CAT: A Hierarchical Image Browser Using a Rectangle Packing Technique

Ai Gomi*, Reiko Miyazaki*, Takayuki Itoh*, Jia Li**
*Ochanomizu University, **The Pennsylvania State University
{gomiai, reiko, itot}@itolab.is.ocha.ac.jp, jiali@psu.edu

Abstract

The recent revolution of digital camera technology has resulted in much larger collections of images. Image browsing techniques thus become increasingly important for overview and retrieval of images in sizable collections. This paper proposes CAT (Clustered Album Thumbnail), a technique for browsing large image collections, and its interface for controlling the level of details (LOD). As a preprocessing, this new system applies tree-structured clustering to images based on their keywords and pixel values, and selects representative images for each cluster. When a user specifies one or multiple keywords, CAT extracts a branch of the tree structure that contains clusters described by the user-specified keywords. A hierarchical data visualization technique is developed to display the tree structured organization of images using nested rectangular regions. Interlocked to the zooming operation, CAT selectively shows representative images while zooming out, or individual images 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. Several image browsing techniques apply dimension reduction schemes for similarity-based image layout [9, 11, 12, 13]. Several others represent structures of images, such as graphs or clusters, for intelligent layout of the images [1, 3, 6]. Focus+context or zooming interfaces are also useful for image browsing [6, 12].

We had a questionnaire, as introduced in Section 3, for our preliminary discussion about what kind of image browsers are desirable for users. As a result, we decided the policy for the image browser as follows:

Policy 1: Since most of users are familiar with exploring hierarchy of file systems with GUIs, we would like to form hierarchical structures of images, and allow non-uniform depths in a hierarchical structure.

Policy 2: It is preferable to divide images according to metadata and keywords first, and then divide according to contents, to construct the hierarchy of images.

Policy 3: It is better to display many images in well-aligned, no-overlapped style, since users may look every image as if they use image browsing function of file system GUI (i.e. Microsoft Windows Explorer).

Policy 4: It is also better to display images for each cluster of images, as if famous file system GUI displays several representative images for each folder.

This paper presents CAT (Clustered Album Thumbnails), a technique for browsing clustered images, and its interface for controlling the level of detail (LOD), based on the above policies. It is assumed that unstructured collections of images are given, and the images are assigned with one or more annotation keywords. CAT clusters images according to keywords as well as pixel values. When a user specified one or more keywords, CAT constructs the subset of tree structure of images which the keywords are assigned, and then visualizes the tree structure of images using a hierarchical data visualization technique [4, 5], which represents the clusters by nested rectangular regions. While a user zooms out, it visualizes representative images of high-level clusters. Zooming in, it visualizes the images in each cluster. Figure 1 shows an example of zoom in and out states of our browsing technique.

A main feature of CAT is the effective control of the LOD based on the clustering hierarchy. For example, a user has ten thousand images, which cannot be simultaneously displayed at a reasonable resolution by a regular size of personal display unit. CAT addresses this issue by showing representative images of clusters formed at different levels of hierarchy. Specifically, it first displays representative images of the highest level clusters generated based on annotation keywords, such as flower, sky, or ocean. Then, the user can select preferable categories by viewing the representative images, and interactively zoom in the selected clusters. Upon this moment, CAT will show representative images of the lower level clusters, and the user may find visually similar groups of images, for example, red flowers and emerald ocean. By the further zoom in operation, the user can fly into the preferred cluster, and finally see individual images in the cluster. Such an operation is friendly to users because they are used to operate GUIs for file systems exploring down from the top level of the hierarchy. Also, this mechanisim is efficient in I/O

Figure 1: Overview of CAT. (Left) While a user zooms out, CAT displays representative images of higher-level clusters. (Center) While a user zooms in, CAT displays representative images of lower-level clusters. (Right) While a user further zooms in, CAT displays independent thumbnail images.

time, because CAT only loads high-level representative images first, and then only loads images in zoomed area, and free defocused images.

