

ABSTRACT

Title of Dissertation: IMAGE MANAGEMENT USING PATTERN RECOGNITION SYSTEMS

Bongwon Suh, Doctor of Philosophy, 2005

Dissertation Directed By: Associate Professor Benjamin B. Bederson,
Department of Computer Science

With the popular usage of personal image devices and the continued increase of computing power, casual users need to handle a large number of images on computers. Image management is challenging because in addition to searching and browsing textual metadata, we also need to address two additional challenges. First, thumbnails, which are representative forms of original images, require significant screen space to be represented meaningfully. Second, while image metadata is crucial for managing images, creating metadata for images is expensive. My research on these issues is composed of three components which address these problems.

First, I explore a new way of browsing a large number of images. I redesign and implement a zoomable image browser, PhotoMesa, which is capable of showing thousands of images clustered by metadata. Combined with its simple navigation strategy, the zoomable image environment allows users to scale up the size of an image collection they can comfortably browse.

Second, I examine tradeoffs of displaying thumbnails in limited screen space. While bigger thumbnails use more screen space, smaller thumbnails are hard to recognize. I introduce an automatic thumbnail cropping algorithm based on a computer vision saliency model. The cropped thumbnails keep the core informative part and remove the less informative periphery. My user study shows that users performed visual searches more than 18% faster with cropped thumbnails.

Finally, I explore semi-automatic annotation techniques to help users make accurate annotations with low effort. Automatic metadata extraction is typically fast but inaccurate while manual annotation is slow but accurate. I investigate techniques to combine these two approaches. My semi-automatic annotation prototype, SAPHARI, generates image clusters which facilitate efficient bulk annotation. For automatic clustering, I present hierarchical event clustering and clothing based human recognition. Experimental results demonstrate the effectiveness of the semi-automatic annotation when applied on personal photo collections. Users were able to make annotation 49% and 6% faster with the semi-automatic annotation interface on event and face tasks, respectively.

Image Management Using Pattern Recognition Systems

By

Bongwon Suh

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Professor Benjamin B. Bederson, Chair and Advisor

Professor David W. Jacobs

Professor Kent Norman

Dr. Catherine Plaisant

Professor Ben Shneiderman

© Copyright by

Bongwon Suh

2005

Dedication

To Won Kyung and Little William

Acknowledgements

First and foremost, I would like to thank my advisor Dr. Ben Bederson for his unwavering support, guidance, and patience throughout my research. Working with such an outstanding mentor and a great person has been a wonderful experience. Ben taught me what research is and how to do good research. I thank him for helping me to become a researcher, and most importantly, an independent thinker.

I would also like to thank other members of my dissertation committee, Dr. Ben Shneiderman is a source of inspiration, and I want to thank him for his encouragement and enthusiasm. I am grateful to Dr. David Jacobs for his invaluable support. Not only is he on my dissertation committee, but he also guided me to gain experience on computer vision research. I want to thank Catherine Plaisant for constructive discussions when I was designing user studies. I also want to thank Dr. Norman for his valuable comments on my dissertation.

Thanks to thumbnail team members at Palo Alto Research Center. I want to thank Allison Woodruff for her wonderful mentoring and support. I would also like to thank Alyssa Glass and Ruth Rosenholtz.

It was a pleasure working with all these wonderful people. I would like to thank all the members of Human Computer Interaction Laboratory for sharing all the knowledge, stimulating discussions, and participating in my user studies. Thanks to Alex Aris, Aaron Clamage, Allision Druin, Jerry Fails, Lance Good, Francois Guimbretiere, Jesse Grosjean, J-P Hourcade, Hilary Browne Hutchison, Hyunmo

Kang, Bill Kullas, Jack Kustanowitz, Bongshin Lee, Sabrina Liao, Jaime Montemayor, Anne Rose, Jinwook Seo, and Haixia Zhao.

Special thanks go to my Korean friends in Computer Science departments who made my life as a graduate student enjoyable. I want to thank them all for being wonderful friends. I wish to express my appreciation to Kyongil Yoon and Bohyung Han for their help and invaluable comments on my research. I also thank Haibin Ling for his wonderful ideas and hard work.

Thanks to my friend, Edmund Injae Lee for his editorial comments. I wish all the best to him and his family, especially little Angie.

I am also especially grateful to my family. Thanks to my parents for their love and supports. My boy, William has been an unlimited source of happiness of my life. I thank him for all the laughter he has brought to me.

Finally, and perhaps most important, I would like to thank my wonderful wife Won Kyung for her endless love, support, encouragement, and inexhaustible belief in me. I would never have completed this work without her support.

Table of Contents

Dedication	ii
Acknowledgements.....	iii
Table of Contents	v
List of Tables	viii
List of Figures	ix
Chapter 1 Introduction.....	1
1.1 Image Management – The Problem.....	1
1.2 Research Components.....	4
1.2.1 Zoomable User Interface.....	5
1.2.2 Automatic Thumbnail Cropping	7
1.2.3 Semi-Automatic Annotation.....	8
1.3 Dissertation Overview	9
Chapter 2 Related Work	11
2.1 Digital Image Browsing and Searching	11
2.2 Zoomable User Interfaces	26
2.2.1 Fundamentals	26
2.2.2 Jazz: A Zoomable User Interface Toolkit.....	27
2.3 Treemap Algorithm.....	29
2.3.1 Fundamentals	29
2.3.2 Quantum Strip TreeMap	31
2.4 Annotation and Metadata.....	32
2.5 Saliency and Thumbnail Cropping	35
2.6 Automatic Event Identification.....	37
Chapter 3 Preliminary Work: PhotoMesa and Its Applications	41
3.1 Overview.....	41
3.2 Multi-level Thumbnails	44

3.3 Implementation Issues	46
3.4 ZPhotoMesa Component	54
3.4.1 ZPhotoMesa Component Interface	54
3.5 Integration with Other Applications	59
3.5.1 International Children's Digital Library (ICDL)	59
3.5.2 Maryland Interactive System for Image Searching	62
3.6 Web Deployment and Other Applications	69
3.6 Summary and Discussion.....	71
 Chapter 4 Automatic Thumbnail Cropping.....	72
4.1 Saliency and Thumbnails.....	73
4.2 Saliency Based Thumbnail Cropping	74
4.2.1 Find Cropping Rectangle with Fixed Threshold using Brute Force Algorithm	74
4.2.2 Find Cropping Rectangle with Fixed Threshold using Greedy Algorithm	75
4.2.3 Find Cropping Rectangle with Dynamic Threshold	77
4.3 Face Detection Based Thumbnail Cropping	78
4.4 User Study Design	80
4.4.1 Participants.....	80
4.4.2 Image Sets.....	81
4.4.3 Thumbnail Techniques.....	83
4.4.4 Recognition Task	84
4.4.5 Visual Search Task	86
4.5 Recognition Task Result	88
4.6 Visual Search Task Result	92
4.7 Summary and Discussion.....	96
 Chapter 5 Semi-Automatic Photo Annotation.....	99
5.1 Metadata and Annotation.....	100
5.1.1 Metadata Acquisition	100
5.1.2 Metadata for Personal Photos	102
5.2 Semi-Automatic Annotation	104

5.3 Semi-Automatic Annotation Design Principles.....	105
5.4 Semi-Automatic Photo Annotation and Recognition Interface (SAPHARI) .	107
5.5 Event Identification.....	110
5.5.1 Event Hierarchy	111
5.5.2 Update Event Boundaries	112
5.5.3 Event Identification Algorithm	116
5.5.4 Annotation Strategy	121
5.6 Clothing Based Human Recognition	124
5.6.1 Face Recognition for Personal Photos	124
5.6.2 Human Model	126
5.6.3. Annotation Strategy	132
5.7 Semi-automatic Annotation User Study	134
5.7.1 Participants.....	135
5.7.2 Method	136
5.7.3 Event Task	138
5.7.4 Face Task	140
5.7.5 Event Task Result	142
5.7.6 Face Task Result.....	145
5.7.7 Subjective Satisfaction.....	148
5.8 Summary and Discussion.....	151
Chapter 6 Conclusion	155
6.1 Summary of Work and Contributions.....	155
6.2 Future Work	158
Appendix A User Study Material	162
A1. Consent Form Used for Automatic Thumbnail Cropping User Study.....	162
A2. Pre-user study Questionnaire for Semi-automatic Annotation Interface User Study	163
A3. Post-user study Questionnaire for Semi-automatic Annotation Interface User Study	165
Bibliography	166

List of Tables

Table 4.1 Design condition. 3X3 within subject factorial design. Two conditions were omitted because they are not applicable.	81
Table 4.2 Ratio of cropped to original image size.....	84
Table 4.3 Analysis results of Recognition Task (Paired T-Test). Every curve in Figure 4.8 is significantly different from each other.....	90
Table 4.4 List of ANOVA results from the visual search task	94
Table 5.1 Acquiring metadata associated with images	100
Table 5.2 Participants Information	136
Table 5.3 Four types of tasks were designed to compare the semi-automatic annotation strategy with conventional manual annotation approaches.....	137

List of Figures

Figure 1.1 Two different representations of the same folder. The left shows image files in the <i>detail view</i> mode of Microsoft Windows Explorer. The image files are represented as a list of files with additional information such as size, type and date. The right shows the same folder in the <i>thumbnail</i> mode. Although users can easily identify the content of the images, users are limited to view less than half the number of files compared with the <i>detail view</i>	3
Figure 1.2 PhotoMesa	6
Figure 2.1 FotoFile image organization and retrieval system [42].....	12
Figure 2.2 Adobe PhotoShop Album. Keyword tags can be dragged and dropped onto photos to associate them with keyword. Users can customize keyword tags and use them to find photos later.....	13
Figure 2.3 PhotoFinder. Users can annotate photos with drag-and-drop interface, also known as direct annotation. The name of a person can be dragged from the list onto photos. This annotation is used for keyword search for finding photos....	14
Figure 2.4 ACDSee™ Image Browser.....	16
Figure 2.5 Personal Digital Historian (PDH) from MERL [62]	17
Figure 2.6 Flamenco Interface [74].	19
Figure 2.7 PhotoTOC [56] user interface. The left panel shows representatives photos of clusters. As users click a cluster in the left panel, the right panel scrolls so that the first photo of the cluster should be shown on the screen with red borders...	20
Figure 2.8 Apple iPhoto [33]. Users can select and drag photos from the main screen onto the icon representing an album as shown in the right image.	21
Figure 2.9 Microsoft Office 2003 Picture Manager. Left: Filmstrip view, Right: Thumbnail View.	22
Figure 2.10 Picasa image browser [54].....	23
Figure 2.11 Face annotation interface for FXPAL prototype [28]	25
Figure 2.12 An example scene graph structure. The partial scene graph example on the right side is represented on the screen as shown in the left figure.....	28

Figure 2.13 The slice and dice treemap layout. The left image shows a hierarchical application of the treemap algorithm. The right image shows a single level treemap.....	30
Figure 2.14 Low aspect ratio layouts. Shading indicates order, which is not preserved.	31
Figure 2.15 Strip treemap algorithm applied to 20 rectangles.....	31
Figure 2.16 MiAlbum interface. Users are allowed to input relevance feedback by clicking thumbs-up and thumbs-down icon on the lower right corner of each image.....	33
Figure 2.17 FXPAL Photo Application [27].....	38
Figure 3.1 Detail view (zoomed-in view) of PhotoMesa.....	42
Figure 3.2 PhotoMesa software architecture	47
Figure 3.3 Previewing an image under the mouse cursor.....	52
Figure 3.4 Adding <i>ZPhotoMesa</i> component inside a Java JPanel.	55
Figure 3.5 <i>PhotoMesaData</i> is a data type to hold information of images. It can be independently prepared without any restriction. PhotoMesa scene graph is built based on this information.....	56
Figure 3.6 An example of linking a <i>ZPhotoMesa</i> component with a <i>PhotoMesaData</i> object. A statement, <i>photomesa.layout(data)</i> ; enables <i>ZPhotoMesa</i> to build a scene graph by using information stored in <i>PhotoMesaData</i> and to show the images on the screen.	58
Figure 3.7 International Children’s Digital Library (ICDL) query interface.....	59
Figure 3.8 PhotoMesa is embedded as an image browser inside ICDL.	60
Figure 3.9 ICDL book reading interface. The example shows the Comic Strip reader out of three readers.....	61
Figure 3.10 ISIS (Interactive System for Image Searching) interface. Search results are shown inside a long html page. Users have to scroll up and down to examine images in the results.....	62
Figure 3.11 PhotoMesa ISIS. This figure shows an example of dynamic query preview. As a user types in a keyword, images that have matching metadata are highlighted so that users can easily identify patterns in results.	65

Figure 3.12 Double slider for specifying time conditions	66
Figure 3.13 PhotoMesa ISIS search options	66
Figure 3.14 Grouping and Searching options	67
Figure 3.15 Dynamic Grouping.	68
Figure 3.16 PhotoMesa can be run in a web browser	69
Figure 3.17 PhotoMesa is adapted to build a virtual microscope.	70
Figure 4.1: An example saliency map.....	73
Figure 4.2: A cropped image from the previous example (Figure 4.1) and thumbnails from the original image and the cropped image	74
Figure 4.3: Greedy Cropping algorithm.....	76
Figure 4.4: The solid line represents the area-threshold graph. The dotted lines show the process of searching for the best threshold. The numbers indicate the sequence of searching	78
Figure 4.5 Left: An example face detection cropping. Original image (A) and face detection result (B). Right: Comparing three types of thumbnails. Plain shrinking (D), saliency based cropped thumbnail (E), and face-detection based cropped thumbnail (F).	79
Figure 4.6 Recognition task interfaces. Participants were asked to click what they saw or the "I'm not sure" button. Left: Face Set recognition interface, Right: Animal Set recognition interface	85
Figure 4.7 Visual search task interface. Participant were asked to find an image that matches a given task description. Users can zoom in, zoom out, and pan freely until they find the right image.....	87
Figure 4.8 Recognition Task Results. Dashed lines are interpolated from jagged data points.....	89
Figure 4.9 Visual search task results.....	94
Figure 5.1 The information flow cycle of semi-automatic annotation	104
Figure 5.2 SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface)	109
Figure 5.3 An example event hierarchy. The units in the upper row represent coarsely grouped events and the units in the lower row are tightly grouped events.....	112

Figure 5.4 An example event hierarchy. When a node “May 2 nd ~3 rd ” is to be added into the hierarchy, it can be either i) merged into the previous period, ii) merged into the next period, or iii) separated as an independent node.....	114
Figure 5.5 The example event hierarchy shown in Figure 5.4 is changed after being updated by a user. The left example shows the result after merging the “May 2 nd - 3 rd ” event into the previous group, “Birthday Party” (denoted as case <i>i</i>) in Figure 5.4). In the right, photos of the “May 2 nd - 3 rd ” event are merged into the next group, “My Camping Trip” (denoted as case <i>ii</i>) in Figure 5.4).	115
Figure 5.6 Pseudo code for building hierarchical event clusters	118
Figure 5.7 Merging two adjacent events.....	119
Figure 5.8 The upper shows a result from event identification with a coarse granularity where all images from one day are identified as a single event. The bottom shows event grouping with a finer granularity. Different levels of events can be obtained by changing the identification granularity. The <i>K</i> and <i>d</i> values on the right side are constants used to detect clusters, which is used in Figure 5.6.....	120
Figure 5.9 Annotation by drag-and-drop. Users can drag a text label onto a photo or a group of photos to make annotations.....	121
Figure 5.10 Fixing event boundaries which have been automatically identified.	123
Figure 5.11 Locating clothing from detected faces	127
Figure 5.12 More weight is given to the upper center part of clothing.	128
Figure 5.13 Human model based on clothing	130
Figure 5.14 People in photos are cropped and laid out on the screen grouped by their clothing similarities. People who wear similar clothing are clustered together.	131
Figure 5.15 Make bulk annotations by drag-and-dropping a name label on a face group.....	132
Figure 5.16 Fix a misclassified face image. Moving a face image into a different face group updates the association between the face image and the name of the person.....	133
Figure 5.17 Context menu for the clothing (face) group layout.	134

Figure 5.18 Event tasks with two different settings. Left: photographs are grouped by events which have been automatically identified by SAPHARI. Right: photos are laid out by using participants own directory structure.....	139
Figure 5.19 Face annotation task interfaces. Participants were asked to annotate people in photos with two different interfaces. Left: Clothing based annotation. Right: Manual Annotation	140
Figure 5.20 The relationships between the time per annotation and the total of annotations per participants with the two different user interface techniques. Due to the bulk annotation, time per annotation has a tendency to decrease as the total number of annotations increases.....	142
Figure 5.21 Time per annotation results from the event tasks. The left figure shows individual performance of participants and the right figure shows the average and the standard deviation of time spent per annotation.....	143
Figure 5.22 Time per annotation results for the face task. The left shows individual performance of participants and the right shows the average and the standard deviation of time spent per annotation.....	145
Figure 5.23 Time per annotation with two different user interfaces. While the left scatter plot does not have any noticeable pattern, the right graph shows a clear decreasing pattern as the number of annotated faces increases.	148
Figure 5.24 The result of participants' subjective satisfaction which was measured by the post-user study questionnaire.....	149
Figure 6.1 Expected relationship between the accuracy of automatic recognition systems and users' annotation performance.	160

Chapter 1

Introduction

A Picture is Worth a Thousand Words. – Fred Barnard¹

Images have been a crucial medium for information sharing and communication even before the invention of letters. While signs and languages take a larger role in everyday communication, images are still used widely as a fundamental way of communication. They are more intuitive and sometimes contain more information than other media.

As the use of digital image devices such as digital cameras and video recorders becomes more popular [10], more images are created and stored in personal computers and shared over the Internet. As the volume of images one person needs to handle increases, it becomes a challenge to manage them. This has created the demand for computing tools which can efficiently organize, search, browse and distribute images.

1.1 Image Management – The Problem

Image management tools share the same principles of general document management systems. They include the ability to index, organize, search, browse and share

¹ The phrase was first used in a trade journal of the printing press, "Printer's Ink". Fred Barnard, then editor of the magazine, coined it in 1921.

documents. These principles can be easily implemented in an image management tool given that the images have supporting data such as captions and keywords. However, this type of information doesn't exist for all images and creating such data is often slow and tedious.

My research identifies two additional challenges that are required to support an efficient image management system – thumbnail presentation and metadata acquisition. Each challenge is detailed below.

- Thumbnails and the limited screen space

The use of thumbnails is one of the most popular techniques to show images on the computer screen. Thumbnails, created by shrinking original images, are easy to generate and are very intuitive. However, as shown in Figure 1.1, thumbnails reduce the density of information available on a screen – there are over twice as many items shown on the screen with *detail view* mode as *thumbnail* mode. Low information density requires a user to perform additional actions such as scrolling down or clicking “Next” button. Rodden, *et al.* [58] observed that users prefer to see a large number of images at once.

On the other hand, increasing information density, i.e. reducing the size of a thumbnail, causes another problem - small thumbnails often become illegible.

The challenge of this research lies in providing a solution to these contradicting requirements.

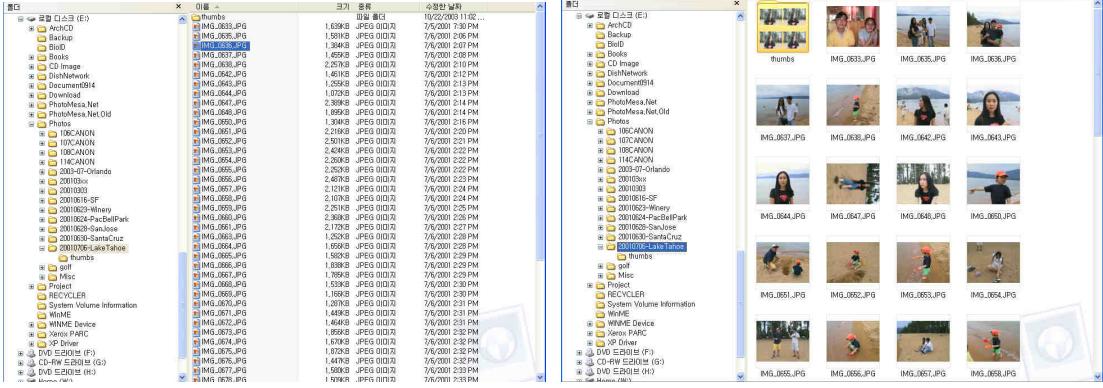


Figure 1.1 Two different representations of the same folder. The left shows image files in the *detail view* mode of Microsoft Windows Explorer². The image files are represented as a list of files with additional information such as size, type and date. The right shows the same folder in the *thumbnail* mode. Although users can easily identify the content of the images, users are limited to view less than half the number of files compared with the *detail view*.

- Lack of metadata

Unlike other textual types of documents which are typically composed of alphanumeric characters, images are usually a stream of color pixels. It is, therefore, not easy to extract metadata directly from images. For example, it is relatively difficult to automatically extract the metadata “cat” from a picture of a cat. In many cases, extensive computation is required to detect meaningful information within images.

There has been various research about automatically extracting metadata from images. The research has focused on areas such as object identification, face detection/recognition, content-based categorization, and so on [73][75][76]. However,

² Windows Explorer is the registered trademark of the Microsoft Corporation.

automatic metadata extraction is often inaccurate and irrelevant. The irrelevancy rises due to the fact that the limited amount of extracted metadata. The obtained metadata may be too general to satisfy the need of every individual user. Each user needs various types of metadata according to his/her own interest. Furthermore, there are numerous cases where it is even impossible to automatically obtain metadata without the intervention of humans. Extracting event information about which a picture was taken, such as a birthday party, is a good example.

The actual users, as information consumers, can function as the most reliable source of accurate and relevant metadata associated with images. But, it is well known that most users are not motivated enough to spend much time creating and annotating metadata for images [58]. Some researchers have tried to enhance the manual image annotation process [39]. However, users still found it tedious to make annotations on their photographs.

The research challenge is to design an easy-to-use, fast annotation system which is capable of helping users generate accurate metadata with low manual effort.

1.2 Research Components

As stated in the previous section, I have identified two important challenges for designing user interfaces for image management. My approach to these problems is to integrate improved automatic recognition systems with novel user interfaces. This strategy helps achieve my research goal of designing intuitive, efficient and enjoyable image management systems.

My research is composed of three components; (1) applying zoomable user interface techniques to the image browsing environment; (2) enhancing thumbnails so that they can be more useful within a limited screen space; and (3) designing and evaluating semi-automatic annotation strategies for personal photo collections. I discuss these in more detail in the following sections.

1.2.1 Zoomable User Interface

Conventional image browsers often use the WIMP (Windows, Icons, Mice and Pointing) style interface - the direct manipulation interface using the desktop metaphor. They arrange *folders* on the screen as shown in Figure 1.1. Typically, users navigate through images by opening and closing folders. Unless images are well organized inside folders, users may need to comb through several folders before they are able to locate a specific image.

Zoomable User Interfaces (ZUI) use a metaphor designed as a successor to the desktop interface. Compared to the desktop interface where the 2D space does not have any depth, ZUIs enable users to move their point of view with depth. The primary navigation techniques for ZUIs are zooming and panning. Users can zoom and pan to any specific area in 2D space. The animation that occurs during zooming and panning helps the user to remember where things fit together based on spatial relationships. The spatial relationship is reduced in a folder-based desktop, which relies on the user's ability to recall specific information about folders.

My research on zoomable image browsing is based on concepts introduced by Bederson [4]. He applied zoomable user interface techniques into an image browsing environment as a solution to increase the browsability of image retrieval systems. For my research, I started by enhancing PhotoMesa [4], a zoomable image browser (Figure 1.2), and applying zoomable browsing techniques to several front-end interfaces of image retrieval systems. I defined a set of programming interfaces so that other applications can embed PhotoMesa as one of their internal components. The enhanced PhotoMesa can handle richer metadata information through the interface. These features allow complex searches, fast query previewing, and dynamic query refinement.

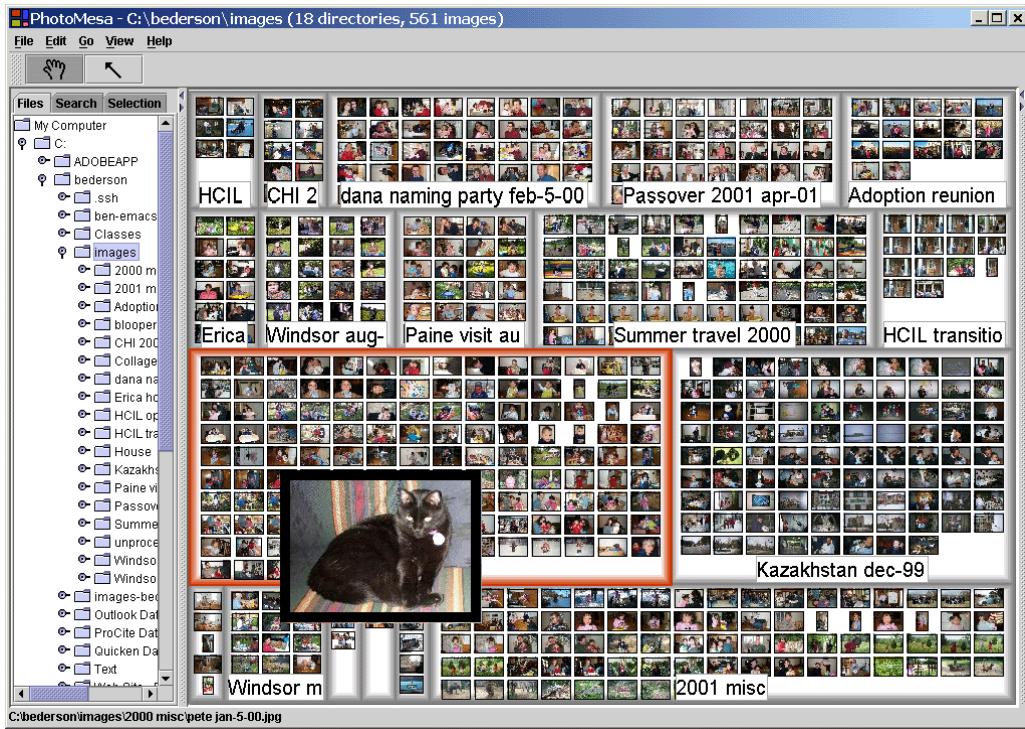


Figure 1.2 PhotoMesa

1.2.2 Automatic Thumbnail Cropping

While thumbnails - generated by shrinking the original image - are one of the most widely used techniques for representing images, they are often rendered too small and illegible. To increase the legibility of small thumbnails, I studied how to detect key components of images so that intelligent cropping, prior to shrinking, can render objects more recognizable. Along with colleagues at the University of Maryland, I developed and evaluated two automatic cropping techniques: 1) based on a method that detects salient portions of general images, and 2) based on automatic face detection.

The general thumbnail cropping method, which is based on a saliency model, finds the informative portion of images and cuts out the non-core part of the image. Cropped thumbnails increase the users' ability to recognize the image and help the users' visual search. This technique is general and can be used without any prior assumption about images since it uses only low level visual features such as color, brightness and orientation (see Chapter 4). Additionally, this technique also reduces the over or under cropping of an image by analyzing the visual content of the image.

When semantic information such as a face is available, we are able to target the crop area more effectively. Keeping a face (or faces) visible in a thumbnail is critical in identifying the people in it. Face detection based cropping demonstrates how semantic information can be used to enhance thumbnail cropping.

We also performed a study that shows strong empirical evidence that users recognize cropped thumbnails more accurately. We also show that using cropped thumbnails increases users' visual search performance.

1.2.3 Semi-Automatic Annotation

Annotation is defined as a process which involves labeling the semantic content of images (or objects in images) with a set of keywords or semantic information. Annotated information is very important for image retrieval since it allows keyword-based search and helps organizing photos. There are roughly three ways of acquiring metadata for images. They are 1) automatic extraction through image analysis, 2) manual annotation, and 3) semi-automatic annotation such as suggested in [71]. Automatic metadata extraction by analyzing images is typically fast but often generates inaccurate and irrelevant results, while manual annotation is slow but accurate. Semi-automatic annotation combines the two approaches. Initial metadata obtained automatically is updated incrementally by relevance feedback from users.

When the metadata has reasonable accuracy, for example, when the amount of erratic information is less than that of correct information, the process of correcting errors can be faster and easier than adding new information from scratch. The process of correcting errors can be even faster as in many cases where users can focus on the important errors and disregard the less important ones.

The goal of my research is to provide users with an efficient and accurate annotation mechanism using the semi-automatic approach and prove its validity. A proper

interface is very important when dealing with automatic suggestions from a system. Fixing many errors tends to frustrate users very easily. I focus on transparent automatic suggestion, in which users have total control over the annotation process.

To achieve these goals, I designed and implemented a semi-automatic annotation prototype, SAPHARI (see Chapter 5). The goal of SAPHARI is to provide an annotation framework which helps users to make accurate annotations with less effort than manual annotation. SAPHARI generates image clusters which facilitate efficient bulk annotation. SAPHARI automatically creates these clusters with hierarchical event clustering and clothing based human recognition. Experimental results demonstrate the effectiveness of the semi-automatic annotation when applied on personal photo collections.

1.3 Dissertation Overview

Chapter 2 discusses related work and the background in image management, image browsing environments, zoomable user interfaces, and related automatic recognition systems.

Chapter 3 presents my work on a zoomable image browser, PhotoMesa. Detailed design challenges are described along with the explanation of the software architecture.

Chapter 4 explains two innovative automatic thumbnail cropping techniques. While small thumbnails are expected in devices with limited screen space or in a zoomed out view of a zoomable user interface, thumbnails easily become illegible. I present

how to create useful thumbnails and evaluate its effectiveness through a series of user studies.

Chapter 5 introduces the semi-automatic photo annotation strategy. I explain the design and implementation of a semi-automatic annototatation prototype, SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface). I discuss how SAPHARI serves as a semi-automatic annotation tool for personal photo collections. In chapter 5, I also report a series of user studies on the semi-automatic photo annotation strategy. I evaluate the effectiveness and usability of SAPHARI through semi-controlled experiments and observational user studies.

Chapter 6 summarizes the main findings of this research and discusses future work.

Chapter 2

Related Work

There have been a number of research prototypes and commercial products to support image management on computers. In this chapter, I explain features of notable image management applications and prototypes. I pay particular attention to searching and browsing as well as annotation strategies of each application. In addition, I describe the technologies which my research is based on. I detail zoomable user interfaces (ZUI), treemaps, and saliency algorithms. I also present a number of automatic recognition techniques which are utilized for extracting useful information from images.

2.1 Digital Image Browsing and Searching

FotoFile is a prototype system for multimedia organization and retrieval [42]. Through informal user studies, Kuchinsky *et al.* [42] found that: 1) users did not want to spend a lot of time organizing their photos with keyboard annotations; and that 2) they wanted to browse through photos, not just perform direct search activities. To facilitate easy annotation, they added bulk annotation and face recognition in their prototype. In their bulk annotation method, users select multiple images on the display, choose attribute/value pairs from a menu, and then press the “Annotate” button so that users can add the same set of keywords on many images at the same time. FotoFile also added a facial feature extraction tool to recognize faces in photos.

This tool allows users to assign a name to a face, and then automatically annotates new photos when the same face is recognized, freeing users from having to do this annotation themselves.

As shown in Figure 2.1, FotoFile allows users to browse photos grouped in albums. When an album is selected, images in the album are laid out on the screen for viewing and editing. When there are more photos in one album than the screen can hold, photos are partitioned into many pages and users can see additional photos by pressing the next page button. As well as its standard interface, FotoFile also added the ability to visualize photos with a hyperbolic tree [43] built from the values of various metadata facets applied to a set of photos. In both ways, the interfaces let users navigate through photos without performing searches. Kuchinsky *et al.* also noted that people like to tell stories with photos and allowed users to create small groups of photos called “scraplets” to represent single narrative episodes.

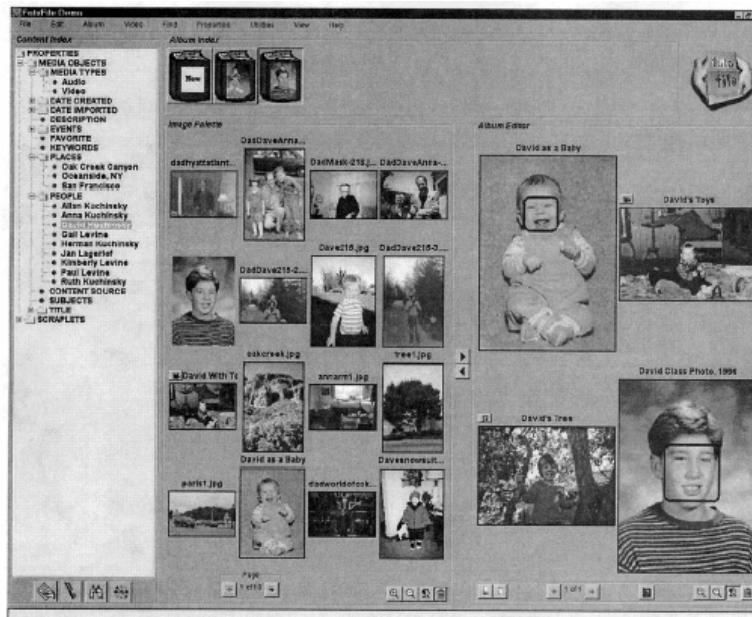


Figure 2.1 FotoFile image organization and retrieval system [42]



Figure 2.2 Adobe PhotoShop Album³. Keyword tags can be dragged and dropped onto photos to associate them with keyword. Users can customize keyword tags and use them to find photos later.

Adobe PhotoShop Album [2] is a commercial product from a well known image application developer, Adobe Systems Inc. Adobe PhotoShop Album gathers all photos in the users' computer and lets users see those photos in one convenient place, organized by date or any chosen subject. On the top of the interface, it has a timeline showing the distribution of photos over time. The timeline has two sliding knobs and users can filter photos by the date they were taken. One of the interesting features it

³ Adobe PhotoShop Album is a trademark of Adobe Systems Inc.

(<http://www.adobe.com/products/photoshopalbum/>)

incorporates is a keyword tag. Users are allowed to create customized keyword tags that represent special people, places, or events, and drag them onto photos so that pictures can be found by subject later. As shown in Figure 2.2, a keyword tag on the right panel can be dragged onto a photo or a group of photos to annotate photos with the keyword. When users drop a keyword tag onto the search panel, photos that have been annotated with the keyword will be found and shown on the center panel as a search result. Users can add more tags to narrow down the search result further.



Figure 2.3 PhotoFinder. Users can annotate photos with drag-and-drop interface, also known as direct annotation. The name of a person can be dragged from the list onto photos. This annotation is used for keyword search for finding photos.

PhotoFinder [39] is a research prototype focusing on enabling non-technical users of personal photo collections to search and browse easily. PhotoFinder allows rapid browsing of large number of photos organized in collections. It provides a set of visual conjunctive Boolean query interfaces and query preview features. PhotoFinder offers a technique known as *direct annotation* to enable personal names to be placed on a photo. In PhotoFinder, annotation is achieved by drag-and-drop. Users can drag keywords (usually a person's name) onto any place on the photos. The content and the position of keywords are automatically saved in a database so that they can be used for searching. Kang *et al.* [39] found that rich annotations and captions are the basis for successful story telling among people.

