MIAOW: A 3D Image Browser Applying a Location- and Time-Based Hierarchical Data Visualization Technique

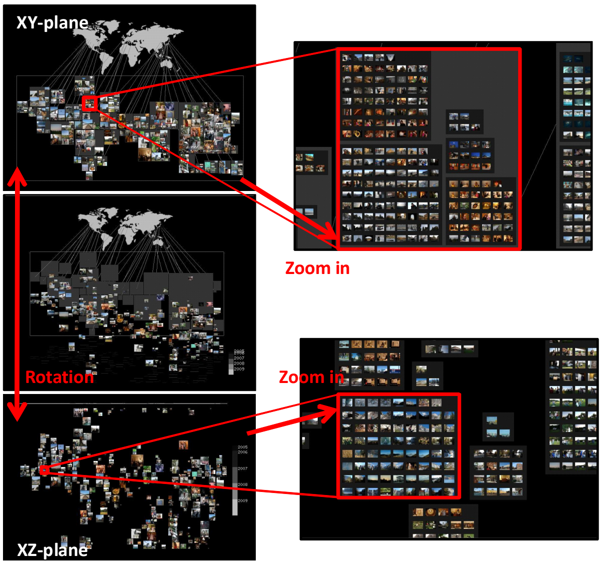

Figure 1: 3D image browser MIAOW. It represents hierarchical clusters of 9,500 photographs as nested rectangular regions. (Upper-Left) XY-plane which the two axes correspond to shooting longitude and latitude of the photographs, displaying representative photographs of the clusters. (Lower-Left) XZ-plane which the two axes correspond to shooting longitude and date of the photographs, displaying representative photographs of the clusters. (Right) Displaying all photographs in the clusters while zooming in.

1 Introduction

Browsing of image collections is important and useful for the overview and retrieval of images. Many existing image browsing techniques focus on intelligent layout and navigation of images. Organization of collections of images is one of the most important key points to effectively browse them. Several image browsing techniques applies manually assigned metadata or constructed structures. Another solution for effective image browsing is automatic image retrieval. Content based image retrieval (CBIR) is an extensively explored topic and has been applied to various problems in image search and categorization.

We think timestamp is one of the best information to automatically and effectively organize collections of personal photographs. Also, We think location is also good information for image browsers. We discuss to effectively utilize time and location information to develop a preferable image browser in this paper.

This paper presents a 3D image browser MIAOW (Memorized Images Album Organized by When/Where), which represents hierarchically clustered photographs based on their shooting locations and times. MIAOW supposes a 3D space with an orthogonal coordinate system to place a set of photographs; it assigns two axes (X and Y axes in this paper) to the shooting locations (longitudes and latitudes) of the photographs, and the other axis (Z axis in this paper) to their shooting time. We expect that MIAOW makes users easier to search for personal photographs associated to their memories with their locations and times. Figure 1 shows the snapshot of image display by MIAOW.

As a preprocessing, MIAOW firstly applies the bottom-up agglomerative clustering algorithm twice, to divide the photographs based on their longitudes and latitudes, and then based on their times. MIAOW then places all clusters of the images onto XY-, XZ-, and YZ-planes of the 3D space, while it represents them as nested rectangular regions. This representation is similar to a previously presented image browser CAT (Clustered Album Thumbnail) [1], which applies hierarchical rectangle packing algorithm to effectively place a set of images.

Our implementation of MIAOW provides orientation and zooming user interface. The orientation interface navigates users to shuttle between location and time spaces, so that they can easily find interested clusters of photographs. Meanwhile, the zooming interface is useful to focus on interested clusters of photographs. Current our implementation firstly displays representative images for each cluster, and switches them into the individual images in the clusters when a user zooms into them. This mechanism is preferable both for visual recognition and system performance, as discussed in [1].

2 Presented Technique

Figure 2 shows the processing flow of image clustering and browsing processes. Here, we assume that a collection of images each assigned with location (longitude and latitude) and time are given.

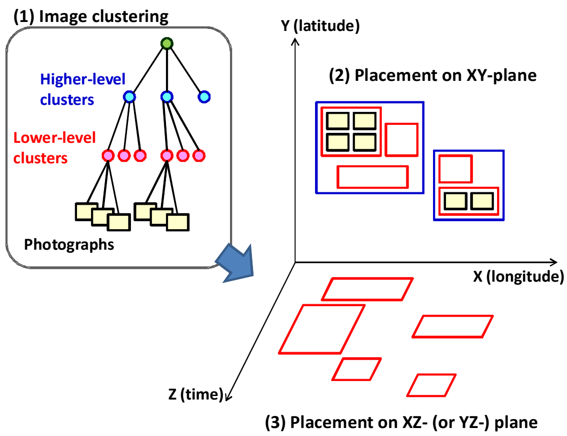

Figure 2: Processing flow of MIAOW.

Image Clustering

MIAOW constructs two-level hierarchy of images by applying two-level clustering.

The first clustering is based on the shooting locations. Our implementation assigns longitude and latitude values to X- and Y-coordinates. Our implementation of the first clustering simply calculates distances between arbitrary pairs of photographs on the XY-plane, then constructs a dendrogram based on the distances, and finally divides the images according to predefined threshold of the distances. The second clustering is based on the shooting times. Our implementation simply sorts the photographs in each of higher-level clusters based on their shooting times, then constructs a dendrogram based on the differences of the times, and finally divides the images according to predefined threshold of the differences.

