AUTOMATIC ORGANIZATION OF LARGE PHOTO COLLECTIONS

by

Michael N. Wallick

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

June 2007

ACKNOWLEDGMENTS

There are several people who I need to thank for helping me get to this point, and they could never all fit on a single page of a dissertation. First and foremost, I must thank my adviser, Michael Gleicher, and the other members of my thesis committee (Charles Dyer, Xiaojin "Jerry" Zhu, Kurt Squire, and Mark Harrower). Without your discussions, contributions and other help this would not have been possible. I also want to thank Yong Rui and Steven Drucker of Microsoft Research who have served as "mentors" to me during my internships under their direction. I also want to thank all of the members of the UW Graphics Group, of which I have had the pleasure of working with during these past six years. Especially, Rachel Heck, who started graduate school at the same time as me on the same project. Although our research interests have diverged, I am glad to have had her as a friend and ally.

Photographs used in this research were provided by Howard Richman (and were of the Wallick and Richman families), Richard Urich, Michael Gleicher, and several other people. Without your generous donations, this work would not have been possible. Microsoft Research and the National Science Foundation have also provided funding support to make this work possible.

Finally, I want to thank my family. I thank my wife Christine for not only proofreading this entire document but also following me up to Wisconsin and sticking with me through this process. I also want to thank my parents, siblings, and grandparents for all of their love and support. Last but not least, my son Ethan for waiting until just after my defense to make his arrival in the world.

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

Appendix

Page

DISCARD THIS PAGE

LIST OF TABLES

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

Appendix
Figure Page

DISCARD THIS PAGE

NOMENCLATURE

Image	a digital 2 dimensional representation of some scene. Unless otherwise noted, images are represented in RGB (Red-Green-Blue) color space.
EXIF	Exchangeable Image File Format. A standard set of meta-data that is recorded with each image on a modern camera. EXIF data includes camera settings, camera model information, time and date, and other information depending on the camera model.
Photograph	an image containing EXIF data.

AUTOMATIC ORGANIZATION OF LARGE PHOTO COLLECTIONS

Michael N. Wallick

Under the supervision of Associate Professor Michael L. Gleicher

At the University of Wisconsin-Madison

Modern digital photography allows users to capture, store, and share thousands of digital photographs at one time. As a result, simply browsing the photo collection becomes a daunting task. A user must see and deal with every single photograph in the collection. Tasks related to browsing, such as searching for a specific photograph, or choosing a few photographs to share become equally difficult. Organizing the photographs and exploiting this organization is one way to simplify these tasks; a user may take advantage of the organization when carrying out any of the above tasks. Unfortunately organizing the photographs by hand often requires more effort than most users want to apply.

In this dissertation I show how using cues from metadata and image content, large collections of photographs can be automatically organized. The photograph collection is automatically partitioned into a hierarchy (or tree) of related "events" and then a single photograph for each event can be automatically selected to represent that group. For any given node of the tree, the user is shown only the representative photographs from the children of the node, thus reducing the visual information that they must deal with at any one time. Browsing the photographs is equivalent to traversing the tree. Other interactions with the photograph (e.g. tagging, culling, image adjustments, etc.) can be carried out on individual photographs or entire sub-trees.

The methods that I developed were informed by two user studies which I carried out. The first study shows that representative (and non-representative) photographs exist within a large collection of photographs, and that humans are able to perform such selection. The second study helps illuminate the process that humans carry out when asked to select a representative photograph. The findings of these user studies helped inform the development of new methods for automatic selection of representative photographs. I present a full implementation of these methods. The

implementation allows a user to browse, tag, and search photographs either on a desktop PC or over the World Wide Web, using an AJAX implementation of these methods.

ABSTRACT

Modern digital photography allows users to capture, store, and share thousands of digital photographs at one time. As a result, simply browsing the photo collection becomes a daunting task. A user must see and deal with every single photograph in the collection. Tasks related to browsing, such as searching for a specific photograph, or choosing a few photographs to share become equally difficult. Organizing the photographs and exploiting this organization is one way to simplify these tasks; a user may take advantage of the organization when carrying out any of the above tasks. Unfortunately organizing the photographs by hand often requires more effort than most users want to apply.

In this dissertation I show how using cues from metadata and image content, large collections of photographs can be automatically organized. The photograph collection is automatically partitioned into a hierarchy (or tree) of related "events" and then a single photograph for each event can be automatically selected to represent that group. For any given node of the tree, the user is shown only the representative photographs from the children of the node, thus reducing the visual information that they must deal with at any one time. Browsing the photographs is equivalent to traversing the tree. Other interactions with the photograph (e.g. tagging, culling, image adjustments, etc.) can be carried out on individual photographs or entire sub-trees.

The methods that I developed were informed by two user studies which I carried out. The first study shows that representative (and non-representative) photographs exist within a large collection of photographs, and that humans are able to perform such selection. The second study helps illuminate the process that humans carry out when asked to select a representative photograph. The findings of these user studies helped inform the development of new methods for automatic

selection of representative photographs. I present a full implementation of these methods. The implementation allows a user to browse, tag, and search photographs either on a desktop PC or over the World Wide Web, using an AJAX implementation of these methods.

Chapter 1

Introduction

Advances in digital photography offer the power to collect, store and share more photographs than ever before. An aggressive digital camera owner may accumulate as many as 3000 to 6000 photographs per year [18]. In addition to collecting so many pictures, picture size and quality continues to improve. At the time of this thesis, a standard consumer digital camera can capture images around 10 megapixels. This number is likely to continue to increase; this means that amateur photographers will continue to capture more photographs with higher resolution.

This dissertation addresses the problem of applying an organization to a set of photographs to aid in common tasks and make further organization simpler without requiring extra human intervention or training. A primary interaction with large photograph collections is to browse those photographs.[1] Within the context of browsing, a user may be simply enjoying the photographs, searching for a specific photograph (or set of photos), curating a specific story to tell, or performing some other browsing operation.

Unfortunately, having massively sized photo streams[2] makes it difficult to carry out basic tasks with the collection. For example, consider a user trying to find a specific picture in a minimally ordered set of thousands of photographs. The user would have to search through each and every picture in order to find the one that is desired. Likewise, the sheer number of photographs would prevent the user from being able to share all of the photographs with friends or family, as people are not willing to sit through long photograph presentations. Instead, a small set of photographs

[1] While there are other important interactions that are carried out with digital photograph collections, this dissertation only focuses on browsing specific operations.

[2] In this context, I define a "photo stream" as a collection of photographs taken over some period of time by a single photographer. Each photograph in the stream has the time it was captured associated with it.

would need to be selected for sharing; again it is a daunting task having to go through each and every picture to select the ones that are best for sharing.

Adding some type of organization, or structure, to the photo stream can make these tasks easier to perform. For example, if the photographs are organized by time, a user can use this information to narrow down the search for a specific photograph. In reality, the photo stream is never completely unorganized - at the very least the operating system will enforce some structure on the stream, such as ordering the photographs by time taken, alphabetical by file name, date last viewed, etc. However, the more organization that is given to images, the easier the tasks become. A simple temporal organization does not help a user select good photographs for sharing; and the act of finding a single photograph can still be improved. This dissertation describes automatic methods that can be applied to a photograph collection to provide further organization to the photographs.

1.1 Problem Statement

We are able to produce more pictures than ever before. Without some type of organization scheme, it becomes extremely difficult for a user to browse, search or share personal photographs. A computer operating system can help organize files, but as the number of files increase, the organization becomes less effective. To this end there have been several products and research projects to help organize photographs. Digital photograph browsers (for example, Adobe Lightroom, Apple iPhoto, Google Picassa, etc.) automatically organize photographs based on a time-line; the photographs are ordered by the time they were taken in one dimension (possibly more if the user takes the time to organize a deeper structure by hand). As the collection grows larger, the user may become overwhelmed by the number of photographs, as such a view shows all of the photographs at once.

At the beginning of this research, I conducted an informal survey among computer science graduate students who actively take pictures, asking how they generally organize their sets of pictures. Overwhelming, the response of those who do not use one of the photo tools listed above[3],

[3]Only approximately 20% of the respondents said they use a photo organization tool, the remaining use the file system.

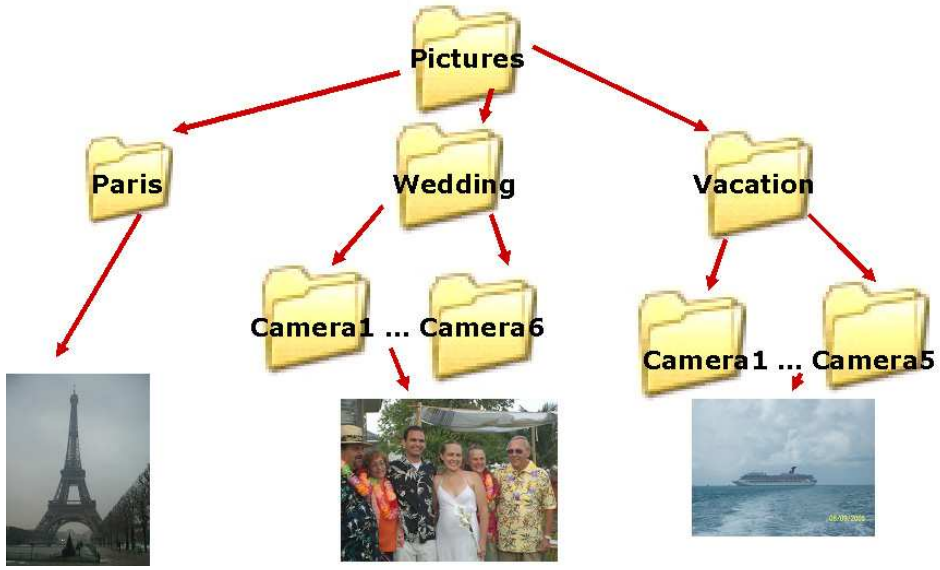

Figure 1.1 Example of typical photo storage solution, using the file system.

was that they simply use the file system. There is a single folder marked ”pictures” and each event, or days worth of pictures is given a subfolder. If more than one camera is taking pictures, then each camera is given a subfolder under the event. Figure 1.1 is a graphical representation of such a storage scheme. In general this does not allow flexibility in browsing, sharing, or finding individual pictures.

In order to create an organization that is not based on time, semantic knowledge of the set is required. In other words, knowledge of the event: who participated, what was happening, where did it take place, why was it being photographed, how does it relate to other photographs in the collection, etc. Current computer vision and other technology does not provide a generic, robust method for automatically acquiring this information. Rather, the best way to get the most semantic information to is to have a user supply this by hand. Most existing tools allow a user to give this information (generally in the form of tags), however they require that each image is tagged individually, although better tagging interfaces have been added to photo management tools. Additionally, existing tools will not easily transfer the metadata provided by the user to the other applications, meaning that a user will have to supply the information for every single photograph in every single application.

As the photograph set becomes more organized, a user should be able to leverage that organization, making it easier (or at the least not any harder) to interact with the collection. However, giving more organization to the photograph sets often requires more up-front work by the user than the perceived payoff [41]. In this dissertation, I propose a set of methods that will aid in the automatic organization of large photo collections. These methods can be implemented as a new browsing tool and interface for photographs. As such, it can be a part of a file system browser, integrated into existing photo organization tools, or as a stand-alone photo browser. I implement each of these tools as either a stand-alone browser, a web application, or both; see Chapter 6 for further details. The organizational methods proposed will not solve every possible task that may be encountered when dealing with photographs, however they provide an initial (or "first pass") organizational structure that is more detailed than what current tools provide, and address the primary task of browsing photographs. The user may either interact with the photographs directly in the new organized structure, or use it as a starting point to further organize the photo set.

It should be noted that the problem I address is similar to that faced by professional photographers, i.e. what is the best way to organize a large collection of digital assets. Professional solutions are available [25], however they are time and resource consuming. Most photographers are amateurs and do not have the time or money to invest in a professional solution. The methods proposed in this thesis allow some level of organization (although not professional) without incurring any extra work for the user. Professional solutions generally include a meticulous labeling of every single photograph, based on attributes such as time/date, subjects, event type, poses, lighting conditions, copyright holder, etc. Often this is carried out by an assistant, rather than photographer.

1.2 Key Insight

Automatic organization of photographs can be a challenging problem. When performing this task manually, humans rely on the contents of the photographs, and possibly their knowledge of the event, to determine the context of the stream in order to make organizational choices. A computer lacks the knowledge of the event, and has no way of determining an unconstrained "context" for

an arbitrary stream. Determining context within images remains an open problem in the computer vision field. While there have been significant advances it is still far from a solved problem.

Rather than trying to determine this intangible, or abstract, high-level information, I rely on the following key insight, on which this work is based. Photographs taken over a relatively short period of time, taken by the same photographer, are related to each other. It would be physically impossible to have two completely unrelated photographs that are taken within a few moments of each other. This is a variation on Tobler's First Law of Geography [56], which states that "everything is related, but near things are more related than distant things." This insight leads to the methods that I developed to automatically organize photographs.

1.3 Proposed Solution

This dissertation addresses the problem of dealing with large collections of photographs by organizing the collection. When a collection is organized, it is easier for a user to browse or find individual photographs in the collection. **It is my thesis that a stream of photographs can be automatically organized into a tree of groups which can in turn be abstracted by displaying a small representative subset of the entire photo stream; this organization simplifies the task of browsing, and thus tasks relating to browsing, by providing further automatic photo organization.** In my approach there are three distinct steps that are carried out to achieve this goal.

In the first step, the photographs are grouped into smaller related sets (Chapter 3). The sets are grouped as a tree (or hierarchy), where each set is a node that represents an event that was photographed. A node further down in the tree (a subset of the parent) represents a sub-event of the parent node. For example assume a node in the tree is all of the photographs from a birthday party. The children nodes may include the party games, the cake, and opening the presents. The key insight, that photographs *relatively close* together in time are related, is what makes this organization scheme possible. Different levels of the tree have a different meaning for relatively close.

The second step is to summarize each set of photographs (Chapter 4). This is done by selecting a single image from each set to represent the entire set. This reduces the number of images that

have to be displayed at any one time, rather than displaying the entire photo stream as traditional software does. Again, the key insight says that photographs taken close together in time are related. This implies that there should be at least one photograph in each set that can serve as a representative image of the entire set. Continuing with the birthday party example, a set of representative images may include a photograph of people playing a game, blowing out the candles, and someone opening the presents. I show how different automatic selection methods compare, and present a new method for carrying out this task automatically.

The final step is laying out the photographs (Chapter 5). Each node in the tree has several photographs associated with it, however the representative photographs from the child nodes are the only ones that need to be displayed, again reducing the visual complexity. Depending on the desired use for the photographs different layouts may be applied. I have implemented four different types of layouts: a grid layout, a time line layout, and two collage layouts. The grid layout displays each representative photograph in temporal order. This layout is useful when a user is trying to find specific photographs. The collage layouts are a more artistic display. Each photograph is given a different size in the display and is not ordered by the time that it was taken.

The three steps outlined above are combined together to both automatically organize photographs and create an interface to interact with the photos within the organizational structure. Any node in the tree is shown by a representative photograph for that node. A user can browse the stream by selecting a representative photograph and move down the tree to the child node. The user can use this interface to browse, sort, or tag the photographs. Sharing can be done by sharing the tree, either entirely or specific branches; or a specific path through the tree may be shared. With this approach every single photograph may be shared, without requiring the recipient to view every single image. I describe my implementation of these applications in Chapter 6.

1.4 Requirements for New Methods

In considering existing photo browsing and organization software, there are several features that almost all existing software lacks. I consider the methods that I present to be successful if they include, address, or improve these areas. In Chapter 2, I describe other systems and explain how

they address (or fail to address) each of these areas. In Chapter 7, I revisit these requirements and discuss how my methods address or improve on each of these requirements.

Automatic and Reliable Organization. As the size of digital photograph collections grow, it becomes more burdensome to organize photographs by hand. A good system should provide an automatic and robust organization scheme, which makes sense to the user, so that pictures can be logically grouped together.

Reduce Visual Information In Principled Manner. Many of the photographs in a large collection are visually redundant. A system can exploit this fact by only using a small number of images from each subset found by the automatic organization (above) to represent the whole collection. However, including a bad image as representative can confuse the user. This can be avoided by selecting several images, but selecting too many images increases the visual complexity.

Provide Simple and Understandable Navigation. Since the user has to deal with many pictures which have been organized and "visually reduced," a simple and understandable navigation scheme is necessary. Such navigation should be initiative and/or similar to methods that are already familiar to users.

The main goal of this dissertation is to aid in the tasks of browsing, searching, and sharing large collections of digital photographs. These requirements work together to build a new interface that aids with these tasks.

1.5 Contributions

The main contribution of this dissertation is the development of a new organization and interface for dealing with large collections of digital photographs. This new interface gets away from the traditional album-like organization which is a holdover from traditional print photographs. In order to achieve this, several other contributions in the field of computer science have been made. The following is a list, in the order that they are discussed in this dissertation:

Photograph Clustering. I present a hierarchial clustering method to work with photographs. Because it is tailored to photographs, it works faster and more reliably than standard generic clustering algorithms. (Chapter 3.)

Comparison of Different Image Selection Algorithms. Many photograph organization applications rely on the idea that a single image can represent a larger set. I present multiple studies which compare the standard methods for selecting a single image. Further, I have built a database of annotated images (marked as being representative, non-representative, or neither) that can be used as a benchmark for new applications as they are developed. I also show a formula to model human behavior for selecting representative images. (Chapter 4.)

Implementation of a New Image Selection Algorithm. Based on the results of my user studies, I show a new method for automatic representative image selection. This new method seems to outperform the existing techniques, and requires no human interaction. (Chapter 4.)

Photograph Organization User Interface. By combining the methods described, I present a new interface model for dealing with large collections of photographs. The interface uses the tree structure combined with representative image selection and layouts. (Chapters 5 and 6.)

Additional Photo Collection Applications. Using the photo tree concept, I present a new method for quickly tagging photographs. A tag can be applied to any node in the tree and the tag is propagated to all of the children photographs of the node. This approach can also be used for other tasks, such as image processing. (Chapter 6.)

The contributions that I make in this dissertation work in concert to meet the requirements that I described in 1.4. The photo clustering gives a reliable and automatic organization scheme. The user studies that I conducted inspired and lead to the development of a new image selection algorithm. This gives a principled method for automatically selecting representative images from a larger set. Finally, the user interface and additional applications provides a simple and understandable navigation scheme for interacting with the collection. In Chapter 7, I revisit these requirements and describe how I met each of them.

