizes them noticeably. Yada [23] analyzed movements of customers in a supermarket, and searched for popular sections. He 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.

3. Presented Visualization Tool

This section presents the processing flow and detailed description of technical components of the presented technique. Figure 1 shows the whole processing flow of our proposed technique. After the preprocessing including trajectory recording, conversion to strings and feature extraction, users can interact with the visualization.

Section 3.1 defines the people flow datasets. Section 3.2 describes a technique to convert the trajectories into sets of char-

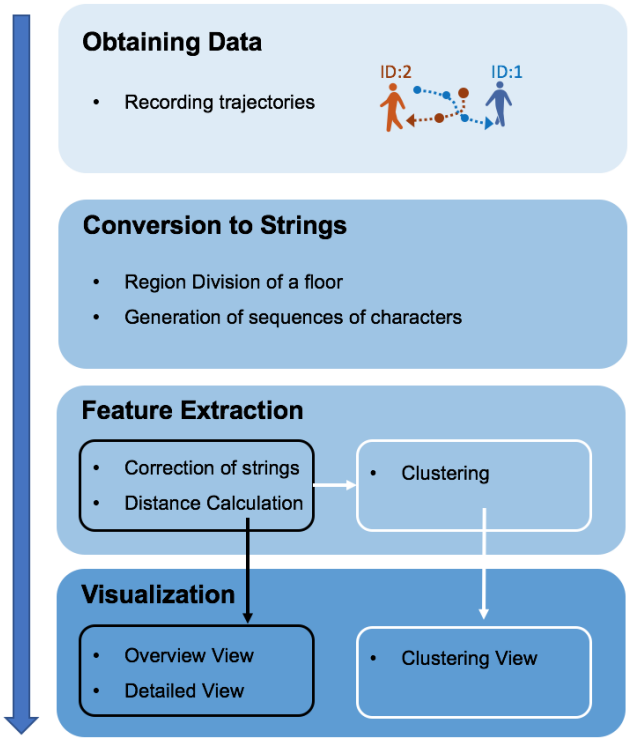

Figure 1: Processing flow of the proposed technique.

acters, following by the discovery of typical movement features described in Section 3.3. Lastly, we describe a visualization technique which emphatically displays the features in Section 3.4.

3.1. Recording People Flow Data

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

- Time at which 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 proposed in [24]. We assigned a particular ID to each walker, and measured positions of heads of pedestrians every dozens of milliseconds. Xtion can measure positions of pedestrians in a real three dimensional space; however, we adopted only two dimensional coordinates on floors.

3.2. Converting 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 2 illustrates this process. We apply UniversalSAX [1],

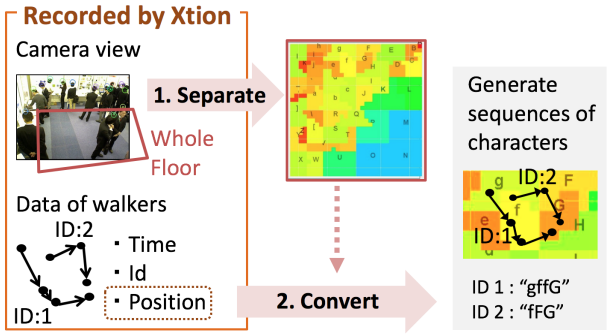

Figure 2: Flow of converting position values.

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 distances 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 region 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 to 2^q segments. Then, we generate a Hilbert curve that passes every block once. A block is a smallest square which composes regions. Each block is assigned a sequential number that indicates the order which the Hilbert curve passes. Unlike other space-filling curves such as Z-ordering, 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 assigns 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 distribution of data items (i.e. walkers).

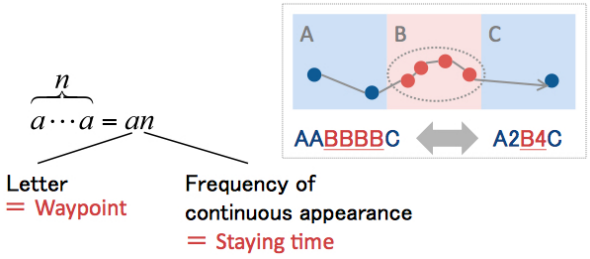

Figure 3: Applying run length encoding to a walking route.