CAT is quite analogous to PhotoMesa [1], a famous hierarchical image browser, since it divides a display space into rectangular subregions, and places images in grid-like layout without overlapping each other. As discussed in Section 2.2, we would like to argue two features of CAT comparing with PhotoMesa, including better aspect ratios of rectangular subregions, and representation of deep or inhomogeneous hierarchical structures.

2 Related Work

This section introduces related works on image browsing and hierarchical data visualization.

2.1 Image Browsing

Many image browsing interfaces, such as image search engine Web sites, simply provide a set of images in grid layout in the ranking order of a certain similarity measure with respect to the query. This kind of interfaces is not always effective for quickly finding all the desired images.

Some image browsing techniques focus on the layout of image thumbnails so that it can finely represent content similarity among the images. Design Galleries [9] and Semantic Image Browser (SIB) [13] applied multidimensional scaling (MDS) for the similarity-based layout of image thumbnails. Rubner et al. [11] defined Earth Mover's Distance (EMD) for measurements of distances among images, and displayed images using MDS and EMD. Walter et al. presented Hyperbolic Image Browser [12] which scatters images onto a hyperbolic space applying MDS. These are good at representing distances among images; however, it often overlaps many images each other on the display. We would like to argue that users may prefer grid-like layout for the display of collection of images, since the layout never overlap images each other, and the users used to browse images in this style on the file system GUIs.

Some other known techniques focus on visualization of network of images [3, 6]. Again, we would like to argue that users may prefer hierarchical structure, since they used to explore the hierarchy of file systems on GUIs.

Bederson presented PhotoMesa [1], which places groups of images into subregions of display space. Kustanowith et al. presented a technique [8] that radially places clusters of images, and provides capabilities to interactively resize the layout for focus+context representation of the collections of images. These techniques are somewhat analogous to our technique, since it represents clusters of images, and display the images by grid-like layout. We discuss the trade-off between CAT and PhotoMesa later.

2.2 Hierarchical Data Visualization

There are many works on hierarchical data visualization, where many of them are categorized as tree-based approaches, and the others are space-filling approaches.

CAT utilizes our hierarchical data visualization method [4, 5]. It represents a hierarchy as nested rectangles, and leaf-nodes as painted icons, satisfying the following conditions: 1) It never overlaps the leaf-nodes and branch-nodes in a single hierarchy of other nodes. 2) It attempts to minimize the display area requirement. 3) It draws all leaf-nodes by equally shaped and sized icons. 4) It attempts to minimize aspect ratio and area of rectangular subspaces.

A desirable trait of our technique is the representation of lower-level data items as clickable and equally-sized thumbnails. Our goal is very similar to the goal of the Quantum Treemap, and actually Quantum Treemap has been applied as the core technology of PhotoMesa [1]. Experiments described in [4] discusses trade-offs between

Quantum Treemap and our technique, where our technique yielded better results in aspect ratio of subregions and stability of layout among similar hierarchical data. As discussed in Section 1, CAT displays representative images by mapping onto the rectangular subregions while zooming out; aspect ratio of subregion is therefore important.

Another feature of our hierarchical data visualization technique is completely equally-sized representation of leaf-nodes, even if the depth of hierarchy is deep or inhomogeneous. Bederson et al. also discussed in [2] that it is much better to display images as equally-sized and well-aligned. However, it is unclear if their technique can display images as equally-sized, if the depth of hierarchy is deep or inhomogeneous.

One more advantage of our technique is the flexible control of the placement of rectangles, as discussed in the last paragraph of Section 3.7 of this paper.

3 Presented Technique

This section presents overview and technical components of CAT.

3.1 Preliminary Discussion

We had a questionnaire about retrieval operations for image stored in personal computers of users, to discuss what kind of image browsers are familiar to users. Questions were as follows:

Question 1: How do you look for specific images stored in your computers?

Question 2: What kind of categorization of images should be primary for you?