1.6 Impact

Digital photograph technology has allowed casual photographers to capture hundreds (and even thousands) of photographs in a very short period of time, with virtually no cost. This has lead many casual photographers to experience the "digital shoe box" problem; that is, all of the same problems of browsing and interacting with large collections of printed photographs stored in a shoe box still exist, only now with even more photographs.

The methods presented in this dissertation are designed to aid the casual photographer when dealing with large collections of digital photographs, without any additional investment of money or work. New applications and interfaces are presented which will help to reduce the problems and frustrations of the digital shoe box. A further discussion of the impact of this dissertation is provided in the Conclusion, Chapter 7.

1.7 Limitations

The methods presented in this dissertation are based on specific assumptions and heuristics associated with digital photography. These assumptions and heuristics allow the applications to function without requiring the user to go through any special training process or tune parameters for different sets. These methods should work for any appropriate photograph set (described below) or any user.

The assumptions and heuristics, however, do present a set of limitations to the methods presented here. Briefly, the limitations are: the time stamp must be included in each photograph's metadata, photographs are not taken at a constant interval (such as a web cam taking a picture every minute), photographs come from a single source (one camera or one photographer, not multiple photographers at different events or pictures randomly collected from the web). Further discussion of the limitations is presented in the Conclusion, Chapter 7.

Chapter 2

Related Work

This dissertation covers many different aspects of computer science, including areas in computer graphics, multimedia, computer vision and user interfaces; each having their own unique methods for dealing with large collections of photographs. In this dissertation I combine several of the ideas already presented along with new methods for approaching this problem. This chapter briefly describes some of the related work in each of these fields, as they relate to the work and ideas presented in this dissertation.

2.1 Organizing Photo Collections

The problem of organizing large collections of photographs is older than digital photography. In 2002, Frohlich et. al. [13] presented an in depth study of how 11 different families organize their photograph collections. The families that were chosen used both digital and traditional photograph technology. One of the findings of the study was how photograph organization differed between traditional and digital photographs. The study showed that while both types of photographs tended not to have large amounts of organization, the digital photographs got even less organization than printed photographs.

Another finding of the study is the suggestion that digital photograph organization should move more towards a social experience. There are several research projects and commercial ventures that help consumers deal with large sets of images which include social interactions. Three such commercial systems are Flickr [30], Kodak Easy Share Gallery [7], and Tag World [33]. These are webpages which allow users to upload pictures, label the photographs, and share them with

others around through the web interface. Flickr and Tag World both allow community labeling of photographs. This means that anyone (with permission) can apply labels to a picture, regardless of ownership. These web pages are based on photo album organization. That is, the user uploads the photographs into a specific folder or album. There is no automatic organization of the photographs. These systems can benefit from the methods that I present in this dissertation.

Similarly there are several pieces of commercial software for organizing photo collections. Most notably is Picasa [6], which is a free photo management program by Google. It is designed to store the photos and to allow for quick searches. Adobe's Photoshop Elements [5] is another piece of commercial software that includes tools for storage and organization of photographs. Elements is designed to help the user with the task of organizing photographs; for example, it employs automatic face detection and has the user manually label each found face in the photo set. Again, these systems could benefit from the added automatic organization methods that I present in this dissertation. These systems do not meet the requirements of Section 1.4 since they rely on the user for organization. There is no attempt to reduce the visual information presented, making the system less scalable. My methods provide automatic organization and reduce the visual complexity of the photograph set.

In addition to commercial ventures, the problem of dealing with large sets of images remains an open one that has been investigated by several user interface researchers. Drucker et al. [10] developed MediaBrowser. In this system, users label individual photographs and videos. The system can then put together thematically-related sets, as well perform searches on the set of images. Similar to MediaBrowser is the MiAlbum system [63]. It uses user labeling to help manage a "typical family's" digital photographs. Again, these systems rely on the user to handle the organization. When they do reduce the visual complexity it is by methods which I show in Chapter 4.3 to be no better than random chance.

Shaft and Ramakrishnan [46] developed a system which uses image classifiers and a database to organize images. The images that are placed in the database have information, such as edge map and color histogram, automatically extracted to help provide information about the photographs. In addition, the user can apply labels to objects within the image allowing the user to carry out

Figure 2.1 Example of user interface in the Photo Triage program [9].

queries to search for images. This is one example from the Image Based Content Retrieval (IBCR) field [17, 48, 49]. A major difference between the work in this dissertation and IBCR is that photographs in ICBR are related through the content of the photographs. The photographs in this dissertation are related by events being taken by the same photographer. For example, in an IBCR database, there may be many images of cats which are all related by virtue of the image contents. By contrast, if photographs where taken of several different animals in a zoo, they would be grouped together in the system that I describe, regardless of content.

Each of the above systems tries to handle the entire set of photographs but does not do much to reduce the size of the set of images. In the Photo Triage Project [9], an interface allows the user to quickly "triage" their photographs. Photographs are presented to the user in a spread-out stack, and through a rapid mouse interaction mark a photograph as "like" or "dislike." The disliked photographs can be discarded while the liked photos can be moved to some type of album for display. The user is then free to concentrate on trying to fix those photographs that received neither label. Figure 2.1 shows an example of the Photo Triage UI.

2.2 Image Clustering

Related images are often clustered together. Many systems, both research and commercial, try to use clustering to help organize the photographs. A key idea presented in this dissertation is that photographs can be automatically clustered at multiple levels in order to produce this organization. The systems described here do not do as much clustering as I present in Chapter 3. The systems described below try to break the photograph collection into separate albums, giving a two-level

organization scheme. The clustering that I propose organizes the photographs into a tree structure, so that there is a much deeper set of clusters.

AutoAlbum, developed at Microsoft Research, by Platt [38] is a system for clustering photographs. Like my proposed work, it takes the time stamp of each photograph in order to generate a clustering. In this scheme, the photographs are only organized into a single level. In Chapter 3, I argue for a multi-level event scheme. The single level works for AutoAlbum since only albums are being created; there is no concept of searches or more in-depth organization. While AutoAlbum has a good navigation, it fails to meet the other two requirements that I described in Section 1.4. In Chapter 4, I show that using the average histogram is not a reliable method for selecting a representative image to reduce the visual complexity. The methods used for automatic organization do a good job of making individual albums, but do not further organize the images. This may lead to very large albums which do not scale well.

Loui presents an alternate time and content based clustering approach [29] to automatically create photo albums based on the time that images are captured. In his approach, K-Means clustering, based on the time stamp of the photograph is used to create the albums. Computer vision techniques are also employed to further match similar pictures, as well as remove poorly taken photographs from the albums. A general problem with K-Means clustering is that the value of "K" needs to be known in advance, in order to prevent unnatural relationships from being formed.

Similar to Loui, several other researchers have proposed that photographs can be clustered by finding bursts within the time stream [4, 15, 16, 54]. A central idea of each of these works, as well as [29, 38] is that digital photographs are taken in bursts. This is because without the traditional constraints of film, a photographer will take multiple pictures of the same event (or subject) to capture the action as it unfolds or to ensure that at least one image of interest was taken. Graham, et. al [16] describes this phenomena as follows: "People tend to take personal photographs in bursts. For instance, lots of pictures may be taken at a birthday party, but few, if any, pictures may be taken until another significant event takes place... Without realizing it, the user gives structure to his personal photo collection by the way that he takes it." The works presented in [16, 54] describe how photographs can be clustered at multiple levels of the time line, a fact that I too exploit.

Both [16] and [54] use a hierarchy for clustering and their methods are most similar to my own. There are, however, some differences in the approach that the methods I present operate compared to their implementation. In [16], a constant is required to boot strap the clustering process, i.e. this constant is used at the top level to determine where the cluster boundaries should be placed. The method employed by [54] requires the tuning of three different constants in order to determine the boundaries at all levels of the tree. The methods which I present require that a single constant (which I provide) be set in order to aid in automatically determining the correct boundaries between each cluster. The other systems that I described above do not do a hierarchy of clustering. Rather, the other systems only cluster at a single level.

Other metadata, such as global position location [22, 36], has been proposed. In this case, images that are close together in physical space, are likely to be related. This information can be further leveraged against a database of known locations to help further identify and tag the photographers. The disadvantage to this approach is that while GPS location is part of the EXIF data specifications, it requires the photographer to be equipped with some type of GPS system to collect this information. Although I believe that cameras will come equipped with such capability as a standard feature in the future, current camera models that do contain GPS capabilities have a very high price point. A different approach is to allow the user to specify this information by hand, either as a tag or directly on a map [50, 30]. The disadvantage to this approach, as I describe in the next section, is that tagging photographs by hand can be a difficult and time consuming process. Although I would like to see some type of position data used as part of my methods, I do not include it as it is not practical at this time. However, I do describe how it can be incorporated in the future.

Other researchers have presented ideas on clustering images based on visual content rather than relying on the metadata of the image. Schaffalitzky and Zisserman [45] present a system for clustering images based on computer vision. Unlike previous work in computer vision, their system will cluster images of the same scene even if there is a large disparity between the two images, i.e. it does not require a small baseline. This approach works well for clustering if the photographer returns to the same place at multiple points in time. Figure 2.2 shows an example of

Figure 2.2 15 Images from a church in Valbonne, France. from [45]. Despite having a wide baseline and differing features, the system presented by Schaffalitzky and Zisserman is able to cluster these as one group.

several different photographs taken (at different angles and orientations) of a church. Their method is able to cluster these images together despite the wide baseline and other differences. Puzicha1 et. al [39] presents an in depth study of several different computer vision based techniques for clustering images together.

The above works show that images can be reasonably clustered using either metadata or visual content (or both in the case of [29]). For my work, I chose to use only the photograph metadata (time in particular) for clustering. The disadvantage to this approach is that it will not work on photographs that do not contain this data (such as images pulled randomly from the web, or photographs where this data has been lost for operations such as manipulating the photo in some program that does not preserve the metadata). The advantage, however, to using the metadata is a small but highly representative amount of information. Regardless of the size of the image, the processing time to cluster will scale linearly. In general, computer vision algorithms become slower as the photograph grows larger (or require that the image be down sampled).

2.3 Labeling Photographs

An alternative approach to clustering photographs is labeling them. If every photograph in a set has at least one label attached to it (even if that label is "unlabeled"), then there is some organization that can be applied to the photograph set. In many ways, clustering (described above) is a specific form of labeling. One of the problems with labeling photographs is that it is time consuming and tedious. Many users will not want to spend the time it takes to label every single image, because the perceived benefit does not outweigh the cost. As such, most labeling research has been looking at making labeling automatic or at least more fun.

The methods presented in this dissertation do no rely on labeling the photographs. However, the tree structure does allow a new interface for aiding in this task. Rather than requiring the user to label photographs individually, branches of the tree can be labeled. This labeling can be combined with any of the methods that I describe below. In Chapter 6 I describe how existing labeling methods can be combined with my methods to further ease the task of labeling large collections of photographs.

Much research has been done to use classifiers [19, 20] (such as face classifiers) in order to label images. Wei and Sethi [62] present an algorithm for detecting faces in images, which can in turn be used for labeling. In the most recent edition of Adobe Photoshop Elements [5], the photo album tool includes a tool that locates all of the faces in the set of photographs. The user can then label all of the faces individually.

It should be noted that while face detection works quite well, face recognition (determining the person) is still in development. Despite this, very recently a new web service, Riya [32], entered the market. This service allows users to train the system to perform face recognition, rather than simple face detection. The system also reads text on signs (and other features) in the photograph. This gives users a way to automatically label many of their photographs. At the time this proposal was written, the Riya system is able to interact with some other web-based image databases, with more on the way. Once it is more developed, I will look at incorporating Riya output into my

system. At the writing of this dissertation, Riya was not far enough along to perform a meaningful test.

In "Show & Tell," Srihari and Zhang [52] describe a system for semi-automatic annotation of images. They use a combination of image classifiers along with natural language processing to create the labeling. In their system they concentrate on medical images, as doctors are used to dictate information about imagery. Beyond using metadata for clustering, others have employed computer vision to do this task. Jeon et al. [23] developed a system for automatic labeling of images. The system takes a training set of manually-labeled images. When it encounters a new image, it attempts to match the image based on the training data. The shortcoming of this system is that it is only as good as its training set. For example, if the system encounters an image of a lion but only has images of cats in its training set, it will label the lion image as a cat. If the system encounters a picture of a lion, but only has a limited type of training data, such as architectural images, then the labeling will be wrong. Others have also presented work on automatic image labeling [14, 27, 64].

In 2000, Shneiderman and Kang [47] developed a labeling system for drag-and-drop labeling of images. In their approach, the user selects a photograph and labels, and simply drops the labels in place. This provides a simple and quick method for labeling photographs.

Recently, researchers have found that if tagging photographs is made into a game, then people will be more willing to carry out the tagging process. Notably in this area is the ESP Game and Peekaboom [58, 59]. Both games pair anonymous players with each other and have the players try to guess the same word. In doing this the users are labeling photographs on the web. The popularity of these games has allowed the creators to label millions of images from the World Wide Web. Such a system, however, does not work to label one's own personal photo collection. Meyers, et al. [35] describes a game for labeling a personal collection of photographs. They use a video game dance pad to have the labels described by the dance actions. In other words a photograph is given the labels by the dance moves. This approach is limited by the mapping of dance moves to labels.

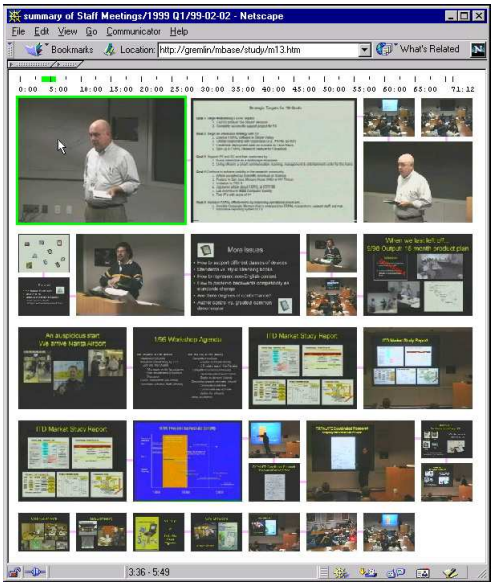

Figure 2.3 Summarized video from [2]. The more important the key frame, the larger it is in the final display.

2.4 Layout and Collage Generation

The problem of laying out many images or video frames is one that has been explored by several researchers. Work carried out at FX Palo Alto Laboratory [2, 57] looked at summarizing a video in a comic book (or Japanese Manga) style. In this system key frames are selected from the source video. The algorithmically-determined importance of the key frame dictates how much space the final image would take up. A unique packing algorithm is used to determine the final layout. Figure 2.3 shows an example of a summarized video.

Many programs and researchers create a very simple collage by laying out thumbnails of each image. This is done in Photoshop Elements and Picasa [5, 6]. In Photomesa [1], Bederson employs a ZUI or "Zoomable User Interface" to display the photograph thumbnails. In this system all of the photographs are displayed as thumbnails. The more photographs being displayed, the smaller they appear. What makes the system unique is that as the user mouses over and clicks different parts of the display the interface will zoom in on photographs in that area. The user can then drive down to show a photograph in full resolution. These systems do not address the requirements presented

in Section 1.4 as they require the user to provide the organization; and do not reduce the visual information being presented.

Fogarty et al. [12] built a system for making collages that are both aesthetically pleasing and convey information. They describe having a large digital display that is suitable to be hung as a piece of art. The system collects information that can be displayed with information that does not require constant attention, such as e-mail or news group headers. Most of the time, the collage functions as decorative artwork, however when the viewer wants to give it full attention, other information (for example the arrival of new e-mail) can be gleaned from the collage. The collages they design are very different from the collages I intend to build. Most notable is that they are not using images to design the collage; further, this system is more interested in the artistic side of collage medium, rather than the informational properties.

Recently, Diakopoulos and Essa [8] presented a system for creating a photo collage. In their system, the user selects a set of photographs and a template such as the one in Figure 2.4. The system will optimize the layout of the photographs based on the selected template. Figure 2.5 shows an example of the completed collage. Unlike what I am proposing, this system uses entire photographs, requires the user to select the photos to include, and limits the input size for the collage to that of the template. A similar collage layout program was developed by Wang et. al. [60].

Work at Microsoft Research Cambridge has lead to the "Digital Tapestry" [43] system. This approach uses saliency to identify the important features in an image. The salient regions are rendered together using a graph cut algorithm to minimize the energy (or difference) between different elements and create a reasonable looking composite. Their system is different from what I propose in several ways. While the collages do show elements, the elements can sometimes be cut off. This is because their method for collecting the images just uses saliency rather than an input labeling. This can be seen on the horse/waterfall in Figure 2.6. Second, their system does an importance sampling only by image epitome, so two very related subjects may appear different and be included. Finally, since there is no notion of labeling the images, if the user wants to design a thematic tapestry, the collection must be compiled by hand.

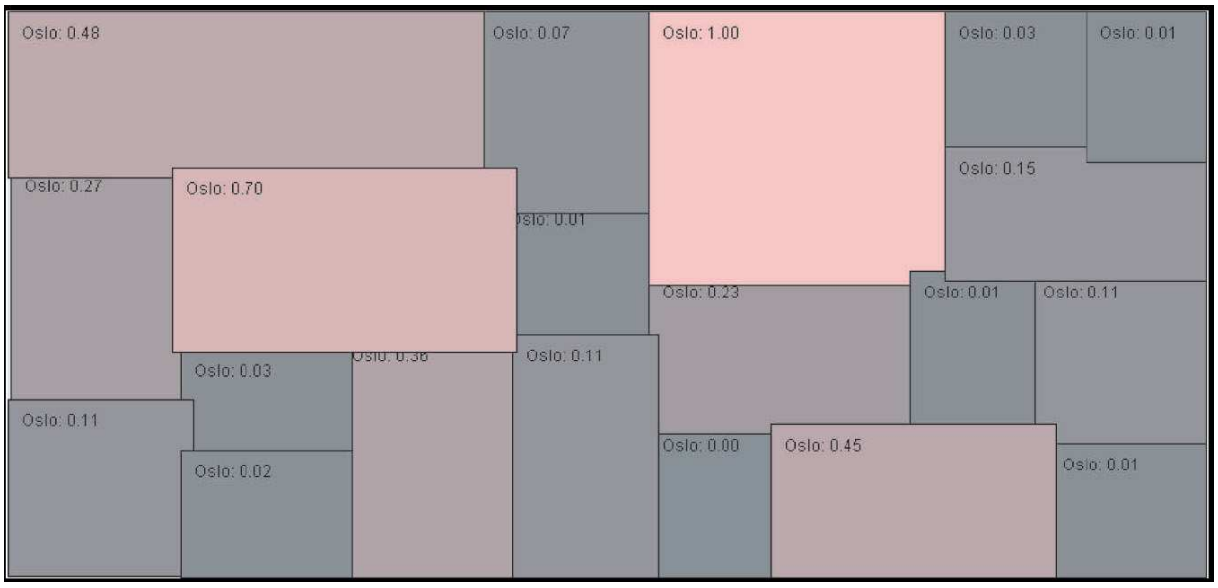

Figure 2.4 Collage template from [8]. The system runs an optimization to best place the selected images.