Smaller number of blocks are assigned to regions that many data items belong to. On the contrary, larger number of blocks are assigned to a region if few data items belong to. There are two reasons to adjust areas of regions according to density of data items, not regularly separating the space in a grid pattern. One is to keep fine resolution to preserve more accurate spatial information in high density regions such as crowded places. Crowded place is usually worth paying attention to understand walking patterns of people. The other is to avoid lack of characters to assign as names of regions by too fine separation at sparse regions. At the same time, this process saves a table of distances among all regions conveniently. 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. The following set G is a group of characters for names of regions.

$$G = \{g_1\ g_2\ \ldots\ g_z\} \tag{1}$$

g_i ($i = 1, 2, \ldots, z$) are characters as names of regions, selected from 'A'-'Z', 'a'-'z', and some symbols such as '\', '[', and ']' in this order. Then, the process chooses l letters as s_j ($j = 1, 2, \ldots, l$) from g_i allowing duplication in order to represent a walking route. The selected s_j are chronologically ordered, and compose one string S_k ($k = 1, 2, \ldots, p$) which mean a walking route. p is the number of recorded walkers. Then we generate P as a collections of walking routes.

$$P = \{S_1\ S_2,\ \ldots\ S_p\} \tag{2}$$

The sequence of characters usually construct much smaller datasets comparing with the original people flow datasets. To compress the datasets further, we apply the Run-Length encoding to the sequences of characters, as shown in Figure 3. 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 which include repetitious appearance of any letters; 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 stay for a while, by searching for continuously appearing characters.

Above process is deeply related to complexity and scalability of our technique. We suppose that one character corresponds to a name of a region. The maximum number of regions are approximately 60 which is a total of available alphabets and other characters in our implementation. Users have to apply region division and generation of graphs to datasets in each Xtion which can record about 5 square meters, if they analyze wider areas at the same time.

3.3. Extracting Features of People Flow

We extract features in the sequences of characters to understand property of people flow datasets. In this study, we focused on following features to find.

- Congestions.
- Changes of populations in different time periods.
- Representative walking routes.

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.

Before calculating distances, we correct abnormally long strings generated by the people flow conversion process described in Section 3.2 to calculate appropriate distances. (See Figure 4.) Usually, Run-Length codes generated from movements covering wide 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. It happens that 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 repetitions 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 regardless of a degree of difference, and set new staying times by accumulating staying times at each region. In this implementation, we correct parts which have repetitions of patterns "ABA" of way points (like "ABABA"). We do not adjust only one appearing (like "ABA"), since it is likely a real turning.

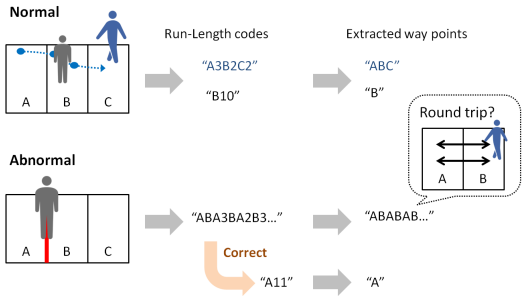

Figure 4: Correction of abnormal strings.

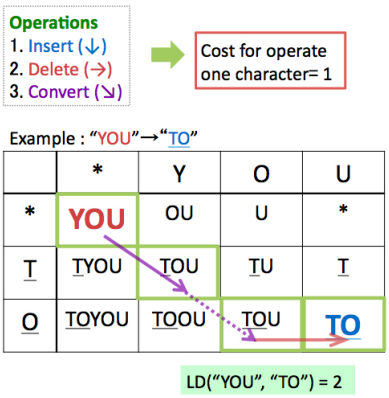

Figure 5: An example of calculating original Levenshtein Distance.

After this process, we calculate distances among S'_k. We adopt Levenshtein Distance (LD), one of popular definitions of distances between sequences of characters. An advantage of applying LD is that we can easily extend LD fitting to data, because the algorithm of LD calculation is simple. We suppose to use the distances in searching for specified trajectories and clustering. Users can control conditions of these processes by replacing the formula of the distance. We can calculate LD between every (S'_1, S'_2) pair, even if 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 $O(m_1 m_2)$. 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 5.)