PhotoFinder can create collections from the folders in file explorer by dragging the selected folders onto the library viewer and the photos in the collections can be sorted by the selected attribute such as date, location, title, and so on.

ACDSee [1] is one of the most successful image browsers on the market. Based of the file system, it provides users a total environment to view and browse image and graphics files quickly, even large images or thousands of thumbnail previews at a time. It enables users to organize pictures efficiently by assigning images to categories and keywords in batches. It allows users to find files fast by searching on categories, keywords, metadata, date, type, description, or other properties--or by clicking dates on a calendar. Users also can view all photos from a particular year, month, week, or day.

One disadvantage of ACDSee is its lack of ability to support managing photo groups. It depends on low level file structures and does not allow images to be included in more than one group (or folder) without copying the original and having multiple copies. ACDSee supports basic features such as ‘Favorites’ and ‘Folders’, but users might still need more convenient functions rather than folder-based grouping for managing their image collections especially when the collections contain a large number of photos.

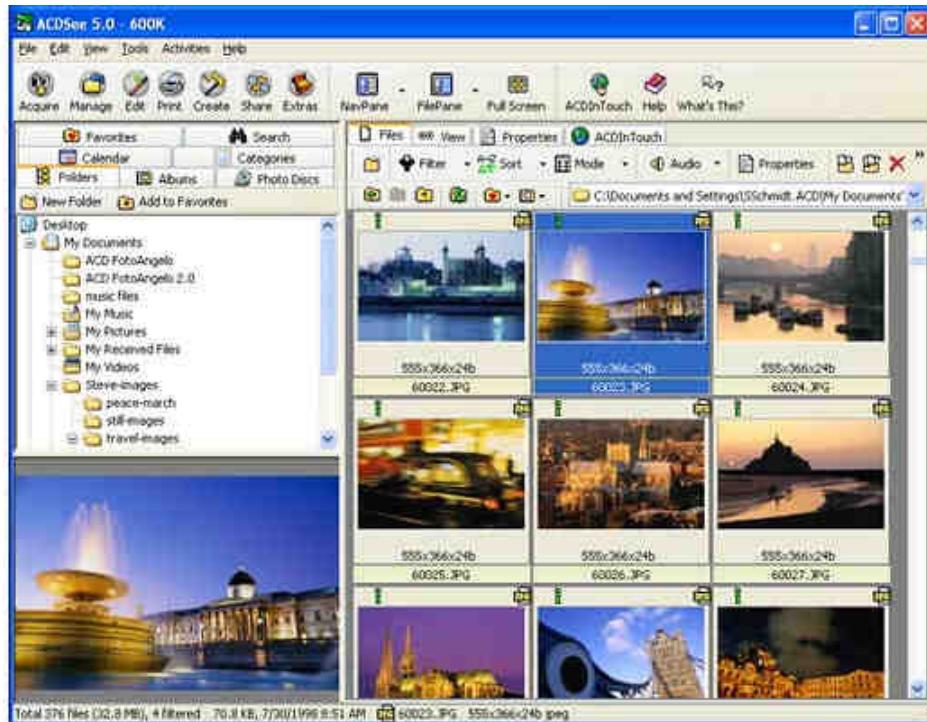


Figure 2.4 ACDSee™ Image Browser⁴

The Personal Digital Historian (PDH) research project [62] presents visualization and layout schemes developed for a novel circular user interface designed for a round, tabletop display. The overall goal of PDH is to investigate ways to effectively and

⁴ ACDSee is a trademark of ACD Systems <http://www.acdsystems.com>

intuitively organize, navigate, browse, present and visualize digital data in an interactive multi-person conversational setting. Shen *et al.* [62] discuss the direct implications of such a circular interface on document orientation and describe the circular layout as shown in Figure 2.5 and explain how to use them in a multi-person collaborative interface. This type of collaborative environment that adapts to the needs of group workers would allow the computer as a device to disappear in the architecture of office spaces, while its functionality remains ubiquitously available.

PDH is an example of a non-WIMP (Windows, Icons, Mice and Pointing, which refer to the desk top, direct manipulation style of user interface) image browser. Unlike the previous examples, PDH introduces a novel environment which is intended to facilitate cooperative browsing.



Figure 2.5 Personal Digital Historian (PDH) from MERL [62]

Query-By-Image-Content (QBIC) is one of the well-known content-based image retrieval systems [22][23]. IBM developed the system which lets users make queries of large image databases based on visual image content -- properties such as color percentages, color layout, and textures occurring in the images. Such queries use the visual properties of images, so users can match colors, textures and their positions without describing them in words. This approach can be effective when users have a clear idea about searching targets such as color, shapes and so on. However, when users have no idea about what the targets look like, this approach is less useful. In addition, QBIC has limitations in specifying semantic elements in images. The system records color and shapes without understanding the meaning of objects in images. However, content based queries can be combined with text and keyword predicates to get powerful retrieval methods for image and multimedia databases. In this dissertation, one of my research goals is to increase textual metadata to facilitate this kind of retrieval.

Flamenco [74] is a web based prototype, whose primary design goal is to allow users to move through large information spaces in a flexible manner without feeling lost. A key property of the interface is the explicit exposure of hierarchical faceted metadata, both to guide the user toward possible choices, and to organize the results of keyword searches. The interface uses metadata in a manner that allows users to both refine and expand the current query, while maintaining a consistent representation of the collection's structure. This use of metadata is integrated with free-text search, allowing the user to follow links, then add search terms, then follow more links, without interrupting the interaction flow. The results of usability studies find strong

preference results for the faceted category interface over that of the standard approach.

Flamenco is useful for searching images when the user has only a vague idea of what they are looking for. The system allows users to follow their information needs. However, the result pane, as shown in Figure 2.6, can show a very limited number of images at once and does not allow users to preview images in result categories. Users have to navigate into image groups to see the result in a group. While Flamenco provides a very flexible searching and browsing environment, its web-based interface limits richer interactions. In addition, Flamenco requires refined metadata and pre-classified categories, which is often not available.

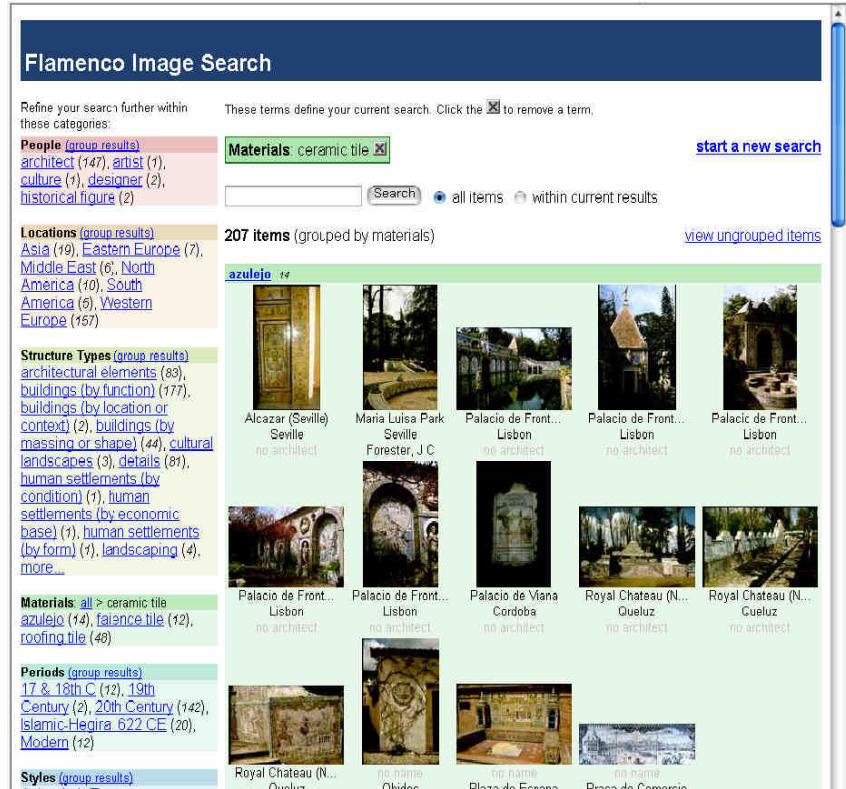


Figure 2.6 Flamenco Interface [74].

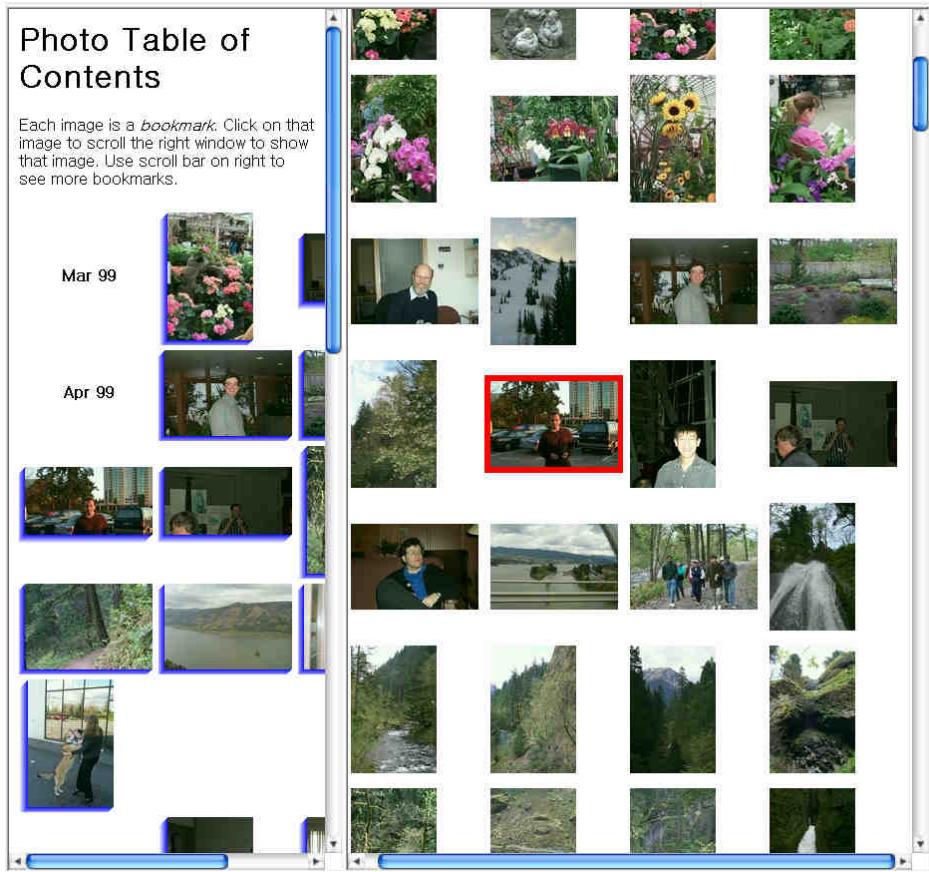


Figure 2.7 PhotoTOC [56] user interface. The left panel shows representatives photos of clusters. As users click a cluster in the left panel, the right panel scrolls so that the first photo of the cluster should be shown on the screen with red borders.

Photo Table Of Contents (PhotoTOC) [56] is an interface that helps users find digital photographs in their own collection of hundreds or thousands of photographs. PhotoTOC is a browsing user interface that uses an overview+ detail design. The detail view is a temporally ordered list of all of the user's photographs. The overview of the user's collection is automatically generated by an image clustering algorithm, which clusters on the creation time and the color of the photographs. PhotoTOC was developed by design iteration on an earlier clustering user interface: AutoAlbum.

PhotoTOC was tested on users' own photographs against three other browsers: a hierarchical folder browser (with image thumbnails and the user's own folder structure), a flat detail view with no automatically generated overview, and AutoAlbum. Searching for images with PhotoTOC was subjectively rated easier than all of the other browsers and PhotoTOC's task performance was not slower than any other browser. This result shows that an automatic organization of personal photographs can be effective: it requires no organization effort by the user and yet facilitates efficient and satisfying search.



Figure 2.8 Apple iPhoto⁵ [33]. Users can select and drag photos from the main screen onto the icon representing an album as shown in the right image.

iPhoto [33] by Apple Computer Inc. is an all-in-one application for importing, organizing, editing and sharing digital photos. It allows users to arrange the pictures by theme (such as vacations and ball games), subject (people, places, pets and so on), or any other way they prefer by dragging photos onto the icon representing an album. Users can rearrange the sequence of photos in the albums any way they choose. Users are also able to make as many albums as they like using any images from the photo

⁵ iPhoto is a trademark of Apple Computer Inc. <http://www.apple.com/iPhoto>

library, and even include the same photo in several albums without making multiple copies of it.

iPhoto lets users categorize photos and make them searchable by keyword or comment. Keywords are essentially labels assigned to different categories of photos, and comments are the captions written for individual photos. iPhoto enables users to find the photos by keyword, or by searching for any of the words or phrases in comments.

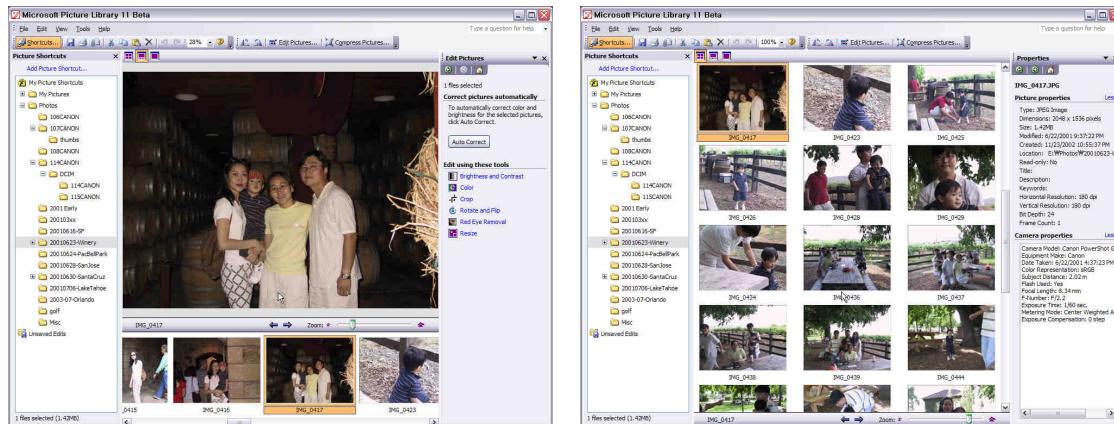


Figure 2.9 Microsoft Office 2003 Picture Manager⁶. Left: Filmstrip view, Right: Thumbnail View.

Microsoft Office Picture Manager [51] provides a flexible way to manage, edit, and share users' pictures. Users can view all the pictures no matter where they are stored; the Locate Pictures feature helps users find images scattered in disk. Instead of navigating between locations and lists of folders each time, users can add shortcuts to all the locations that contain pictures. Office Picture Manager does not require users to create new categories or import pictures. Once users add a shortcut, they can work

⁶ Microsoft Office is a trademark of Microsoft Inc. <http://office.microsoft.com>

with pictures from that location as if they were working from the file system. Office Picture Manager can also automatically perform corrections to your pictures such as brightness and contrast, color, crop, rotate and flip, red eye removal, and resize.

Office Picture Manager allows users to use Microsoft SharePoint [61] for a rich collaboration experience. Through SharePoint, users can share images across the intranet and download picture versions at any size or resolution, while efficiently storing the original pictures. When sharing pictures, users can also compress files to a size that is most efficient for the way they intend to use the picture.

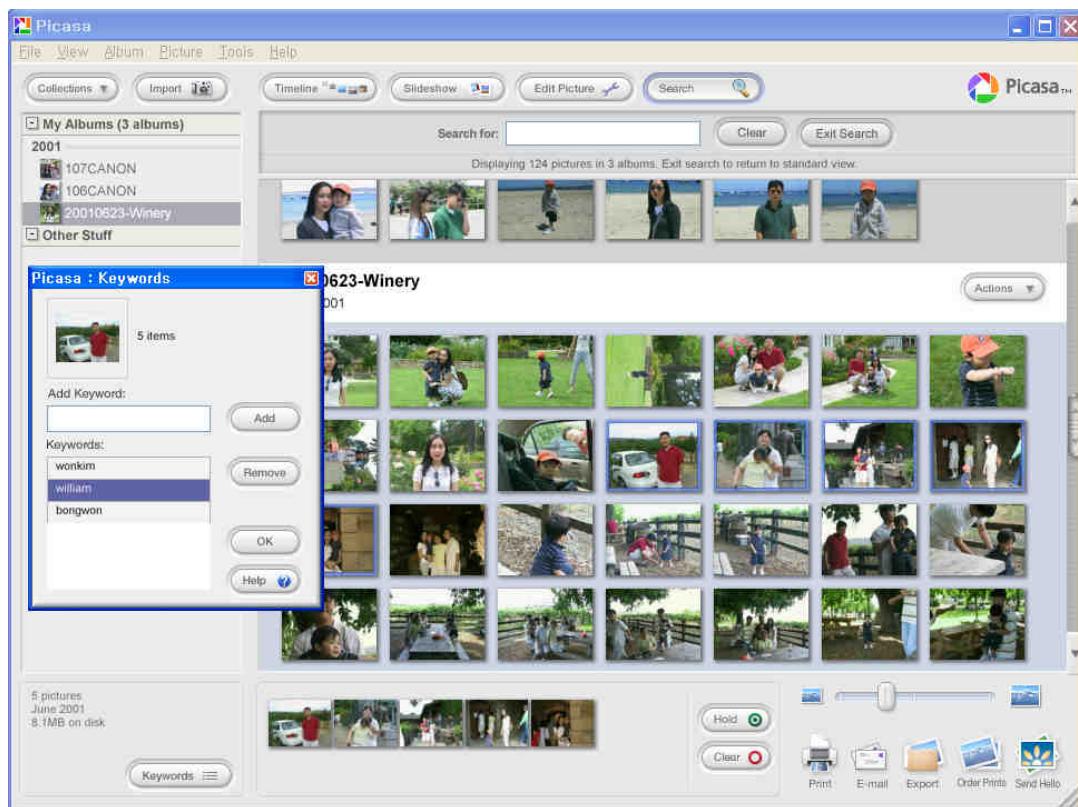


Figure 2.10 Picasa⁷ image browser [54]

⁷ Picasa is a trademark of Picasa, Inc. <http://www.picasa.com>

When first launched, Picasa [54] begin to search the entire folder on a computer and created an album per folder. It supports all of the pictures of general formats as well as standard camera movie files. Users can easily organize folders by merging and renaming them. Photos are laid out on the screen by the albums and users can use scroll bars to navigate them. A slider on the bottom right of screen allows users to resize thumbnails. On the left panel, the list of albums is arranged by timeline.

Picasa also provides image editing functions such as red-eye removal, cropping, rotation, and auto-correction. It also allows users to send pictures quickly by using users' e-mail clients such as Microsoft Outlook. In addition, Picasa supports users to publish albums as web pages.

While users are allowed to add any keyword to photos, the annotation process is very time-consuming. When users are adding keywords, a separate window is provided as shown in Figure 2.10, and users are required to type keywords manually. Furthermore, Picasa does not support a list of existing keywords, which makes annotation very difficult. Even though Picasa is capable of searching all the keywords that users have entered, annotation is limited with manually entered keywords.

Girgensohn[28] *et al.* created a photo management application (Figure 2.11). The prototype provides a semi-automatic approach to facilitate the task of labeling photos with people. They used a face detector to automatically extract faces from photos while the less accurate face recognizer to sort faces by their similarity. The sorted faces are presented as candidate as shown in Figure 2.11. Users are allowed to drag

faces onto name labels to make annotations. Their simulation study showed that on average 60% of faces could be assigned successfully with three or four steps.

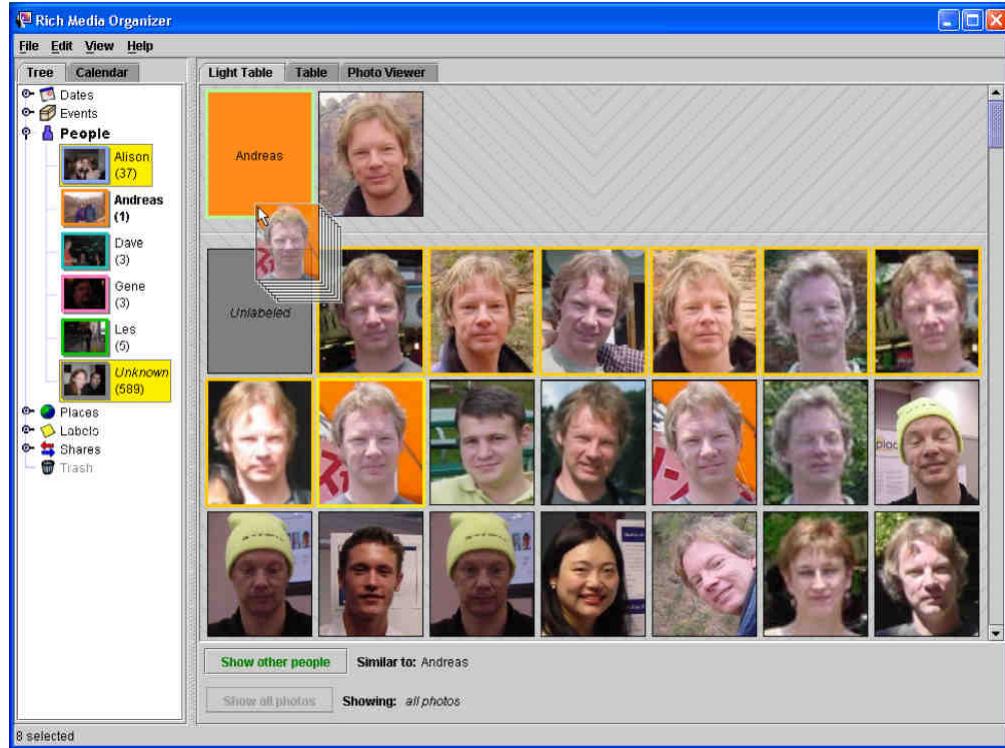


Figure 2.11 Face annotation interface for FXPAL prototype [28]

While the semi-automatic approach of Grgensohn [28] *et al.* showed a great potential, there are scalability and usability issues. As the number of faces in the system increases, it is expected that users are required to use scroll bars frequently. Furthermore, as the number of people increases, the face recognition accuracy decreases significantly and it makes bulk annotation harder. Since the prototype does not limit the number of faces on the screen, users might have problems when they try to label a large number of faces at once. In addition, the prototype solely depends on a face recognizer. Even with state-of-the-art systems, it is known that the face recognition accuracy for outdoor photos is around 50% [53]. Even though the

prototype circumvents the poor performance of the face recognition approach, getting help from other non-facial features such as timestamp and clothing would increase the accuracy of the initial face assigning.

2.2 Zoomable User Interfaces

In this section, I explain the key features of zoomable user interface techniques and a zoomable user interface toolkit, Jazz. Zoomable image browsing introduced by Bederson [4] showed a great potential to increase the browsability of image retrieval systems. In my research, I apply various zoomable user interface techniques to enhance image management systems.

2.2.1 Fundamentals

Zoomable User Interfaces (ZUI) use a metaphor designed as a successor to the desktop interface [9]. Compared to the desktop interface where the 2D space does not have any depth, ZUIs enable users to move their point of view with depth. Users can zoom in to any specific area in 2D space and zoom out to see the larger overview of an area. The animation that occurs during zooming and panning helps the user to remember where things fit together based on spatial relationships. On the other hand, a folder-based desktop relies on the user's ability to recall particular information about folders.

In ZUIs, unlike pure 3D systems, the axis of zooming is fixed to be perpendicular to 2D space so that space can be zoomed in and out only in that direction. The primary navigation techniques for ZUIs are panning and zooming, and rotation, tilt, and

distortion are not allowed. Users can zoom out to see a bigger area and zoom in to see things in more detail. Users also can pan the viewing window without zooming. The simple navigation in ZUIs prevents the general drawbacks of 3D systems such as disorientation and navigation problems, while providing full power of space navigation.

Since ZUIs are dependent on humans' ability to remember where things are in space, it is crucial for users to perceive where they are in space. To lessen users' cognitive load with this perception, the animation during zooming is very important. It is known that users tend to get more lost in space when zooming is not animated. [6]

2.2.2 Jazz: A Zoomable User Interface Toolkit

Jazz [7] is a toolkit that supports Zoomable User Interfaces, designed and developed at the University of Maryland. It is built in pure Java and provides a unique way to create robust, full-featured graphical applications.

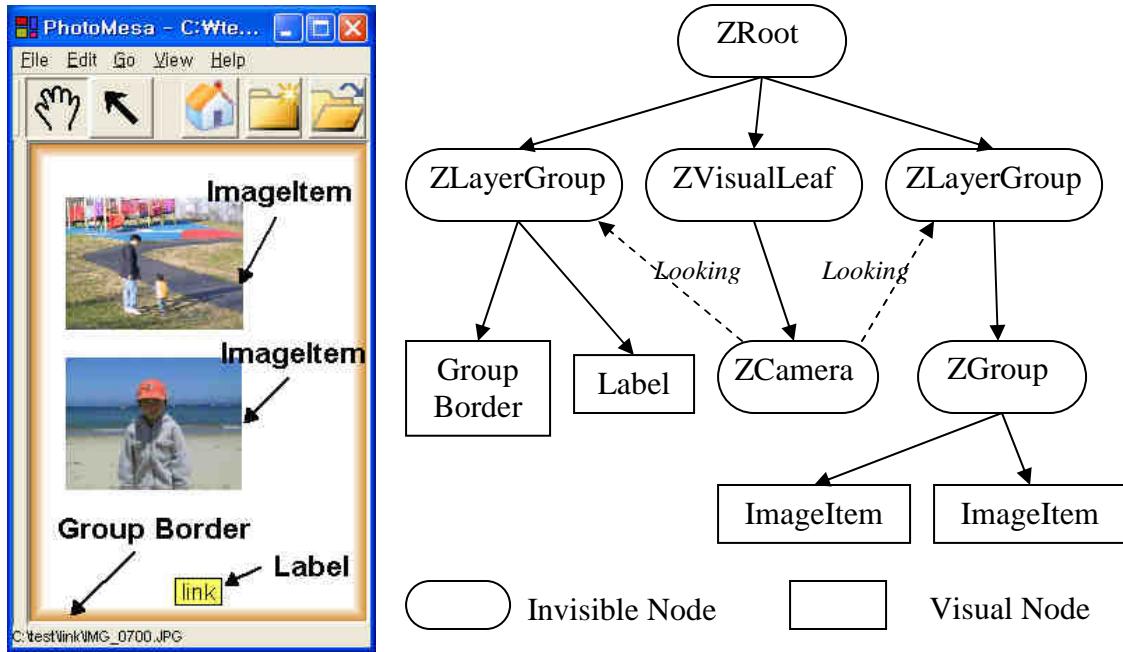


Figure 2.12 An example scene graph structure. The partial scene graph example on the right side is represented on the screen as shown in the left figure.

Jazz is based on a “*polyolithic*” design philosophy. In Jazz, objects are composed by combining simple objects with a scene graph structure. Jazz tackles the complexity of building graphical applications by dividing object functionality into small, easily understandable node types such as **ZLayerGroup**, **ZGroup**, **ZVisualLeaf**, and so on as shown in Figure 2.12.

Figure 2.12 shows an example application with a scene graph structure. The right scene graph of Figure 2.12 is rendered on the screen as in the left screen shot. Photographs on the screen in the left screen shot are represented by **ImageItem** in the right scene graph. But, the **ImageItem** does not have to include all the required functions to draw images on the screen. Its upper level invisible parent, **ZGroup**, takes care of its coordination on the screen and let the **ImageItem** focus on rendering the

associated image without considering the detail about its location and scale. By separating complex functions into small, easy extendable parts, Jazz helps to build applications clearly.

Jazz has been used in a number of user interface applications including Fisheye Menus and tree viewers. [5][31]. It also inspired developing other toolkits such as Piccolo [55].

2.3 Treemap Algorithm

Treemap algorithms are very useful to display a large volume of information on the screen. Combined with zoomable user interfaces, quantum strip treemap algorithm [4] is capable of showing a large number of images in a 2D zoomable space.

2.3.1 Fundamentals

Treemap algorithms are a space-filling visualization method which is capable of representing large hierarchical collections of quantitative data in a compact display [36][64]. A treemap (Figure 2.13) works by dividing the display area into a nested sequence of rectangles whose areas correspond to an attribute of the data set, effectively combining aspects of a Venn diagram and a pie chart.

A key ingredient of a treemap is the algorithm used to create the nested rectangles that make up the map. This set of rectangles is referred to as the layout of the treemap. The slice-and-dice algorithm of the original treemap paper [64] uses parallel lines to divide a rectangle representing an item into smaller rectangles representing its

children. At each level of hierarchy the orientation of the lines - vertical or horizontal - is switched. As seen in the right image in Figure 2.13, each cell represented by a rectangle is encoded with area to convey one of its attributes. Single level treemaps are nested hierarchically to form a whole map. (the left image in Figure 2.13)

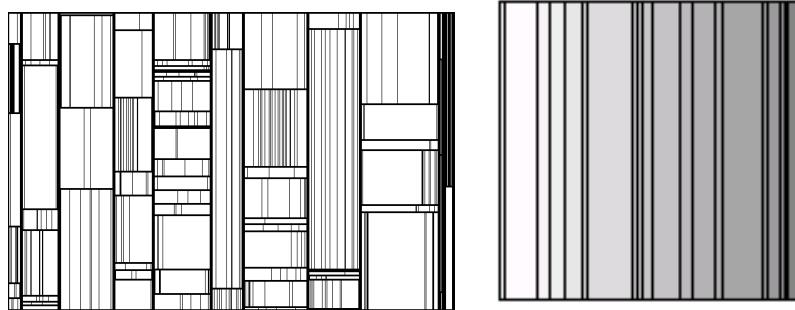


Figure 2.13 The slice and dice treemap layout. The left image shows a hierarchical application of the treemap algorithm. The right image shows a single level treemap.

Treemaps scale up well, and are useful even for a million items on a single display. However, the slice-and-dice layout often creates layouts that contain many rectangles with a high aspect ratio. Such long skinny rectangles can be hard to see, select, compare in size, and label. Hence, many modified versions such as [12], [70] have been developed to give a better visual representation as shown in Figure 2.14.

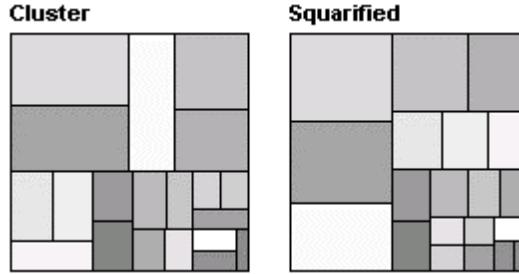


Figure 2.14 Low aspect ratio layouts. Shading indicates order, which is not preserved.

2.3.2 Quantum Strip TreeMap

The quantum strip treemap algorithm [4] is a modification of the existing Squarified Treemap algorithm [12]. The quantum treemap algorithm is similar to other treemap algorithms, but instead of generating rectangles of arbitrary aspect ratios, it generates rectangles with widths and heights that are integer multiples of a given elemental size. The basic idea is to start the regular treemap algorithm and then as rectangles are generated, they are *quantized*. The dimensions of rectangles are expanded or shrunk so that each dimension is an integral multiple of the input element size. The total area of the rectangle is no less than that needed to layout a grid of the requested number of objects.

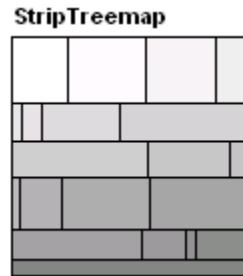


Figure 2.15 Strip treemap algorithm applied to 20 rectangles

It works by processing input rectangles in order, and laying them out in horizontal (or vertical) strips of varying thicknesses (Figure 2.15). While maintaining a current strip, and then for each rectangle, the algorithm checks if adding the rectangle to the current strip will increase or decrease the average aspect ratio of all the rectangles in the strip. If the average aspect ratio decreases (or stays the same), the new rectangle is added. If it increases, a new strip is started with the rectangle. For each rectangle, the algorithm computes the average aspect ratio of the current strip. Each strip will be, on average, of length equal to the square root of the total number of rectangles. Thus, the strip treemap algorithm runs in $O(\sqrt{n})$ time on average.

PhotoMesa [4] lays out images by using the quantum strip treemap algorithm and appears to be the only use of treemaps to display non-quantitative data within each rectangle.

2.4 Annotation and Metadata

Annotation is defined as a process that labels the semantic content of images (or object) with users' metadata. Annotation is especially important for image collections because it allows enhanced searching and browsing which is not possible without annotated information.

As described in section 2.2, PhotoFinder [39] offers a drag-and-drop technique known as *direct annotation* to enable personal names to be placed on a photo or a group of photos. Users can drag keywords (usually person's name) onto any place on the photos to save typing. Similarly, Adobe PhotoShop Album [2] incorporates a

keyword tag. Users can create customized keyword tags that represent special people, places, or events, and drag them onto photos so that pictures can be found by subject later. (Figure 2.2) These improvements help users make annotations efficiently in comparison with the manual annotation strategy where users are required to type-in keywords. However, it is still a burden for users to make annotation on a large number of images.

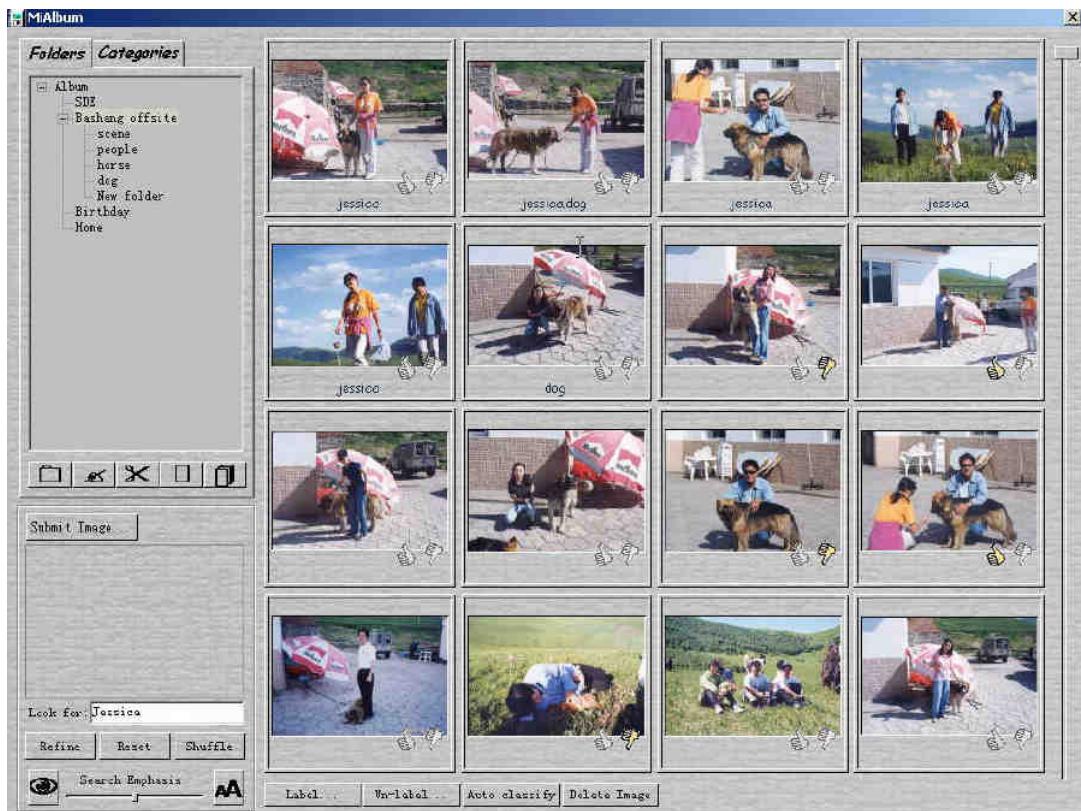


Figure 2.16 MiAlbum interface. Users are allowed to input relevance feedback by clicking thumbs-up and thumbs-down icon on the lower right corner of each image.

