UC Berkeley

Title
Finding Lost Children

Permalink
https://escholarship.org/uc/item/7sn371vm

Author
Eden, Ashley Michelle

Publication Date
2010

Peer reviewed|Thesis/dissertation

Finding Lost Children

by

Ashley Michelle Eden

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Trevor Darrell, Co-Chair
Professor Jitendra Malik, Co-Chair
Professor Karen DeValois
Professor Maneesh Agrawala

Fall 2010

Finding Lost Children

Abstract

Finding Lost Children

by

Ashley Michelle Eden

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Trevor Darrell, Co-Chair
Professor Jitendra Malik, Co-Chair

During a disaster, children may be quickly wrenched from their families. Research shows that children in such circumstances are often unable or unwilling to give their names or other identifying information. Currently in the US, there is no existing system in the public health infrastructure that exploits image-based analysis to effectively expedite reunification when children can't be identified. Working with the Children's Hospital Boston, we have engineered a system to speed reunification of children with their families, should they get separated in a disaster. Our system is based on a Content-Based Image Retrieval search using attributes and user feedback. In this thesis we will describe the system and a series of evaluations, including a realistic disaster drill set up and run jointly with the Children's Hospital.

To my parents,
Hank and Elinor,
who have always encouraged me.

Contents

List of Figures

List of Tables

Acknowledgments

Graduate school has been an amazing journey, and I have been extremely fortunate in having had the opportunity to work with many extraordinary people. I would like to take the opportunity to thank some of the people most integral in shaping my path.

First, I would like to thank my advisors Trevor Darrell and Jitendra Malik for their continual support and invaluable guidance. Jitendra inspired me with his deep intellect, allowed me to explore so many different areas of research, and let me find my way. Trevor has been the perfect advisor – brilliant, insightful, and ever patient. I'm fortunate, indeed, to have him as an advisor, and owe him so much.

I wish to express my deep appreciation to the other members of my committee, Karen DeValois and Maneesh Agrawala, for their generosity in giving valuable time and feedback. I have also had the honor and pleasure of taking classes from them both, which have been indispensable.

I am deeply indebted to Mario Christoudias, who has been an unofficial third advisor. He has continuously believed in, encouraged, and helped me through the roughest times. Without him, this wouldn't have happened. He's a great friend and colleague, treasured by all fortunate enough to know him.

I would also like to thank my mentors at Microsoft Research, Rick Szeliski and Matt Uyttendaele, for helping spark my interest in research and providing such a nurturing environment at an early, critical part of my journey.

In addition, many thanks, to so many others who have been there for me along the way, including:

In Jitendra's group: Pablo Arbalaez, Jon Barron, Alex Berg, Lubomir Bourdev, Thomas Brox, Alyosha Efros, Andras Ferencz, Charless Fowlkes, Andrea Frome, Chunhui Gu, Joseph Lim, Mike Maire, Subhransu Maji, Greg Mori, Chetan Nandakumar, Bjorn Ommer, Xiaofeng Ren, Patrik Sundberg, and Hao Zhang, among others. I value knowing them all.

In Trevor's group: Mario Christoudias, Nicholas Cebron, Carl Ek, Ryan Farrell, Sanja Fidler, Mario Fritz, Yangqing Jia, Sergey Karayev, Brian Kulis, Trevor Owens, Kate Saenko, Mathieu Salzmann, Alex Shyr, Hyun Oh Song, Peer Stelldinger, and Raquel Urtasun, among others. They've been a family to me.

At Children's Hospital Boston: Sarita Chung, Leslie Kalish, Paula Klaman, Brittany Kronick, Ryan Licata, Stephen Monteiro, Sandy Wong, and Ryan Licata. A special thanks to Michael Shannon, who conceived of the child-parent reunification project, but – so tragically – did not live to see it come to fruition. I'm appreciative of the opportunity to have collaborated with the outstanding, dedicated staff of Children's Hospital Boston.

Many thanks also to Phuc Nyugen, a great undergrad student at Cal who helped out with this thesis, and showed great initiative. LaShana Polaris, Student Services Advisor, EECS, who has always been so gracious and helpful, especially when I've needed it the most. Valerie DeLeon for sharing with me her vast knowledge and expertise in craniofacial morphometrics.

Other graduate students: Okan Arikan, Adam Bargteil, Tamara Berg, Robert Carroll, Nuttapong Chentanez, Lillian Chu, Jaety Edwards, Bryan Feldman, Tolga Goktekin, Floraine Grabler, Hayley Iben, Leslie Ikemoto, Pushkar Joshi, Kenrick Kin, Adam Kirk, Brian Klingner, Tony Lobay, Deva Ramanan, Jason Sanders, Chen Shen, Jordan Smith, and Ryan White, among others. They've supported me in so many ways, and have been a joy to know. They've made my time at Cal particularly special.

Finally, and importantly, I'd like to express deep gratitude to my mother, father, brother Greg, and grandparents for their love, support, and encouragement.

Chapter 1

Introduction

Content-Based Image Retrieval, or CBIR, is a diverse field of research that has yielded many successful approaches for image search. Contrastingly, there is still a dire need in this country and beyond for efficient and expeditious methods to reunify children with their families if they get separated in a disaster. There are some recent advances in disaster reunification in general, but these methods mostly do not help the most at-risk subgroup of children – those who are unable to identify themselves – and none offer an efficient way for parents to search for their children. This thesis presents the design, implementation and testing of a novel Content-Based Image Retrieval system to be used specifically for fast pediatric reunification after a disaster especially for those children most at-risk. Our method incorporates both user feedback and attribute search over automatically classified attribute labels.

1.1 The Reunification Problem

There are 70 million children in the US, 22 million of whom are 5 years old or younger. After a disaster, children can be quickly wrenched from their families due to limited space in rescue vehicles [61], the rapid pace of evacuation efforts [61], and a number of other issues including the disaster striking at a time of day when children are at school/daycare and parents are at work. After a disaster, children are at at most risk for adverse consequences. Children may be unable or unwilling to give their name, address, or phone number, especially those in the following subgroups: < 4-5 years of age, developmentally disabled, severely injured, and (of course) dead [24].

After Hurricane Katrina hit in 2005, more than 5000 children were separated from their families for as long as 6 months [21], and more than 34,000 calls were placed on the special hotline set up by the National Center for Missing and Exploited Children [12]. In addition to the US, children are at high risk after disasters worldwide, as seen following the Haiti earthquake of 2010 and the Sichuan, China earthquake of 2008 where more than 7000 schoolrooms

collapsed [22].

Currently in the US, there is no system in the public health infrastructure that effectively expedites reunification when children can't be identified. After Katrina, parents had to drive around, checking hospitals, shelters, etc in a large, multi-state radius. Even then, it was difficult for parents to check. If their child had been unable to provide standard identification information, either verbally or by written information on their person, the next quickest option would be for parents to look through photos of the admitted. However, there are currently no hospital standards to photograph any patients, even in a disaster scenario.

International Red Cross data state that current methods for reunification remain primitive around the globe [21]. As mentioned in the 2010 report by the U.S. Department of Health and Human Services to the President and Congress [76], it is important to have a reunification system ready. This system should also be able to handle children who are unable to identify themselves.

There are some existing systems intended to help with reunification after a disaster in general [76]. First, several states have begun implementing ways to track evacuees. For example, Texas and Louisiana have implemented an RFID wristband system, so that as people are moved around to different shelters, their location can be tracked [93, 90]. In the Texas version, as people enter into shelters, they will be given an RFID wristband, and the list of RFID wristbands given out will be uploaded to a central database. However, no identifying information accompanies the RFID list. In the Louisiana version, people are given RFID wristbands and that information is associated with drivers license or state ID information. The same RFID number is given to all people in a family unit, including pets. Neither system, however, would reliably handle cases where children are separated from families before the tagging happens, *e.g.* if they are at school or daycare when the disaster hits, or are separated in emergency vehicles. States can also use the NMETS system [50], which is also an RFID tagging system associated with name, gender, DOB, address, medical needs, etc. Again, this does not reliably deal with the most at-risk groups of children who would be unable to provide the information with the tag. Additionally, any state-based system does not work when patients cross state lines.

In addition to tracking, the 2010 report [76] also mentioned several tools designed to help with disaster reunification in cases where the separated people do not necessarily have any tracking info stored with them. It mentions the National Emergency Child Locator, of the NEMC [75]. The NEMC encourages parents to keep photos of their children on their person, and to keep identification information on their children at all times. They will also have additional resources, in a disaster, including a hotline and a website with information about missing children. It seems, however, that the information about the missing children is gathered from the parents, which would mean that volunteers at the hospitals or shelters would have to continually check the system to report to see if they had admitted had any of

the children reported missing[1]. Another federal emergency system is the National Emergency Family Registry and Location System [42]. This system allows people to voluntarily upload information about their location and well-being so that concerned family members or friends could check. Again, it is unclear how this would help the most at-risk pool of children, as children would not be able to utilize this system if they were unable to identify themselves. Other similar tools were created by the American Red Cross [2] and Google [52, 51] for deployment in the US and globally.

1.2   Photo Indexing

The tool most similar to our system is the Lost Person Finder (LPF) by the U.S. National Library of Medicine [77]. This system allows hospital workers or volunteers to take photos of people as they are found and to hand-tag the photos with information such as: name, health status, gender, age, location, and 'image tags'. It also allows people to upload photos and information about missing family members and friends. The information is then uploaded to a central database, where family members can search the photos. In addition to looking through the photos, they can also search by tag information. The system is deployed both on computers and as a cell phone application, to facilitate taking/uploading photos and allowing searches to happen anywhere. It also emails registered users – people actively looking for lost family members or friends – photos and information as they are uploaded. Allowing users to check the information as it is added would mean fewer large-scale searches at once, and could give searchers peace of mind knowing that they will be automatically contacted as soon as the information is uploaded. In general, the LPF is a valuable system that is easy to use, and, to some degree, handles the most at-risk subgroups of children; a person's photograph is another good form of identification. However, there is some room for improvement. Currently, when searching, a parent must browse through all of the photos, and there are no user feedback or content-based image retrieval methods to help speed the process. The parent could search over tags, but these mostly include identification information (which would not necessarily be helpful in the most at-risk subgroups), status information (which is not actually part of the child's identity or something the parent would be aware of), and user-generated image tags (which means a huge set of possible keywords). The tags do, however, include gender and age, which should help lower the number of images a parent would have to look through. However, currently all tags need to be hand-entered by a volunteer, which means they are subject to some amount of error. Our system is similar to the LPF system, but we focus on a more standardized enrollment protocol, plus attribute and browsing algorithms involving parent feedback to help reduce search time, even in the presence of error.

---

[1]It is actually unclear how the information would be collected, as the system has not been in use since its creation.

1.3   Our Approach

Based on the findings in [21], and working with the Children's Hospital Boston, we have designed and engineered a browsing system using Content-Based Image Retrieval, and a central database, to quickly reunify children with their families, should they get separated in a disaster. (See Figure 1.2.) The work is part of a grant from the Health Resources and Services Administration (part of the U.S. Department of Health and Human Services), and is based on the hospital's findings that many children can not give their information, either verbally or in writing. The system overview is as follows: 1) Obtain digital images of each child as he/she enters a health care facility/triage/etc. 2) Automatically index images and archive them. All lost children's photos and information from any health care facility/triage/etc. in the area will be uploaded to the same central database. 3) Parents can then go to a designated center, input facial characteristics of their child, and search. The search uses the attributes, Content-Based Image Retrieval, and parent feedback to help expedite the process. Upon finding their child's photo in the database, they will have access to the information stored with the photo, including their child's location. Each parent will work with at least one social worker throughout the entire process, who will help the parent use the system and perform the search, deal with and alleviate mental anguish and stress involved with having a lost child, and make sure that the parent is really the parent of the child, among other tasks. (There is also the possibility of parental stress if the photos have facial trauma.)

For privacy, efficiency, and the reduction of mental anguish, the parent should look through as few images as possible. Also, based on information from previous events [24], the system must be easy to use, and must address the need for surge capacity – *i.e.* be as automated as possible. Similarly, based on information provided by hospital ER workers who have worked through several disasters, we must assume that many parents have no photos of their children.

We can frame step 3 of the system as a specific application of mental image search. In particular, we want to be able to search a set of images and find an image that matches a particular one that the user has in mind – *i.e.* a *mental image.* In our case, the images are images of children's faces. We will discuss previous work on mental image searching in Chapter 4, where we will also discuss the browsing algorithm we used for the search. This algorithm is based on iterative feedback from the parent, in which she is presented with screens of children's faces from the database of lost children, and can select those faces that look similar to her own child.

In addition to the similarity choices, we also wanted each parent to be able to search over semantic attributes for the child she is trying to find. Working with the Children's Hospital Boston, we came up with a list of distinguishing attributes, such as eye color, skin color, and age. When the children come to the hospital and have their photos uploaded, attributes may be extracted (either automatically or manually) and stored with the database of photos. Then, when a parent comes to search, she can enter her child's attributes and search over

photos with the same set of labels, which could reduce the number of images she needs to look through.

Attributes have been used in the past as a natural way to perform image search without an input image. *E.g.* , Kumar *et al.* [65] successfully used attributes in their FaceTracer system, which was able to perform a better facial image search using attribute text queries. In their system, however, the image display order was only calculated once, based on the initial query text. We hope to shorten the search time further by including user feedback, which is especially important in the presence of 'error'. There may be error in automatic attribute classification or differences between a parent's assessment of attributes and a volunteer's hand-labels. Thus, a parent should not have to look through all children of a certain set of attributes before seeing children with other attributes. It is important to integrate the attribute information with the browsing, and incorporate user feedback in order to reach the missing child faster. Please see Subsection 4.3.4 for a more detailed explanation of why it is important to integrate a CBIR browsing system with attributes.

In the remainder of this introduction we overview our system design, summarize some of the key technical challenges in creating the system, and finally summarize our contributions and conclusions.

1.4   System Overview

Our system has two separate phases: The first phase begins when a lost child enters a hospital/triage/etc., and the second phase begins when a parent comes to a designated center to search for her child's photo. The first phase will be ongoing. As children are admitted to hospitals, they will have their photo taken and uploaded to a central database, along with any necessary semantic attribute information (such as eye color, skin color, and age), and location information (such as hospital name and hospital ID). All hospitals in the affected and surrounding area will upload photos and information to the same central database. That way, parents can go to any nearby designated search center and have access to the same information as they would anywhere else. This is important both to simplify the process of locating a child who could be in one of many hospitals/triages/etc. in a multi-state radius, and for the safety of the parents, keeping them off the roads and in a facility that has trained social and mental health workers.

1.4.1   When Children Are Admitted

In a real disaster, when children are admitted to hospitals, triages, etc., they will have their photos taken and their attributes labeled, and this information will be uploaded to the central database. The hospital worker uploading the photos will first be prompted to enter the child's location (hospital, room number, etc.) along with the file path of the photo. The photo will then be read into the system and stored. Additionally, any other information
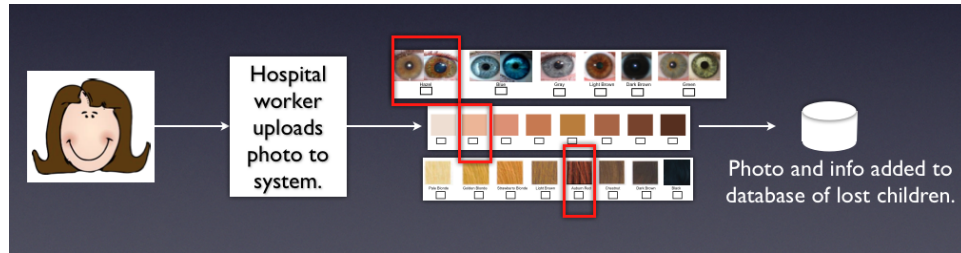
Figure 1.1: Overview of what happens when a child is enrolled in the system. The child first has her photo taken, and the hospital worker uploads the photo and identification information, such as the hospital name and room number, to the system. Then the hospital worker can select attributes that describe the child. These attributes can also be automatically determined by the system. When the worker is satisfied, the photo and all associated information (location, attributes) will be added to the global database of lost children. This database will be used when the parent comes to search.

necessary for browsing later will be calculated and stored with each photo. These calculations include automatically extracting attributes, and determining non-semantic image feature information, such as PCA features. The hospital worker has a chance to check over and/or change any of this information before it is stored. Attribute extraction and image enrollment are further discussed in Chapter 2. Figure 1.1 shows the general overview of what happens when a child is admitted.

1.4.2   Parent Search

Figure 1.2 shows the general overview of a parent's search. The parent first enters some semantic attributes about her child. The set of attributes and possible labels the parent may choose from is the same as when the children were admitted. Figure 1.4 shows a screen the parent sees before beginning the search. This particular screen allows the parent to enter the eye color of the child she is trying to find, or to skip the question.

After entering attributes, the parent browses the photos. The browsing GUI we use can be seen in Figure 1.3. The parent will be presented with a screen of children's faces, and will be allowed to click on zero or more faces that are similar to her mental image. She can then refine her search by clicking the "Refine" button. Doing this will allow the parent to see a new set of images and make new similarity decisions. Once she sees her child in the set, she can click "Found". If the parent feels that she is continually seeing screens of faces that look nothing like her child, she can choose "Select Random", which displays a random set of images on the next screen. Using this random screen for the next update should pull the parent out of the area in which she's stuck. Other buttons, "Back" and "Forward", are available if the parent feels that she has made a mistake, or changes her mind about having made a mistake, respectively. There is also a progress bar so that the parent can
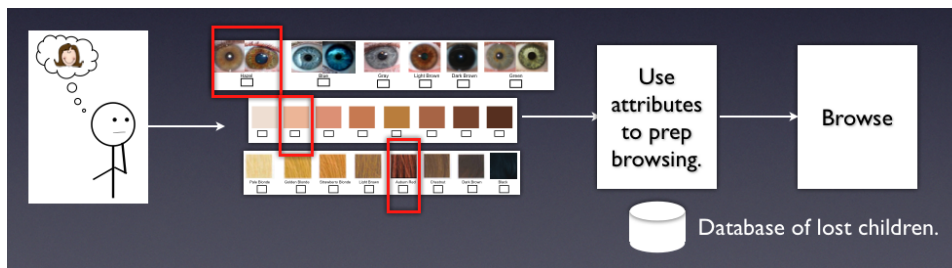
Figure 1.2: System overview of what happens when the parent come in to search. The parent is first asked to choose attributes that best describe her child. For each attribute category, *e.g.* eye color, the parent may only choose one attribute label, although she is also allowed to skip. In the user studies, we only allowed the parent to choose eye color, skin color, and age. These attributes were used to prep the input to the browsing algorithm (see Section 4.3.2), before the parent browses.

what fraction of the images she has looked through so far.

1.5 Technical Challenges

There were many technical and implementational challenges in creating this system. First, how to gather images for testing? Because our system is supposed to handle a large-scale disaster, we needed to collect a large number of children's images in order to test. Because it proved too slow to gather such a database by hand, we ended up needing to get the images from the Internet, and hand-choose those that were front-facing and of high enough quality. (See Chapter 5.)

We also needed to get attribute information for the images. What attribute labels should we use? How would we get ground-truth labels? Attribute classification by humans is somewhat subjective. When we collected ground truth information across multiple users, there was often some amount of disagreement. (See Section 5.2.1.) Also, how could we automatically classify attributes? How can we robustly extract feature information for use in classification? (See Chapter 2.1.) How could we train a classifier, and what parameters should we use for the purpose of achieving high enough accuracy given our specific setup? (See Section 5.2.2.)

How would we incorporate the attributes and browsing? Attributes, when accurate, can help cull the number of images a parent needs to look through. However, due to human inconsistencies in labeling or errors in automatic extraction, we couldn't just have the parent look through images of one set of attributes at a time. We needed a way to parametrize this 'error', and to incorporate it with the user feedback. Additionally, attributes would not be as helpful if there were a large, homogeneous population, and, similarly, there would be a greater need for browsing in a very large disaster. (See Section 4.3.2.)
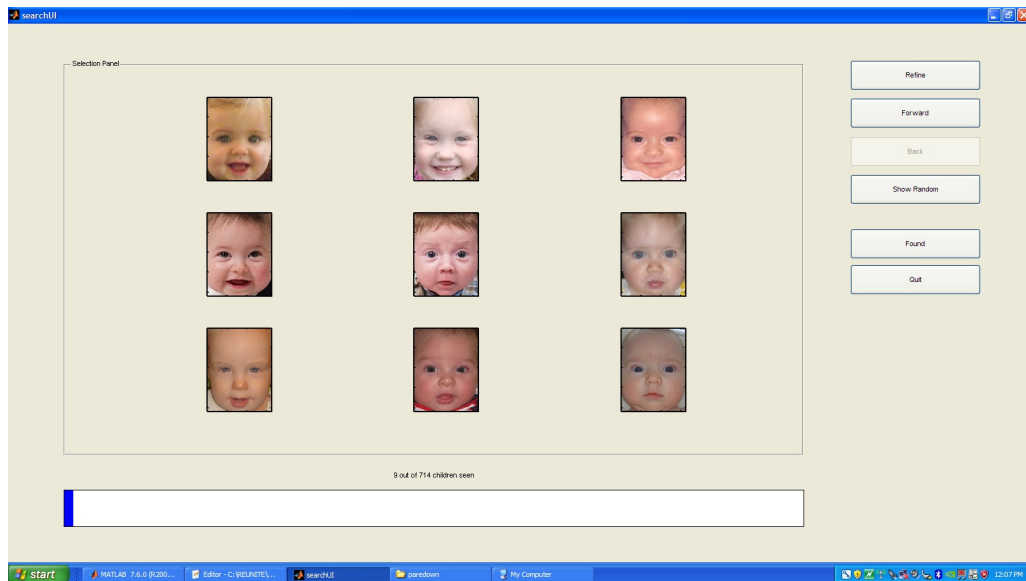
Figure 1.3: Screenshot of the browsing GUI (browsing box from Figure 1.2). On the left are 9 faces that the parent can choose from. If a parent wants to select images as similar, she will click on them and the border of each chosen image will become red. Once the parent is happy with her selection, she can click the "Refine" button on the right. If she sees her child, she can select the image and click the "Found" button on the right. The "Back" and "Forward" buttons are useful if she feels she has made a mistake in a previous screen, and the "Show Random" button is useful if she is extremely stuck.
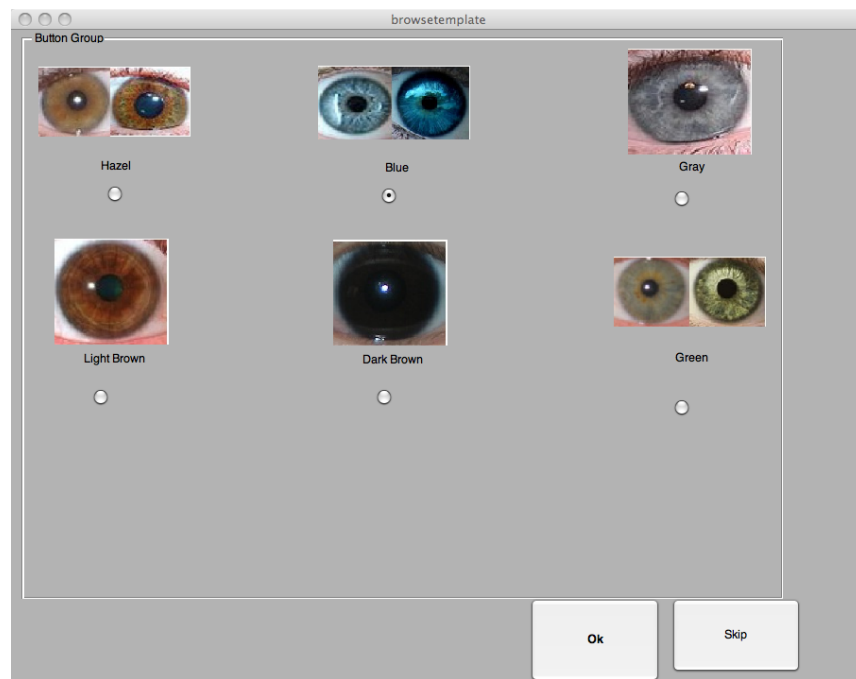
Figure 1.4: Screenshot of the screen the parent sees when choosing her child's eye color. The parent can click on any one of the radio buttons beneath the images, and click "Ok" when satisfied with her choice. If she doesn't want to choose any of the choices, she may click on the "Skip" button, and it will mark that attribute choice as unknown.

How should we browse in general? We have to assume that parents won't have any photos of their child available, so we need a search algorithm that does not require an input image. It should, however, acquire and use iterative feedback from the parent with the goal of reducing search time. The interface should be simple and easy to learn how to use. It should also allow flexibility in trying different features and metrics for comparison. (See Chapter 4.)

This raises the related question of what features and metrics to use? And is there a way to weight features based on human perceptual judgments? If so, what is the best form of perceptual judgments to use, and how could we collect that information ourselves? Also, to reduce the effects of noise, how could we collect information from many people? (See Chapter 3.)

