ICIC Express Letters
Part B: Applications
Volume **10**, Number **6**, June **2019**

pp. 475–482

BIG DATA CONSTRUCTION AND DATA ANALYSIS BY LIFE LOG DATA ACQUISITION APPLICATION

Eri Domoto[1], Junko Shibata[2], Mizuki Fukushima[3] and Koji Okuhara[4]

[1]Department of Media Business
Faculty of Economics
Hiroshima University of Economics
5-37-1 Gion, Asaminami-Ku, Hiroshima 731-0192, Japan
er-domo@hue.ac.jp

[2]Faculty of Economics
Kobe Gakuin University
518 Arise, Ikawadani, Nishi-ku, Kobe, Hyogo 651-2180, Japan
shibata@eb.kobegakuin.ac.jp

[3]Fujitsu Hokuriku Systems Limited
3-4-30 Masuizumi, Kanazawa, Ishikawa 921-8611, Japan

[4]Graduate School of Engineering
Toyama Prefectural University
5180 Kurokawa, Imizu, Toyama 939-0398, Japan
okuhara@pu-toyama.ac.jp

Received November 2018; accepted February 2019

Abstract. *In this study, we develop a system that detects similar behaviors and event behaviors from life log data. The system consists of a text data collection part by cooperation of smart glass and image recognition API (Application Programming Interface) and an activity recognition part. Hierarchical cluster analysis for classifying data and co-occurrence network analysis are incorporated in the activity recognition part. Then, it is possible to grasp the behavior transition on the self-organizing maps. Big data construction and data analysis were performed using developed life log data acquisition application, and similarity patterns of events and events were detected.*
Keywords: Life log, Behavior pattern, Text data mining, SOM (Self-Organizing Map)

1. **Introduction.** In modern times, it is common for many people to carry a smartphone or a wearable device, and from the development of rapid information technology it is possible to acquire and record personal lives and behaviors as data [1]. It is thought that behavior data obtained by using such smartphone or wearable device can be utilized in individual life or utilized in society. There are many applications that acquire and analyze GPS (Global Positioning System, Global Positioning Satellite) of a smartphone or wearable device as a life log, and researches for highly receptive life logs are being conducted [2].

The purpose of this study is to develop a life log system that is widely accepted by many people, to detect and consider similarity and event characteristics of behavior patterns. Focusing on protecting personal information and taking time and effort as a life log widely accepted by many people, we develop an application that automatically acquires life log data. From the data acquired by this application, identify action patterns and consider similarity and event property.

To achieve this objective, we develop an application to acquire real-time visibility information using smart glass and image recognition. In addition, we use image recognition

DOI: 10.24507/icicelb.10.06.475

API to develop this application, and use EPSON MOVERIOTM BT-300 for the device. Analysis of data can be analyzed by conventional methods such as hierarchical cluster analysis and co-occurrence network analysis, and SOM. By comparing them, it is shown that similarity and event property of life log data can be detected.

2. Behavior Pattern Analysis from Life Log.

2.1. Life log for behavior pattern. A life log is a record (log) of human activity (life), and an act of acquiring a log related to individual activity with a sensor or the like is considered as a word origin of a life log. In this study, this activity is a life log, and big data generated based on personal behavior history is called life log data. There is no definition that a long time record or a huge amount of data is necessary for the life log.

By linking with smart ware, there is an application that can acquire life log data such as the number of steps and heart rate, and make use of it for the user's own health management. Also, there is an application that maps the position information automatically to the map or manually maps it. Since these applications can take action records, they can be used as daily lives and travel records. The above applications require GPS access permission, and other life log applications often require GPS.

2.2. Smart glass as device for environment recognition. In this study, we use SEIKO EPSON's see-through mobile viewer EPSON MOVERIOTM BT-300 as a smart glass (Figure 1). The reason for using it is that it is relatively inexpensive even among smart glasses, and it is possible to create and operate an Android application. EPSON MOVERIOTM BT-300 performs a see-through display in which the virtual object generated by the computer is superimposed on the real space the user sees. There are two types of see-through display (Figure 2), video see-through system and optical see-through system, and EPSON MOVERIOTM BT-300 can use optical see-through system.