Wenyin *et al.* [71] introduce a novel approach to semi-automatically and progressively make annotations on images. The progressive annotation process is

embedded in the course of integrated keyword-based and content-based image retrieval. When a user submits a keyword query, the system retrieves and arranges images on the screen as shown in Figure 2.16. The search results include images which are relevant to the search keyword, as well as images found based on their visual feature similarity to the images matched with the query and/or a set of *randomly* selected images. When an image receives positive feedback from users (by clicking the thumb-up icon), the search keywords are automatically added to the images so that the images can be retrieved by keyword-based image retrieval in the future. The coverage and quality of image annotation is improved progressively as the cycle of search and feedback increases.

Wenyin *et al.* [71] report that the semi-automatic image annotation strategy is better than manual annotation methods in terms of efficiency, and is better than automatic annotation techniques in terms of accuracy. But the authors also detail that the MiAlbum user interface needs enhancements. The simple thumbs-up/down metaphor was not enough for users to understand the built-in underlying automatic algorithm. They also report a problem in the discoverability of relevance feedback.

Rodden *et al.* [58] observed users' behavior with their digital personal photographs and found that two features are essential for users. They are 1) automatically sorting photos in chronological order; and 2) displaying a large number of thumbnails at once. And they also found that the participants in their study most commonly wanted to browse their personal photos by event, rather than querying them based on more specific properties. This result is not surprising and matches well with my intuition.

Users just want to have a simple and meaningful way of browsing. One more thing has to be clarified is about location or place information. Some users think it is another very important type of information. However, in most cases, location information is tightly coupled with event information. When personal photos are taken in a relatively short period time, the photos usually tend to have the same event and location.

Along with the chronological information, people in photos are regarded as one of the most important pieces of information because a great many pictures of interest show human faces many of which are central objects in the images. It is not surprising that many image browsing prototypes and products [2][39][42][62] include features for labeling persons with metadata such as names. Rodden *et al.* also [58] hinted that robust face recognition would help users to browse their personal photo collections.

2.5 Saliency and Thumbnail Cropping

Thumbnails - generated by shrinking the original image - are one of the most widely used techniques for representing images. However, when used in limited screen space, they are often rendered too small and illegible. In this dissertation, I focus on intelligent cropping so that key components of images can be more recognizable in small thumbnails. I use a visual saliency model for cropping images.

Visual attention is the ability of biological visual systems to detect interesting parts of the visual input [34][35][49][50][72]. The saliency map of an image describes the degree of saliency of each position in the image. The saliency map is a matrix

corresponding to the input image that describes the degree of saliency of each position in the input image.

Itti and Koch [34][35] provided an approach to compute a saliency map for images. Their method first uses pyramid technology to compute three feature maps for three low level features: color, intensity, and orientation. For each feature, saliency is detected when a portion of an image differs in that feature from neighboring regions. Then these feature maps are combined together to form a single saliency map. After this, in a series of iterations, salient pixels suppress the saliency of their neighbors, to concentrate saliency in a few key points.

Chen *et al.* [14] proposed using semantic models together with the saliency model of Itti and Koch to identify important portions of an image, prior to cropping. Their method is based on an attention model that uses attention objects as the basic elements. The overall attention value of each attention object is calculated by combining attention values from different models. For semantic attention models they use a face detection technique [45] and a text detection technique [15] to compute two different attention values. The method provides a way to combine semantic information with low-level features. However, when combining the different values, their method uses heuristic weights that are different for five different predefined image types. Images need to be manually categorized into these five categories prior to applying their method. Furthermore, it heavily relies on semantic extraction techniques. When the corresponding semantic technique is not available or when the

technique fails to provide a good result (e.g. no face found in the image), it is hard to expect a good result from the method.

2.6 Automatic Event Identification

There are a number of approaches to automatically identify event clusters from digital photo collections. Cooper *et al.* [16] introduced a temporal similarity-based approach to cluster digital photographs by time and image content. Cooper *et al.*'s algorithm is general and unsupervised. It calculates event boundaries by computing temporal similarity between photos. For example, as photos are closer in time, they have higher similarity. [16] defines a confidence measure to determine the goodness of event boundaries. The confidence measure is calculated by combining each cluster's average self-similarity and the dissimilarity between adjacent clusters. Cooper *et al.*'s algorithm chooses event boundaries that maximize the confidence measure. Along with the temporal similarity, they also include content based similarity. Using low frequency discrete cosine transform (DCT) coefficients from each photos, they calculated visual similarities between photos. Cooper *et al.* applied their techniques and measure the accuracy of the algorithm. While their experimental results show that their algorithm had around F-score 0.85, it was not significantly better in comparison with other algorithms in [44][41]. Also, using the content similarity did not make significant contribution to detect events.

Girgensohn *et al.* [27] presented a prototype photo manager based on Cooper *et al.*'s event detection algorithm [16]. As shown in Figure 2.17, photographs are grouped by

automatically identified events. The left panel shows identified events as a tree view and the main panel displays thumbnails of individual photographs grouped by event.

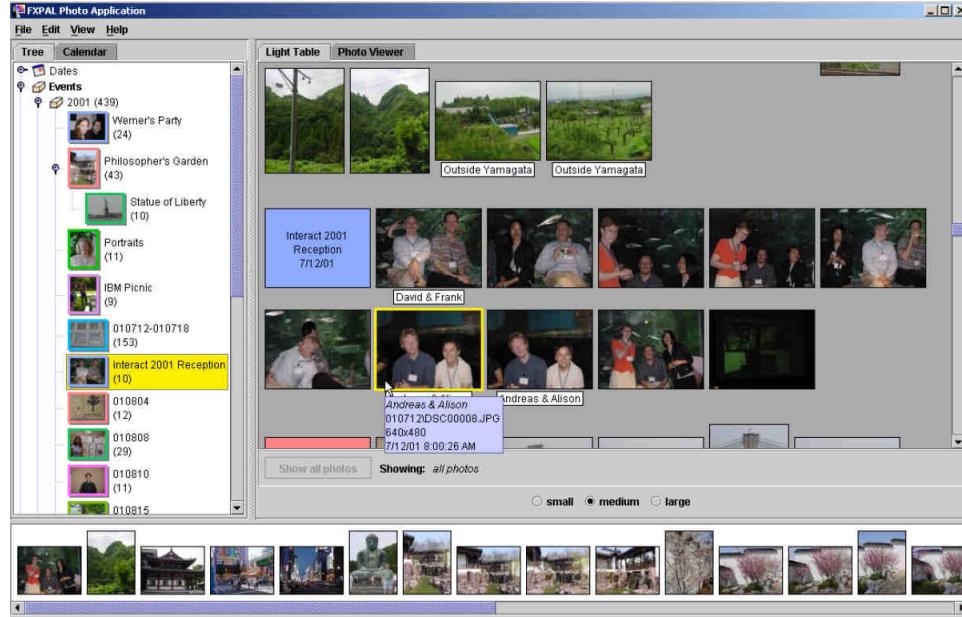


Figure 2.17 FXPAL Photo Application [27]

Platt *et al.* [56] use an adaptive local threshold method to detect event boundaries.

Platt et al.'s algorithm [56] compares a time interval to its local average interval. If a temporal gap between adjacent two photos is considerably larger than its weighted local average, the algorithm decides the gap to be an event boundary. Unlike the algorithm in [16], this algorithm requires additional parameters, a threshold for sensitivity and a windows size, which should be empirically chosen and can be subjective. Cooper *et al.* [16] also reports that the accuracy of this algorithm was not very good as compared with other clustering algorithms.

Scale-space analysis [44] is a technique for accessing structure at multiple scales in a data set. It assigns a Gaussian kernel per data to form a Gaussian mixture. The result

mixture is used to form clusters by finding points where its second derivative value is zero (peak point). By using varying standard deviation, it allows to construct hierarchical segmentation.

Loui *et al.* [47] use the K-means algorithm combined with content-based post-processing for automatic albuming of photographs. They checked the color similarity of images at event boundaries to verify that the images indeed differ.

Graham *et al.* [30] use time information for creating event hierarchies for personal photo collection. Based on [56], they create initial clusters. Then, they build an event hierarchy based on the initial clusters. For each cluster, a summarization photograph is selected to represent the event. With this clustering and summarization technique, they built a prototype, “Hierarchical Browser” and performed a user study. They found that users completed given tasks better with the hierarchical browser. They also showed that the summarization technique significantly reduced users’ browsing completion time.

Gargi [25] [26] presented an analysis of consumer media capture behavior based on timestamp metadata. He reported *bursty* behavior of personal photo collections [25]. Date sets used in [25] shows that photos are taken on approximately one day during a ten day period. However, on the day that photos are taken, users take about twelve photos at once.

As shown in this section, there have been numerous researches on automatic event identification. Nevertheless, automatically identified events are not perfect and

require users' amendment. Most of the above approaches did not consider users' feedback on event boundaries. Once event boundaries are set by an algorithm, they are not adaptable and the algorithms do not allow further interaction with users.

Chapter 3

Preliminary Work: PhotoMesa and Its Applications

In this chapter, I explain the design and implementation issues of PhotoMesa. While the design challenge is to support efficient browsing without losing the intuitive interface, there are crucial performance issues, especially because PhotoMesa handles a large number of images at the same time. I examine the issues in detail and explain techniques that I applied when designing and developing PhotoMesa.

3.1 Overview

Many conventional image browsers follow the WIMP style (Windows, Icons, Mice and Pointing, which refer to the desk top, direct manipulation style of user interface) and they usually use *folders*. Unless images are well organized inside folders, users need to keep opening folders before they are able to locate a specific image.

On the other hand, PhotoMesa [4] allows users to view a large set of images on one screen in a zoomable environment. Users can zoom in to see the detail image (Figure 3.1) and zoom out to view the overview of images as in Figure 1.2. PhotoMesa allows users to view multiple directories of images with a simple set of navigation functions. The name *PhotoMesa* derives from the Spanish word *mesa* which means table, but is commonly used in the US southwestern states to describe the natural volcanic

plateaus which are high and have flat tops. Standing atop a mesa, you can see the entire valley below, much as you can see an overview of many photos in PhotoMesa.

As the user moves the mouse, the directory under the mouse cursor is highlighted, and the label is shown in full. Then when the user clicks the left mouse button, the view is smoothly zoomed in to that directory. At any point, the user can press the right button to zoom out to the previous magnification.

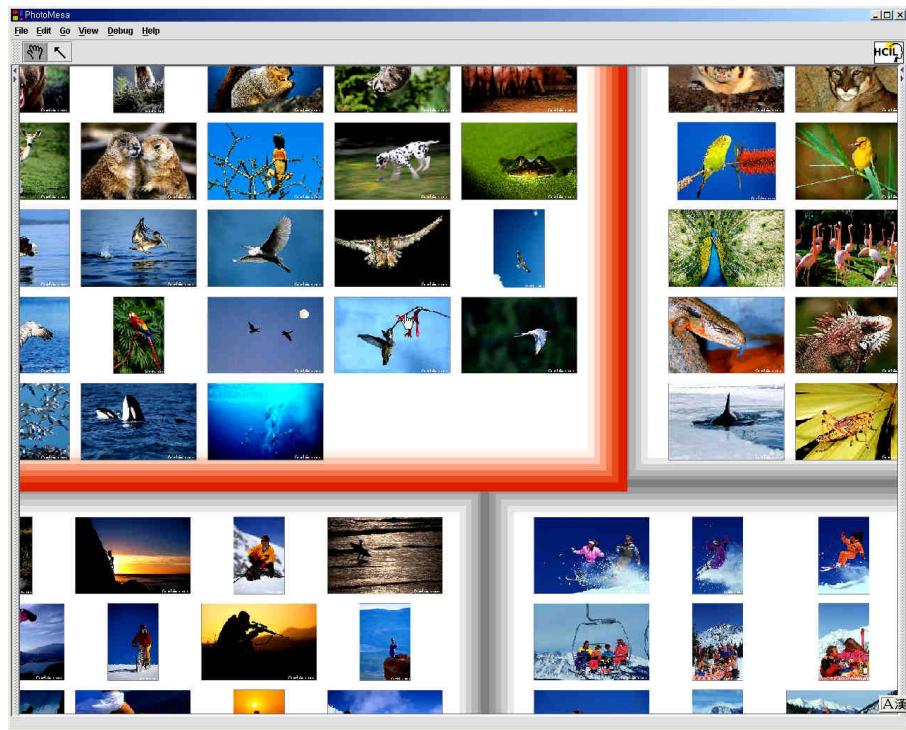


Figure 3.1 Detail view (zoomed-in view) of PhotoMesa

One of the goals for designing PhotoMesa is to provide a simple and intuitive interface. Thus, simplifying navigation was a very important challenge. However, pure zooming is known to have a navigation problem. Users are easily disoriented when extremely zoomed in [37]. Users often have no idea where they are looking at, and which direction they should move. Furthermore, this situation is easily confused

with when extremely zoomed out because all that users can see on the screen is empty space.

A constrained zooming technique is designed for PhotoMesa to prevent the disorientation problem. In PhotoMesa, users are only allowed to zoom into the highlighted area which is easily recognizable prior to navigation, and to zoom out only to the previous magnification. The users' navigation actions are restricted to left-click (or space key) to zoom in and right click (or enter key) to zoom out. According to pilot studies, we observed that most users liked the constrained zooming and they also found it easy and intuitive.

Another novel technique introduced in PhotoMesa is the use of Quantum Strip Treemaps [8] with which PhotoMesa lays out images in 2D zoomable space as shown in Figure 1.2. Treemap is a space-filling visualization method which is capable of representing large hierarchical collections in a compact display (see section 2.3).

One interesting assumption that PhotoMesa made is that it is not necessary to show the hierarchies in which photos are arranged. The rationale for this is that users looking at images are primarily interested in groups of photos, not at the structure of the groups. In addition, the interface for presenting and managing hierarchies of groups would become more complicated for users. This postulate enables simple and effective algorithms for image layouts.

While the initial version of PhotoMesa provided a novel image browsing interface, it had room for improvement. The initial version did not support any metadata other

than the native file structure information such as filename, directory and date information. While it is easy for users to begin to use PhotoMesa, the inability to handle rich metadata limits its capabilities in some crucial activities such as adding captions and keywords. Furthermore, the initial version had very limited search functions, allowing users to search through image file names only.

Based on the initial version of PhotoMesa, I redesigned and re-implemented PhotoMesa to control rich metadata while consuming fewer computing resources. While the initial version focused on personal usage, I extended PhotoMesa into a general image search interface. Through a set of software interfaces, PhotoMesa can be plugged in as a front-end user interface for general image browsing environments. The new PhotoMesa can be integrated with database systems and handle richer metadata, enabling users to query images by a set of keywords. Furthermore, the new version allows users to control grouping. Images can be grouped in meaningful clusters based on users' search category. Search results can be dynamically regrouped as users refine their search conditions. Compared to conventional image search interfaces, PhotoMesa shows great potential as a general image retrieval interface.

I also designed the new PhotoMesa to be web-deployable and it can be run as an applet in web-based applications.

3.2 Multi-level Thumbnails

PhotoMesa typically handles a large number of images at once. When zoomed out, users can see the overview of images which are shown as small thumbnails. When

zoomed in, images are presented in larger dimensions and users can see more detail about the images. PhotoMesa should support rapid transitions between various levels of magnification. As a user navigates the zoomable space, PhotoMesa is required to render a large number of images on the screen. But, the problem is that it is not possible to hold all the images inside the main memory. For example, suppose that PhotoMesa is showing 1000 images and each image is about 1000X1000 pixels in size. The required memory is roughly 3GB⁸, which is far beyond typical computer systems.

PhotoMesa uses multi-level thumbnail images as a solution to this problem. Instead of loading all images inside the main memory, PhotoMesa holds only minimum sized thumbnails. When zooming, PhotoMesa dynamically determines the right thumbnail level and loads thumbnails of that level as well as releasing thumbnails of off-screen images. When zoomed out, PhotoMesa loads a *large* number of *small* sized thumbnails and, when zoomed in, it loads a *small* number of *large* sized thumbnails. This technique ensures that approximately one screenful of data is loaded on the main memory at a time. PhotoMesa is implemented to use four levels of thumbnails with maximum of 10, 50, 100, and 200 pixels, and it can limit its memory usage to 256 MB even when interacting with two thousand images.

In addition, there are other benefits to having multi-level thumbnails. When PhotoMesa is using images over the networks, it can regulate the data transfer

⁸ For rendering, images often have to be transformed into a RGB format where one color pixel is composed of three bytes.

bandwidth. Since PhotoMesa requires only one screenful of data at a time, thumbnail data is continuously transferred as users navigate among images. This technique provides shorter response time and balanced network traffic. Furthermore, low resolution thumbnails can be used as a cache. While high resolution images are being downloaded slowly, a low resolution thumbnail can be used to give faster feedback to users.

However, there are some tradeoffs with multi-level thumbnails. It takes a while to generate thumbnails. It also requires additional disk space to store them. To minimize thumbnail generation overhead, PhotoMesa generates thumbnails only when it sees a new image and stores them in a disk cache which is invisible to users. PhotoMesa reuses thumbnails whenever possible.

3.3 Implementation Issues

As I mentioned earlier, based on the initial version of PhotoMesa, I improved PhotoMesa to control rich metadata while consuming fewer computing resources. In this section, I address the issues about enhancing PhotoMesa.

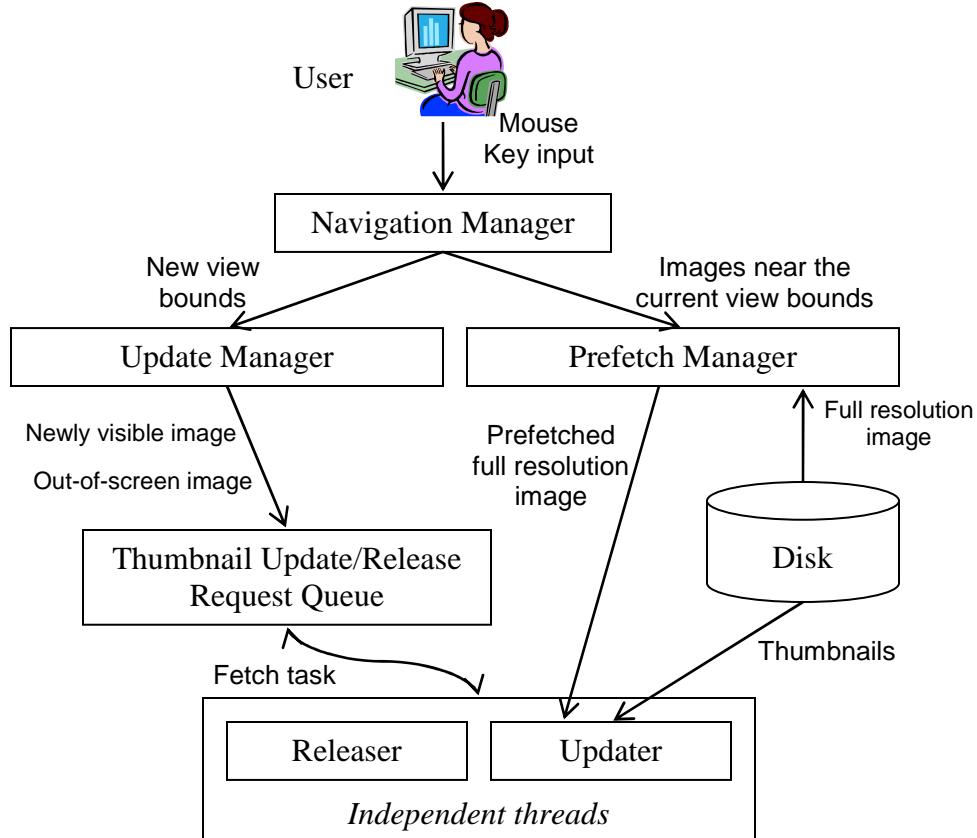


Figure 3.2 PhotoMesa software architecture

- Asynchronous Thumbnails Updating

PhotoMesa uses multi-level thumbnails and those thumbnails need to be dynamically loaded or released very efficiently due to the memory limitation. For example, when zoomed in, low resolution thumbnails need to be replaced with higher resolution images. In the case when users pan or zoom, some images on the screen become out of visible bounds and those off-screen thumbnails need to be quickly released from the memory.

PhotoMesa uses two independent threads, “*Updater*” and “*Releaser*” to achieve efficient updating and releasing thumbnails. When users zoom or pan, the update

manager identifies which images should be updated and released. Then it notifies each thread with this information. Newly visible and zoomed-in images should be updated with corresponding thumbnails and panned out images (not visible any more on the screen) should release their thumbnails from the main memory. The update thread keeps replacing thumbnails with the right sized thumbnails and the release thread frees unused thumbnails. These operations are performed independently to avoid blocking the interaction. In this way, users are allowed to navigate without waiting for thumbnail loading and/or releasing to be completed. *Updater* and *Releaser* respectively merge multiple requests into a single request for efficient thumbnail management.

- Smooth Animation

Animated zooming and panning is a very important feature in zoomable user interfaces. Since zoomable user interface techniques take advantage of users' human ability of remembering spatial relationships, zooming and panning should be animated smoothly to help the users' cognitive loads.

Animation between two views is achieved by redrawing a series of in-between frames quickly. However, it is not an easy task to redraw thousands of thumbnails at a minimum target rate of ten frames per second. To speed up this rendering, PhotoMesa uses a native type image class in Java. The native type image has the same color model and structure as the native machine uses. Also, this type of image can reside in the VRAM of the graphics card so that they can be processed by hardware accelerated

graphics engine without using CPU cycle. When properly used, the native image can be rendered more than five times faster than non native type images.

Along with the technical enhancement, I also took advantage of humans' cognitive capabilities. I found that detailed information need not be drawn when animating. Users did not notice that some auxiliary information on the screen such as labels, image borders, and group borders are not drawn during the animation. In addition, they did not perceive low quality thumbnails, instead of high quality thumbnails which are used in a static scene, are rendered on the screen during the animation.

These techniques enable PhotoMesa to render approximately 30 frames per seconds with 1,500 images on the screen when run on a 2.4GHz Pentium 4 machine with 512MB memory.

- Prefetch high resolution images

When a single image is zoomed in, PhotoMesa shows the image in its full resolution. Performance degradation also occurs at this point. Since the original images are usually compressed with popular formats such as *jpeg*, *png*, and *gif*, the image is required to be decompressed and transformed before being rendered on the screen. As a result of this processing, there is a delay between users' navigation and the system's rendering.

However, we observed that users tend to have some patterns when browsing images. Once a user selects an image as a full resolution view, he/she tends to see the next image also at full resolution. PhotoMesa takes advantage of this behavior and tries to

preload the next image while users see a current image. If the user keeps on navigating to the next image as predicted, the delay stated above can be avoided and an immediate response can be provided.

This kind of prefetching technique is widely used in commercial image browsers [1][2]. But, prefetching in PhotoMesa is a little bit different from others. In other applications, the next image can be easily determined because users are allowed to move in one dimensional direction, back and next. But, in PhotoMesa, users can pan freely in two-dimensional space. In other words, users can pan into four directions, up, down, right, and left. Initially, PhotoMesa was design to prefetch all the four neighboring images.

However, I found that prefetching the four images at the same time produced too much overhead and it had little benefit over no-prefetching. As an alternative, PhotoMesa is implemented is to prefetch only one image at a time. I designed PhotoMesa to remember the last direction of users' navigation and to prefetch the next image in that direction. For example, if a user pans to the right by pressing the right arrow key, PhotoMesa remembers the direction and prefetches the right neighbor of the next image. This preference is kept until the user changes the navigation direction. It is observed that users do not pan randomly and they have a tendency to navigate in one direction for a period of time. This adaptable prefetching approach minimizes the prefetching overhead while providing good performance in practice.

- Use built-in thumbnails in EXIF [20]

PhotoMesa uses multi-level thumbnails and reuses thumbnails in the disk cache as much as possible. If it fails to find pre-generated thumbnails, it creates thumbnails only for those images. Usually, this procedure is performed when there are newly added. However, it often takes more than one second to generate multi-level thumbnails per image. When importing a large set of new images, it can take several minutes to finish generating thumbnails.

One solution is to use EXIF [20], an industry standard for digital images. Recently, many digital camera manufacturers follow the EXIF format which defines various types of information about digital images. Most of them are low level, camera specific information such as focal length, shutter speed, and so on. But, in the EXIF format, a thumbnail is also included. Before generating thumbnails, PhotoMesa checks whether images to be loaded contain EXIF headers and corresponding thumbnails. If available, PhotoMesa imports the EXIF thumbnails rather than generating thumbnails from scratch. This technique enables PhotoMesa to load a set of new images swiftly and to reduce the initial delay of executing PhotoMesa.

However, there is a tradeoff when using the embedded EXIF thumbnails. The thumbnails embedded in EXIF images are usually small and have low quality. Thus, the overall image quality of thumbnails can be decreased. Due to this characteristic, I made this feature as an option. When users choose to use EXIF thumbnails, PhotoMesa can skip the thumbnail generation process and reduce the initial delay. If

users decide not to use EXIF thumbnails, PhotoMesa creates and uses high quality thumbnails.

- Preview

With the WIMP style interfaces such as ACDSee [1] and Microsoft Windows Explorer, users are required to keep clicking (or opening) folders until they find search targets. Before opening a folder, the folder name is the only clue that users can have. Unless images are well organized inside folders, users have to look up many folders repeatedly.



Figure 3.3 Previewing an image under the mouse cursor

On the other hand, PhotoMesa shows all images at once on the screen grouped by their directory and allows users to do a visual search instantly. However, there are tradeoffs. Since all the images are shown on one screen, there are too many thumbnails on the screen. In addition, the thumbnails are usually small and often tiny. To help users with these problems, PhotoMesa provides a *preview*, an enlarged

thumbnail under the mouse cursor as shown in Figure 3.3. I implemented two more options for the PhotoMesa preview in addition to *delayed preview* that was included in the initial version of PhotoMesa.

1) *Immediate preview*

The preview follows the user’s mouse cursor. Since users are usually gazing at the mouse cursor, the visual distance to the preview is very short. It enables users to identify images underneath the mouse easily as they hover mouse on thumbnails.

PhotoMesa uses *immediate preview* as its default preview option.

2) *Tooltip preview*

Once the mouse moves over an image, the preview is attached under the image. It stays there until the mouse cursor moves out of the image. This type of preview is widely known as “tool tip” for GUI components.

3) *Delayed preview*

The preview is shown only when there is no user’s activity. With the *immediate preview* and *tooltip preview*, a preview image can obscure other thumbnails behind. When users are actively navigating with the mouse and keyboard, previewing is refrained. When the user stops to move the mouse or to type, a preview of the image under the mouse cursor is shown over the thumbnail.

While I was performing a related user study (see Chapter 4), I observed that preview was very useful especially combined with zoomable interface techniques. Users were

able to identify images under the cursor very easily as they were hovering the mouse. I found that the *immediate preview* technique is more useful than other preview techniques especially when thumbnails are small on the screen. Often, users were able to get sufficient information about images without zooming in.

3.4 ZPhotoMesa Component

The initial version of PhotoMesa focused on browsing personal photos on disk and it had very limited extensibility. I have defined a set of software interfaces to apply the PhotoMesa style interface to general image retrieval environments. I redesigned PhotoMesa to be an open software component so that other applications can embed it easily.

3.4.1 ZPhotoMesa Component Interface

ZPhotoMesa is named after general Jazz [7] components by using the Jazz naming convention that a component name begins with the capital letter Z. As its name implies, the PhotoMesa component, *ZPhotoMesa* can be treated as other Jazz components. It can be embedded in a scene graph structure and represented in zoomable spaces just like other Jazz component can be. Figure 3.4 shows an example of how *ZPhotoMesa* can be added into a JPanel.

```

public class PhotoMesaPanel extends JPanel {

    ZCanvas canvas;
    ZPhotoMesa photomesa;

    public PhotoMesaPanel() {
        // The canvas prepare a basic scene graph structure when created.
        canvas = new ZCanvas();                                // Create canvas
        canvas.setNavEventHandlersActive(false);

        // Create PhotoMesa Component under canvas
        photomesa = new ZPhotoMesa(canvas);

        // Enable PhotoMesa event handler
        photomesa.setEventHandlersActive(true);

        // Options for the PhotoMesa Component
        photomesa.setThumbnailBase(null);
        photomesa.setAllowDrop(false);
        photomesa.setImageBorderWidth(0);
        photomesa.setShowProgress(true);
        photomesa.setConstantAnimationSpeed(500);

        // Event handler can be added to capture events
        // from the inside of PhotoMesa component
        photomesa.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if(e.getID() == ZPhotoMesa.ACTION_IMAGE_ON_FOCUS) {
                    ImageItem imageItem = (ImageItem)e.getSource();
                }
            }
        });

        this.setLayout(new BorderLayout());
        this.add(canvas, "Center");                            // Add PhotoMesa canvas to JPanel
    }
}

```

Figure 3.4 Adding *ZPhotoMesa* component inside a Java JPanel.

In Figure 3.4, ZCanvas is a basic Jazz component. It is a simple Swing component onto which other Jazz objects can be rendered. It also defines a default Jazz scene graph structure consisting of a root, a camera, and one node. Once a canvas is created, *ZPhotoMesa* can be added as one of its children. As shown in the example, creating and adding a *ZPhotoMesa* component is achieved essentially in one line, `photomesa = new ZPhotoMesa(canvas);`

When *ZPhotoMesa* is newly created, it does not have any information about images and, thus, it draws nothing on the screen. *PhotoMesaData* is another data structure which defines where *ZPhotoMesa* should look for images.

```
public class PhotoMesaData {
    public Vector getRegions();
    public void sort();
    public ImageItem copyImageItem(ImageItem src, Region region) throws Exception;
    public ImageItem linkImageItem(ImageItem src, Region region) throws Exception;

    public ImageItem add(Region region, ImageItem imageItem) throws Exception;
    public void rename(ImageItem imageItem, String newName) throws Exception;
    public void remove(ImageItem imageItem) throws Exception;

    public void addRegion(Region region) throws Exception;
    public void renameRegion(Region region, String newName) throws Exception;

    public Dimension getPreferredSize() throws Exception;
}
```

Figure 3.5 *PhotoMesaData* is a data type to hold information of images. It can be independently prepared without any restriction. PhotoMesa scene graph is built based on this information.

As shown in Figure 3.5, *PhotoMesaData* is used to store a list of images to be fetched by *ZPhotoMesa*. This data is totally independent from drawing. It only defines the way that images can be handled such as copy, link, add, and remove. Therefore, by using custom *PhotoMesaData*, *ZPhotoMesa* can be easily extended to load images from various sources such as local hard disk, web server, or database.

Applications which embed *ZPhotoMesa* should implement appropriate methods of *PhotoMesaData*, which can be achieved by creating a new class extending *PhotoMesaData*. The core method of *PhotoMesaData* is *getRegions()* which must be implemented for every subclass. Based on the return value of the *getRegions()* method, a *ZPhotoMesa* components builds a corresponding internal scene graph structure. Other methods in *PhotoMesaData* support supplementary actions such as add, remove, and link images. These non-core methods are required to be defined if not needed. For example, when there is no dynamic addition or removal of images, *add()* and *remove()* are never invoked. According to the interaction strategies of applications, only part of *PhotoMesaData* methods can be implemented.

Once *ZPhotoMesa* and *PhotoMesaData* are ready, loading is quite simple. Figure 3.6 shows an example procedure that links *ZPhotoMesa* with *PhotoMesaData*. In Figure 3.6, *SimplePhotoMesaData* is defined as an example subclass of *PhotoMesaData*. After creating regions by using *createRegion()* method, *SimplePhotoMesaData* adds a set of images by using *addImage()* method. The linkage is achieved by one simple line of code, *photomesa.layout(data);*.

```

ZPhotoMesa photomesa = Somewhere.getZPhotoMesa();
PhotoMesaData data = new SimplePhotoMesaData();

// Adding a new region "Frog"
Region region = data.createRegion("Frog");
data.addImage(region,
    new URL("file:///c:/queryKidsImages/umich/brown bat.jpg"));
data.addImage(region,
    new URL("file:///c:/queryKidsImages/umich/green frog.jpg"));
data.addImage(region,
    new URL("file:///c:/queryKidsImages/umich/wood frog.jpg"));
data.addImage(region,
    new URL("file:///c:/queryKidsImages/umich/wood frog3.jpg"));

// Adding a new region "Fish"
region = data.createRegion("Fish");
data.addImage(region,
    new URL("file:///c:/queryKidsImages/fish/aba aba.jpg"));
data.addImage(region,
    new URL("file:///c:/queryKidsImages/fish/protopterus.jpg"));

// Clear the PhotoMesa screen
photomesa.clear(true);

// Add the prepared regions on the screen
photomesa.layout(data);

```

Figure 3.6 An example of linking a *ZPhotoMesa* component with a *PhotoMesaData* object. A statement, *photomesa.layout(data);* enables *ZPhotoMesa* to build a scene graph by using information stored in *PhotoMesaData* and to show the images on the screen.

3.5 Integration with Other Applications

As explained in the previous section, PhotoMesa is redesigned to be a pluggable software component. In this section, I explain a couple of notable applications which embed PhotoMesa in their image navigation interfaces.

3.5.1 International Children's Digital Library (ICDL)

The International Children's Digital Library (ICDL) is a research project to develop innovative software and a collection of books that specifically address the needs of children as readers [19][32][57] and is currently deployed at <http://www.icdlbooks.org>. The primary goal of the research project is to provide access to literature that can enable children to understand the world around them and the global society. With participants from around the world, the ICDL is building an international collection that reflects both the diversity and quality of children's literature. Currently, the collection includes over 500 books in 27 languages.



Figure 3.7 International Children's Digital Library (ICDL) query interface.

Figure 3.7 shows how a PhotoMesa component is embed in the ICDL search interface. When users click a search category denoted by the red rectangle in the left figure, ICDL interface shows sub level categories on the main window (the center window of the right figure). When user clicks a leaf category, the chosen category is moved over the green caterpillar on the top (the first red circle in the right figure) and matching books begin to be loaded in the small PhotoMesa component (the second red circle in the right figure). Users can add/remove query conditions by clicking categories or caterpillar (Figure 3.7). Each added categories will be used to filter out books conjunctively. For example, adding “Spanish” under the language category will limit the result to books written in Spanish. This conjunctive Boolean filtering is known to be effective to younger audiences according to [57].