Also, how to test the system? There are many parameters to tweak, and different settings to try, but a practical limit on the number of people we can ask to test the system. Therefore, we needed to devise a set of experiments that show the power of the system. Additionally, we needed to figure out how to best tweak the parameters, in smaller-scale testing while coding, so that we could have good settings for some of the parameters and leave them unchanged across experiments. For example, we needed to figure out at what scales to test the system; scale affects when browsing is necessary. Testing some parts of the system required culling the database so much that gathering a large enough initial database would be infeasible. Therefore, we needed to figure out how to mimic large-scale situations through different settings in a smaller database. There were also many tweakable parameters in the browsing algorithm. (See Chapter 6.)

In general, if we were unable to get enough parents to test the system, how could we test using people who weren't actual parents and therefore not familiar enough with any one child's face? (See Section 6.2.)

These challenges drove the design of our system and will be described further in the following chapters.

1.6   Thesis Outline

In this thesis we will describe the system and a series of evaluations, including a realistic disaster drill set up and run jointly with the Children's Hospital Boston. First, in Chapter 2 we will talk in more detail about what happens when a photo is added, including how attributes are automatically extracted. After that, in Chapter 3, we will discuss possible features we could use to compare images during browsing, including distances and ratios of landmarks on the face. We will also give a survey of perceptual features in previous face-related research. We will go on to discuss two psychophysics experiments we performed in order to learn the feature importance of distances and ratios in facial similarity. Then we will go over the browsing algorithm and previous CBIR work in Chapter 4. This will include how we incorporated the use of semantic attributes. We will next go over how we created

a dataset of images so that we could test the browsing, how we got ground truth labels for the dataset, and what attribute labels we decided to use in the system. These topics and the accuracy of the automatic attribute extraction will be covered in Chapter 5. Next, in Chapter 6 we will describe and show the results of experiments that tested the entire system. This will include the realistic disaster drill, which was performed at the Children's Hospital with hospital workers, social workers, parents and children. After that, we will discuss other possible features we could have used to search over in the browsing, including distances and ratios of landmarks on the face. We will also give a survey of perceptual features in previous face-related research. We will go on to discuss two psychophysics experiments we performed in order to learn the feature importance of distances and ratios in facial similarity. We will then discuss future work related to what we observed and learned while making the system in Chapter 7.

1.7 Contributions and Results Overview

- We've built the first system that uses CBIR for pediatric disaster scenarios.

- We've investigated a novel scheme to initialize the prior of a Bayesian CBIR mental image search with attribute information.(Chapter 4).

- We've automated the extraction and classification of attribute information (Chapter 2) and demonstrated its effectiveness (Chapter 5).

- We've developed a real prototype system and evaluated it in the field with real parents, through disaster drills and dyad tests conducted collaboratively with Children's Hospital Boston Emergency Department personnel. (Chapter 6.)

- In synthetic tests conducted in our lab, we found a statistically significant reduction in search time using browsing over baseline. (Section 6.2.)

- In dyad tests with Children's Hospital we found a significant improvement over random presentation, even when using automatic attributes (which were somewhat prone to error) and over a homogeneous population, in which attributes are not helpful. (Section 6.3)

Chapter 2

Attribute Processing

When a child comes into the hospital, triage, etc, a volunteer or hospital worker will take her photo and upload it to the system. There, they will be able to enter or alter attribute information. Before we discuss how attributes can be automatically determined, it is important to discuss the first step of detecting and cropping the face.

To do this, we used the Everingham face detector [36], although any face detector that finds a bounding box for the face and eyes would also work. If the detector is unable to find a face, we have a pop-up GUI that lets the user specify the bounding box for the face and eyes.

Figure 2.1 shows the screen the user will see after the image has been cropped. There are more potential attribute categories than the ones shown here. These categories were determined in part with the Children's Hospital Boston.

Once the image has been cropped, the user is presented with a screen showing the cropped image and attributes. If automatic attributes are used, there will be guesses already set in the attribute categories. If not, they will be blank. The user is able to manually change any of the attribute labels. Additionally, the user can re-specify the bounding boxes of the face by selecting "Re-crop". If the user re-specifies the bounding boxes, the automatic attribute code is also rerun, and the new automatic attribute label results are updated in the GUI.

We should note that because the images are cropped based on eye location only, re-specifying the face bounding box will not affect the way the image crop looks. However, the placement of the face bounding box does affect the result of automatic extraction. (In order to have uniform scaling in alignment, we can't both align the eyes and fit the face to the exact face bounding box.)

If hand-labeled attributes are used, the person adding the photos to the system can determine the labels directly from the cropped photo displayed on the upload screen. Alternatively, the volunteer taking the photos can mark the attributes while looking at the actual child. The advantage to this is that the volunteer will have access to the child at a much "higher resolution", and it will save time during the uploading step. (This can make a difference if there is a gap between the time the photos are taken and when someone uploads

Figure 2.1: Screen the user sees upon photo upload. On the left is the image cropped as it will be in the parent's search. If the user is unhappy with the crop, she may click on the "Re-crop" button. The images next to the attribute categories are the current choice. If an attribute was automatically determined, its choice will be visible when the screen first appears. The user may also choose "Browse", or click on the pull-down menus, in order to change the attribute label.

them to the system.) If the volunteer marks the information, she enters it onto a REUNITE sheet, as seen in Figure 2.2. In addition to marking the attributes, she also marks the image name from the camera. That way, the person uploading the images just has to go through the stack of REUNITE forms, upload the specified image names, and enter the attributes already noted on the sheets corresponding to each image.

The REUNITE form in Figure 2.2 also shows the set of attributes and attribute labels that the Children's Hospital originally deemed as possible identifiers. These are: age, gender, eye color, skin color, hair color, hair length, hair texture, shirt color, shirt pattern, eyeglasses, and birthmark/scar. Of these attributes, we focused on skin color, eye color and age. We chose these attributes because they were less likely to change between when the parent last saw their child and when the child's photo is taken, and they are stronger identifiers. Also, in order to test the effects of attributes with browsing, we needed a large enough dataset, and using more attributes would have the effect of paring down the dataset more. (See Section 4.3.4 for a more detailed explanation of using attributes with browsing, and Chapter 6 for the results of user tests.) In future work, we could try having the parents label more attributes, and see if the extra attribute "resolution" ends up helping or hurting due to additional noise.

The attribute labels in Figure 2.2 were determined iteratively between the collaborators at Berkeley and the Children's Hospital. Most of these labels, however, were somewhat subjective, and prone to error across labelers. See Chapter 5.2 for labeling accuracy – both by hand and automatic – as well as how we dealt with it and what attribute labels we ultimately used.

We also tried automatic attribute classification. In this chapter (Section 2.1), we will explain how we extracted the features for use in classification, but will save the classification specifics and results for Section 5.2.2, after we have discussed the labels we used in more detail.

2.1 Automatic Attributes

Instead of hand-labeling attributes, the attributes may be automatically determined. This section goes over our methods for automatic classification of skin and eye color, as well as our joint work on age estimation. More specifically, we will go over some previous work in automatic attribute classification, and then go over our feature extraction methods. We will save a discussion of the actual labels classified over for Section 5.2.
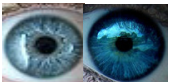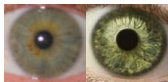
2.1.1 Previous Work

There has been much previous work in computer vision and related fields that use attribute classification, although it remains an active area of research, and there are many different methods and features used. Some works mention using attribute classification or feature

Figure 2.2: Sheet the volunteer uses to organize hand-labeled attributes.

extraction, but the specific methods used are not described. Developing a new and better method for attribute classification was not our goal. Rather our goal was to use techniques that worked well for our specific purposes. We do not make any claims as to our methods' uniqueness, novelty, or improvement over other methods. However, before describing our methods in Section 2.1.2, we will first present some previous work in the area. Because we are concerned with labeling "permanent" attributes of the face – *i.e.* , attributes that would be common to both a parent's general "mental image" and an actual image taken within a few weeks of when the parent last saw the child – we will not discuss such features as facial expression. Also, because we are working with children, we are not concerned with facial hair.

Kumar *et al.*   [65] recently used attribute classification in their FaceTracer system. They were specifically concerned with gender, age (baby, child, youth, middle age, senior), hair color, smiling, mustache, race ('white', 'black', 'asian'), eyewear, blurry, lighting, and environment. Moreover, the photos they wished to label spanned a wide range of ages. They hand labeled 1700 images, but wanted to automatically label millions more. To do this, they performed Adaboost [45] on a set of 'weak', 'local' SVM classifiers [28] over face region, types of pixel data, normalizations, and aggregations, and then passed the newly determined weights into one 'global' SVM classifier. The features used were pixel based and on different regions of the face.

Golomb *et al.* [49] and Moghaddam and Yang [74] learned gender using neural networks and SVMs respectively, over adult faces. We chose not to classify gender for the current system, because it is a much harder task over children's faces. (We determined this by attempting to personally ground truth some infant's genders.) For the current system, however, we decided that if the different genders looked similar, then there would just be more similar images to choose from in the browsing (see Chapter 4). However, as discussed in Section 6.3.2, parents of older children often did not want to choose similar images of the opposite gender, so gender would be something to add as part of future work (see Chapter 7).

There has been much previous work in detecting skin versus non-skin pixels in an image. Vezhnevets *et al.* [96] and Kakumanu *et al.* [62] gave a survey of pixel-based skin detection techniques. These techniques included: explicitly defining skin regions by learning a range in color space that defines skin. The issues here were the wide range of skin tones, lightings, etc. They also discussed nonparametric modeling, including a lookup table based on histogram binning of skin and non-skin pixels of training images, and using those bins to make a Bayesian classifier. These methods are useful in terms of which pixels are skin and not skin, though our end goal is to classify the image into a specific skin color class. White *et al.* [98] determined skin color by finding a patch co-located with the cheek to get around this problem.

In terms of iris estimation, Grabler *et al.*   [54] extracted the iris from the eye, first by using face detection techniques to roughly locate the eye, then looking for a circular region with minimum average luminance. We use iris estimation both in determining eye color and iris size. However, we use specifically tweaked algorithms for each in order to maximize

accuracy with respect to getting color of just the iris, and of finding the most accurate iris diameter.

2.1.2   Skin and Eye Color

Our general method for predicting the class label of skin and eye color is to extract color features for the skin and eyes, and then use a nearest neighbor classifier to predict the class. Extracting the color features relies on a face detector that also marks the position of the eyes. (This was necessary anyway in order to rectify and align the images for PCA.) The Everingham face detector we used finds a bounding box for the face, and bounding boxes for each eye (upper left (x, y) coordinate and the height and width). During the upload, the user also has a chance to re-specify the face and eye bounding boxes. Once we have the position of the eyes, we determine the skin color by first finding patches of skin corresponding to the left and right cheeks. We do this by looking at a patch of skin some fraction of the way down the face, dependent on the distance from the eye to the bottom of the face, and the height/width of the eye bounding box, for both the left and right eyes. We combine the pixel information from patches, convert to LAB space, and get the mean pixel color. When converted back into sRGB space, this 3x1 vector represents the skin color feature vector for that face.

The left and right cheek patches are determined using the face and eye bounding boxes. The face, eye, and cheek bounding boxes are axis aligned with each other and the image. Thus we may define each as an (x, y) position (of the upper left corner), and a width and height. We calculated the upper left corners of the left and right cheeks as follows:

$$
\begin{aligned}
left\ cheek_x &= left\ eye_x \\
left\ cheek_y &= left\ eye_y + left\ eye_{height} + \frac{\mu_l}{4} \\
right\ cheek_x &= right\ eye_x + \frac{right\ eye_{width}}{2} \\
right\ cheek_y &= right\ eye_y + right\ eye_{height} + \frac{\mu_r}{4}
\end{aligned}
\tag{2.1}
$$

where μ_l is the distance in y between the bottom of the left eye bounding box to the bottom of the face. Similarly, we do this for μ_r, but using the right eye bounding box instead. The width and height are the same for both cheeks; the width of the cheeks is $\frac{left\ eye_{width}}{2}$, and the height is $\frac{1}{12}$ of μ_l.

To get the eye feature vector, we looked at the bounding boxes of the eyes as determined by the face detector. However, not all pixels in the eye region are of the iris – the part of the eye that defines the eye color. The eye regions also include skin, eyelashes, sclera, and pupil. To determine the eye color, we chose to first identify what parts of the eye are iris. To do this, we first calculated a skin match probability for each eye region, using the skin

color mean and covariance from the cheeks, in LAB space. (The mean was a 3x1 vector and the covariance was a 3x3 matrix.) For each eye, we also ran k-means on the original (LAB) eye patch, and sorted the regions based on how many pixels, some threshold away from the edge, are less than some skin probability. Through empirical tests, we found that the region with the most "valid" pixels was usually the iris. We also found that when this assumption failed, it is usually because the regions with more "valid" pixels were extremely large, especially in comparison to the region containing the iris. Because there are only a certain number of anatomical regions of the eye, these large regions were usually mostly skin and miscellany. Therefore, after the initial sort, we ignored any regions whose average color was above some skin probability, and then took the region with the most "valid" pixels as iris region. Once we identified which k-means patch was the iris region, we took the average LAB color of that region and converted it back to sRGB. The final eye color feature vector is the average of the sRGB colors for each eye.

In order to get the eye and skin color class labels, we perform k-nearest neighbors for each. In the final system, the training set was 37 faces hand picked from a set supplied by the Children's Hospital. The images were chosen because they had similar, broad-spectrum flash lighting, and were strong examples of their class labels. Please see Section 5.2.2 for a more detailed description of the actual classifier, as well as performance numbers, and Section 5.2 in general for a discussion of the label categories we ultimately classified over.

2.1.3   Age – Work In Progress

In order to predict age, we worked with a forensic anthropologist at The Johns Hopkins University. She devised an algorithm based on landmark points on the face and iris size, which we provided for her.

The iris size algorithm was devised/implemented in part with an undergraduate researcher. We started with a rough cutout of each eye, including eyelashes, sclera, surrounding skin, and sometimes eyebrow. The eye images were automatically generated from the face detection algorithm we used, which provided estimated locations of the bounding box of the eyes. (During uploading, the hospital worker also had the opportunity to re-specify the bounding boxes for each eye.) Each eye image was first processed using Canny edge detection in order to get the edge image. Because of the visual structure of the eye, these edges usually include the boundary of the eyelid with the eye and some part of the iris boundary, in addition to noise. They also sometimes includes the pupil boundary.

We used a circular Hough Transform [5] to determine the radius and center of the circle that best fits the boundaries, and therefore, if the edge image is good, best estimates the location and size of the iris. We only accumulated counts for radius values between 10 pixels and half the height or width of the image, whichever was larger. Also, for robustness, we passed in an original estimate of the center of the pupil. (We will discuss how we calculated that original estimate below.) If the newly calculated center of the pupil was more than $\frac{1}{3}$ of the $\max(height, width)$ of the image away from the original estimate, we re-chose the best

radius, only considering radii whose maximum accumulated value was located closer to the originally estimated center of the pupil.

The original estimate of the pupil began by calculating the mean and covariance in LAB space of the cheeks, the same as in calculating the skin and eye color features. Using those values, we calculated the log probability that any given pixel in the eye patch was skin. This resulted in a probability image, Γ, for each eye. (For simplicity we will explain the rest of the calculations over a single eye, but these calculations were performed separately for both eyes, using each eye's corresponding Γ image.) We then convolved Γ with a "mask". In practice, this mask was just a 7x7 matrix of 1s, except the first and last row and column, which were $-.5$. The mask essentially functioned as a box filter, which blurred the probability image and got rid of small anomalous regions. We then calculated $\min(\Gamma)$ – the minimum probability of being skin. Any pixel in Γ with this value was likely to correspond to a location containing actual eyeball in the eye patch image. In order to avoid anomalies due to noise, we then found all pixel locations in Γ with a value $\leq \min(\Gamma) + \frac{1}{10}|\min(\Gamma)|$. The mean of these locations became our new best estimate of the centers of the eyes.

The face landmarks we provided were automatically determined by the Gu *et al.* face detector [55]. This detector placed 83 landmarks, as seen in Figure 2.3. Using the iris size and landmark positions, our collaborator is still determining an algorithm to estimate age. (In the tests in Chapter 6, age was either not used as an attribute, hand-labeled, or a temporary age classifier was described and used.) The iris detector failed if there were glasses, or if the child was squinting or looking away from the camera to a considerable degree. From a set of 110 images hand picked for quality (open eyes, front facing), only one seems to have completely failed on both eyes. After adding up the automatically determined iris radii of both eyes, and comparing them to the sum of the ground truthed radii (as determined by an anthropologist), the mean error was $-.045$ of the actual size, with a standard deviation of .12.
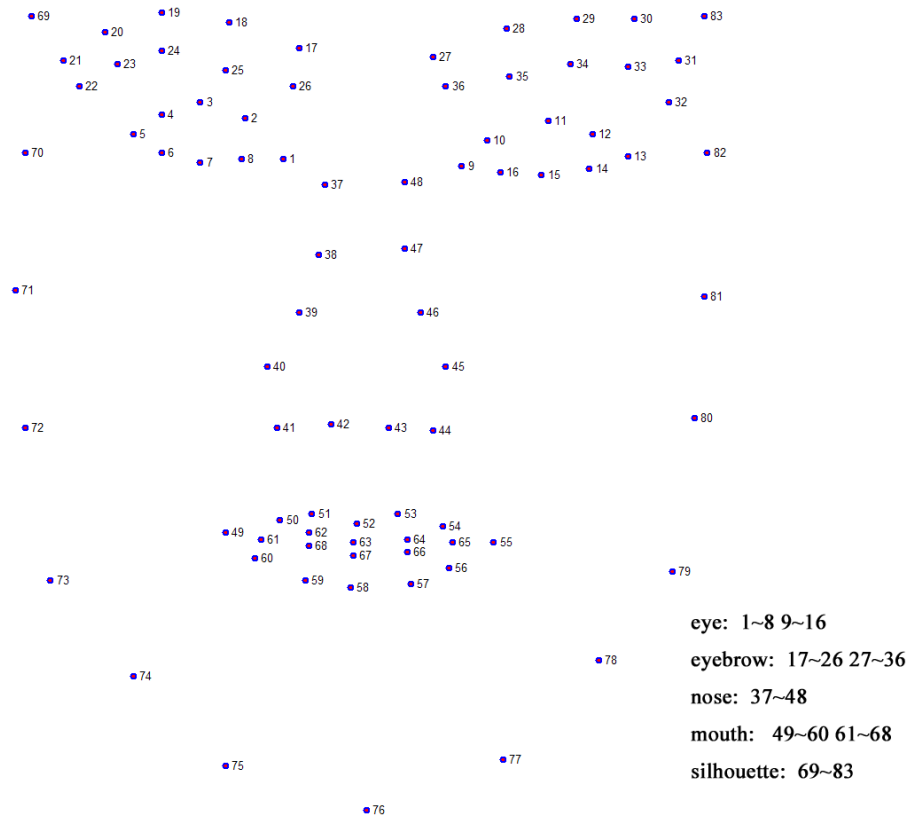
Figure 2.3: Landmarks from the Gu *et al.* face detector. There are 83 landmarks altogether.

Chapter 3

Perceptual Representation

Regardless of the browsing algorithm used, we will need to rank the images based on similarity information. Therefore, we need a good measure of visual similarity that is perceptually based. Using no measure of visual similarity would give a browsing performance equivalent to random. In the experiments of Chapter 6 we used $L2$ unweighted distances on PCA features, which proved effective. However, it was worth exploring more features, and a possibly better comparison metric, to further improve performance.

In this chapter we will first go over features used previously in face recognition research. Then we will go over some psychology research on facial similarity. We will then give a more detailed background of how distances and ratios are important for facial similarity research in fields other than computer vision. Even with a good set of features, we still need a metric to compare faces. We will next discuss how to learn a better metric for comparing features, including going over the method we used. Metric learning is based on both feature information and similarity judgments. Because we ultimately want to use the metric in a user-feedback scenario, our similarity judgments need to be perceptually based – *i.e.* actual human judgments. In the next section we will go over previous work in learning a metric for facial similarity using perceptually-based similarity judgments. Then, when we lastly describe our own setup for learning feature importance, it will include our experimental setup for gathering the perceptually-based similarity judgments in addition to the features used and metric learning parameters.

We should note that the additional features and weightings described in this chapter were not actually used in our pediatric reunification system. Our results here ended up being inconclusive, so we continued to exclusively use PCA features when testing the entire system. However, we still believe that features such as distances and ratios are important and will be beneficial in the system. As such, this chapter should be thought of as a guide to future work.

3.1   Previous Features Used in Face Recognition

Face recognition (and face verification) is a rich field with much previous research [103, 68]. The goal of face recognition is to verify that pairs of images are of the same person, or to search a database to find other examples of a query person. Challenges arise due to differences in lighting, pose, and expression, as well as changes to the person's appearance. Here we will go over some of the features used historically in computer face recognition techniques, which also draw on some techniques used in face detection [100].

We will be focusing on face detection and verification techniques that do not use a 3D model, and are pose-dependent. These techniques can be divided into "Appearance-based" approaches which consider the entire face as living in some subspace, "Feature-based" which focus on specific parts of the face, and hybrid methods. Some of the most widely used features for appearance-based methods are as follows [103]: Eigenfaces [94], based on a projection of the face into a PCA basis. Fisherfaces and subspace LDA [7] based on a Fisher Linear Discriminant and is invariant to lighting and facial expressions. Independent Component Analysis [6] is a generalization of PCA. There have also been successful face recognition methods based on the Active Shape Models of Cootes *et al.* [25]. Brunelli and Poggio [17] tried both a template-based and feature-based method, and found that the template-based method performed better.

Feature-based methods include the use of purely geometric features based on distances and/or ratios of landmark points on the face [31], [63]. There is also graph jet matching based on Gabor-type wavelets found on the face [99]. Here, there was geometry information comprising the graph, and local features stored in the jets.

Hybrid methods, which have arguably had the best performance, combine both sets of features. For example, Pentland *et al.* [80] used both eigenfaces and semantic features in creation of Photobook., an image browsing tool. In it, the users first enter semantic information such as race, age, facial expressions, and gender, and are presented with a set of images of people fitting the description. Each person in the database is represented by at least two photographs with possibly varying expressions, hairstyle, etc. After being presented with images that fit the semantic descriptions, the user may select a face. Photobook will then reorganize the rest of the faces in the database according to similarity (over the eigenface representation). Here they assessed the accuracy of their system by whether the same person as selected showed up in the top matches.

3.2   Human Face Recognition Results

There are many papers in psychology and cognition that explore how humans process faces [14, 102, 13, 58, 16]. This is different than computer face recognition research. In computer face recognition, one is only concerned with the accuracy of the recognition, not necessarily in drawing any conclusions about whether the recognition is performed similarly to what

humans do. Analogously, features found to be important in human recognition can not necessarily be directly assumed to be useful or the best line of attack for our purposes; who is to say that the metric we end up using is the same as that used by humans? Nonetheless, it is important to have an idea of what the psychology literature has found on this subject [88]. In this section we will briefly go over some psychology research on facial similarity judgments in humans.

Sinha *et al.* [88] go over nineteen results from human face recognition that may be relevant to making an automated face recognition system. Paraphrasing from the paper, the first fourteen results are as follows: 1) "Humans can recognize familiar faces in very low-resolution images". 2) The more familiar a person is with a face, the lower the quality of the image can be for the person to still recognize it. 3) High-frequency information, such as an edge image, is not enough for good face recognition performance. 4) The appearance of facial features such as eyes, nose, mouth, etc, are not sufficient for recognition; their placement is important as well. 5) Eyebrows are important. Without seeing eyebrows, human face recognition gets significantly worse. 6) A face may be compressed in the x or y dimension with no reduction in face recognition performance. This may mean that ratios of distances within the same dimension are important. 7) Caricatures are recognized better than the veridical face. This may mean that humans use a "norm-based" encoding of faces, where the faces are stored in a mental "face-space" based on how far away from an average face they are. 8) Prolonged exposure to a face can lead to "after-effects", such as "anti-faces", which also supports the "norm-based" encoding theory. 9) Pigmentation cues are important in addition to shape. 10) Color cues, in addition to luminance, are important, especially if the face is very low resolution. 11) Viewing a negative of a face greatly reduces performance. 12) Humans can recognize faces under large illumination differences, but their performance is affected. 13) Humans can recognize the same face under different viewing angles, but when shown other faces inbetween, this seems to hurt their performance; the temporal proximity of the presentation of faces seems to be important. 14) Face motion, especially non-rigid expressive movements, facilitate face recognition. The expressive movements may help evince facial structure. The rest of the results go over developmental progression and neural underpinnings, which are beyond the scope of relevance for our purposes currently.

Bruce *et al.* [15] give a brief survey of some key human face recognition results and two computer face recognition methods. The computer methods use PCA [94] and graph jets based on Gabor-type wavelets [99] as features respectively. They also ran their own psychophysics tests on facial memory and similarity tasks and compared the results to the rankings of the computer methods on the same set of images. They found that both give statistically significant correlations with the human results, and that PCA may be more useful when seeing an unknown face, and the graph-matching may be more useful on familiar faces.

Steyvers and Busey [91] tried to predict similarity ratings using a combination of features used in computational approaches and a mapping to abstract features. They found that the mappings correlated to 'age' and 'facial adiposity'. The features used were PCA and

geometric distances, and Gabor jet and geometric distances. They found that both gave similar results, and that the geometric distances on their own seemed to be important. They mentioned that the faces they used, however, were fairly homogeneous, and with a more heterogeneous population the PCA and Gabor information may make more of a difference.

