A STORY AUTHORING TOOL WITH MAPPED PHOTO COLLECTIONS

Hideyuki Fujita and Masatoshi Arikawa
Center for Spatial Information Science, the University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8568, Japan
{fujita, arikawa}@csis.u-tokyo.ac.jp

Abstract

Our research goal is to facilitate the sharing of stories with digital maps. The main technical novelty of this paper is a method for semi-automated creation of photo slideshows with geotagged photographs. Story authoring software like movie editors and slideshow editors enable us to create stories by arranging elements like movie clips, audio clips, photographs, etc. in the order of the storyline. Those software have functions of sorting the elements by using their time stamps. For example, we can sort photographs in time series in common slideshow editors. On the other hand, people don't arrange all the part of stories in time series. Our research target is a slideshow with geotagged photographs. According to our analysis on spatial and temporal relationships between every two successive frames of the slideshows, in average, over 20% of two frames are arranged not in time series but in "spatial series." Spatial series means elements are arranged based on their spatial proximity or relationships. They make panoramic sequence, walkthrough sequence, etc, in slideshows. We therefore propose a method for sequencing photographs based on their temporal and also on spatial relationships. We have released a software for mapping personal photo collections and representing them as stories such as route guidances, sightseeing guidances, personal diaries, and so on. Each story is conveyed through a sequence of mapped photographs, presented as a synchronized animation of a map and an enhanced photo slideshow. We implemented the sorting function proposed in this paper on the software.

1. Introduction

Photo mapping services, which let people organize personal photo collections with digital maps, are now popular. For example, Flickr (Flickr, 2005) collects over eight hundred thousand geotagged photographs and Panoramio (Panoramio, 2005) collects over nine millions geotagged photographs at this moment. In these services, a map is mainly help with visualizing, organizing, and searching photo collections. On the other hand, one of the user requirements across the range of future photo sharing technologies is to support storytelling with digital photographs (Frohlich, 2002). Our research goal is also to facilitate the sharing of stories with a map. Thanks to various web mapping services, it is easy today to build our own collections of places and publish them on the Web. Many people map collections of places along with text narrations, photographs, audio tracks, movies, etc. Their aim should not be only organizing and searching collections of places, but also expressing and communicating their stories. However the common framework of web mapping services is not sufficiently expressive to communicate stories in a narrative fashion. For example, when the number of the mapped collections of places is particularly large, it is neither easy for viewers to interpret the map nor is it easy for the creator to express a story as a series of events in the real world. This is because each narrative, in the form of a sequence of textual narratives, a sequence of photographs, a movie, or audio is mapped to just one point. As a result, it is up to the viewer to decide which points on the map must be read, and in what order.

We therefore proposed a new framework, *Spatial Slideshow*, for mapping personal photo collections and representing them as stories such as route guidances, sightseeing guidances, historical topics, fieldwork records, personal diaries, and so on (Fujita, 2008a). Each story is conveyed through a sequence of mapped photographs, presented as synchronized animations of a map and an enhanced photo slideshow. We have released a software for editing and playing spatial slideshows.

The aim of this paper is proposing a method for semi-automated creation of spatial slideshows as slideshows with geotagged photographs. Story authoring software like movie editors and slideshow editors enable us to create stories by arranging elements like movie clips, audio clips, photographs, etc. in the order of the storyline. Those software have functions of sorting the elements by using their time stamps. For example, we can sort photographs in time series in common slideshow editors. On the other hand, people don't arrange all the part of stories in time series. According to our analysis on spatial and temporal relationships between every two successive frames of the spatial slideshows created by users, in average, over 20% of two frames are arranged not in time series but in "spatial series." (Fujita, 2008b) Spatial series means that elements are arranged based on their spatial proximity or relationships. They make panoramic sequence, walkthrough sequence, etc, in slideshows. With the aim of semi-automated creation of spatial slideshows, the main technical novelty of this paper is therefore as follows:

a method for sequencing mapped photographs based on their spatial relationships

The remainder of this paper is organized as follows: Related works and the features of our approach are described in Section 2. The structure and characteristics of a spatial slideshow are described in Section 3. A spatially organized photo data structure, the basic unit in this study, is also defined in this section. A method for sequencing photographs placed on a map is described in Section 4; this is the main feature of this paper. An implemented prototype system is introduced in Section 5, and some concluding remarks and future works are described in Section 6.