FIGURE 1. Smart glasses

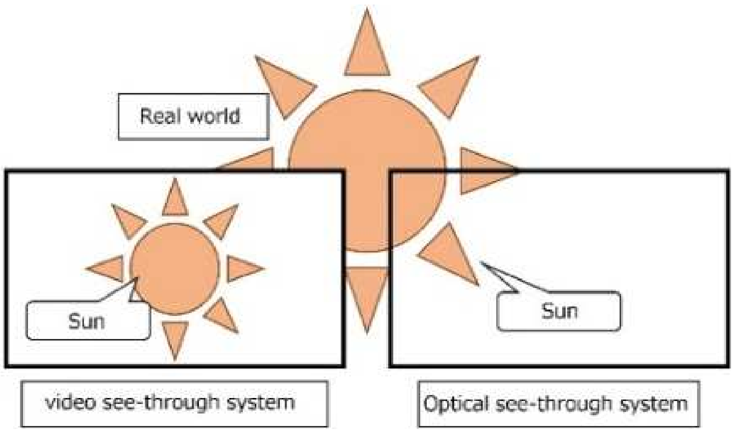

FIGURE 2. See-through display

2.3. **Text data from image recognition API.** The image recognition technology is a technique to allow a computer to understand an image. Pattern recognition that extracts meanings from the pattern of pixel signals in the image allows the computer to process the human visual function. By using the image recognition technology, various kinds of information possessed by the image can be acquired. One of them is acquisition of text information.

Typical image recognition API are Google Cloud Vision API, Computer Vision API, and IBM Bluemix Alchemy API. By using the image recognition API, since it is possible to use a large amount of data which is difficult to acquire by individuals alone, it is possible to develop an application which recognizes camera images acquired by a smartphone or a wearable device.

In this study, Computer Vision API that can acquire tags and captions is used as text data that can be obtained from images.

3. Proposal Method for Similarity and Event Detection.

3.1. **Developed system.** The development engine uses the Unity 5 development engine for game development provided by Unity Technologies. Unity 5 mainly deals with 3D objects and also supports output to mobile terminals. Image recognition API is developed using Computer Vision API. Since EPSON MOVERIOTM BT-300 is Android 5.1, create an Android application with API level 22. The activity recognition part incorporates hierarchical cluster analysis, co-occurrence network analysis, and SOM.

Figure 3 shows a flowchart of the life log data acquisition application in the data collection part. The screen is dark when activated, so that it does not display anything on the screen even when the camera is activated. Since MOVERIOTM BT-300 has the property of transmitting black screens, it is possible to acquire life log data without disturbing visibility. In this study, in order to always check whether data can be acquired, we use a program that displays the acquired tag in a size that does not interfere. The Computer Vision API machine reads the subject of the photographic image and has various functions such as reading and labeling the text in the image. Even labeling alone has the function of tags and captions. Tags calculate tag information based on elements in an image, based on more than 2,000 recognition elements, living things, landscapes, and the like. Captions display a summary as a human readable language in sentences.