Valentine *et al.* [95] analyzed the effects of distinctiveness, inversion and race in human face recognition. Davidenko [33] used silhouetted face profiles and ran a series of psychophysics tests to predict gender, age, and attractiveness. He found that the results on the silhouette-only images were reliable. Rhodes *et al.* [83] and Benson and Perrett [9], to name a few, found that caricatures of a face helped in human face recognition. The caricatures were created by warping the face image after exaggerating landmarks on the face based on their distance from their corresponding positions on the average face [11]. In Maloney and Martello [72] two sets of participants rate the absolute similarity of pairs of childrens' faces, and classify sibling/non-sibling of the same faces respectively. They found that the similarity ratings were a good predictor of the sibling classifications. Also, the similarity ratings did not vary with age or gender differences between the images in the pair even though the subjects were not told how to interpret 'similarity'. Goffaux *et al.* [48] found that high spatial frequency information was useful when comparing faces with featural differences (such as in the eyes, nose, mouth), and low spatial frequency information was useful when comparing faces with configural differences.

3.3 Distances and Ratios

Distances and ratios have long been used as perceptual measures in fields other than computer vision [35]. Means and standard deviations of distances and ratios are the basis of *anthropometric data*. Anthropometry is the science of measuring the human body and gathering statistics across different populations. Anthropometric information is useful, for example, in forensic anthropology. A forensic anthropologist may use this information to age a photo of a missing child, classify remains, or virtually reconstruct skulls. These same distances and ratios are used in medicine by plastic surgeons and orthodontists in restructuring faces [84, 41, 39]. Distances and ratios can also be used in clinical genetics and in facial deformity research. Anthropometric methods sometimes employ morphometrics, tools used in the more general study of differences in size and shape between objects [10]. Using these tools, landmark positions and distances may be transformed into information on shape.

Artists have also long used distances and ratios both as guidelines to help draw faces, and to assess beauty. Medical studies have shown a correlation between beautiful faces and ratios on the face that follow 'The Golden Ratio', or 1 : .618, and Facial Thirds. [8, 101]. Ricketts [84] also discusses these ratios and other angles in his paper on planning for maxillofacial surgery.

Additionally, both statistical analysis and machine learning have been applied to distances and ratios of faces in order to respectively determine if there were any significant

features that correlated with beauty, and in order to beautify existing photographs. Gunes *et al.* [57, 56] made classifiers based on the ratios established in the medical studies and collected attractiveness ground truth ratings. They were able to predict the ratings with a standardized difference of one class out of ten. They intend this measurement to be helpful in plastic surgery planning. Farkas [66] took anthropometric measurements and color photos, both frontal and lateral, of 315 young adults. First, the photos were shown to a team of 6 'experts', who classified their attractiveness as an integer on a scale of 1 to 6. These numbers were averages and grouped into three labels: below-average, average, and above-average. They then reported on the means and standard deviations of a series of distances and ratios for the below-average and above-average faces.

Leyvand *et al.* [67] also used distances and ratios to automatically beautify photos of faces. They took feature points on the face and created a 2D facial mesh, where the edge lengths, *i.e.* the distances between points, form the feature vector that describes each face. For each face, they found the k nearest neighbor faces based on their beauty-weighted feature vectors, and modified the current face's feature vector in the direction of the neighbors' beauty-weighted mean. They recreated new landmark points for the warped mesh through stress minimization, and warped the photo based on the new mesh. Through user studies, they showed that the resulting faces were in fact more attractive.

Distances and ratios were also used in other graphics research, such as in DeCarlo *et al.* [35]. They used anthropometric measurements, based on those listed in Farkas, to automatically generate varied, realistic 3D face meshes. They first generated a set of distances and angles of landmark points on the face, following those described by Farkas [66], that reflected the distributions recorded for a given population. Then they used these measurements as constraints and used a fairing method to generate a smooth 3D facial mesh.

3.3.1 Set of Measurements

In terms of what measurements have been historically used to describe faces, Farkas [66, 39, 41] has written multiple books on anthropometry, including distance, ratio, height, weight, and angle statistics for children and young adults. He uses standard tools from physical anthropology, including the sliding and spreading caliper , and marks the landmark positions directly on the skin. Some landmarks correspond exactly with their anatomical 'bony' counterparts, and some are nearby, their locations determined by underlying soft tissue. Some of the landmark point on the face can be seen in Figure 3.1. Paraphrased from Farkas, the distance measurements he includes are all linear or angular. The linear measurements are either projective or tangential. Projective measurements are the shortest distance between two landmark points, and tangential are taken using a soft measuring tape along the facial tissue. The measurements can be divided into three dimensions: horizontal, vertical, and depth. Angular measurements are taken by first aligning the face to a standard position, the Frankfort Horizontal, or the line connecting the bottom of the orbitale (eye socket) to the highest point on the external auditory canal.
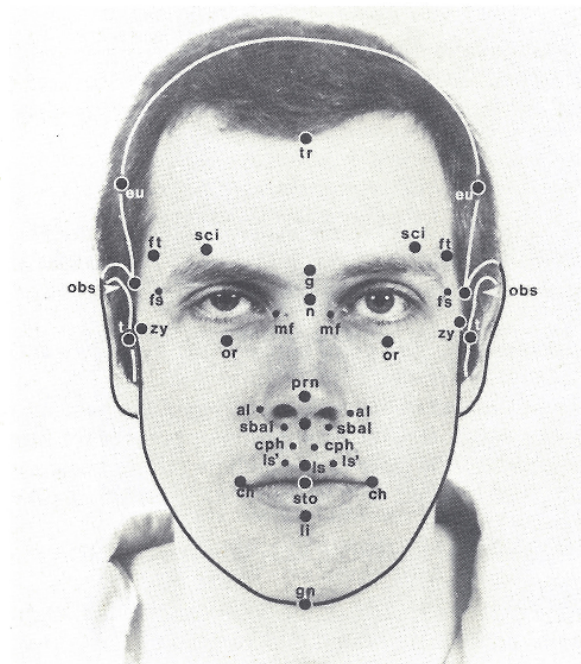
Figure 3.1: Taken from [39]. Some landmarks on the face

Figure 3.2: Taken from [66], originally from [40]. Landmark points on the face to be used in photogrammetry.

Some can be approximated on a photograph by first marking landmark points on the face and then measuring directly on the photograph. Farkas [66] ran an experiment to check the error in such landmarks. He first marked points on the faces of 36 young adults, as seen in Figure 3.2. From these, he took 100 measurements directly on the face, and 60 measurements from life-size front and lateral photos of the faces. In order for a measurement to be considered accurate, it had to be $\leq$ 1mm or 2 degrees from the direct facial measurement. He found that only 20 of the 60 measurements from the photos were 'accurate'. These reliable measurements can be found in Figure 3.3.

We used this information when determining what measurements to use as our features in the metric learning. One of the set of features was from the 20 distances deemed accurate from this experiment. However, it should be noted that even though these features were deemed accurate, that does not mean they are useful or important for our task, in addition to the inherent difference in obtaining them. In the Farkas experiment, he first marked points on the face and then took measurements from photos of those marked landmarks. In our experiment, we do not mark up the actual faces. Instead, we rely on an automatic landmarking algorithm to mark landmarks on the photo of the face. Therefore, many of the
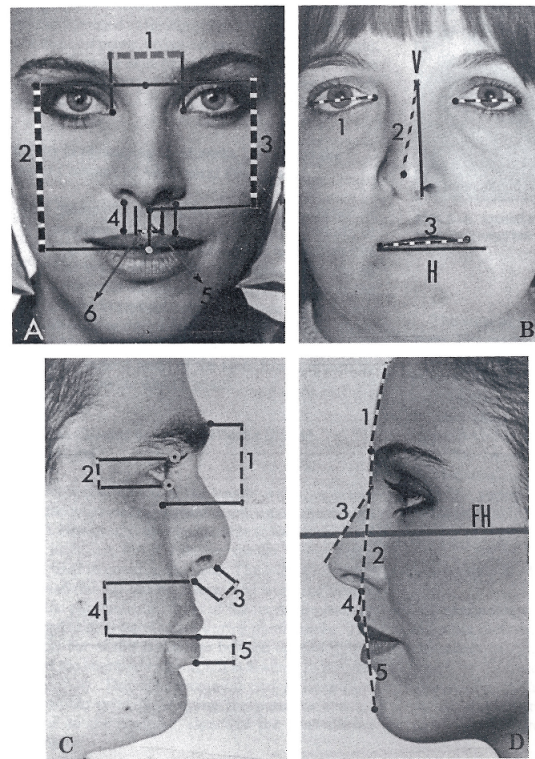
Figure 3.3: Taken from [66]. The 20 landmarks found to be consistent for use in photogrammetry.

landmark positions probably start off in inaccurate positions on the photographic projection; we are unable to check where the bony protrusions or soft tissue is when 'positioning' the marks on the face. This can create problems, e.g., for the tragion and nasion positions, among others. Also, even with the features that are along visual image edges, e.g. the endocanthion and exocanthion, there are natural errors in the automatic landmarking. Finally, Farkas's experiment was done with a careful photogrammetric setup as outlined by Reid and Farkas in Chapter 15 of [66]. Our photogrammetric setup is not as controlled, so there may be slight variations in the angle of the faces when the photos are taken. According to Reid and Farkas, this can create large errors in distances and angles.

The other set of measurements we used from Farkas were estimates of those reported on in the tables of Farkas and Munro [41]. These measurements were taken from children younger than 6 years old, and between 6 and 18 years old. Measurements from points that did not match up, even loosely, with those automatically determined by the landmarking algorithm we used were ignored. We used most features, however, e.g. estimating the nasion point by averaging points 37 and 48 from Figure 2.3.

3.4   Metric Learning

In addition to choosing good features, one should have a good distance metric when comparing images; some features may be more important than others, so it is valuable to learn a weighting over the features. Because we will ultimately be using the metric in order to present a user with a set of images she hopefully thinks are similar to ones they've already chosen, this weighting should be based on perceptual information.

Metric learning takes image similarity information and features and learns a weighting that reflects the similarity information. In this section we will discuss some previous metric learning methods, including the one we used in our own experiments. In the next section we will discuss how we got the perceptually-based image similarity information, as well as our overall experimental setup and results.

Because we are interested in using the metric in an information retrieval setting, not a classification setting, we want to be able to accommodate similarity information that is not class based. *I.e.* , we need to use a metric learning method that doesn't assume similarity constraints based on images having similar/dissimilar class labels. Additionally, because it is more accurate to collect relative judgments than absolute ones (see Section 3.5.1) we want the learning method to be able to use constraints based on pairs of distances, *e.g.* [87, 46, 34, 64]. Ultimately, we ended up using the method of Kulis *et al.* [64], which was based on Davis *et al.* [34]. The method of Frome *et al.* [46] was specifically geared towards computer vision applications and also yielded excellent results, but it was more general than we needed for our setup. In particular, it learned a different parametrization for each reference image, which is useful when images have different numbers of features, *e.g.* when using SIFT features. Since we anticipated using the same numbers of features for all

images, we chose to use Kulis *et al.* for speed and simplicity of implementation. (We had access to the source code.)

These methods and most others learn a Mahalanobis distance. A simple Euclidean distance between two feature vectors can be written as

$$\|x - y\|^2 = (x - y)^T (x - y) \tag{3.1}$$

The Mahalanobis distance may be written as the weighted distance

$$\|Mx - My\|^2 = (x - y)^T A (x - y) \tag{3.2}$$

where A is the positive semi-definite matrix $A = M^T M$. Metric learning of a Mahalanobis distance function can therefore be rephrased as learning M given a set of constraints. Because we are interested in relative comparison constraints, we may rewrite our problem as the distance function and constraints:

$$\text{distance function: } d_A(x, y) = (x - y)^T A (x - y)$$
$$\text{constraints: } d_A(x, y) < d_A(x, z)$$
$$A \succeq 0 \tag{3.3}$$

where the constraints can be thought of as triples describing that x is more similar to y than it is to z. (Section 3.5 will go over how we collected a set of triples to use for the learning.) What makes this problem difficult is that there are two possible problems that can arise: 1) No A satisfies all constraints, and 2) Many As satisfy all constraints. Finding the best possible A turns it into an optimization problem. More specifically, we want to optimize

$$\min_{A,P} D(A, I) + \lambda \sum_{(x,y,z) \in S} P_{xyz}$$
$$\text{s.t. } d_A(x, y) < d_A(x, z) + P_{xyz}$$
$$A \succeq 0$$
$$P_{xyz} \geq 0 \tag{3.4}$$

where P_{xyz} is a constant value or *penalty* for each triple that reflects how important it is that that particular triple constraint be met. Allowing the constraint to be rewritten as $d_A(x, y) < d_A(x, z) + P_{xyz}$ addresses problem 1), where no A satisfies all constraints by loosening the constraints. However, the term $\lambda \sum_{(x,y,z) \in S} P_{xyz}$ in the optimization function pays for these penalties, and λ determines how much penalty to pay. The first term is important if problem 2) arises, *i.e.* many As satisfy all constraints. In this case, we wish to choose an A that is as similar as possible to the identity matrix in order to avoid over-fitting.

Using the identity is equivalent to what's done in a simple Euclidean distance, so it is more natural. One can use a different initialization matrix A_0 instead of I.

The method used by [64] solves Equation 3.4 as a convex optimization problem using Bregman's algorithm [18]. It is an iterative process where at each iteration a different constraint is chosen and a new A_{t+1} is chosen that minimizes $D(A_{t+1}, A_t)$ for that particular constraint using Lagrange multipliers. Because at each minimization step the previously seen constraints do not necessarily remain satisfied, they loop through this process multiple times.

In the next section we will discuss how we used this metric learning framework to try to learn feature weights based on perceptual information. This includes going over how we collected the constraints, what parameter settings we used in the metric learning framework, and what features we used to describe each image x.

3.5   Learning Feature Weights Based on Perceptual Information

We wanted to learn a good perceptually-based metric for image similarity comparisons, using good features. In this section we go over previous work aimed at the same goal, then describe our own experimental setup and results.

Remember we are ultimately using the metric to show to the user faces similar to ones she's chosen. Because she chooses faces based on their similarity to her own lost child, we therefore want to show the user images she would consider similar looking to her child. Because our choice of images depends on our metric, and our metric depends on similarity judgments, our similarity judgments should be perceptually based.

In this section we will first give an overview of previous work related to learning a metric for performing facial similarity based on perceptual information. Then we will go over our experimental setup and results, including how we collected perceptually-based facial similarity judgments, and what features we used in the metric learning.

3.5.1   Previous Work

There has been some previous work on learning a metric to predict facial similarity based on perceptual information. Holub *et al.* [59] ran a series of psychophysics experiments to learn feature importance in predicting facial similarity. He first had subjects rate the similarity of faces in a relative and absolute setting. In the relative setting, the subject was shown a target and 24 other faces. He was asked to choose which of the 24 faces was the most similar to the target. For the absolute task, the subject was shown pairs of faces and asked to rate their similarity on a scale of $1\dot{7}$. They found that inter and between subject consistency was high. In order to determine which method was more accurate, they performed a synthetic test. They first generated 100 random 10 dimensional vectors, and

simulated the relative and absolute scores by looking at the Euclidean distance between the vectors. They performed Multidimensional Scaling, using only the rankings as constraints, and found that the relative ratings were able to better recreate the original distances. This test, however, was highly artificial and depended on the metric used. Additionally, because the features were randomly generated, they were totally unrelated to what might be used in a real facial similarity experiment. Finally, the ratings were not perceptually based. For the next part of their experiment, they had 2 subjects perform relative similarity ratings on 180 images of adult faces. From each task they were able to get 23 triplets, and used those triplets as the input to a metric learning setup in order to learn feature importance. The features they used were image patches of the eyes and mouth, reduced in dimensionality by PCA to 200 dimensions. Their method of metric learning, however, did not assume a positive semi-definite matrix, so the results are not interpretable.

Santini and Jain [86] also ran a psychophysics experiment to get similarity ratings and to learn geometric feature importance. He ran it on sketches, not photographs. The experimental setup involved asking 4 subjects to rank the similarity of 9 sketches to a single target sketch. They were asked to give relative ratings, in this case to put the images in order of similarity to the target, in addition to dividing the images into groups labeled "very similar", "not very similar", and "completely different". They compared the rankings against applying several different metrics to the geometric feature vectors, in order to determine which metric was most accurate. The results were inconclusive.

Cox *et al.* [29], and Papathomas *et al.* [79] used a two alternative forced choice, 2AFC, psychophysics experiment to evaluate what distance metric to use in the Bayesian CBIR Pichunter system (that our CBIR setup is similar to). In a 2AFC setup, the subject is shown a target image and two other images, and is asked to choose which of the two images is more similar to the target. They both claim that the data support the use of a distance metric for comparison, and even discuss using a weighting, but do not go over what the distance metric or the weighting is. They also discuss how the 2AFC experiment matches up with relative and absolute similarity judgments.

3.5.2   Our Experiments

We will now go over our experimental setup for gathering human facial similarity judgments, as well as how we used the judgments in a metric learning framework. It should be noted that in these experiments, we are not trying to claim that this is what the human brain does when judging facial similarity. We only care about learning a metric that will perform well on a test set of similarity judgments.

XM2VTS Experiment

One of the first experiments we ran was metric learning on perceptually based triples to try to learn feature importance. In order to get triples of similarity judgments, we set up
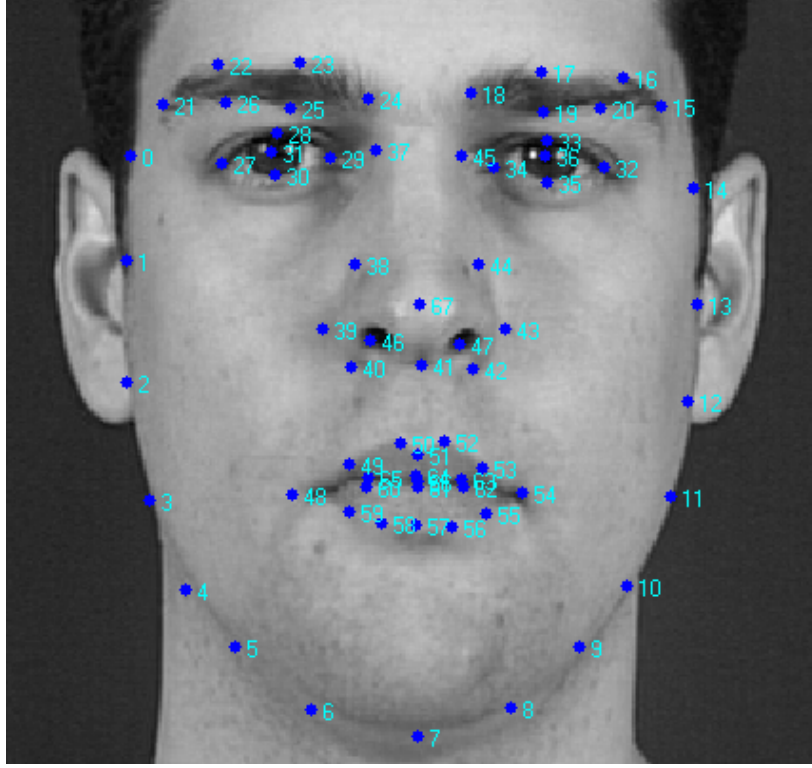
Figure 3.4: The 68 landmark positions used to landmark the XM2VTS database.

a psychophysics experiment. We wanted to use distance and ratio features, so we chose to use images from a face database that had already been landmarked – the XM2VTS database [73]. This database consists of color face images of 295 different subjects. It also had hand-landmarked positions for the corners of the eyes, corresponding with the position of the endocanthion and exocanthion. Cootes *et al.* [26] landmarked additional points on the face so that there were 68 landmark points total, as seen in Figure 3.4.

In the experiment, subjects would look at a series of screens with a reference face in the middle, and 4 comparison faces in the corners. On each screen, they would be asked to rank the comparison faces in order of similarity to the reference face. There was no time limit per screen. The reference and comparison faces for each screen were chosen from the XM2VTS database. We first hand chose 20 male, caucasian reference faces from the database. For each reference face we hand-chose 8 similar faces. Showing each subject $\binom{8}{4}$ screens per reference face would make the experiment too long. Instead, we showed 6 screens per reference face, such that the combinations of comparison faces had a minimal Hamming distance. We ran the experiment over 12 subjects and, for speed, ran half of the subjects with all of the reference faces, and the other half with 10 reference faces each. Subjects took about an hour to finish the complete task. All of the reference face sets were seen by 9 subjects total. Each

of the faces viewed was first eye-aligned (from the centers of each eye), and cropped to the same size.

The inter-rater agreement between subjects was low, but we created 6 triples per comparison set, which meant 36 triples per reference image, and 324 triple responses for that reference image total. (If the data was consistent across subjects, there would be many duplicate triples.)

We ran the metric learning algorithm of Kulis *et al.* [64] on these triples, using feature positions, distances as noted from Farkas [39], and PCA. (We separated the training and test set right before performing the metric learning. Because some of the comparison faces were used for multiple reference images, we ended up having to throw out some of the triples.) We also tried adding extra triples based on the assumption that any of the images not chosen as one of the 8 comparison faces per reference was therefore less similar than any that were chosen. This led to 5670 triples total, although most of those triples were only pseudo-perceptually based.

The results of the metric learning, in addition to the baseline, were inconclusive. There were many problems with how we used the dataset, and with our original methodology in general. There was a lot of noise in the consistency between users. The faces were difficult to ground truth, either because none looked particularly similar, or too many looked similar. The task of ranking four images at a time was also difficult, although this was intentional; it is good to get user data on the edge of what is too difficult of a task. We could have only taken, *e.g.* , the first and last place ranked faces if the second and third place were too inconsistent due to the difficulty of the task. However, noise and the difficulty of the task was compounded by the fact that were had very few triples, and a very small dataset to choose from. We decided to rerun a similar experiment on a new, much larger dataset; that way we would have many more triples to overcome noise.

Experiment on Children's Database

We decided to re-do the experiment, but this time on children's faces from the database used in testing the system (see Chapter 5). In addition, we used more triples, and more similarity data. The experiment can be divided into three sections: 1) Making triples of images and getting similarity data on them, 2) Landmarking the face and calculating distances and ratios, and 3) Running metric learning on the features and triples.

Making image triples means gathering sets of three images, where one is a reference and the other two are used for comparison. Before we can run experiments to get the inequality data on which of the comparison images is more similar to the reference, we first need to determine which sets of three images to use. We call these triples (prior to getting the similarity data on them) 'unweighted' triples.

To make the 'unweighted triples' for Step 1), we used faces from the Parenting database. Because we eventually used these triples in a metric learning framework, we wanted to create both a train and test set. (No image should be in both the train and test set.) We iteratively

added triples to the train then test set, and repeated. We kept track of which images were used as reference and comparison images in both sets.

For ease of explanation, we will discuss here how any given triple was built for the training set. In order to automatically determine the triples, we first randomly chose a reference image not used before in either set. Then, we took the $L2$ distances of the rest of the images (not used in the test set), and sorted the distances in descending order. We chose 11 similar images from the 50 with the closest distances, and made 11 choose 2 triples corresponding to the previously chosen reference image. Then we made a set of triples for the test set, and back and forth. In order to reduce noise, we made separate triples for the two groupings of skin classes: 1-4 and 5-8. In other words, we made triples consisting only of skin 1-4 for train, then skin 1-4 for test, then skin 5-8 for train, then skin 5-8 for test, and then repeated. After we iterated 25 times on test, we made 45 more sets of triples for train only. This resulted in 3850 triples of 487 unique images.

In order to get similarity information on the triples, we ran them through Mechanical Turk. We had each user evaluate 10 different triples at a time, and each triple was evaluated by 10 different users. Users who took less than 30 seconds to complete a task had their information thrown away. From the triples, we only took those that had $\geq 60\%$ agreement.

In order to get the landmarks on the faces, for Step 2), we used the Gu *et al.* face detection system which annotated 83 points on each face. We manually went through the faces and marked which landmarkings were bad, ok, or good. We only used those images with ok or good landmarks. This, coupled with the inter-rater agreement threshold, yielded 290 triples in train and 150 triples in test.

We tried a variety of distances and ratios as features for the metric learning: the 83 landmark positions on the face, distances deemed anthropometrically consistent from a photograph from Farkas [66], the landmark positions minus the eyebrows (also often inaccurate), the distances used in the XM2VTS test, just the left side of the face minus the eyebrows (the right side was sometimes less accurate), and ratios taken from Farkas and Munro [41].

Step 3), the metric learning, was similar to that done for the XM2VTS experiment, and based again on Kulis *et al.* [64]. We passed in the train and test triples, previously deemed to be within the thresholds. We tried a variety of *eta* and *margin* parameters from .001 to 100, using factors of 10. For the features that were distances, some included angles, so we also tried taking the absolute value of the features, and normalizing them across all images. Unfortunately, all the accuracies we had were around 50%, for both the learned and the unweighted. Whenever we got a higher accuracy, it fluctuated over runs, which meant it was overtrained. We also tried just using PCA, as a baseline. Its unweighted and weighted accuracy were both around 50% as well.