We gathered the answers from 12 students. For the question 1, only 8 % of the answerers said that they used search engines, and 83 % of the answerers said that they mainly used GUIs for file systems (i.e. Microsoft Windows Explorer). 67 % of the answerers said that they mainly explored folders based on their memories, and 17 % of the answerers said that they mainly looked every folder and image one-by-one by using thumbnail display function of the GUIs. For the question 2, 50 % of the answerers said that metadata (i.e. date, place) should be primary, and 50 % of the answerers said that keywords (i.e. names of objects shot in the images) should be primary. All answerers agreed that contents (i.e. color, texture) might be useful, but all of them said that it should be secondary or thirdly.

Based on the above results, we led the policies for the image browser, as described in Section 1. Policy 1 reflects the above result that image browsing operation should be familiar if it is analogous to exploration of file systems. Policy 2 reflects the result that metadata and keywords should be most desirable for hierarchy construction. Policies 3 and 4 reflect the result that some users may look over every folder and image. The below presented technique is an image browsing technique satisfying the above policies.

3.2 Technical Overview

Figure 2 shows the flow of image clustering and browsing processes of CAT. Here, assume that a collection of images each assigned with one or more keywords are given, where metadata (i.e. date, place) can be given as keywords.

As a preprocessing, CAT first divides images according to their keywords, and construct higher-level clusters of images. It then divides the images in each of clusters according to their contents (colors and textures), and constructs lower-level clusters of images. After the clustering process, it selects a representative image for each cluster. Finally, it loads the above hierarchical data.

The initial screen of CAT displays a list of keywords. When a user selects one or more keywords, CAT constructs the subset of tree structure consists of images which have all the user-specified keywords. CAT then places the clusters and images onto a display space, as described in Section 3.7. CAT selectively displays representative or original images interlocking to zooming operation, for the LOD control described in Section 3.8.

3.3 Keyword-based Image Clustering

As the first step of preprocessing, CAT constructs clusters of images based on their keywords. Let the whole vocabulary of keywords be V, and the set of keywords for image X_i be W_i, where

$$W_i = \{w_{i,1}, ..., w_{i,m_i}\}, w_{i,j} \in V \qquad (1)$$

and m_i denotes the number of keywords for image X_i. If W_i and W_j are entirely equal, CAT put the images X_i and X_j into the same cluster.

3.4 Content-based Image Clustering

As the second step of preprocessing, CAT further divides images in the higher-level clusters, accordig to feature vectors calculated from color and texture.

To calculate the color part of the feature vectors, CAT first converts RGB color components into LUV or YCbCr. We experimented with several ways of forming the color part of the feature vector; as a result, current our implementation simply divides the image pixel coordinates into two dimensional grids, and calculate the average values of color components in each of grid-subspaces.

To calculate the texture part of the feature vectors, CAT uses a Daubechies 4 wavelet transform to obtain high frequency band image, and then applies a feature vector calculation scheme presented in [7] from the wavelet images.

CAT then normalizes the feature vector of each image, and calculates cosine values of all possible pairs of images. It finally generates clusters according to the cosine values, applying the bottom-up agglomerative clustering algorithm. This step may be time-consuming if the image collection is extremely large; however, it can be a batch process.