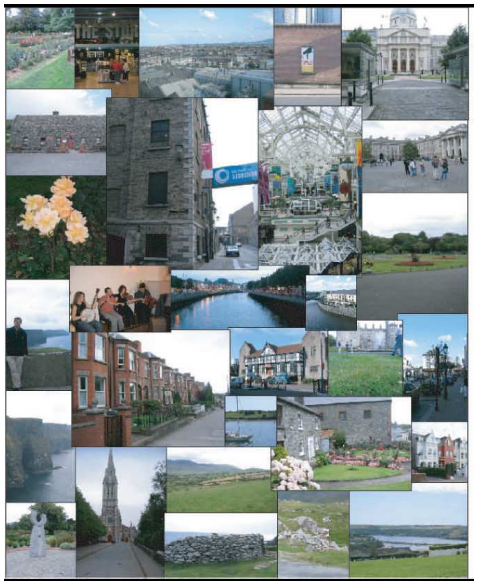

Figure 2.5 Collage from [8].

Figure 2.6 Digital Tapestry from [43].

Figure 2.7 Collage generated automatically in the Kodak Easy Share Gallery.

Companies have also begun to offer collage systems. Kodak Easy Share Gallery [7] allows users to upload photographs and create a collage. The system randomly lays out the images in an $n \times n$ grid. If there are not enough pictures to complete this grid (i.e. a perfect square), then some pictures are repeated. If a picture is not the correct aspect ratio, then it is cropped to fit. Figure 2.7 shows an example of a user-created collage. This program has several drawbacks. First, when the pictures are cropped, important information may be lost. This is seen in Figure 2.7 (bottom row/center picture) where the child's face is partially cropped. A similar failing is when the images are rotated and perturbed (for artistic purposes); these translations can cover important information in other photos; this is seen in top row/center picture of Figure 2.7. Again, in this system the entire photograph is shown rather than individual elements.

With the exception of Digital Tapestry [43], none of the collage based systems meet any of the requirements that I describe in Section 1.4. These systems do not scale, since they try to lay out every image provided, have no organization nor navigation. Digital Tapestry does scale well, since it looks for image epitomes to generate the collage. However, there is no built in organization to the images. Further, there is no natural navigation through the actual image collection. Adding a natural navigation to such a system is not trivial, since there are no well defined boundaries

between images or sets, the user would be unsure what parts of the tapestry leads where. Using my methods, users can navigate through the tree of photographs. Visual cues are provided as the user mouses over images, to help prevent the user from getting lost or confused.

2.5 Estimating Semantics from Low Level Cues

Much of the work in this thesis is based on the idea that high level semantic knowledge about an image can be approximated from low level cues. This includes the visual content of the images as well as the metadata that the camera records. This idea is often used in the area of Image Retargeting, as the goal is to find the important information in an image to make sure that it is retained when the image is altered to fit on a smaller screen.

Two notable efforts for retargeting images are by Suh et al. [55] and Liu and Gleicher [28]. Both of these systems only operate on single images rather than large sets. I show how similar techniques of extracting low level information can lead to an approximation of high level understanding.

Chapter 3

Automatic Photograph Clustering

The first method in this dissertation is for organizing photographs in a richer manner than a single time-line view. This is done by creating a hierarchy or tree of events that the photographs represent. Researchers [4, 15, 16, 29, 38, 54] have shown that photographs tend to be taken in clusters, or are "bursty." That is, several photographs are taken around events that the photographer wishes to capture. This cluster pattern can be exploited to automatically organize the photographs into broad groupings. The time that photographs are taken (and not taken) gives an approximation of the temporal boundaries around the events that the photographer wishes to capture. In this chapter, I explain the idea of the "Burst Pattern of Photography" and present my mechanism for automatically organizing large collections of photographs into a tree of related groups.

3.1 Burst Pattern of Photography

Photographs are often captured with a burst pattern and can be automatically grouped into subgroups by looking for these bursts. As researchers have pointed out, this behavior should not be unexpected; a photographer often tries to take multiple pictures of something of interest, either to capture the entire event as it unfolds, or to make sure to get at least one good picture of the subject. Researchers, myself included, have pointed out that this burst pattern exists at different levels of the time-line. At the same time that I was developing this work, Suh [54] came to the same conclusions as I present here. My findings match with those reported by Suh, implying that this theory is likely to be correct. This idea is also proposed in [16].

To confirm this idea, I have looked at approximately 40 different photo streams of time varying from several years to a few hours. In looking at the different streams of photographs, I have found that the burst pattern can be seen at any level in the time line. It is my conjecture that naturally captured photograph streams all convey this pattern. Consider, as an example, a stream which contains (among other events) a week long trip to Paris, France. Looking at the entire stream, there is a large cluster of photographs during the trip to Paris. If I were to "zoom-in" around that time frame, there would be different clusters around the different events of that trip, perhaps photographs from each museum visit. Again, zooming-in further around the time of the visit to the Louvre would reveal a cluster about each room, and further would show clusters around individual works of art. I have studied approximately 40 different photo streams that were "donated" for this research or publicly posted on photo sharing web sites. All of them had this burst pattern. The only time when I have not observed this pattern is when the camera is set to take a picture at a regular interval, such a web cam set to take a picture every minute.

I refer to this phenomenon as the "Burst Pattern of Photography." This implies that a photo stream is not necessarily a single line (or one dimensional time line), but rather can be structured in a tree. Each node in the tree represents an event that has taken place. From the Paris example above, the root of the tree is all of the events that the stream captures. The trip to Paris would be one of the children. Each subtree represents a sub-event of the parent. The visit to the Louvre is a child node of the Paris node and likewise a sub-event of the trip to Paris.

The study presented by [13] suggests that while most people would like to organize their personal photo collection (either printed or digital) they often do not find the time to be able to complete this task, unless there is some specific forcing function (such as an assignment). When dealing with large collections of digital photographs in general, Rodden and Wood present a study [41] which concludes that most photographers do not feel that the effort of organizing photographs pays off compared to the work required. Whether it is a lack of time, motivation, or a combination of the two, most peoples photographs have very little organization to them.

The tree based organization is one way that people would ultimately like to organize their photographs [13] if they had the time. Consider a printed album. A motivated person could create

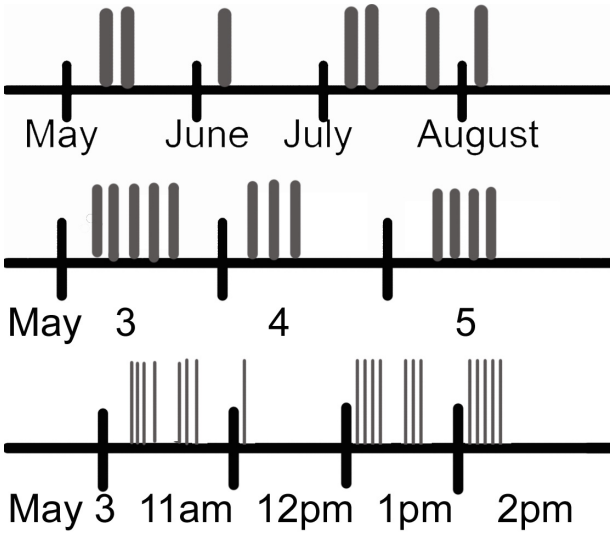

Figure 3.1 Example of photographs that may exist in a time-line, showing how photographs are taken in bursts, regardless of the "zoom" level of the time-line.

an album with not only pages, but also sections, subsections, etc. With properly designed dividers, browsing through such a photo album would be very similar to navigating the collection in the tree structure. The downside to organizing photographs in a tree is that it can be tedious and time consuming; just as building a complex physical photo album. In the next section, I present methods for automatically organizing the photographs into a multi-level tree. This provides the benefits of a tree (described above) without any of the costs associated with building such a structure.

3.2 Automatic Photo Tree Construction

In order to fully exploit the advantages of the Burst Pattern of Photography, the photographs can be organized into a tree structure. As explained above, doing this by hand is unattractive because of the time and effort involved in doing so. Using the metadata embedded in each picture, it is possible to automatically organize the photographs in a stream into a tree structure, where each node in the tree is an event, and the children of that node are sub-events.

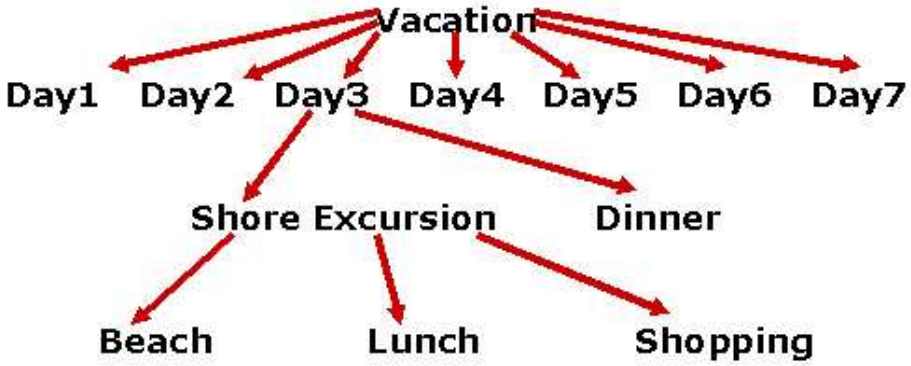

Figure 3.2 Example of a what a tree structure may look like based on the photographs of a one-week vacation.

Virtually every digital camera on the market today provides the time that the photograph was taken as part of the metadata information. This information alone is enough to be able to automatically build the photo tree. The only requirement is that the camera's clock remains relatively precise, so that the burst patterns can be detected within a stream. The clock does not need to be accurate since the capture time of the photographs will be compared against other photographs in the stream. This means that the methods will still work if the photographer does not change the camera's clock when traveling to different time zones; or even set it at all, provided that it keeps moving forward.

I use a single-link hierarchical clustering to organize the photographs. The first reason that I chose to use this method is because the photographs are naturally sorted by the time that they are taken. Single-link clustering exploits this fact and runs quickly. Second, the number of clusters is automatically determined. Third, the branching factor can be automatically determined based on the input given. Finally, the algorithm can be applied recursively, to automatically build the tree structure.

The following is a description of the algorithm to automatically cluster the photographs in a stream S:

1. Sort S by the time that each photograph was captured (starting with the first photograph taken). This is the default manner that photographs are stored on the card, as well as organized by most operating systems, so this step may usually be skipped.

2. Determine the average distance, in time, (t_{avg}) between each consecutive photograph in S. That is, determine the average amount of time between photograph captures. Mathematically it is found by:
   $t_{avg} = \frac{(T_2-T_1)+(T_3-T_2)+...+(T_z-T_{z-1})}{z}$,
   where T_i is the time the i^{th} picture was taken and z is the number of images in S.

3. Create a cluster boundary between any two consecutive photographs where the average time between them is greater than t times the average. In other words, if $T_i - T_{i-1} < t_{avg} \times t$ then there is a boundary between T_{i-1} and T_i and they are placed into separate clusters.

4. To build the tree, recursively perform steps 2 and 3 on each cluster. Stop when each photograph is in a cluster of its own, i.e. a leaf node, or the number of photographs left in the cluster is "small enough," depending on the application (i.e. each photograph in the cluster can fit on a display screen).[1]

Step 3 is where the algorithm finds the boundaries between photographs. The value for t in this step, was experimentally determined and set to 3. As an example, consider two consecutive photographs that are taken approximately 5 minutes apart. At a higher level of the time-line (or tree), if the average time between each photograph is 20 minutes, then those two photographs will be placed in the same cluster. In the recursive step (4), the cluster containing the two photographs will again be examined and a new average will be determined. If the average distance in this new cluster is 1 minute apart, then the photographs will be separated into distinct clusters in this new pass.

In using this algorithm, the assumption that is made is that the photo set follows the behavior of the Burst Pattern of Photography. More specifically that means there is an assumption that each photograph has a time stamp associated with it. Further, it is assumed that the photographs are related by being taken by the same photographer/camera or several photographers and cameras capturing the same event. If they are taken with different cameras, then it is assumed that the offset

[1] In my implementation, I use a cluster size of 20 as the default stopping size. This is because I have found that 20 images can comfortably fit on most computer screens. This number, however, should be altered depending on the application.

between the clocks of the different cameras is known, and can be adjusted to properly sort the photographs. In practice, this can be found by finding the difference of the time stamp between multiple photographs that are known to have been taken very close together in time, such as two pictures of the same subject/action. These assumptions preclude this algorithm from working with a set of pictures that have been collected from different sources at different times, such as collecting pictures from different web pages or an image search. It also will not work if the metadata associated with the different pictures is not intact; the most likely reasons for this is that the photograph was altered (e.g. resized) by a program that does not maintain the metadata. Since this method (as well as the others presented in this dissertation) are intended as a first pass organization of photographs, this scenario is unlikely.

Other researchers ([16, 54]) present their own methods for building a tree of photographs. Rather than use their methods, I chose to use the single-link clustering algorithm. The method that I use is very similar to that of Graham et al. [16]. The main difference is that at the top level of the tree, the distance between consecutive photographs in two different clusters is a hard coded variable. Although my method also requires a hard coded constant, it is a constant times the average distance between photographs. Hard coding the root level spacing may result in a wider tree, as this distance prevents overly large clusters, which may be appropriate over very large photo streams. The method used by Suh [54] requires fine tuning three different constants; again my method requires only one parameter to be set.

3.2.1 Efficiency of Clustering

The method presented here is extremely fast. Unsorted hierarchical clustering algorithms are $O(n^2)$ (for n being the number of objects being clustered together), since the distance between each pair of objects must be computed. For more information on clustering algorithms, please see [42]. In some cases, such as with photographs, the objects have a natural ordering and the time can be reduced by first sorting. Because photographs are naturally sorted, the sorting step can often be skipped. A notable exception would be when combining two or more streams together, but in this case a merge sort can be used. Computing a single node in the tree requires computing

the average distance between the photographs which takes $O(m)$ time (where m is the number of photographs in the node) and then finding the distance between each pair of photographs; thus computing a single node is $O(m)$. In this case if m is the number of photographs in the node and n is the total number of photographs in the set then $m \leq n$. Computing an entire level of the tree is $O(n)$.

As the tree gets wider, it must also become shallower. A wider tree means that there are more clusters (or nodes); and more nodes means that each cluster must have a small number of photographs. Less photographs means that there can be fewer children. The worst case for such a tree would be there were two bursts at every level where one burst contained 1 picture and the second burst contains the remaining photographs; in this rare instance the tree will have n levels, causing the time to build the tree to be $O(n^2)$. In practice, the tree tends to be much shallower than that (closer to, but not exactly $\log n$), reducing the build time significantly.

In practice, the tree can be computed dynamically, i.e. one level or one node at any one time, as requested by the user, rather than computing the entire tree at once. Again, this is an $O(n)$ operation. Computing the root of the tree (the largest node) from a stream of several hundred pictures takes less than a second. Each subsequent node will take less time to compute as there are fewer pictures in each node further down the tree. In practice I build each node as requested by the user.

3.3 Clustering Evaluation

To evaluate the clustering, I compared the results to those of a tree built by hand. The main difference was that the hand-built tree tended to stop clustering at a higher level than the automatic method. This is not necessarily a disadvantage or incorrect result. For example, one photo stream that I investigated contained a trip to a local zoo during the photographers vacation. The hand built tree stopped clustering after the zoo. However, the automatic algorithm created additional sub-groups. One group was all the pictures of bears (Figure 3.3), and another sub-group was the photographs from the area of the zoo devoted to African animals. In a similar example, all of the Van Gogh pictures which are all displayed in one room of the Musee d'Orsay were clustered

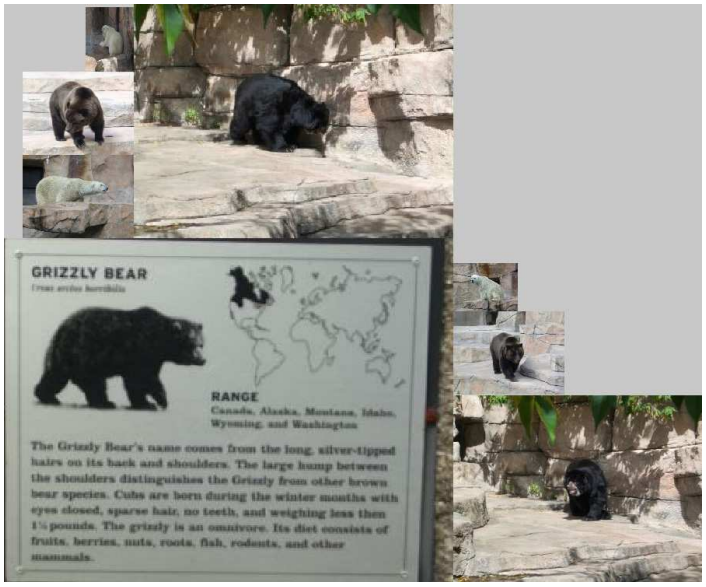

Figure 3.3 A cluster, in a collage layout, of photographs of the bears taken during a trip to a zoo.

together, this is shown in Figure 3.4. Both Figure 3.3 and 3.4 are laid out using a collage layout algorithm. Both of these figures are at the second to last level (just above individual photographs) of two different photo streams. Chapter 5 describes different layouts.