When users click the PhotoMesa component, it is zoomed in and provides a full view of book covers as in Figure 3.8.



Figure 3.8 PhotoMesa is embedded as an image browser inside ICDL.

The area inside the red circle in Figure 3.8 is embedded PhotoMesa. The same navigation strategy is used in the ICDL. Users can click the left mouse button to zoom in and the right mouse button (or press the ‘Enter’ key) to zoom out. A highlight rectangle that follows the cursor represents the area that users can zoom into. When users click the left button, the area denoted by the rectangle will be zoomed and fit into the whole screen. According to pilot studies, children at early ages can use the interface without big problems.

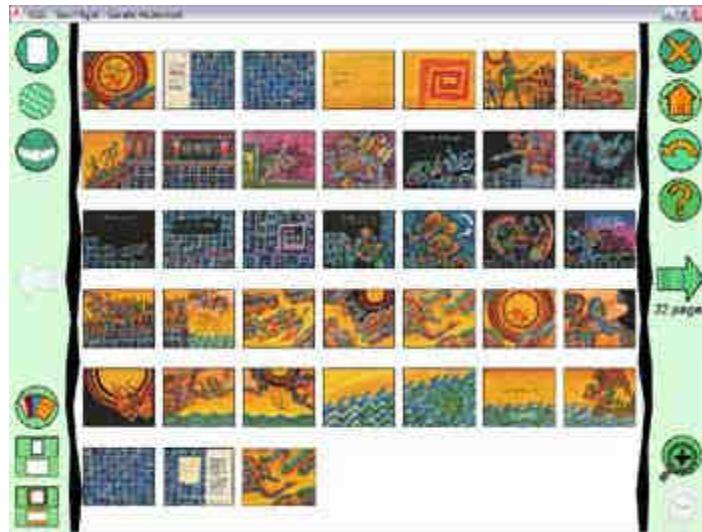


Figure 3.9 ICDL book reading interface. The example shows the Comic Strip reader out of three readers.

A book reader is shown on the screen (Figure 3.9) after users pick a book in the PhotoMesa component by selecting one book cover. In ICDL, three different book readers are provided for reading the content. Figure 3.9 shows one of the readers, the Comic Strip reader in which all the pages in a book are arranged on 2D grid. Users can use arrow keys or mouse to jump to any page that they want to see. This reader is

motivated by PhotoMesa. It follows the design ideas originated from PhotoMesa such as zoomable interface and multi-level thumbnails.

3.5.2 Maryland Interactive System for Image Searching

The department of art history and archaeology in the University of Maryland keeps a collection of approximately 300,000 slides, more than 10,000 digitized images, and several hundred archaeological artifacts. As the collection is used primarily by faculty and graduate students in the department, its content reflects the curriculum of the department. It is maintained also as a resource for the college of arts and humanities and is available to the entire university community.

The screenshot shows a web-based search interface titled "ISIS - Result for UM member". The search term "picasso" is entered in the search bar. Below the search bar, it says "General Search Terms: picasso" and "14 records displayed". The results are presented in a grid of four rows, each containing a thumbnail image and detailed information about a specific painting by Picasso. The first row shows "Still Life with Chair-Caneing" (1911-12), the second row shows "Three Musicians" (1921), the third row shows "Guernica" (1937), and the fourth row shows "The Old Guitarist" (1903). Each result includes the title, artist, creation date, creation place, copyright holder, dimensions, and material/medium.

Title	Artist	Creation Date	Creation Place	Copyright Holder	Dimensions	Material/Medium
Still Life with Chair-Caneing	Picasso, Pablo	20th Century,=1911-12		{SO:Gardner's Art Through the Ages, 9th ed., 22-9}	{Di:10 5/8" x13 3/4}	Oil and pasted paper simulating chair caning on canvas
Three Musicians	Picasso, Pablo	20th Century,=1921		{SO:Gardner's Art Through the Ages, 9th ed., 22-11}	{Di:6' 7" x7' 3 3/4}	Oil on canvas
Guernica	Picasso, Pablo	20th Century,=1937		{SO:Gardner's Art Through the Ages, ninth ed., 22-83}	{Di:approx. 11'6" X 25'8}	Mural
The Old Guitarist	Picasso, Pablo	20th Century,=1903		{SO:Janson, History of Art, 4th ed., Fig 960}	{Di:47 3/4 x 32 1/2}	Oil on panel

Figure 3.10 ISIS (Interactive System for Image Searching) interface. Search results are shown inside a long html page. Users have to scroll up and down to examine images in the results.

The department is actively digitizing the slides and has built a web-based image browsing prototype system called ISIS (Interactive System for Image Searching) [48] as shown in Figure 3.10. ISIS accepts keywords from users and returns matching images. However, the current prototype has some crucial interface issues.

First, search results are shown in web pages. This strategy has some obvious benefits. Users can use any web browser for querying images without installing any special software and the system can be accessed anywhere through the Internet. However, the web-based interface can show only about 5 images per page and users have to scroll up and down to examine the results.

Secondly, there is no notion of grouping in the result. Grouping the result can help users find the right information quickly; especially when users have no idea about what the result might be [13]. Grouping the results helps users filter out unwanted groups and focus on the relevant images.

Thirdly, comparison between images is not directly supported. Users have to remember what they want to compare and need to control scrollbar to locate them. ISIS interface sometimes returns more than 500 rows of information. Users have to scroll up and down to compare images, which is typically ineffective.

I began to address these issues by interviewing a group of art historians who are the intended users for the system. As a result, I identified a number of requirements for an art history image retrieval system. They are listed as follows:

- Fast preview

The size of ISIS search results is often large. For example, there are more than 700 images coming up when searching with keyword “renaissance”. Users must be able to review the result and filter out unwanted images efficiently. Fast preview is crucial for this task. Users should be provided with the fast visual summary of search results.

- Grouping related images

When using ISIS, typical tasks include choosing images from a set of related images. Therefore, an image retrieval system is required to present search results in meaningful groups. But, the way of forming group is not fixed for every search. For example, users want to group images by artist, by century, or by medium etc.

- Rapid Filtering (Query refinement)

The number of result images from the system is typically large and a search interface should allow users to filter out unwanted image efficiently. In many cases, this filtering is repeated as the user adds more conditions.

Some of the above requirements can be satisfied with the direct application of PhotoMesa techniques. PhotoMesa is capable of showing a large set of images aligned in groups and helping users recognize the characteristics of each group; hovering the mouse over images will popup a preview of them.

Motivated by this potential, I designed and implemented PhotoMesa ISIS to support the art history image collection. (Figure 3.11)

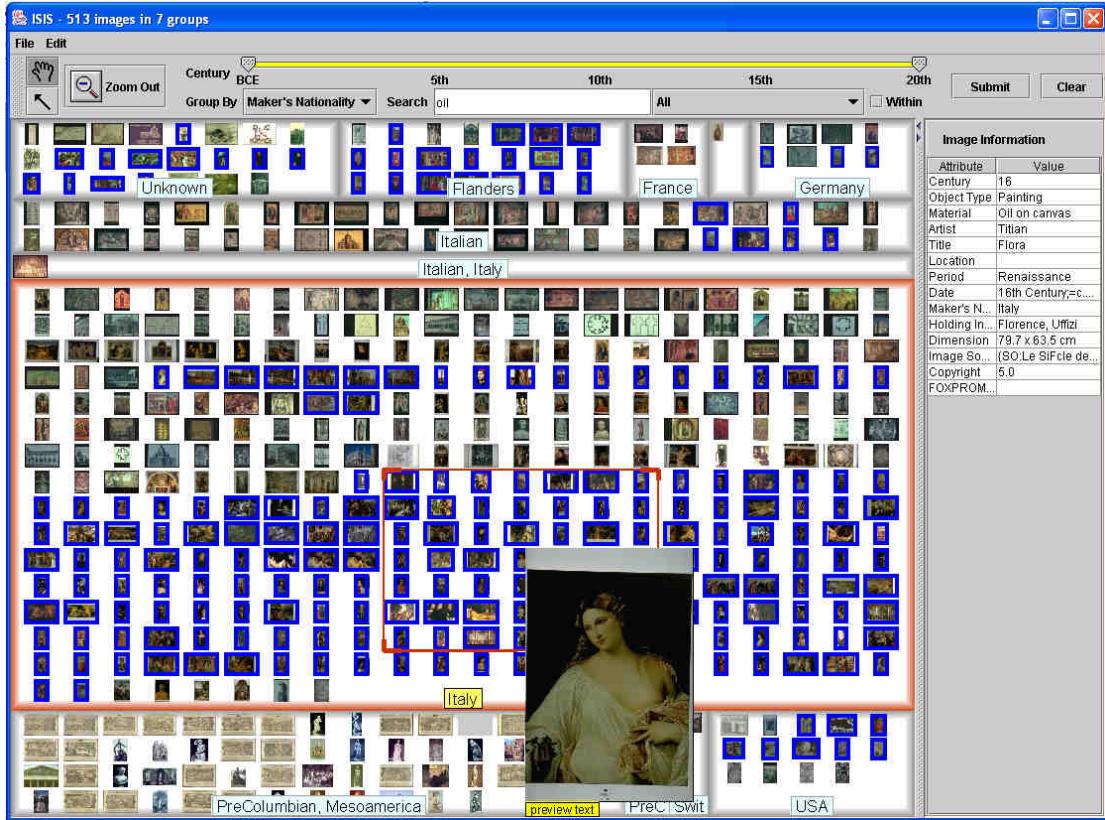


Figure 3.11 PhotoMesa ISIS. This figure shows an example of dynamic query preview. As a user types in a keyword, images that have matching metadata are highlighted so that users can easily identify patterns in results.

PhotoMesa ISIS embeds PhotoMesa as its core components as in Figure 3.11. The main window in the center is PhotoMesa canvas where search results are displayed. In addition to the basic navigation functions that PhotoMesa can provide, I also added a number of interface techniques to support art historians to specify sophisticated search conditions. The new techniques of PhotoMesa ISIS are as follows.

- Time slider



Figure 3.12 Double slider for specifying time conditions

The double slider, which is added on the top of the screen, allows users to specify time conditions. Users can slide each knob to choose a time period in which artifacts were created. As in Figure 3.12, the yellow region between the knobs represents a time period that a user selects.

- Search by keyword and dynamic preview



Figure 3.13 PhotoMesa ISIS search options

Figure 3.13 shows a text box with search options. Users can narrow down the search range by limiting the search category. For example, typing a keyword, “impression” in “Period” category will show only images that contain that string in the period field.

As a user is typing in a keyword, images on the screen that match the search condition are highlighted automatically with blue thick borders. For example, as a

user types “oil”, all the image that contains “oil” in their metadata are highlighted as shown in Figure 3.11. This dynamic preview is especially useful when users want to find patterns in the search results.

The keyword field also can be used when images are queried from database. When users click the “*Submit*” button on the top of the screen (Figure 3.11), specified search conditions are used to retrieve images from the database.

- Dynamic Grouping

As specified earlier, grouping is a crucial function for showing related images. PhotoMesa ISIS provides six categories under which images can be grouped as shown in Figure 3.14. The search results are displayed on the screen grouped by the chosen category (Figure 3.11). These categories are determined by domain experts (art historians), and chosen from metadata within the ISIS database.



Figure 3.14 Grouping and Searching options

Once the search results are retrieved, PhotoMesa ISIS allows users to regroup them dynamically on the screen. When users want to group the search results by a different

category, they can select one in “Group-By” category as in Figure 3.14. PhotoMesa ISIS immediately regroups them.

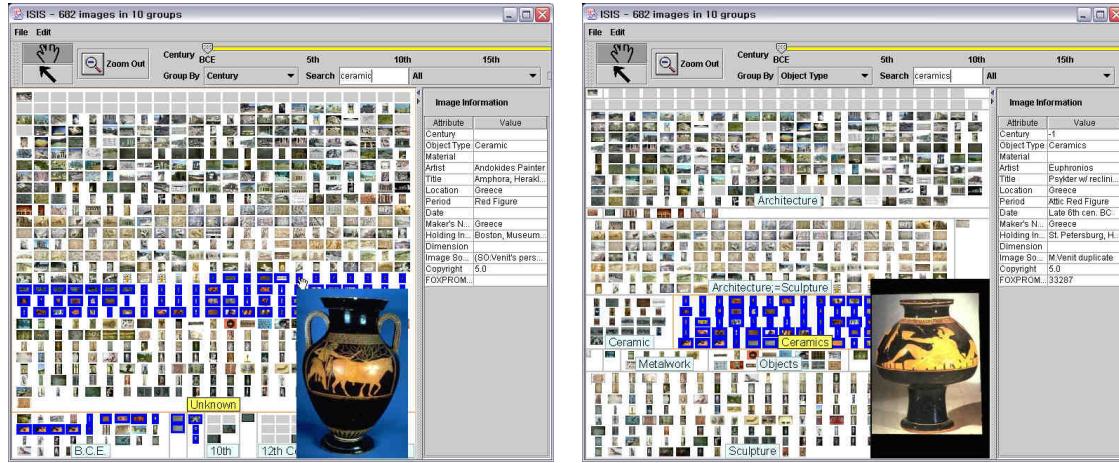


Figure 3.15 Dynamic Grouping.

Figure 3.15 shows an example of regrouping. In the left figure, the search results are grouped and ordered by “Century”. When a user selects “Object Type” in the combo box (Figure 3.14), images on the screen are regrouped by their object type (denoting types of artifacts such as oil painting, porcelain, building, etc.). This feature allows users to freely group images the way they want it.

Dynamic grouping can be especially useful when combined with dynamic preview. In the left figure of Figure 3.15, some images are highlighted by using dynamic preview. The images matching with a keyword, “ceramics” are highlighted. In this case, the highlighted images are scattered on the screen as shown in the left image. As a user regroups the result by “Object Type”, all the matching images become clustered into a single group and users are allowed to browse them much efficiently (the right figure of Figure 3.15).

3.6 Web Deployment and Other Applications

PhotoMesa is designed to be easily extensible and portable. Since PhotoMesa is entirely written in Java, it can be ported to be run in web browsers. Figure 3.16 exemplifies the application of the PhotoMesa applet running in Microsoft Internet Explorer.

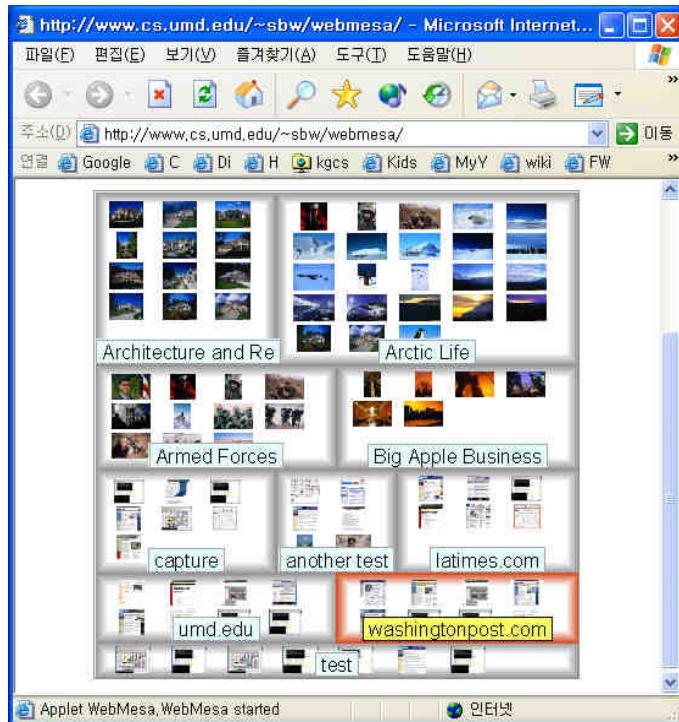


Figure 3.16 PhotoMesa can be run in a web browser

This ability opens various adaptation possibilities. It can be used as a front-end interface of image retrieval systems over the web. For example, search results of images.google.com [29] could be visualized using PhotoMesa. Since more and more information is available on the web, this ability can contribute to make the web more usable.

PhotoMesa can be easily adapted into other types of image browsing environments. As an example, the software architecture developed in PhotoMesa was applied to implement a virtual microscope. A virtual microscope enables users to explore huge-sized samples in a manner that is similar to real-life microscopes. With a simple modification – removing space between images on the screen, PhotoMesa can show one big image as a mosaic of smaller pieces of images. Figure 3.17 shows a running example of a virtual microscope, which can be used to handle big image files (>20MB) without loading them at once.

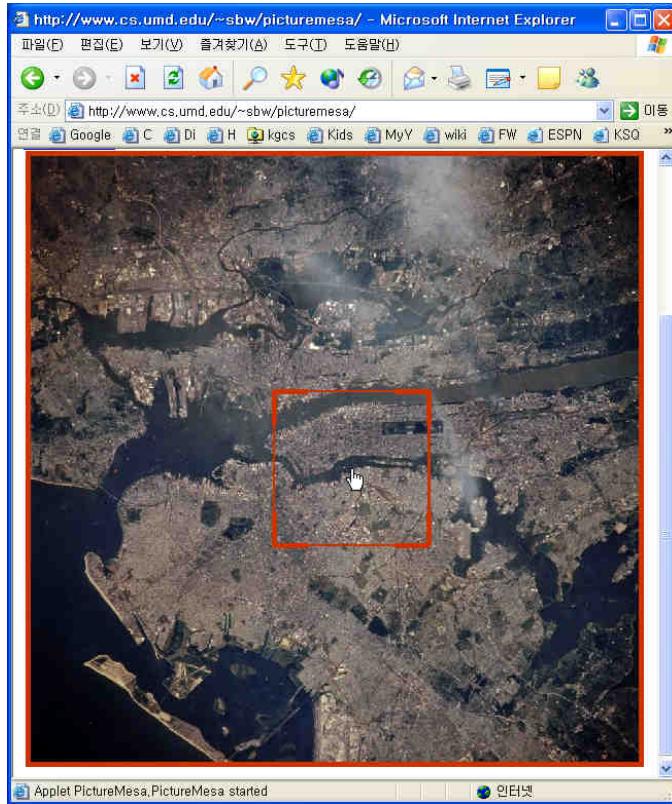


Figure 3.17 PhotoMesa is adapted to build a virtual microscope.

3.6 Summary and Discussion

In this chapter, I present my work on a zoomable image browser, PhotoMesa. Zoomable image browsing was introduced by Bederson [4]. He applied zoomable interface techniques into an image browsing environment as a solution to increase the browsability of image retrieval systems. As preliminary work, I enhanced PhotoMesa and applied zoomable image browsing techniques to several image retrieval systems such as ICDL and ISIS.

While PhotoMesa focuses on user interfaces for efficient browsing, there are also critical performance issues. I apply a number of techniques to enable PhotoMesa to show thousands of images on the screen with reasonable performance.

In addition, I define a set of programming interfaces so that other applications can embed PhotoMesa as their software component. I also demonstrate that PhotoMesa can be run in a commercial web browser and it can be easily extended into other type of applications such as a virtual microscope.

The experience gained in this preliminary research becomes a valuable starting point for the series of research in this dissertation.

Chapter 4

Automatic Thumbnail Cropping⁹

What we see depends mainly on what we look for. – John Lubbock

Thumbnails, miniature versions of original images, are widely used as abstract forms of original images. Combined with zoomable user interfaces, thumbnails provide seamless integration with original images. They are intuitive and easy to use. Thumbnails enable users to quickly scan large numbers of images on the screen in zoomed out views.

Recognizing the objects in an image is important in many retrieval tasks, but thumbnails generated by shrinking the original image often render objects illegible. We studied the ability of computer vision systems to detect key components of images so that intelligent cropping, prior to shrinking, can render objects more recognizable. We evaluate automatic cropping techniques 1) based on a method that detects salient portions of general images, and 2) based on automatic face detection. Our user study shows that these methods result in small thumbnails that are substantially more recognizable and easier to find in the context of visual search. This research has been collaborated with fellow graduate student Haibin Ling, and professors Dr. Benjamin B. Bederson and Dr. David Jacobs.

⁹ This research was published in the proceedings of UIST 2003 conference [66] and received the best student paper award.

4.1 Saliency and Thumbnails

Many image browsers generate thumbnails by shrinking the original image. [1][2][42] This method is simple. However, thumbnails generated this way can be difficult to recognize, especially when the thumbnails are very small. This phenomenon is not unexpected, since shrinking an image causes detailed information to be lost. An intuitive solution is to keep the more informative part of the image and cut less informative regions before shrinking. Our first method is a general cropping method based on the saliency map of Itti and Koch which uses a model of human visual attention [34][35]. A saliency map of a given image describes the importance of each position in the image. In our method, we use the saliency map directly as an indication of how much information each position in images contains. The merit of this method is that the saliency map is built up from low-level features only, so it can be applied to any image. We then select the portion of the image of maximal informativeness.



Figure 4.1: An example saliency map

4.2 Saliency Based Thumbnail Cropping¹⁰

We define the thumbnail cropping problem as follows: Given an image I , the goal of thumbnail cropping is to find a rectangle R_C , containing a subset of the image I_C so that the main objects in the image are visible in the subimage. We then shrink I_C to a thumbnail.

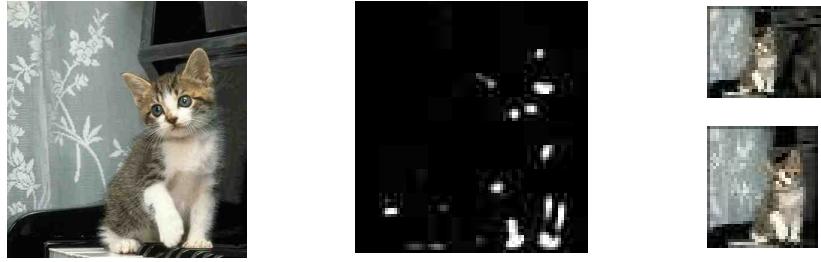


Figure 4.2: A cropped image from the previous example (Figure 4.1) and thumbnails from the original image and the cropped image

4.2.1 Find Cropping Rectangle with Fixed Threshold using Brute Force Algorithm

We use Itti and Koch's saliency algorithm because their method is based on low-level features and hence independent of semantic information in images.

Once the saliency map S_I is ready, our goal is to find the crop rectangle R_C that is expected to contain the most informative part of the image. Since the saliency map is used as the criteria of importance, the sum of saliency within R_C should contain most of the saliency value in S_I . Based on this idea, we can find R_C as the smallest

¹⁰ Haibin Ling and Dr. David Jacobs originally introduced this research.

rectangle containing a fixed fraction of saliency. To illustrate this formally, we define candidates set $\mathfrak{R}(\lambda)$ for R_C and the fraction threshold λ as

$$\mathfrak{R}(\lambda) = \left\{ r : \frac{\sum_{(x,y) \in r} S_I(x, y)}{\sum_{(x,y)} S_I(x, y)} > \lambda \right\}$$

Then R_C is given by

$$R_C = \arg \min_{r \in \mathfrak{R}(\lambda)} (\text{area}(r))$$

R_C denotes the minimum rectangle that satisfies the threshold defined above. A brute force algorithm was developed to compute R_C .

4.2.2 Find Cropping Rectangle with Fixed Threshold using Greedy Algorithm

The brute force method works, however, it is not time efficient. Two main factors slow down the computation. First, the algorithm to compute the saliency map involves several series of iterations. Some of the iterations involve convolutions using very large filter templates (on the order of the size of the saliency map). These convolutions make the computation very time consuming.

Second, the brute force algorithm basically searches all sub-rectangles exhaustively. While techniques exist to speed up this exhaustive search, it still takes a lot of time.

We found that we can achieve results that are nearly as good much more efficiently by: 1) squaring the saliency to enhance it; 2) using a greedy search instead of brute force method by only considering rectangles that include the peaks of the saliency.

```

Rectangle GREEDY_CROPPING (s, λ )

thresholdSum ← λ * Total saliency value in S
RC ← the center of S
currentSaliencySum ← saliency value of RC
WHILE currentSaliencySum < thresholdSum DO
    P ← Maximum saliency point outside RC
    R' ← Small rectangle centered at P
    RC ← UNION(RC, R')
    UPDATE currentSaliencySum with new region RC
ENDWHILE
RETURN RC
```

Figure 4.3: Greedy Cropping algorithm

Figure 4.3 shows the algorithm GREEDY_CROPPING to find the cropping rectangle with fixed saliency threshold λ . The greedy algorithm calculates R_C by incrementally including the next most salient peak point P . Also, when including a salient point P in R_C , we compute the union of R_C with a small rectangle centered at P . This is because if P is within the foreground object, it is expected that a small region surrounding P would also contain the object. When we initialize R_C we assume that the center of the input saliency map always falls in R_C . This is reasonable, since even when the most salient part does not contain the center (this rarely happens), it will not create much harm to our purpose of thumbnail generation. With this assumption, we initialize R_C to contain the center of the input saliency map.

Suppose we are finding a cropping rectangle inside an image of $n \times n$ dimension (n^2 pixels). With the brute force algorithm, we need to evaluate all possible sub-rectangles. Therefore, it requires $O(n^4)$ time¹¹.

However, with the greedy cropping algorithm, it takes only $O(n^2 \log n)$ time. First, sort the pixels in an image by order of saliency values ($O(n^2 \log n)$). Once the pixels are sorted, each pixel is processed just once ($O(n^2)$) if a smart data structure is utilized. Therefore, the total processing time is bounded by the sorting time $O(n^2 \log n)$.

4.2.3 Find Cropping Rectangle with Dynamic Threshold

Experience shows that the most effective threshold varies from image to image. We therefore have developed a method for adaptively determining the threshold λ .

Intuitively, we want to choose a threshold at a point of diminishing returns, where adding small amounts of additional saliency requires a large increase in the rectangle. We use an area-threshold graph to visualize this. The X axis indicates the threshold (fraction of saliency) while the Y axis shows the normalized area of the cropping rectangle as the result of the greedy algorithm mentioned above. Here the normalized area has a value between 0 and 1. The solid curve in Figure 4.4 gives an example of an area-threshold graph.

A natural solution is to use the threshold with maximum gradient in the area-threshold graph. We approximate this using a binary search method to find the

¹¹ Each sub-rectangle can be decided by two points, upper left corner and lower right corner. Therefore, its computing complexity is equal to choosing two points out of n^2 points, which is $\binom{n^2}{2} = O(n^4)$

threshold in three steps: First, we calculate the area-threshold graph for the given image. Second, we use a binary search method to find the threshold where the graph goes up quickly. Third, the threshold is tuned back to the position where a local maximum gradient exists. The dotted lines in Figure 4.4 demonstrate the process of finding the threshold for the image given in Figure 4.1.

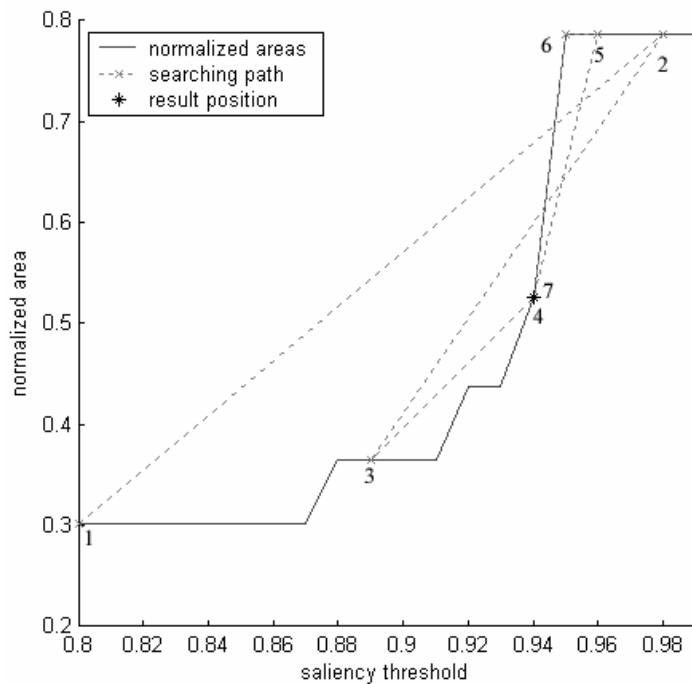


Figure 4.4: The solid line represents the area-threshold graph. The dotted lines show the process of searching for the best threshold. The numbers indicate the sequence of searching

4.3 Face Detection Based Thumbnail Cropping

Although the general saliency based method just described is useful, it does not consider semantic information in images. If our goal is to make the objects of interest in an image more recognizable, we can clearly do this more effectively when we are

able to automatically detect the position of these objects. We show that semantic information can be used to further improve thumbnail cropping, using automatic face detection. We choose this domain because a great many pictures of interest show human faces, and also because face detection methods have begun to achieve high accuracy and efficiency [73].

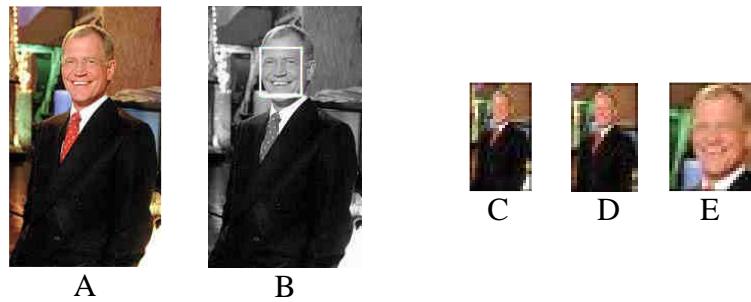


Figure 4.5 Left: An example face detection cropping. Original image (A) and face detection result (B). **Right:** Comparing three types of thumbnails. Plain shrinking (D), saliency based cropped thumbnail (E), and face-detection based cropped thumbnail (F).

For human image thumbnails, we claim that recognizability will increase if we crop the image to contain only the face region. Based on this claim, we designed a thumbnail cropping approach based on face detection. First, we identify faces by applying CMU’s on-line face detection [21][60] to the given images. Then, the cropping rectangle R_C is computed as containing all the detected faces. After that, the thumbnail is generated from the image cropped from the original image by R_C .

4.4 User Study Design

I ran a controlled empirical study to examine the effect of different thumbnail generation methods on the ability of users to recognize objects in images. The experiment is divided into two parts. First, I measured how recognition rates change depending on thumbnail size and thumbnail generation techniques. Participants were asked to recognize objects in small thumbnails (Recognition Task). Second, I measured how the thumbnail generation technique affects search performance (Visual Search Task). Participants were asked to find images that match given descriptions.

The recognition tasks were designed to measure the successful recognition rate of thumbnail images on three conditions, image set, thumbnail technique, and thumbnail size. The recognition correctness was measured as a dependent variable.

The visual search task conditions were designed to measure the effectiveness of image search with thumbnails generated with different techniques. The experiment employed a 3x3 within-subjects factorial design, with image set and thumbnail technique as independent variables. I measured search time as a dependant variable. But, since the face-detection clipping is not applicable to the Animal Set and the Corbis Set, the visual search tasks were omitted with those conditions as in Table 4.1. The total duration of the experiment for each participant was about 45 minutes.

4.4.1 Participants

There were 20 participants in this study (see Appendix A1 for user study material). Participants were college or graduate students at the University of Maryland at

College Park recruited on the campus. All participants were familiar with computers. Before the tasks began, all participants were asked to pick ten familiar persons out of fifteen candidates. Two participants had difficulty choosing them. Since the participants must recognize the people whose images are used for identification, the results from those two participants were excluded from the analysis.

4.4.2 Image Sets

Three image sets were used for the experiment. There were also filler images as distracters to minimize the duplicate exposure of images in the visual search tasks. There were 500 filler images and images were randomly chosen from this set as needed. These images were carefully chosen so that none of them were similar to images in the three test image sets.

Thumbnail Technique	Image Set		
	Animal Set	Corbis Set	Face Set
Plain shrunken thumbnail	√	√	√
Saliency based cropping	√	√	√
Face detection based cropping	X	X	√

Table 4.1 Design condition. 3X3 within subject factorial design. Two conditions were omitted because they are not applicable.

- Animal Set (AS)

The “Animal Set” includes images of ten different animals and there are five images per animal. All images were gathered from various sources of the Web. The reason I

chose animals as the target image was to test recognition and visual search performance of familiar objects. The basic criteria of choosing animals were 1) that the animals should be very familiar so that participants could recognize them without prior learning; and 2) they should be easily distinguishable from each other. As an example, donkeys and horses are too similar to each other. To prevent confusion, I only used horses.

- Corbis Set (CS)

Corbis is a well known source for digital images and provides various types of tailored digital photos [17]. Its images are professionally taken and manually cropped. The goal of this set is to represent images already in the best possible shape. I randomly selected 100 images out of 10,000 images. I used only 10 images as search targets for visual search tasks to reduce the experimental errors. But during the experiment, I found that one task was problematic because there were very similar images in the fillers and sometimes participants picked unintended images as an answer. Therefore, I discarded the result from the task. A total of five observations were discarded due to this condition.

- Face Set (FS)

This set includes images of fifteen well known people who are either politicians or entertainers. Five images per person were used for this experiment. All images were gathered from the Web. I used this set to test the effectiveness of face detection based

cropping technique and to see how the participants' recognition rate varies with different types of images.

Some images in this set contained more than one face. In this case, I cropped the image so that the resulting image contains all the faces in the original image. Out of 75 images, multiple faces were detected in 25 images. I found that 13 of them contained erratic detections. All erroneously detected faces were included in the cropped thumbnail sets since I intended to test our cropping method with available face detection techniques, which are not perfect.

4.4.3 Thumbnail Techniques

- Plain shrinking without cropping

The images were scaled down to smaller dimensions. Ten levels of thumbnails were prepared from 32 to 68 pixels in the larger dimension. The thumbnail size was increased by four pixels per level. But, for the Face Set images, I increased the number of levels to twelve with a maximum dimension of 76 pixels because I found that some faces are not identifiable even in a 68 pixel thumbnail.

- Saliency based cropping

By using the saliency based cropping algorithms described above, I cropped out the background of the images. Then the cropped images were shrunken to ten sizes of thumbnails. Table 4.2 shows how much area was cropped for each technique.

Cropping Technique and Image Set	Ratio	Variance
Saliency based cropping	Corbis Set	61.3%
	Animal Set	53.9%
	Face Set	54.3%
	All	57.6%
Face detection based cropping (Face Set)	16.1%	0.120

Table 4.2 Ratio of cropped to original image size

- Face detection based cropping

Faces were detected by CMU’s algorithm [21][60] as described above. If there were multiple faces detected, I chose the bounding region that contains all detected faces. Then twelve levels of thumbnails from 36 to 80 pixels were prepared for the experiment.

4.4.4 Recognition Task

The Animal Set and the Face Set images were used to measure how accurately participants could recognize objects in small thumbnails. First, users were asked to identify animals in thumbnails. The thumbnails in this task were chosen randomly from all levels of the Animal Set images. This task was repeated 50 times.