2. Related Work

Our research concerns the facilitation of personal digital storytelling, which is currently one of active topics in the CHI (computer-human interaction) research community. More specifically, we deal with storytelling with maps, and propose a map-based story authoring tool especially in this paper. From these aspects, we summarize the related works and describe the features of our approach as follows.

2.1. Personal Digital Storytelling

From their field study at the Digital Storytelling Workshop, Landry et al. clarifies the requirements of personal storytelling software. As one of requirements, they conclude that personal storytelling software should enable users to not only manipulate media but also seriously consider the story they wish to tell. In this light, some tools indeed provide functions for visualizing and editing the story structure or storyline (Landry 2006); however, we use a map to realize this functionality. For example, photographs, as elements of the story, are visualized on a map, and the map as well as the mapping process itself also support the establishment of the storyline.

2.2. Map-based story authoring

Personal photo album software and movie authoring software like iPhoto (Apple inc.), iMovie (Apple inc.), and Picasa (Google inc.) now treat geotagged photographs or movie tracks. At the moment they only have functions for visualizing on maps. Pongnumkul et al. suggest a map-based storyboard as a novel user interface for browsing and authoring a long tour video. To create such a storyboard, they employ an automatic pre-processing component to parse the video into coherent shots, and an authoring tool to enable the user to tie the shots with landmarks on the map by hand (Pongnumkul 2008). In order to help creating storylines, the method uses only video image processing, and not utilizes geospatial information. In contrast, our method utilizes spatial relationships of geotagged photographs for semi-automated creation of storylines.

2. Spatial Slideshow

A spatial slideshow is a slideshow of photographs placed on a background map. Photographs in a spatial slideshow are arranged as follows:

mapped to arrows on the background map
lined up in the order of the storyline of the slideshow
with attached text narration

We call the arrows that represent photographs *photo vectors*; as explained later, they indicate from where to where the photograph was taken. To be more precise, we define a spatial slideshow as a sequence of frames. Each frame is composed of a photo image, a photo vector, and text narration. Text narration of each frame is a part of a whole narrative expressed by a spatial slideshow. While playing a spatial slideshow, a highlighted photo vector moves to next one in sync with a highlighted photo image switches to next one. In order to control a spatial slideshow, the viewers play, pause, step forward, and rewind it just like a usual slideshow, or directly select a photo vector on the map. In order to create a spatial slideshow, users draw arrows called photo vectors on a map and then line them up according to the storyline of the slideshow. The features of a spatial slideshow are as follows:

Figure 1. A photo sequence of a spatial slideshow

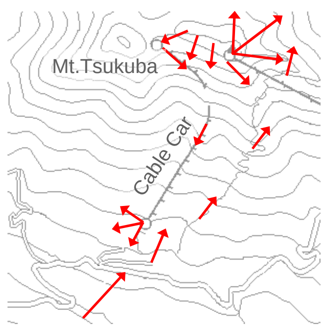


Figure 2. Photo vectors of a spatial slideshow

(a) Storytelling with a map

The author creates a spatial slideshow by selecting and arranging photographs based on a storyline. In spatial slideshows, photographs on a map is not scattered independently but arranged in the order of the storyline. Therefore, we can rather passively view photographs on a map as a story in the order specified by the creator without being required to select or search on a map. In addition, a spatial slideshow has a spatial and temporal structure in a machine readable form. For example, time stamp indicating shooting datetime, photo vector indicating from where to where the photograph was taken, their relationships between successive frames, and so on. We assume these structures are applicable to analyzing stories, developing software functions like its semi-automatic creation described in this paper, and so on.