While the automatic clustering did perform very well, there was one example where the result was incorrect. In this case, the photographer was on vacation, and upon returning he found that his house was damaged in a storm. He immediately took pictures of the damage for insurance purposes. Since that particular stream spanned a very long time (3 years) those two events were clustered together as one at the top level of the tree. However, at the next level, the storm damage and vacation were separated into two different sub-events. Although time is a very strong cue, this example shows that it is not necessarily always enough; and additional information, such as image content or other camera metadata, may help further improve these results. This is shown in Figure 3.5.

It should be noted that although the adaptive clustering method requires a parameter to be set a priori, the t value in the adaptive clustering is not very brittle. Altering the value of t slightly does not significantly change the results of the clustering. Further, the value for t can be kept constant

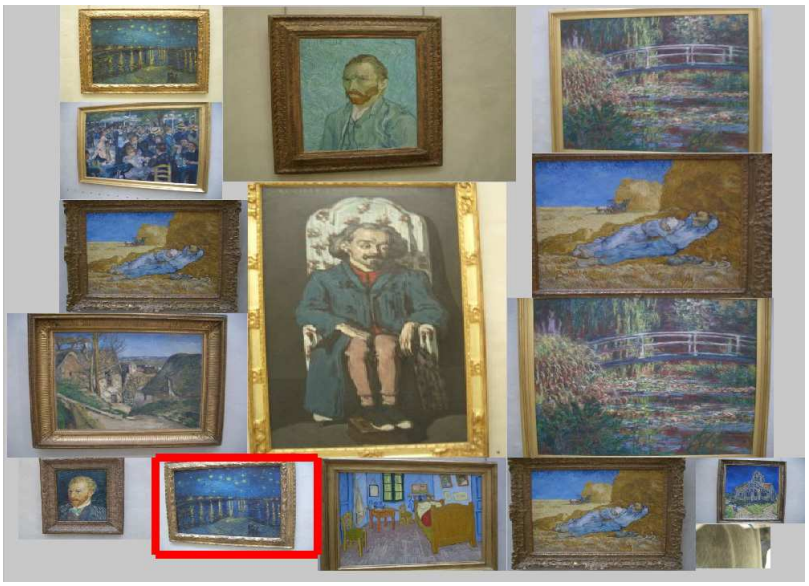

Figure 3.4 A cluster, in a collage layout, of photographs of Van Gogh paintings. These paintings are all displayed in the same room of the Musee d'Orsay in Paris France.

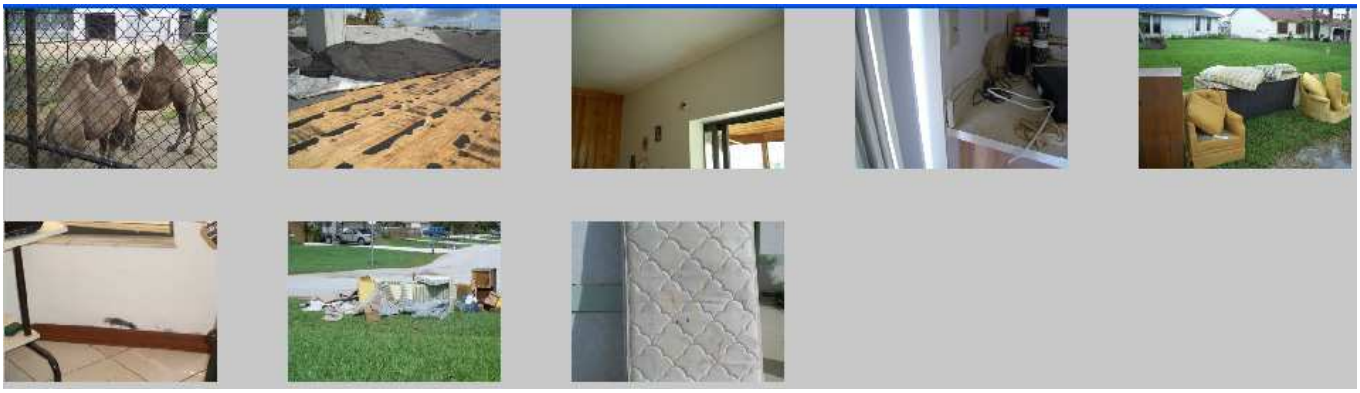

Figure 3.5 A cluster, in a grid layout, where vacation photos (the first photo) is clustered with photographs of storm damage that happened while the photographer was on vacation.

regardless of the photo streams or level of the tree. Photographs that are taken close together remain together, even with small changes of t.

Chapter 4

Selecting Representative Photographs

Most photograph organization methods, including those presented in this thesis, depend on the idea that a single photograph (or small number of photographs) can be used to represent the larger group. Selecting a single photograph to summarize a large collection can help by condensing the visual information that is presented to a viewer at any one time.

The Burst Pattern of Photography (Section 3.1) would seem to imply that this is correct. Photographs are representative of an event that has taken place. Intuitively, a single photograph from that set should contain enough context of the event to serve as representative of the entire event.

To date, however, there has been little, or no evidence that this intuition is correct. Since many methods (including the ones that I present) rely on the fact that a representative image is chosen it is very important to make sure that this happens robustly. If a non-representative image is selected, this can present false information to the user, giving the wrong (or no) idea about what the larger collection contains. This chapter first presents the results of a user study in which I test multiple commonly used automatic methods for selecting a single image from a set. Then I present the results of a talk aloud study, where participants are asked to select representative images. The results of these studies inform a new model for automatic photograph selection as well as provide a further comparison of how the different methods perform.

It should be noted that I am interested in "representative" photographs, and not "good" (or "best") photograph. These are subjective terms that vary depending on the viewer's mood, relationship with the photographs, and other intangible factors. I describe a "representative photograph" as a photograph that carries information to summarize the other photographs in the set; or could be applied as a label to an album (or box) containing the full set.

4.1 Standard Representative Selection Methods

There are many methods that are used in different tools for selecting a representative photograph from a set of images. Below, I list five common methods that are often employed as they are well defined, and relatively fast to compute.

First Photograph. The first photograph in the set is selected as being most representative. This is the method that is employed by web sites such as Flickr, and Windows operating system (when in "thumbnail mode"). This method is very simple to employ and is probably the most commonly used method today.

Middle Photograph. The photograph that is closest to the middle of the list of photographs when ordered by time. For example, if there are five photographs in the set, then the third photograph would be considered the one in the center. This method was first tried in AutoAlbum [38]. It was abandoned when this method selected an image pointing towards the ceiling in a set of pictures of people at a party.

Average Histogram. The average of all of the photograph histograms is computed. The photograph with the histogram that is closest (most similar) to the average is considered to be the most representative, since that image would have a color distribution closest to the "average" image. This is the method that was ultimately used in AutoAlbum when the middle photograph failed.

Image Contrast. Since the human eye is often drawn to contrast [21], the photograph with the most internal contrast is considered to be the most representative. This method itself has not been used in representative selection. However, since it is often used in other image processing tasks [28, 53, 54] I decided to test this method as well.

Appearance of Faces. The photograph with the most visible faces is considered to be representative. This is because the appearance of people can often carry information about who participated in the event. Currently this tends to be used more in research areas such as [54].

As face detection becomes more common, I believe that this method will be used more often in the future.

Rather than using a single method at a time, systems will often combine these methods together, such as in [16] and use a feature vector to determine the representative photograph. Again, however, there is only intuition given as to why the specific combination of methods works. I consider each method independently to gain a better understanding of which ones work. Later, when I develop my own representative selection method, I also use a feature vector. The main difference, however, is that the feature vector I use is based on the results of my studies on how humans perform the task of representative selection.

The list of representative selection methods that I presented are not exhaustive. The methods studied in this dissertation, with the exception of internal contrast, all have been used in other systems and can either be found in the metadata of the photograph or computed very quickly. Internal contrast was included because it is used in other image processing systems, and is also very quick to compute. All of these methods tested in this dissertation can be carried out very quickly even as the photograph sets continue to increase in size.

4.2 Testing Representative Methods

Each of the above listed methods have been used in various systems (research and commercial) in order to select the most representative image. Whenever any of these methods are employed, if any justification for its use is given it is always an intuition why it is correct. Strong evidence or proof is not provided.

In this section I present the results of a user study which shows that human selection does the best at finding the most representative image. While this does not prove that it is possible to use a single image as representative, it does provide evidence that it can be done. However, the automatic methods are not as robust as having an actual person select the image.

For my experiment I used twenty-one sets of twenty images each. Six of the 21 sets were donated explicitly for use in this research project. No one person donated more than two image

sets, so if a donor participated in the study, his or her familiarity with the photographs should have a minimal impact on the final results. The remaining fifteen sets were albums acquired from the Flickr web site, and are under a Creative Commons license, allowing for redistribution and modification of the original images. Only the first twenty images in each set was used in the experiment. For each set, six of the 20 images were selected as being potentially the most representative image in the set. The five methods listed above accounted for the first five selected images, the sixth image was selected by me (using "human knowledge") as being the most representative. In the case where the total number of images was less than 6 (either because there were no faces in the set or because one image was selected by multiple methods) then I used either a randomly selected image or my choice for least representative. In all, 17 of the 21 sets had at least one image with at least one face, 11 had a least representative image, and 9 had a random image. Each of the other methods were represented in all of the sets.

I invited participants to take part in the study over the World Wide Web. Initial invitations were sent to mailing lists for computer science and education graduate students at the University of Wisconsin-Madison. The invitation encouraged participants to forward the invitation to friends and family who they thought may be interested in participating. The human subjects approval prohibited collection of any demographic or geographic information about the participants. After agreeing to participate in the study, each user was shown a set of 20 image. After that they were shown the 6 candidate images (on the same screen) to select the one image that they felt was most representative. This was repeated a total of 21 times. The order of the sets and order of the candidate images were independently random for each volunteer. Incomplete surveys were not recorded. Volunteers were also given the opportunity to leave comments about their experience at the conclusion of the survey, however this information was separated from individual answers. In total 63 people completed the survey. Figure 4.1 shows a screen shot of a single image trial from our user study.

The hypothesis is that at least one method should out perform the others. If this is the case, then there is evidence that this method performs better than other methods to select a representative image. To test this, I performed a χ^2 test with a null hypothesis that each method should perform

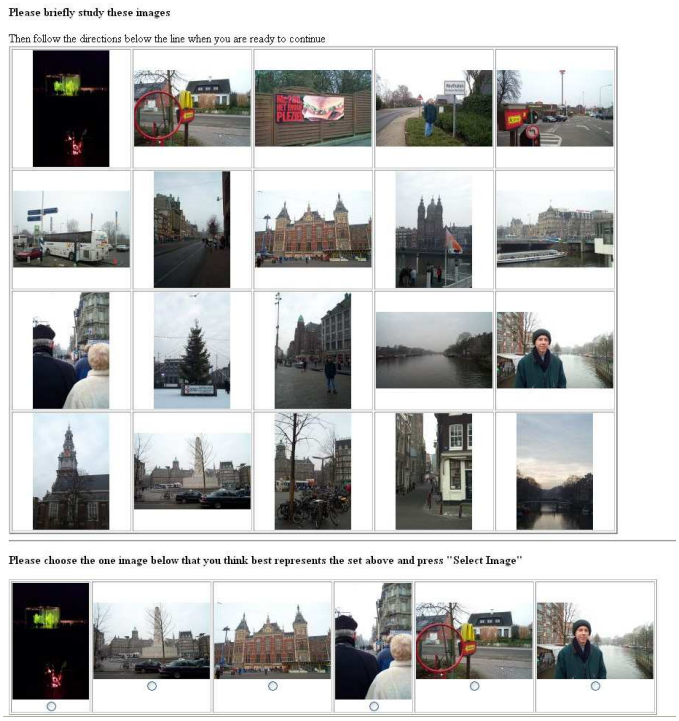

Figure 4.1 Screen shot of our user study.

Selection Method	Total Votes	Expected Vote
First Image	131	225.792
Middle Image	170	225.792
Average Histogram	218	225.792
Faces	194	184.32
Contrast	154	225.792
Least Representative	20	118.272
Random	107	96.768
Most Representative	542	225.792

Table 4.1 The total number of "votes" for each selection method and the expected number of votes.

with roughly the same results. Table 4.1 shows number of times an image of each method was selected and the expected selections, assuming that each method should perform equally. Faces, least representative, and random selection have a lower expectation than the other methods since they were not used in all 21 sets.

For the results, $\chi^2 = 602.752$ with 7 degrees of freedom. The P value is less than 0.0001. With extreme confidence we may reject the null hypothesis that all methods perform equally. The fact that the human selected image performs best is very telling. It implies that when selecting a single representative image, the simple methods do not perform as well as a human selected image.

The single selection design creates a masking effect that makes it difficult to infer either the absolute performance of the top choice, or much about the methods that were not chosen. However, the extremely large number of times the human-identified best images was chosen and the extremely low number of times the human-identified worst image was chosen supports the notion that humans can reliably make the best choice for representative and non-representative images. I address this problem by redesigning (although not implementing) the study in Appendix A.

The study alone does not prove that a single image can be used to represent the entire set. It is possible that the human selection is simply the "best of the worst;" and no representative image

actually exists. Such a scenario is unlikely, but possible; for example there may be an entire set of images where the lens cap is left on, or photographs are randomly gathered from the web (both cases that this dissertation does not address). However, in general, I believe that the results of the study, combined with the "Burst Pattern of Photography" implies that representative images do exist and humans are able to find them. Recall that the Burst Pattern of Photography is the photograph analog of Tobler's First Law of Geography [56]. Photographs that are close in time (i.e. a burst) are of the same general subject. Therefore, there should be one or more photographs that capture the subject and can represent the entire set of photographs.

A common comment among participants in our study was that for some sets, they would have chosen a different image that was not one of the six choices. Thus, participants had a different opinion of what the most representative image was from the experimenter. This suggests that there are multiple good answers. The implication is that finding one of this set of sufficiently good answers is sufficient for the selection process. This is further addressed in the next chapter.

The reliable existence of non-representative images has an important implication for implementations: bad choices exist, and should be avoided. Therefore, systems should avoid random or fixed index methods that may inadvertently select a bad choice. This refers to the first, middle and random selection methods. Rather some type of image processing is necessary when making the decision.

The results show that if it is possible for a single image to be representative, then humans are best at performing this task. While this study cannot confirm that existing algorithms cannot perform selection reliably, I feel that the data suggests that they do not. In the remainder of this chapter, I further address this problem, and create a formula for image selection based on how humans carry out this task as well as a new implementation for representative image selection.

4.3 Human Representative Photograph Selection Study

The first study I presented shows that humans can reliably select a representative photograph from a large collection. The results did not tell how it is that humans are able to perform this task. I now present the results of a follow-up study, in which I derive a formula for how humans select

representative images from a collection. In the following section, I show an implementation of this result.

4.3.1 Talk Aloud Study

The second study that I performed was a talk aloud study. Rather than trying to get a broad idea of how many participants behave, this study focuses in depth on a few participants, and how they reason about and solve a specific task. If multiple participants use the same method to solve a problem, then a broader conclusion can be drawn, even if the participants come to different results. This is true, especially for subjective decisions, such as selecting a representative photograph. A typical sample size for a talk aloud study is between three and seven participants. For more information about such studies see [26, 44].

There are both advantages and disadvantages to a study such as this one. The first disadvantage is that the sample size is small. If the population is very similar in behavior, but dissimilar to the overall population, then the results may be skewed. However, this is unlikely and small sample sizes are generally used in this style of study [26]. The next disadvantage is that the study tends to a take a longer time for each participant to complete. Unlike the previous web study that many participants could complete in a short amount of time without any interaction, only a few people could complete the study in a much longer amount of time. This also makes it more difficult to find participants. Finally, there is a certain amount of self-consciousness that is normal in this type of study, since the participants are being observed and recorded. This may cause the participant to not fully vocalize their thoughts or be more concerned in reporting answers that they believe are desired rather than what he or she actually thinks.

Despite these disadvantages, the talk aloud study has several advantages and the disadvantages can be overcome. The main advantage to the study is that it provides an in depth look at how a participant carries out the task in question. If the facilitator notices that the participant is beginning to act self-conscious then he can offer encouragement to the participant to get him or her back on track. Finally the design of my study provided both qualitative data (the details of how the participants make the selection) and quantitative data (the actual selections) to offer further insight

into the problem. Since I was trying to understand how humans perform representative selection, I decided that the advantages of a talk aloud study outweighed the disadvantages. Again, this is because the study provides the most details about how humans perform this task.

For my study, five participants (three males and two females, computer science students) were asked to look at multiple sets of images and mark those that they found to be representative, and those that they found to be non-representative. The photograph names in each set were listed on a sheet of paper where the participant was able to mark any image, along with room for a small comment justifying his or her decision. Additionally, each participant was video taped in order to record their utterances, as he or she described his or her thought process. At the end of the study each participant was asked to briefly summarize his or her selection strategy.

Each photograph set was shown to the participants individually. The photographs were displayed in Windows Thumbnail mode, ordered by the time the photographs were taken. The participants were free to display any (or all) photographs using the IrfanView photo viewer. Some of the participants viewed the photographs using IrfanView to display full screen sized images. Other participants would only display those images that were of interest to them.

In total, the participants were asked to work with six different sets, varying in size from 88 images to 25. Additionally, there were four more sets that were subsets of one of the original 6 sets. All of the sets came from one of two photograph streams, however other than the three related sets, each set was separated significantly in time so that the context of each set was different.

4.3.2 Qualitative Results of Study

Overall, the agreement between the participants about which photographs were representative and which ones were not representative varied greatly. A few photographs were marked as both representative by one participant, and not representative by another. The subjective nature of the question, along with the different life experiences of different participants can account for this disparity between answers. Figure 4.2 shows one such picture that was marked as being representative by one participant but not representative by another one. The picture comes from a family vacation on a cruise ship. The participant who marked the picture as being representative has been on the

same cruise line within the part year, and recognized the decorations as being very specific to the particular cruise line. To him, it was a perfect example of a family having fun on a cruise vacation. The participant who marked the photograph as being non-representative was never on that cruise line. He said that the picture, while it does show the participants, does not really give any context about the fact that the family is on a boat.

