Group-Oriented User Interface for Digital Image Management

Munehiro Nakazato
Beckman Institute
University of Illinois at Urbana-Champaign
405 N. Mathews Ave.
Urbana, IL 61801 USA
+1-217-244-1089
nakazazato@uiuc.edu

Thomas S. Huang
Beckman Institute
University of Illinois at Urbana-Champaign
405 N. Mathews Ave.
Urbana, IL 61801 USA
huang@ifp.uiuc.edu

ABSTRACT
ImageGrouper is an Object-oriented graphical user interface for digital image management. It eases experimental query for the image retrieval as well as hierarchical arrangement of photo stocks. The concept is simple and can be applied to the other types of objects. The system is being used as a front-end for our digital image toolkit.

KEYWORD: Group-oriented interaction, direct manipulation, multimedia objects, digital images

INTRODUCTION
While many researchers have investigated user interfaces for text document searching and visualization [1][3], few research has been done for the user interface for multimedia object management [4][5]. To design a user interface for multimedia objects, we need an extra care.

First, in the text databases, the user retrieves a specific document with keywords and boolean operators. Thus, many user interfaces were designed for visual manipulation of boolean query [2][3]. Meanwhile, multimedia objects are often searched by visual similarity [6]. Therefore, the traditional user interfaces are not necessarily suitable for multimedia databases: we cannot apply AND/OR on images.

Next, in the text document systems, the documents are often represented by icons. However, in order to examine the contents, the user had to see the detail in another window. On the other hand, multimedia objects are usually visible as they are. Thus, the users should be able to manipulate the objects directly with the thumbnails of images or key frames of videos. *Direct manipulation* [3] is applicable to multimedia objects more naturally than texts.

THE IMAGEGROUPER
We are developing *ImageGrouper*, a user interface for digital image retrieval and arrangement. In *ImageGrouper*, we introduced a new concept, *Group-Oriented Interaction* (GOI.) In this method, the user conducts various tasks by dragging and grouping objects on the workspace.

Figure 1. shows the workspace of *ImageGrouper.* The workspace is divided into two panes. The left pane (*Result View*) is used to show the results of various image search methods. Currently, the system supports *random retrieval*, *similarity-based search* and *keyword-based search*. Similarity-based retrieval is often called *content-based image retrieval* (CBIR.) The right pane (*Group Palette*) is the workspace where the user manipulates image groups. Both panes can be resized individually. Only the *result view* is scrollable.

To make a visual similarity query, a user first drags sample images from the left pane to the right pane. Then, s/he creates a group by drawing a rectangle that encloses those images. When s/he presses the query button, the system retrieves images that are similar to the sample images. The user can also create a group of negative sample images. The system avoids retrieving those images which are similar to the negative examples. The user can create multiple groups on the workspace. The image groups are also used for annotating keywords and arranging images in the database as described below.

THE BENEFITS OF GROUP-ORIENTED QUERY
In many user interfaces for image retrieval [5][6], both queries and results are displayed on the same workspace. Therefore, the user's query is overwritten by the result display. We separated the workspace into two panes to pre-

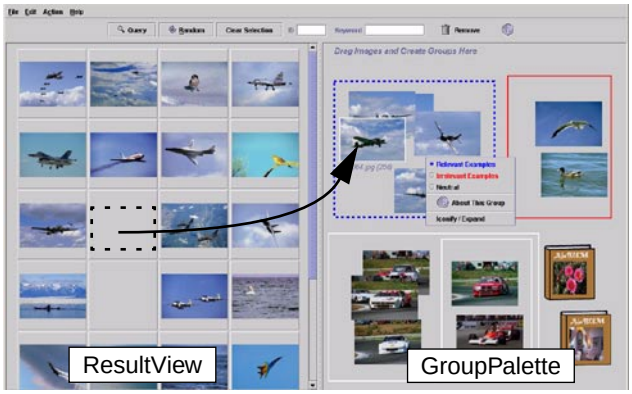

Figure 1: ImageGrouper workspace. Rectangles in the right pane are image groups.

serve the user's query specification so that s/he can always know what s/he did. This approach makes experimental query much easier: the user can try different combinations of query examples quickly by moving images between the groups on the workspace.

