Feature Extraction and Visualization for Symbolic People Flow Data

Yuri Miyagi[*] Masaki Onishi[†] Chiemi Watanabe[‡] Takayuki Itoh[*] Masahiro Takatsuka[§]
[*]Ochanomizu University [†]AIST [‡]University of Tsukuba [§]The University of Sydney
[*]{miyagi, itot}@itolab.is.ocha.ac.jp [†]onishi@ni.aist.go.jp
[‡]chiemi@cs.tsukuba.ac.jp [§]masa.takatsuka@sydney.edu.au

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 suppose compression and data mining techniques are especially important for analysis and visualization of large-scale people flow datasets. This paper presents a visualization tool for large-scale people flow dataset featuring compression and data mining techniques. This tool firstly compresses the people flow datasets using UniversalSAX, an extended method of SAX (Symbolic Aggregate Approximation). Next, we apply natural language algorithms to extract movement patterns. Finally, we visualize trajectories of people flow and extracted features to represent popular walking routes and congestions.

Keywords--- **People Flow, 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. 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. Finally, the technique visualizes the people flow datasets focusing on the movement patterns.

This paper shows a case study with a real-world people flow dataset in an exhibition and discusses effectiveness of the presented tool.

2. Related Work

This section introduces existing studies to analyze people flow and non-walker trajectory datasets.

There have been many existing methods which classify and visualize spatio-temporal people flow data recorded as real values. Yabushita et al. [2] 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. Fukute et al. [3] 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. Wang et al. [4] extracted and visualized features of automobiles passing at particular positions, applying datasets collected using many sensors on roads. Guo et al. [5] 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. Guo et al. [6] 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. Andrienko et al. [7] 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. Different from the study, our methods supposes that datasets are collected by cameras in small area like one floor in a shop.

Also, there have been many techniques on compression and pattern discovery techniques for people flow datasets. Thack et al. [8] converted the spatio-temporal 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. [9] 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. Different from these methods, our technique firstly extracts typical movement patterns and then visualizes them noticeably.
Yada [10] 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. Section 3.1 defines the people flow datasets. Section 3.2 describes a technique to convert the trajectories into sets of characters, 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 Reordering 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 motion capture device Xtion to record the people flow data. We assigned a particular ID to each walker, and measured positions of heads of pedestrians every dozens of micro seconds. 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

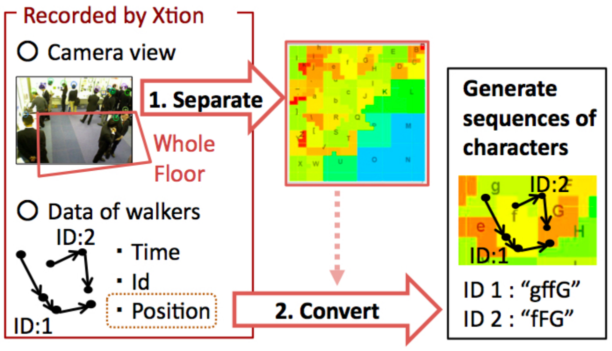

Figure 1. Flow of converting position values.

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 1 illustrates this process. We apply UniversalSAX [1], 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^k : number of partitions in each axis
c : 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^k segments. Then, we generate a Hilbert curve that passes every block once.

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 c regions by grouping these blocks, and assigns characters alphabetically to each region in the order of the sequential numbers of the blocks. Number of blocks belonging to one region depends on distribution of data items. 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 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 2. Run-length encoding is a reversible compression method which is especially effective for sequences that 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.

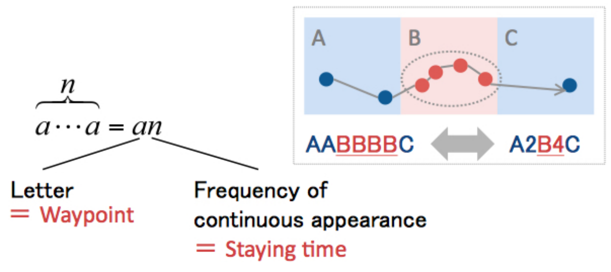

Figure 2. Applying run length encoding to a walking route.

3.3 Extracting Features of People Flow

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 characters which continuously appear in the sequences. Furthermore, we calculate distances among the sequences of characters to summarize or search for particular walking routes. In our current implementation, the distance just reflects positions and shapes of walking routes, while staying time is not considered.

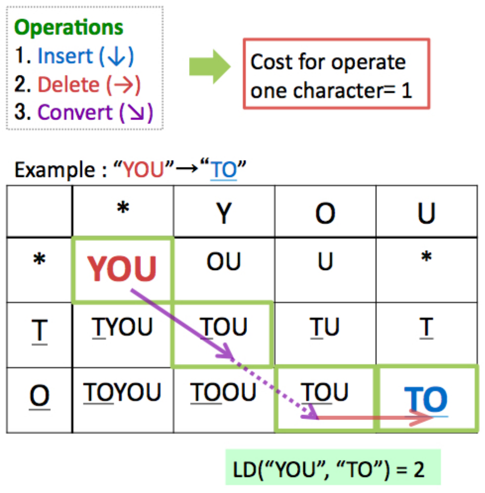

Figure 3. An example of calculating original Levenshtein Distance.