Despite the variance among results from the participants, the methods employed for making the selection were remarkably similar. Each participant first looked through the entire set to try and figure out what was happening in the set, i.e. give an overall theme or context to the selected set. Any photographs that clearly did not fit within that theme were immediately removed from consideration and marked as being non-representative. Next a participant would search for faces. When a participant was questioned about this tactic he responded that knowing who was part of the event is important. Some participants went as far as too look for multiple occurrences of the same person, and give a stronger emphasis to photographs with the same people. Finally participants looked for images that were aesthetically pleasing, such as properly taken, in focus, etc.

Based on the results of the study and discussions with the different participants, I developed a formula for scoring photographs as being representative, modeling the behavior of humans. For any given set, the photograph with the highest score is chosen as the representative photograph. If more than one photograph is desired, then the set should be further divided (see Chapter 3), and a photograph from each sub-cluster can be chosen. The following is the formula that I developed:

$$R_i = \alpha \times C(P_i, S) + \beta \times F(P_i) + \gamma \times I(P_i) \qquad [4.1]$$

Where P_i is the i^{th} photograph in set S, C is a function that returns the numerical score of context of P_i relative to the set, F returns a score of the people in the image, I is a function that measures the interestingness of the image, and α, β, and γ are each normalization and weighting constants to adjust the relative importance of each measure. R_i is the score of P_i The influence of context should be greater than the influence of faces which should be greater than the influence of aesthetics. I discuss this formula (and a practical implementation) in Section 4.5.

For the given formula, the representative photograph, P_r, in set S is simply given as:

Figure 4.2 Example of an ambiguous photograph. It was marked both as representative and non-representative by different participants.

$$R = \max(R_i \in S) \quad [4.2]$$

Alternatively, we can say that P_r is all photographs in the set that are greater than some value.

This formula is designed to model and approximate human behavior for selecting representative photographs. The participants did not actually assign scores and mathematically determine representative values, at least not cognitively. Further, the values of the weighting parameters and way that the functions (context, faces, interestingness) varied by participant, based on personal preference.

4.4 Comparing Human and Automatic Selection Methods

In addition to the qualitative data, the study also provided a plethora of quantitative data. This data gives a further means of comparing the automatic selection methods from the original user study against how a human performs. Each participant gave each image a value of being representative, non-representative or neither (the image does not stand out in any way). The automatic selection methods are able to make a single choice from each set. The "goal" for the selection method is to select an image that was marked as being representative or at least avoid those images that were marked as being non-representative.

Within each image group, there are sets of images that are similar. These are images that convey the same information, even if one may be considered, by some metric, better than another. For example, this may include images of the same scene but different camera orientation (landscape or portrait), different camera settings (flash or no flash), or different combination of people in the scene (all of the men in the group or all of the woman in the group). Although the instructions were to select all of the images that a participant considered to be representative, sometimes a participant would only select one of these images

Often the participants in the study would select a single image from such a set, most likely in order to save time. To accommodate for this, the score for all equivalent images were summed together for the following analysis. If a single participant marked multiple images in an equivalent group, then the votes only counted once.

Equivalent classes were determined by hand. In order for two images to be considered equivalent, they had to meet several criteria. First, images had to be consecutive. Next they had to be taken within two minutes of each other. The images have to be clearly of the same scene. Finally each image must also contain the same participants. If any of these criteria were not met, then the images were not grouped together.

The different sized photo sets and probabilities makes standard statistical tests either difficult (or impossible) to perform and/or give less accurate results, since the test data does not meet the standard requirements. Since random simulation makes no presumptions about the data, I choose to use that technique to analyze the results. The random simulation tests can only show if a method is performing effectively statistically randomly. It, however cannot be used to make direct comparisons of one method against another. Despite these limitations, however, the results are very telling.

Using the quantitative data provided through the study, I will test each of the methods to determine the likelihood that each method performs better than random chance when selecting representative images. For this test, I consider an image to be representative if two or more participants marked it as such since this indicates that there is agreement between participants. Likewise, a non-representative image is one in which two or more participants marked as being non-representative. Although there were images that were marked by different participants as being both representative and non-representative, there were no images that could be considered both by the above definition.

For each of the six sets[1], the probabilities were determined that a representative or non-representative image would be selected at random. These probabilities are given below in Table 4.2. I then ran a simulation of selecting images based on the given probability of selecting a representative (and non-representative) image 100,000 times. The value of each test is given by Equation 4.3. There are 64 possible values that any trial can take on. The number of occurrences of each value was represented as a histogram.

[1] In these tests, I do not consider the four subsets, to avoid some photographs getting more influence than the others

Set Number	Representative Image	Non-Representative Image
Set 1	$\frac{4}{85}$ (4.7%)	$\frac{1}{85}$ (1.18%)
Set 2	$\frac{6}{37}$ (16.22%)	$\frac{6}{37}$ (16.22%)
Set 3	$\frac{5}{88}$ (5.68%)	$\frac{2}{88}$ (2.27%)
Set 4	$\frac{5}{25}$ (20.00%)	$\frac{2}{25}$ (8.00%)
Set 5	$\frac{3}{25}$ (12.00%)	$\frac{4}{25}$ (16.00%)
Set 6	$\frac{12}{71}$ (16.90%)	$\frac{4}{71}$ (5.63%)

Table 4.2 The probability mass (or likelihood) that each selection method performs as random chance.

$$\sum_{i=1}^{n} \frac{1}{p_i} \times x_i \sim p_i \qquad [4.3]$$

Each of the tests were re-run and a value (t) was found using Equation 4.3. The area under the curve of the histogram to the right of t gives the probability mass that each method performs like random selection. The smaller the probability mass, the less likely it is that the method is no better than random selection. A good method should have a low probability mass for finding representative images, and a high probability mass for finding non-representative images. Table 4.3 displays the probability mass for several automatic selection methods.

At first glance, it may appear that using the first image in the set is the best method for selecting a representative image, since it has a "representative" probability mass of only 0.8%. However, the "non-representative" probability mass is still low enough (4.6%) to be considered statistically significant. These results can be explained in one of two ways. Either the first image in the set will tend to contain a very good (representative) or very bad (non-representative) image. The other explanation is that participants felt compelled to rate the first image in a set, since it is the first one that was viewed in each test. Either way, being the first image alone does not seem reasonable grounds for selecting it as representative.

In the test there were two other temporal position based methods: middle image and 10^{th} image in the set. These were tested to see if there is likely to be any position (besides first) that

Selection Method	P.M. Representative	P.M. Non-Representative
First Image	0.8%	4.6%
Middle Image	10.945%	15.883%
10^{th} Image	100%	100%
Closest to Average Histogram	54.98%	100%
Furthest from Average Histogram	100%	28.883%
Contrast	54.98%	0.46%
Faces (Computer)	2.14%	100%
Random Sample	35.145%	100%

Table 4.3 The probability mass (or likelihood) that each selection method performs as random chance.

may be reasonably used for selecting representative images. The results show that neither method can statistically outperform random selection. There appears to be a large difference between the middle and 10^{th} image in the set. Despite this difference, however, statistically both methods performed with the same results as with random selection. The reason for this difference is that most of the random trials resulted in a score of 0 (not picking a representative or non-representative image), or a probability mass of 100%. With such a small trial set (6 sets of photographs) there is a large difference of probability mass between picking a single representative image (or non-representative image) and not picking any representative images. With a larger trail set, I believe that the 10^{th} and middle image would have a closer similarity. It should be further noted that the results show that selecting the middle image is equally likely to result in a bad image as it is in a good image. This result was confirmed by [38]. In that work, the middle image was originally taken as the representative image. However, they discovered a case where the middle image was pointing towards the ceiling in a set of pictures from a party, where most pictures had faces, decorations, and other cues to indicate that the photographs where taken at a party. This led them to abandon the middle image as the representative image in the set.

Similarly, the histogram based methods do not seem to outperform random selection either. Intuitively, an image that is close to the average histogram of all the images, should be representative since its color distribution is close to the average color distribution of all of the images. However, in practice, this metric only holds up for very small sets of images (roughly less than 10). This makes the histogram an undesirable choice for selecting representative images.

Taking the image with the largest amount of internal contrast may lead to non-representative images. While the human eye is drawn to contrast (or high-frequencies), this metric alone does not imply that an image will be representative. For example, Figure 4.4 shows an example of a chalkboard filled with writing. Although this photograph contains more internal contrast than any other image, it was marked as non-representative by most of the participants.

The appearance of faces satisfies both metrics of having a low representative probability mass, and high non-representative probability mass. This would make it an ideal candidate for automatic representative selection. However, there are some problems with this approach. Most importantly,

there is no guarantee that faces will appear in any given set. When this happens, there needs to be another mechanism for making a selection. Also, faces alone do not always convey enough information; for example, too many faces may block the context of the scene and do not reliably represent the set.

4.4.1 Representativeness at Multiple Levels

In addition to the six sets of images, I also tested 4 subsets. In these tests, the participants were given the same task: selecting representative and non-representative images. Every image that a participant marked as being representative in a set was also marked as being representative in the subset. This implies that given sets of images S and S' such that $S' \subset S$ and an image $i \in S', S$. If i is representative of S then i is also representative of S'. However, i being representative of S' does not necessarily mean that i is representative of S.

4.5 Implementation of Representative Selection

The main components of Equation 4.2 are: Context, Faces, and Aesthetics. A good automatic representative selection method should try to take each of these properties into account when making an image selection. Each of these aspects, however, are subjective terms that can mean different things to different people. In fact, this explains why the participants each came to different results, despite using the same approach. Since there is no defined way of deciding each of these metrics, I developed approximations of each in order to perform representative selection. In the following sections, an algorithm is described that scores an image based on these metrics.

Since the high level information that Equation 4.2 requires is not automatically attainable, it is necessary to approximate such information with simple, easily attainable, low-level cues. The implementation for each of the metrics can be carried out very quickly, or even implemented on a cameras hardware directly. As technology and visual understanding methods improve, the methods presented here can be replaced with newer, more accurate approximations. Approximating high-level information using low-level cues is often done in computer vision and multimedia tasks of this sort [28, 53]

4.5.1 Approximating Context

Participants in the study always started by looking for the general context of the photo set. In other words, they were trying to figure out "what is happening" or "what story is being told by these photographs?" Humans are very good at determining context from the set of images, and can quickly identify the outliers, or those photographs that do not match with the theme of the rest. While several specific purpose object detectors exist [23], computer vision technology in general does not give a way of determining a general context of a set of images.

Some participants in the study initially started looking for images that contained text, i.e. signs that would be useful in identifying where the images were captured and what was happening. While this seems to be a reasonable approach, several of the pictures in the example set had signs that were "cute" but did not carry any useful information, and in fact can detract from understanding the context. One participant pointed this out and said that he would avoid images with signs for just that reason. Figure 4.3 shows an example of an image where the written information on the sign detracts rather than provides context. Many of the participants marked that image as being non-representative for this reason. Additionally, images with text alone do not guarantee enough information to be representative on their own. Figure 4.4 shows an image taken of a chalkboard, which would not be representative of the entire set. Although there are reliable techniques for finding unconstrained text within an image, I do not rely on the appearance of signs, as they do not offer a strong enough guarantee that the image contains enough information to represent the context of the photo set.

The color histograms of the photographs seem as though they should be able to approximate the context. Photographs that are similar in context should have a similar color scheme. Likewise, photographs that are different from the rest of the set will likely have a different color distribution. The histogram is often included as part of the photograph metadata, so it can be accessed without having to load the photograph into memory. Even if the metadata does not include histogram information it can still be computed very quickly. This is a similar idea to that used by AutoAlbum [38] for making the selection of a representative image.

Figure 4.3 Example of a photograph with a sign point that on its own detracts, rather than provides information. The sign lists many cities, states and countries that have nothing to do with the context of the overall set.

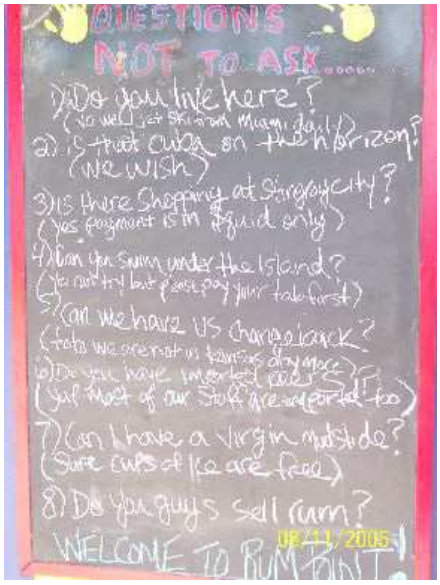

Figure 4.4 Example of a chalkboard with a lot of writing and internal contrast. However, this photograph is not representative of the set it is in.

If a photograph is very different from the others in the set, then its color distribution is likely to be different as well. This outlier should not greatly affect the average histogram, and it should result in a large C and is likely related to the context.

Unfortunately, the analysis of the study shows that the histogram is not likely to perform much better than random chance (Table 4.3). As the photo set grows, the histogram becomes a less reliable indicator of context. This is likely to be caused by two factors. First, the photographer may change various camera settings when capturing the same picture. This will cause the color distribution to change for the same scene. Second, as the photo set grows, the histogram tends more towards a uniform distribution.

In order to compute the context score of the image I rely on metadata of the set rather than the image contents. For any given set of photographs, I subdivide the set based on the time taken (Chapter 3). The context score is then given to the entire cluster, based on the number of photographs in the cluster rather than individual photographs. This is based on the idea that the more important something is, the more photos will be taken of it. The context score of an image i is given as follows:

$$C_i = \begin{cases} |S'|, & |S'| < 3 \\ 4, & 3 < |S'| \geq 20 \\ 5, & 20 < |S'| \geq 50 \\ 6, & |S'| \geq 51 \end{cases}, i \in S' \qquad [4.4]$$

In Equation 4.4, S' refers to a cluster of photographs that contains image i. The length of S' is given by $|S'|$. The values and cut off points were experimentally determined. They were chosen so that sets that are only slightly larger are not given a strong extra preference. Minor changes to these values should not greatly affect the final performance of the algorithm.

This method of approximating context cannot find the representative image alone. In fact, the highest scores go to the most number of images. Rather, it helps give a range of where the most representative image may be located. The other metrics will be key in determining which image to select. It is possible for an image to have a low context score, but still be the representative image

in the set. Again, this is based on the assumption that the photographer is taking many pictures around the event that is of interest.

Since the context score is dependent on the set dynamics, the size of the set in particular, it must be computed at run time. However, this score only needs to be computed once per set, rather than once per image. Further, the process is very fast so it does not add any noticeable computation time to the process.

4.5.2 Approximating Faces

In the quantitative analysis of the study data against different selection methods, faces did the best. Fortunately, faces are perhaps the easiest of the three metrics to automatically measure. However, the participants in the talk aloud study each approached this task in different ways. Some participants simply counted the number of faces in each image, or picked images that seemed to show a lot of people. Others looked for the same faces repeatedly so that images containing the same person could be given a higher weight. Participants also commented that they did not know who the important people in the set were, and if they did, that may have influenced their decision.

For the most part, face detection is a solved problem in computer vision [34]. There are many algorithms which take a photograph as input and return rectangles indicating the location of faces. Face recognition, on the other hand, is a more difficult task. Computer vision algorithms are continually getting better at it, but they are far from perfect. Further, while a face recognition algorithm may be able to find if the same face appears in multiple images, it cannot determine (automatically) which faces are important. This level of sophistication requires some amount of training by someone familiar with the photo set.

In this application, I used the Intel Open Source Computer Vision Library (OpenCV) implementation of face detection, which uses Principle Component Analysis [24] to find faces. The algorithm takes a single image and returns a list of rectangles enclosing each face in the image. For a given picture in the set, the face score is given by:

$$F_i = |f(P_i)| \qquad [4.5]$$

In the above equation, the face score, F_i is equal to the size of the set of rectangles returned from the face detection (f) for photograph P_i.

Unlike the context of the image, the faces score of each image does not change relative to the set. This score can be computed once per image and can be computed off-line. While face detection in general is relatively fast (and can be implemented in hardware), the OpenCV implementation can take a few seconds on a large image. When dealing with several hundred images, being able to perform this operation off-line is very useful. Additionally, performing this operation off-line allows for the introduction of more computationally-expensive operations (such as face recognition) in the future.

4.5.3 Approximating Aesthetics

Figuring out if an image is aesthetically pleasing is an on-going research topic in computer science and psychology. A true and full understanding is outside of the scope of this dissertation. Rather, as with the other metrics, I approximate the aesthetics of an image using simple cues.

The human eye is finely tuned to detecting contrast, and it is one of the most low-level visual responses [21]. I exploit this human attribute in the approximation of aesthetics. An image with high contrast is very likely to have something interesting or aesthetically pleasing happening, or at the very least attract the viewer's attention. An image with little or no contrast is likely to have been poorly captured: taken out of focus, over/under exposed, etc.

In order to score a photograph, I use the method presented in [31]. This method creates a bitmap of the image marking the pixels with high contrast. The aesthetic score is given by the percentage of the image covered by high contrast. The following equation shows how the aesthetic score is computed for a given image.

$$A_i = \frac{\sum_{p=0}^{w} \sum_{q=0}^{h} (K_{p,q})}{w \times h} \qquad [4.6]$$

In the above equation K is the binary mask of contrast in the image, w is the width, and h is the height of the image.

This method has two advantages. First, it gives a good approximation of the visual interest of any given photograph. At the same time, photographs that are poorly taken (e.g. out of focus, over-exposed, etc.) will be scored low, as these images will not have high contrast. This serves as a means of removing such undesirable images from consideration as being most representative.

The study indicates that contrast alone does not provide a reliable method for selecting representative images. For this reason, the overall aesthetic contribution is small relative to the other two metrics. However, the contribution should be enough to ensure that well taken photographs are given more importance than a poorly taken photograph.

As with the face score, the aesthetic score is independent of the other images in the set. It can therefore be computed once for each image, in an off-line setting. In my implementation, the face and aesthetics scores are computed the first time an image is encountered and stored for later use.

4.6 Automatically Selecting a Representative Image

Using the approximations described above, a system can be built for automatically selecting a single representative image. As previously stated, the face and aesthetics score is computed once, the first time each image is encountered. The context score changes with each photograph, relative to the other photographs in a set.