We still believe distances and ratios to be important features. Future work in this area should include using more triples. Because selecting only those triples that corresponded to images with good landmarks yielded very few triples (on par with the XM2VTS experiment), we may need to hand-landmark. We could also generate even more triples in the first place, over many more images. It would also be useful to try combinations of features, e.g. distances

and ratios with PCA, although either some sort of normalization or different starting weight matrix would probably be necessary. Also, it's possible that even though we weren't able to learn a set of weights for distances and ratios using this setup, the same set of features could still be helpful in our reunification system. It would be worthwhile to try adding the features and running some user tests.

Chapter 4

Browsing

The initial inputs to the search algorithm are the database of lost children's photos and the semantic information stored with them, as well as the parent's semantic labels for the same set of attributes. The parent's choice of semantic labels will influence the initial ranking of the images in the database, when the images are passed to the search algorithm for browsing.

As the parent searches, she is presented with screens of images. On each screen, she can choose photos similar looking to her missing child. When the parent finds her child in the database, the output of the algorithm will be the other meta-information stored with the image – *e.g.* the hospital name and room number where the child is.

The parent is performing a mental image search when browsing for her child. In other words, there is no initial input from the parent other than semantic information. In order to hopefully shorten the browsing process, the parent needs to iteratively interact with the visual information displayed at each screen. This kind of search falls in the field of Content-Based Image Retrieval, or CBIR.

In this chapter, we will first describe some relevant previous work in CBIR. The algorithm used in this system is based on the browsing system by Cox *et al.* [29], but with the addition of semantic features. Because of this, we will then give an overview of the Cox *et al.* method. Finally we will describe how attributes were added specifically for our system.

4.1 Previous Work in CBIR

To our knowledge, there are no previously implemented systems specifically designed for pediatric reunification. However, there has been previous work on mental image searching. Mental image search is a sub-field of the rich field of Content-Based Image Retrieval, or CBIR [32, 89]. CBIR can be divided into three main categories: open browsing ([44, 97, 43, 53] to name a few), category browsing ([60, 92, 69, 23] to name a few) and target search ([81, 29, 37] to name a few). Open browsing is when the user isn't sure what she is looking for, and can

change her mind partway through. Category browsing is when the user is looking for an image in a particular category. One of the main challenges here is that, even if the user provides an example image, it may be unclear what specific category the user wishes to search for. (For example, if a user provides an image of a red car, she may want to see other red objects, or more cars.) A target search is where the user is looking for a specific thing (object, person, etc.). Since in our case the parent is looking for her specific child, browsing the database of images can be thought of as a target CBIR search.[1]

Target CBIR searches can be further divided into query-by-example (*e.g.* [81]) and mental image search (*e.g.* [29, 37, 47, 4]). In query-by-example, the user supplies a photo of the specific thing she's looking for, and then tries to find more of that same thing. An example of this is Pentland *et al.* 's Photobook search engine [81]. In a mental image search, since there is no initial query image, nor does there have to be any initial information of any kind, there needs to be *relevance feedback* [85]. Relevance feedback allows the user to be in the loop, iteratively interacting with the browsing system.

There has been some previous work on mental image retrieval with relevance feedback, in particular, [29, 38, 37, 78]. Zhou and Huang [104] give a survey of relevance feedback approaches in CBIR. Navarrete *et al.* [78] use self-organizing maps. With each iteration of relevance feedback, the weights on the map are updated and those nodes with the highest weights are chosen as the images to display at the next step. The works of Cox *et al.* [29], Fang and Geman [38] and Fang *et al.* [37] are more closely related to the system described here. Cox *et al.* describe a Bayesian method that, at each iteration, updates the probability $P(image\ is\ target|browser\ history)$. Here, the browser history includes all previously displayed images and actions. Fang and Geman and Fang *et al.* are a specific applications of Cox *et al.* using faces. In both cases, the user is presented with a set of images and can select some subset of images similar to her mental target. Neither is specifically tried on children's faces. Additionally, these methods do not exploit extracted or labeled attributes, as we describe in Section 4.3.2.

Exploration versus exploitation is an important issue in relevance feedback. When determining which images to present to the user at each iteration, a CBIR system can either try to "zoom in" to areas most closely related to the images chosen by the user so far, or can try taking a more global approach. Exploitative approaches select the images with the highest probability of being the target conditioned on the user's actions so far, thus effectively zooming into a portion of the search space. Analogously, explorative methods take a more global view of the space, considering more possible images even if those images are less likely under the model.

Glowacka *et al.* [47], Auer and Leung [4], and Cox *et al.* [29] study the exploration versus exploitation trade-off. Glowacka *et al.* use a Dirichlet distribution to model the

---

[1]It should be noted that some CBIR methods are relevant to multiple categories, *e.g.* [29] can be used with open browsing. However, because open browsing is difficult to evaluate empirically, the evaluation is generally performed on the other categories.

system's beliefs about which image is the target. By combining global and local approaches in updating the user model and choosing the display, they have a nice trade-off between exploration and exploitation. Synthetic experiments show that this model outperforms the *PicHunter* system of Cox *et al.* for a particular setting of the system. Our method follows the search algorithm of Cox *et al.* , however, the approach of Glowacka *et al.* could also be used and modified in a similar fashion to incorporate attribute information.

Auer and Leung [4] analyze exploration versus exploitation in conjunction with two paradigms for modeling user action: those based on selecting a single image as similar, and those that give a binary labeling to displayed images as either relevant or irrelevant. (These two paradigms are akin to the absolute and relative distance models of Cox *et al.* ; see Section 4.2.) They then go over three display algorithms, and how they trade-off exploitation versus exploration when used with the two models.

Cox *et al.* [29] discuss both explorative and exploitative methods, and the approaches are described in Section 4.2.

4.2   Cox *et al.* Method

The browsing system most similar to ours is the *PicHunter* system of Cox *et al.* [29]. There are three main parts to the retrieval problem as outlined by Cox *et al.* : 1) Which images to show at each iteration, 2) How the user interacts with the data, and 3) How we interpret the user's feedback. Figure 4.1 shows the three steps, denoted by $1), 2), 3)$, and defines some terms. Here, T is some mental representation of the thing the user is trying to find, and $T_1...T_n$ is the entire database of images. At each timestep t of relevance feedback, D_t is the set of images currently shown, A_t is the user action taken on those images, and $H_t = D_1, A_1, ..., D_t, A_t$ is the browser history.

We can rewrite the probability $P(image\ is\ target|browser\ history)$ as

$$P(T = T_i|H_t) = \frac{P(A_t|T = T_i, D_t, H_{t-1})P(T = T_i|H_{t-1})}{normalization} \tag{4.1}$$

$P(A_t|T = T_i, D_t, H_{t-1})$ is called the user model, and is a probability of the user's action at that timestep given that T_i is the mental image, and the entire history. Cox *et al.* make the assumption that the user model is time invariant, so H_{t-1} is actually dropped. $P(T = T_i|H_t)$ is the posterior at timestep t, and is an update of the user model times the prior, $P(T = T_i|H_{t-1})$.

One can see that after the user takes an action, we can evaluate the user model of that action A_t for each $T = T_i$. Because the prior is just the posterior from the previous round (initialized with a uniform distribution), we can easily update the posterior for the current iteration.

Thus, step 3, how Cox *et al.* interpret the user's feedback, can be reformed as how they update the user model. First, we must answer step 2, how the user interacts with the data,
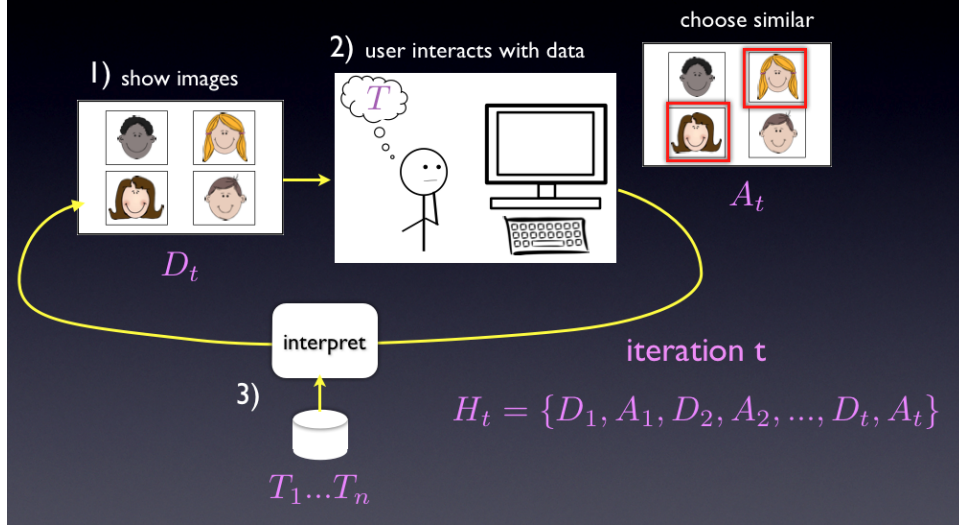
Figure 4.1: Overview of Cox *et al.*

in order to get A_t. Although it depends somewhat on the way the user model is updated, in general the user may select zero or more similar images per screen.

Cox *et al.* describe two main methods for evaluating the user model: Relative and Absolute Distance. In the relative distance framework, the set of selected images (from the current screen D_t) is denoted by X_+ and the set of unselected images is X_-. For each T_i, and each pair $(x_+, x_-), x_+ \in X_+, x_- \in X_-$, they calculate the distance $Dist = d(T_i, x_+) - d(T_i, x_-)$, put it through a sigmoid, and combine. Thus, in the relative distance framework, they are assuming that all images not chosen are specifically not similar to the target image. In the absolute distance framework, only one image is chosen per screen. Here, there are no assumptions on the images that weren't chosen. Instead, only the distance between each T_i and the chosen image X_+, i.e. $d(T_i, X_+)$, is calculated and put through a monotonically decreasing function such that images closer to X_+ have a higher value.

Once the user model is updated, and thus the posterior, Cox *et al.* use the new posterior to determine step 1, i.e., which images to show at each iteration. Cox *et al.* go over two main display algorithms: Most Probable and Most Informative. In the most probable framework, the new display D_{t+1} is chosen from the highest probabilities in the current posterior. The idea behind the most informative method is to minimize the number of expected iterations by choosing the new display that minimizes the entropy of the posterior distribution.

4.3 Implementation Details of Our Browsing Method

4.3.1 General Implementation Details

In our system, "Refining" updates the user model (a uniform distribution if no similar images are selected). We have functionality to update the user model using either relative or absolute distances, but our experiments are with relative distance. The features used in calculating distances between faces are PCA features on the faces after alignment and cropping. As part of future work, we could also add distances and ratios of landmarks points on the face. After the user model and posterior are updated, we show a new set of images D_t. We have functionality to choose each D_t based on highest probability or sampling, but our experiments so far use highest probability. Because we want to reduce the number of photos parents view, once a face has been seen it won't be shown again unless "Back" or "Forward" is used. To do this, we set the posterior of those images to be 0. Once a probability is set as 0, it will remain so. Choosing "Select Random" updates the user model with a uniform distribution, thus keeping the posterior the same as at the previous iteration. However, after "Select Random" is chosen, the next D_t is chosen randomly from any images whose posterior $\neq 0$.

We should note that three buttons "Back", "Forward", "Show Random", and the progress bar were included based on feedback from our collaborators at the Children's Hospital Boston. The parents using the system often said they wanted to be able to fix a mistake – thus "Back" and "Forward". They were also inclined to become frustrated or worried they they hadn't found their child yet if they did not know how many more images they had to look through – thus the creation of the progress bar. Some parents also became frustrated if they kept seeing several screens in a row of children that did not look similar. The "Show Random" button helped alleviate this, and also mathematically helped get the parents out of a search that was too "zoomed-in".

The following subsections go over some more of the details on how we specifically implemented the browsing method.

Updating the User Model

We based the update of the user model on equation 7 from Cox *et al.* [30], which gave implementation details of Cox *et al.* [29]:

$$P_{indep}(A = a_1..a_k|X_1..X_n, T) = \alpha \prod_i p(A = a_i|X_{a_i}, X_{unselected}, T) \qquad (4.2)$$

where $X_{unselected} = \{X_1..X_n\}\backslash\{X_{a_1}..X_{a_k}\}$, $\{X_{a_1}..X_{a_k}\}$ are the selected images, and α is a normalizing term. Assuming each image was chosen independently, they reduce this equation to the product of softmins (Equation 4.3), for each image selected. The softmin equation is

$$P_{softmin}(A = a|X_1..X_n, T) = \frac{exp(-d(X_a, T)/\sigma)}{\sum_i exp(-d(X_i, T)/\sigma)} \qquad (4.3)$$

In this case, X_i will be $\{X_{a_i}, X_{unselected}\}$, *i.e.* the chosen image, and all unselected images. (The function d is the distance metric. For the user tests, we used an unweighted $L2$ distance on image features. We will talk about the features used in the next subsection, and more possible features in Chapter 3.) If multiple images are chosen, then we calculate Equation 4.3 for one chosen image at a time, and mutliply the resulting terms.

Although this is the method we used in our user tests for updating the user model, we also included functionality for another way to calculate this "Relative" update. One can still assume that the images are chosen independently, but chosen as independent sets. Therefore, one can compare all of the k selections at once, against all possible selection of $\binom{n}{k}$, where n is the total number of images seen on a screen. Therefore, X_a in Equation 4.3 would be the set of k images chosen, and each X_i in the denominator would be a possible set of k images. (There would be $\binom{n}{k}$ sets total in X_i.) In order to compare multiple selections at once, one would take the sum of the distances for each element in X_a before putting it through the exponential – *i.e.* the numerator of Equation 4.3 would be $exp(-\sum d(X_a, T)/\sigma)$. Similarly, because each X_i is now a set of sets, the denominator would be $\sum_i exp(-\sum_j d(X_{i_j}, T)/\sigma)$. In trying both settings, however, it appeared that the first method yielded better results, so we used that method in the tests.

Extracting Features

The function d in Equation 4.3 denotes a distance metric. In our experiments we used an unweighted $L2$ distance metric on PCA features. In order to get PCA features, we first needed to align the faces. We transformed and cropped the images so that the center of each eye was at a fixed location. (We assumed that before alignment, the center of an eye was the center of its bounding box.) We first transformed the full size image to reference eye locations so that the entire image fit inside a much larger, set-size image, and so that all of the eye centers lined up. The fixed 'reference' eye locations were such that both the left and right eye centers had the same Y coordinate, placed halfway down the image. The 'reference' X coordinates of the left and right eye centers were also equidistant from the middle of the image.

We then calculated a baseline cropping based on average ratio of the original distance between the eyes to the width of the face. The baseline width of the crop was the average original distance between the eyes times this ratio. The baseline height of the crop was 1.3 times the width. Depending on the amount of cropping we wanted, we would take some percentage of the width and height across all images. For PCA, we used 80% of the width and height. This gave us a bunch of images that were all the same size, where the eyes were at the same locations across all of the images. Because not all faces were the same size, however, some faces had a tighter cropping than others. For PCA we decided to avoid

having any background, which meant that some images had part of the face cropped off. We also used the same alignment technique to make the thumbnails for viewing the images in the search. For these thumbnails, however, we loosened the crop so that no images had part of the face cropped off, although this meant that some images then had more background visible. For the dyad tests, described in Chapter 6 the cropping was 100% of the baseline crop. For the synthetic and disaster drill, we had used the same cropping as the PCA. We changed it for the dyad tests based on user comments (see Section 6.4).

Once we aligned the images and properly cropped them for PCA, we converted them to grayscale and calculated the PCA features. This was done, for any given image, by first subtracting the mean image then multiplying by the basis. The basis was calculated on a set of images, by finding the number of principal components that comprised 95% of the total variance. Additionally, the first three components were removed, since they historically represent lighting differences only.

The PCA basis and mean image was calculated once, offline, prior to deployment of the system. In our experiments, we calculated it on the pre-made database we used to initially populate the search. (See Section 5.1 for a more detailed explanation of the pre-made database and how it was created.) The images used in the pre-made database sometimes varied per experiment, so the basis and mean was re-calculated and stored prior to each experiment. The baseline width and height was also calculated once, on the pre-made database, and stored. The reason why these calculations were done offline was because we needed them to be consistent across all images in the database, including new images to be uploaded. The subsequent PCA feature calculation is performed when each new photo is uploaded, and is stored with the photo in the database, along with the cropped thumbnail to be shown at browsing. The percentage of crop for browsing may be re-set per run, but if a pre-made database is used for experimentation, their crops must also be recalculated and re-stored as well.

Additionally, the σ variable used for Equation 4.3 was calculated once, offline, and stored. Because there is only one σ for all images, it can also be calculated once, on the fly, at the beginning of a parent's search, without having to change any stored information per photo. However, we pre-calculated for speed, based on the images in the pre-made database, per experiment. The σ is the average of the distances between all feature vectors.

In the field, there would be no pre-made database to initially populate the search. In fact, adding a pre-made database would only increase search time and add 'fake' children to the set of children who were actually lost. In the field, we would calculate the baseline width, height, basis, and mean based on some pre-made database and store, even though the faces used in pre-calculation may not represent the faces actually being uploaded. There are a few ways to fix this: First, we may re-calculate this information any time a parent comes in to search; because we are also storing the original, uncropped photo, we may recalculate the feature information for all of the photos at the start of the browsing program. This, however, would probably create considerable lag time before the parent is allowed to search. Instead, the information could be recalculated, offline, once a day, on any of the images currently in

the database, and re-stored. It would be important that no photos were concurrently being uploaded while this calculation was being performed.

4.3.2 Adding Attributes

Previous work in CBIR has combined relevance feedback with semantic attributes. Lu *et al.* [71] did this combination in their iFind system for general images, but did not have a Bayesian framework. They also allow the user to initially input any set of keywords. When the user later chooses similar images during relevance feedback, those image will then become linked with the initial keywords for any subsequent searches. At each round of relevance feedback, new images to be shown are chosen using a ranking which combines keyword association weights and the positive and negative examples chosen in the previous round. Cox *et al.* [29] mentioned adding an initial semantic query in a discussion of future work. Liu *et al.* [70] surveyed some recent advances in combining CBIR with high-level semantics. In particular, they discuss semantic templates which show the user exemplar images based on their relevance feedback, initial keywords, and initial query image (non mental-search). In the CLUE system of Chen *et al.* [20] also shows cluster centers, but is an unsupervised learning framework, and is concerned with updating the similarity measure and thus dividing the images into the appropriate keyword clusters with the most accuracy. There has also been some previous work in using other methods of semantic input with a CBIR system. For example, using an initial query sketch or 3D mockup [19, 3].

We combined attributes with our Bayesian system by initializing the ranking of the images input to the browsing system based on some semantic attributes of the child. At the start of the search, a parent chooses a label for each attribute from a discrete set. (If she isn't sure of the answer or wishes to skip, the label would be 'unknown'.) We call the her labels $L = (l_1, l_2, .., l_m)$ where each l_j is her label for the jth attribute, with m attributes total. Each image i in the database also has labeled attributes stored with it, $L'_i = (l'_{i1}, l'_{i2}, ...l'_{im})$. Because our Bayesian browsing method is based on probabilities over the images in the database, we want the attribute information to influence these probabilities. More specifically, we chose to initially weight the prior for those images with matching labels higher than those without. For example, if L and L_i have two of the same labels, but L and L_j have none of the same labels, then i will have a higher prior value than j. A prior is calculated for each attribute type. Then they are multiplied together and the result is re-normalized. We also add a constant at the end so that none of the probabilities is actually 0. Zero probability is reserved for images we never want to view again. If a parent skips an attribute, the prior for that attribute is uniform across all images.

Attributes with binary labels are assigned prior values of either 1 or S, a *softness value*. This value can be tuned per attribute type. These softness values affect how much the attributes influence the overall search. If all the softness values are 1, then the attributes would make no difference and there is just browsing. If they are all 0, then, depending on the shape of the sigmoid used in the user model, the parent will probably have to look through

all images with exactly the same attribute labels that she listed, before seeing any images with different attribute labels. If we are sure there is no error in the attribute labeling, then it would make sense to make the softness equal to 0. However, as we will discuss in Chapter 5, error can happen even when using "ground truth" attribute labels in the image database.

An attribute with $n > 2$ possible labels will have n discrete values in its prior – i.e. n softness values. The softness values for such attributes are determined by monotonically decreasing functions over discrete variables, where each possible label is assigned a different function. We assume the labels are evenly spaced and ordered semantically, and the input to each function is the number of labels away from the target. The shape of each function is a tunable parameter ρ. The prior for each image i and attribute j in the database:

$$p(T = T_i | l'_{ij}, l_j) = \frac{e^{\frac{||l_j - l'_{ij}||}{\rho}}}{normalization} \tag{4.4}$$

where the normalization term is the sum of the numerator for all i. This way, images with the same label are given a prior of 1. The variance effects the amount of falloff in the value of the prior for neighboring labels. In order to make attributes make less of a difference, we would have a high variance. A low variance could assign a relatively high prior to labels one away, but a much smaller prior to labels several away.

One can consider this the 'discrete' case of creating the initial prior to the Bayesian CBIR system. Remember that the last step here is to multiply together the priors for each attribute type j and normalize. Thus, the complete prior can be written as:

$$p(T = T_i | L'_i, L) = \frac{\prod_j p(T = T_i | l_j, l'_{ij})}{normalization} \tag{4.5}$$

where the normalization term is the sum of the numerator for all i. This terminology is important to remember when we discuss a 'continuous' way to create the prior in Section 4.3.3.

4.3.3 Continuous Attributes

In our tests we used discrete labels over the attributes. In future work, it would be interesting to look into using continuous labels. One way to do this is to not use discrete labels at all; for each attribute, the parent could select from a continuous range of values, corresponding to the range of possible feature vectors. Then the initial prior to the browsing system would be based on the L2 distance between the parent's chosen feature vector and the feature vectors of the images in the database. In the case of skin and eye color, the feature vector is a 3-dimensional color, so the parent could choose from a color wheel in the same color space. Because age is over PCA features, however, it would probably be difficult for the parent to choose this way.

Another more principled approach is to still allow the parent to choose from discrete labels, but to calculate the prior over features through use of generative models; for each category there would be a set of generative models, one for each possible label. Then, when the parent chooses a label, we would evaluate the probability that each image is the target, under the model corresponding to the parent's choice.

In Section 4.3.2 we described the 'discrete' method for calculating the initial prior (Equation 4.5). For the 'continuous' case, the calculation of the prior would be rewritten as follows:

$$p(T = T_i | L, X_i) = \frac{\prod_j p(T = T_i | l_j, x_{ij})}{normalization} \tag{4.6}$$

where, as before $L = (l_1, l_2, .., l_m)$, where each l_j is the parent's label for the jth attribute, with m attributes total. However, instead of conditioning over the labels of the images in the database, we condition over X_i, the feature vectors for image i in the database. Notationally, $X_i = (x_{i1}, x_{i2}, ...x_{im})$, where each x_{ij} is the feature vector for the attribute j.

One way to represent each attribute's prior could be with a Gaussian. In other words:

$$p(T = T_i | l_j, x_{ij}) = \frac{1}{z} e^{(x_{ij} - \mu_{l_j})^T \Sigma_{l_j}^{-1} (x_{ij} - \mu_{l_j})} \tag{4.7}$$

where μ_{l_j} could be the average of the feature vectors for all images in the database with label l_j, and $\Sigma_{l_j}^{-1}$ could be the inverse of the covariance over all feature vectors corresponding to images in the database with label l_j. A Gaussian is just one possible approximation, however, and testing would need to be required.

4.3.4 Why Attributes Plus Browsing is Important

It is important to note why using both attributes and browsing is useful. While attributes have been shown to be useful in searches, they are generic. Depending on the population in the area of the disaster and the number of children affected, thousands of children might have the same set of attribute labels. We could refine the class labels, or add more attribute types, but doing so increases the error (since the ground truth task itself becomes more difficult), and it becomes burdensome for the parent. Thus, the parent needs to be able to look efficiently through the large number of images with the specified labels. Not only that, but extraction is not perfect. Errors in the extraction/classification and differences in user judgment are common. So not only is there a need for browsing, but there is a need to account for and accommodate the initial attribute error with user feedback. Hence, our use of a Bayesian CBIR browsing system.

Chapter 5

Dataset and Attribute Evaluation

5.1   Dataset

Testing the system required collecting a dataset of front-facing children's faces, over a range of ethnicities, eye colors, and ages. These images were used as if they were from the children separated during a disaster. We downloaded thousands of images from the Parenting.com website [27], uploaded as part of a modeling contest. From those, we hand-selected images, trying to only choose ones that were high enough resolution, front facing, and preferably with a natural wide-spectrum indoor or outdoor lighting. Because of the quality of most of the images on the site, however, many of the images chosen still had widely varying lighting and other noise. All of the tests we performed and will report on in Chapter 6 were using some subset of 1213 of these images.

Note that in the field, a standard camera, *e.g.* a Canon PowerShot SD1100 IS, will be used, with flash, preferably indoors, to try to standardize the lighting. We have written an instruction manual (see Appendix A for how to set the camera, and how to take the photos – front-facing, eye-level, little or no out-of-plane rotation, and preferably against a neutral background.)