Our implementation firstly deletes the numeric characters from the run-length codes and deals with the sequences of alphabetical characters. 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, and that of s_2 is n, time complexity to calculate LD between them is $O(mn)$. 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 D to convert s_1 to s_2 applying dynamic programming. (See Figure 3).

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 example, while comparing two pairs of words, between "mine" and "nine", and that of "mine" and "fine", we may feel pronunciations of the second pair is more different. However, original definition of LD unnaturally determines both D("mine", "nine") and D("mine", "fine") as 1, because both the conversions just replaces the first character. To solve this problem, we apply weighted LD which can reflect differences between arbitrary pairs of characters by flexibly adjusting costs of insert, delete and replace operations. In our study, a difference between a pair of characters corresponds to a distance between two regions which has been defined in the lookup-table generated by the setup process described in Section 3.2. We simply refer the table to identify a difference between two regions $r1$ and $r2$ as $d(r1, r2)$. Figure 4 illustrates the costs of the operations. For example, the cost c to convert "abxd" to "abd" (delete 'x') or "abd" to "abxd" (insert 'x') is calculated as follows:

$$c = |n + d(b,x) + d(x,d) - d(b,d)|$$

Here, $d(a, b)$ is a cost while moving from a to b.

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 one 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.

Distance calculation among sequences of characters is useful to search for particular walking routes, or apply clustering algorithms to summarize the large amount of walking routes. We can adjust costs of insert, delete or replace operations independently by adopting the weighted LD, and therefore we can control conditions for search and clustering processes. For example, if we want to classify walking routes according to the wideness of movements, it is effective to set larger costs to insert and delete operations to distinguish various lengths of the sequences.

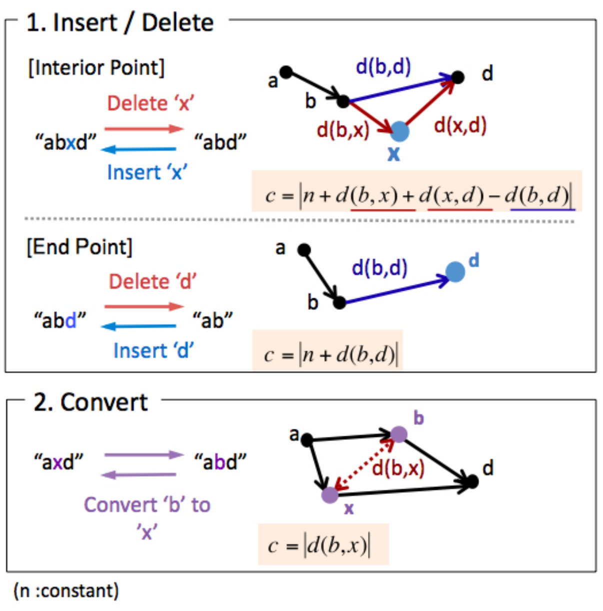

Figure 4. Costs of each operation.

3.4 Visualizing 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 3.2. 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 5(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 5(right). This representation avoids too complex visualization results caused by huge number of visible segments.

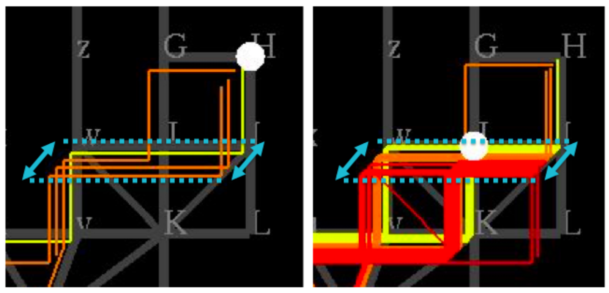

Figure 5. (Left) Visualizations of small number of segments. (Right) Bundling similar segments.

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. 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.

4. Example

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 6. 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', '\', ']' , '^' , '-', '‘', and 'a'-'1' to the regions. We firstly visualized congestions to briefly understand the tendency of the people flow as shown in Figure 7, 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, which depicted exhibits on the circles got attention of the participants.

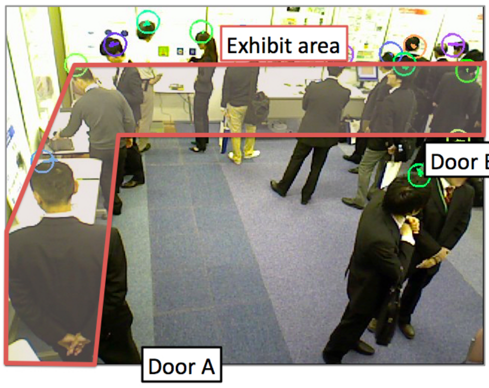

Figure 6. Camera view.