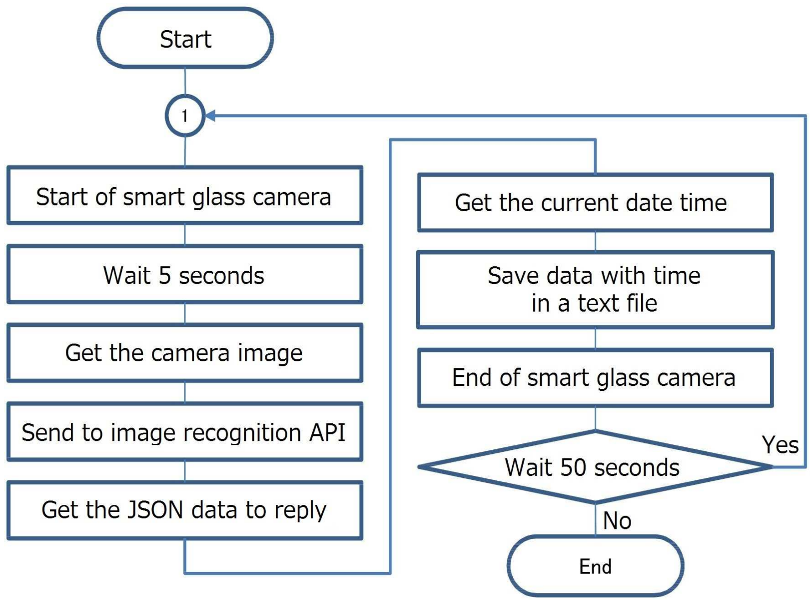

FIGURE 3. Application flowchart

When the camera image is acquired, it is transmitted to the image recognition API. Through the image recognition API, camera image information can be acquired as JSON data. Date and time of acquisition are added to this JSON data, and it is saved in MOVERIO as text data. The text file is saved as "long ¥ _report ¥ _yyyymm ¥¥ dd.txt". If there is no text file of the same name in MOVERIOTM BT-300, a new text file is created and text data is saved. If there is a text file with the same name, add text data to the last line of the text file and save it.

3.2. **Hierarchical cluster and co-occurrence network.** Hierarchical cluster analysis is a method of collecting clusters sequentially from the most similar individuals based on similarity or dissimilarity (distance) between individuals [3]. There are several methods for hierarchical cluster analysis, but in either case the following steps are taken.

1) Find the distance (or similarity) from the data. 2) Select the cluster analysis method (nearest neighbor method, nearest neighbor method, etc.). 3) Find the cophenetic matrix of the selected method. 4) Create a tree diagram based on cophenetic matrix. 5) Review the results.

In hierarchical cluster analysis, a matrix of distances from data is obtained, a cophenetic matrix is obtained from a matrix of distance, and a step of drawing a tree diagram based on cophenetic matrix is taken.

Data Matrix ⇒ Distance Matrix ⇒ Cophenetic Matrix ⇒ Result

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & & & \\ d_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ d_{m1} & d_{m2} & \cdots & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & & & \\ c_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ c_{m1} & c_{m2} & \cdots & 0 \end{bmatrix}$$

⇒ Draw tree diagram

There are several ways to generate a cophenetic matrix from the distance matrix, but the first stage is the same, and the distance between two individuals with the closest distance is the cophenetic distance. How the cophenetic distance is determined after the first stage is different depends on the method of cluster analysis.

Co-occurrence network analysis is a network that links words and words co-occurrence with words used in text as nodes, and expresses the strength of the link by Jaccard coefficients. The Jaccard coefficient $sim(v, q)$ is a similarity between sets and is defined by the following equation.

$$sim(v, q) = |v \cap q|/|v \cup q|$$

where v and q are the frequency of appearance of two words in the text.

3.3. **Visualization and unification by SOM.** SOM is a neural network algorithm proposed by T. Kohonen, a data analysis method for nonlinearly projecting high-dimensional data onto a two-dimensional plane, and is a multidimensional scale method, cluster analysis method. The basic structure of the self-organizing map is a two-layered neural network composed of an input layer and an output layer. The output layer is also called a competitive layer [4].

Assume that the variable layer of the individual j $(j = 1, 2, \ldots, n)$ to be analyzed is $\boldsymbol{x}_j$ $(x_{j1}, x_{j2}, \ldots, x_{np})$ in the input layer and k $(i = 1, 2, \ldots, k)$ units $\boldsymbol{m}_i$ are in the output layer. Any one unit in the output layer is linked to all of the variable vectors in the input layer. In the initial stage, weights $\boldsymbol{m}_i$ $(m_{i1}, m_{i2}, \ldots, m_{in})$ are attached to each variable by random numbers.