(b) Spatial slideshow animation

Spatial slideshow animation is a transitional animation of photo images and photo vectors that visualizes the progression of the story as movements from photo to photo. It is a synchronized animation of the following two animations.

- A three-dimensional animation of photo images as though moving from the current photograph toward the next photograph
- A two-dimensional animation of photo vectors on a map as though moving from the current photo vector toward the next photo vector

It is created automatically in playing spatial slideshows by using spatial relationships of photo vectors. Features and a creating method of a spatial slideshow animation are described in another paper (Fujita 2008b).

Sample

We have released software for editing and playing spatial slideshows, as described in Section 5, and have, until now, acquired about a hundred spatial slideshows created by users. The types of stories created by users include route guides, sightseeing guides, guides on historical subjects, fieldwork records, personal diaries, and so on. Here, we present an example of a spatial slideshow created by a user; it is entitled “Climbing Mt. Tsukuba.” Figure 1 shows an abstracted photo sequence that constitutes the spatial slideshow. All the photographs are mapped as shown in Figure 2. The spatial slideshow is a record of a short trip, of climbing the mountain to see the red autumn leaves.

Photo Vector

A photo vector is an arrow indicating from where to where the corresponding photograph was taken. It is a map symbol and a geographic feature for a photograph. We use photo vectors for mapping photographs. A photo vector P is defined as follows:

$$P = (x_S, x_E)$$

x_S is the viewpoint; it is the position of the camera at a time when the photograph was taken. x_E is the gazing point; it represents the position of the object focused in the image. Figure 3 shows a geometric model of a photo vector. At the present time, we assume a user manually set these points by hand drawing arrows on a map. The reason why we use photo vectors for mapping photographs is described in another paper (Fujita 2008a).

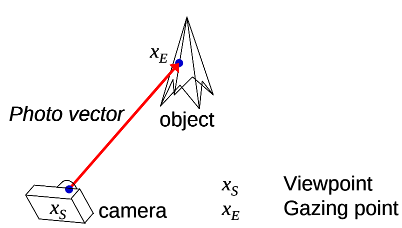


Figure 3. Photo vector

3. Sequencing photo vectors

This section proposes a method for semi-automated creation of photo sequences of spatial slideshows by using spatial relationships of their photo vectors. The aim of the method is sorting photo vectors in a spatially sequential order. We perform the following steps for sequencing photo vectors.

(1) Clustering photo vectors (Section 4.1)
(2) Sorting photo vectors in each cluster (Section 4.2)

Each step is explained in detail in the following subsections. We perform clustering before sorting because in many cases, all photographs on a map compose not only one sequence, rather they compose multiple sequences.

3.1. Clustering photo vectors

We apply nearest neighbour clustering method to points composing photo vectors. The reason why we select the method from many other clustering methods is that the method is commonly said to be appropriate when points are scattered in a line. As initial condition of the method, all points are assigned to different clusters containing only one point, and the clustering proceeds successively by merging smaller clusters into larger ones. Two clusters whose dissimilarity is the smallest are merged into one cluster. We use geographic distances as dissimilarity between clusters. As an exception, we set dissimilarity between a viewpoint and a gazing point composing one same photo vector to 0. By the exception, every two clusters containing them are merged at first in the merging process. In other words, clusters containing one photo vector are made at first. It is a feature of our method. Users interactively control how many groups photo vectors are divided into. The procedure in detail and the user interface are as follows:

Definition

At first we define some variables and functions. X is defined as a set of all points composing all photo vectors to process. It is the target of the clustering and sorting process.

$$X = \{x_1, x_2, ..., x_n\}$$

where x_i, $1 \le i \le n$ is a point. It is a viewpoint or a gazing point of photo vectors. C_i is defined as a cluster containing multiple points.

$$C_i = \{x_a, x_b, ... \mid x_a, x_b, ... \in X\}$$

Function $\mathrm{IsPhotoVector}$ is defined as a function to decide whether two points are on one photo vector or not. It returns true when one of the two arguments is the viewpoint and the other is the gazing point of the same photo vector.