Location-Based Hierarchical Data Placement

MIAOW applies Treemap-like space-filling hierarchical data visualization technique [1]. It places a set of images onto a display space based on a bottom-up packing algorithm consists of the following three phases:
Phase 1-1: The technique firstly places a set of image thumbnails in a lower-level cluster in grid layout, and encloses them by a rectangular border. It repeats this process for all the lower-level clusters.
Phase 1-2: The technique then packs and encloses all the rectangles corresponding to the lower-level clusters that belong to the same higher-level cluster by a rectangular border. It repeats this process for each of the higher-level clusters.
Phase 1-3: The technique finally packs the rectangles of all the higher-level clusters, and encloses them by a rectangular border. While packing the rectangles, it reflects longitudes and latitudes of the clusters.

Time-Based Hierarchical Data Placement

After completing the rectangle placement on the XY-plane, MIAOW places rectangles on the XZ- and YZ-planes, where Z-axis corresponds to the shooting time of the images. Here, we would like to design the image browser so that we can smoothly shuttle location-based and time-based planes. To realize seamless switch between XY- and XZ- (or YZ-) planes, we implemented a time-based hierarchical data placement algorithm, as it preserves the X- or Y-coordinates values calculated on the XY-plane. The algorithm consists of the following three phases to place a set of images onto the XZ-plane:
Phase 2-1: The algorithm firstly packs all rectangles corresponding to lower-level clusters onto the XZ-plane while preserving their X-coordinate values calculated by Phase 1-2.
Phase 2-2: It then adjusts Z-coordinate values of the rectangles so that their positions well represent the times of the clusters.
Phase 2-3: It finally calculates the positions of images in the rectangles.

Remark that this process only places lower-level clusters on XZ- and YZ-planes. In other words, it does not pack the rectangles corresponding to the lower-level clusters independently for each higher-level cluster, but pack them all-in-one.

Orientation and Zooming User Interface

Our implementation supports various mouse operations for image browsing. It assigns left button to translation, right button to orientation, and wheel to zooming operation.

MIAOW supports rotation around X- or Y-axis according to the mouse operation. When a user moves a cursor vertically, it transfers the movement to the rotation around X-axis. When he/she moves horizontally, it applies the rotation around Y-axis. While this operation, MIAOW assigns positions of images in the 3D space. For example, If an image places at (x_0, y_0) on the XY-plane and (x_0, z_0) on the XZ-plane, the image floats at (x_0, y_0, z_0) in the 3D space, while rotating around X-axis. Also, MIAOW inversely rotates the images against the operation, so that images always face towards the view direction.

MIAOW switches the displaying image according to zooming operation. It displays representative images of lower-level clusters while zooming out, and switches to each image thumbnails while zooming in. The representative images are displayed inside the rectangular borders representing the clusters.

3 Example and Experiment

Figure 3 shows an example of display of XY-plane. Our implementation displays a world map behind the XY-plane that images are placed, and draws edges between higher-level clusters and corresponding positions on the map. These results demonstrate that MIAOW places representative images of clusters while it avoids overlap among the images, attempts to minimize the display area, and preserve the geospatial adjacency among the clusters.

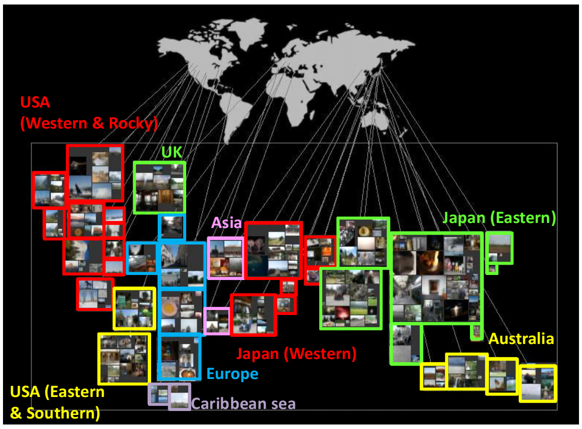

Figure 3: Result (on the XY-plane).

We had an experimental test with 12 students majoring computer science in our laboratory. We asked them to play with MIAOW and an existing 2D space-filling image browser. Then, we asked them to search for images specified by various conditions, such as "an image of Kawaguchiko-lake taken in May 2009". Here, our image collection included photographs taken at the events of our laboratory. We asked the examinees to search for photographs of the events, which should be memorized to their memories with their locations and times. We measured the times taken to search for the images using Dataset 3.

Table 1 shows maximum, minimum, and average times taken to search for specific images by examinees. This result demonstrates that MIAOW is effective to search for specific images if they are associated with locations and times to the memories of users.

Table 1: Time taken to search for specific images (sec.).

	Minimun	Average	Maximum
CAT	46.6	211.25	637.4
MIAOW	22.6	62.4	162.8

Our potential future works include as follows: sophisticated selection of representative images, smooth switch of representative images and thumbnails while zooming operations, experiments with larger image collections, numerical evaluations of the layout results, and consideration of more metadata such as photograph owners or photogenic subjects.

References

[1] A. Gomi, R. Miyazaki, T. Itoh, J. Li, CAT: A Hierarchical Image Browser Using a Rectangle Packing Technique, *12th International Conference on Information Visualization (IV08)*, pp. 82-87, 2008.