The Computer Vision API machine reads the subject of the photographic image and has various functions such as reading and labeling the text in the image. We can obtain a word vector $\boldsymbol{x}_j$ $(x_{j1}, x_{j2}, \ldots, x_{np})$ at time j $(j = 1, 2, \ldots, n)$, if an ith word is involved in text from the Computer Vision API at time j, then $x_{ji} = 1$ otherwise 0. And the vector is sent to SOM.

SOM algorithm

1) Find the most similar unit $\boldsymbol{m}_c$ compared to all units of input $\boldsymbol{x}_j$ and output layer, and make that unit the winner.

$$||\boldsymbol{x}_j - \boldsymbol{m}_c|| = \min_i\{||\boldsymbol{x}_j - \boldsymbol{m}_i||\}$$

2) Update the weight vector $\boldsymbol{m}_i$ of the unit found and its neighboring unit. Updating is done by the following formula.

$$\boldsymbol{m}_i(t+1) = \begin{cases} \boldsymbol{m}_i(t) + h_{ci}(t)[\boldsymbol{x}_j(t) - \boldsymbol{m}_i(t)] & i \in N_c \\ \boldsymbol{m}_i(t) & i \notin N_c \end{cases}$$

$$h_{ci}(t) = \alpha(t)\exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right)$$

$h_{ci}(t)$ in the equation is a neighborhood function, and the influence of $\boldsymbol{x}_j$ is adjusted by the proximity of the unit c and the neighboring unit i. $\alpha(t)$ in the expression $h_{ci}(t)$ is a coefficient of the learning rate, and r_c and r_i are coordinate position vectors on two dimensions of the units c and i.

$\sigma^2(t)$ is a function to adjust the radius of the neighboring region N_c of the unit c. $\alpha(t)$ and $\sigma^2(t)$ are monotonically decreasing functions with the learning number (or time) as a variable. The simplest monotonically decreasing function with the number of learning as a variable is $1 - \frac{t}{T}$. This t is the number of learning times (or times $1, 2, 3, \ldots, T$), and T is the total number of learnings set in advance.

3) Repeat (1) and (2) on the feature vector $\boldsymbol{x}_j$ ($j = 1, 2, \ldots, n$) of all inputs.

4. Numerical Example and Discussion.

4.1. Target text data corrected by smart glass and API.
Here we use the developed application to acquire life log data. We also use multivariate analysis from the acquired data to detect similarity of behavior patterns and event properties. The acquisition date of the life log data is 180 minutes from 10:30 to 13:30 on January 27 and 28, 2018. The acquisition time was 180 minutes because acquisition was made from the state where the device was charged at 100% until the charge was cut off. Although data was acquired approximately once a minute, the number of data is 190 because there is a wave in the processing capability of the device. Data acquired on 27th is data 1, and data acquired on 28th is data 2.

In order to facilitate comparison between data 1 and data 2, data 1 and data 2 are converted into one csv file (data 3). At this time, the label column stores delimiters between data 1 and data 2. Line numbers 1 to 190 are data 1, and 191 to 380 are data 2. By using data 3, it is possible to summarize multivariate analysis results of data 1 and data 2. For cluster analysis, this label column is included in the text data column. Using KH Coder, natural language processing is performed as preprocessing of the obtained text file to extract words [5,6].

4.2. Behavior pattern clustering by text data.
We perform hierarchical cluster analysis and co-occurrence network analysis from the acquired data with KH Coder. Perform cluster analysis of data 3. At this time, since there are no significant changes in the cluster numbers 6, 7, 8, the number of clusters was set to 5. The cluster showing what actions from the composed word is shown in Figure 4. Data 1 is close to a pink cluster, and since the same cluster contains the words "computer" and "desktop" data 1 shows that there is much work on the desktop PC. Also, data 2 contained lots of words such as "laptop" and "screen shot", thought that there was a lot of notebook PC work, but thought that data 1 has strong relationship with "computer". Both the red cluster and the blue cluster contain something that represents going out. The yellow cluster contains words representing kitchens, meals, furniture, and these three actions are considered to