$$\mathrm{IsPhotoVector}(x_a, x_b) = \begin{cases} \mathit{true} & (x_a \ne x_b,\ x_a \text{ and } x_b \text{ compose the same photo vector}) \\ \mathit{false} & (\text{otherwise}) \end{cases}$$

Function D is defined as dissimilarity between two clusters or points.

$$\mathrm{D}(C_a, C_b) = \min_{x_a \in C_a, x_b \in C_b} (x_a, x_b)$$

$$\mathrm{D}(x_a, x_b) = \begin{cases} 0 & (\mathrm{IsPhotoVector}(x_a, x_b) = \mathit{true}) \\ \text{Geographic Distance between } x_a \text{ and } x_b & (\text{otherwise}) \end{cases}$$

With these definitions, the procedure is summarized as follows:

Procedure

1. Make empty clusters C_i, $1 \le i \le n$. n is the size of X. Assign each one point in X to each different cluster.

$$C_i = \{x_i\}$$

2. Select two clusters whose dissimilarity D is the smallest, and make a new cluster by merging them. Selected clusters are not selected again in this process. Suppose C_a and C_b are selected and C_c is made below.

$$C_c = C_a \cup C_b$$

   C_a and C_b are not selected again in this process.

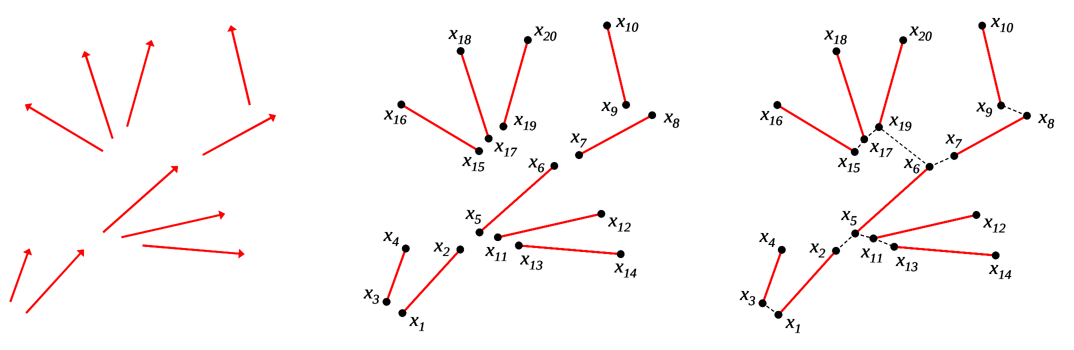
3. Stop the process if one cluster containing all points in X are made else back to step 3. A dendrogram of clusters is made at the end of the clustering process.

User Interface

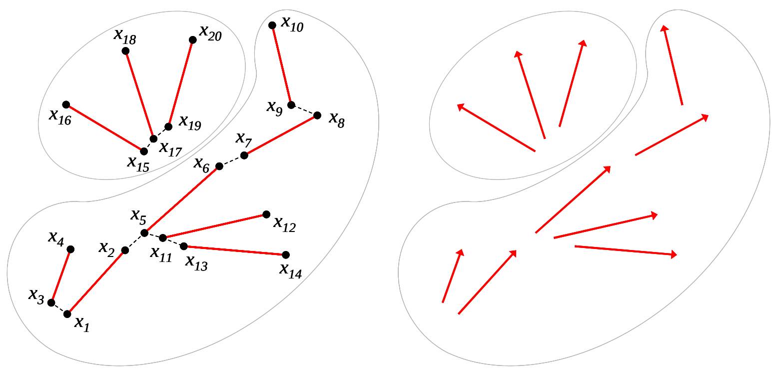
A user controls a clustering level interactively with a map and a slider bar. The clustering level is controlled by moving the slider bar. When the level is changed, the map is updated in real time and the clustering result according to the specified level is visualized. A clustering level is corresponding to criteria for distances between clusters which is used to decide whether they are merged or not. Since we made a complete dendrogram of the clusters at first, clustering result at any level is visualized in real time by cutting the dendrogram at the desired level.