Next, we visualized particular walking routes and compared populations in each of the time periods, as shown in Figure 8. We colored trajectories based on HSV model, where larger hue values correspond that time of walkers' movements were later. Each of colors corresponding 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, 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.

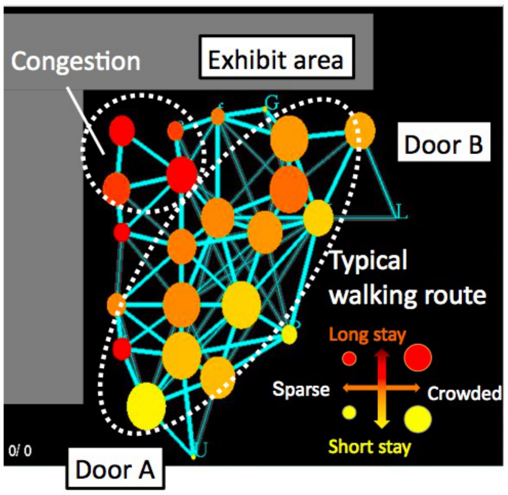

Figure 7. Visualization of congestions.

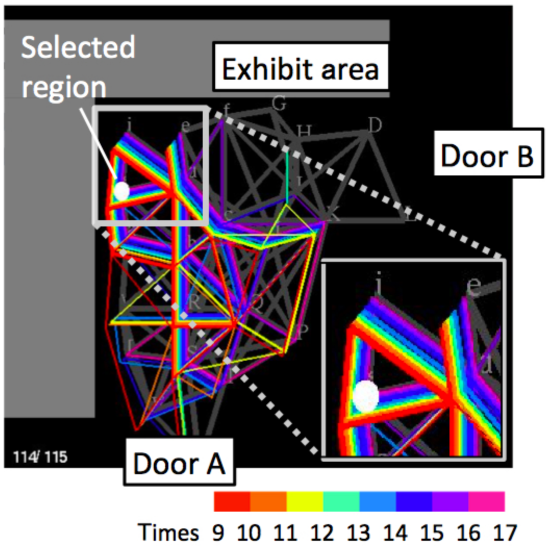

Figure 8. Visualizing walking routes passed the selected region.

Then, we selected more specific walking routes as sequential characters, and counted numbers of people who passed the routes by the search function. Figure 9 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 waypoints. There were larger number of walkers (corresponding to routes 1, 3, and 5 in Figure 9) came from door A, while smaller number of other walkers (corresponding to routes 2, 4, and 6 in Figure 9) 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 9) 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 9) turned buck after moving to the upper left exhibition. These indicate that we ought to devise arrangement of the exhibition to attract more visitors.

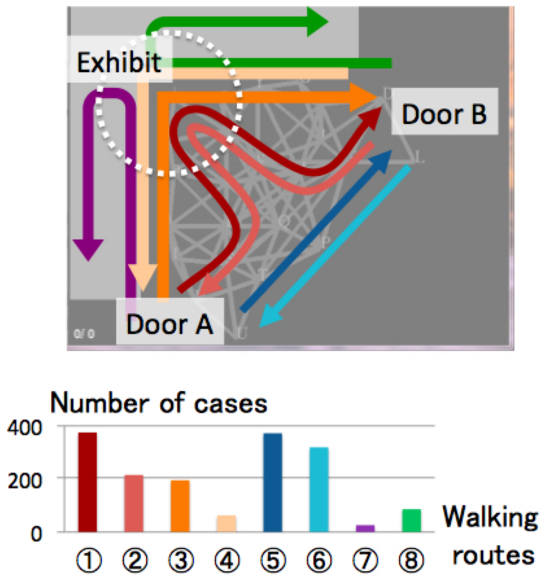

Figure 9. Visualizing walking routes passed the selected region.

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 demonstrated that the presented visualization technique succeeds to reduce the costs required to analyze people flow data.

5. Conclusions and Future Work

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.

Future issues of this study are follows.

- More abstract representation of walking routes.
- More functionality for searching for walking routes.
- Manual specification of region division in the setup process.
- Inferring the semantics of walking actions.

One problem is that too many segments may be drawn and therefore visualization results may be too complex. Concerning the example in this paper, it may be difficult to clearly visualize small number of segments separately. We would like to develop simpler representations, to emphasis representative or exceptional trajectories.

Also, 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.

References

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

[9] T. Oates, A. P. Boedihardjo, J. Lin, C. Chen, S. Franken-stein, S. Gandhi, “Motif Discovery in Spatial Trajectories using Grammar Inference”, ACM International Confer-ence on Information and Knowledge Management (CIKM 2013), 1465-1468, 2013.

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