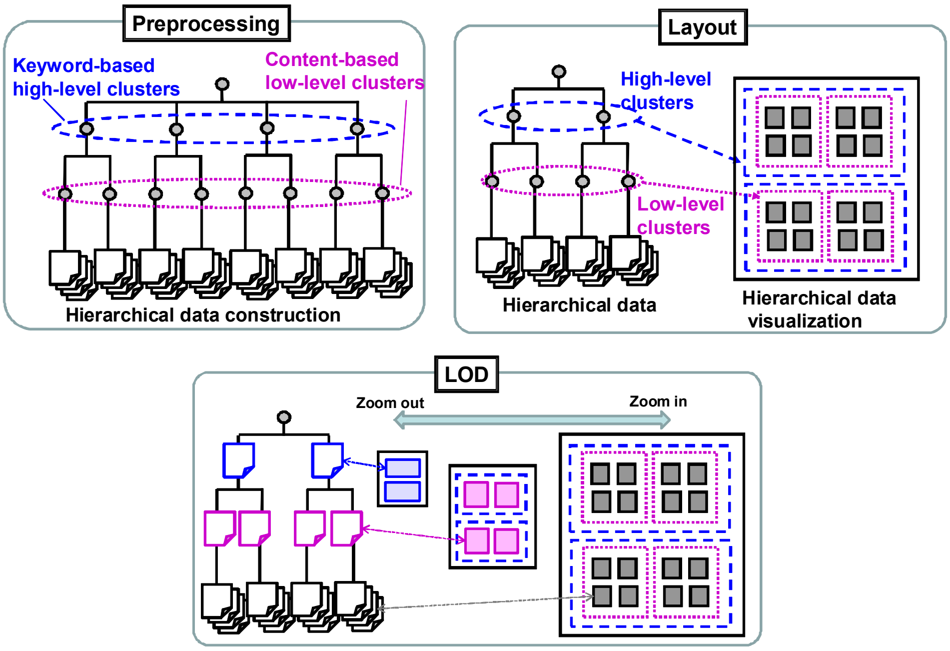

Figure 2: Overview of processing flow of CAT. (Upper-Left) Hierarchy construction as a preprocessing. (Upper-Right) Layout of hierarchically structured images. (Lower) LOD control interlocking to the zooming operation.

We think it is visually much better if we can avoid generating too small or too large clusters. Our implementation of clustering algrithm therefore aggressively merges small clusters to adjacent clusters, and avoids merging large clusters with others. However, it may by still difficult to construct good hierarchy if number of images is huge. For example, if we have ten thousands of images in a cluster, we may form one thousand clusters of ten images, or ten clusters of one thousand images. Our implementation allows applying multiple sets of thresholds to construct multi-level or inhomogeneous hierarchy. Adequately controlling the thresholds, our implementation limits the maximum number of images or clusters in their parent cluster, and adjusts the depth of the hierarchy. However, we just experimentally define the thresholds currently. Optimal threshold definition will be one of our future works.

3.5 Representative Image Selection

As the final step of preprocessing, CAT selects representative images for each cluster. There is a variety of ways to select representative images, but currently CAT simply selects according to pixel information.

In many cases, the image that is closest to the center of the cluster in the feature vector space looks average in the cluster, and therefore the image is preferable as the representative of the cluster. Our implementation therefore simply selects the image closest to the center of the cluster as the representative image.

3.6 Interactive Keyword Selection

After hierarchy construction and representative image selection as a preprocessing, CAT loads the entire hierarchical structure and paths of image files in order: however, it does not initially load any images themselves. At the moment, CAT displays a list of keywords as an initial screen, and waits for events to select one or more keywords.

As described in Section 3.3, top-level clusters of hierarchy are based on keywords of images. Let the set of keywords for cluster C_i be W_i, where

$$W_i = \{w_{i,1}, ..., w_{i,m_i}\}, w_{i,j} \in V \qquad (2)$$

and m_i denotes the number of keywords for cluster C_i. Also, let the set of user-specified keywords be S, where

$$S = \{s_1, ..., s_M\}, s_i \in V \qquad (3)$$

and M denotes the number of user-specified keywords.

When a user selects the keywords, CAT constructs a subset tree structure consists of images which have all the user-specified keywords. If the set of keywords W_i includes all keywords in S, cluster C_i will remain in the subset tree; otherwise, C_i will not remain.

3.7 Display of Hierarchically Clustered Images

As described in Section 2.2, CAT applies our hierarchical data visualization technique for image browsing. It places images based on a bottom-up packing algorithm consists of the following three phases:

Phase 1: CAT first places 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 2: CAT 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 3: CAT finally packs the rectangles of all the higher-level clusters, and encloses them by a rectangular border.

Since CAT places representative images of clusters into the rectangular borders, aspect ratios of the rectangular areas should be as close as possible to the aspect ratios of the representative images. For this requirement, CAT calculates the horizontal and vertical numbers of images in the grid layout so that the ratio of the numbers is as close as possible to the aspect ratio of the representative image of the cluster. Also, we slightly modify the condition of rectangle placement described in [4] for Phases 2 and 3, where the condition attempts to minimize the penalty value e, where $e = aA + rR + dD$, a, r, and d are user-defined positive constant values, A is the ratio of areas of rectangular border between the before and after rectangle placement process, R is the ratio of aspect ratios of rectangular border between the before and after rectangle placement process, and D is the distance between actual position of the rectangle and its ideal position described in a template.