Example

As an example, we process photo vectors shown in figure 4(a). Figure 4(b) shows points x_1, x_2,…, x_{20} which compose the photo vectors. Solid lines in the figure specify that each two points linked by the lines compose photo vectors. Each two clusters contained a point linked by the lines are merged at first at step 2. After repeating step 2, all clusters are merged and one cluster containing all points are made as shown in figure 3, and the cluster merging process stops. Dashed lines in figure 4(c) specify that each two points linked by the lines are selected to merge two clusters in the cluster merging process. All points are linked by one or more solid lines or dashed lines when the clustering process stops as shown in the figure. Then a user controls a clustering level after the cluster merging process. For example, the points are divided into two clusters in figure 4(d). It is realized by cutting off the longest dashed line $x_6 x_{19}$ in figure 4(c). Figure 4(e) shows the final result. Photo vectors are clustered based on how points are clustered.



(a) Photo vectors (b) Points composing photo Vectors (c) The result of the cluster merging process



(d) Clustered Points (e) Clustered Photo Vectors

Figure 4. Clustering process

3.2. Sorting photo vectors

We define a graph with photo vectors and links between them, and process sorting of photo vectors as a graph search (graph traversal) problem. A graph is generally defined by a set of vertices and a set of edges. We apply points composing photo vectors to vertices. We make two types of edges; an edge composing a photo vector and an edge connecting two photo vectors. It is a feature of our method. Ordered list of photo vectors are made by performing depth-first search of the graph. Sorting process in this section is assuming that the clustering process in Section 4.1 has been performed, and it is a method for sorting photo vectors in each cluster. The procedure in detail is as follows.

Definition

At first we define some variables and functions in addition to those of the clustering process. Below we use X as a set of points contained in one cluster, and sort points in

X. We define two types of edge $e^{(P)}$ and $e^{(L)}$. An edge $e^{(P)}$ connects a photo vector's viewpoint and its gazing point.

$$e^{(P)} = \{x_a, x_b\}, \text{ IsPhotoVector}(x_a, x_b) = true$$

$E^{(P)}$ is defined as a set of edge $e^{(P)}$.

$$E^{(P)} = \{e_1^{(P)}, e_2^{(P)}, \ldots\}$$

An edge $e^{(L)}$ connects two photo vectors. It connects a viewpoint or a gazing point of one photo vector to a viewpoint or a gazing point of another photo vector.

$$e^{(L)} = \{x_a, x_b\}, \text{ IsPhotoVector}(x_a, x_b) = false$$

$E^{(L)}$ is defined as a set of edge $e^{(L)}$.

$$E^{(L)} = \{e_1^{(L)}, e_2^{(L)}, \ldots\}$$

Function $\mathrm{T}^{(P)}$ is defined as a function which returns an adjacent vertex based on $E^{(P)}$.

$$\mathrm{T}^{(P)}(x_a) = x_b$$

where x_a and x_b satisfy the following formula:

$$\{x_a, x_b\} \in E^{(P)}$$

Function $\mathrm{T}^{(L)}$ is defined as a function which returns multiple adjacent vertices based on $E^{(L)}$.

$$\mathrm{T}^{(L)}(x_a) = x_b, x_c$$

where x_a, x_b and x_c satisfy the following formula:

$$\{x_a, x_b\} \in E^{(L)}, \ \{x_a, x_c\} \in E^{(L)}$$

With these definitions, the procedure is summarized as follows:

Preparation of the graph

1. Set $E^{(P)}$ and $E^{(L)}$ empty.
2. Make an edge $e^{(P)} = \{x_a, x_b\}$ from each photo vector, and add the edge $e^{(P)}$ to $E^{(P)}$. Make edges from all photo vectors and assign them to $E^{(P)}$.
3. In step 2 of the clustering process (Section 4.1), two points, suppose x_a and x_b, are selected as the smallest dissimilarity between two clusters to merge. At that time, make a new edge $e^{(L)} = \{x_a, x_b\}$ by using the selected two points x_a and x_b, and add the edge $e^{(L)}$ to $E^{(L)}$. When the clustering process stopped, each photo vector is connected to one or more other photo vectors by edges.