When the user clicked the “Next” button, a thumbnail was shown as in Figure 4.6 for two seconds. Since I intended to measure pure recognizability of thumbnails, I limited the time thumbnails were shown. According to a pilot user study, users tended to guess answers even though they could not clearly identify objects in thumbnails when they saw them for a long time. To discourage participants’ from guessing, the

thumbnails were hidden after a short period of time (two seconds). For the same reason, I introduced more animals in the answer list. Although only ten animals were used in this experiment, 30 animals are listed as possible answers as seen in Figure 4.6, to limit the subject's ability to guess identity based on crude cues. In this way, participants were prevented from choosing similarly shaped animals by guess. For example, when participants think that they saw a bird-ish animal, they would select swan if it is the only avian animal. By having multiple birds in the candidate list, those undesired behaviors could be prevented.

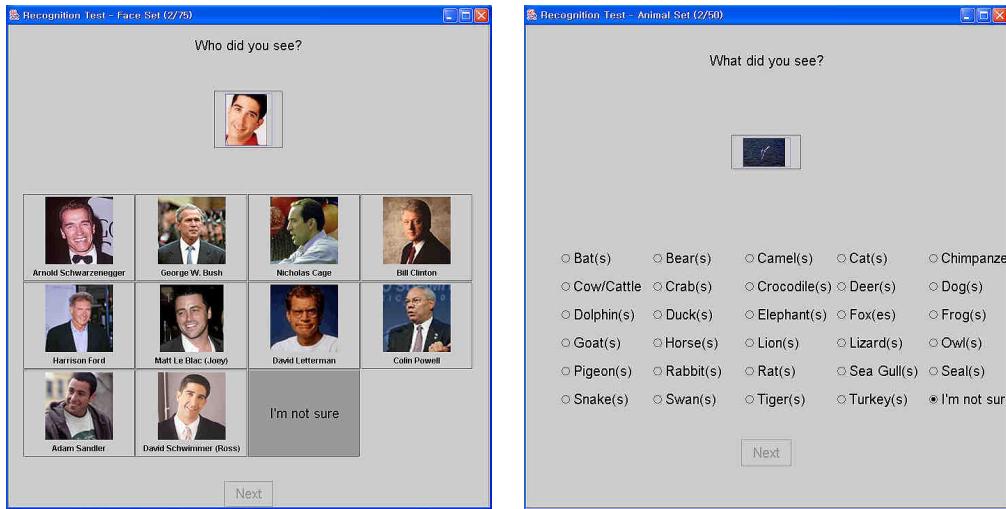


Figure 4.6 Recognition task interfaces. Participants were asked to click what they saw or the "I'm not sure" button. Left: Face Set recognition interface, Right: Animal Set recognition interface

After the Animal Set recognition task, users were asked to identify a person in the same way. This Face Set recognition task was repeated 75 times. In this session, the candidates were shown as portraits in addition to names as seen in Figure 4.6.

4.4.5 Visual Search Task

For each testing condition in Table 4.1, participants were given two tasks. Thus, for each visual search session, fourteen search tasks were assigned per participant. The order of tasks was randomized to reduce learning effects.

As shown in Figure 4.7, participants were asked to find one image among 100 images. For the visual search task, it was important to provide equal search conditions for each task and participant. To ensure fairness, I designed the search condition carefully. I suppressed the duplicate occurrences of images and manipulated the locations of the target images.

For the Animal Set search tasks, one target image was chosen randomly out of 50 Animal Set images. Then, 25 non-similar looking animal images were carefully selected. After that they were mixed with 49 more images which were randomly chosen from the filler set as distractors. For the Face Set and Corbis Set tasks, the task image sets were prepared in the same way.

The tasks were given as verbal descriptions for the Animal Set and Corbis Set tasks. For the Face Set tasks, a portrait of a target person was given as well as the person's name. The given portraits were separately chosen from an independent collection so that they were not duplicated with images used for the tasks.

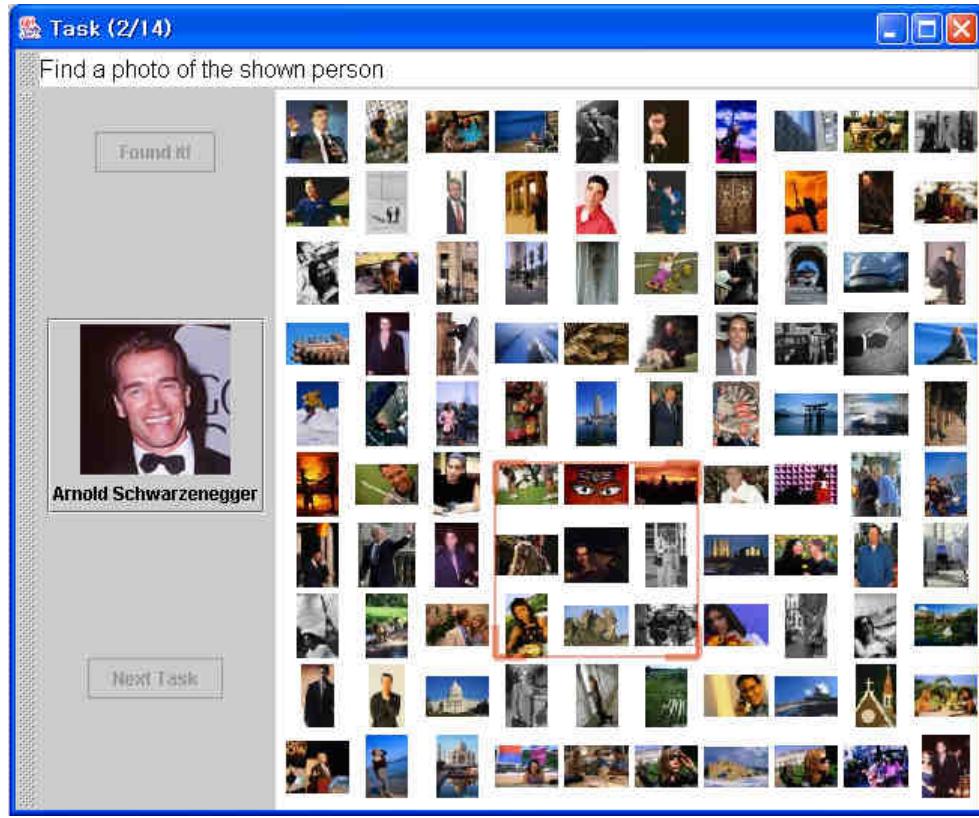


Figure 4.7 Visual search task interface. Participant were asked to find an image that matches a given task description. Users can zoom in, zoom out, and pan freely until they find the right image.

I used a custom-made image browser based on PhotoMesa [4] as our visual search interface. PhotoMesa provides a zooming environment for image navigation with a simple set of control functions. Users click the left mouse button to zoom into a group of images (as indicated by a red rectangle) to see the images in detail and click the right mouse button to zoom out to see more images to overview. Panning is supported either by mouse dragging or arrow keys. PhotoMesa can display a large number of thumbnails in groups on the screen at the same time. Since this user study was

intended to test pure visual search, all images were presented in a single cluster as in Figure 4.7.

Participants were allowed to zoom in, zoom out and pan freely for navigation. When users identify the target image, they were asked to zoom into the full scale of the image and click the “Found it” button located on the upper left corner of the interface to finish the task. Before the visual search session, they were given as much time as they wanted until they found it comfortable to use the zoomable interface. Most participants found it very easy to navigate and reported no problem with the navigation during the session.

4.5 Recognition Task Result

Figure 4.8 shows the results from the recognition tasks. The horizontal axis represents the size of thumbnails and the vertical axis denotes the recognition accuracy. Each data point in the graph denotes the successful recognition rate of the thumbnails at that level. As shown, the bigger the thumbnails are, the more accurately participants recognize objects in the thumbnails. And this fits well with our intuition. But the interesting point here is that the automatic cropping techniques perform significantly better than the original thumbnails.

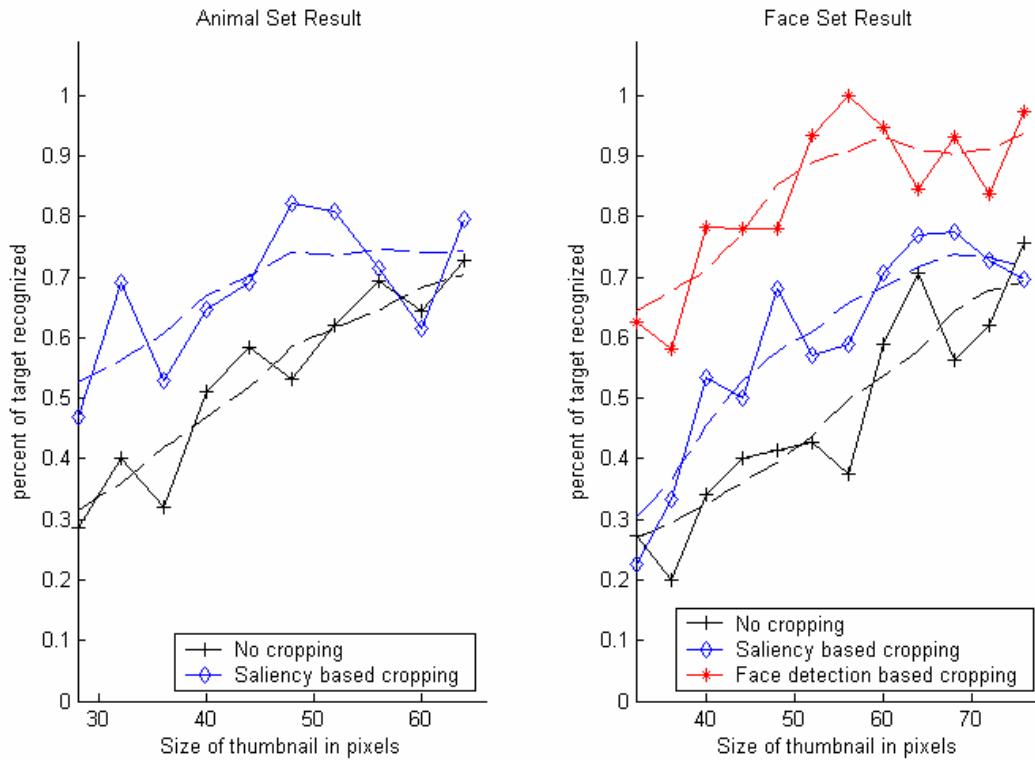


Figure 4.8 Recognition Task Results. Dashed lines are interpolated from jagged data points

There were clear correlations in the results. Participants recognized objects in bigger thumbnails more accurately regardless of the thumbnail techniques. Therefore, Paired T-test (two tailed) was used to analyze the results. The results are shown in Table 4.3.

The first graph shows the results from the “Animal Set” with two different thumbnail techniques, no cropping and saliency based cropping. As shown in Figure 4.8, users were able to recognize objects more accurately with saliency based cropped thumbnails than with plain thumbnails with no cropping. One of the major reasons for the difference can be attributed to the fact that the effective portion of images is drawn relatively larger in saliency based cropped images. But, if the main object region is cropped out, this would not be true. In this case, the users would see more

non-core part of images and the recognition rate of the cropped thumbnails would be less than that of plain thumbnails. The goal of this test is to measure if saliency based cropping cut out the right part of images. Even when there were errors in cropping, I included them in the user study test sets. As shown in Figure 4.8, the recognition test result showed that participants recognized objects better with saliency based thumbnails than plain thumbnails. Therefore, I can conclude that saliency based cropping does not cut out the core part of images.

Condition	<i>t</i> -Value	P value
No cropping vs. Saliency based cropping on Animal Set	$t(9) = 4.33$	0.002
No cropping vs. Saliency based cropping on Face Set	$t(11) = 4.158$	0.002
No cropping vs. Face Detection based cropping on Face Set	$t(11) = 9.556$	< 0.001
Saliency based cropping vs. Face detection based cropping on Face Set	$t(11) = 7.337$	< 0.001
Animal Set vs. Face Set with no cropping	$t(9) = 4.997$	0.001
Animal Set vs. Face Set with saliency based cropping	$t(9) = 3.077$	0.005

Table 4.3 Analysis results of Recognition Task (Paired T-Test). Every curve in Figure 4.8 is significantly different from each other.

During the experiment, participants mentioned that the background sometimes helped with recognition. For example, when they saw blue background, they immediately suspected that the images would be about sea animals. Similarly, the camel was well

identified in every thumbnail technique even in very small scale thumbnails because the images have unique desert backgrounds (4 out of 5 images).

Since saliency based cropping cuts out large portion of background (42.4%), I suspected that this might harm recognition. But the result shows that it is not true. Users performed better with cropped images. Even when background was cut out, users still could see some of background and they got enough help from the information. It implies that the saliency based cropping is well balanced. The cropped image shows main objects bigger while giving enough background information.

The second graph shows results similar to the first. The second graph represents the results from the “Face Set” with three different types of thumbnail techniques, no cropping, saliency based cropping, and face detection based cropping. As seen in the graph, participants perform much better with face detection based thumbnails. It is not surprising that users can identify a person more easily with images with bigger faces.

Compared to the Animal Set result, the Face Set images are less accurately identified. This is because humans have similar visual characteristics while animals have more distinguishing features. In other words, animals can be identified with overall shapes and colors but humans cannot be distinguished easily with those features. The main feature that distinguishes humans is the face. The experimental results clearly show that participants recognized persons better with face detection based thumbnails.

However, the results also show that saliency cropped thumbnails is useful for recognizing humans. I found that people in photos are usually included in saliency based cropped images. The test results show that the saliency based cropping does increase the recognition rate of identifying people in photos.

In this study, I used two types of image sets and three different thumbnail techniques. To achieve a higher recognition rate, it is important to show major distinguishing features. If well cropped, small sized thumbnail would be sufficient to represent the whole image. Face detection based cropping shows benefits when this type of feature extraction is possible. But, in a real image browsing task, it is not always possible to know users' searching intention. For the same image, users' focus might be different for browsing purposes. For example, users might want to find a person at some point, but the next time, they would like to focus on costumes only. I believe that the saliency based cropping technique can be applied in most cases when semantic object detection is not available or users' search behavior is not known.

In addition, the recognition rate is not the same for different types of images. It implies that the minimum recognizable size should be different depending on image types.

4.6 Visual Search Task Result

Figure 4.9 shows the result of the visual search tasks. Most participants were able to finish the tasks within the 120 second timeout (15 timeouts out of 231 tasks) and also

chose the desired answer (5 wrong answers out of 231 tasks). Wrong answers and timed out tasks were excluded from the analysis.

A two way analysis of variance (ANOVA) was conducted on the search time for two conditions, thumbnail technique and image sets. As shown, participants found the answer images faster with cropped thumbnails. Overall, there was a strong difference for visual search performance depending to thumbnail techniques, $F(2, 219) = 5.58$, $p = 0.004$.

Since I did not look at face detection cropping for the Animal Set and the Corbis Set, another analysis was performed with the two thumbnail techniques (plain thumbnail, saliency based cropped thumbnail) to see if the saliency based algorithm is better. The result shows a significant improvement on visual search with saliency based cropping, $F(1, 190) = 3.823$, $p = 0.05$. I therefore believe that the proposed saliency based cropping algorithm make a significant contribution to visual search.

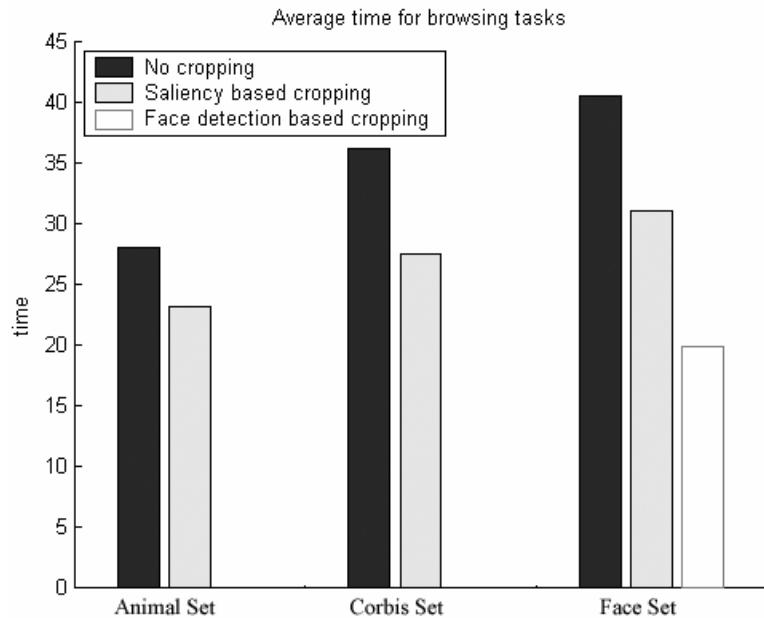


Figure 4.9 Visual search task results.

Condition	F value	P value
Thumbnail techniques on three sets	$F(2, 219) = 5.58$	0.004
Thumbnail techniques on Face Set	$F(2, 87) = 4.56$	0.013
No cropping vs. Saliency based thumbnail on three image sets	$F(1, 190) = 3.82$	0.052
Three image sets regardless of thumbnail techniques	$F(2, 219) = 2.44$	0.089

Table 4.4 List of ANOVA results from the visual search task

When the results from the Face Set alone were analyzed by one way ANOVA with three thumbnail technique conditions, there also was a significant effect, $F(2, 87)=4.56$, $p = 0.013$. But for the Animal Set and the Corbis Set, there was only a borderline significant effect over different techniques. I think that this is due to the

small number of observations. I believe those results would also be significant if there were more participants because there was a clear trend showing an improvement of 18% on the Animal Set and 24% on the Corbis Set. Lack of significance can also be attributed to the fact that the search task itself has large variances by its nature. I found that the location of a search target affects the visual search performance. Users begin to look for images from anywhere in the image space (Figure 4.7). Participants scanned the image space from the upper-left corner, from the lower-right corner, or sometimes randomly. If the search target image is located in the initial position of users' attention, it would be found much earlier. Since I could not control users' behavior, I randomized the location of the search target images. But as a result, there was large variance.

Before the experiment, I was afraid that the cropped thumbnails of the Corbis Set images would affect the search result negatively since the images in the Corbis Set are already in good shape – professionally taken and manually cropped - and I was concerned that cutting off their background would harm participants' visual search. But according to our result, saliency based cropped thumbnails does not harm users' visual search. Rather, it showed a tendency to increase participants' search performance. I think that this is because saliency based cropping algorithm cut the right amount of information without removing core information in the images. At least, I can conclude that it did not make visual search worse to use the cropped thumbnails.

Another interesting thing I found is that the visual search task with the Animal Set tends to take less time than with the Corbis Set and the Face Set, $F(2, 219) = 2.44$, $p = 0.089$. This might be because the given Corbis Set and Face Set tasks were harder than the Animal Set. But, there was another interesting factor. During the experiment, when he found the answer image after a while, one participant said that “*Oh... This is not what I expected. I expected blue background when I'm supposed to find an airplane.*” During the experiment sessions, it was observed that the participant passed over the correct answer image during the search even though he saw the image at reasonably big scale. Since the Animal Set and the Corbis Set tasks were given as verbal descriptions, users did not have any information about what the search target images would be like. I think that this verbal description was one of the factors in performance differences between image sets because it was observed that animals are easier to find by guessing background than other image sets.

4.7 Summary and Discussion

We developed and evaluated two automatic thumbnail generating methods. A general thumbnail cropping method based on a saliency model finds the informative portion of images and cuts out the non-core periphery. Thumbnail images generated from the cropped part of images increases users’ recognition and helps users in visual search. This technique is general and can be used without any prior assumption about images since it uses only low level features. Furthermore, the technique is safe to be used for pre-cropped images because it reduces the over or under cropping of an image.

When semantic information such as a face is available, the crop area can be determined more effectively. The face detection based cropping technique demonstrates how semantic information can be used to enhance thumbnail cropping.

I performed a user study that shows strong empirical evidence supporting our hypotheses. I assumed that the more salient a portion of image, the more informative it is. I also presumed that using more recognizable thumbnails would increase visual search performance.

During the experiment, I found it interesting that users had a tendency to have mental models about search targets. Some users develop a specific model about what a target will look like by guessing its color and shape. It was observed that participants spent more time searching when the actual search target was different from what they had in mind, their mental model. Some participants even skipped the correct search target when their model and the actual target did not match. The same thing happened when participants were unable to guess because of the ambiguity of the given tasks. It is known that humans have an “attentional control setting” – a mental setting about what they are (and are not) looking for while performing a given task. Interestingly, it is also known that humans have difficulty in switching their attentional control setting instantaneously [24]. This theory explains my observation. I think that this phenomenon should be regarded in designing image browsing interfaces especially in situations where users need to skim a large number of images or when users are required to visually search information such as in [67].

It was also observed that participants used various visual search strategies. Some participants searched images from the upper left corner and scanned images horizontally while some others begun to search from the bottom right corner and scanned images vertically. Some of them did not seem to have any search pattern at all and their eyes randomly traversed the image space. On the other hand, with scroll bar interfaces, most users tend to scan images from left to right and from up to down just like they read a book.

The saliency based thumbnail cropping is based on the idea that the saliency is a measure for the informativeness. I think this idea can be applied to other domains. For example, sometimes it is useful to identify which part of web pages tends to attract humans' attention. Or it can be extended to recognize which parts of video clips have more information. I hope future research will extend this research for other domains.

One practical concern in promoting the use of the automatic thumbnail cropping is its performance. Since the thumbnail cropping algorithm is written in Matlab, it is very slow and not practical in a real world setting. The reimplementation of the algorithm in more efficient environments such as C/C++ will speed up the thumbnail generation significantly. Performance issues did not effect these studies since all cropping was performed offline.

Currently, the cropping algorithm does not involve users in deciding its cropping regions. I think that interactive image cropping is another good example of automatic recognition systems might help users. The automatic cropping can provide users with a firsthand suggestion and let users confirm what an automatic system provides.

Chapter 5

Semi-Automatic Photo Annotation

Premature optimization is the root of all evil. – Donald Knuth

It is better to have enough ideas for some of them to be wrong, than to be always right

by having no ideas at all. – Edward de Bono

Thus far, I have described work done to navigate and browse images on the screen.

Along with browsing, searching is another important axis of information retrieval.

Especially when users have to deal with a huge volume of information, search is a very useful technique for locating information efficiently. However, searching usually requires information to be pre-indexed. As explained in the earlier chapters, metadata associated with images is hard to be obtained for many reasons. In this chapter, I detail the problems with metadata acquisition. I, then, explain the concept of semi-automatic annotation and how this approach can benefit acquiring metadata associated with photographs.

5.1 Metadata and Annotation

5.1.1 Metadata Acquisition

Annotation is defined as a process which involves labeling the semantic content of images (or objects in images) with a set of keywords or semantic information. Annotated information is very important for image retrieval since it allows keyword-based search. There has been much research to ease this annotation process.

From Devices	File name, file size, EXIF [20] information such as shutter speed
Image Analysis	Low level visual features such as texture, color, blobs
From Context	Captions, surrounding text in a web page
Manual Annotation	Accurate, relevant annotation. Very slow and users don't like to do manual annotation.

Table 5.1 Acquiring metadata associated with images

Some basic information can be directly obtained from images or image devices. File names, file size, file date and EXIF information can be easily acquired. But, these metadata does not have much value for users, especially for casual users who want to manage their own personal photos. For example, an image file name “*IMG_2345.jpg*” is not very useful.

There have been a number of research studies to extract useful metadata directly from images. QBIC [23] tried to use image-based analysis techniques to extract metadata. QBIC allows users to specify search conditions based on low level visual features

such as color, texture, and so on. For example, users can issue queries such as “find images which have red objects in the center”. However, the metadata extracted by automatic feature extraction is not very relevant in many cases. For personal photos, higher level information such as location, event, or person in photos would be more relevant and interesting to users.

As an alternative way of obtaining metadata associated with images, some researchers have used the context of images to improve understanding. Shen *et al.* [63] used the textual context of web pages to extract descriptive information of images on the same pages. This type of approach can be applied to images with captions or with pre-annotated keywords. But, this approach is not applicable for general images since it assumes appropriate context. It may not work for images without further information.

While these automatic approaches can provide limited metadata, the automatically obtained information inevitably involves recognition errors. The errors usually hinder direct usage of the acquired metadata in image retrieval systems. Furthermore, even though the acquired metadata is correct for general usage, it might not be useful to all users. The obtained metadata may be too general to satisfy the need of every individual user. Each user needs various types of metadata according to his/her own interest. Furthermore, there are numerous cases where it is even impossible to automatically obtain metadata without the intervention of humans. The inaccuracy and irrelevancy are the fundamental problems with automatic recognition systems.

On the other hand, there is a manual approach where users can explicitly decide which information should be added on a specific image. The actual users, as

information consumers, can function as the most reliable source of accurate and relevant metadata associated with images. But, it is well known that most users do not want to spend much time creating and annotating metadata for images. Kang *et al.* [39] developed a direct annotation method that focuses on labeling names of people in photos. While it saves users typing work, users still have to perform drag and drop many times. Manual annotation is usually labor intensive and tedious.

Semi-automatic annotation combines the two techniques, automatic metadata extraction and manual annotation. The basic idea of semi-automatic annotation is to add users' feedback onto metadata that was automatically extracted. When the metadata has reasonable accuracy; the amount of erratic information is less than that of correct information, the correcting errors can be faster and easier than adding new information. The goal of the strategy is to provide users with an efficient annotation method and accurate search results.

5.1.2 Metadata for Personal Photos

Various types of metadata can be associated with images through either manual annotation or automatic acquisition. The metadata can vary from low level features such as colors and texture to high level abstract information such as captions and keywords. Some researchers tried to identify common types of metadata that are general enough so that they can be useful for most users.

Rodden *et al.* [58] observed users' behavior with their digital personal photographs and found that there are specific types of metadata that the participants in their study

commonly wanted to use to browse their personal photos. The participants wanted to browse photos by event, rather than querying them based on more specific properties. Along with event information, some users regarded location as another very important type of information. However, in most cases, location information is tightly coupled with event information. When personal photos are taken in a relatively short period time, the photos usually tend to have the same event and location. For example, an event, “camping trip on June 10th”, would be held on a single location. Thus, event information and location information usually have strong association with each other especially for person photo collections.

Rodden *et al.* [58] also found that the participants in their study were 1) automatically sorting photos in chronological order; and 2) displaying a large number of thumbnails at once. The first observation clearly emphasizes the importance of the chronological order of photos.

People in photos are regarded as one of the most important pieces of information because a great many pictures of interest show human faces many of which are central objects in the images. It is not surprising that many image browsing prototypes and products [2][39][42][62] include features of labeling persons with metadata such as names. Rodden *et al.* [58] also hinted that robust face recognition would help users to browse their personal photo collections.

It is obvious that tools for managing personal photos are required to support the above three types of metadata, event, chronological order, and people in photos, as well as other subjectively preferred types.

5.2 Semi-Automatic Annotation

The semi-automatic strategy is to let users correct automatically extracted metadata based on the hypothesis that such automatically extracted metadata will have errors and that correcting those errors will be faster than completing manual annotation. The semi-automatic strategy allows users to incrementally and interactively increase metadata on photo collection.

The conceptual information flow of semi-automatic annotation is shown in Figure 5.1.

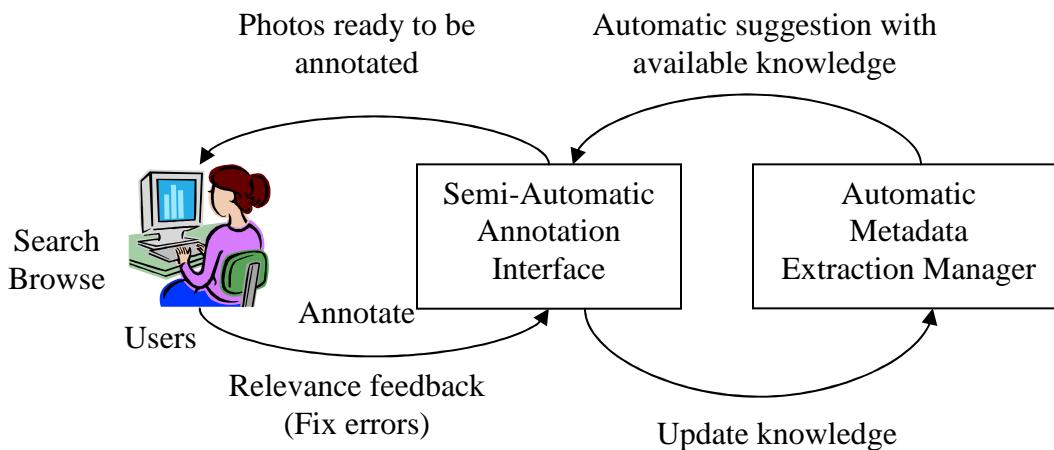


Figure 5.1 The information flow cycle of semi-automatic annotation

By its nature, automatic metadata extraction generates results compromised by recognition errors. Initially, a semi-automatic annotation interface accepts the raw results from the automatic metadata extraction manager. The user interface provides users the opportunity to give feedback while browsing and searching. Users are allowed to correct the errors in the extracted information. The users' correction (or relevance feedback) is used as an input for the automatic metadata extraction manager.

to increase its accuracy. Users' annotations are also fed back into the automatic extraction manager and used to generate more accurate metadata extraction. As users keep using the system, the overall accuracy, as well as the quantity of metadata, increases since more reliable metadata are added by the users.

Among the data flow in Figure 5.1, my research focuses on the interaction between users and semi-automatic user interfaces.

5.3 Semi-Automatic Annotation Design Principles

While designing a semi-automatic annotation interface, I considered a number of principles. In this section, I present some of principles that are focused on facilitating efficient annotations as well as searching and browsing images.

- Bulk annotation

Bulk annotation, where multiple images are annotated with a single user action, can accelerate users' performance when adding metadata to images. Rather than repeatedly selecting images and making annotations one by one, making annotations on selected multiple images can speed up the annotation process. However, the speed-up is achieved only when selecting multiple annotation targets is easy enough. If the selection takes too long, there will be no benefit. A semi-automatic annotation interface, therefore, should be carefully designed to allow users to choose multiple images efficiently. For example, when images that share common or similar information are located closely together on the screen, they can be a good candidate

for bulk annotation. Items which are semantically close with each other are desired to be laid out together to facilitate bulk annotation.

- Transparent interface

The relationship between automatic extraction and users' relevance feedback mechanism should be understood clearly. Koenemann and Belkin [40] observed that users perform better when they understand underlying algorithms. They showed that increasing the transparency of relevance feedback improves how effectively users take advantage of it. An interface should provide clear information about how it processes information. For example, MiAlbum [71] allows users to make decisions on automatically extracted information by using thumbs up/down metaphor. [71] reports that their feedback metaphor was not very clear to users and confused users because of its lack of transparency.

- Users in control

Users should be in control at all times. Automatically extracted metadata should be a suggestion to users. Users must have a freedom to make their own annotation as they want to. A semi-automatic annotation interface should not block or interfere users' manual overriding.

- Show context information

While showing alternatives is one nice feature for general user interfaces, it is especially important for semi-automatic annotation user interfaces. Since

automatically extracted metadata often contains errors, users have to be provided with options to choose substitute information.

However, providing all the available information on the screen is not a good design strategy either. A user interface should prioritize available information and provide just the right number of alternatives.

- Incremental and interactive annotation

Users must not be forced to make annotations. An interface should allow users to make annotations at any time. Users should be allowed to make annotations on important and interesting images first and other images later when they feel like it.

5.4 Semi-Automatic Photo Annotation and Recognition

Interface (SAPHARI)

Based on the design principles in the previous section, I designed and implemented a research prototype, SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface) to help users manage their personal photo collections by using automatic recognition systems.

SAPHARI is not only an annotation interface. It also allows users to browse and search their photo collections. As shown in Figure 5.2, SAPHARI uses zoomable user interface techniques that were applied to PhotoMesa (see Chapter 3). Users can navigate a 2D zoomable image space with zooming and panning. Photographs are also laid out on the screen by using the quantum strip treemap algorithm [4].

However, while PhotoMesa depends on basic metadata such as directory, date, and filename to form image groups, SAPHARI takes advantage of automatic recognition algorithms. SAPHARI generates image clusters which facilitate efficient bulk annotation. SAPHARI uses hierarchical event identification (see section 5.5) and clothing based human recognition (see section 5.6) to cluster photos along with the basic metadata. By using the acquired metadata, SAPHARI provides multiple views for users' photo collections. SAPHARI is capable of creating photo groups by event, month, year, directory, and person. Those groups play very important role in assisting users to make bulk annotations. For example, when users want to annotate event information, providing photos grouped by event will be very useful because users can easily choose multiple photos in the target event.

Users can start to use SAPHARI by choosing directories that they want to manage. Once they choose folders, SAPHARI automatically searches all the image files in the folders and stores the image information into a database. Users can choose the “*Grouping Tab*” to load images in the database. SAPHARI provides “*Fine*” event grouping, “*Regular*” event grouping, “*People*” grouping, month grouping, year grouping, and directory grouping. It also allows users to form custom grouping. As users select a tab in the “*Grouping Tab*”, SAPHARI immediately lays out photos based on the selected grouping method. Users can make annotations by dragging a label from “*Metadata Pane*” onto a photo or a group of photos.

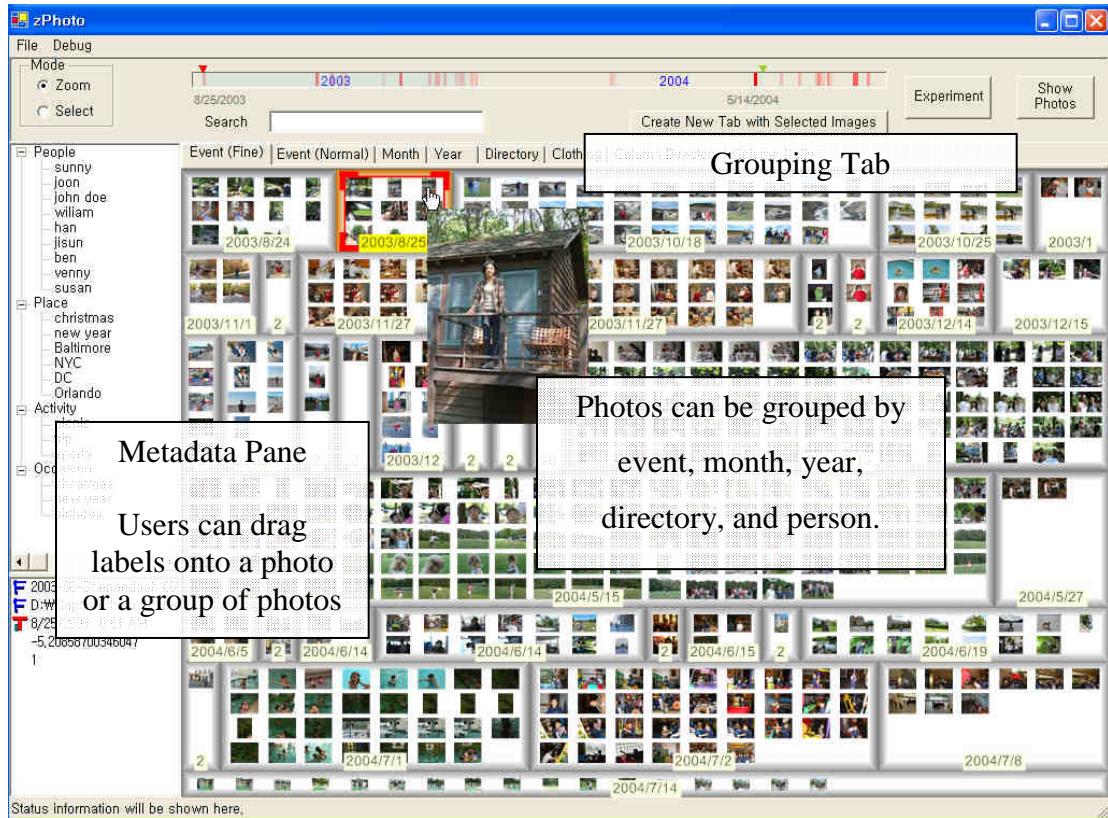


Figure 5.2 SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface)

SAPHARI does not require users to make annotations. As users browse and search photos collections, they can make annotations whenever they want to. Also, users can modify inaccurate suggestions that automatic recognition systems have made. Users' amendments are fed back into SAPHARI and used to increase the accuracy of automatic suggestions. In this way, SAPHARI enables users to make annotations interactively and incrementally. The detailed design and implementation of SAPHARI are discussed in the following sections.