Experimental query is very important in similarity-based image retrieval for several reasons. First, unlike text retrieval where a query is a set of keywords, a query of image retrieval is a set of sample images. Thus, it is harder to remember query specifications made. Second, since those systems are using low-level visual features such as color and texture, similarity in these features is not necessarily equivalent to similarity in the user's high-level concepts such as "car." Hence, the results are less predictable. This problem is known as "semantic gaps [5]"

ADDITIVE BULK ANNOTATION AND HIERARCHY MANAGEMENT

The group-oriented user interface has also several benefits for digital image arrangement. First of all, annotating keywords on a large number of images is a tedious task. The *ImageGrouper* allows the user to annotate keywords not on individual images but on image groups. This realizes *bulk annotation* [4]. Furthermore, by dragging a new image into an annotated group (rectangles in Figure 1.) the new image is automatically annotated with the same words (*additive annotation*.) In the other systems, once bulk annotation is done, there was no way to annotate the same set of keywords on new images quickly. Annotated keywords are used for keyword-based retrieval.

Next, the most common user interface to manage hierarchy is a tree view. Accordingly, most GUI toolkits provide APIs for trees. Our group-based user interface provides a new way to create a hierarchy among objects. In Figure 2. images of various types of cars are collected on the workspace. By grouping all images and annotating this group with "car," all images belong to a class of "any types of car." Within this class, the user can create a subclass of images by grouping a subset of the images with a new rectangle. In the figure, images are separated into two subclasses: red cars and other cars. By annotating the left subgroup with "red," this subclass automatically becomes a class of "red cars." The right diagram in Figure 2. is the generated hierarchy. Visual hierarchy management by groups is not limited to images. The concept can be also applied to hierarchy creation of other objects.

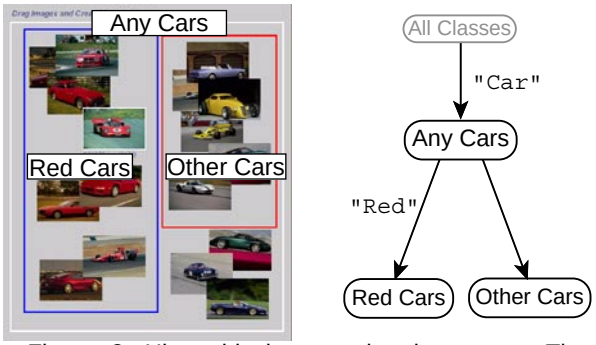

Figure 2: Hierarchical annotation by groups. The right figure is generated hierarchy.

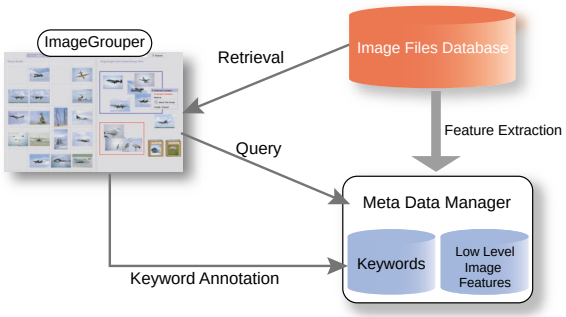

Figure 3: The digital image toolkit

IMPLEMENTATION

ImageGrouper is written in Java2 with Swing API. Currently, it is used for a front-end of our digital image toolkit (Figure 3.) In this system, *Meta-data manager* extracts visual features from an image stock and compares images based on these features. The distance metric is dynamically adjusted based on group configuration on *ImageGrouper* [6]. The meta-data manager also maintains keywords that are annotated by *ImageGropuer*.

CONCLUSION AND FUTURE WORK

ImageGrouper enables various tasks in simple and consistent operations. It makes experimental query and additive bulk annotation easier. In addition, it provides a new visual way to create a hierarchy. Although ImageGrouper is developed for multimedia objects, the concept of *group-oriented interaction* is also applicable to other types of objects.

We are investigating hierarchy management of our user interface. Group-oriented GUI is suitable for creating a new hierarchy from scratch, while the traditional trees are suitable for browsing a deep hierarchy. We plan to integrate the group-oriented view and the tree view so that the user can go back and forth between two views.

An online demo of *ImageGrouper* is available at
http://www.ifp.uiuc.edu/~nakazato/grouper

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation Grant CDA 96-24396.

REFERENCES

1. Card, S.K., Mackinlay, J.D., and Shneiderman, B. (Editors) Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann, 1999.
2. Cousins, S.B., et al. The Digital Library Integrated Task Environment (DLITE). In ACM Digital Libraries, 1997.
3. Jones, S. Graphical Query Specification and Dynamic Result Previews for a Digital Library. In UIST'98.
4. Kuchinsky, A. et al. FotoFile: A Consumer Multimedia Organization and Retrieval System. In Proc. of CHI'99.
5. Santini, S. and Jain, R. Integrated Browsing and Querying for Image Database. IEEE Multimedia, July 2000.
6. Zhou, X. S. et al. Comparing Discriminating Transformations and SVM for Learning during Multimedia Retrieval. In Proc. of ACM Multimedia'01.