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 'D' is close to 'B', and region

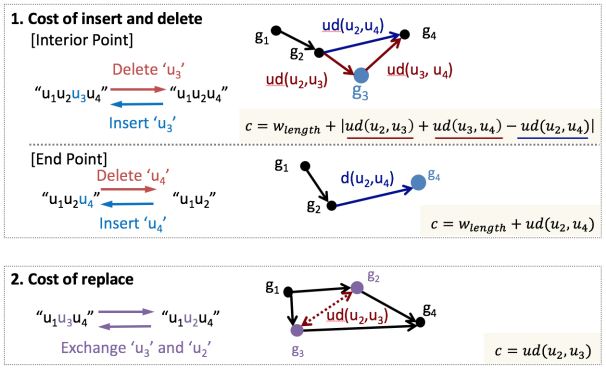

Figure 6: Costs of each operation.

'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) [25] 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 $u_{j'}$ $(j' = 1, 2, \ldots, l')$ and calculate sum of cost of operating one unit. We define $S'_k = (u_1, u_2, ..., u_{l'})$, where $t_{j'}$ denotes how many times $g_{j'}$ continuously appeared:

$$u_{j'} = \begin{cases} g_{j'} \ (t_{j'} = 1) \\ g_{j'} t_{j'} \ (otherwise) \end{cases} \tag{3}$$

We define costs of each operations as shown in Figure 6 using these definitions.

$ud(u_{j'_1}, u_{j'_2})$ denotes distances between two units defined as:

$$ud(u_{j'_1}, u_{j'_2}) = d(g_{j'_1}, g_{j'_2}) + w_{time} \sqrt{|t_{j'_1} - t_{j'_2}|} \tag{4}$$

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 3.2, to identify a distance between two regions $g_{j'_1}$ and $g_{j'_2}$ as $d(g_{j'_1}, g_{j'_2})$. $c(g_{j'_1}, g_{j'_2})$, which is a cost to operate using $g_{j'_1}$ and $g_{j'_2}$, increases when the region $g_{j'_1}$ and $g_{j'_2}$ are far away since $d(g_{j'_1}, g_{j'_2})$ gets higher value. 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. We treat the formula as inserting operation if the modified unit is at the top of S'_k. Conversely, we treat it as deleting operation if the unit is at the last. We can change in values of costs by not only which type of operation is applied but also to which letter we apply operations, by using $d(g_{j'_1}, g_{j'_2})$. 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 $u_{j'}$. 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.
- Lengths of the walking routes.
- Directions.
- Staying times.

We can classify trajectories or search for particular walking routes using results of these calculations, and show walking patterns such as typical walking routes 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.

We select K-medoids as a method of clustering. K-medoids is a non-hierarchical clustering method which is suitable for datasets containing different lengths of trajectories. 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. Users have to carry out the calculation of the matrix before the clustering only once. Hence, we have not developed user interfaces for the processes which we mentioned in section 3.1, 3.2, and 3.3. This process takes the largest part of the computation time in our technique, because the system compares all the possible pairs of trajectories.

3.4. Visualizing Walking Routes

Finally, we visualize people flow data as sequences of characters emphasizing its features. Briefly, users can observe three types of visualizations as shown in Figure 1. Overview and detailed views consume strings generated by the process introduced in Section 3.2. Clustering view needs a result of clustering which is mentioned in Section 3.3. This process displays the connections of the nodes based on the sequences of characters to represent the walking routes. Figure 7 shows an interface

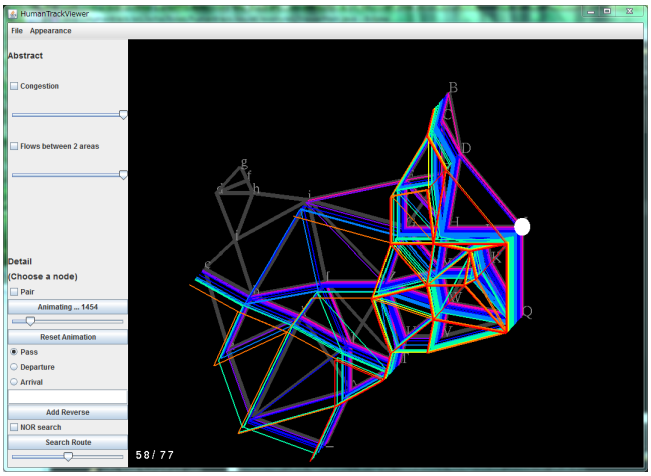

Figure 7: An interface of the visualization tool.