5.5 Event Identification

Time information plays an important role when classifying personal photos since they usually have *temporal locality* [25]. In other words, when photos are close in time with each other, they have a high probability of sharing common or similar information. For example, photos that have been shot in one day would have a better chance of sharing common information than photos taken several months apart. The motivation of time-based event identification is based on the assumption that the effort needed for annotation can be reduced dramatically because of temporal locality in personal photo collection. Given the temporal locality, photos can be prepared in groups according to their timestamps so that they can be bulk-annotated. Users can make annotations on automatically prepared image groups rather than on a single image one by one.

As stated earlier, "event" is one of the most important units for personal photo organization. There has been a number of research to find meaningful event clusters from image collections [16][25][42][56]. Time based event identification is achievable due to the fact that personal photo collection is usually *bursty* or *episodic* with respect to the temporal order of photos in it [25]. In most cases, casual users don't take photos on a regular basis, such as one shot a day. When there are interesting things and a user has a camera, he or she usually takes a relative large number of photos in a short period of time. Then, there may be a relatively long pause followed by another burst of activity. For example, when a user goes on a camping trip, he/she would take a larger number of photos than he/she would take on usual

workdays. Based on this characteristic, event boundaries are identified by detecting relatively long pauses in the collection. When a temporal gap between timestamps of ordered photos are significantly bigger than its neighbors, the gap is identified as an event boundary (see Figure 5.6).

5.5.1 Event Hierarchy

In addition to burstiness, I found another interesting pattern in identified events in person photo collections. Photos in personal collection tend to have a temporal hierarchy. In other words, events can be defined in multiple ways with different granularity as in Figure 5.3. For example, “Summer Camping Trip”, which spans June 13th - 17th, can contain multiple subordinate event units such as “Hiking” on 14th, “Canoeing” on 15th, and “Santa Cruz” on 16th. I found that users want to identify each separate event, as well as “Camping Trip” as a whole (Figure 5.3). There are a number of event identification techniques [16][56] which try to find a single level of events. However, as seen in Figure 5.3, a single level event detection technique cannot identify all the meaningful events in photo collections.

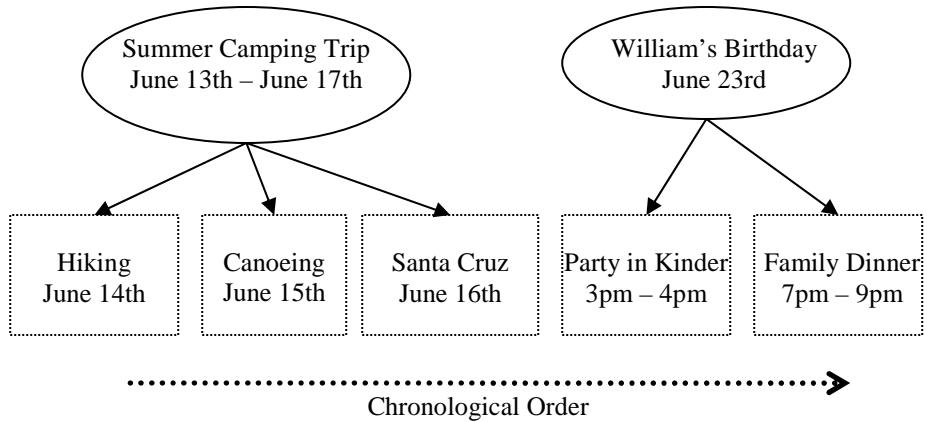


Figure 5.3 An example event hierarchy. The units in the upper row represent coarsely grouped events and the units in the lower row are tightly grouped events.

Hierarchical event identification enables more flexible grouping. By changing the granularity of event grouping, users are provided with coarsely grouped events as well as tightly grouped events according to users' grouping flavors.

5.5.2 Update Event Boundaries

Suppose that a user is about to add a number of photographs into his/her photo collection. Then, the system needs to identify how these new images would fit in the pre-identified events. In addition, users might want to redefine event boundaries that have been automatically identified. This subsection details updating event boundaries.

Temporal information has implicit semantics of sequence. In other words, any given moment in time can be located in a timeline and compared with other points in time. This aspect implies that it is possible to pick neighbor events in an event hierarchy as

in Figure 5.4. For example, as neighbors of the event “May 2nd - 3rd”, we can easily identify two neighboring events, “April” and “May 12th-16th”.

The neighboring events are important because they have a high probability of sharing common information with a given event. This feature is particularly useful because it can be utilized to fix errors in event boundaries which have been automatically identified.

When users find automatically identified events inappropriate, it might be because either 1) that images in the cluster should have been included in one of the neighboring groups or 2) the automatic algorithm creates event groups with unsuitable granularity, which usually causes too coarse or too tight events.

Figure 5.4 explains an example how to locate an event (a group of photos) into an event hierarchy. As in Figure 5.4, suppose that an event May 2nd - 3rd is identified automatically. When users find that the automatically identified event is consistent with users’ intention, users don’t have to do any additional manipulation. A new temporal event “May 2nd - 3rd” is created and added into the event hierarchy as in the case *iii)* in Figure 5.4.

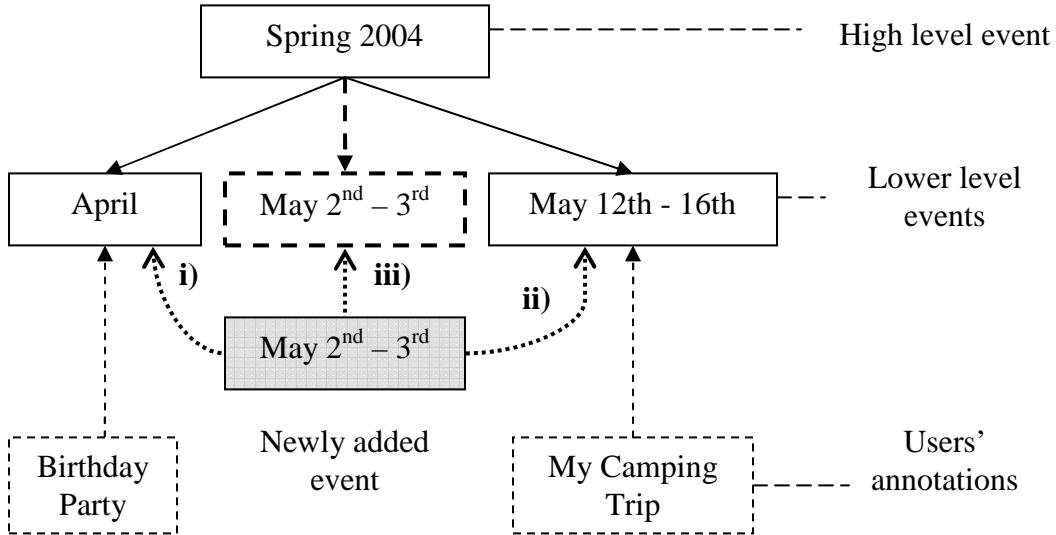


Figure 5.4 An example event hierarchy. When a node “May 2nd~3rd” is to be added into the hierarchy, it can be either i) merged into the previous period, ii) merged into the next period, or iii) separated as an independent node.

However, users may well find that the identification of the event May 2nd - 3rd as an independent event is inaccurate. In this case, users can intervene and fix the inaccurately identified event. As stated above, neighboring events have a high probability to share common information with the given event. In this example, the event “May 2nd - 3rd” has a decent chance to share information with its neighbors, “April” and “May 12th-16th”. In other words, the event “May 2nd - 3rd” can be merged into one of its neighbors. In Figure 5.4, the photos from event May 2nd - 3rd can be merged into either “Birthday Party” event (denoted as *i*) in Figure 5.4) or “My camping Trip” event (denoted as case *ii*) in Figure 5.4) according to the user’s discretion.

In some cases, users may find that grouping photos from May 2nd - 3rd as a single event is inappropriate because the photos can be separated further into multiple sub-events. With these cases, users can 4) split the event “May 2nd - 3rd” into finer sub-events.

In some cases, users may find that the grouping of photos from May 2nd - 3rd is too broad and is needed to be separated into multiple sub-events. With these cases, users can split the event “May 2nd - 3rd” into finer sub-events and the sub-events are added into an event hierarchy as independent events.

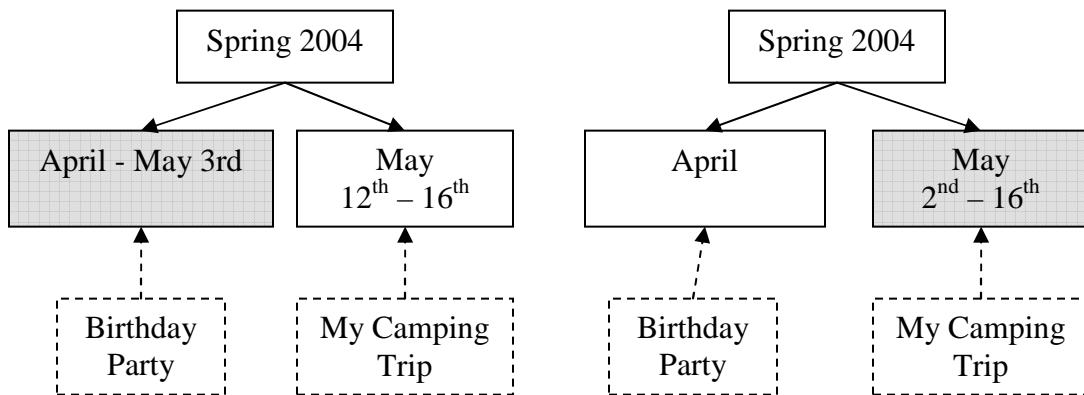


Figure 5.5 The example event hierarchy shown in Figure 5.4 is changed after being updated by a user. The left example shows the result after merging the “May 2nd - 3rd” event into the previous group, “Birthday Party” (denoted as case *i*) in Figure 5.4). In the right, photos of the “May 2nd - 3rd” event are merged into the next group, “My Camping Trip” (denoted as case *ii*) in Figure 5.4).

To summarize, there are four major choices that users can make for automatically identified events. The newly identified event can be 1) merged into the previous event or 2) the next event. If neither makes sense, the photos in the events may be totally

independent from the surrounding events; and 3) is added to an event hierarchy as a self-governing event. When a user finds that the event is too broad, the events are 4) divided into sub-events according to the user's discretion.

While updating event boundaries, user interfaces for this type of tasks have a crucial requirement. Users have to see the photos in the previous and the next event as well as images in the current event to determine the validity of event grouping. Without understanding characteristics of neighboring events, it would not be easy for users to decide what to do with the current group. In SAPHARI, I provide context information, photos in neighbor events, by zoomable interface techniques. When zoomed out, SAPHARI provides a natural overview of adjacent event groups as shown in Figure 5.2.

5.5.3 Event Identification Algorithm

As explained earlier, I assume that events are separated by a relatively long temporal pause. Some researchers have used this burstiness character of photo collections to detect event information inside them. Cooper *et al.* [16] present similarity-based method to cluster digital photographs by time and image content. Platt *et al.* [56] develop an adaptive local threshold applied to the inter-photo time intervals. Loui *et al.* [47] use K-means algorithm combined with content-based post-processing.

In SAPHARI, I develop an algorithm based on Platt *et al.* [56]. While Platt's algorithm focuses on detecting event boundaries on static collections, I improve the algorithm so that it can be used to support hierarchical event structure and dynamic update.

The basic idea behind [56] is to compare a time interval to its local average interval. Suppose that timestamps of photographs are ordered as $[t_1 \dots t_n]$, then a list of time intervals $[g_2 \dots g_n]$ can be easily computed where g_i , is defined by $t_i - t_{i-1}$. Then, for each time interval g_i , the algorithm looks up adjacent time intervals $[g_{i-d} \dots g_{i+d}]$, where the parameter d controls the size of neighbors to be considered. If the current gap is considerably larger than its weighted local average, the algorithm decides the gap to be an event boundary. Platt *et al.* formulate the idea as follows.

$$\log(t_i - t_{i-1}) \geq K + \frac{1}{2d+1} \sum_{j=-d}^d \log(t_{i+j} - t_{i+j-1}) \quad [56],$$

where t_i is a timestamp from an ordered list of photographs, K is a threshold for sensitivity, and d is a windows size. While this formula can detect event boundaries, it has some drawbacks if used as is. Cooper *et al.* [16] reports that the accuracy of this algorithm was not quite as good as other clustering algorithm. One of the reasons for its inaccuracy can be attributed to the fact that the algorithm requires empirical parameters, K and d , which are subjective. Users might need to spend some time to decide an adequate K and d values for their photo collections. Another problem is that the algorithm does not consider users' feedback on event boundaries. Once the event boundaries are set, it is not possible to update them. Furthermore, hierarchies in events are not supported.

Based on Platt's algorithm, I developed an interactive and adaptive algorithm (Figure 5.6) that can support multiple event levels. My algorithm allows users to put their updates inside the clustering algorithm as well as to insert extra photographs any time without breaking pre-existing event boundaries. By changing the K and d parameters,

event detection granularity can be controlled. In SAPHARI, I use empirically chosen $K = \{25, 200\}$ and $d = \{10, 20\}$.

```

BUILD_HIERARCHICAL_EVENT_CLUSTER(images in collection, current event level)

foreach image in images

    if(image.eventBoundary[finer granularities].merge is true) {

        // Case 1

        image.eventboundary[current event level].merge = true;

    } else if(image.eventBoundary[coarser granularities].split is true) {

        // Case 2

        image.eventboundary[current event level].split = true;

    } else if(image.eventboundary[current event level] is not defined) {

        // Case 3

        image.eventboundary[current event level].split =

            SPLIT_BEFORE(images, index, current event level)

    } else {

        // Case 4

        // keep the current image.eventboundary[current level]

    }

end foreach

END

BOOL SPLIT_BEFORE(images, i as current index, l as current event level)

T[] = ordered timestamps collected from images

KL = K[l]

dL = d[l]

If log(T[i] - T[i-1]) ≥ KL +  $\frac{1}{2d_L + 1} \sum_{j=-d_L}^{d_L} \log(T[i+j] - T[i+j-1])$ 

    return true

Else

    Return false

END

```

Figure 5.6 Pseudo code for building hierarchical event clusters

Since I assume an event hierarchy in personal photo collection, the logical structure among events should also be maintained. In other words, an event hierarchy should be kept as tree-like structure, where its root event represents the whole collection. For example, event groups in the same level cannot overlap and a photo cannot be included in multiple event coarse groups. However, as users change the original event grouping that have been automatically identified, the change may affect events in other levels in hierarchy. Figure 5.7 shows two examples which explain possible problems when merging two adjacent events.

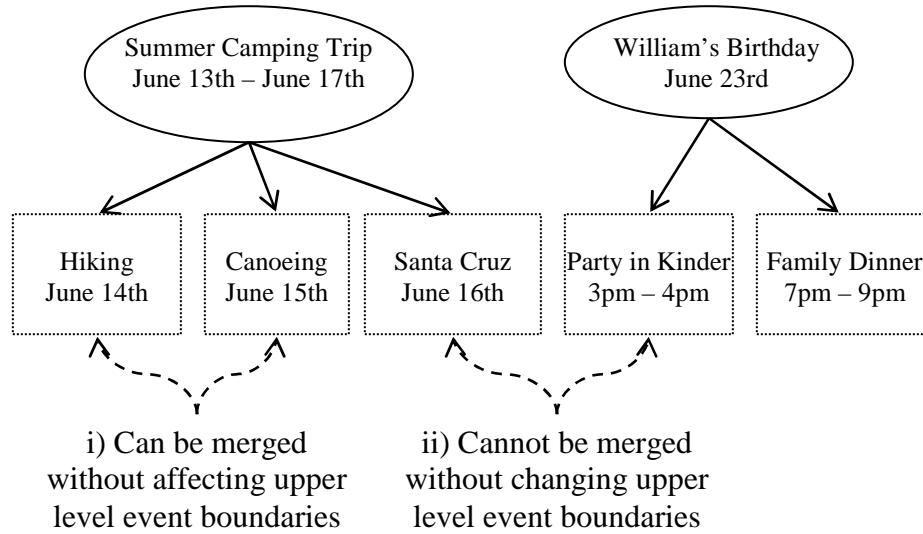


Figure 5.7 Merging two adjacent events

In order to keep the logical integrity of event hierarchy, two conditions should be kept. They are: 1) when events are merged at a finer level, the event groups cannot be split at coarser levels; 2) When events are split at a coarser level, those events cannot be merged in finer levels. Keeping rules ensures the validity of the structure of event hierarchies. More importantly, these rules can be used to propagate users' feedback

into other levels in the hierarchy. For example, when users split an event into two events at a coarse level, the change is automatically applied to every finer level. If a user merges two adjacent events at a finer level, the update may merge events at a coarse level (denoted as case *ii*) in Figure 5.7).

In SAPHARI, users are allowed to choose different levels of events to make annotations. According to users' preferred event granularity, they can select a level in the event hierarchy and make annotations. While users browse their photo collection, they also can fix event boundaries, which are automatically propagated into event grouping of the different levels in the event hierarchy. Hierarchical event identification enables a more flexible way to annotate photos compared to fixed event clustering techniques such as [16][56].



Figure 5.8 The upper shows a result from event identification with a coarse granularity where all images from one day are identified as a single event. The bottom shows event grouping with a finer granularity. Different levels of events

can be obtained by changing the identification granularity. The K and d values on the right side are constants used to detect clusters, which is used in Figure 5.6.

5.5.4 Annotation Strategy

As explained earlier in the semi-automatic annotation interface design guidelines, bulk annotation is a valuable accelerator for creating metadata. SAPHARI is designed to help users make bulk-annotations efficiently. When users like to annotation event information on photos, SAPHARI arranges photos by event groups on the screen so that users can make annotations on event groups not on a single photo repeatedly. While SAPHARI allows users to make bulk annotation by drag-and-dropping a metadata label on a photo group, users always have the freedom to annotate a single photo at any time.

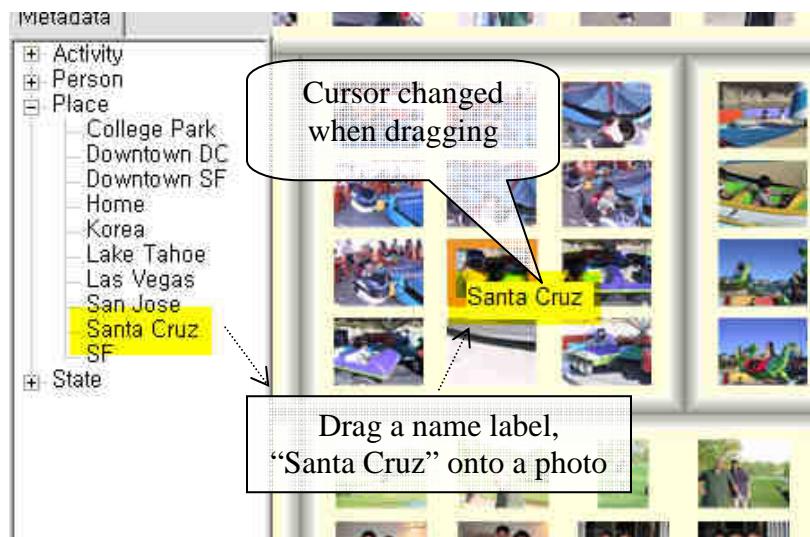


Figure 5.9 Annotation by drag-and-drop. Users can drag a text label onto a photo or a group of photos to make annotations.

Users can drag a label onto a photo or a group of photos to make annotations. Figure 5.9 shows an example annotation, adding “Santa Cruz” on a single photo. Users can

begin dragging by selecting an entry in the metadata tab on the left of the interface. With the mouse dragged, the cursor is changed into the text label. At the same time, the drop target is highlighted to give visual feedback to users. In Figure 5.9, the borders of a photo under the mouse cursor are highlighted with orange color. As with the standard drag-and-drop metaphor, the metadata is annotated onto a highlighted photo as the mouse button is released.

With the shift key pressed, SAPHARI chooses all the photos in the event group under the mouse cursor as its drop target instead of a single photo. When a label is dropped on a group, photos in that group are annotated with the dragging label at once (Bulk annotation). In addition, SAPHARI supports bulk annotation on any arbitrary pre-selected group of photos as well as on a single photos and event groups.

While SAPHARI supports bulk annotation on event groups, not all events are appropriately grouped with the granularity that users want. According to users' taste, they may want to have finer or coarser event granularities. SAPHARI supports two levels of event grouping as explained in the previous section. While making annotations, users can change the event granularity and SAPHARI immediately changes its grouping and show an alternative event grouping. For example, when users find that the "*Fine*" grouping has too much detail, they can switch the grouping to the "*Regular*" instantly and make annotations on coarsely grouped events.

However, there are cases when the automatic event identification fails to detect events correctly. Since the event detection is automatically calculated, it does not always match users' intention [16]. SAPHARI allows users to manually override any

event boundary that has been automatically identified. As shown in the previous section, there are three types of modification when changing the boundaries of an event. They are: 1) merging the current event with the previous event, 2) merging the current event group with the next event, and 3) splitting the current group. As shown in Figure 5.10, SAPHARI provides a context menu for these types of modifications. When event boundaries are updated, the changes are propagated into event boundaries of other levels. For example, when a coarse event group is splitted, the split point is propagated to finer event groups. When two adjacent fine event groups are merged, the merge is propagated to coarser event groups.

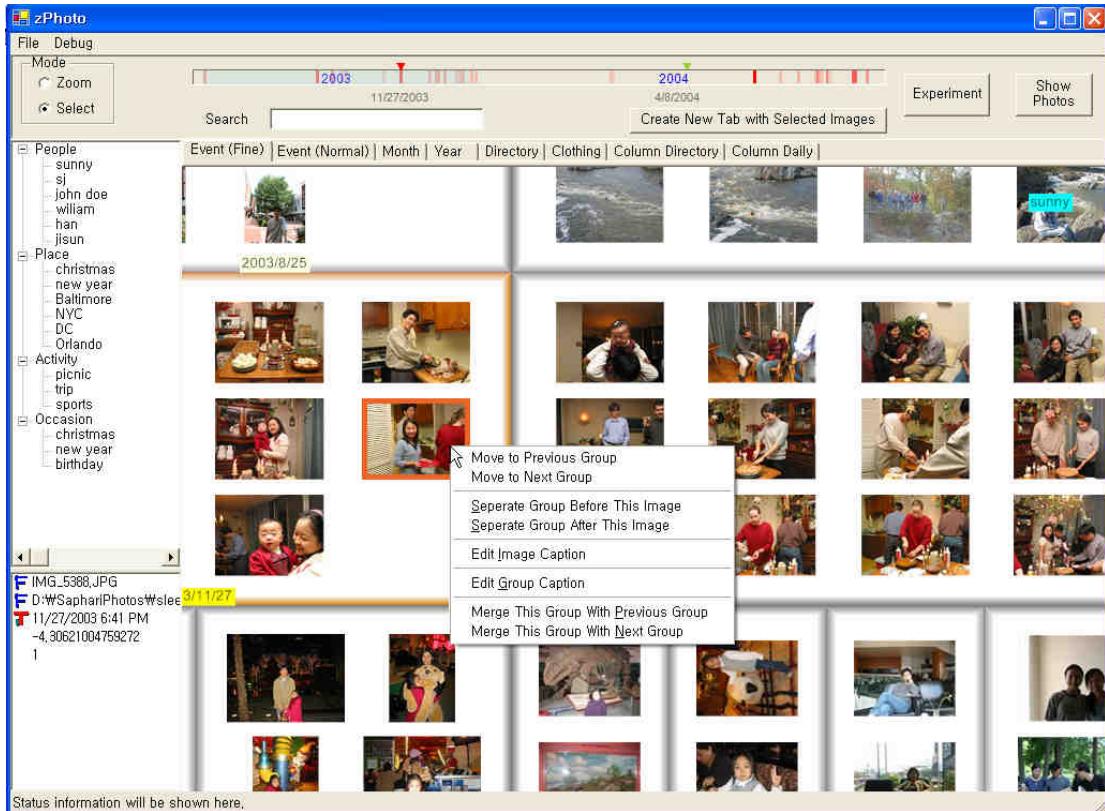


Figure 5.10 Fixing event boundaries which have been automatically identified.

5.6 Clothing Based Human Recognition

People in photos are regarded as one of the most important information in photos because many photographs include people as central objects. It is not surprising that many image browsing prototypes [2][39][62] focus on labeling people with metadata such as names. SAPHARI allows users to make bulk annotations on people in photos.

5.6.1 Face Recognition for Personal Photos

Faces are the most crucial information for identifying people. There has been much research recently about the use of facial features to recognize people in images [76]. Roughly, there are two approaches to the application of face recognizers. First, face recognizers can provide a *similarity* metric between faces. In this approach, the metric can be used to cluster faces, which is important for bulk annotation because similar faces can be grouped together. However, additional steps are required to label clustered faces.

On the other hand, *labeling* of faces, which are provided by face recognizers, can be directly used. In this case, the face recognition software must be trained with a learning set of photos. Users have to provide initial mappings between faces and labels so that face recognizers can suggest labeling for unseen faces in the future. With this strategy, the result of users' manual annotation can be used as training examples.

Since SAPHARI is designed to facilitate bulk annotation, clustering faces is important. Grouping similar faces would help users to select multiple annotation

targets effectively. Furthermore, labeling of faces requires intensive training on early stages of interaction. For example, some faces have to be manually annotated in the beginning. For these reasons, I focused on face recognition techniques which provide similarity metrics.

However, research about recognizing human faces have had limited success and face recognition in an uncontrolled environment is still very challenging. For example, even for the best face recognition systems, the recognition rate for faces captured outdoors, at a false rate of 1%, was only about 50% [53]. Also, many state-of-the-art face recognition systems are commercial products and not available for public use [53].

As preliminary research, I used the HMM face recognizer included in OpenCV [52]. Even though the face recognizer produced reasonable results when applied for controlled face sets – indoor, controlled lighting, and frontal view, the accuracy was dramatically decreased when used on personal photos. I found the accuracy to be less than 10% on my personal photos, which was unacceptable. The face recognizer was very sensitive about lighting condition and tilted faces, which are not unusual cases for personal photos. People in personal photo collections frequently are not gazing at the camera, which aggravates the hardship in face recognition. Some faces might be turned away, averted, or even occluded. Therefore, I concluded that I cannot solely rely on human face recognition to identify people in photos.

As an alternative, I have observed an interesting pattern in person photo collection that can help with identifying people. People usually don't change their clothing

during a day. Given this condition, clothing information can be used to assist the identification of people. I hypothesize that people who wear similar *clothing* and appear in photos taken in one day are very likely to be, in fact, the same person. Furthermore, the episodic aspect of personal photo collections facilitates the assumption. As stated earlier, many photos are often taken within one day. Based on these two assumptions, we can use information about the clothing a person is wearing to identify people in personal photo collection.

5.6.2 Human Model

In this section, I present a human model based on clothing information. For modeling clothing in photos, it is first necessary to locate the clothing of people in photos. However, it is not an easy task because the shape of human body is not rigid. Human can move their body parts such as arms and legs rather freely and the shape of clothing is quite variable.

While many researchers have focused on detecting human body movement, it is still challenging to detect human bodies from a single static scene [3][59]. Rather than trying to detect the human body directly, I use a face detection technique to locate clothing in photographs. While it is not useful to do face recognition analysis on personal photos because of low accuracy, we can use face detection technique, where its goal is to locate faces in images [73]. Some systems have reached around 90% accuracy for detecting faces in images. I develop a clothing-based human model as in Figure 5.11 by a using face detection technique. I use the Viola-Jones face detector [46][52][69] to locate faces in a photo.

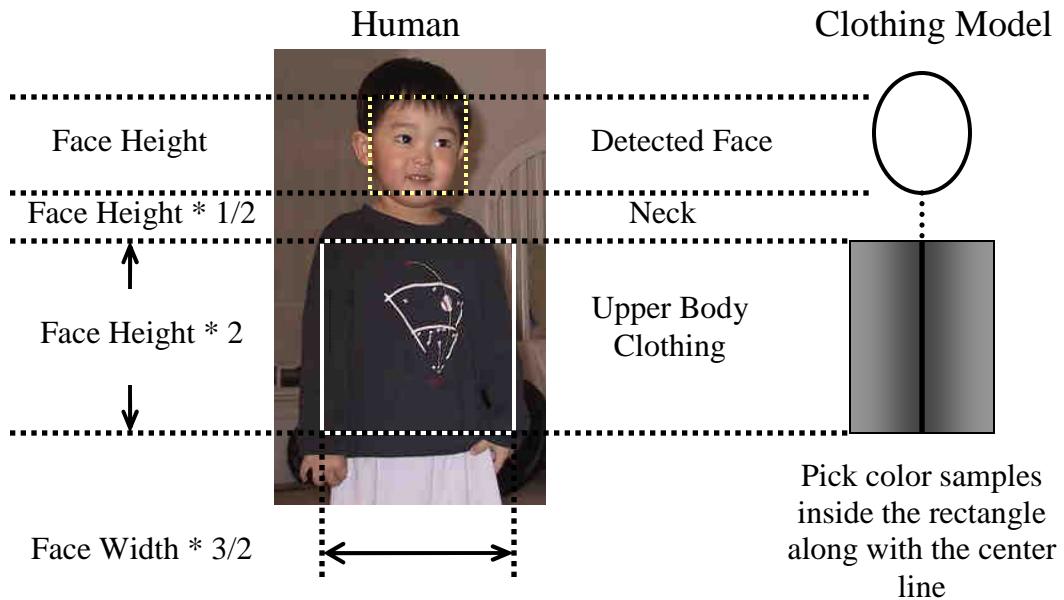


Figure 5.11 Locating clothing from detected faces

I use a clothing area defined as a rectangular region under the face as in Figure 5.11. Since the face detector also provides sizes of faces, the size of the upper body clothing region is calculated based on the size of the face. As shown in Figure 5.11, I construct the human model only with upper body clothing. Theoretically, it would be optimal to use the whole body information. However, as explained earlier, identifying a human body causes a whole set of problems and is beyond the scope of this dissertation. I also find that the upper body clothing is, sometimes, more useful than whole body information. The whole body information does not exist in photos such as in portraits, people sitting in front of a desk. As a quick alternative, I use the upper body part alone based on the heuristics that an upper body is tightly coupled with a face.

Once a face is detected, I skip some area underneath the face as I assume the area is belongs to a neck. Then, along with the center line of the face, I pick a rectangular region under the neck as clothing. However, my clothing model does not necessarily assume clothing as a rectangle. The rectangle simply represents bounds inside where color samples are collected. Inside the rectangle, samples are picked based on the probability distribution as in Figure 5.12.

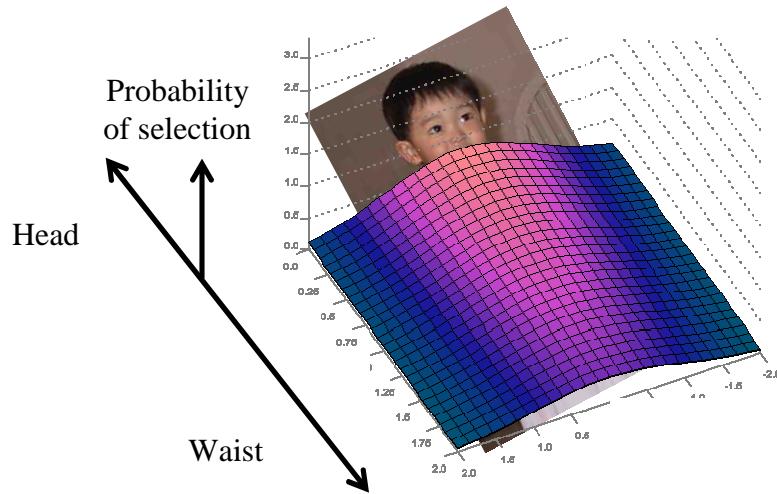


Figure 5.12 More weight is given to the upper center part of clothing.

Figure 5.12 shows the probability distribution of selecting samples inside the clothing. Each horizontal row follows a normal distribution of which mean is on the center line of the face and of which standard deviation is $3/8$ of the detected face width. As shown in the figure, the upper parts have more weight. The weight of the topmost row has twice as much as that of the bottom-most row. Based of this probability, I pick a number of samples and turn them into a human model.

In this paper, four dimensional feature vector $X = (y\text{-distance}, \text{red}, \text{green}, \text{blue})$ is employed to model the clothing, where $y\text{-distance}$ is defined as a relative vertical

position of a sample and *red*, *green*, and *blue* are color information, respectively. In SAPHARI, about 900 samples are picked in the upper body region. With this four dimensional vectors, I estimate a four dimensional probability density function per clothing of a person by the following kernel density estimation formula. [65]

$$\hat{f}(X) = \frac{1}{n} \sum_{i=1}^n \frac{1}{(2\pi)^{d/2} |H_i|^{d/2}} * \exp\left(-\frac{1}{2}(X - X_i)^T H_i^{-1} (X - X_i)\right) \quad [65]$$

, where n is a number of samples and d is 4, respectively. H is a diagonal matrix with independent four variances as follows.

$$H = \begin{bmatrix} \sigma_{y-distance} & 0 & 0 & 0 \\ 0 & \sigma_{red} & 0 & 0 \\ 0 & 0 & \sigma_{green} & 0 \\ 0 & 0 & 0 & \sigma_{blue} \end{bmatrix}$$

As shown in H matrix, I assume that there is no correlation between *y-distance* and *red*, *green*, and *blue*. For my prototype, I use $\sigma_{y-distance} = 0.08$ (of the clothing height), $\sigma_{red} = 0.04$, $\sigma_{green} = 0.04$, and $\sigma_{blue} = 0.04$, respectively.