5.2   Attribute Labels

5.2.1   Ground Truth

In order to get the ground truth attribute labels for each of the browsing sets, we ran Amazon Mechanical Turk [1][1] experiments on all 1213 images from the Parenting.com dataset. We

---

[1]Mechanical Turk [1], a subsidiary of Amazon.com, matches workers and tasks, as submitted by requesters, so that a large number of tasks can be done in parallel, online. Requesters can ask for the same task to be performed by multiple workers, and can set minimum worker requirements (such as rating, experience, etc) in order to filter who is allowed to perform the task. Workers are paid a set amount of money per task,

decided to use eye color, skin color and age as attributes. In the tests we describe in Chapter 6, parents search over some subset of the Parenting.com dataset, in addition sometimes to some small number of photos of children recruited for the test. In all cases, the pre-prepped Parenting.com dataset makes up most of the browsing set. The attributes stored with the Parenting.com images are the "ground truth" labels as determined from Mechanical Turk (or in the case of age, downloaded directly from Parenting.com). Even these ground truth labels, however, have a lot of noise. Below we will report on the amount of inter-rater noise, which is important to keep in mind when evaluating the automatic attribute classification.

For our Mechanical Turk tasks, each image was labeled by 5 different people. Each person was asked to label the following attributes for a given face: eye color, skin color, hair color, gender, and age. See Figure 5.1 to see the possible labels for eye and skin color. Out of the 1213 images, only 1027 had an inter-rater agreement of 60% or more for eye color, and 714 for skin color. In addition, the mean interrater agreement for eye color was .73 with a standard deviation of .21. The mean interrater agreement for skin color was .56 with a standard deviation of .17. We determined that a natural grouping of eye colors would be "Hazel", "Light Brown" and "Dark Brown" as one category, and "Blue", "Green", and "Gray" as another. With this new binary labeling, and grouping the original labels from Mechanical Turk, 1211 had an interrater agreement of 60% or more. The mean interrater agreement for eye color became .92 with a standard deviation of .13. Similarly for skin color, we grouped skin colors 1-4 and 5-8. With this new binary labeling, and grouping the labels from Mechanical Turk, 1210 had an interrater agreement of 60% or more for skin color. The mean interrater agreement for skin color became .91 with a standard deviation of .14. See Figure 5.2 for an example why the eye and skin color may be difficult, even for humans, to hand label.

We should note that for skin color, we also tried having the raters classify the skin as 'light' or 'dark', asking them to take into account all possible skin tones when labeling. The mean inter-rater agreement for these labels was .86 with a standard deviation of .16. Because this was a worse inter-rater agreement than grouping the skin colors, and the labels had potentially more subjectivity, we chose to keep the skin labels $1 - 8$ and then group.

The mean inter-rater agreement for age was .67 with a standard deviation of .18, where the age choices were "0-12 months", "13-23 months", "2-4 years", "5-10 years", "11-12 years". None of the children on the site were over 12 years old. Because age was a field the parents had to fill in when uploading photos to the Parenting.com website, we also had the true ages for all of the children we downloaded (within the specified ranges). Of the 1213 images, 994 had an inter-rater agreement of 60% or more. Taking the mode from the Mechanical Turk results of these 'good' images and comparing against the ground truth from the original website yielded correct ages 66.2% of the time.

For those images that had an inter-rater agreement of $< 60\%$ using the binary grouping for either skin or eye color, we hand-labeled the respective attributes. These hand-labels,

---

which they know before starting, and requesters can rate their performance afterwords.

Figure 5.1: GUI of the choices for eye and skin color.
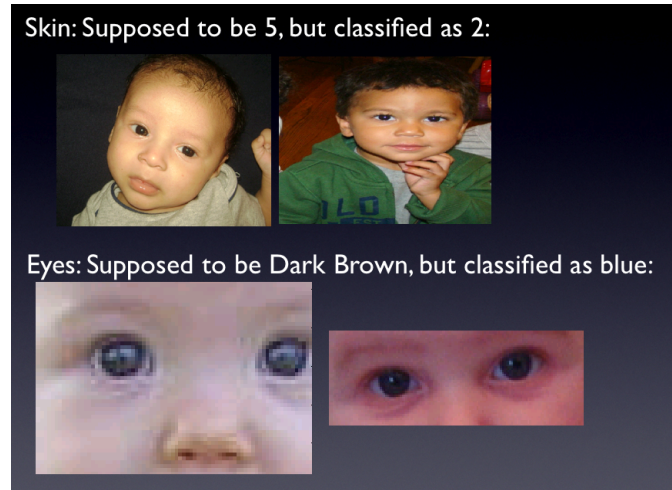
Figure 5.2: Example of why ground truthing eye and skin color is hard. In the top row, the raters classified both children as having a skin color of 5, but the automatic method classified them as having a skin color of 2. Looking at the color patches in 2 and 5, it's unclear which is more accurate. In the bottom row, the raters classified the children as having dark brown eyes, but the automatic method classified them has having blue eyes. Looking at the zoomed-in images of the eyes, they do appear to have a blue tint, but this could be due to the lighting. Again, it's unclear which classification is actually 'correct'.

the mode of the Mechanical Turk labels when the interrater agreement was $\geq 60\%$, and the ground-truth age from the Parenting.com website were the labels used for the Parenting.com dataset when testing in Chapter 6.

5.2.2 Automatic Attribute Results

In this section we evaluate the skin and eye color classifiers. For both skin and eye color, we classify using only two possible labels for each – the grouped labels. Because we used grouped labels throughout the classification, a new image could only be classified with grouped label choices. (A different way to do this would have been to classify based on the original non-grouped labels, and then group after classification. However, this method would be more inaccurate in the same way that the non-grouped inter-rater agreement was poor.)

Some of the tests described in Chapter 6 use automatic attributes. In these cases, when a new image is loaded into the system we run K-nn with a training set of 37 hand-picked images, which we will call the *development set*. These images were chosen from a set that the Children's Hospital collected, along with hand-labeled annotations from one person. The

22	1
3	11

Table 5.1: Confusion matrix for the leave-one-out performance of skin color classification on the development set, with $k = 5$. The rows correspond to the ground truth labels, and the columns correspond to the automatic labels. The top row corresponds to skin color $1 - 4$ and the bottom row corresponds to skin color $5 - 8$.

37 images were chosen because they had similar, broad-spectrum flash lighting, and were strong examples of their class labels – *i.e.* we would expect a strong inter-rater agreement. The development set was similar in image quality to those that will be captured in the field; they were taken using the same camera and flash/lighting specifications.

Here we will present performance results for several tests: performance on the development set, performance on the browsing set, and performance on new images entered during the disaster drill, to be discussed in Chapter 6. The browsing set is that used to populate the database for the disaster drill and other tests. Note that the browsing set has a much wider range of lighting conditions and image qualities. Because the 'ground truth' labels for the browsing set are based on the Mechanical Turk results, it only makes sense to check the accuracy of images with a high enough inter-rater agreement.

Skin Color

For skin color, we classify using only two possible labels, corresponding to either skin color $1 - 4$ or $5 - 8$. First we will show the accuracy of leave-one-out K-nn on the development set. The features used are the average skin color of the cheeks in sRGB, as described earlier. The distance metric was an unweighted $L2$ Euclidean distance. Shown in Figure 5.3 is the correct classification rate versus the number of nearest neighbors. The confusion matrix for $k = 5$ is shown in Table 5.1. We get about 90% performance for skin color with $k = 5$.

We also tested the accuracy of the skin classifier on the browsing set used in the disaster drill and other tests. Again, we tried leave-one-out K-nn. The features, labels, and distance metric were the same as when testing the development set. Note that we only checked the accuracy of images with a high enough inter-rater agreement. Therefore, we only tested the 714 images with $\geq 60\%$ interrater agreement for skin. (We could have also chosen those images with a high post-grouped interrater agreement.) Figure 5.4 shows the correct classification rate versus the number of neighbors for the leave-one-out accuracy. The confusion matrix for $k = 5$ is shown in Table 5.2. We get about 91% performance, but the number of images in each skin color group is very skewed. In classifying images labeled with a skin color of $5 - 8$, the classifier was only accurate 34% of the time.

We also tried checking the accuracy of the skin classifier on the browsing set, this time using the development set for training. Using the development set for training is how the
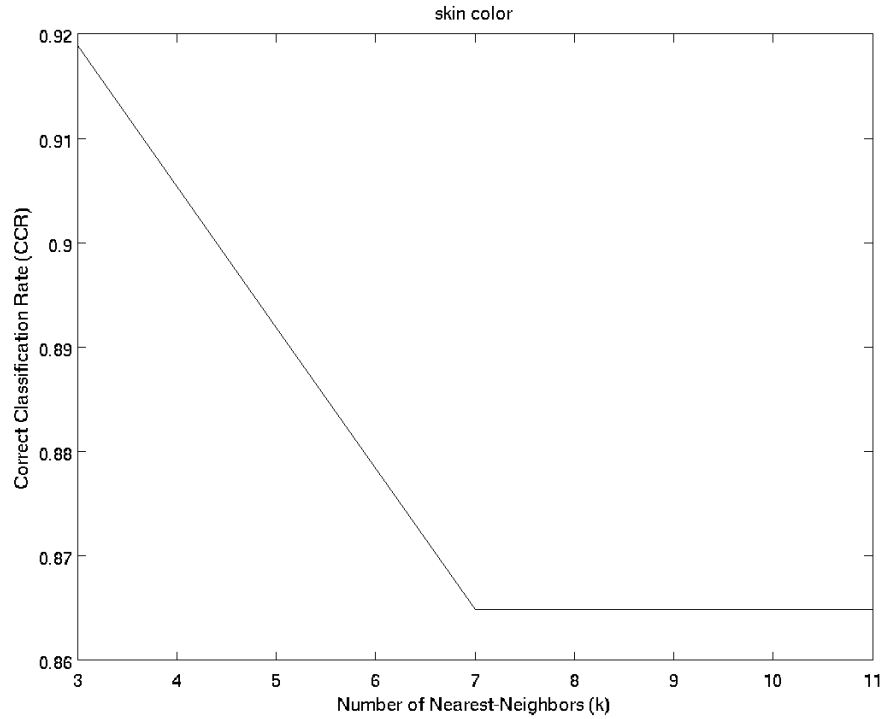
Figure 5.3: Performance of the skin color classifier on the development set of 37 images.

632	18
42	22

Table 5.2: Confusion matrix for the leave-one-out performance of skin color classification on the browsing set, with $k = 5$. The rows correspond to the ground truth labels, and the columns correspond to the automatic labels. The top row corresponds to skin color $1 - 4$ and the bottom row corresponds to skin color $5 - 8$. Only those images with $\geq 60\%$ pre-grouped interrater agreement were tested.
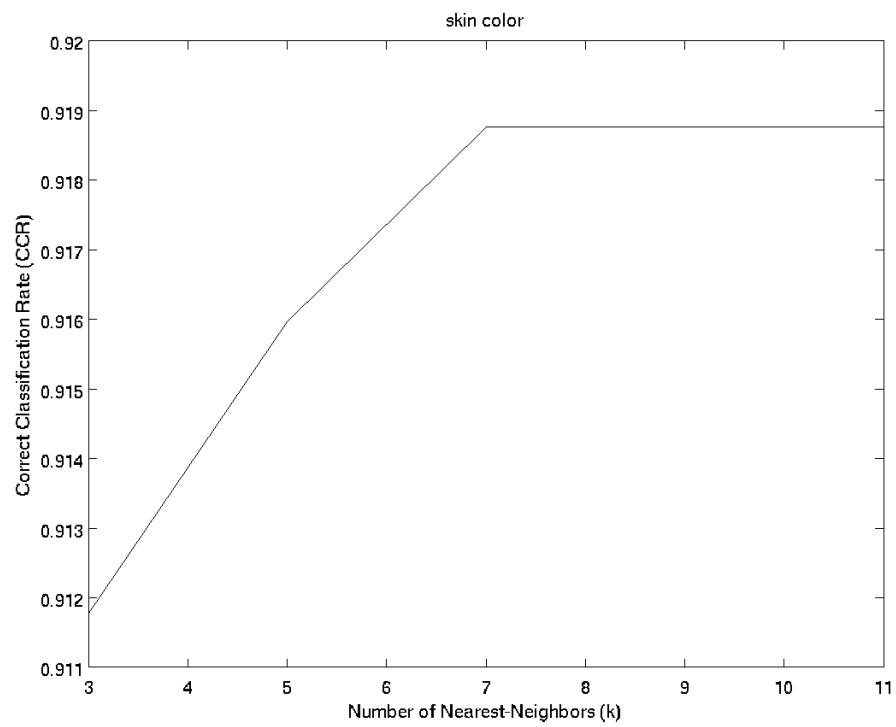
Figure 5.4: Performance of skin color classifier on the browsing set, using leave-one-out K-nn.
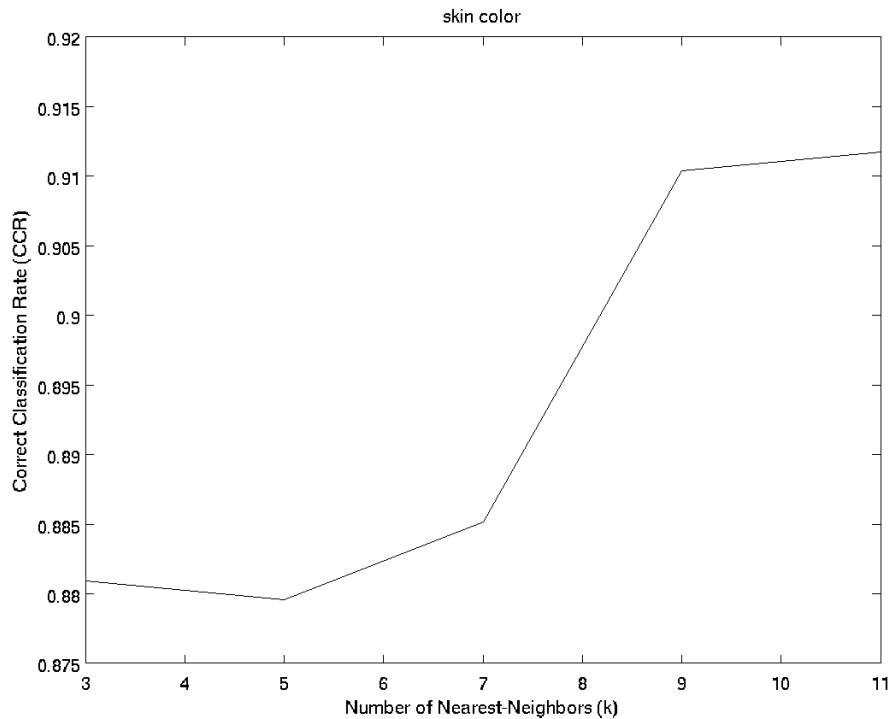
Figure 5.5: Performance of the skin color classifier on the browsing set, using K-nn with the development set for training.

classification is done when a new image is loaded into the system. Again, the features, labels, and distance metric were the same, and we still only tested against 714 images. Figure 5.5 shows the correct classification rate versus the number of neighbors. The confusion matrix for $k = 5$ is shown in Table 5.3. We get about 88% performance, this time with 67% performance on skin color $5 - 8$.

Table 5.4 shows the confusion matrix for how well the automatic skin color classifier performed on the 17 children uploaded for the disaster drill. We used a K-nn classifier with $k = 5$, and the development set for training. The features, labels, and distance metric were the same as the previous tests. Note that although we are reporting the performance of the skin and eye color classification on the images uploaded for the disaster drill, the attributes used in the disaster drill are hand-labeled. The purpose of this confusion matrix is to show the accuracy on images whose quality is similar to what will be used in the field. See Table 6.1 for the confusion matrix of parent labels versus volunteer labels.

585	65
21	43

Table 5.3: Confusion matrix for the performance of skin color classification on the browsing set, using the development set for training, and with $k = 5$. The rows correspond to the ground truth labels, and the columns correspond to the automatic labels. The top row corresponds to skin color $1 - 4$ and the bottom row corresponds to skin color $5 - 8$. Only those images with $\geq 60\%$ pre-grouped interrater agreement were tested.

8	5
0	5

2 parents chose unknown for skin color.

Table 5.4: Confusion matrix for the performance of skin color classification on the 17 new children added in the disaster drill, using the development set for training, and with $k = 5$. (3 children were found, and thus labeled, by both parents, so there were 20 parent labels total.) The rows correspond to what the parent entered, and the columns correspond to the automatic labels. The top row corresponds to skin color $1 - 4$, and the bottom corresponds to skin color $5 - 8$.

Eye Color

For eye color, we classify using only two possible labels, corresponding to either eye colors "Hazel", "Light Brown" and "Dark Brown" as one category, and "Blue", "Green", and "Gray" as another. We will call the set containing the "Hazel" category 1, and the set containing the "Blue" category 2. First we will show the accuracy of leave-one-out K-nn on the development set. The features used are those described earlier. The distance metric was an unweighted $L2$ Euclidean distance. Shown in Figure 5.6 is the correct classification rate versus the number of nearest neighbors. The confusion matrix for $k = 5$ is shown in Table 5.5. We get about 95% performance for skin color with $k = 5$.

We also tested the accuracy of the eye color classifier on the browsing set used in the disaster drill and other tests. Again, we tried leave-one-out K-nn. The features, labels, and distance metric were the same as when testing the development set. Note that we only checked the accuracy of images with a high enough inter-rater agreement. Therefore, we only tested the 1027 images with $\geq 60\%$ interrater agreement for skin. (We could have also chosen those images with a high post-grouped interrater agreement.) Figure 5.7 shows the correct classification rate versus the number of neighbors for the leave-one-out accuracy. The confusion matrix for $k = 5$ is shown in Table 5.6. We get about 87% performance.

We also tried checking the accuracy of the eye color classifier on the browsing set, this time using the development set for training. Again, the features, labels, and distance metric
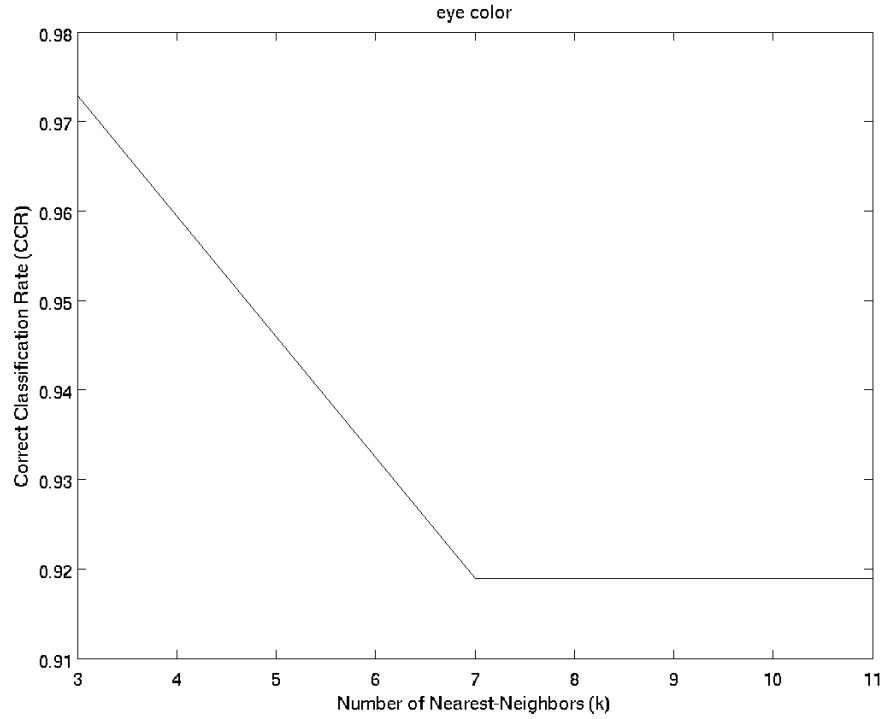
Figure 5.6: Performance of the eye color classifier on the development set of 37 images.

18	1
1	17

Table 5.5: Confusion matrix for the leave-one-out performance of eye color classification on the development set, with $k = 5$. The rows correspond to the ground truth labels, and the columns correspond to the automatic labels. The top row corresponds to eye category 1, *i.e.* "Hazel", "Light Brown" and "Dark Brown", and the bottom row corresponds to eye category 2, *i.e.* "Blue", "Green", and "Gray".
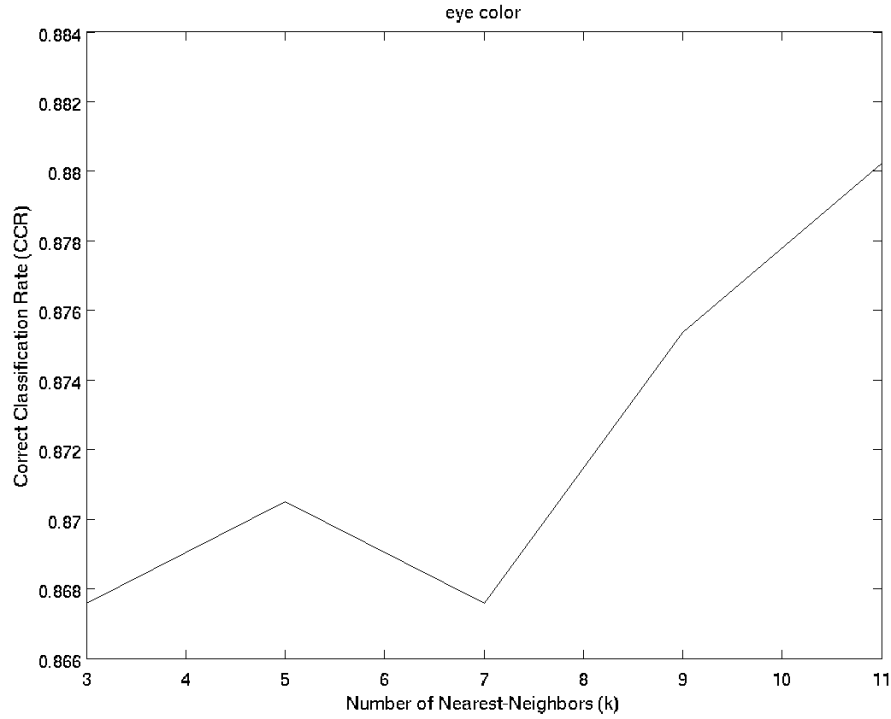
Figure 5.7: Performance of eye color classifier on the browsing set, using leave-one-out K-nn.

$$\begin{array}{|cc|} \hline 670 & 55 \\ 78 & 224 \\ \hline \end{array}$$

Table 5.6: Confusion matrix for the leave-one-out performance of eye color classification on the browsing set, with $k = 5$. The rows correspond to the ground truth labels, and the columns correspond to the automatic labels. The top row corresponds to eye category 1, *i.e.* "Hazel", "Light Brown" and "Dark Brown", and the bottom row corresponds to eye category 2, *i.e.* "Blue", "Green", and "Gray". Only those images with $\geq 60\%$ pre-grouped interrater agreement were tested.
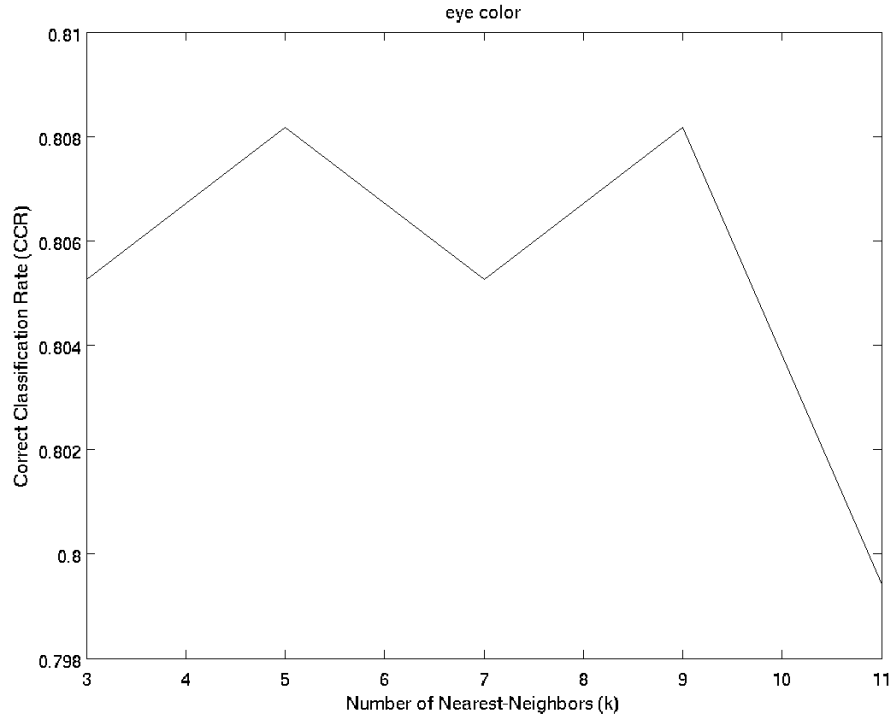
Figure 5.8: Performance of the eye color classifier on the browsing set, using K-nn with the development set for training.

were the same, and we still only tested against 1027 images. Figure 5.8 shows the correct classification rate versus the number of neighbors. The confusion matrix for $k = 5$ is shown in Table 5.7. We get about 81% performance.