Recall Equation 4.1 describes how humans perform representative image selection. Based on the approximations of each metric in 4.1, the equation can be rewritten as follows:

$$P_i' = \alpha \times C_i + \beta \times F_i + \gamma \times A_i \qquad [4.7]$$

Essentially the subjective values of context, faces, and aesthetics are replaced with the approximations. Again, the values α, β, and γ are used to weight and normalize each of the three components of the formula. In my implementation, α and β (the scalars for context and faces respectively) are both taken to be 1. This states that a photograph in a large set with lots of people will have a higher score than a photograph in a small set with few or no people. The value of γ scales the percentage of the image that is covered by contrast. It was experimentally determined to be 5; in other words, an image that is 10% covered in high contrast pixels will have an aesthetics

Selection Method	P.M. Representative	P.M. Non-Representative
First Image	0.8%	4.6%
Faces (Computer)	2.14%	100%
New Method	**0.187%**	**100%**

Table 4.4 The performance of First Image in the Set, Face Detection (the two highest performing methods shown in Table 4.3), and the new method presented above.

score of 0.5. This number was chosen to be small, so that the overall contribution of the aesthetics score does not dominate the other two scores, as the studies showed that it is the least important of the three metrics. In practice, the aesthetics score often acts as a tie-breaking vote.

Again, the representative image (P'_r) in a set (S) is selected as being the image with the highest overall score. Equation 4.2 can be rewritten as:

$$P'_r = \max(P'_i \in S) \quad [4.8]$$

If more than one representative image is desired, then the set should first be divided as described in Section 3.2 and then the representative selection performed on the smaller subsets.

4.7 Representative Selection Evaluation

Selecting a representative image is highly subjective. The participants in the study often disagreed depending on personal knowledge, experience and tastes. As a result, this makes formally evaluating any type of automatic representative selection method difficult.

I first evaluate this new representative image selection as compared with the previous methods. Table 4.4 shows how the automatic selection method fairs, compared with the first image in the set, the image with the most faces, and my new method. Recall that for a method to perform well it should have a low probability mass for representative selection (i.e. does not act like random chance) and a high mass for non-representative selection (i.e. does act closer to random chance).

From the results in Section 4.4, the method to try and out perform is simply finding faces. The results imply that my new method works better than finding faces alone. However, there is not enough of a statistical difference to make a strong claim to that effect. The main difference between my method and simply relying on face detection is that although my method does incorporate face detection, it also uses other information to fine tune the selection. Further, my method will also work when there are no faces present in the set.

The setup of Equations 4.7 and 4.8 ensures that photographs with common "mistakes," such as being out of focus, will be avoided. I have not seen any instance of a poorly taken image selected as representative. The methods proposed do seem to fail, however, in the case of a picture where there are so many people that they block the context of the image. This is because the face score may dominate the other scores. Figure 4.5 shows one such example. The photo set was of several people swimming with stingrays. However, because there are many people in this image, the face score drives the total score of the image up. Such a selection is not entirely incorrect, as it does convey information about who was participating in the event; however it does not convey information about what is happening to someone who is not familiar with or did not participate in the event.

Figures 4.6 through 4.9 shows results of the representative selection technique. Justification for each selection is provided in the caption.

4.8 Summary

Many applications try to use a single image or multiple images for representative image selection. The methods that I present are no different. However, little justification has been given for why different methods are used; only an intuition why the method is used.

I have tested several commonly used representative selection methods. Although the methods will often be combined as a feature vector, I tested each method separately in order to gauge how well each method works independently. The first user study shows that humans do a better job than any simple method, at selecting a representative image. This implies that a single method alone cannot reliably find a representative image. The second user study gave an idea of how humans

Figure 4.5 Example of a poor automatic selection. While several participants are shown, there is very little context of the overall set.

Figure 4.6 (Left) Subset of images from a photograph stream. (Right) Image that was automatically selected. This image shows several people, as well as sky and water background.

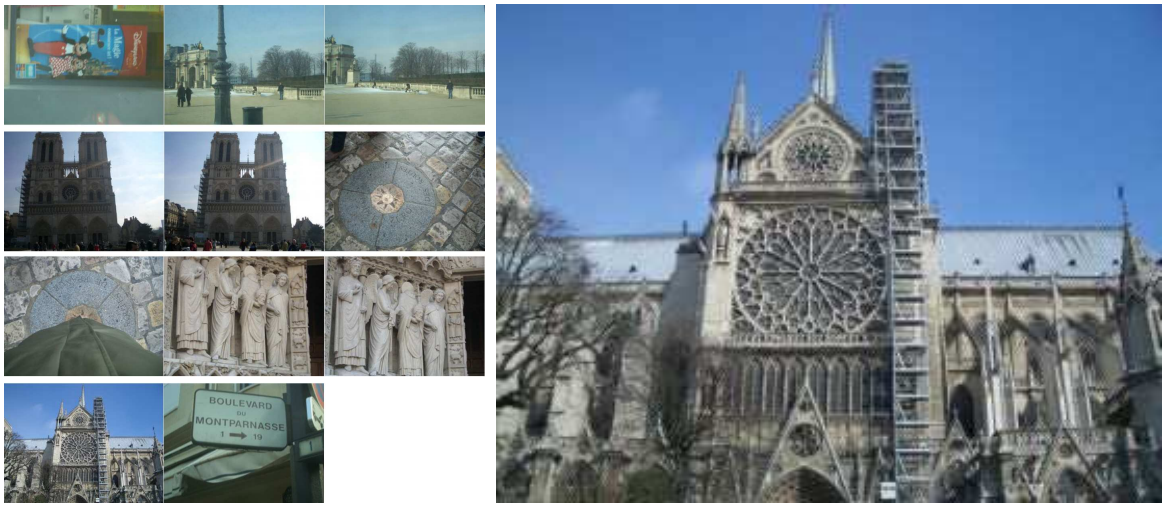

Figure 4.7 (Left) Subset of images from a photograph stream. (Right) Image that was automatically selected. The entire set was taken around Notre Dame in Paris, France. The picture selected is one of the chapel, which has more contrast than those taken of the ground ("point zero").

Figure 4.8 (Left) Subset of images from a photograph stream. (Right) Image that was automatically selected. The set was taken around San Francisco, CA and more specifically the Golden Gate bridge. This photograph has two faces and contrast of the red bridge against the natural background.

Figure 4.9 (Left) Subset of images from a photograph stream. (Right) Image that was automatically selected. This image shows the boat trip that the set was capturing. Two boats where approaching each other, which is what was being captured.

perform this task. It also provided data to retest the methods that I explored in the first study. Other than face detection, the methods do not perform any better than random chance. Since there are few non-representative images in a set, random selection may do a reasonable job most of the time, but there is no guarantee that a bad image will not be selected.

In order to approximate human behavior, a method should be a combination of context, faces and aesthetics. Different low level cues can be used to approximate each of these metrics. Using the idea, I presented a new method for automatic representative image selection.

Chapter 5

Photograph Layout

When displaying images, many programs will show all of the images in the set or folder, in a standard grid layout. Displaying all of the images at one time may overwhelm the viewer. Rather than displaying all of the images at once, I propose using the methods previously described to create a simpler layout that will still convey the meaning of the photograph set.

The methods presented thus far in this dissertation can be combined to provide a new interface model for employing layouts. The photographs are clustered into a tree structure. Rather than displaying the entire tree, a representative sample from the root node can be displayed, taking one photograph from each child of the root. This reduces the total number of images that need to be displayed at any one time. In theory, if the representative image was well chosen, then there should be no reduction of visual information. In practice, only a small amount of visual information is actually lost. This is different from most existing photograph applications in that the existing applications will either show all of the photographs, which may overwhelm the viewer; or use a different selection method, such as first image in the set; which may not contain as much information as the representative selection method that I describe. In the next chapter (6), I describe how the layouts can be used as a navigation tool to rapidly browse through all of the photographs.

I propose modifying the way that existing layout methods are used in order to take advantage of the methods that I present in this dissertation. Virtually any existing layout can be altered to make use of this new model. To demonstrate this, I have implemented four different types of layouts. The first two are temporal based: a grid and a time line layout. The other two are collage-based layouts. Each layout serves a different informational and aesthetic purpose. Again,

these are only four layouts, to show how the methods I present can be combined with different layouts. From these examples, it should be possible to see how other layouts can be modified in a similar manner.

We are constantly bombarded with more information than can be displayed on a given media. Photographs are only one example of this phenomena. Other examples of such visualization of large collections of data include documents [37] or information passively collected throughout the day [11]. Although there is a large literature on information visualization techniques [3, 51, 61], I have focused my efforts on using layouts that are common within photograph systems. In Section 1.4, I listed several different requirements for a successful system. One of those requirements is that it must include a simple and understandable navigation system. While it should be possible to modify techniques proposed for visualization, I chose to use traditional photograph layout mechanisms as these are familiar to users. This should definitely meet the requirement of an understandable interface without requiring the user to learn a new interface model.

5.1 Existing Layout Mechanisms

I use standard layout mechanisms, augmented with the methods that I present in this dissertation. In this section I describe four standard methods and show how they can be altered to make use of the methods that I present in this thesis. The main idea is that each grouping of images is abstracted by a single image, thus reducing the visual complexity of the entire set. This idea is similar to [43], in that the total amount of visual information is greatly reduced.

A major difference between my implementation and [43] is that using my methods allows a means of moving throughout the photo tree. Rother et. al, provides one visual summary of the photo collection; instead I propose creating a new layout for each node in the tree. The representative photographs that are displayed at each level of the tree also serve as a gateway to the next level.

Below, I describe four standard layout algorithms that I have augmented using my methods. I show how the layouts can be used in conjunction with my new methods and when each one would

Figure 5.1 A standard grid layout.

be useful. It is important to stress that I am not creating new layout methods, but rather showing how existing ones can be improved with these methods.

5.1.1 Grid Layout

In a grid layout, the photographs are organized into a simple grid. Virtually all photo programs and file systems offer this style of layout. Such a layout is useful as it is both simple to implement and simple for the viewer to understand. In a grid layout, there is an implied ordering of the images, which makes it easier for the viewer to "read."

In my implementation, one image from each event group is taken. Those images are placed in the order that they are taken. I have found that this method is useful when trying to find a specific photo (or photos) in the collection, or go through and rapidly tag the photos. Figure 5.1 shows an example of photos laid out in a grid.

Figure 5.2 A time-line layout.

5.1.2 Time-Line Layout

Photographs are often displayed in a time-line. In such a layout, the images are ordered by the time that they are taken. There may be varying amounts of spacing between images to visually display the temporal space between the time the images were captured. Generally the temporal layout requires a lot of horizontal space on the screen, but not much vertical space.

The time-line view is useful for searching for a specific image. If the user knows roughly when the event occurred, ordering the images in a straight temporal sequence allows for a manual variation of a binary search through the images. By using the tree structure and representative image, the time-line can be condensed showing a smaller set of images, over a larger amount of time. A single image from each group is selected and displayed in a straight line, ordered by the time that it was taken. Figure 5.2 shows an example of a time-line layout.

5.1.3 Collage Layouts

While a grid or time-line layout is useful for quickly finding a specific photograph (or event), I have found that a collage layout is one way to create more aesthetically interesting renderings. I have developed two different collage layout algorithms. The first method is free form generation. The images are laid out in order of their score (Section 4.6) starting at the highest and working towards the lowest scored photograph. The size of each photograph is based on the score, relative to the other photographs in the set. Each photograph is placed on the canvas so as to maximize the amount of space that it borders with other (already placed) photographs and be as close to the center as possible, without overlapping other photographs.

The second method uses a predefined collage template, similar to the method presented in [8], to place the photographs on the canvas. Each entry in the template is numbered, and photographs

Figure 5.3 A freeform collage layout.

Figure 5.4 A template based collage layout.

are again placed in on the canvas ordered by score. The highest scoring photograph goes into position one of the template, the second highest scoring photograph goes into position 2, etc. The template is ordered so that position 1 is in the center of the canvas, and positions 2 and 3 are on either side of it. Positions 4 through 8 are directly above, and 9 through 12 are directly below. Positions 13 to 16 and 17 to 20 are columns on the side. This pattern continues until all of the photographs are placed.

5.2 Modifying Layouts

The changes made to each layout method are identical. The actual implementation of the layout methods are not changed. The difference is, rather than displaying all of the images on a single layout, the visual information is reduced. Using the organization provided by the tree structure (Chapter 3), a layout is representative of a single node in the tree. A representative photograph from each child of that node is used to populate the layout. An image set with several hundred pictures will probably only have tens of pictures displayed on the layout, rather than the entire photo set.

Above, I have described and shown this idea of reducing the visual information by using the representative image selection and organization tree, for four different layout mechanisms. However, the idea is not specific to those four layouts alone. Any layout mechanism should be able to

be modified in this way to reduce the visual information being displayed. These layouts may also be used for navigation through the collection, an idea that I describe in greater detail in Chapter 6.

Chapter 6

Applications

I have built several different photo browsing applications. All of these applications are based on the idea of using a tree of photographs (Chapter 3), the representative image selection (Chapter 4), and different layout methods (Chapter 5). In this chapter, I describe the construction and use of these applications.

6.1 Photo Browsing Tool

Using my methods as the control structure, I have developed a desktop photo browsing tool similar to Photomesa or Picasa [1, 6]. In this section, I briefly describe the implementation and workings of the tool in order to give a description of the interface that I have developed and an understanding of how one would use the tool. In Chapter 7 I describe how the tool is used for different tasks. Whenever displaying a non-leaf node, the user is shown a layout that summarizes the photographs that are underneath the node. A single photograph is shown whenever the user reaches a leaf of the tree. The tree of photographs and layout are dynamically generated at run-time; only part of the photograph score is computed off-line. In order to reduce computation time and memory usage, layouts are generated as requested by the user. The specific layout style is left to the user to decide and can be changed dynamically. This is useful if the user wishes to go between searching for a specific photograph (using a grid layout) to browsing the photographs for enjoyment (using a collage layout).

Traversing the tree, or browsing the collection, is done using the mouse. Left-clicking on an element of a layout moves down one level, bringing up a new layout based on the group the element

represents. Right-clicking anywhere on the canvas will move up one level back to the parent layout. Examples of paths through a collage layout can be seen in Figures 6.1 and 6.2; the root node for each layout is Figure 5.3.

As the user mouses over elements of the layout, the thumbnails of the photographs that are represented by the element are displayed at the bottom of the screen. The number of photographs and time range of the cluster is also displayed for the user. Figure 6.3 shows an example of an image that was selected with the thumbnails for that image underneath.

When moving between two layouts, a transition may be displayed. The transition between the layout helps to avoid jarring the viewer and give a visual connection between the two layouts. The transition I have implemented slowly fills the canvas, starting with the photograph that was clicked and continuing in descending order of score. It should not be difficult to imagine the construction of other types of transitions. Finally, the user is also given the ability to set the background color to help visually separate the background from the photo elements. Figure 6.4 shows a screen shot of the collage program.

6.1.1 Web-based Browsing Tool

In addition to the desktop photo browser, I have also developed a web-based photo browser, also using the same methods. The web-based browser was built as an AJAX script. The photographs can be placed on a web server and the script does not need to be adjusted for different sets. Again, clicking on a photograph will traverse down one level. A button is displayed for moving back up the tree to a higher level. For the web-based implementation I do not include transitions because it is not possible to ensure that the photographs will be transferred in a timely manner and the correct order. Figure 6.5 shows a screen shot of the web based browsing tool.

The photo browsing tool also works for viewing images from the Flickr web site. There is no scoring information for the photographs stored on Flickr, which is required for my methods. There are two ways to work around this. First would be to randomly select a single image to be representative. The study in Chapter 4.3 suggests that while this will not produce great results, it is likely to be reasonable. The alternative method is to locally download the photographs for scoring

Figure 6.1 A path through the tree.

Figure 6.2 A path through the tree.

Figure 6.3 (Top) Image selected. (Bottom) Thumbnails displayed from set that top image represents.

Figure 6.4 Screen shot of the photo tree browsing program.

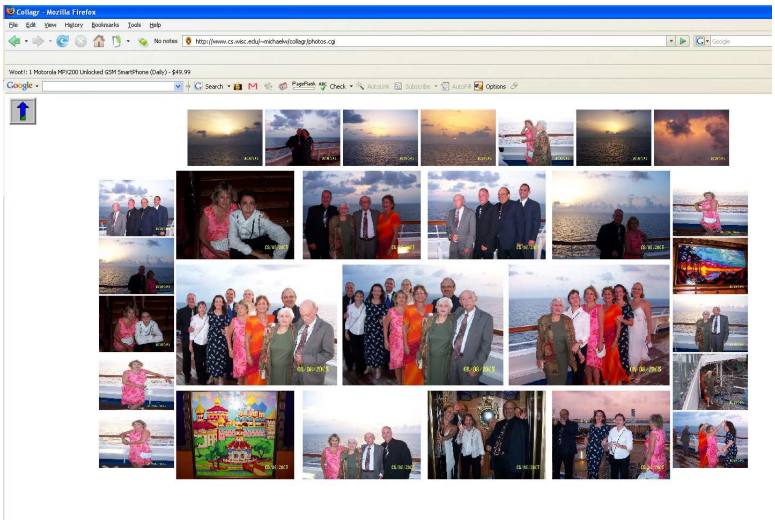

Figure 6.5 Photo viewing program displayed in Mozilla Firefox.

and then attach the score as a Flickr tag for the photograph. This has the restriction that it must be carried out by the owner of the photograph, or at least someone who was authorized by the owner to add such information. Using the Flickr API I have implemented the first approach as an AJAX application, as well as a desktop tool for downloading an entire collection of Flickr photographs, which can then be viewed in the desktop application.

6.2 Tagging

There have been many methods presented to speed up the process of tagging photographs, such as using a "drag and drop" [47] method, or using some type of computer vision approach [23]. I implemented a novel approach to tagging photographs by employing the methods described in this dissertation. By combining existing tagging methods with methods presented in this dissertation, tagging methods can be improved. If several photographs along a branch of the tree are given the same tag, then all of the photographs along that branch may be given that tag as well.

Since each sub-tree represents a specific event in the set, a label given to a node can be propagated down to the children of that node rather than having to separately label each photograph in the set. I was able to label complicated streams containing hundreds of photographs, to a point where the tree could be searched and every photograph tagged with multiple tags, in approximately 10 minutes. The labeling can be used as either a method creating new combinations of trees, or to correct the event clustering when temporal information is not enough. Figures 6.6 and 6.7 show two examples of new collages that were generated based on different tags within the same photo stream. Figure 6.6 shows the photographs that were tagged as "Cayman Island," representing all of the events from the single day spent there. Figure 6.7 shows all of the photographs that were tagged as being of the "Ship." The photographs in this group spanned the entire set of photographs in different days and events.