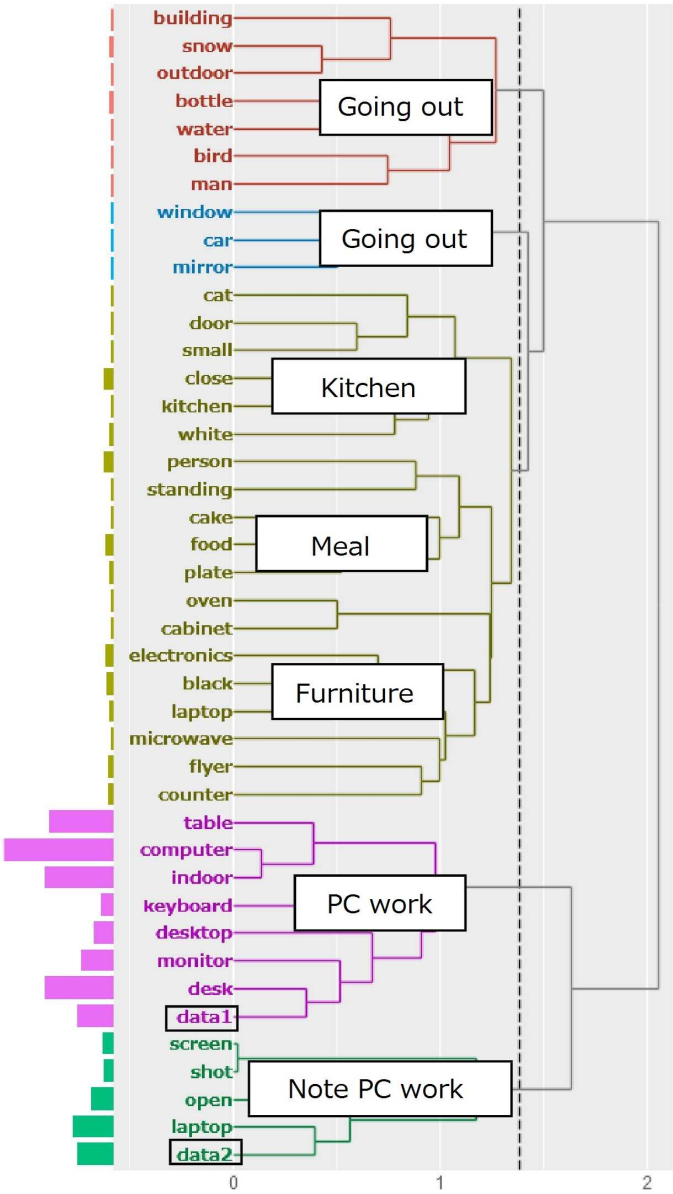

FIGURE 4. (color online) Results of hierarchical cluster analysis

be close behaviors. Also, it can be seen that both data 1 and data 2 are all about the same distance except for PC work.

In order to simultaneously output co-occurrence relations between data 1 and data 2, co-occurrence network analysis of data 3 is performed (Figure 5). In addition, we marked the word that we feel that we showed behavior in Figure 5. Both Data 1 and Data 2 have a co-occurrence relationship with extracted terms "table", "desk", "computer", and "indoor", both of which have a strong co-occurrence relationship with a Jaccard coefficient of 0.2 or more, and it is possible to confirm behavior with similarity. Also, you can see that the same PC work is shown, but "data 1" includes "desktop" and data 2 contains "laptop". From this we can see that the same PC work also uses different PCs.