for the visualization. Users can select contents to visualize using a panel on the left side. At the same time, the system draws a graph on the right side. We generate nodes at the centers of the regions divided by the setup process described in Section 3.2. We propose 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. Users can find multiple regions that are worth observing by selectively displaying the following:

- Regions in 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 divided by area of each region, and 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. 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

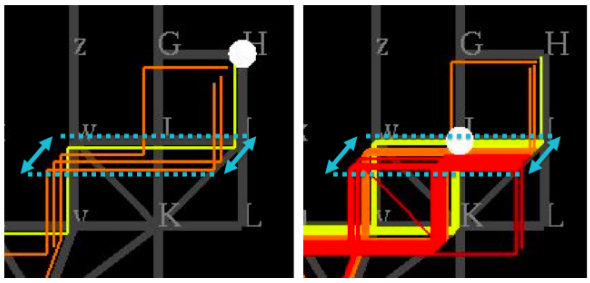

Figure 8: Visualizations of small number of segments (left) and bundles of similar segments(right).

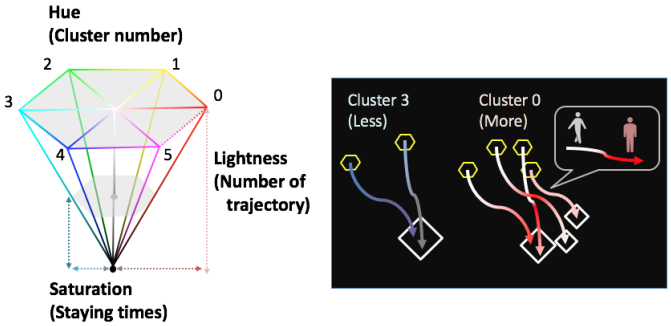

Figure 9: Selecting colors of trajectories.

draws each segment to directly compare them, as shown in Figure 8(left), if there are small number of segments to display. Otherwise, our implementation bundles similar segments so that they look like a single thick segment, as shown in Figure 8(right). This representation avoids too complex visualization results caused by huge number of scattered visible segments. 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. Keyword is a Run-Length code like "A2B4C", or a connection of region names like "ABC". 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.

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 as shown in Figure 9.

$$Hue = \frac{300\ v}{(V-1)} \tag{5}$$

$$Saturation = s_{min} + w_{sat}\frac{t_{gv}}{p_v} \tag{6}$$

$$Brightness = b_{min} + w_{bright}\ (255 - b_{min})\ \frac{p_v}{p} \tag{7}$$

In fomula (5), 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, for example cluster 0 gets red, and cluster 5 gets blue.

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} 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 rather 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. v_{sat} 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.

4. Experiments

4.1. Exhibition use case

We visualized people flow data recorded in an exhibition. There were two doorways in the room of the exhibition. 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 10. 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. The size of original dataset was 669MB. It was compressed into 115KB including 62KB sequences of characters and 53KB region information after applying UniversalSAX. We firstly visualized congestions to briefly understand the tendency of the people flow as shown in Figure 11, 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 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,

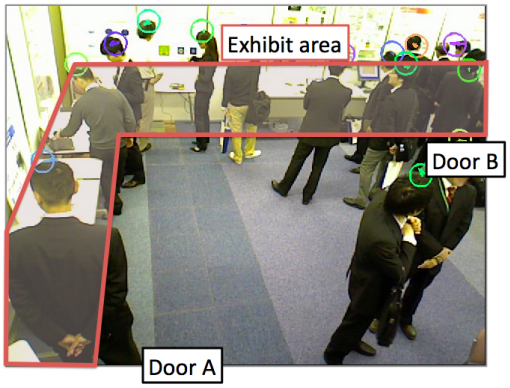

Figure 10: Camera view.