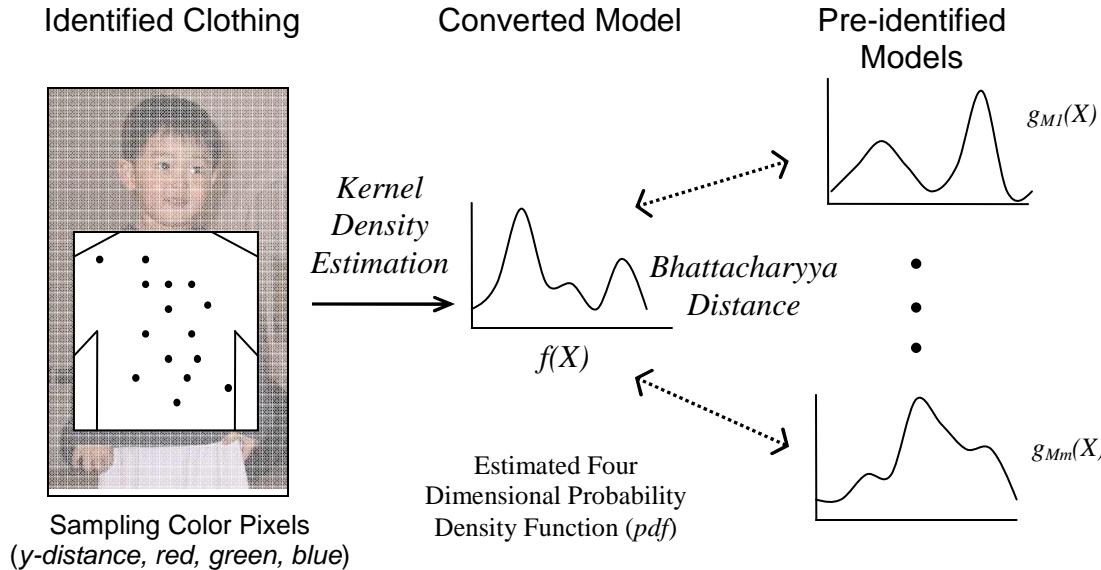


Figure 5.13 Human model based on clothing

The basic idea of the clothing based human recognition is to use the estimated four dimensional probability density function (*pdf*) as a proxy of a person. After mapping clothing into an estimated four dimensional probability density function (*pdf*), we can measure the visual distance between two pieces of clothing. I use *Bhattacharyya* distance to measure the distance between two *pdfs*.

The *Bhattacharyya* distance [11] is defined as follows.

$$\text{Bhattacharyya Distance} = \sqrt{1 - \sum_{i=1}^n \sqrt{p(X_i)q(X_i)}}, \text{ where } n \text{ is a number of}$$

samples picked for comparison and p, q are *probability density functions* (*pdf*) which have been estimated by using [65]. As shown in Figure 5.13, the visual distance between two human models can be measured by using the *Bhattacharyya* distance.

With the distances between human models, SAPHARI classifies people in photos. When a measured distance is below an empirically determined threshold, the system

classifies the models as the same one. When a distance between two models is above a threshold, SAPHARI catalogs them as two different ones. SAPHARI uses an empirically chosen threshold, 0.4.

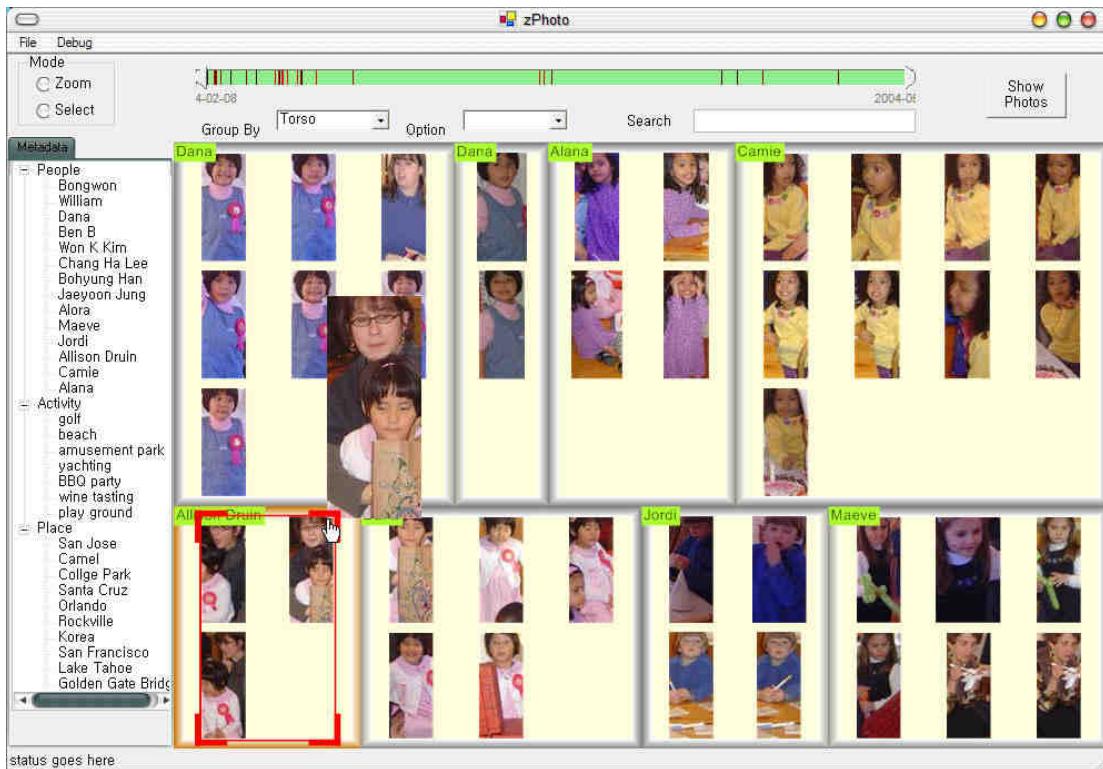


Figure 5.14 People in photos are cropped and laid out on the screen grouped by their clothing similarities. People who wear similar clothing are clustered together.

As explained earlier, the clothing based analysis is performed on photos taken in one day. When users choose the “Clothing” tab in SAPHARI, users are asked to pick a date among the list of dates when users have taken any photos.

For each photo taken in a day, SAPHARI identifies locations of faces and crops out the faces with associated upper bodies. Then, SAPHARI clusters the torso image (portrait of face and upper body) based on clothing as shown in Figure 5.14. When

there are multiple faces in one photo, SAPHARI prepares a torso image per identified face. Every cropped face is normalized (shrunken or enlarged) to the equal size on the screen as shown in Figure 5.14.

5.6.3. Annotation Strategy

SAPHARI lays out cropped faces on the screen based on visual features of the clothing. With the face clustering, users can make bulk annotations on a group of people by dragging a name label. Rather than annotating photos individually, annotation is allowed only on a face group. Upon dropping a name label, all faces in the group under the cursor are annotated with the name (Figure 5.15). After a name is assigned to a face group, the group itself is associated with the name.

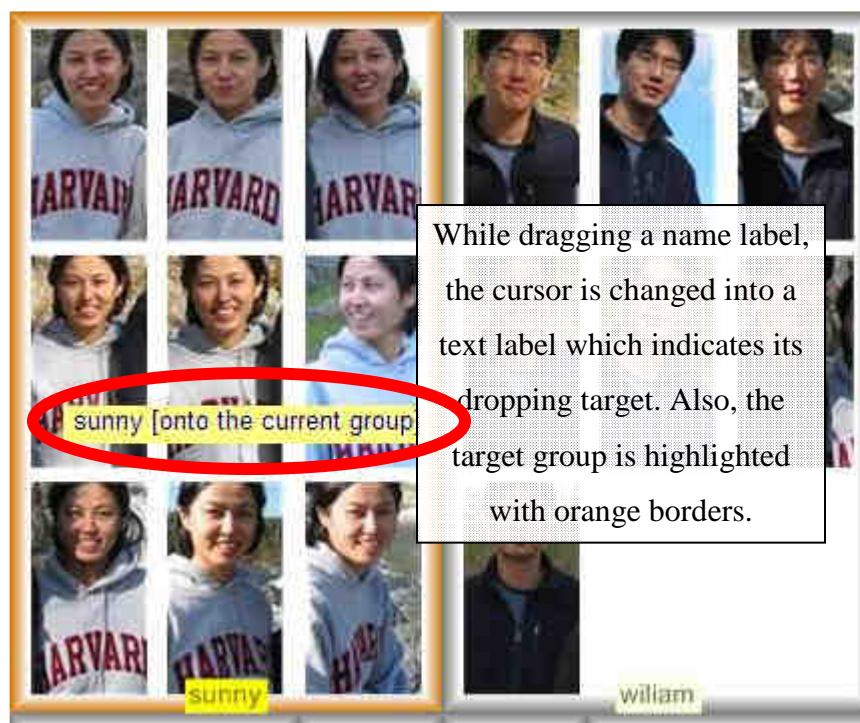


Figure 5.15 Make bulk annotations by drag-and-dropping a name label on a face group.

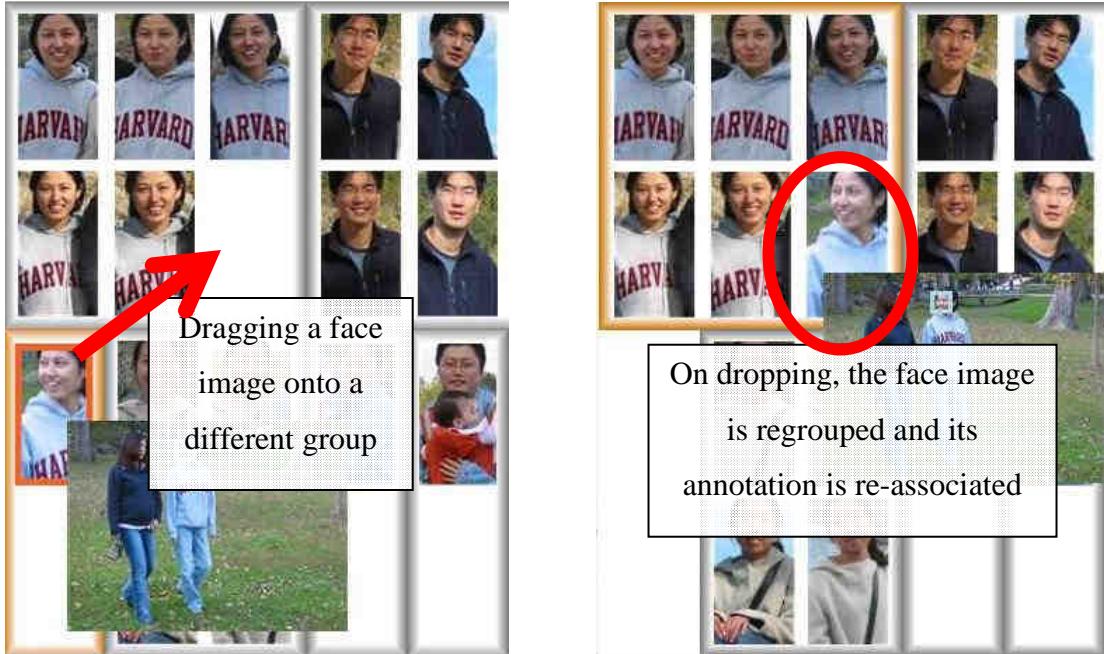


Figure 5.16 Fix a misclassified face image. Moving a face image into a different face group updates the association between the face image and the name of the person.

However, clothing based human recognition also generates recognition errors due to many reasons. The errors can be easily corrected by relocating face images into the correct person group. As shown in Figure 5.16, users are allowed to move a face image or a group of face images into another group. Once face images are moved into other groups, the face images don't maintain face annotations which have been made on it earlier. Instead, the moved face images are automatically annotated with the new name label of the target group. For example, suppose that a face, F is misclassified in a face group G . As a user relocates the image F into a group H , F is automatically annotated with the name label of H . This concept, where a group is associated with a set of metadata, is introduced by Kang [38] and called as *Semantic regions*. Face groups in SAPHARI are *semantic groups* and can be annotated with a name. Adding

face images into a face group will make annotations on them with the associated name.

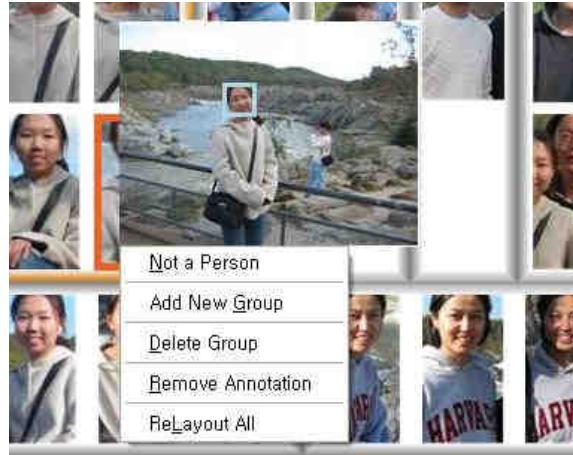


Figure 5.17 Context menu for the clothing (face) group layout.

SAPHARI also provides other utility functions for easy manual regrouping. Figure 5.17 shows a context menu that SAPHARI supports. Users can create a new face group, remove a face group, remove annotation, and remove an unnecessary face image (“*Not a Person*” menu item). Due to the errors in face detection, sometimes, non-face images are recognized as a face. Clicking “*Not a Person*” removes the image from the face group. By using these functions, along with drag and drop techniques, users can relocate misclassified faces into where they belong.

5.7 Semi-automatic Annotation User Study

I conducted an user study to examine the effect of semi-automatic annotation strategies on personal photo collection, and to observe the strategies users employed. The user study was divided into two parts. First, I observed and measured how event-based clustering effects users’ annotation. Participants were asked to examine

automatically identified event groups laid out in a 2D zoomable space and to annotate some given key events (Event Task). Second, I compared the clothing based human recognition and manual annotation. Participants were asked to identify people in a set of photos and to annotate them with appropriate name labels (Face Task).

While I measured the task completion time for comparison, the user study was not a controlled user study. A controlled experiment requires that the condition of each task should be identical to each other, which is not true in this user study. Rather than using a fixed photo collection, I used users' own photo collection. Since the user study focuses on personal photo collection, using non-personal photos is not the intended target. In practice, it was also hard to recruit participants who share common experience. Furthermore, some contents of photos in personal collections were private. It was not practical to design controlled user studies with limited time and resources.

The study results showed some interesting patterns that provide valuable insight about semi-automatic annotation techniques. In addition, I was able to observe various behaviors from users while they were using SAPHARI. I will explain the details in the following sections.

5.7.1 Participants

There were seven participants in this study. Participants were college or graduate students at the University of Maryland. There were four men and three women. All participants were familiar with computers. The summary of their photo collections are list as shown in Table 5.2.

Participant	Digital photo experience	Estimated total size of the collection	Size of the collection provided for the study
P1	Three years	600	579
P2	Three years	2000	1245
P3	Four years	2000	1664
P4	Three years	1000	727
P5	2.5 Years	1000	464
P6	Three years	3000	1309
P7	Two years	1000	758

Table 5.2 Participants Information

Each participant was asked to provide more than five hundred photographs which had been taken over more than a six month period. It was not easy to recruit participants who were willing to provide their personal photos. Some of them were very concerned about their privacy especially because some of the photos had sensitive private contents.

5.7.2 Method

A few days before meeting with participants, I asked them to fill out a pre-user study questionnaire (see Appendix). From the questions in the questionnaire, I identified events and people that the participants thought important to them. The list of events and people was used in the actual annotation tasks.

On the day of the user study, I began the meeting by explaining the functions of SAPHARI. I gave details about navigation through a zoomable space and clarified the meaning of groups on the screen. With the semi-automatic annotation interface,

participants were reminded that there were two levels of event grouping and they could freely switch their views between them. I provided a couple of sample browsing and annotation tasks to make sure that participants were able to perform intended operations.

Annotation Strategy	Task Type	
	Event Annotation	Face Annotation
Semi-automatic annotation	Photographs are laid on a 2D zoomable space grouped by events that have been automatically identified by the system.	Cropped portraits of people are grouped by the similarity of clothing they wear.
Manual annotation	Photographs are laid out in a scrollbar canvas with grouped by their directory structures.	Photographs are laid out in a scrollbar canvas ordered by the date on which they were taken.

Table 5.3 Four types of tasks were designed to compare the semi-automatic annotation strategy with conventional manual annotation approaches.

Table 5.3 shows the task design. The user study employed a 2x2 design, with annotation technique and task type as independent variables. I measured the time per completion and the number of annotated items as dependent variables. All participants were asked to finish *Event Task* first, followed by *Face Task*. However, the order of annotation techniques was counter-balanced to minimize the learning effect. The total time duration of the user study for each participant was about one hour. I used *talk-aloud* methods to gain more insight about users' behaviors.

5.7.3 Event Task

In the *Event Task*, I asked participants to annotate a set of specific events and measured the task completion time. This task was intended to measure the efficiency and usability of the event based grouping compared to users' own folder-based photo organization. Through the pre-user study questionnaire (see Appendix A2), I found that all participating users were using directories or folders to organize their photographs.

Participants were provided with two different types of interfaces: 1) semi-automatic annotation interface where photos are grouped by automatically identified events and are laid out in a zoomable space and 2) manual annotation interface where photos are grouped by their directory structure and are laid out on a non-zoomable canvas equipped with a vertical scroll bar (Figure 5.18).

I provided each participant with four event annotation tasks, two events for the semi-automatic annotation interface and the other two events for manual annotation interface. In each task, participants were asked to annotate any number of photos that matched the given event. The order of tasks was counterbalanced. Half of the participants finished the semi-automatic annotation tasks first followed by the manual annotation tasks. The other half was asked to begin with the manual annotation tasks. For each task, I recorded the completion time, the number of annotated photos as well as taking memos on the participant's annotation and navigation strategies.

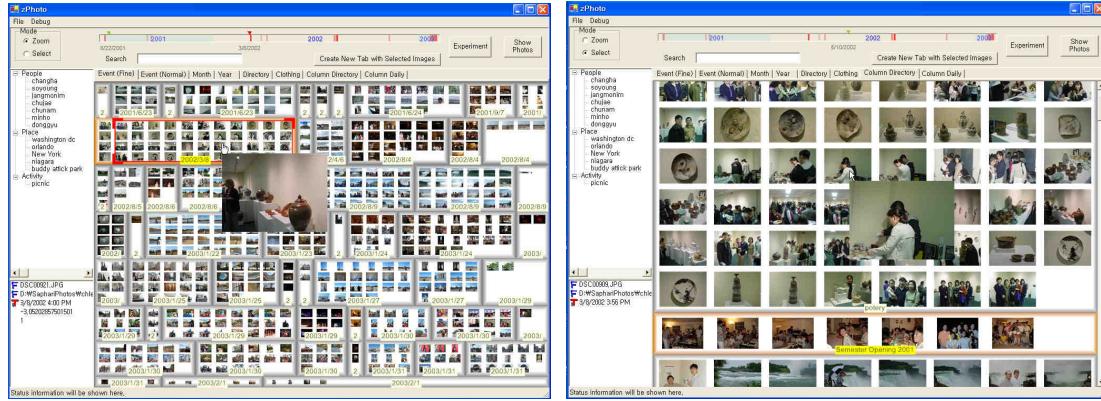


Figure 5.18 Event tasks with two different settings. Left: photographs are grouped by events which have been automatically identified by SAPHARI. Right: photos are laid out by using participants own directory structure.

As briefly mentioned in the previous sub-section, participants were asked to fill in the question, “*Please state at least five interesting events or places in your photo collection.*” I collected a list of events for each collection. The four events, which were used in this task, were randomly chosen from this list. Therefore, each participant was given different events, which makes this user study non-controlled. However, with this study design, the tasks are more consistent with real life situation than annotating unrelated event on non-personal collection.

With the semi-automatic interface, participants were allowed to make annotations on a single photo as well as on an identified group. With the manual annotation interface, participants also were allowed to use conventional selection techniques for choosing annotation targets – 1) clicking with the control key pressed to add the clicked photograph to the current selection group, 2) clicking with the shift key pressed to add a range of photographs to the current selection group, and 3) selecting photographs by a marquee rectangle. Participants can drag a label onto a group of selected photos to annotate them at once.

5.7.4 Face Task

The second part of the user study was designed to measure the efficiency of clothing based annotation. Participants were asked to annotate a given set of photos with a number of people as quickly and accurately as possible. They were provided with two types of interfaces: 1) semi-automatic annotation interface where faces are grouped by clothing based human recognition and 2) manual annotation interface where photos are laid out on a canvas with scroll bars as shown in Figure 5.19.

With the manual annotation interface, participants also were allowed to use conventional selection techniques as in the event task for making bulk annotation.

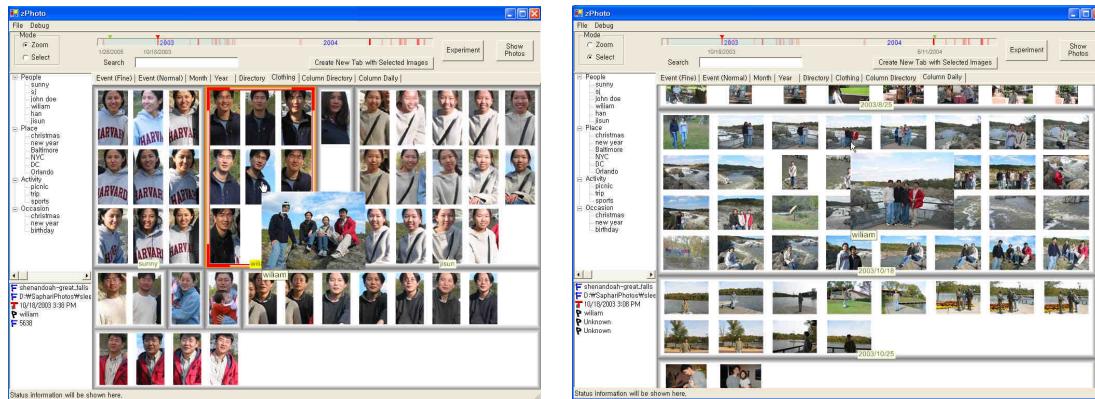


Figure 5.19 Face annotation task interfaces. Participants were asked to annotate people in photos with two different interfaces. Left: Clothing based annotation. Right: Manual Annotation

Each participant was given four different face annotation tasks. The face annotation task was given with the instruction, “*Please annotate the photos taken on [a specific sample date] with [a list of person]*”. The task was easy to understand and realistic

because participants were asked to make annotations on their own photos with people who they are familiar with.

Before meeting with participants for the user study, I manually picked two days per participant. Like event information, the list of people was also extracted from the pre-study questionnaire. The following question was given to participants and their answer was inserted into the metadata field of SAPHARI before meeting for the user study.

“Please state at least five people appear in your photo collection. You don’t have to list all the people. However please include people who are important to you – people who you want to find in your digital photo collection.”

With the two picked dates and the two interface techniques, there are four possible combinations, *date one* with clothing based annotation, *date two* with clothing based annotation, *date one* with manual annotation, and *date two* with manual annotation. Participants were asked to perform each task. The order of tasks was counterbalanced. For each task, I recorded the completion time and the number of annotated faces. I also observed participants’ annotation strategies.

Since the goal of the face task was to examine the feasibility and usefulness of clothing based human recognition, I manually selected the date that were used in the user study. Among the dates on which participants took photos, I picked two dates per participant where 1) at least three people appeared in the photos taken during a day

and 2) at least more than ten photos were taken in a day. For example, I excluded a photo set which were composed of landscape scenes or solo shots.

5.7.5 Event Task Result

Figure 5.20 shows the results from the *Event Task* with two annotation techniques, semi-automatic annotation with manual annotation with directory based grouping vs. automatic event clustering.

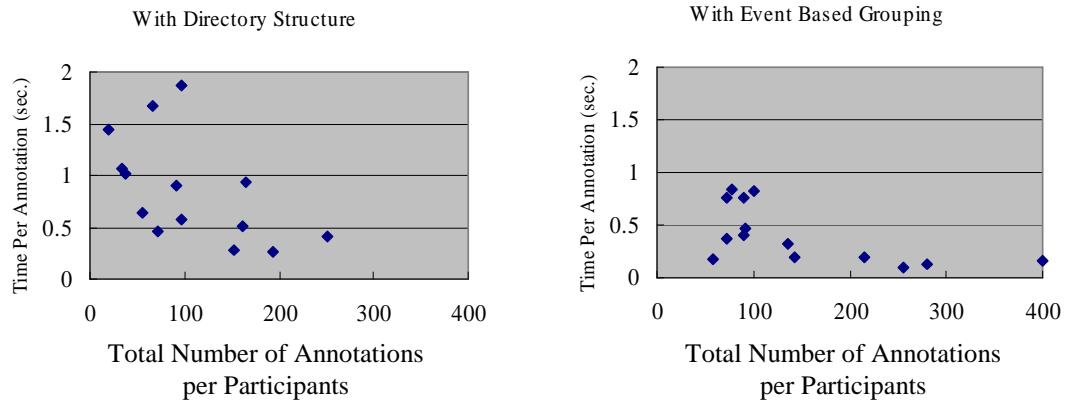


Figure 5.20 The relationships between the time per annotation and the total number of annotations per participant with the two different user interface techniques. Due to bulk annotation, time per annotation has a tendency to decrease as the total number of annotations increases.

As shown in Figure 5.20, the time per annotation was decreasing as the number of annotation made increases. This was mainly due to bulk annotation. When photos are well grouped according to users' events, users could easily select multiples photos and make annotations on the group of photos. In both groupings, automatic event

groups and users' directory structures, participants were able to take advantage of bulk annotation.

However, with the automatic event grouping, users performed much better. As shown in Figure 5.21, the time per annotation was reduced 49% with the event grouping; 0.367 second with semi-automatic annotation (event based grouping) and 0.720 second with manual annotation interface (directory based grouping).

A paired sample *t*-test was conducted on the task completion time and there was a strong statistical difference for the *Event Task* depending on the annotation techniques, $t(1, 6) = 3.16$, $p = 0.019$.

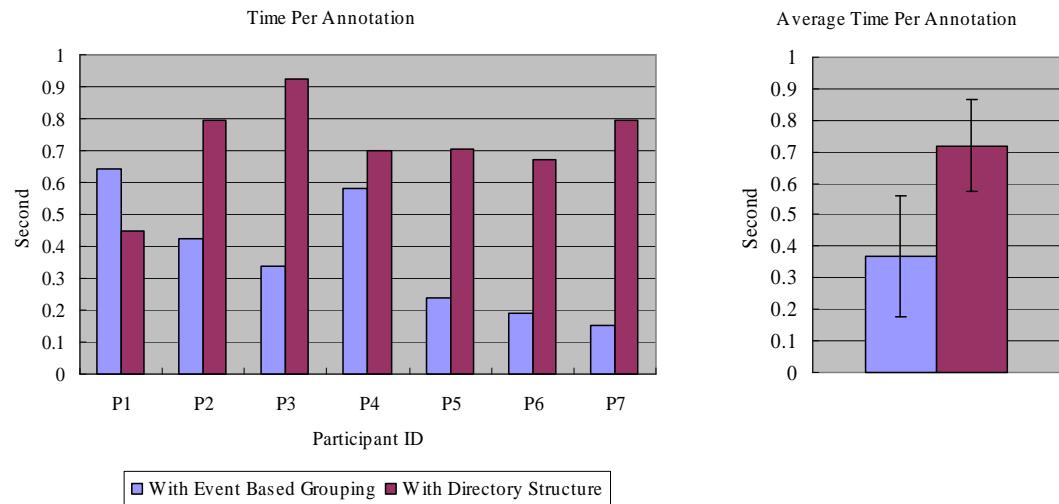


Figure 5.21 Time per annotation results from the event tasks. The left figure shows individual performance of participants and the right figure shows the average and the standard deviation of time spent per annotation.

During the user study, all participants complained about the repetition of selecting photos and drag-and-drops. One participant reported that “*Sometimes, it's very*

difficult to select a group of pictures, especially if they are not adjacent." Especially with the manual annotation, even though participants were trying to make bulk-annotations as much as possible, selecting multiple annotation targets required significant effort from the users. Sometimes, participants had to scroll when selecting annotation targets. On the other hand, with semi-automatic annotation interfaces, participants were allowed to make annotations on pre-clustered event groups and participants took advantage of event groups.

Overall, participants were positive about automatically identified events. They immediately noticed the meaning of each event. One participant said, "*This is Thanksgiving dinner and this is Christmas. And this is when my parents were here.*" He was very satisfied with the automatic event groups and reported, "*This grouping is much better than my directories.*"

SAPHARI provided two event granularities; "*Regular*" and "*Fine*" (see section 5.5). When asked which event grouping was best, six out of seven participants answered that they preferred "*Regular*" grouping. Participants did not care much about detail events. Since they could remember most of events in their photo collections, they preferred to find a high level event and then do a visual search among the photos in that event.

On the other hand, some problems were also observed during the user study. One participant in particular had problems with event groups. Some photos in her collection were altered when rotated and shrunken and the timestamps of the photos were not accurate. EXIF headers of the photos were destroyed and their timestamp

showed the date of the modification, not the actual date of photo-taking. Due to this problem, some events were wrongfully grouped and she was required to unscramble the spoiled event groups. In addition, some participants reported there were a few errors in event boundaries. However, they were able to fix the boundaries very easily.

5.7.6 Face Task Result

The user study with participants was not a controlled experiment and I manually picked the tasks. The goal of the face task was to investigate potential benefit of clothing based human recognition.

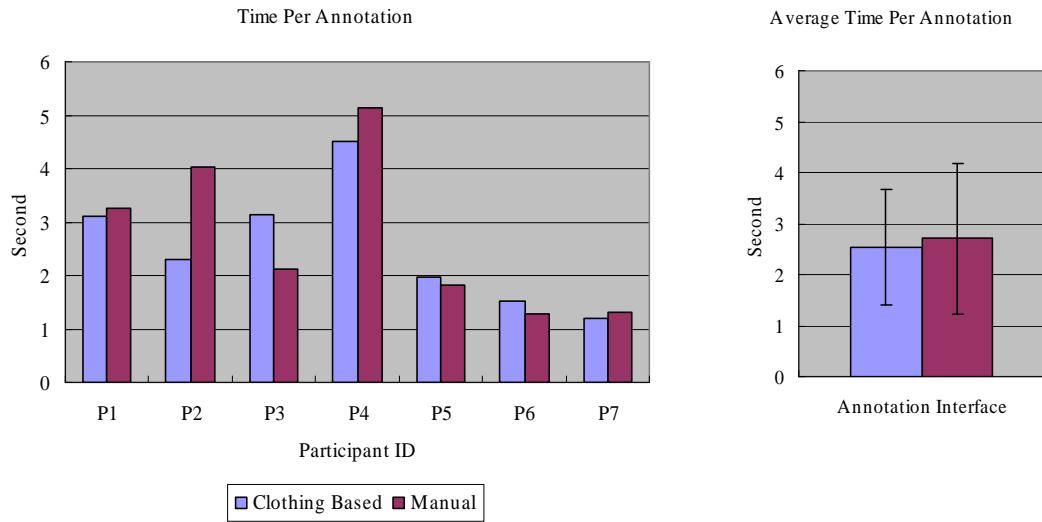


Figure 5.22 Time per annotation results for the face task. The left shows individual performance of participants and the right shows the average and the standard deviation of time spent per annotation.

With the Face Task, there was only 6% difference between two techniques: semi-automatic annotation with clothing based human recognition (2.54 sec per annotation)

vs. manual annotation (2.71 per annotation). As shown in Figure 5.22, the time per annotation did not show significant difference.

However, there emerged very interesting results with the *Face Task*. Even though there were only 6% difference on the time per annotation between manual annotation interface and semi-automatic annotation interface with clothing based human recognition, participants gave very high ratings on “*quick task completion*” with clothing based face grouping (see the next section 5.7.7).

During performing the tasks, one participant reacted that “*It requires too much effort to use face annotation even though it doesn’t seem to take much time.*” Another participant even complained about fatigues on her wrist after finishing the *Face Task*. After the user study, all participants agreed that the face annotation task was too time consuming especially because they had to select target photos one by one. In the *Event Task*, participants were able to take advantage of bulk annotation even with the manual annotation interface because the photos are grouped by the directory. However, with manual annotation on the *Face Task*, it is not easy to make bulk annotation. Photographs that contain a specific person are not necessarily located together on the screen. They were scattered on the screen and the user was required to identify the people in photos one by one. Even though there was no significant difference in the task completion time, participants clearly became more tired with the manual annotation strategy. On the other hand, with the semi-automatic annotation interface, faces were grouped by clothing features. Participants were immediately able to understand the meaning of face groups and made annotations on

the face groups. However, as shown in Figure 5.22, there was no significant speed up with the semi-automatic annotation.

During the user study, I observed a few interesting patterns when participants were making annotations using the clothing based annotation. First, due to inaccurate results from the face detection algorithm, there were a quite number of non-faces that were recognized as faces. It caused SAPHARI to include those non-face images in the face groups. Participants spent some effort to remove those non-faces images from face groups and it slowed down the annotation process. Second, while performing the *Face Task*, participants were also provided with images of unrelated people on the screen. The *Face Task* asked participants to make annotations only with the given list of people. But, in many cases, SAPHARI also provided images of people who were not on the list. For example, one participant was asked to annotate her family members, but the given task also showed a lot of images of her friends. While she was organizing face groups, she also identified all her friends as well as her family members. However, the result only counted annotated family members excluding other identified persons. If all the annotated faces were included in the results, the semi-automatic annotation could have shown additional speed up. In addition, I observed that participants spent more time looking at photos with the semi-automatic annotation than with the manual annotation. With the manual annotation interface, participants immediately began to annotate. But, with the semi-automatic annotation, participants spent some time observing the grouping result before actually performing annotation. When asked what they were doing, participants responded that they were examining the results and planning how to fix errors.

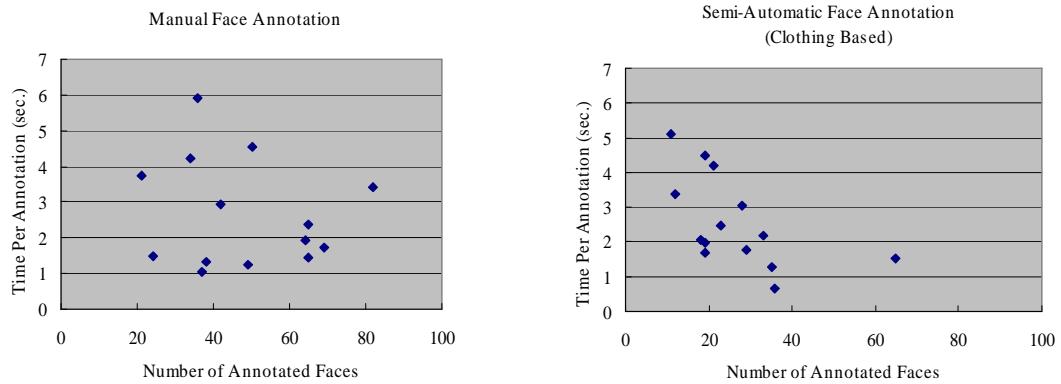


Figure 5.23 Time per annotation with two different user interfaces. While the left scatter plot does not have any noticeable pattern, the right graph shows a clear decreasing pattern as the number of annotated faces increases.