Table 5.8 shows the confusion matrix for how well the automatic eye color classifier performed on the 17 children uploaded for the disaster drill. We used a K-nn classifier with $k = 5$, and the development set for training. The features, labels, and distance metric were the same as the previous tests. See Table 6.1 for the confusion matrix of parent labels versus volunteer labels.

562	163
34	268

Table 5.7: Confusion matrix for the performance of skin color classification on the browsing set, using the development set for training, and with $k = 5$. The rows correspond to the ground truth labels, and the columns correspond to the automatic labels. The top row corresponds to eye category 1, *i.e.* "Hazel", "Light Brown" and "Dark Brown", and the bottom row corresponds to eye category 2, *i.e.* "Blue", "Green", and "Gray". Only those images with $\geq 60\%$ pre-grouped interrater agreement were tested.

11	1
0	8

Table 5.8: Confusion matrix for the performance of eye color classification on the 17 new children added in the disaster drill, using the development set for training, and with $k = 5$. (3 children were found, and thus labeled, by both parents, so there were 20 parent labels total.) The rows correspond to what the parent entered, and the columns correspond to the automatic labels. The top row corresponds eye category 1, *i.e.* "Hazel", "Light Brown", and "Dark Brown", and the bottom row corresponds to eye category 2, *i.e.* "Blue", "Green", and "Gray".

Chapter 6

Tests/Results

In order to test the system we ran three sets of experiments with the Children's Hospital Boston: 1) a disaster drill to test the overall 'attributes plus browsing' mental image search system on actual parents, 2) a series of tests on 'synthetic' parents to evaluate the effectiveness of browsing, and 3) a set of dyad tests, run on actual parents, designed to evaluate the 'attributes plus browsing' system with automatic attributes, and to demonstrate the merit of browsing in a realistic setup. In this chapter we will go over all three sets of experiments, and the implications for each.

6.1   Disaster Drill

One of the difficulties in testing the browsing system is that it's very difficult to perform a mental image search unless one is very familiar with the person he/she is looking for. Since we're searching for children, aside from teachers or other care workers who see the same children every day, parents/guardians are probably the only group of people familiar enough with a child to perform a purely mental image search. Since we are ultimately gearing the system to parents/guardians, it is therefore important to run the experiment using actual parents.

To do this, we ran a complete disaster drill in collaboration with the Children's Hospital Boston. For this test, we used real parents and ran the mental image search as we might in the field using attributes and browsing, although the attributes of the children enrolled were hand-labeled by a volunteer, and not automatically determined. Seventeen children[1] from 8 families were enrolled. The parents and children were both given family identifiers, and the children were given extra identifiers according to their age. The children were taken into a room where they had their photos taken. The volunteer taking the photos hand-noted the children's eye and skin color, and asked them their age, if they were willing to provide it.

---

[1]Technically there were 18 children, but one of the Reunite sheets was either lost or not marked for the 18th child, so that child's photo was never uploaded.

For each child, the volunteer recorded the necessary information on a Reunite sheet, similar to the one pictured in Figure 2.2. When the volunteer was done, the information and photos were uploaded to the system and onto three laptops. The person in charge of uploading entered the attribute information as it had been noted on each child's Reunite sheet. That person could have also hand-chosen the attributes herself while looking at each photo during the upload.

The laptops were then taken to a separate room, where the parents filed in as space permitted. The parents had been prepped in a separate room by social workers, who evaluated their mental stress. The procedure followed was the same as if there had been an actual disaster. The parents did not work the system on their own, but instead pointed to the screen and communicated with a volunteer. Each volunteer and parent pair was monitored by a social worker, and some of the parents were instructed to be argumentative. The volunteers read standardized instructions (see Appendix B). Because some families had more than one parent participate, there were 20 searches total. Figure 6.1 shows a photo of the drill in progress.

The searches were performed on 730 images: 713 images chosen from the pre-made Parenting.com set, plus the 17 children who were participating in the drill. The images from the Parenting.com set were chosen from the full 1213 so that the eye, skin, and age distributions were roughly even. Parents saw 9 images per screen. The first screen of images was chosen randomly from those images with a maximum initial prior value. If there were fewer than 9 images with that value, images would then be chosen from those with the next highest prior value, and so on. For the disaster drill, however, this was never the case.

For the drill, we used eye and skin color, and age attributes. For skin and eye color, binary labels were used. The labels for the dataset (aside from the new children enrolled) were determined by the Mechanical Turk results. For those images with an inter-rater agreement $> 60\%$, the mode label was used, and then grouped into the appropriate binary class. Images with too low an inter-rater agreement were hand-fixed with the appropriate binary classes. When a parent was asked to enter skin and eye color, he or she chose labels from the finer-level as input, and these labels were then grouped into the corresponding binary value. The softness values used for the eye and skin attributes were both .6.

For age, the dataset (aside from the new children enrolled) was labeled using information directly from the original Parenting.com website the photos came from – the age, within a range, was stored with the photos on the site, as it was a required field for parents to fill in when uploading the photos. Because there were many fewer children 5 years or older on the site, we condensed those age ranges into one. Thus, the age ranges used for labeling both the original dataset and children added during the drill was: "1-12 months", "13-23 months", "2-4 years", "5 years or older". When the parents were asked to input the age, they chose from this same set of ranges. The variances used to determine the softness values for age ranges were [5 5 3 2].

Parents looked at an average of 7.10 screens with a variance of 6.83 to find their child, with chance performance being 40.6 screens. This experiment shows the validity of the overall

Figure 6.1: Photo taken during the disaster drill. Sitting closest to the laptop is the volunteer, helping the parent (also sitting) through the system. We decided it was best for the volunteer to be the one to physically make the choices, based on what the parent points to or says. On the floor and standing up are two social workers, calming the parent down. (The parents in the drill were instructed to act upset.)

13	0
2	3

12	1
0	7

0	0	0	0
0	2	0	0
0	0	3	0
0	0	0	15

2 parents chose unknown for skin color.

Table 6.1: Confusion matrix for the performance of the volunteer-labeled attributes versus the parents' ground truth attributes, for the images uploaded during the disaster drill. The rows correspond to what the parent entered, and the columns correspond to what the volunteer entered. 3 children were found, and thus labeled, by both parents. The table on the **left** is for skin color. The first row corresponds to images with a ground truth of skin color 1-4, and the second row corresponds to images with a ground truth of skin color 5-8. The **middle** table is for eye color. The first row corresponds to images with a ground truth eye color of category 1, *i.e.* "Hazel", "Light Brown" and "Dark Brown". The second row corresponds to images with a ground truth of eye category 2, *i.e.* "Blue", "Green" and "Gray". The **right** table is for age. The rows correspond to images with the ground truth age categories of "0-12 months", "13-24 months", "2-4 years", "5 years or older". When possible, the volunteer asked the child his/her age and recorded that.

system over random browsing. Looking at the ranking of the images in the prior, browsing seems to take the same number of screens as would be required by randomly showing images with the maximum prior. However, we believe that because the attributes pared down the number of images considerably, there were too few images to draw any conclusions about browsing from the disaster drill. (There were only about 45 images of each eye, skin, and age combination, and browsing works on iterations of user feedback.) With a larger scale test, i.e. browsing over more images – more indicative of a very large disaster – we believe browsing would have a significant effect. The same would be true for a disaster on a homogeneous population. In this experiment, we used hand-labeled attributes, and the parents often chose the same labels for their children as the volunteer who ground truthed them, especially for age[2]. See Table 6.1 to see the 'accuracy' of the volunteer-labeled attributes versus the parents' 'ground truth' attributes. Depending on time or personnel restrictions after a disaster, it might be necessary to use automatic attribute labels. This will likely mean more 'error' in the attributes, and therefore more of a need for the level of softness we used, and for browsing in general. We ran more experiments on real parents in Section 6.3 to evaluate these scenarios.

---

[2]When possible, the volunteer asked the child his/her age. Most children were old enough to be likely to respond properly, which probably influenced why age was always correct. Note that even if age were hand-labeled in the field, it might be done by someone looking at the photos later. This person is less likely to be as accurate.

6.2   Synthetic Parents

We have also run a series of experiments on "synthetic parents" – i.e. people who are not parents, but who are allowed to look at the target image throughout the search. These experiments demonstrate the effectiveness of browsing, albeit not in a purely mental image framework. This section goes over the setup and results of the two main tests.

In the first experiment using synthetic parents, we tested browsing only. The synthetic parents searched over 861 images taken from the pre-made Parenting.com dataset. Even grouping the labels, this set of images had a very uneven distribution of skin colors. We displayed only black and white versions of the images, and the only user options were "Refine" and "Found". We had 7 people run trials; 5 people ran 10 trials, 1 person ran 6, and 1 person ran 4. Because they could see the actual image they were looking for, we asked them to take into account not only identity, but also hairstyle and facial expression. Random performance was 47.8 screens (a total of 861 images, 9 images per screen), and with browsing it took on average 16.3 screens with a variance of 13.4 to find the missing child. For this test, the first screen was chosen randomly.

In the second test, we still used synthetic parents, but we also tried adding attributes. This time the search was performed on 1213 images from the pre-made dataset, such that the distribution of skin colors was more even. The user options were still only "Refine" and "Found", but this time we used color images. Five users evaluated the system with and without attributes. Most users performed 5 trials for each of the two settings, with the exception of one user who only did 2 trials for the browse only setting. See Figure 6.2 for a distribution over eye color, skin color, and age for those datasets. Each triple in the figure represents a (skin color, eye color, age) triple.

For the 'attributes and browsing' setting, only skin and eye color binary attributes were used. The labels for the browsing set were determined the same way as in the disaster drill. When the synthetic parent searched, she chose labels from the finer-level as input, and these labels were then grouped into the corresponding binary value. She chose the attributes being able to look at the target image. However, errors could still be introduced if there was a difference between what the synthetic parent thought and the consensus on Mechanical Turk.

For these tests, the prior was set to "all or nothing"; *i.e.* if the image had both of the labels the same, it was given a high score, otherwise it was given a low score. The "softness" was set to the equivalent of $1 - 10^{-6}$; *i.e.* , error in attributes would make a large difference. The first screen was chosen from images with the maximum initial prior. However, for this test, the initial prior was purely sorted in descending order, and images with the same maximum prior were not randomly permuted. Thus, it was likely that two different users could start with the same initial screen.

Because there were 1213 images in the browsing set with 9 images per screen, chance performance was 67.4 screens. Browsing only yields 23.5 screens on average with a variance of 22.2, and browsing with attributes yielded 16.3 screens with a variance of 15.7.
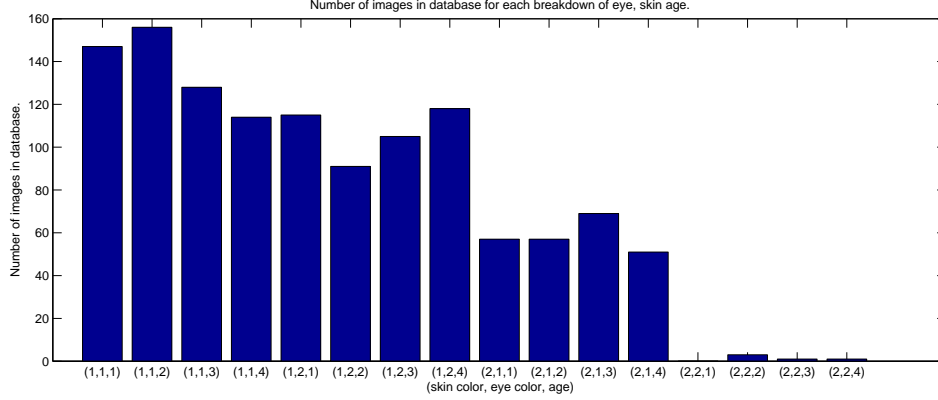
Figure 6.2: Data distribution of the full browsing dataset. Each triple represents a (skin color, eye color, age) triple. 1 and 2 for skin color represent the grouping of the original skin color labels $1 - 4$ and $5 - 8$ respectively. Eye color 1 represents the grouping of the original eye colors "Hazel", "Light Brown", "Dark Brown", and eye color 2 represents the grouping of the original eye colors "Blue", "Gray", "Green". The ages $1 - 4$ represent the ages "0-12 months", "13-24 months", "2-4 years" and "5 years or older".

6.3 Dyad Tests

6.3.1 Dyad Setup

In order to more fully test the system and determine the efficacy of browsing, automatic attributes, and a combination of both, the Children's Hospital ran a series of dyad tests. In particular, we wanted to see whether browsing aids in the presence of noisy attribute labels, or when given a large, homogeneous population. The tests were performed over series of a few weeks, and not done all at once, as with the disaster drill. When the parents came in to search, they were read standardized instructions like in the disaster drill (see Appendix B).

With the Children's Hospital, we set up two different dyad tests. The first dyad test, which we will call *Main Dyad*, had the parents run the system twice: once where the system used attributes and browsing, and once with just browsing. (As a reminder, attribute and browsing means using the attribute information to pre-set the initial prior, and just browsing means that the initial prior is uniform.)

Each child was enrolled into the system similarly to what was done in the disaster drill; the child was given a unique identifier, and his photo was taken out of view from the parent and uploaded to the system. Unlike the disaster drill, however, the child's attributes were automatically determined. The attributes used were skin color, eye color, and age.

For skin and eye color classification of each new image, the feature vectors were first automatically extracted using the techniques described in Section 2.1.2. They were then assigned labels using the classification scheme from Section 5.2.2 – K-nn with $k = 5$, and

trained over the development set of 37 hand-picked images. The class labels assigned to the training set were the grouped binary labels – skin category 1 or 2 (corresponding to skin colors $1-4$ and $5-8$ respectively), and eye category 1 or 2 (corresponding to eye colors "Hazel", "Light Brown", and "Dark Brown" for the first, and colors "Blue", "Green", and "Gray" for the second). See Table 6.2 for the skin and eye color confusion matrices of the children enrolled in this dyad test.

We did not yet have the age classifier from our anthropology collaborator at Johns Hopkins, but test runs of the system with Children's Hospital revealed that parents became frustrated looking at images of children with very different ages than their own child. We decided to add an automatic age classifier to the system with the main goal of avoiding these extreme cases, not in perfecting the per-class accuracy of the age estimation. More specifically, it was acceptable if the classification was one label off, but not a few; *e.g.* if the child was > 5, he shouldn't be classified as < 1. The age labels used were the same as in the disaster drill: "0-12 months", "13-24 months", "2-4 years", and "5 years or older". Age was automatically determined using a Gaussian process classifier [82] on the PCA features, with the entire 1213 image browsing database as the training set. See Table 6.2 for the age confusion matrix for children enrolled in this dyad test.

When each parent came in to search, she first entered her child's identifier and attribute information. The system than randomly chose whether the first search would be attributes and browsing or just browsing. After the first search was complete, a dialog box would pop up asking the parent if she was ready to move onto the next test, and then the parent would search again. To the parent, both tests looked the same. The parent was told that she would be looking for the same child, but that the photo used might be different. However, the same photo was used.

For both methods, the parent browsed over 1214 images – the full 1213 image Parenting.com browsing set, and her child. When a parent browsed, she would be browsing over the full 1213 Parenting.com database, with the addition of her child. No other children enrolled in the test were shown to the parents. The labels assigned to the browsing database (aside from the new children enrolled) were taken from the Mechanical Turk results the same as in the disaster drill. Parent were shown 9 images at a time, and the skin and eye softness was set to .6, and the age variance was set to [5 5 3 2], which worked with the level of accuracy we wished to achieve from our classifier. We will show the results of 19 runs of the Main Dyad test. (In this case, 19 parents searched for 19 different children. No parent searched for more than one child, and no child was looked for by more than one parent.)

The second dyad test, which we will call *Homogeneous Dyad*, was performed on a homogeneous population; all of the photos in the database searched over had the same set of attributes, as well as the children recruited. We performed this test in order to simulate searching over a large, homogeneous population, or to simulate searching over a very large population after attributes had been selected. (Because of our softness parameter, technically after attributes were selected, images of children from other sets of attributes could be shown. In order to say that this dyad test was similar to searching after selecting attributes,

we would have to assume that the attributes selected were "correct" – the same as what was stored with the child – and that the softness was set to 0.)

We chose the homogeneous population based on the distribution of children we had in the Parenting database. The largest sample was of children with skin group 1, eye group 1, and age group "13-24 months" at 156 children. However, using only those images in the database would give a random performance of 9 screens, which we believe was too few to be able to take advantage of the relevance feedback. We decided to also add images of eye group 2, and "0-12" months, ending up with 509 images total.

Because of the difficulty of finding parents with eligible children to perform the task, some parents were asked to provide photos of their children when they were between "0-24" months. In these cases, the volunteer adding the images to the system chose one and told the parent around what age to picture her child. (This was mostly important to distinguish between infant and 2 years old, though in an ideal situation the database would be more homogeneous.) In one instance the volunteer was also the user, but in this case she waited a day before searching in order to forget the details of what her child looked like in the photo. Two parents also searched for a child (the same child) whose photo was from when she was slightly older than 24 months.

Children were enrolled into the system similarly to the Main Dyad test; their photo was taken out of view from the parent and they were uploaded to the system, along with their identifier. Because the population was homogeneous, we did not extract attributes. Similarly, when the parents came in to search, they were not asked to enter attributes.

The parents browsed over 510 images – the 509 images chosen from the Parenting.com database, and their own child. No other children enrolled in the test were shown to the parents. Parents were shown 9 images at a time. Overall, there were 7 runs of the Homogeneous Dyad test, performed by 4 parents searching for 4 children.

For both the Main Dyad and Homogeneous Dyad tests, the first screen of images was chosen randomly from those images with a maximum initial prior value.

6.3.2   Dyad Results

In this section, we will show the results of the dyad tests, and discuss some of their implications. For the Main Dyad test, remember that each parent searched for her child using both the 'attributes plus browsing' and the 'just browsing' methods. Because the database was the same for both tests including the image of the new child uploaded, the image features were also the same, and it is reasonable to compare the search time and number of screens between each. In the discussion of the Main Dyad test, we will also often compare the number of screens seen to a 'just attributes' test. This test wasn't run, but estimated from the attribute information, and defined as follows: Similar to the 'attributes plus browsing' method, attribute information is used to set the initial prior for browsing. However, instead of browsing using CBIR, the images would then be shown in descending order of the initial prior. Images with the same prior value would be shown in random order. Therefore, the

10	3
0	6

12	5
1	1

1	2	0	0
1	0	0	0
0	2	4	1
0	0	2	6

Table 6.2: Confusion matrix for the performance of the automatic attributes versus the parents' ground truth attributes, for the images uploaded during the dyad tests. The table on the **left** is for skin color. The first row corresponds to images with a ground truth of skin color $1 - 4$, and the second row corresponds to images with a ground truth of skin color $5 - 8$. The **middle** table is for eye color. The first row corresponds to images with a ground truth eye color of category 1, *i.e.* "Hazel", "Light Brown" and "Dark Brown". The second row corresponds to images with a ground truth of eye category 2, *i.e.* "Blue", "Green" and "Gray". The **right** table is for age. The rows correspond to images with the ground truth age categories of "0-12 months", "13-24 months", "2-4 years", "5 years or older".

number of screens needed to find the ith child in the database, using the 'just attributes' method, and assuming an initial prior P is: $|\{x : x \in P, P(x) > P(i)\}| + \frac{|\{x : x \in P, P(x) = P(i)\}|}{2}$. (Here, $|\cdot|$ is used as cardinality.) In other words, the total number of screens would equal the number of images whose prior is $> P(i)$ plus half the number of images whose prior $= P(i)$. Because images of the same prior are shown in random order, this total is actually the average over multiple searches for the same child.

It should be noted that in this discussion, there will not be comparisons of absolute performance values across different subsets of runs; *e.g.* it is not interpretable to talk about how the mean of 'just attributes' was higher in Figure 6.13 than in Figure 6.3. When, in this discussion, there is talk about performing 'relatively worse' or 'relatively better', the only statistic being compared across runs is the *relative* performance between methods. *E.g.* , looking at the same figures, one could say that 'just attributes' performs relatively worse than the other two methods in Figure 6.13 than in Figure 6.3.

Main Results

Figure 6.3 shows the distribution of the number of screens seen for 19 runs of the Main Dyad test. The dotted horizontal line denotes chance performance, which is 67.4. (1214 images, divided by 2, seen 9 images per screen.) Running a T-test of each method against chance performance shows statistical significance for both, with $3.51e^{-7}$ as the p-value for 'attributes plus browsing', and $3.5e^{-3}$ as the p-value for 'just browsing'. The mean number of screens seen using 'attributes plus browsing' was 27.4 with a variance of 23.1. The mean number of screens seen using 'just browsing' was 41.4 with a variance of 37.7. One can see that the mean number of screens for 'attributes plus browsing' is better than either method alone, although the distributions appear similar.
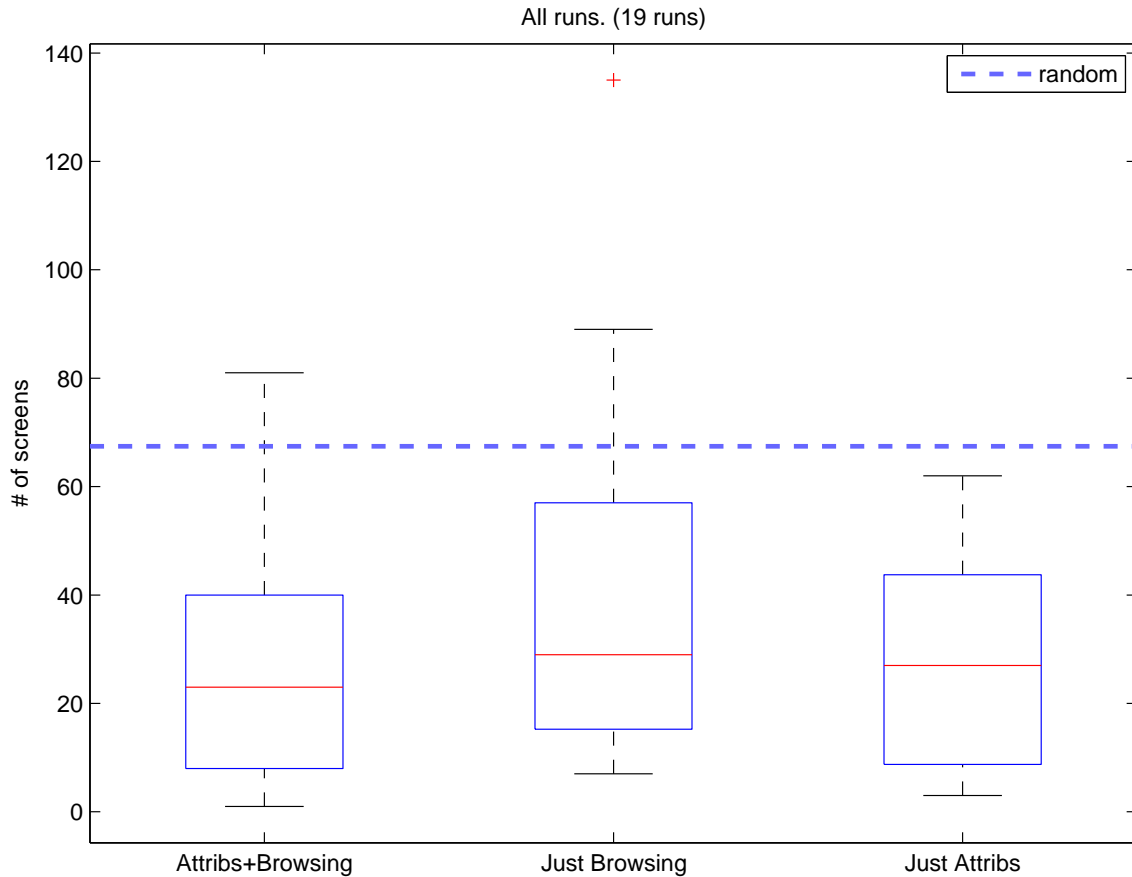
Figure 6.3: Box plot of the number of screens seen over all runs of the Main Dyad test.

In the Main Dyad test, neither 'attributes and browsing' nor 'just browsing' performed statistically significantly better than 'just attributes'. However, looking at Figure 6.2, one can see that once all three attributes are entered, this significantly pares down the database. If the automatic attributes are correct, then the number of screens for 'just attributes' would be the number of images in the database with those attributes divided by 18. In that case, the database is still not large enough to be able to confidently measure the effects of browsing. Section **Attribute Errors** below will go over performance when there is automatic attribute error.