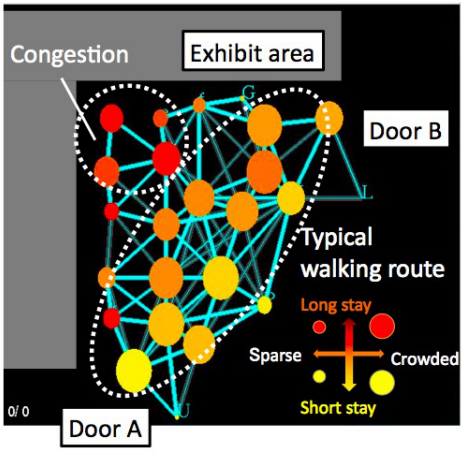

Figure 11: Visualization of congestions.

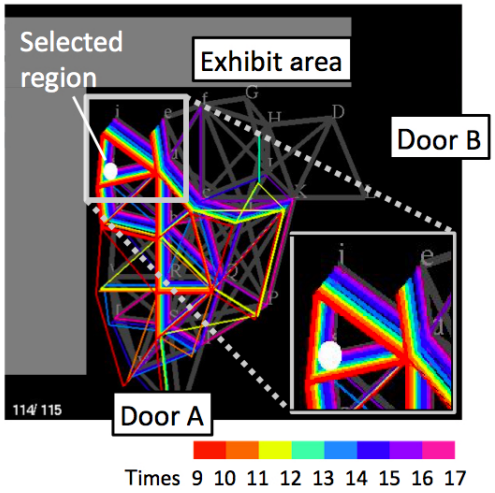

Figure 12: Visualizing walking routes passing through the selected region.

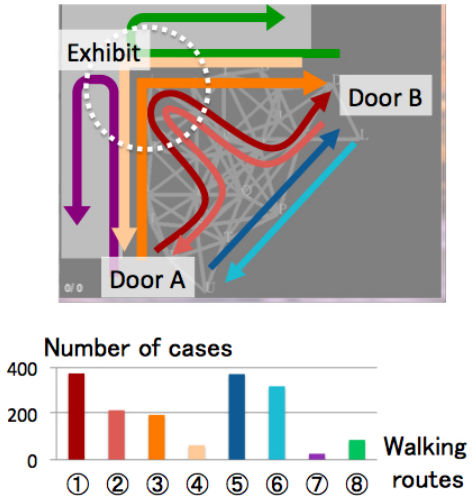

Figure 13: Result of counting numbers of people.

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 12. The system generated a graph shown in Figure 12. We added annotations regarding the room and colors of nodes. We colored trajectories based on HSV model, where higher hue values correspond to movements that occurred later. Each colors correspond to a 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, there were more dark blue segments corresponding to participants visited during 14:00 to 15:00, which illustrates more 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 13 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. We manually drew eight arrows which depict each routes as input. There were larger number of walkers (corresponding to routes 1, 3, and 5 in Figure 13) coming from door A, while fewer other walkers (corresponding to routes 2, 4, and 6 in Figure 13) 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 13) 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 13) turned buck after moving to the upper left exhibition. These indicate that we ought to devise arrangement of the exhibition to attract more visitors.

As introduced above, we successfully visualized various features of the people flow. There might be several lacks of fine information comparing with the original data, such as exact passing time and smooth geometry of the walking routes. On the other hand, we could quickly find the properties of people flow using only 62KB sequences of characters and 53KB region information, compressed from 669MB of the original data. This section illustrated that the presented visualization

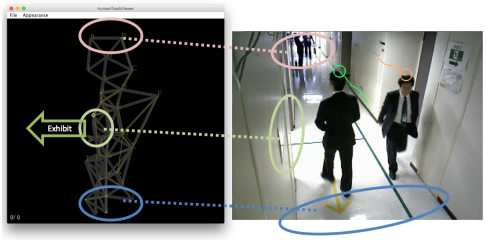

Figure 14: Camera view at a corridor and a result of visualizing passable routes.

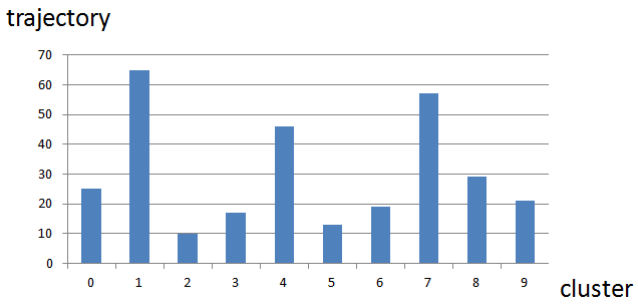

Figure 15: Generated clusters excluding staying time information.