We modify the variable R for the above requirement. CAT calculates R_1, which is the error of aspect ratio of a rectangle against the ideal aspect ratio, before the placement. Similarly it calculates R_2, which is the similar error after the placement. Here ideal aspect ratio is the aspect ratio of the representative image in Phase 2, or the aspect ratio of the window space in Phase 3. CAT calculates R as the ratio of R_2 to R_1, and attempts to minimize e.

Another requirement for the data layout is similarity-based placement of clusters. We can obtain feature vectors of clusters, from set of keywords for higher-level clusters, and pixel-based feature values for lower-level clusters. Dimension reduction schemes, such as principal component analysis (PCA) and MDS, are useful to calculate positions of clusters from the feature vectors, so that clusters containing similar images are closer on the display space. CAT has a capability to calculate the positions and record it as a template, and refer to it while placing image thumbnails and rectangular borders. Detail of template-based data layout algorithm is described in Section 5 of [4].

3.8 User Interface and LOD Control

CAT provides a user interface for zooming operation, which is somewhat similar to ZUI (Zoomable User Interface) applied by PhotoMesa. Our implementation zooms into the specific clusters by a double-click of left button of a mouse, and zooms out by a double-click of right button. It also shifts focus areas by drag operation. When the zooming operation arrived at the lowest hierarchy, it magnifies an image by a click operation.

In our implementation, CAT initially zooms out and loads only representative images of higher-level clusters from the hard disk drive into the main memory, and then loads each image thumbnails in the focused clusters on the fly, or frees memory space for image thumbnails in defocused clusters. This mechanism is effective for frame rate and memory usage.

4 Example and Evaluation

We implemented the image clustering part of CAT on GNU gcc 3.4 with OpenCV [1] 1.0, and two versions of image browsing part of CAT on Microsoft Visual C++. We tested CAT on IBM ThinkPad X60 (CPU 1.8GHz, RAM 1GB) with Windows XP, using an image collection which contains 2360 images [2] of size 384x256 and stored in JPEG format. Figure 1 is a browsing result using the same image collection, where we selected a keyword "flower".

We had several experiments with 10 examinees, where all of them were female university students. We evaluated CAT according to several criteria [3], including subjective appearance evaluation, measurement of time to search for specific images, and aspect ratio of cluster regions.

4.1 Subjective Appearance Evaluation

Examinees played five variations of CAT for several minutes, and then evaluated them by ranking. The five variations are as follows:

"no cluster": All images are placed in a grid layout.

"low-level, without-representative": Pixel-based clustering is applied to form one-level clusters. Representative images are NOT displayed.

"low-level, with-representative": Pixel-based clustering is applied to form one-level clusters. Representative images are displayed.

"two-level, without-representative": Both keyword- and pixel-based clustering are applied to form two-level clusters. Representative images are NOT displayed.

"two-level, with-representative": Both keyword- and pixel-based clustering are applied to form two-level clusters. Representative images are displayed.

We asked examinees to give ranks 1 to 5 to each of the five variations of CAT from the aspects of visual impres-

[1] Open Source Computer Vision Library, distributed at http://www.intel.com/technology/computing/opencv/.

[2] The images are provided at http://www.stat.psu.edu/~jiali/index.download.html.

[3] Detailed statistics of the evaluation is uploaded at http://itolab.is.ocha.ac.jp/~gomiai/CAT-evaluations.pdf.

sion and usability, where 1 is the best, and 5 is the worst evaluation. We found from the statistics of the answers that existence of representative images is effective for users even higher-level clustering was not applied.

4.2 Time to Search for Specific Images

Next we measured the time to search for specific images. We provided an image printed on paper to examinees, and they searched for the image from the image collection displayed by the variations of CAT. We measured the time to search for the image.