4.3. **Similarity and event detection.** Create SOM of data 3 and compare the similarity of time series of data 1 and data 2. Figure 6 shows the result of directly applying the data 3 to the SOM. It turns out that it is difficult to obtain useful knowledge as it is. Figure 7 shows the results of the proposed system. Data 1 is indicated by a red line segment, and data 2 is indicated by a blue line segment. The number of clusters is set to 9, which is the number of clusters of data 1 and data 2. From this, it can be seen that the similarity

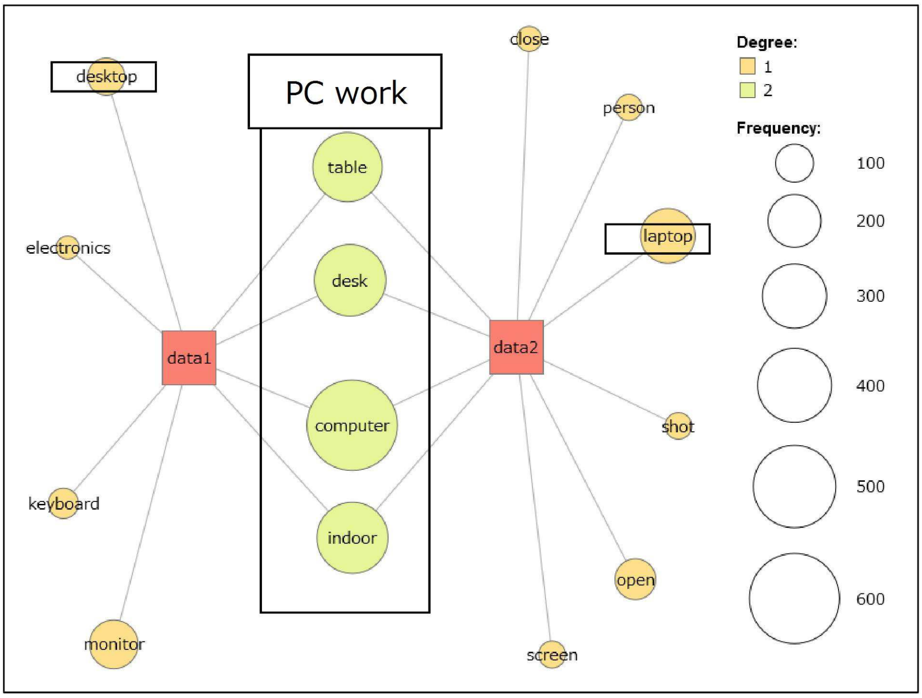

FIGURE 5. Results of co-occurrence network analysis

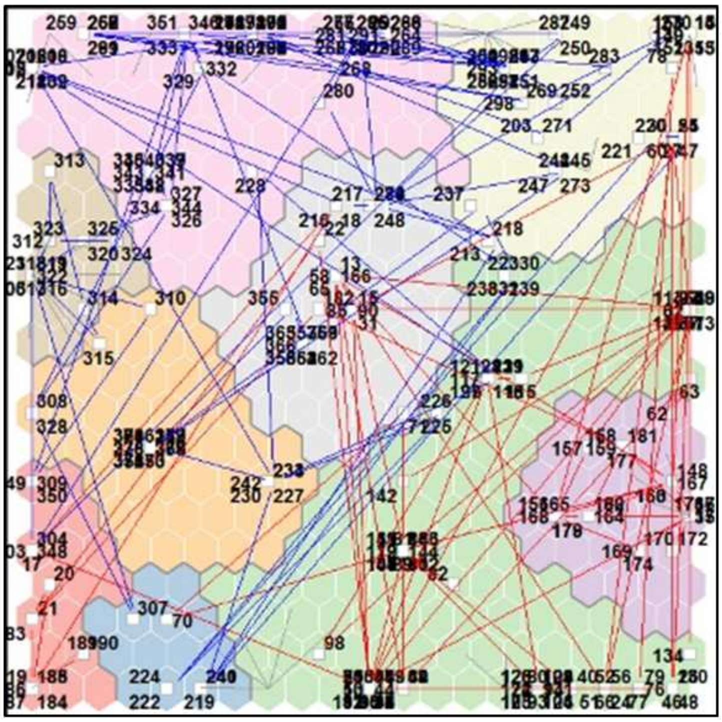

FIGURE 6. Results of SOM

of the time series of data 1 and 2 is low. The reason for this is that even though the same PC work is performed using another PC of the desktop PC and the notebook PC, it is because the behavior is distinguished from the difference in the object seen in the sight, even in the same action. Also, when you check the text of the time when both the red line and the blue line are connected, you are working closer to the computer screen such as "a screen shot of a computer" or "a close up of a computer" and the word "person" was included in the text. We think that this recognizes the image of the person on the PC screen and the poster at home.