Figure 6.4 shows the distribution of the number of screens seen for the Homogeneous Dyad test. The dotted horizontal line denotes chance performance, which is 28.3. (510 images, divided by 2, with 9 images presented per screen.) It is important to note that over the homogeneous dataset, chance performance is equivalent to the performance of 'just attributes'. Running a T-test of browsing against the chance performance shows statistical significance, with a p-value of $3.4e^-2$. The average number of screens seen using browsing
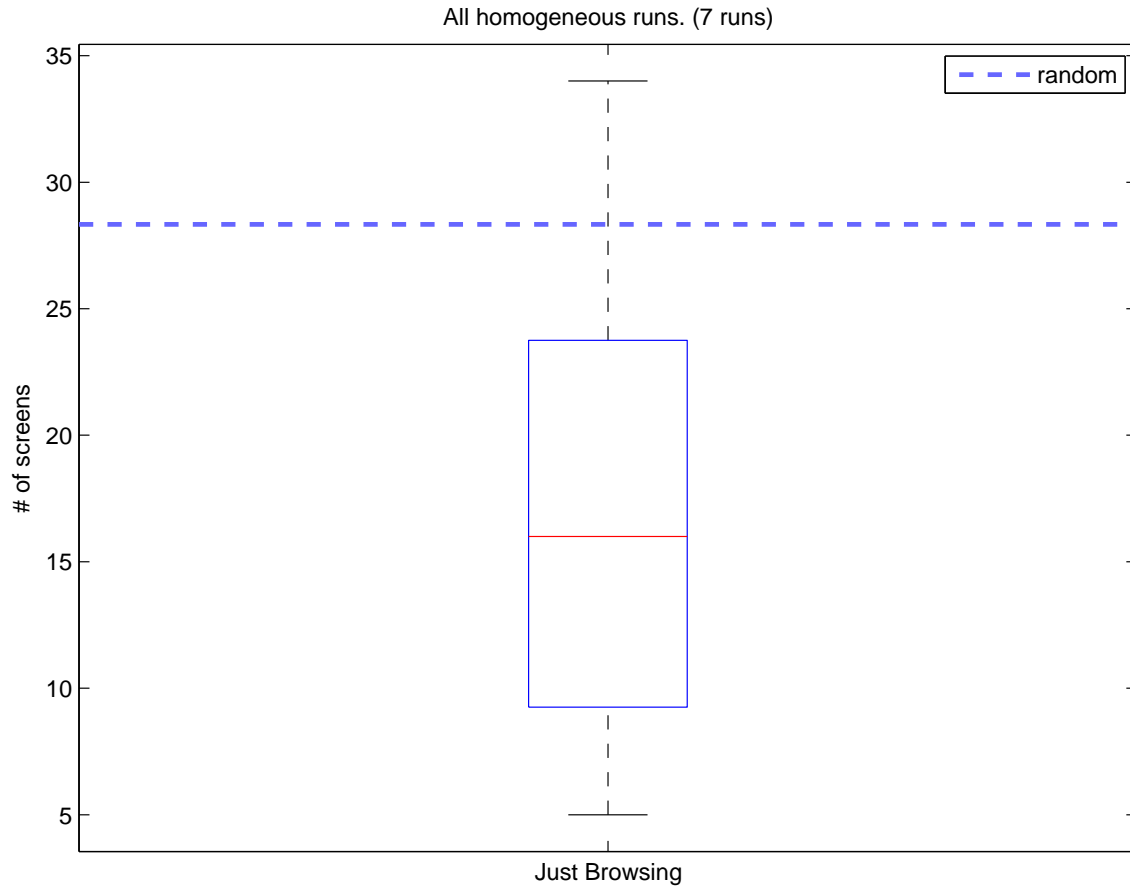
Figure 6.4: Box plot of the number of screens seen over all runs of the Homogeneous Dyad test. Here, chance performance is equivalent to the performance of 'just attributes'.

was 17.8 with a variance of 10.1.

One can see that using automatic attributes and browsing is better than randomly showing images. Also, in the case where a large number of images have the same set of attributes, browsing does better than 'just attributes' (which in this case is equivalent to random performance over the homogeneous dataset). This would imply that, if performed over a much larger heterogeneous browsing database, 'attributes plus browsing' should perform significantly better than 'just attributes'. This test would be an interesting future work.

There was one outlier case for the browsing run in Figure 6.3. In this case, the parent did not click on any similar images. In the absence of user feedback, images were shown in the order in which they were added to the database, although a random presentation of these images could also be used.[3] The children enrolled in the dyad test were always added to the

---

[3]This may have introduced a slight bias in our results in cases where parents took several screens before beginning to give feedback, especially since children, for the most part, were added to the database in order

end of the database, which explains why the parent looked through all of the images, and why it is a strong outlier. One could ask, however, why the parent chose no similar images. In this case, the child added was syndromic, and in that respect unlike any of the images in the database, which could be a reason. This raises the question of whether being syndromic should itself be an attribute. See Section **Interactivity** below for a more in depth discussion on parent 'interactivity'.

We also measured the total time the parent took to find her child, starting when the parent sees the first screen of children and ending she selects found (and confirms). Figure 6.5 shows the distribution of time in seconds for all 19 runs. One can see that both the 'attributes plus browsing' and 'just browsing' methods are equivalent. The same is true for the amount of time taken, on average, per screen. There was one datapoint per run, calculated by dividing the total time by the total number of screens seen. Figure 6.6 shows the distribution, and again it's roughly equivalent.

In future tests, it would be interesting to run the additional baseline that measures how long it takes for a parent to find her child with random (non-CBIR) search. It would also be interesting evaluate this baseline while varying the number of images per screen, or, *e.g.* , allowing the parent to scroll through a screen containing all of the images.

Attribute Error

In Chapter 5, we saw that attribute labeling error happens both because of automatic classification error, and the inherent subjectivity of the labeling task. As such, it is important to analyze how the system is affected by error. Figure 6.7 shows the performance over the number of screens seen when all three automatic attributes – skin color, eye color, and age – were correct. Here, correct is measured as being the same as the set of attribute labels chosen by the parent. One can see that in these cases, using attributes is extremely beneficial. Adding browsing to attributes doesn't seem to help, but it also doesn't appear to hurt.

Figure 6.8 shows performance over the number of screens seen when any one of the attribute categories had error, 14 runs total. It also shows the distribution of 'Correct Attribs', *i.e.* what 'just attributes' would have been if the automatic attributes had had no error. Looking at the distributions of the first three methods, 'just browsing' is slightly wider than the rest. It is interesting to note the simulated performance of 'Correct Attribs', however; one can check that the performance of 'just attributes' is mostly affected by error, and not very much by, *e.g.* , some skewed population distribution effect. Overall, the mean performance of 'just attributes' is worse than 'attributes plus browsing' and 'just browsing' (though only slightly). Browsing improves over just attributes in the case of attribute error.

---

of age. Using a random permutation would likely further improve our results.
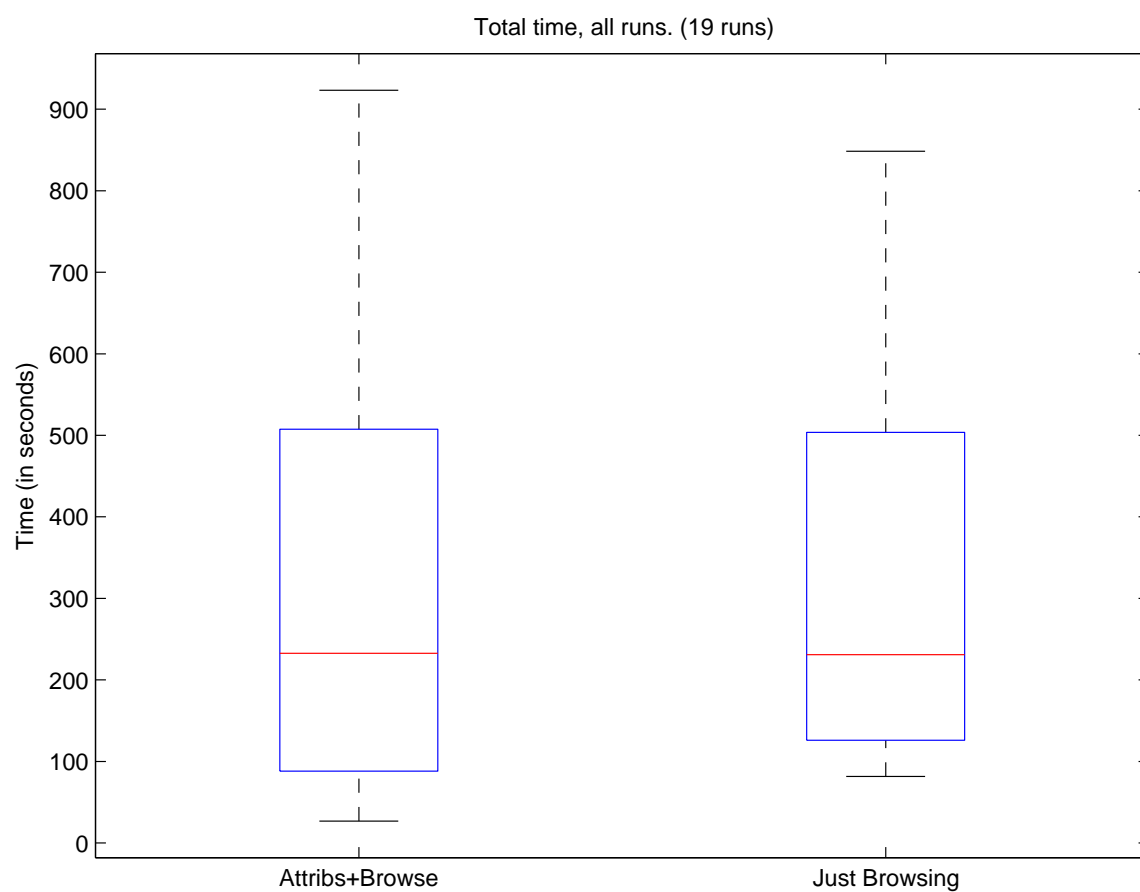
Figure 6.5: Box plot of the total time taken for all of the runs in the Main Dyad test.
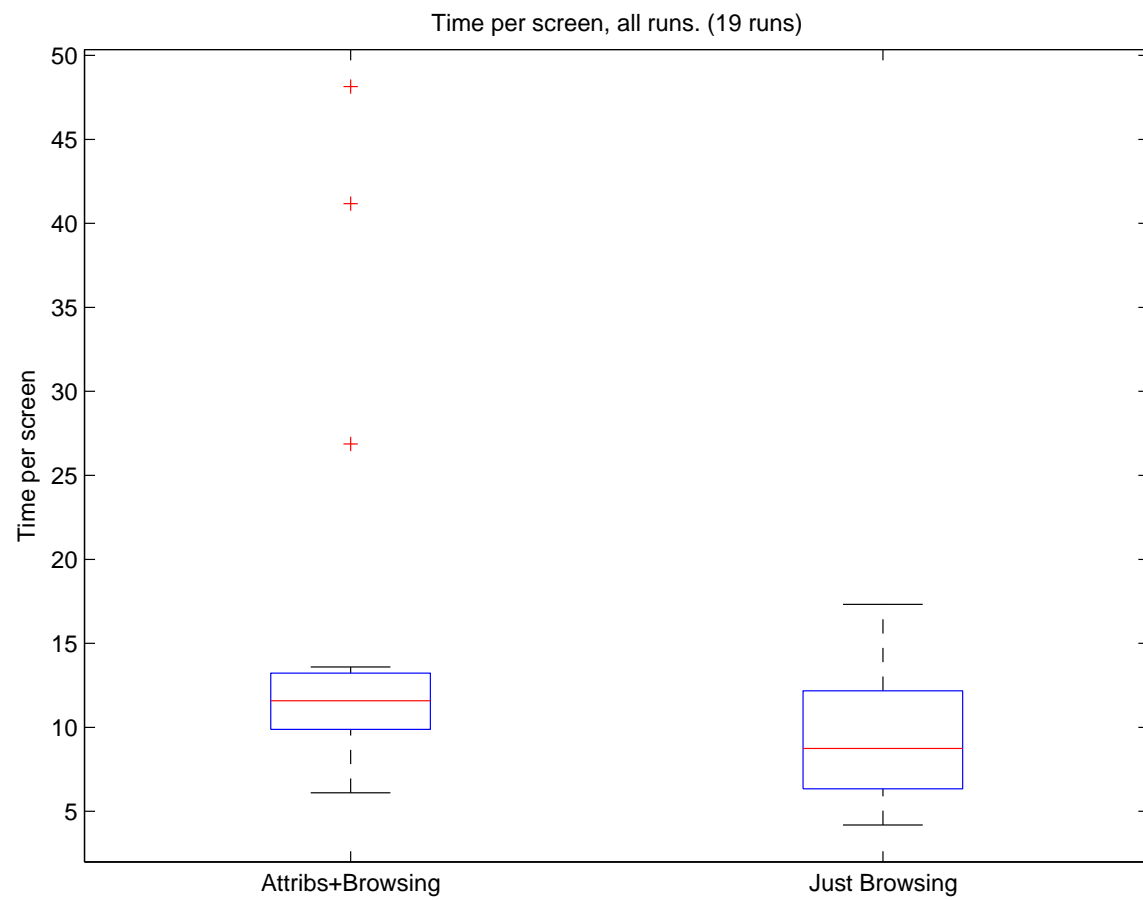
Figure 6.6: Box plot of the time per screen for all of the runs in the Main Dyad test.
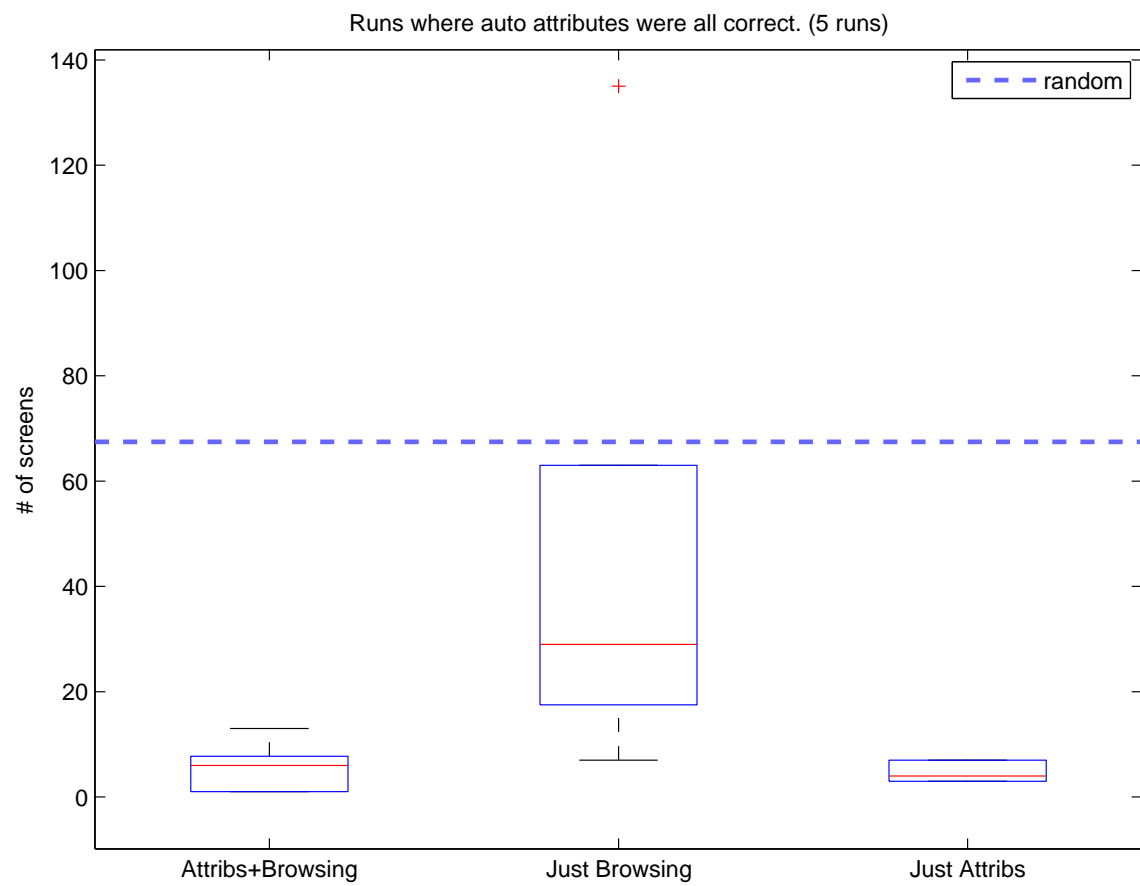
Figure 6.7: Box plot of the number of screens seen for those runs of the Main Dyad test where the automatic attributes were the same as the parent chosen attributes.
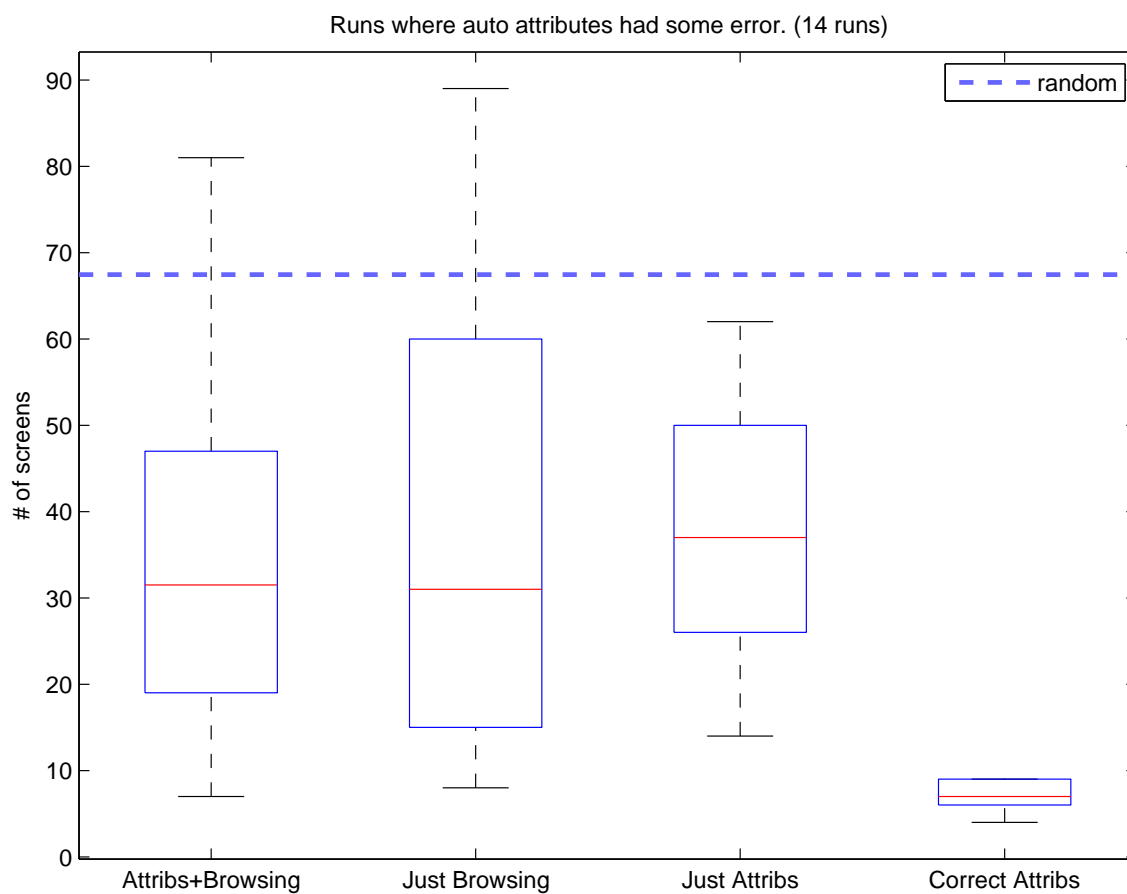
Figure 6.8: Box plot of the number of screens seen for those runs of the Main Dyad test where there was a discrepancy between the automatic attributes and the parent chosen attributes.
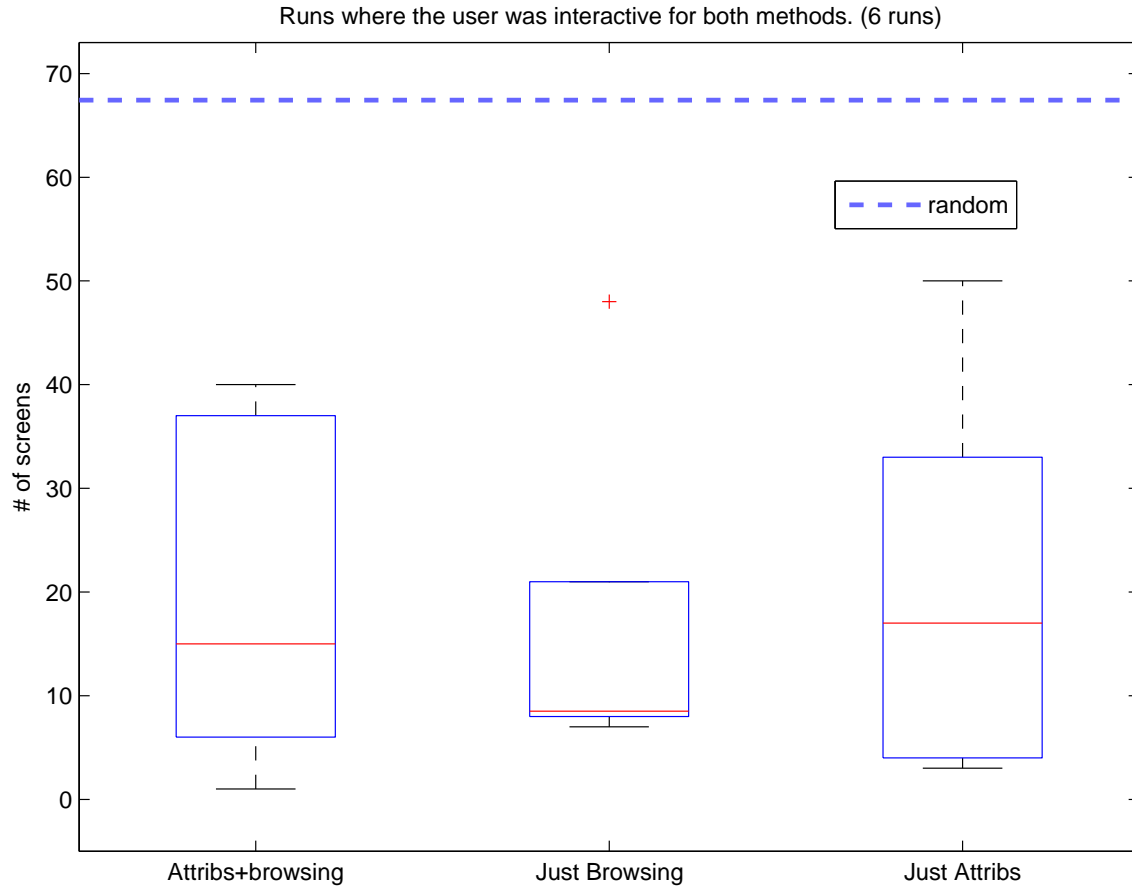
Figure 6.9: Box plot of the number of screens seen for those runs of the Main Dyad test where the parent was 'interactive' in both the 'attributes plus browsing' and 'just browsing' methods.

Interactivity

Some users had a very low level of 'interactivity', and rarely chose similar images. Here, we define an interactive user as one who chose, in total, a number of similar images $\geq$ half the number of screens she saw for that run. A non-interactive user is one who chose fewer than that threshold. If a parent is non-interactive, then she is, on average, selecting 0 similar images on more than half the screens, and the rest of the time only choosing 1.

Figure 6.9 shows the performance on runs where the parent was interactive for both the 'attributes plus browsing' and 'just browsing' methods, 6 runs total. In this case, 'just browsing' performs better than the other methods, and 'attributes plus browsing' performs on par with 'just attributes'.