Future investigation in this area includes integrating additional labeling mechanisms with our methods. For example, a drag and drop interface can easily be combined with the tagging that we describe. Further, newer cameras are beginning to come with GPS data as part of the captured meta information, the location can be translated into labels for the tree [36].

Figure 6.6 A collage layout from the vacation stream for photos with the label "Cayman Island." This represents several groups in the original tree.

Figure 6.7 A collage layout from the vacation stream for photos with the label "Ship." This represents several groups in the original tree.

6.3 Digital Photo Frame

Digital photo frames are becoming extremely popular. The frame allows users to upload photographs and displays each photograph for a preset amount of time. The frame is essentially a low-end computer with a small LCD screen that is always running a screen saver application.

Using the methods presented in this thesis, a similar device can be constructed; and software was written for this task. Rather than displaying individual photographs in a slide show format, the screen can display a collage layout of the photographs. A collage from some level of the tree is displayed, every n seconds a new collage is displayed randomly moving either up or down the tree. Whenever the system is at the top or bottom of the tree it will move down or up, respectively. Otherwise, the choice to move up or down is made randomly, with a slight bias towards moving down. When moving down, the direction is also randomly selected.

6.4 Photograph Sharing

One of the main uses of digital photographs is to share them with friends and family. However, it is not feasible (or socially acceptable) to share hundreds or thousands of photographs at once. It is too burdensome to expect others to flip through many photographs.

Using the methods described in this dissertation, there are multiple solutions to the problem of sharing. First, all of the photographs can be shared, along with the tree and representative selection information. This way the recipient can browse through the photograph tree, looking at many pictures at once; and only follow those branches that are of interest.

Giving a predefined path, or tour, through the photograph tree is another solution for sharing photographs. A narration can be included with the path, to create a variation on a slide show. The recipient still gets to view many of the images, without having to go through all of the individual images.

Chapter 7

Conclusion

Browsing is one of the fundamental operations in which people interact with their digital photograph collection. This may take the form of simply enjoying the collection, searching for a specific photograph, or finding a set of photographs to share with others. As the collection grows, browsing becomes more difficult. This is because large photograph collections require more organization to be able to browse (or search) in an efficient manner. In this dissertation, I presented methods for automatically organizing large collections of digital photographs without requiring additional user interaction. I also presented applications that make use of these methods for interacting with the collection.

7.1 Contributions

In the introduction (Chapter 1) I presented a list of five contributions that this dissertation makes towards the problem of automatically organizing large collections of photographs. I now revisit each and briefly recap my contribution in each area.

7.1.1 Photograph Clustering

In Chapter 3, I make the claim, along with several other researchers, that photographs tend to be taken in bursts. By investigating several different photograph streams (approximately 40), I have shown further evidence that this claim is true. This burst pattern can be seen at any zoom level of the time-line. For example, a photo stream many contain the pictures taken at a birthday party. There is a large burst of pictures on the day (and the hours) of the birthday party. If we

were to zoom in at the time of the party, we would be likely to see separate bursts around the cake, opening the presents, and each of the party games.

I have shown how a single-link hierarchial clustering algorithm can be used to automatically find the clusters within the set. By recursively finding the bursts at every level, the entire set of photographs can be clustered in a tree structure. Each level of the tree can be built in $O(n)$ time. The entire tree can be computed off-line if desired, however, it can also be computed in realtime, as a user requests each new level. While other researchers use a similar clustering technique, the method that I present does not require bootstrapping the clustering ([16]) and only has one variable that needs to be set ([54]). The value that I use has been found to be acceptable for every stream that I have investigated.

7.1.2 Comparison of Different Image Selection Algorithms

Automatically selecting a single image to represent a larger set is often done by many different photo organization applications. However, in general there is no justification given for how a representative image is selected. In Chapter 4, I show two different studies that look at addressing how well different applications perform.

In the first study, I have shown that humans can do a better job at selecting a representative image than five commonly used automatic methods. However, I was not able to draw any conclusions about relative performance of the other methods.

In the second study, I asked several participants to select all of the representative (and non-representative images) in various sets. From this study a formula was developed that models human behavior for selecting representative images. I also retested various automatic methods for selecting a representative image. The findings show that most commonly employed methods do not perform much better than randomly selecting an image. The main exception is using face detection; however, this would not work when a set of images does not contain any faces.

7.1.3 Implementation of a new Image Selection Algorithm

Based on the results of my user studies, I developed a formula for modeling how humans select images as being representative. This formula is a linear combination of context, appearance of people (or faces), and quality of the image. Unfortunately each of these are intangible metrics that cannot be automatically determined.

Rather than relying on these metrics directly, I use low-level heuristics as approximations. Context is approximated by the number of images that were taken close together in time; i.e. if the photographer takes many images at once then there is likely to be something important that is being captured. Appearance of people is done using standard face detection technology. Quality or aesthetics of the image is approximated by looking for internal contrast in the image. The new method that I present seems to out perform all of the other standard methods that were tested.

7.1.4 Photograph Organization User Interface

In Chapter 5 I describe how the methods that I present can be used to improve existing layout algorithms. Rather than displaying the entire photograph collection at once, the set is organized into a tree structure, and a single photograph from each child of the root node (or node of interest) is displayed in a layout of the users choice.

I present an application for viewing the layout, as well as navigating through the photograph set in Chapter 6. The user is shown a layout at some level of the photograph set. To view more photographs, the user can click on a single image and a new layout containing the photograph set that the clicked image represents appears. The user can right-click on the layout to move back up the tree. This presents a new interface and organization of large collections of photographs.

7.1.5 Additional Photo Collection Applications

In addition to the photo browsing tool, I have also shown how the methods that I present can be used for other applications. For example, the user may interact directly with a single branch of the tree and apply some operation such as tagging or image manipulation. The operation can be applied to the entire branch rather than individually on every single image.

Other applications can replace standard photograph slide shows. For example, a digital photo frame usually shows single images in a slide show format. This can be replaced by randomly walking up and down the photo tree showing the layout at any given level. Another application is to give a narrated path through the photo tree, again displaying the different layouts at each level.

7.2 Limitations

There are some limitations to the methods that I present. Most notably, these methods will only work on streams of photographs, where the temporal metadata is in place. This is because the clustering algorithm relies on this information to build a tree structure. When computing the context of a photograph, it will use the clustering, and thus needs to use the temporal data.

Overall this should not pose a problem, as virtually every camera on the market includes a time stamp when the photograph was taken. The time stamp does not need to be accurate, only precise, since all of the computations are relative to the other photographs. Two or more more streams can be combined together without any extra work, providing that there is no overlap between the events being captured. If there is an overlap, the user needs to select one photo from each stream that corresponds to (roughly) the same photograph in another stream so that the offset between the different camera clocks can be computed. However, if multiple cameras recorded different events with time stamps that are close together, then those events would be clustered together and the representative image selection would not be very accurate.

The methods that I presented will not work on general image collections. For example, if a user were to go through the web and download images from different web pages. This is because there would be no events around which the methods could cluster. If the images collected do not contain any metadata, then the methods cannot function at all.

7.3 Impact of Future Technology and Advances

As technology improves, the methods presented in this dissertation will improve with these advances. For example, many experts predict that cameras will soon come equipped standard with

GPS sensors[1] so that the location of the photograph can be included in the metadata. There are already cameras on the market with such capability and a field in the EXIF specification for such an entry. This location information can be incorporated into the clustering in order to produce better results.

As advances in computer vision are made, approximations for automatic image selection can be improved. For example, improved face detection or recognition will help improve the face component of selection. Likewise a better model for context detection and aesthetics may go a long way in improving the results.

7.4 Comparison of My Methods to Other Browsing Tools

This dissertation attempts to address the problems arising from having extremely large collections of digital photographs. The main contribution of this dissertation is to help aid in tasks involving photograph browsing by offering new methods that automatically organize the photograph collection. I have presented several new methods that combined together create a new interface for browsing large collections of photographs. More specifically, I claim that the organization and interfaces I use aid in browsing, searching and sharing large collections of photographs. In this section, I describe these tasks using my methods versus the Windows File System, Photomesa [1], and Picasa [6].

7.4.1 Comparison of Browsing

The general browsing experience is very different from the other three tools. In the Windows File system as well as Picasa the photographs are simply laid out in a grid. The user can scan through the images one at a time and look at them. Alternatively in both approaches the photographs can be displayed one at a time in a slide show. The ordering of the photographs will be set based on time (which is the default) or some other sorting mechanism specified by the user. Photomesa will also display all of the images in a grid layout. However, since Photomesa is based

[1] At the time of this dissertation it is possible to purchase a camera with GPS sensor, however the price puts it out of the consumer range.

on a zoomable interface, the user can browse through the image collection by zooming into different areas of the grid.

By contrast, my methods do not display all of the images at once. Only the images at the top level of the tree are displayed first. They can be laid out in a grid or collage layout. Other layouts may be employed, but were not implemented for this dissertation. To browse through the collection, the user selects an image that is on display and it will call up a new layout of the photographs underneath. Although I have not conducted formal testing, several people who have used my system have indicated that this is a more enjoyable browsing experience than the means described above.

7.4.2 Comparison of Searching

I describe the task of finding a specific image by the implementation of my methods versus using the Windows File System, Photomesa [1], and Picasa [6]. The photograph in question is the 118^{th} image out of about 400 images, from a personal collection. The photograph itself is of a lizard on the beach. It should be noted that these photographs have no information associated with them other than the metadata captured by the camera, and thus I have to rely on my memory and knowledge of the event alone [40]. If there was additional information, such as tags, then this would be a different process.

First, I searched for the image using the Windows File System, set to "thumbnail" view. A thumbnail of each photograph is displayed. In order to find the photograph, I sorted the photographs based on the time taken. In order to find the photograph, I need to scroll through each image, viewing 16 images at once (the default window size, displaying 4×4 grid of images). A screen shot of the folder in thumbnail view is shown in Figure 7.1. Alternatively I could use the "film-strip" or slide show views, however these would each require my looking at every single image, one at a time; which would be even more time consuming.

Using Photomesa, all of the images are displayed on the screen in grid order. The size of each image is small, so all the images can be fit onto one screen. After scanning through the images, and finding the image, I can click on it multiple times, in order to zoom in on that image. With each

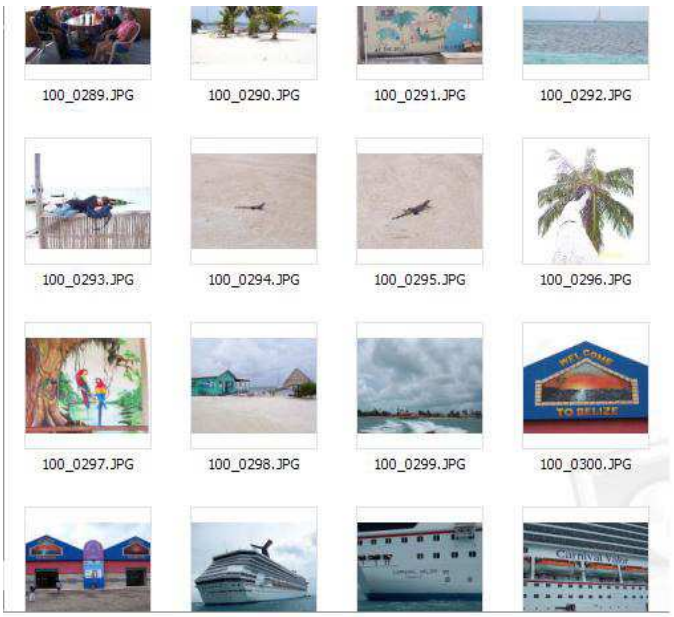

Figure 7.1 Screen shot of windows file system in thumbnail mode. To find the image in question, I need to scroll through the entire contents and look at each image until the desired photograph is located.

click, the images get progressively larger. Since the photographs are in grid layout, while zooming in, other photographs that are in the zoomed in view have very little to do with the photograph of interest. Figure 7.2 shows screen shots zooming in on the image in question.

Next, for the Picasa trial, I first create a new "album" with all of the photographs in question. I then viewed the photographs, again in a grid view. This view is similar to that of the Windows File System. The major difference is that since Picasa catalogs all of the photographs on the computer, going between different image sets is much simpler. Figure 7.3 shows the screen shot from the Picasa program. Like the Windows File System, I could have switched to a different view, however, alternative views would also have taken longer to find the image in question.

Finally, I look at finding the photograph using the methods that I present in this dissertation. In this application, at most 25 images (5×5 grid) are displayed, if more groups are necessary then scrolling would be required. I use the thumbnail display at the bottom of the screen to find the image set that contains the image that I'm interested in. As I select each image, the photographs displayed are all related to the photograph that is desired. This type of searching is possible since I am familiar with the set of photographs and have developed "memory landmarks" [40] for searching through the collection. Figure 7.4 shows the screen shots of the screen using the implementation of the methods presented in this dissertation.

7.4.3 Comparison of Sharing

Of the three systems in question, Picasa [6] is the only one that has a formal sharing mechanism. The desktop version of the program will automatically publish the albums to the web for users to share their photographs with others. The organization of the albums is still the responsibility of the user. The other two methods (Windows File System and Photomesa [1]) require the user to select those images that will be shared (if not all of the images), organize the images, and publish them in whatever way the user chooses (e-mail, CD, web, etc.)

When using my methods, the main way to share the photographs (as described in Chapter 6) is to share the tree structure along with the photographs. In this way, the user does not have to perform any extra organization; as this is built into the tree. The recipient can browse through the

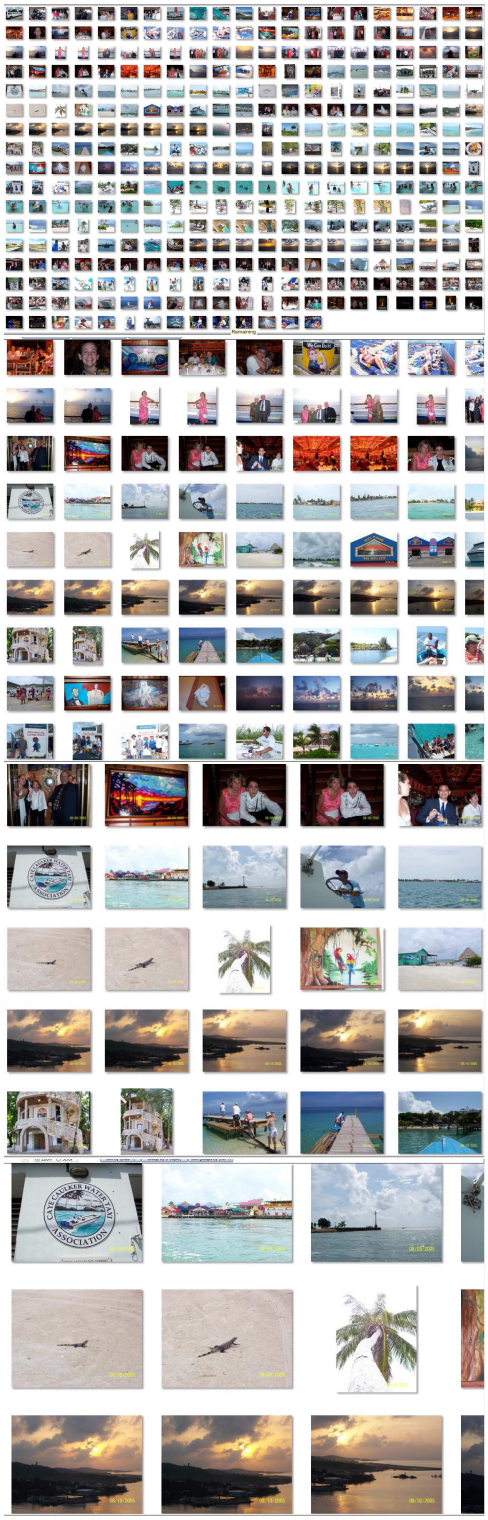

Figure 7.2 Screen shots from Photomesa program, progressively zooming in on the desired image. To find the image in question, I must first locate it within the several hundred small thumbnails and then click on the photograph to zoom in.

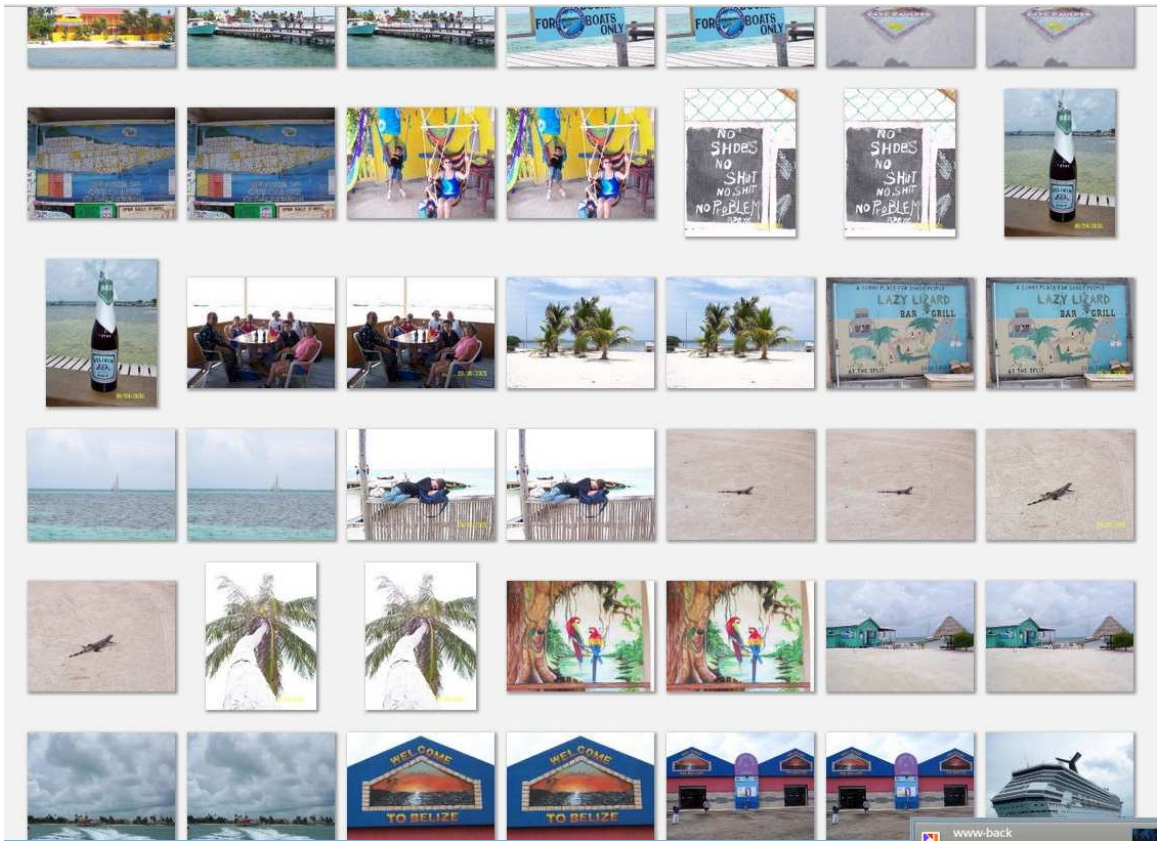

Figure 7.3 A screen shot from Picasa program. This is very similar to the windows layout, however all of the indexed photographs are displayed on the screen.