Setting starting points

In order to decide the starting point x_{start}, we narrow down the points in X by applying the following conditions one by one. The starting point is decided when the number of possible starting point becomes one. If possible starting point doesn't become one even when all the conditions are applied, user manually specifies the starting point from them.

Conditions for setting x_a as the starting point:

(1) $\mathrm{T}^{(L)}(x_a) = null$

(2) $\mathrm{T}^{(P)}(x_a) = x_b, \ \left|\mathrm{T}^{(P)}(x_b)\right| = 1$

(3) $\mathrm{T}^{(P)}(x_a) = x_b$, $\mathrm{T}^{(P)}(x_b) = x_c$, Type of x_b and x_c are one of the following set:

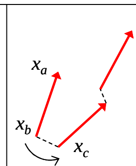
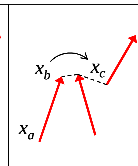
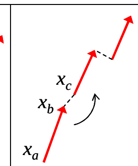
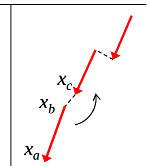
x_b is a viewpoint and x_c is a viewpoint

x_b is a gazing point and x_c is a gazing point

x_b is a gazing point and x_c is a view point

Table 1 shows the condition (3).

Table 1. A condition to be a starting point (Condition (3))

				
x_b	viewpoint	gazing point	gazing point	viewpoint
x_c	viewpoint	gazing point	viewpoint	gazing point
$x_a x_b$ becomes starting photo vector	Yes	Yes	Yes	No

Procedure

Define A and B as ordered lists of points in X. Data structure of A is stack. B gives the result of the sorting process.

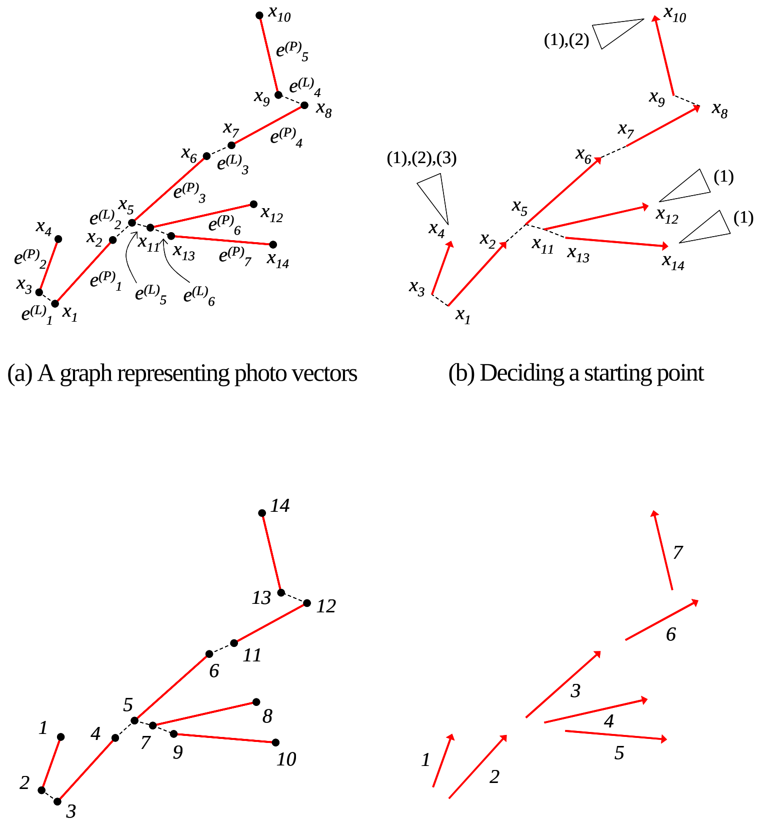
1. Set A and B empty.
2. Assign x_{start} to A and B.
3. Pop an point x_a from A.