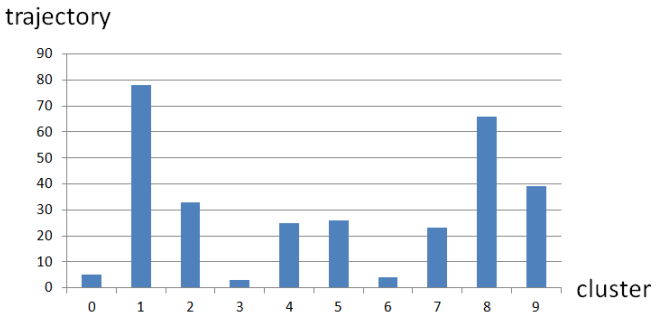

Figure 16: Generated clusters including staying time information.

technique succeeds to reduce the costs required to analyze people flow data.

4.2. Corridor use case

This subsection introduces results of another experiment using datasets recorded at a corridor next to the exhibition room, as shown in Figure 14. We compressed the dataset including 302 trajectories from 5MB to 8KB (99.84% Compression rate). We intentionally used this small dataset, because we repeatedly applied clustering while changing parameter values for distance calculation. Remark that users do not have to repeat clustering in practical usage. We applied clustering to visualize the datasets, changing in numbers of clusters, weights of features while calculating distances. We selected numbers of clusters as 5, 10, and 15. We found 5 was not an appropriate number to represent every known movement patterns. 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.

In this experiment, we verified efficiency of correction for abnormally long strings that we described in section 3.3. The system generated unnatural clusters from strings when we did not apply the correction process. The clusters included particular trajectories that are similarly very long but contain various letters. On the other hand, visualization results seemed to be more natural when we applied the corrected dataset, because such unnatural clusters disappeared. Figure 15 shows the distribution of trajectories while staying time information is not taken into account. Figure 16 shows the result when staying time information is taken into account.

Figure 17 is an example of visualizing results of clustering. Trajectories in these clusters did not disperse to many types of clusters. Clusters in the left column 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 representative departure and arrival points. On the other hand, when w_{times} = 5.0, many trajectories belong to clusters in the right column. Arrows depict how many trajectories belong to clusters in the left column moved to clusters in the right column. Observing the results of visualizing cluster 1 and 9 in the right column, shapes of clusters are similar, though they have different saturations. Whole saturations of cluster 1 is higher than those of cluster 9. That 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 ones 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 18 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 cluster. While setting w_{times} = 5.0, almost 80 percent of trajectories went to five clusters in Figure 18, 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.

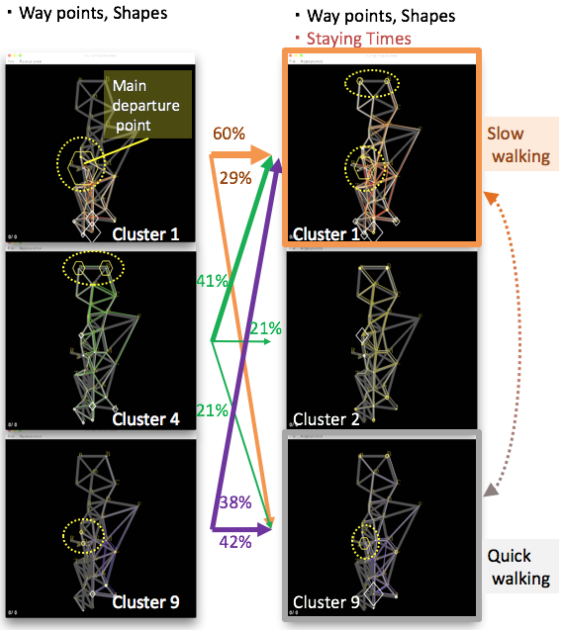

Figure 17: Comparing affiliations of trajectories in particular clusters.