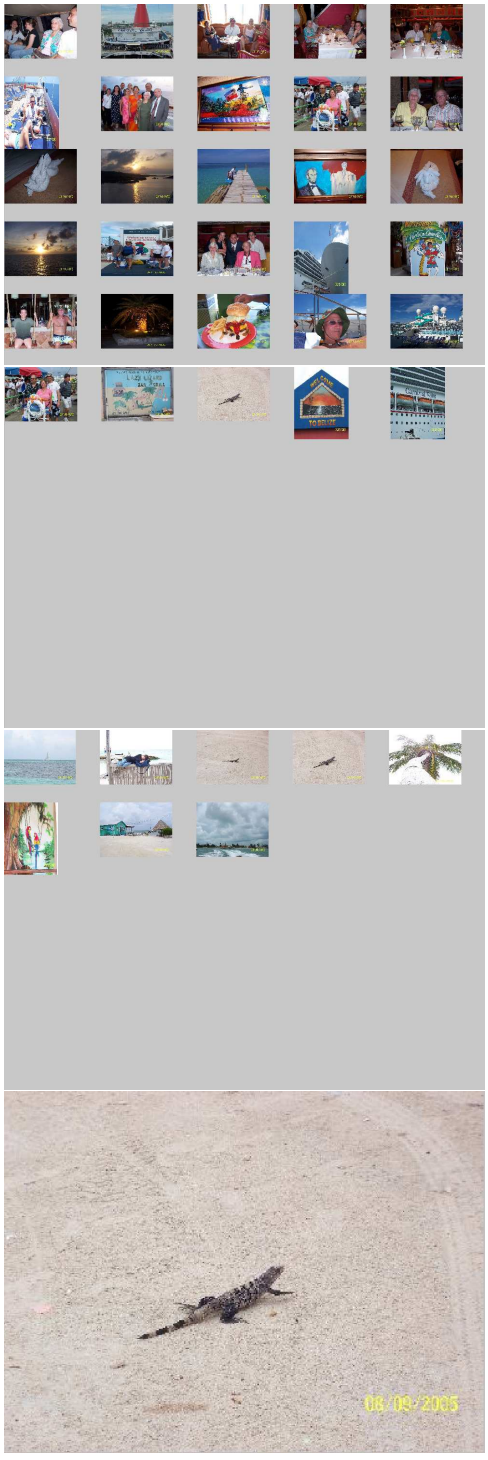

Figure 7.4 Screen shots from the methods presented in this dissertation. To find the image in question, I click on the image within the group that the photograph is located. This progressively narrows the search.

tree without having to view all of the images. Alternatively, the user may also provide a specific path through the tree to create a slide-show like experience for the recipient. There is no set way for the user to publish the photograph collection; the methods that I describe have been implemented as both a desktop program (which can be shared along with the photographs) and an AJAX script so that the photographs can be published on a web site.

7.5 Evaluation of My Methods

In Section 1.4, I laid out three requirements for my methods to address or improve upon in order to be considered successful. Briefly they were: 1) Automatic and reliable organization at any scale; 2) Reduce the visual complexity in a principled manner, without reducing the information conveyed; 3) Provide a simple and/or intuitive navigation scheme. In Chapter 2, I discussed several systems that address similar problems to those addressed in this dissertation and discuss why they do not fully meet each of these three requirements. I now revisit these requirements and describe how the methods I present each meet these requirements.

The first requirement is that the photographs should be automatically and reliably organized at any scale. Several systems will do a one-pass organization to create albums, however they do not prevent the albums from growing to unreasonably large sizes. I address this, as have others, with a recursive clustering method based on time. Photographs are taken in bursts, which can be found at multiple levels. By grouping the photographs in the bursts, similar photographs will be kept together. Providing that the metadata from the camera is kept intact, and no other overlapping events are merged with the photo stream, this method is an automatic and reliable organization scheme.

Since many photographs will contain the same visual information, the second requirement is to reduce the visual complexity in a principled manner. By reducing the number of images displayed, the user can get the general idea of the photo set without having to look at nearly as many pictures. However, if a “non-representative” image is selected to represent the other photographs this will cause the user to get an improper idea of what is contained in the set. I have not found formal justification for any method that is employed in general practice, however I have shown in

Chapter 4.3 that most of the methods employed work as well as random chance. Some systems will try to off-set the likelihood of making an improper selection by choosing multiple images, however this increases the visual complexity that is presented to the user. I present user studies which investigates the way in which humans select representative images as well as evaluate the usefulness of individual methods. I combined these findings and implemented a new method for representative selection. Further, in my system implementation, information about the set as well as thumbnails from the set are displayed to the user whenever mousing over an image. This helps offset confusion in the event that a bad selection was made.

The final requirement is for a simple navigation system. In my system I use layout mechanisms that most users are already familiar with, in order to reduce the learning curve. Navigation is controlled by the tree that was automatically generated. Whenever the user mouses over an image, information about the image set as well as thumbnails from the set are displayed to aid the user in understanding what is happening in that set. A common clicking gesture allows the user to move down through the tree and a right-click moves back up. This is similar to "forward" and "backwards" gestures in other programs. Whenever I have asked participants to test my program they have never had any problems navigating this system.

7.6 Future Work

The problem of automatic photograph organization is wide open and there are still many problems that need to be studied. As mentioned above, as technology improves, there can be many improvements made to the implementations presented in this dissertation, such as improvements to the computer vision algorithms that are employed.

Other areas of future work include using additional metadata to aid in the clustering and representative selection process. For example, GPS data could be included to aid in the clustering. It may also be used to help annotate the photographs, if the photograph is taken in a common or popular location. Other information such as news or events can also be included to help further cluster and classify images.

I have presented several applications that make use of these methods. Other existing applications can be altered to make use of the methods presented. Alternatively, other, new applications may also be created based on the ideas that I have presented in this dissertation.

Finally, an area of interest that requires more work is in dealing with large collections of photographs displayed on devices with small screens [28, 55]. Personal media players, and even cellular telephones, allow users to carry large amounts of personal media virtually anywhere. A common attribute of these devices is to have a small screen, usually no larger than a few inches. Showing a single image on such a screen is a challenging task. As these devices gain in popularity, and increase in storage size, new methods need to be developed for dealing with large collections of photographs under limited display sizes.

LIST OF REFERENCES

[1] Benjamin B. Bederson. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80, New York, NY, USA, 2001. ACM Press.

[2] John Boreczky, Andreas Girgensohn, Gene Golovchinsky, and Shingo Uchihashi. An interactive comic book presentation for exploring video. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 185–192. ACM Press, 2000.

[3] S.K. Card, J.D. Mackinlay, and B. Schneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[4] Matthew Cooper, Jonathan Foote, Andreas Girgensohn, and Lynn Wilcox. Temporal event clustering for digital photo collections. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 1(3):269–288, 2005.

[5] Adobe Corporation. Photoshop elements version 4.0. Computer Software, October 2005.

[6] Google Corporation. Picasa. Computer Software, available at http://picasa.google.com/index.html, November 2005.

[7] Kodak Corporation. Kodak easy share gallery. http://www.kodakgallery.com, November 2005.

[8] Nicholas Diakopoulos and Irfan Essa. Mediating photo collage authoring. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 183–186. ACM Press, 2005.

[9] S. Drucker, C. Wong, A. Roseway, S. Glenner, and S. De Mar. Photo-triage: Rapidly annotating your digital photographs. Technical Report MSR-TR-2003-99, Microsoft Research, December 2003.

[10] Steven M. Drucker, Curtis Wong, Asta Roseway, Steven Glenner, and Steven De Mar. Mediabrowser: reclaiming the shoebox. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 433–436. ACM Press, 2004.

[11] S. Dumais, E. Cutrell, JJ Cadiz, G. Jancke, R. Sarin, and D.C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, 2003.

[12] James Fogarty, Jodi Forlizzi, and Scott E. Hudson. Aesthetic information collages: generating decorative displays that contain information. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 141–150. ACM Press, 2001.

[13] David Frohlich, Allan Kuchinsky, Celine Pering, Abbe Don, and Steven Ariss. Requirements for photoware. In *CSCW*, pages 166–175, 2002.

[14] Yuli Gao, Jianping Fan, Xiangyang Xue, and Ramesh Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 901–910, New York, NY, USA, 2006. ACM Press.

[15] Andreas Girgensohn, John Adcock, Matthew D. Cooper, Jonathan Foote, and Lynn Wilcox. Simplifying the management of large photo collections. *Human-Computer Interaction INTERACT*, 3:196–203, 2003.

[16] Adrian Graham, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Time as essence for photo browsing through personal digital libraries. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 326–335. ACM Press, 2002.

[17] VN Gudivada and VV Raghavan. Content based image retrieval systems. *Computer*, 28(9):18–22, 1995.

[18] David F. Huynh, Steven M. Drucker, Patrick Baudisch, and Curtis Wong. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In *CHI 2005: CHI 2005 extended abstracts on Human factors in computing systems*, pages 1937–1940. ACM Press New York, NY, USA, 2005.

[19] Intel. Intel image processing library. URL http://developer.intel.com/vtune/perflibst/ipl/index.htm.

[20] Intel. Intel open source computer vision library (opencv). URL http://www.intel.com/research/mrl/research/opencv/.

[21] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[22] A. Jaffe, M. Naaman, T. Tassa, and M. Davis. Generating summaries and visualization for large collections of geo-referenced photographs. *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 89–98, 2006.

[23] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126. ACM Press, 2003.

[24] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[25] Peter Krogh. *The DAM Book: Digital Asset Management for Photographers*. O'Reilly Media, Inc., 2006.

[26] Steve Krug. *Don't Make Me Think: A Common Sense Approach to Web Usability*. New Riders Publishing, 2000.

[27] Jia Li and James Z. Wang. Real-time computerized annotation of pictures. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 911–920, New York, NY, USA, 2006. ACM Press.

[28] Feng Liu and Michael Gleicher. Automatic image retargeting with fisheye-view warping. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 153–162. ACM Press, 2005.

[29] A. Loui and A.E. Savakis. Automatic Image Event Segmentation and Quality Screening for Albuming Applications. *Proc. IEEE Intl. Conf. on Multimedia and Expo*, pages 1125–1128, 2000.

[30] Ludicorp Research & Development Ltd. Flickr. http://www.flickr.com, November 2005.

[31] Y.F. Ma and H.J. Zhang. Contrast-based image attention analysis by using fuzzy growing. *Proceedings of the eleventh ACM international conference on Multimedia*, pages 374–381, 2003.

[32] P.C. Magazine. Riya alpha. http://www.pcmag.com/article2/0,1895,1885030,00.asp, November 2005.

[33] P.C. Magazine. Tag world beta review. http://www.pcmag.com/article2/0,1895,1884543,00.asp, November 2005.

[34] T. Maurer, D. Guigonis, I. Maslov, B. Pesenti, A. Tsaregorodtsev, D. West, G. Medioni, and I. Geometrix. Performance of Geometrix ActiveIDˆ TM 3D Face Recognition Engine on the FRGC Data. *Computer Vision and Pattern Recognition, 2005 IEEE Computer Society Conference on*, 3, 2005.

[35] B. Meyers, A.J.B. Brush, S. Drucker, M.A. Smith, and M. Czerwinski. Dance your work away: exploring step user interfaces. *Conference on Human Factors in Computing Systems*, pages 387–392, 2006.

[36] Mor Naaman, Susumu Harada, QianYing Wang, Hector Garcia-Molina, and Andreas Paepcke. Context data in geo-referenced digital photo collections. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 196–203, New York, NY, USA, 2004. ACM Press.

[37] K.A. Olsen, R.R. Korfhage, K.M. Sochats, M.B. Spring, and J.G. Williams. Visualization of a document collection: the vibe system. *Information Processing and Management: an International Journal*, 29(1):69–81, 1993.

[38] J. Platt. AutoAlbum: Clustering digital photographs using probabilistic model merging. *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, 2000.

[39] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Proc. ICCV*, 2:1165–1172, 1999.

[40] M. Ringel, E. Cutrell, S. Dumais, and E. Horvitz. Milestones in Time: The Value of Landmarks in Retrieving Information from Personal Stores. *Proceedings of Interact*, pages 184–191, 2003.

[41] K. Rodden and K.R. Wood. How do people manage their digital photographs? *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 409–416, 2003.

[42] H.C. Romesburg. *Cluster Analysis for Researchers*. Lulu Press, 2004.

[43] Carsten Rother, Sanjiv Kumar, Vladimir Kolmogorov, and Andrew Blake. Digital tapestry [automatic image synthesis]. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 2005.

[44] Jeffrey Rubin. *Handbook Of Usability Testing: How to Plan, Design and Conduct Effective Tests*. John Wiley and Sons, 1994.

[45] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431. Springer-Verlag, 2002.

[46] Uri Shaft and Raghu Ramakrishnan. Data modeling and querying in the piq image dbms. *IEEE Data Eng.Bull.*, 19(4):28–36, 1996.

[47] Ben Shneiderman and H. Kang. Direct annotation: A drag-and-drop strategy for labeling photos. In *IV '00: Proceedings of the International Conference on Information Visualisation*, pages 88–95. IEEE Computer Society, 2000.

[48] AWM Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.

[49] J.R. Smith and S.F. Chang. VisualSEEk: a fully automated content-based image query system. *Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, 1997.

[50] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006.

[51] Robert Spence. *Infomration Visualization: Design for Interaction*. Prentice Hall, 2 edition, 2007.

[52] Rohini K. Srihari and Zhongfei Zhang. Show & tell: A semi-automated image annotation system. *IEEE MultiMedia*, 7(3):61–71, 2000.

[53] B. Suh, H. Ling, B.B. Bederson, and D.W. Jacobs. Automatic thumbnail cropping and its effectiveness. *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 95–104, 2003.

[54] Bongwon Suh. *Image Management Using Pattern Recognition Systems*. Ph.d., University of Maryland, 2005.

[55] Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 95–104. ACM Press, 2003.

[56] WR Tobler. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46:234–240, 1970.

[57] Shingo Uchihashi, Jonathan Foote, Andreas Girgensohn, and John Boreczky. Video manga: generating semantically meaningful video summaries. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 383–392. ACM Press, 1999.

[58] L. von Ahn and L. Dabbish. Labeling images with a computer game. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.

[59] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, 2006.

[60] J. Wang, J. Sun, L. Quan, X. Tang, and H.Y. Shum. Picture Collage. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*, pages 347–354, 2006.

[61] Colin Ware. *Information Visualization: Perception for Design*. Elsevier, 2 edition, 2004.

[62] Gang Wei and Ishwar K. Sethi. Face detection for image annotation. *Pattern Recogn.Lett.*, 20(11-13):1313–1321, 1999.

[63] Liu Wenyin, Yanfeng Sun, and Hongjiang Zhang. Mialbum - a system for home photo managemet using the semi-automatic image annotation approach. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 479–480. ACM Press, 2000. A photo organization tool.

[64] Wen Wu and Jie Yang. Smartlabel: an object labeling tool using iterated harmonic energy minimization. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 891–900, New York, NY, USA, 2006. ACM Press.

Appendix A: Alternate Study Design

In Chapter 4 I described a user study to try and determine which of the different automatic selection methods performs best. However, the design of the study and fact that human selection strictly dominated created a masking effect that prevented comparisons of the other methods against each other.

The second study that I carried out implies that most methods do not perform any better than random selection. However, since the sample size of participants is so small, it is difficult to be able to make such a strong statement. In order to do this, another study needs to be designed and carried out. Due to time constraints, I will only present an alternative design, however, I did not run this study.

Allowing participants to select among several (six) images at a time was the major flaw in the initial study. A better design would have been to use "forced binary selection." That is, only give the participant two choices at once. Doing this should eliminate the masking effect. If the image selected by "method A" is consistently chosen over the image selected by "method B," then it could be said that "method A" performs better than "method B."

The downside to this approach is that it will require many more selection tasks for each image set, where the original study only had one selection task per image set. The original study had a total of 21 sets of images containing 6 potentially representative images each. Each set of images would require 15 different selection tests. The total test with this design would require each participant to make 315 selections.

It was difficult to get the participants to carry out the 21 selection tasks without quitting in the middle of the task. It is unreasonable to expect that participants would be willing complete the study having to make 315 individual selections. A new design would have to reduce the number of selections each participant is being asked to make. This can be done by either reducing the number of sets, the number of comparisons being made (i.e. not checking each possible method against every other of method for every set), or using both techniques. In general, I believe that it would be

better to reduce the number of comparisons being made, rather than reducing the number of sets. By reducing the number of sets, there is not as broad of a sampling of image types

When reducing the number of comparisons, it needs to be decided if the same methods should be compared for each set, or if the comparisons are randomly selected. In this case, I would advocate using the same set of comparisons for each image set. This should make the results more consistent.

Finally, each image set and set of method comparisons should be randomized for each participant. This is the same way that it was carried out in the original study. This way, if users become more tired towards the end and do not provide accurate answers, this will be minimized by being spread thinly through the entire data set, rather than strongly represented in the last few image sets. Along those lines, the original study did not record any incomplete studies, a user had to press the "submit" button before the response was recorded. The new study should record incomplete responses, since this new study is longer than the original and it is likely that many people may choose to not see it to completion.