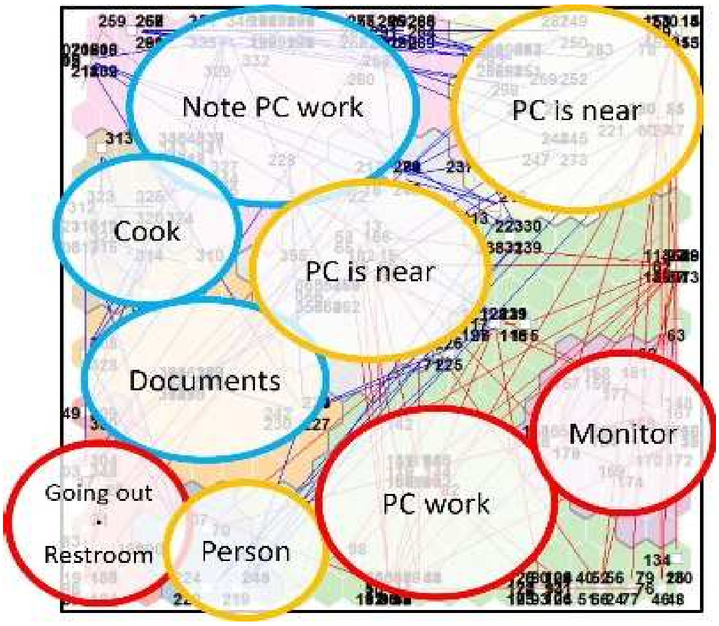

FIGURE 7. (color online) Results of the proposed system

From SOM, we were able to confirm whether there was time to perform similar behavior in the data time series. In this numerical experiment, similar actions were only at the moment of approaching the PC screen, but if data 1 contained meals it can be thought that there were many more overlapping parts. It was found that similar behaviors are also distinguished from objects entering sight.

5. **Conclusions.** The purpose of this study is to pay attention to personal information protection as a life log widely accepted by many people, and to acquire life log data automatically without taking time and effort. Then, it is possible to consider similarity and event property from the acquired data. Big data construction and data analysis were performed using developed life log data acquisition application, and similarity patterns of events and events were detected. In this paper, we first explained the behavior pattern analysis from life log by using smart glass as device for environment recognition. Then we proposed the similarity and event detection by hierarchical cluster, co-occurrence network and visualization of SOM. In numerical example, for similarity and event detection, we showed results of behavior pattern clustering from text data corrected by smart glass and API. As future work, we think that it is important to establish and find a measure for a quantitative comparison to our proposed similarity and event detection by behavior pattern analysis.

REFERENCES

[1] T. Terada, Challenges on activity recognition techniques using wearable sensors, *Computer Software*, vol.28, no.2, pp.42-54, 2011.
[2] K. Aizawa, Life log for practical use, *The Journal of the Institute of Television Engineers of Japan*, vol.63, no.4, pp.445-448, 2009.
[3] K. Yoshihara and H. Tokudaka, Introduction of cluster analysis, *Journal of Surface Analysis*, vol.21, no.1, pp.10-17, 2014.
[4] Y. Nishida and K. Honda, Visualization of technical solution by self-organizing maps for accelerating innovation, *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, vol.30, no.2, pp.543-547, 2018.
[5] T. Kyoko, Network analysis with KH coder and R, *Kurume University Computer Journal*, vol.28, pp.37-52, 2014.
[6] K. Kitamura, T. Yamasaki and K. Aizawa, Food logging and processing: Analysis of food images, *The Journal of the Institute of Television Engineers of Japan*, vol.63, no.3, pp.376-379, 2009.