   For each x_i in $\mathrm{T}^{(P)}(x_a)$, assign x_i to A and B if $x_i \notin B$.

   For each x_i in $\mathrm{T}^{(L)}(x_a)$, assign x_i to A and B if $x_i \notin B$.
4. Stop the process if A is empty else back to step 3.

Example

As an example, we process photo vectors in the right cluster of figure 4(e). A graph is made as a result of the graph preparation process. Figure 5(a) shows the graph which is made of nodes x_1, x_2,…, x_{10}, edges $e_1^{(P)}$, $e_2^{(P)}$,…, $e_7^{(P)}$ and $e_1^{(L)}$, $e_2^{(L)}$,…, $e_6^{(L)}$. Then we decide the starting point. In figure 5(b), points which satisfy one of conditions to be a starting point are marked, and satisfied conditions are specified. In this case, x_4 is decided to the starting point x_{start} because only x_4 satisfies the condition (1), (2), and (3). By using the graph and the starting point described above, the sorting process is carried out as a graph traversal process. In figure 5(a), at the beginning of the process, x_4 is assigned to A and B at step 2. x_4 is popped from A and x_3 is assigned to A and B at step 3 since $\mathrm{T}^{(P)}(x_4) = x_3$. Then x_3 is popped from A and x_1 is assigned to A and B at step 3 since $\mathrm{T}^{(L)}(x_3) = x_1$. The graph traversal process is carried out like above. The result of the process is shown in figure 5(c). The numbers near the points specify the order to traverse. They are the order in B. Figure 5(d) shows the final result. Photo vectors are ordered based on the order of the points.



(a) A graph representing photo vectors
(b) Deciding a starting point
(c) Ordered points
(d) Ordered photo vectors

Figure 5. Clustering process

4. Prototype system

We have been developing and releasing "PhotoField (http://www.s-it.org/photofield/)," a personal photo album application, which we refer to as spatial photo album software. It is a tool for editing and viewing personal photo collections in the form of a story, i.e., a spatial slideshow. We have extended the software to include both clustering and sorting of photo vectors by implementing the technical suggestions of this paper. Figure 6 shows the graphical user interface of the software. It has visual interfaces for a map, thumbnails of photographs, and the selected photograph. As an editor of spatial slideshow, it realizes the following functions:

(a) Mapping photographs

A user can map photographs as photo vectors by drawing arrows on the background map.

(b) Editing spatial slideshows
A user can create spatial slideshows by selecting the photographs in order, and then adding comments or text narrations to each photograph. Spatial clustering and sorting functions described in Section 4 can be used in sequencing photographs. Figure 8 shows examples of the function. Colored buffers around multiple photo vectors represent clusters. Different colors represent different clusters. Photo vectors in each cluster are arranged in-line. In other words, the clusters are short slideshows. Therefore a user can create a whole slideshow by just combining the clusters just like creating a whole movie by short movie clips.

(c) Reading and writing spatial metadata of photographs
The spatial metadata of a photograph such as the viewpoints and gazing points are saved and loaded as an XML file.
As a viewer of spatial slideshow, the system realizes the following functions:

(d) Controlling spatial slideshows
The user can play, pause, step forward, and rewind spatial slideshows by the corresponding control buttons as in common media players or by directly selecting a photo vector on the map. In the latter case, the focus of the map moves to the selected photo vector, and the slideshow also moves to the photograph corresponding to that photo vector.

(e) Playing spatial slideshow animations
While playing spatial slideshows, three-dimensional photo animations moving from photo to photo are created in real time and played. The photo vectors on the map also move in the animation and visualize the route defined for the spatial slideshow.

Figure 7 shows a gallery website of spatial slideshows created by users of the software.