As shown in Figure 5.23, there was an interesting pattern in the results of the two interfaces. With the semi-automatic face annotation interface, the time per annotation decreased as the number of annotated faces increased. This suggests that more bulk annotation was made with the semi-automatic annotation interface. It also implies that, when there are photos to be annotated, the semi-automatic annotate could become more efficient. Even though the statistical evidence is very weak, the pattern shows the positive potential of the semi-automatic annotation interface with clothing based human recognition.

5.7.7 Subjective Satisfaction

In the post-user study questionnaire, much stronger differences emerged. Immediately after the Face Task, participants answered questions about their satisfaction with the interfaces they used in the study (see Appendix A3). Figure 5.24 shows the average and standard deviation scores on a seven point scale (1=disagree, 7=agree) for

participants' responses to a number of ease of use and preference ratings. A one-way analysis of variance (ANOVA) was run on each measure to test for differences between interface techniques.

As shown in Figure 5.24, there was no difference in the first two questions; “*simple to use*” and “*easy to learn*”. Participants found that annotation interfaces easy to use. Participants immediately grasped the main concepts of annotation and event groups. They were also easily acquainted with navigation with zooming in and out. All participants were able to finish the given task successfully without any problem with two interface techniques and they answered very positively for both interfaces when being asked about the ease of use and learnability.

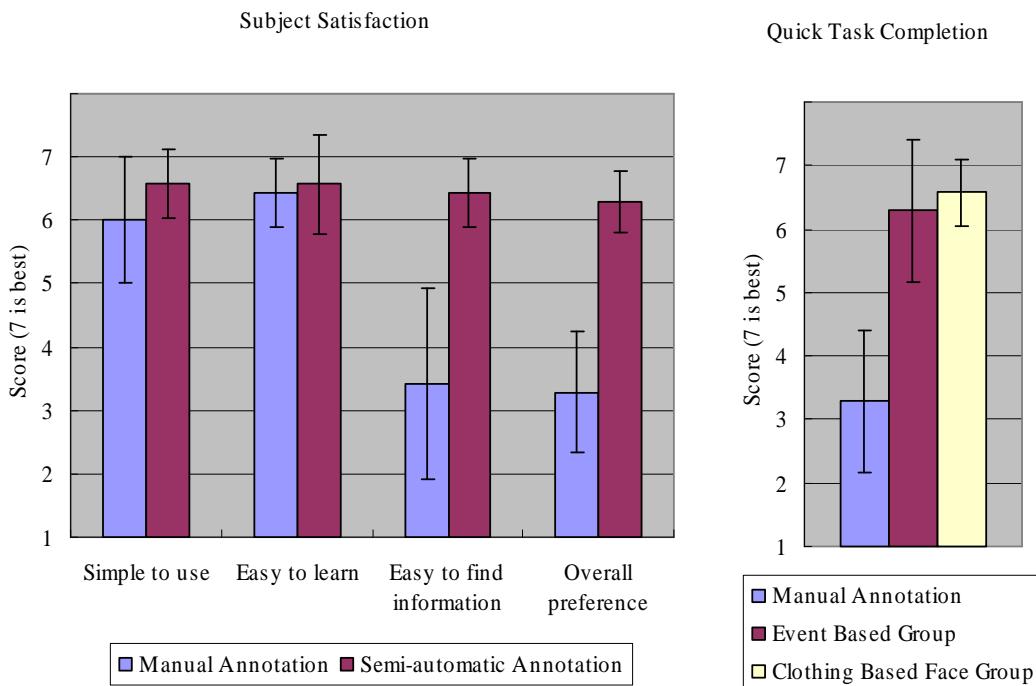


Figure 5.24 The result of participants' subjective satisfaction which was measured by the post-user study questionnaire.

However, the answer to the question, “*easy to find information*” showed clear preference. Participants were significantly positive with semi-automatic annotation interfaces compared to manual annotation interfaces, $t(12) = 6.74, p < 0.001$. Since the semi-automatic interfaces provided photos in a 2D zoomable space, participants were able to easily zoom in any photos. In addition, the zoomable interface provided quick previewing of photos. With mouse hovering, the interface provided a preview (about 200x150 in pixels) of photos under the cursor. Most participants were able to take advantage of zoomable user interface to navigate their photo collections. One participant mentioned that the previewing was “*definitely useful*”. On the other hand, with the manual interfaces, participants were asked to use scrollbars.

For the “*overall satisfaction*” question, participants unanimously preferred semi-automatic annotation interface with very strong statistical significance, $t(12) = 7.42, p < 0.001$. Response was very positive. A few participants were even interested in continuing to use it in his personal computer.

For the “*quick task completion*” question, I separated two semi-automatic annotation interfaces (the right graph in Figure 5.24). The results showed that participants answered differently with very strong significance, $F(2, 18)=21.1, p < 0.001$. Participants typically gave low ratings for manual annotation interfaces, which is not surprising considering the results with the *Event Task*. Participants were able to finish the given tasks in about half the time. However, participants also gave very high ratings for semi-automatic annotation with clothing based human recognition, even though there was only 6% difference in the task completion time. Tognazzini [68]

emphasizes the importance of reducing *subjective time*. Compared to *objective time*, *subjective time* represents the users' engagement with the task. Also, Csikszentmihalyi [18] put a very strong emphasis on users' engagement for better experience. This result is another strong confirmation that using the semi-automatic annotation interface is less tedious than using the manual annotation interface.

In addition, participants showed much more enthusiasm with the clothing based human recognition. One participant was annotating photos which were not in the given task. He stated that he just wanted to annotate everybody in the collection.

5.8 Summary and Discussion

In this chapter, I explored semi-automatic techniques to help users make accurate annotations with low effort. While metadata is very important for browsing and searching photos, it is hard to acquire accurate metadata associated with photos. Automatic metadata extraction is typically fast but inaccurate while manual annotation is slow but accurate. I designed and implemented a semi-automatic annotation prototype, SAPHARI which combines these two techniques by generating image clusters which facilitate efficient bulk annotation. SAPHARI automatically creates these image clusters with hierarchical event clustering and clothing based human recognition. I performed a user study with seven participants. The results showed the potential benefit of the semi-automatic annotation when applied on personal photo collections. In the user study, users were able to make annotation 49% and 6% faster with the semi-automatic annotation interface on event and face tasks, respectively.

In SAPHARI, the semi-automatic annotation interface is integrated with a zoomable user interface. Users are able to navigate using zoomable browsing techniques, zooming in to see more detail and zooming out to see the overview of images. During the user study, I observed that zoomable navigation helped users when searching annotation targets. The participants were able to find events in their collection immediately in a zoomable space. One interesting characteristic of personal photo collections is that users are already well aware of photos in their collection. Combined with zoomable user interface techniques, familiar photos seem to play a very important role in efficient browsing. Compared with previous user studies (see Chapter 4) which were designed to browse non-familiar images, search performance appeared to be improved when participants were browsing with their personal photos because they are familiar with their personal photos. Even when photos were represented in very small thumbnails, participants were able to remember details of the photos. This suggests that zoomable user interfaces have a great potential when used for handling familiar information. Even though this hypothesis is not confirmed, I report a very strong empirical observation.

There are a number of possible technical improvements for the research described in this chapter. The face detector used in SAPHARI can be replaced with one with higher accuracy. The Viola-Jones face detector [46][52][69] used in SAPHARI is often heavily affected by lighting conditions and fails to work properly. In addition, SAPHARI only detects frontal face views. Although the frontal view is the most common form of people in photos, supporting lateral views will increase the accuracy of human recognition. More efficient face detector will increase the effectiveness of

clothing based human recognition. Another important improvement will be updating human models by using users' feedback. This would result in fixing recognition errors more efficiently. For example, when a user moves a face into other face group, it would update the human model associated with the face group and would result in other similar faces being regrouped. For users, fixing one recognition error would effectively correct multiple recognition errors. Further research is required on efficient human model updating and corresponding face group restructuring.

As explained earlier, there are a couple of assumptions that I made when designing SAPHARI; I assumed that: 1) photo collections are episodic; and 2) people usually wear the same clothing within a day. In addition, there are some implicit assumptions. In the unusual case when these implicit assumptions are not met, SAPHARI does not work well. For example, when people wear uniforms or when people are in swimming suit, clothing based human recognition cannot be applied. SAPHARI also assumed all photos have valid timestamps. When the timestamps of photographs are modified, SAPHARI generates inaccurate results. Further research has to be made on cases where these assumptions are not met.

Another important future research is to compare the accuracy of clothing based human recognition with that of face recognition systems. Because of no access to commercial face recognizers, I was not able to compare the quality of face clusters generated by clothing based human recognition. While SAPHARI shows a great potential, clothing based human recognition is still open for comparison.

There also were some usability issues with SAPHARI. Some participants did not like drag-and-drops. Some of them complained about difficulties in marquee-selecting images. Sometimes, users were confused between “selection mode” which allows users to select images and “zoom view” which enable users to navigate a zoomable space. Further refinement is needed on these issues.

Semi-automatic annotation is still in its early stage. Even though computer vision research has developed many useful techniques, only a few are directly applicable to personal photos. I hope further research will provide useful automatic recognition techniques which can be integrated with user interface strategies to help users manage ever-growing personal photo collections.

Chapter 6

Conclusion

6.1 Summary of Work and Contributions

In this dissertation, I propose novel techniques to help users manage their image collection. This research topic becomes increasingly important as users begin to experience the difficulties of having to manage large numbers of digital images.

Two primary challenges associated with designing efficient image management tools are identified - thumbnail presentation and metadata acquisition.

To address these problems, my research spans three areas. First, I applied zoomable user interface techniques into image browsing. I worked on redesigning and implementing PhotoMesa and present two successful cases where PhotoMesa is embedded into their browsing environments. Second, I introduced a better way of generating thumbnails. Based on a human visual attention model, I am able to crop out peripheral regions of images. User studies showed that users perform visual searches better with the cropped thumbnails. Finally, I investigated a semi-automatic annotation approach where users can make efficient and accurate annotations on their personal photos. I designed and implemented a semi-automatic annotation prototype, SAPHARI. It generates image clusters which facilitate efficient bulk annotation. For automatic clustering, I introduce hierarchical event clustering and clothing based

human recognition. Experimental results demonstrate the effectiveness of the semi-automatic annotation when applied on personal photo collections.

The research in this dissertation contributes to bringing human computer interaction and computer vision closer together. Based on the consistent errors of computer vision based object recognition, I have enhanced the user interface of digital image management systems to let users fix those errors. I summarize the contributions into three categories:

- Contributions to image application builders
 - *Application of zoomable user interface techniques on image browsing environments:* I take part in design and implementation of two prototypes, PhotoMesa and SAPHARI, and show that a large number of images can be displayed on the screen with reasonable performances.
 - *Design and implementation of the ZPhotoMesa component:* By using a simple set of software programming interfaces, an application can easily incorporate zoomable image browsing in its interface.
- Contribution to thumbnails
 - *An automatic thumbnail cropping algorithm that creates small but legible thumbnails:* I introduce two new steps – critical area identification and information based cropping – prior to shrinking in the thumbnail generation process.

- *Experimental results verifying that saliency is a reasonable proxy for informativeness in images:* The saliency based cropping algorithm successfully removes the periphery of images while preserving critical areas.
 - *Experimental results confirming that cropping based on semantic information produces more effective thumbnails:* My research uses facial information as an example of semantic information. Using a face detection algorithm as a method of identifying semantic information, I was able to produce better thumbnails which allowed users to perform visual searches 50% faster.
 - *Experimental results showing that cropped thumbnails significantly increased the user's ability to recognize and search images:* The series of studies show that users performed visual searches more than 18% faster with cropped thumbnails.
- Contribution to image annotation
- *Semi-automatic annotation strategies and design principles suggested in SAPHARI:* I propose the use of hierarchical event clustering for annotating events and clothing based clustering for annotating people. I also suggest a set of design guidelines to be used in developing a semi-automatic annotation interface.
 - *Empirical results showing that users annotate events more efficiently with the semi-automatic annotation interface in SAPHARI:* Users were able to make

event annotations 49% faster with the semi-automatic annotation interface compared with the folder based manual annotation.

- *Empirical results indicating that the clothing based human recognition may work reasonably:* Although my study results show a 6% performance increase on average, there was high variance which makes the finding statistically insignificant. However, users clearly preferred the semi-automatic annotation interface over the manual annotation.

6.2 Future Work

I have presented specific pieces of future work within the respective chapters. I will close my dissertation with an overview of my larger research agenda. In this thesis, I incorporate automatic recognition systems into the user interface. I have combined novel user interface techniques with various automatic recognition techniques such as face detection, temporal gap based event identification, Gaussian kernel based probability density function estimation, Bhattacharyya distance and saliency based critical area identification. However, I limited the scope of my research to strategies and techniques which increase users' annotation performance. One important research agenda is to broaden the scope and build a general framework between automatic recognition systems and user interface design techniques.

Automatic recognition systems inevitably bring in inaccurate results and users are required to correct them for accurate metadata. In this dissertation, I have investigated various ways of accelerating the process and introduced useful semi-automatic techniques. However, I expect that semi-automatic approaches are not necessarily

optimal choices over manual approaches at all times. Depending on information types and the accuracy of underlying automatic recognition systems, the decision between manual and semi-automatic interface techniques should be determined. The challenge is to provide general and practical criteria to decide which to choose.

Figure 6.1 shows an expected relationship between users' performance and the accuracy of automatic recognition systems. The x axis represents the recognition accuracy of underlying automatic recognition systems and the y axis represents users' efficiency which can be measured by the amount of metadata annotated with limited resources such as time, users' attention e.g. annotation per second. A manual annotation interface, which does not make use of any automatic recognition, is independent from the accuracy of the automatic recognition systems. Therefore, we can denote manual annotation interfaces as being constant in the figure. However, with semi-automatic annotation interfaces, it is expected that users' performance does interact with the accuracy of automatic recognition systems. It is natural to assume that users' efficiency increases as the accuracy of underlying automatic recognition system enhances. Figure 6.1 shows an example curve for semi-automatic annotation interfaces.

There are a couple of interesting points in this expectation. First, with poor recognition systems, manual annotation may be better than semi-annotation interface (denoted by the region A in Figure 6.1). Second, as the accuracy gets better, the users' performance increases (denoted by the region B in Figure 6.1). However, the

figure is an early prediction and needs refinement. Further research is required to confirm the relationship, and to identify the crossover point between regions A and B.

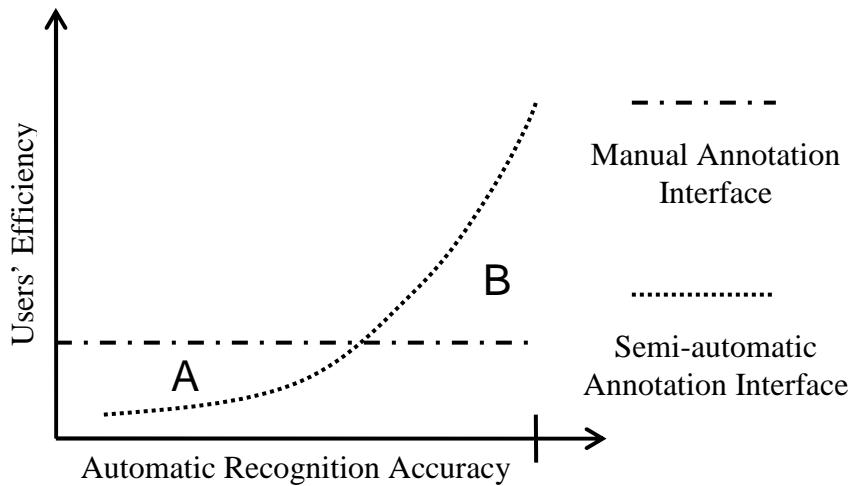


Figure 6.1 Expected relationship between the accuracy of automatic recognition systems and users' annotation performance.

Empirical experiences with SAPHARI go along with the expectation as in described Figure 6.1. The user study results showed that semi-automatic annotation interfaces help users make annotation efficiently. However, the user study contributes only a few data point in the relationship curve in Figure 6.1. The information about other parts of the relationship is still incomplete. For example, it is yet unclear how accurate recognition systems need to be, in order to achieve a certain amount of efficiency. Future research should focus on revealing uncertain parts of the relationship. I hope further research will also provide useful guidelines for designing user interfaces which have to deal with inaccurate information generated by automatic recognition systems.

In the longer term, I also hope to tackle other issues involved in browsing and searching general media information. Along with digital photos, users begin to stack up audio and movie clips on their computers. While browsing and searching these types of information are common tasks for users, they may require different management approaches just like digital images need additional management strategies over conventional document management principles. For example, many researchers have worked on summarizing a movie clip into one or limited number of images. Some have focused on extracting a theme part from arbitrary audio clips. But, there has been relatively little research focused on designing user interfaces for those types of media. I believe that lesson learned in this dissertation may be applied to design user interfaces supporting those types of general media.

Appendix A

User Study Material

A1. Consent Form Used for Automatic Thumbnail Cropping

User Study

CONSENT FORM	
Project Title:	Evaluation of Image Browsing Interface Using Thumbnails
Age of subject: (parental consent) needed for minors)	I am over 18 years of age and wish to participate in a program of research being conducted by Prof. Ben Bederson in the Department of Computer Science at the University of Maryland, College Park, Maryland 20742.
Participation	I understand that participation in this study is voluntary.
Purpose:	The purpose of this research is to compare the effects of various browsing and image manipulation techniques on image browsing, searching, recognition and satisfaction.
Procedures:	I will first read a brief description of the experiment and will be allowed to ask questions. Then I will be shown a tutorial that lasts about 10 minutes. Next, I will select images that match given conditions among images shown on the screen. I will also answer questions if I can recognize the objects in images. Following the experiment, I will complete a questionnaire to gauge subjective satisfaction. The combined duration of the experiment will be about one hour.
Confidentiality:	All information collected in the study is confidential, and my name will not be used in the analysis or reporting of the experiments.
Risks:	I understand that the experiments pose no risk to me and I will use a personal computer for about an hour.
Benefits, Freedom to withdraw and to ask questions:	I understand that the experiment is not designed to test me personally, but that the investigator hopes to learn more about the effectiveness of image browsing interfaces and thumbnail techniques. I understand that I am free to ask questions or to withdraw at any time without any penalty.
Name, Address, Phone Number of Principal Investigator:	Prof. Ben Bederson Computer Science Department, 3171 A.V. Williams Building University of Maryland, College Park, MD 20742 (301) 405-2764
Name and Signature of Subject:	<hr/> <hr/>
Date:	

A2. Pre-user study Questionnaire for Semi-automatic Annotation Interface User Study

Demographic Information for Semi-Automatic Annotation User Study

Thank you for participate in this user study!

Please answer the following questions so that we understand your background.

All data will be anonymized before we publish it.

Sex: Male _____

Female _____

Age: <20 _____

20-29 _____

30-39 _____

40-49 _____

50+ _____

Occupation: _____

Computer use
per week

0-2 Hours

3-5 Hours

6-9 Hours

10-19 Hours

20+ Hours

How long have you been keeping your digital photo collection? _____

How many photos are in your digital photo collection? _____

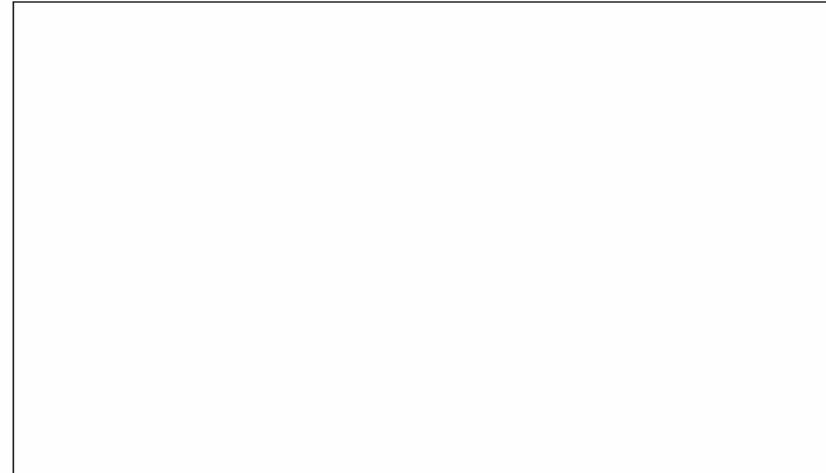
What photo management tools are you using? _____

How often do you browse your collection? _____

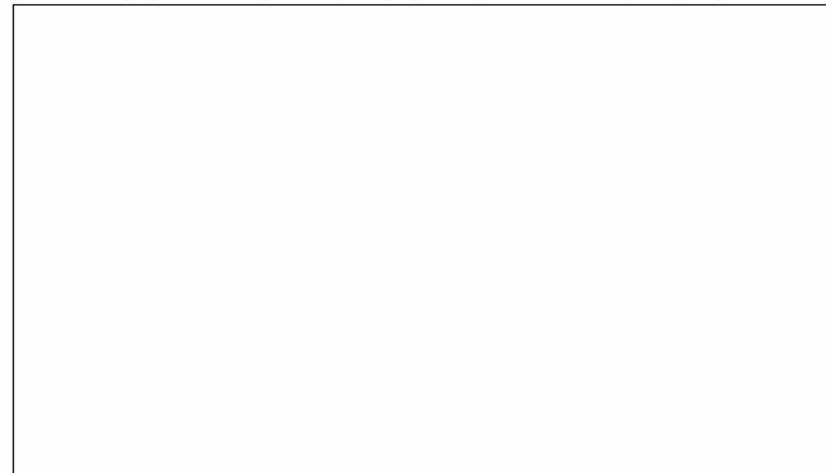
How do you organize your digital photos?

Please state at least five interesting events or places in your photo collection.

(E.g. Halloween, Birthday party, College Park, Camping Trip, etc.)



Please state at least five people appear in your photo collection. You don't have to list all the people. However, please include people who are important to you – people who you want to find in your digital photo collection.



A3. Post-user study Questionnaire for Semi-automatic Annotation Interface User Study

Semi-Automatic Annotation Interface

Question	Strongly Disagree Strongly Agree						
	1	2	3	4	5	6	7
It was simple to use this system							
It was easy to learn to use this system							
I was able to complete <i>event</i> annotations using this system quickly							
I was able to complete <i>face</i> annotations using this system quickly							
It was easy to find the information I needed							
Overall, I liked using the interface of this system							

Additional Comment:

Manual Annotation Interface

Question	Strongly Disagree Strongly Agree						
	1	2	3	4	5	6	7
It was simple to use this system							
It was easy to learn to use this system							
I was able to complete annotations using this system quickly							
It was easy to find the information I needed							
Overall, I liked using the interface of this system							

Additional Comment:

Bibliography

1. ACDSee, ACD Systems, <http://www.acdsystems.com>
2. Adobe Photoshop Album, Adobe Systems Incorporated.,
<http://www.adobe.com/products/photoshopalbum/>
3. Barron, C. and Kakadiaris, I., Estimating anthropometry and pose from a single image. Computer Vision and Image Understanding, 81, No. 3, pp.269-284, 2001.
4. Bederson, B. B. PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. UIST 2001, ACM Symposium on User Interface Software and Technology, CHI Letters, 3(2), pp. 71-80. 2001.
5. Bederson, B. B., Fisheye Menus. UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters, 2(2), pp. 217-225, 2000.
6. Bederson, B. B., and Boltman, A. Does Animation Help Users Build Mental Maps of Spatial Information? In Proceedings of Information Visualization Symposium (InfoVis 99) New York: IEEE, pp. 28-35, 1999.
7. Bederson, B. B., Meyer, J., and Good, L. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *UIST 2000*, CHI Letters, 2(2), pp. 171-180, 2000.
8. Bederson, B. B., Shneiderman, B., and Wattenberg, M. Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, ACM Transactions on Graphics, 21 (4), 833-854, ACM Press, 2002.

9. Bederson, B. B., Stead L., and Hollan, J. D., Pad++: Advances in Multiscale Interfaces, Extended Proceedings of 1994 ACM SIGCHI Conference, 1994.
10. Bell, G., A Personal Digital Store, Communications of the ACM, Vol. 44, No. 1 (January 2001), 86 - 91, 2001.
11. Bhattacharya, A. On a measure of divergence between two statistical populations defined by their probability distributions, Bulletin of Calcutta Maths Society, vol. 35, pp. 99-110, 1943.
12. Bruls, M., Huizing, K., and van Wijk, J. J. Squarified Treemaps. In Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000) IEEE Press, pp. 33-42, 2000.
13. Chen, H., and Dumais, S. T. Bringing order to the web: Automatically categorizing search results. In Proc of CHI 2000, 2000.
14. Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H., and Zhou, H. A Visual attention model for adapting images on small displays, ACM Multimedia Systems Journal, Vol. 9, No.4, pp. 353-364, 2003.
15. Chen, X., and Zhang, H. Text Area Detection from Video Frames. In Proc. Of 2nd IEEE Pacific-Rim Conference on Multimedia (PCM2001), Beijing, China, pp. 222-228, October 2001.
16. Cooper, M., Foote, J., Girgensohn, A., Wilcox, L. Temporal Event Clustering for Digital Photo Collections, Proc. of the 11th ACM International Conference on Multimedia, (MM '03), 2003.
17. Corbis, <http://www.corbis.com>

18. Csikszentmihalyi, M. Finding Flow: The Psychology of Engagement with Everyday Lift, New York, Basic Books, 1997.
19. Druin, A., Bederson, B. B., Hourcade, J. P., Sherman, L., Revelle, G., Platner, M., and Weng, S. Designing a Digital Library for Young Children: An Intergenerational Partnership. In Proceedings of Joint Conference on Digital Libraries (JCDL 2001) ACM Press, pp. pp. 398-405, 2001.
20. Exif, <http://www.exif.org>
21. Face Detection Demonstration. Robotics Institute, Carnegie Mellon University
<http://www.vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi>
22. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W., "Efficient and effective querying by image content," *J. of Intelligent Information Systems*, vol. 3, no. 3-4, pp. 231-262, July 1994.
23. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. Query by Image and Video Content: The QBIC System, *IEEE Computer*, Volume: 28, Issue: 9 , pp.23 -32, Sept. 1995.
24. Folk, C.L., Remington, R.W., and Johnston, J.C. Involuntary covert orienting is contingent on attentional control settings. *Journal of Experimental Psychology: HP&P*, 18:1030-44, 1992.
25. Gargi, Ullas, Consumer Media Capture: Time-Based Analysis and Clustering, Hewlett-Packard Tech. Report HPL-2003-165, 2003.
26. Gargi, U., Deng, Y., Tretter, D.R. Managing and Searching Personal Photo Collections, Hewlet Packard Tech Report, HPL-2002-67, 2002.

27. Grgensohn, A., Adcock, J., Cooper, M., Foote, J., Wilcox, L. Simplifying the Management of Large Photo Collections. Human-Computer Interaction INTERACT '03, IOS Press, pp. 196-203, September 1, 2003.
28. Grgensohn, A., Adcock, J., Wilcox, L., Leveraging face recognition technology to find and organize photos, In Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval, pp. 99-106, 2004.
29. Google image search engine, <http://images.google.com>
30. Graham, A., Garcia-Molina, H., Paepcke, A., Winograd T., Time as essence for photo browsing through personal digital libraries, Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries, July 14-18, 2002.
31. Grosjean, J., Plaisant, C., & Bederson, B. B., SpaceTree: Design Evolution of a Node Link Tree Browser. Proceedings of Information Visualization Symposium (InfoVis 2002) New York: IEEE, pp. 57-64, 2002.
32. Hourcade, J. P., Bederson, B. B., Druin, A., Rose, A., Farber, A., & Takayama, Y. The International Children's Digital Library: Viewing Digital Books Online. *Interacting With Computers*, 15(3), pp. 151-167, 2003.
33. iPhoto, Apple Computer Inc. <http://www.apple.com/iphoto/>
34. Itti, L., and Koch, C. A Comparison of Feature Combination Strategies for Saliency-Based Visual Attention Systems, *SPIE human vision and electronic imaging IV(HVEI'99)*, San Jose, CA, pp. 473-482, 1999.
35. Itti, L., Koch, C., and Niebur, E., A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), pp. 1254-9, 1998.

36. Johnson, B. and Shneiderman, B., Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures, In Proc. IEEE Visualization '91, pp. 284-291, IEEE CS, 1991.
37. Jul, S., and Furnas, G. W. Critical Zones in Desert Fog: Aids to Multiscale Navigation. In Proceedings of User Interface and Software Technology (UIST 98) ACM Press, pp. 97-106, 1998.
38. Kang, H., Personal Media Exploration with Semantic Regions, CHI '03 extended abstracts on Human factors in computing systems, pp.668-669, 2003.
39. Kang, H., and Shneiderman, B. Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder, *In Proc. of IEEE International Conference on Multimedia and Expo (ICME2000)* New York: IEEE, pp. 1539-1542, 2000.
40. Koenemann, J. and Belkin, N.J., A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. Proc. ACM CHI96 pp. 205-212, 1996.
41. Kohonen, T., Self-Organization and Associative Memory, Springer-Verlag, 1989.
42. Kuchinsky, A., Pering, C., Creech, M.L., Freeze, D., Serra, B., and Gwizdka, J. (1999) "FotoFile: A Consumer Multimedia Organization and Retrieval System", *Proc. ACM CHI99 Conference on Human Factors in Computing Systems*, pp. 496-503, 1999.
43. Lamping, L., Rao, R., and Pirolli, P., A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In Proceedings of the Conference on Human Factors in Computing Systems, 1995, 401-408, 1995.

44. Leung, L., Zhang, J.S., Xu, Z.B., Clustering by Scale-space Filtering, IEEE Trans. On Pattern Analysis and Machine Intelligence, 22(12), pp. 1396-1410, 2000.
45. Li, S., Zhu, L., Zhang, Z., Blake, A., Zhang, H., and Shum, H. Statistical Learning of Multi-view Face Detection. In Proc. of European Conference on Computer Vision (ECCV) 2002, Vol. 4, pp. 67-81, 2002.
46. Lienhart, R. and Maydt, J. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.
47. Loui, A. and Savakis, A., Automatic Image Event Segmentation and Quality Screening for Albuming Applications. Proc. IEEE Intl. Conf. on Multimedia and Expo, pp. 1125-1128, 2000.
48. Maryland Interactive System for Image Searching, <http://www.isis.umd.edu>
49. Milanese, R., Wechsler H., Gil S., Bost J., and Pun T. Integration of Bottom-Up and Top-Down Cues for Visual Attention Using Non-Linear Relaxation, In proc. of Computer Vision and Pattern Recognition, IEEE, pp. 781-785, 1994.
50. Milanese, R. Detecting Salient Regions in an Image: from Biological Evidence to Computer Implementation, Ph.D. thesis, Univ. of Geneva, 1993.
51. Office Picture Library, Microsoft Inc. <http://office.microsoft.com>
52. Open CV, Open Source Computer Vision Library, Intel Inc. <http://www.intel.com/research/mrl/research/opencv>
53. Phillips, P. J., Grother, P. J., Michaels, R. J., Blackburn, D. M., Tabassi, E., Bone, J. M. "Face recognition vendor test 2002: Evaluation report." NISTIR 6965, 2003.
54. Picasa, Automatic Photo Organizer, <http://www.picasa.com>
55. Piccolo, <http://www.cs.umd.edu/hcil/piccolo>

56. Platt, J. C., Czerwinski, M., and Field, B. "PhotoTOC: Automatic Clustering for Browsing Personal Photographs", Microsoft Research Technical Report MSR-TR-2002-17, 2002.
57. Revelle, G., Druin, A., Platner, M., Bederson, B. B., Hourcade, J. P., & Sherman, L. E. (2001). Young Children's Search Strategies and Construction of Search Queries. *Journal of Science Education and Technology*, 11(1), pp. 49-57.
58. Rodden, K. and Wood, K., How do People Manage Their Digital Photographs?, ACM Conference on Human Factors in Computing Systems (ACM CHI 2003), Fort Lauderdale, April 2003.
59. Rosales, R. and Sclaroff, S., Inferring body pose without tracking body parts. In Proc. IEEE Computer Vision and Pattern Recognition, 2000.
60. Schneiderman, H., and Kanade, T. A Statistical Model for 3D Object Detection Applied to Faces and Cars. *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, June, 2000.
61. SharePoint, Microsoft Inc., <http://www.microsoft.com/sharepoint/>
62. Shen C., Lesh N., Moghaddam B., Beardsley P. and Bardsley R.S. Personal Digital Historian: User Interface Design. Proceedings Extended Abstract of CHI'2001, ACM Press, pp. 29-30, 2001.
63. Shen, H.T., Chin, O.B., and Tan, K., "Giving Meanings to WWW Images", ACM SIGMM'2000 Multimedia Conference, L.A., 2000.
64. Shneiderman, B., Tree Visualization with tree-maps: A 2-D space filling approach. ACM Transactions on Graphics, 11(1) pp. 92-99, January 1992.

65. Silverman, B. W., Density estimation for statistics and data analysis, Monographs on Statistics and Applied Probability. 26, Chapman & Hall 1986.
66. Suh, B., Ling, H., Bederson, B. B., and Jacobs, D. W. Automatic Thumbnail Cropping and Its Effectiveness, UIST 2003, CHI Letters, 5(2), ACM Press, 2003.
67. Suh, B., Woodruff, A. Rosenholtz, R, and Glass, A. Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces, CHI 2002, CHI Letters, 4(1), pp. 251-258, ACM Press, 2002.
68. Tognazzini, B. T., "Principles, Techniques, and Ethics of Stage Magic and Their Application to Human Interface Design," Proceedings of INTERCHI, 1993 (Amsterdam, The Netherlands, April 24-29, 1993). ACM, New York, pp 355-362, 1993.
69. Viola, P. and Jones, M., Robust real-time object detection. International Journal of Computer Vision, 57(2), pp. 137-154, May 2004.
70. Wattenberg, M. Visualizing the Stock Market. In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 99) ACM Press, pp. 188-189, 1999.
71. Wenyin, L., Dumais, S., Sun, Y., Zhang, H., Czerwinski, M. and Field, B. Semi-Automatic Image Annotation. In Human-Computer Interaction--Interact '01, Hirose, M. (Ed.), IOS Press, pp.326-333, 2001.
72. Wolfe, J.M. Guided Search 2.0: A Revised Model of Visual Search, Psychonomic Bulletin and Review, Vol. 1, No. 2, pp. 202-238, 1994.

73. Yang, M., Kriegman, D., and Ahuja, N. Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis and Mach Intelligence*, 24(1), pp. 34-58, 2002.
74. Yee, P., Swearingen, K., Li, K., and Hearst, M., Faceted Metadata for Image Search and Browsing, In Proc. Of CHI 2003, 2003.
75. Yoshitaka, A., and Ichikawa, T. A Survey on Content-Based Retrieval for Multimedia Databases. *IEEE Trans on Knowledge and Data Engineering*, 11(1): 81–93, 1999.
76. Zhao, W., Chellappa, R., Philips, P.J., Rosenfeld, A., Face Recognition: A Literature Survey, ACM Computing Surveys, Vol 35(4), pp. 399-458, 2003.