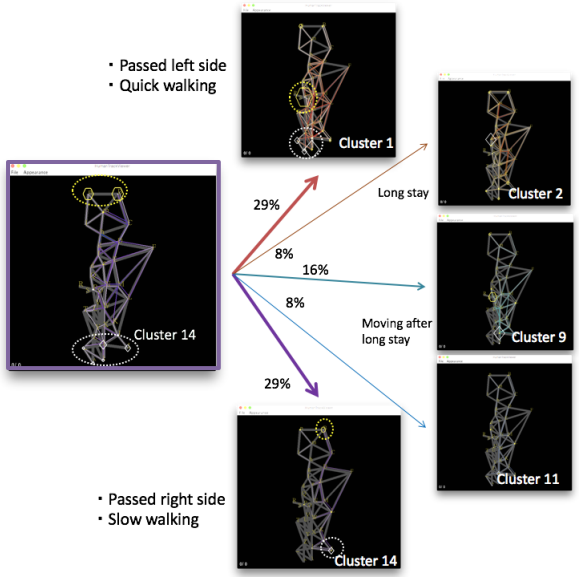

Figure 18: Dividing trajectories which passed almost same places by features of staying times.

5. Discussion

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 1. 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. 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.

On the other hand, we found remaining issues in our technique. First, the bundling is not exactly effective in this experiment. The system visualized trajectories along edges of a graph. However, we cannot easily and accurately understand directions of the walkers. Also, it is also difficult to correctly understand complex patterns, such as patterns generated by people who repeatedly passed the same place. Second, the rule of colors for the visualization of clustering results may cause a couple of faults. When the values get low, users cannot always distinguish the trajectories, since the trajectories get dark. In the current implementation, we defined representative walking patterns and crowded places as primary targets which we want to find. They correspond to highlighted trajectories which have high values of brightness or saturation. Therefore, we could find such trajectories or clusters quickly from the results. On the other hand, even though dark trajectories are not the primary targets of the visualization, they also depict useful information such as irregular movings. So we should develop better rules of colors to catch such trajectories quickly.

6. Conclusions

In this paper, we proposed a technique to visualize features of people flow data by converting real values of walking routes into sequences of characters. 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.

Future issues of this study are the following.

- 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.

References

[1] 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.

[2] M. Höcker, V. Berkhahn, A. Kneidl, A. Borrmann and, W. Klein: Graph-based approaches for simulating pedestrian dynamics in building models, eWork and eBusiness in Architecture, Engineering and Construction, 389-394, 2010.

[3] E. K. Çetinkaya, M. J. F. Alenazi, Y. Cheng, A. M. Peck, J. P. G. Sterbenz: A comparative analysis of geometric graph models for modelling backbone networks, Optical Switching and Networking, 14, 95-106, 2014.

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

[5] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, F. Giannotti: Interactive visual clustering of large collections of trajectories, 2009 IEEE Symposium on Visual Analytics Science and Technology, 3-10, 2009.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. Lu, Z. Wang, X. Yuan: TrajRank: Exploring travel behaviour on a route by trajectory ranking, IEEE Pacific Visualization Symposium, 311-318, 2015.

[11] R. Mehran, B. E. Moore, M. Shah: A Streakline Representation of Flow in Crowded Scenes, 11th European conference on computer vision conference on Computer vision, 439-452, 2010.

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

[13] J. G. Ko, J. H. Yoo: Rectified Trajectory Analysis Based Abnormal Loitering Detection for Video Surveillance, 1st International Conference on Artificial Intelligence, Modelling and Simulation, 289-293, 2013.

[14] 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.

[15] G. Andrienko, N. Andrienko, H. Schumann, C. Tominski: Visualization of Trajectory Attributes in Space-Time Cube and Trajectory Wall, Cartography from Pole to Pole - Lecture Notes in Geoinformation and Cartography, 157-163, 2014.

[16] 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.

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

[18] 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.

[19] 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.

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

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

[22] 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.
[23] K. Yada: String analysis technique for shopping path in a supermarket, Journal of Intelligent Information Systems, 36 (3), 385-402. 2011.
[24] 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.
[25] K. Yarimizu: "Weighted Levenshtein Distance" for Calculating the Distance between Words, Meikai Japanese language journal (18), 179-194, 2013.