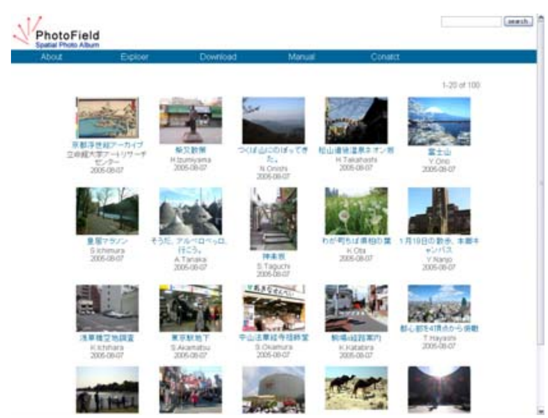


Figure 6. Prototype system

Figure 7. Gallery website

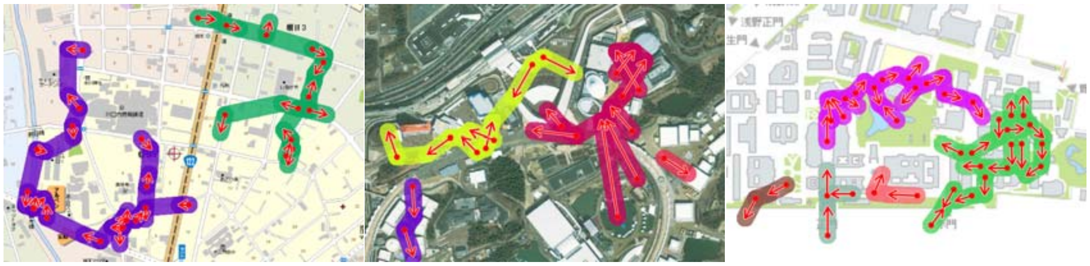


Figure 8. The results of the clustering function

5. Conclusion and future work

We described a method for semi-automated creation of photo slideshows with geotagged photographs. It is realized by clustering and sorting arrows representing photographs on a map. We assume proposed method will lower the complexity barrier in creating slideshows with geotagged photographs. In many common digital maps, objects on a map are represented as icons, and detailed information is shown by clicking each of the icons. Thus it is a bother clicking icons one by one especially when the number of icons is large. Proposed framework of sequencing objects on a map based on their spatial relationships is also applicable to such common cases. The possible directions for future work are as follows:

- Developing a method for extracting spatial patterns

There seem to be some major spatial patterns in photo sequences manually crated by users such as panoramic sequence, walkthrough sequence, etc. We plan to develop a method for automatically extracting such patterns by using spatial relationships of photo vectors.

Acknowledgement

This work was supported by the Grant-in-Aid for Young Scientists (B) No.18709002 of Japan Society for the Promotion of Science.

References

Flickr (2005). Yahoo!, Inc., <http://www.flickr.com/map/>.

Panoramio (2005). Google, Inc., <http://www.panoramio.com/>.

D. Frohlich, A. Kuchinsky, C. Pering, A. Don, and S. Ariss (2002). "Requirements for Photoware," Proc. of the 2002 ACM Conference on Computer supported cooperative work (CSCW2002), pp.166-175.

B. M. Landry and M. Guzdial (2006), "Learning from Human Support: Informing the Design of Personal Digital Story-Authoring Tools," Proc. of CODE 2006,

<http://www.units.muohio.edu/codeconference /proceedings/conference_papers4.htm>.

Pongnumkul, S., Wang, J., and Cohen, M. (2008), “Creating Map-based Storyboards for Browsing Tour Videos,” Proc. of the 21st ACM Symposium on User Interface Software and Technology (UIST2008), pp.13-22.

Fujita, H., Arikwa, M. (2008a). “Storytelling with Animation of Photo Collections Mapped to Arrows (in Japanese),” Proc. of Interaction 2008, IPSJ, pp.83-90.

Fujita, H., Arikwa, M. (2008b). “Animation of Mapped Photo Collections for Storytelling,” IEICE Transactions on Information and Systems 2008 E91-D(6), pp.1681-1692.

Miyamoto, S. (1999). “An introduction to clustering analysis,” Morikita Publishing (in Japanese), pp.89-90.