From the results, We are surprised that the "low-level" version of CAT, in which only pixel-based clustering is applied, did not drastically improve the time against "no-cluster" version of CAT very well. In other words, keyword-based clustering is very effective for image browsing using CAT. Rodden et al. proofed that visual similarity is useful information for image browsing [10]; in addition to that, our result may suggest people to search for images based on semantics, rather than visual properties such as colors and textures. The "two-level" version therefore significantly improves over the other two versions.

4.3 Aspect Ratio of Rectangular Regions

We calculated the aspect ratios of all rectangular regions. Since aspect ratios of all images are $4/3$ in our experiments, the best aspect ratio value is $4/3$ in this evaluation. We found that more than half of many clusters obtained preferable aspect ratios (during 1.2 and 1.4), and the average aspect ratio values were also very close to 4/3.

5 Conclusion

This paper presented CAT (Clustered Album Thumbnails), a technique for browsing clustered images, and its LOD control interface. This paper also provided experiments of CAT, and demonstrated good results from multiple perspectives including subjective evaluation, usability to search for specific images, and statistics of aspect ratios of rectangular regions.

We point out the following as potential future works: experiments with larger image collections, more discussion of representative image selection techniques, experiments with more various keyword and metadata, optimization and subjective evaluation of parameters described in Section 3.7, and application to debugging image annotation and clustering results, as SIB [13] addressed.

Acknowledgements: This work has been partially supported by Japan Society of the Promotion of Science under Grant-in-Aid fir Scientific Research (C) No. 18500074.

References

[1] Bederson B., B., PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps, *Symposium on User Interface Software and Technology*, 71-80, 2001.

[2] Bederson B., Shneiderman B., Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, *ACM Transactions on Graphics*, 21(4), 833-854, 2002.

[3] Chen C., Gagaudakis G., Rosin P., Content-Based Image Visualization, *IEEE Information Visualization, 13-18, 2000.*

[4] Itoh T., Yamaguchi Y., Ikehata Y., Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm, *IEEE Transactions on Visualization and Computer Graphics*, 10(3), 302-313, 2004.

[5] Itoh T., Takakura H., Sawada A., Koyamada K., Hierarchical Visualization of Network Intrusion Detection Data in the IP Address Space, *IEEE Computer Graphics and Applications*, 26(2), 40-47, 2006.

[6] Jankun-Kelly T. J., Ma K.-L., MoireGraphs: Radial Focus+Context Visualization and Interaction for Graphs with Visual Nodes, *IEEE Symposium on Information Visualization*, 59-66, 2003.

[7] Kubo M., Aghbari Z., Makinouchi A., Image Retrieval by Edge Features Using Higher Order Autocorrelation in a SOM Environment, *The Institute of Electronics, Information, and Communication Engineers, Transactions on Information and Systems*, E86-D(8), 1406-1415, 2003.

[8] Kustanowitz J., Shneiderman B., Meaningful Presentations of Photo Libraries: Rationale and Applications of Bi-Level Radial Quantum Layouts, *ACM/IEEE Joint Conference on Digital Libraries*, 188-196, 2005.

[9] Marks J., et al., Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation, *ACM SIGGRAPH*, 389-400, 1997.

[10] Rodden K., Basalaj W., Sinclair D., Wood K., Does Organisation by Similarity Assist Image Browsing?, *ACM SIGCHI 2001*, 190-197, 2001.

[11] Rubner Y., Tomasi C., Guibas L. J., A Metric for Distributions with Applications to Image Databases, *IEEE Computer Vision*, 59-66, 1998.

[12] Walter J. A., A., Webling D., Essig K., Ritter H., Interactive Hyperbolic Image Browsing - Towards an Integrated Multimedia Navigator, *ACM SIGKDD*, 111-118, 2006.

[13] Yang J., Fan J., Hubball D., Gao Y., Luo H., Ribarsky W., Ward M., Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis, *IEEE Visual Analytics in Science and Technology*, 191-198, 2006.