Figure 6.10 shows the performance on runs where the parent was not interactive for either method, 5 runs total. Here, the distribution of 'just browsing' is more widely spread than
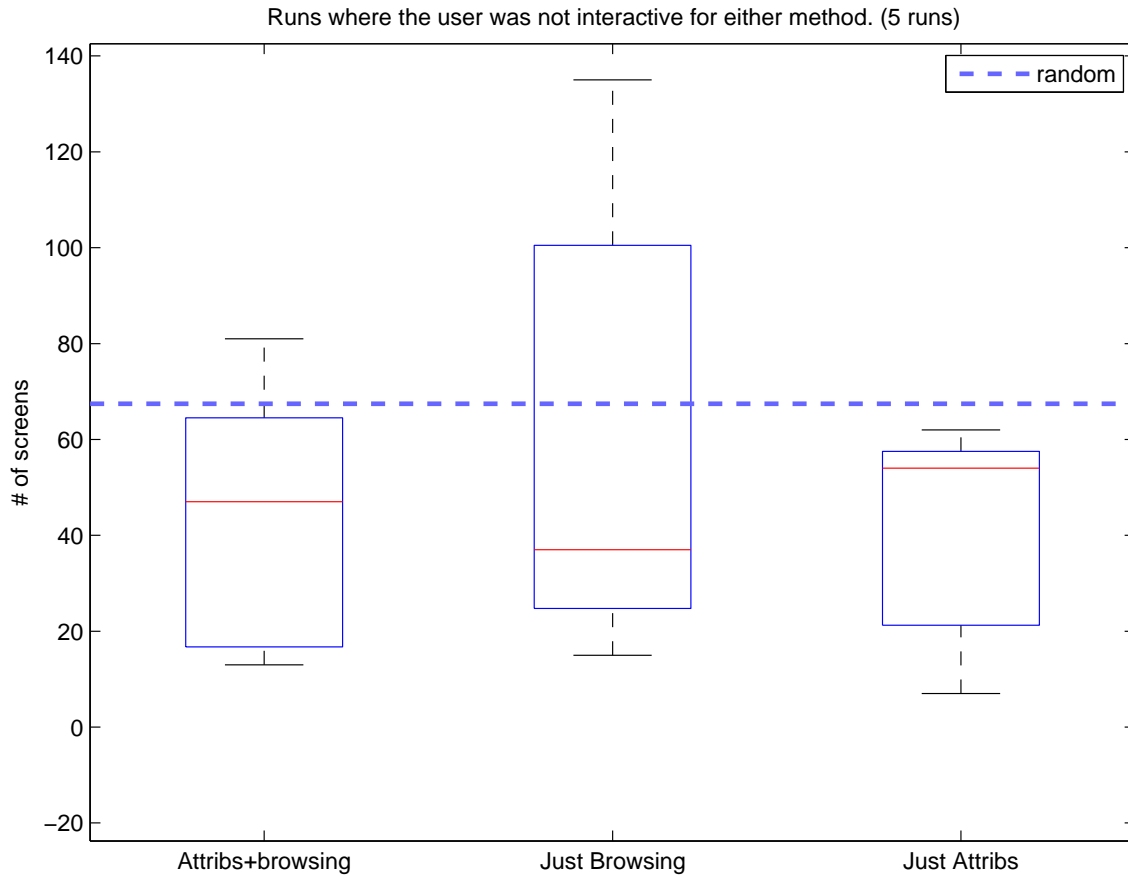
Figure 6.10: Box plot of the number of screens seen for those runs of the Main Dyad test where the parent was 'not interactive' in both the 'attributes plus browsing' and 'just browsing' methods.

the other methods, and is skewed towards longer runs. (Although the average performance is still lower than the other methods.) It would be interesting to adjust the definition of interactivity to be more and less strict, and to see the performance of the resulting sets of runs. However, at this point there are too few runs total to do this. Overall, 'attributes plus browsing' and 'just browsing' outperform than random to a much greater degree when the parent is interactive than when she is not.

One can also look at the interactivity for each method, ignoring whether the other method was performed interactively or not. Figure 6.11a shows the runs where the parent was interactive for 'attributes plus browsing', 12 runs total. Figure 6.11b shows the runs where the parent was not interactive for 'attributes plus browsing', 7 runs total. When the parent is interactive, 'attributes plus browsing' performs better than 'just attributes'. In general, the relative performance of 'attributes plus browsing' compared to 'just attributes' is much

better when the parent is interactive than when she is not. The same can be said for the relative performance of 'attributes plus browsing' compared to random presentation. Similarly, Figure 6.12a shows the runs where the parent was interactive for 'just browsing', 8 runs total, and Figure 6.12b shows the runs where the parent was not interactive for 'just browsing', 11 runs total. Again, when the parent is interactive, 'just browsing' performs much better than 'just attributes', and, in general, the relative performance of 'just browsing' compared to 'just attributes' is much better when the parent is interactive than when she is not. The same can be said for the relative performance of 'just browsing' to random presentation.

Future tests should be designed to determine why a parent may be more or less interactive. One interesting observation from the runs where the parent was either interactive or not interactive for both methods, is that the the number of screens seen with 'just attributes' was much lower when the parent was interactive than when the parent was not interactive. (See Figures 6.9 and 6.10.) 'Just attributes', however, is purely determined by the order of the initial prior, and has nothing to do with interactivity. Thus, one should wonder what may have been going on in the runs that 'caused' the parent to be non-interactive. One possible reason for 'just attributes' taking a large number of screens is that these runs also correlated to attribute error. In fact, looking at the distribution of attribute labels in the dataset (see Figure 6.2), the highest possible number of screens for 'just attributes' without attribute error would be only 9. (This number was also based on the fact that none of the parents skipped choosing any attribute labels.) By contrast, the highest number of screens for 'just attributes' in Figure 6.10 is 62. Thus, most of the runs where the parent was not interactive also correspond to runs where there was some amount of attribute error. One can conjecture why there may be a correlation between the attribute error and the parents' non-interactivity. Possibly, parents may resist choosing images as similar if the images have eye color, skin color, or age attributes different than the labels they chose for their own children. Because of the size and distribution of the dataset browsed over, parents could exhaust the subset of images with the same label quickly and could spend most time looking at images with slightly different labels.

Ideally, we would like all parents to be interactive. However, each parent has a different personal metric for choosing similar images (even if this metric is not semantically expressible). It would be interesting future work to investigate ways to get the parents to choose a greater number of similar images, or to investigate why the task may be inherently difficult for some parents or in some cases. One possible avenue of investigation would be to make sure that the parents understand that they don't have to base their selection of similar images on anything related to the semantic attributes; they can decide an image is similar however they'd like, and don't have to explain or justify their selection. It could be, however, that parents personally choose to base their choices on semantic features such as eye color, skin color, and age; some parents mentioned in passing to the volunteers that they did not like choosing an image as similar, if the child in it appeared to have an age or ethnicity different than their own child. More experiments and surveys should be designed and run

to study this. It would also be interesting to think of the problem in terms of perceptual metrics and feature weightings, as described in Chapter 3. See Section 7.3 for more of a discussion on how to potentially encourage parents to "loosen" their threshold for similarity, and to enforce more user feedback.

The lack of interactivity for some parents, however, does raise the question of why the parents never used the 'Show Random' button. Looking back at the instructions read to the parent during the Dyad Drill (see Appendix B) the part explaining the buttons is noted as 'As Needed', so it is possible that the parents were unaware of what the 'Show Random' button did. Even if they were aware, perhaps in future tests we should instruct the volunteer to suggest that the parent use the button if the parent is not selecting any similar images for many screens in a row. (Having the volunteer strongly suggest using any buttons, however, may interfere with the process or be stressful to the parent. This should be explored in future tests.) The 'Show Random' button is useful if the parent may find herself stuck in an area of the face space that yields few images similar in appearance to her child.

Note that in Appendix B, the parents were possibly not being read the full set of bullets explaining what similar images meant – only if they asked what similar meant. This could also affect the interactivity or performance of the system, and should be reviewed for future tests.

Missed Children

It should be noted that in the dyad tests (and the disaster drill), the system stopped after the parent got to a screen containing her child, even if the parent missed him and choose to continue refining. (This was an experimental design created specifically for the tests, dependent on knowing what child each parent was looking for. In an actual deployment during a disaster, this functionality would not make sense nor work.) There were 4 cases where the parent missed her child, 3 in 'just browsing', and 1 in 'attributes plus browsing'. Figure 6.13 shows only those tests where the parent found her child by both methods. The distributions appear similar, and the relative differences between methods are comparable to those between all runs. Thus, it's reasonable that we performed our other analysis on subsets of runs without removing the runs where the child was missed.

Discussion

It can be seen that 'attributes plus browsing' performs significantly better than random (with a p-value of $3.51e^{-7}$), even when the attributes are classified automatically. Using attributes and browsing together performs better than using any one method individually. Additionally, when there is classification error, the increase in performance from using 'attributes plus browsing' over 'just attributes' is even stronger. This suggests that browsing helps the system recover from attribute error. Browsing is theoretically most useful when attributes don't sufficiently narrow the number of potential images, *i.e.* when there is a homogeneous
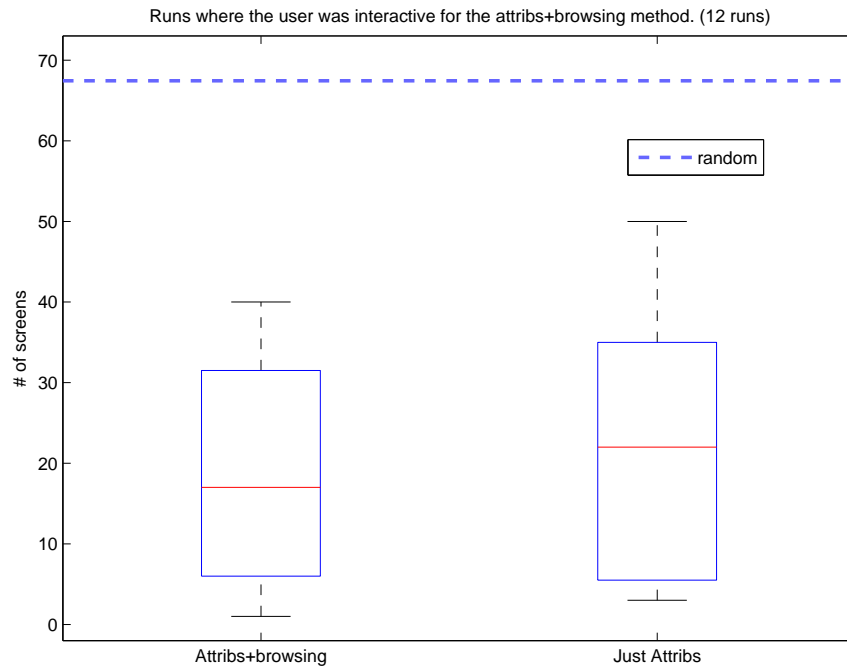
(a) 'Attrib+Browsing' Interactive
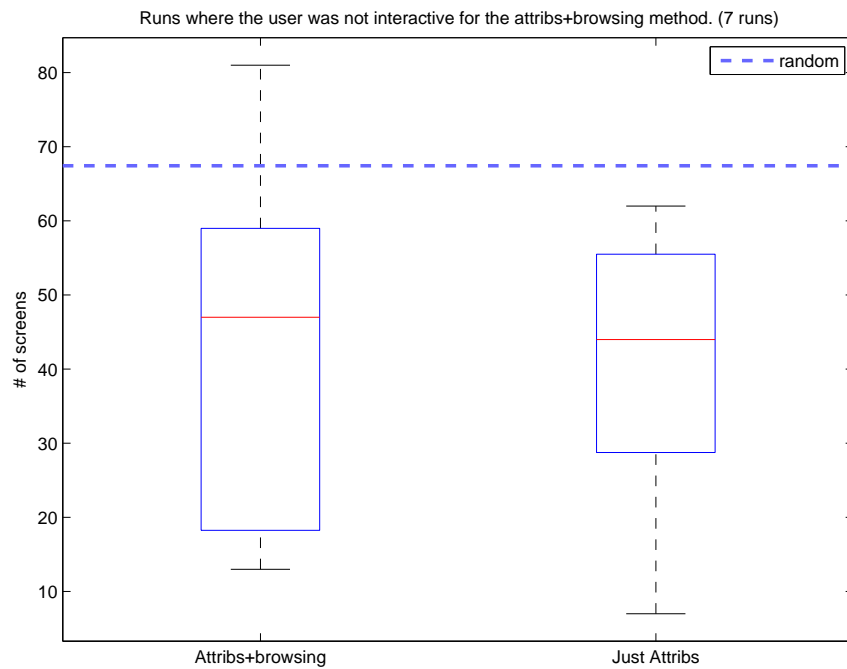
(b) 'Attrib+Browsing' Not Interactive

Figure 6.11: Runs where the parent was interactive and not interactive for 'attributes plus browsing'.

Runs where the user was interactive for the browsing only method. (8 runs)

(a) 'Just Browsing' Interactive

Runs where the user was not interactive for the browsing only method. (11 runs)
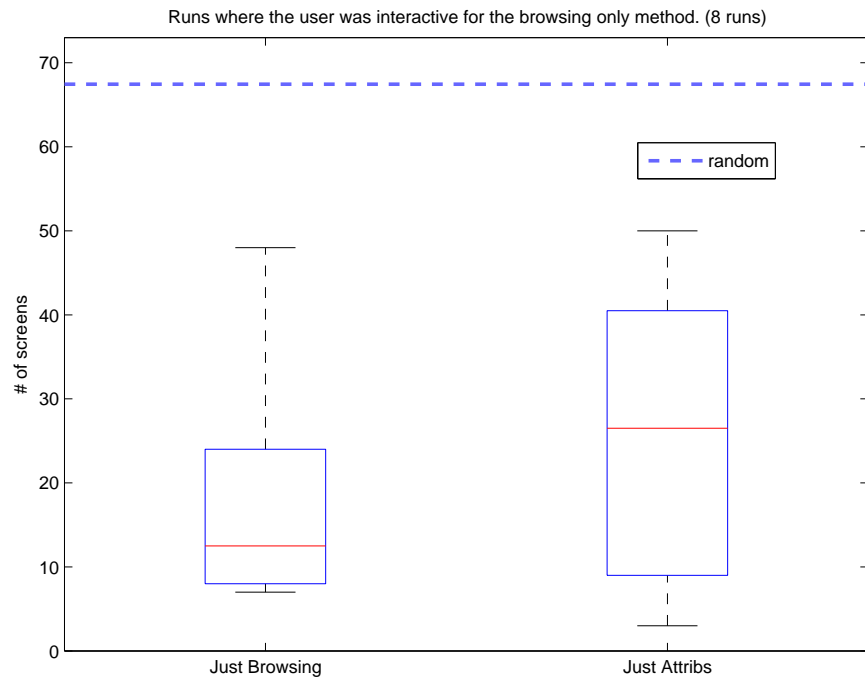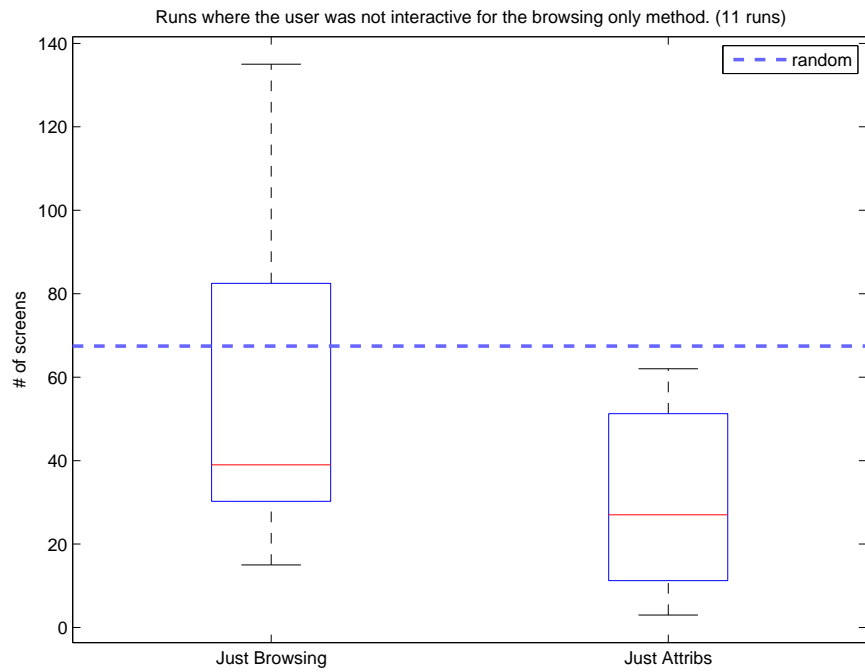

(b) 'Just Browsing' Not Interactive

Figure 6.12: Runs where the parent was interactive and not interactive for 'just browsing'.
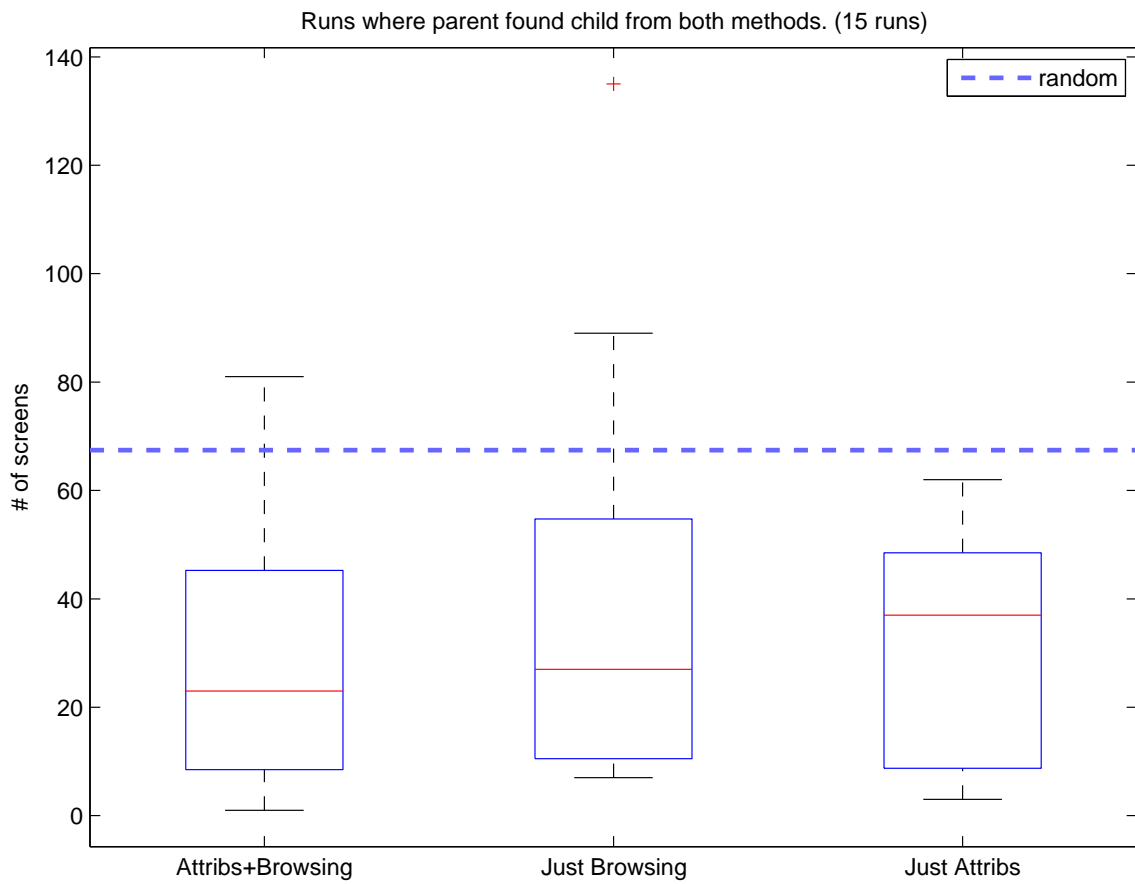
Figure 6.13: Box plot of the number of screens seen for those runs of the Main Dyad test where the parent did not miss their child in either the 'attributes plus browsing' or 'just browsing' method.

population, or a very large population in general. Our tests on a homogeneous population showed that browsing was significantly better (with a p-value of $3.4e^-2$) than 'just attributes'. In this case, 'just attributes' is equivalent to random presentation over the homogeneous dataset.

In general, the amount of user feedback increases proportionally with the number of screens a parent looks over. The more user feedback, the more we would expect browsing to be beneficial. In our tests, as the parents were more interactive, browsing indeed had more of an effect. In fact, just using browsing was better than just using attributes in the case of higher user interactivity. Looking at the runs where the parent was interactive/not interactive while using the 'attributes plus browsing' method, and comparing the relative performance of 'attributes plus browsing' to 'just attributes', one can see that here, as well, that browsing had more of an effect when the parent was interactive. Also, the improvement of 'attributes plus browsing' over 'just attributes' in the interactive case was relatively stronger than the improvement of 'attributes plus browsing' over 'just attributes' when considering all runs.

Another thing to note is that most of the browsing set was made up from the Parenting.com dataset. The attributes stored with the dataset were based on hand labels with mostly high inter-rater agreement. (Those with low inter-rater agreement were hand-fixed.) Thus, these tests did not fully test how the system would perform if attributes had been fully automatic for *all* of the images in the dataset. However, we decided not to run a test using automatic attribute classification over the images from the Parenting.com dataset because of the images' quality; the new images of the recruited children were of a much higher quality and standardization of lighting and color. The images from the Parenting.com dataset had much more variety, and their automatic attribute classification performance was much worse, so it would be somewhat unfair to claim that that was a more realistic scenario. Also, the new children added to the database during the dyad tests *were* automatically classified, and the results of their automatic classification are independent of whether or not the Parenting.com dataset attribute labels are automatically determined. Using automatic attributes on the Parenting.com dataset would, however, affect how many images had any given initial prior. Thus, it could potentially affect the performance of 'just attributes', though depending on the distribution of class labels after automatic classification, the performance could either be better or worse. Similarly, it could slightly affect the performance of 'attributes plus browsing' and 'just browsing', but because the attributes were only used during initialization, the ordering of the initial prior would have less and less of an effect as the parent chose more similar images. Having some amount of error in the automatic attribute classification of the rest of the dataset could add some amount of randomness to the images shown and could in fact help performance, depending on the parent's personal criteria for choosing similar images.

6.4 Survey Results

After the disaster drill and each dyad test, the parents were asked by Children's Hospital staff to fill out a survey. The survey questions were designed by the Children's Hospital with the goal of determining the parent's comfort with the system and system design. In this section we will go over some of the comments from the disaster drill survey, as filled out by 9 of the participating parents. Some comments led us to make slight changes in the system for the dyad tests. Noting these comments and the changes we made in response demonstrate our iterative design process. Many of the survey questions were general to the system used in both the disaster drill and dyad tests, and are generally relevant to designing a CBIR mental image search with attribute information. We present the information from the disaster drill because we wanted to evaluate the parents' reaction to the 'attributes plus browsing' complete system.

First, in the disaster drill, several parents asked for bigger pictures and less cropping. In fact, one parent noted that the close cropping was why she missed her child. We changed this for the dyad test, allowing for a larger cropping in the thumbnails, and larger photos on the screen.

Parents were asked the general question "What would be the problems that you would expect if this system were used in a real disaster?" One parent noted that "The number of pictures to view may be overwhelming. May be difficult for parents to pay attention to instructions." Another parent similarly noted "The number of pictures that the parent would have to go through." The total number of images in the database depends on the scale of the disaster, but if parents are expressing possible stress related to the total number of images they realize they might have to look through, and an inability to pay attention to the instructions, perhaps the way the instructions are written or presented should be changed. This might also warrant a change in the GUI aesthetics or the way the progress bar is displayed. Another parent noted that she wanted the system to "offer exact birthdate, ability to scan in actual photos parents have to use in disaster". In order to address this issue, there could be future focus on refining the automatic age classifier to work with finer age labels. In Section 7.1 we discuss adding functionality to allow parents to input photographs if the parents have some available. Finally, one parent noted "technology glitches", and another noted "great system, they just need to know how to deal with the parents". These problems could be addressed with more training for the volunteers, and in networking the database/image upload (see Section 7.1).

Another question was "What do you like about the REUNITE system?" The parents overall found the system easy to use, and were pleased: "It seems to work", "It really did work", "Its purpose", "Ease of use", "The goal to reunify families in a systematic way. Relatively easy to use.". One parent replied "The limiting at each step to minimize the number of kids that you have to look at." This suggests that parents liked seeing only a few images per screen instead of being presented, *e.g.* , with all of the images on one page they would have to scroll through. Another parent said "Actual pictures", which suggests

that, aside from being an important identifier of at-risk children, photos are a good way for parents to feel confident in the identification of their child. None of the children in the drill were injured, however, and the potential effects on parents of having to look through photos of injured children, and possibly seeing their own child with injuries should be studied.

Parents were also asked "What do you dislike about the REUNITE system?" Three parents noted that the photos were cropped too tightly and that image size on the screen was too small. As previously noted, we fixed these issues for the dyad test. One parent noted that the task was "Harder than I thought", though it's unclear what part of the system she was talking about. Another wrote "Time to get it going", referring to the time it took to take photos of the children and upload them. The time it took was partly an artifact of the drill setup; most of the time was spent synchronizing the laptops with the same database using a single USB key. Also, the automatic attribute classification code could be optimized. (See Chapter 7.) One parent wanted to select gender and exact age, and another said we "Need another way to match parents with children." It is unclear what she was referring to, but there was some general, verbal concern during the drill that another parent could identify and take the wrong child. This issue is orthogonal to our task, and there are separate standards/protocols that would be enacted before allowing a parent to take a child.

The final general question that parents were asked was "How do you think the system can be improved?" Three parents again asked for bigger pictures and less cropping. One parent wanted more ethnic variety in the database. This would be addressed by having a larger database in general (see Section 7.7), though in a real disaster, the ethnic makeup of the database would depend on the ethnic makeup of the children affected by the disaster. Two of the comments were about the attributes: One parent asked for a wider range of skin tones (see Chapter 7). Another noted the difficulty in labeling eye color, saying "Eye color was easy for [my] child with brown eyes. It is difficult for the child with blue/green/hazel eyes". We may want to think about allowing the parents to enter their level of certainty when they choose attribute labels. Another parent said "maybe filtering by gender? Frustrating looking at boys while looking for a girl, and vice versa". We could add gender as an attribute for future work, though it is unclear how easy the task is even for humans, especially with regards to very young children. Two parents asked, in general, for more explanation of the system and more reminders to ask for clarification.

Chapter 7

Future Work

The performance of our system was significantly better than a baseline performance of random browsing. Looking at the test results, parents' comments, and reflecting back on the entire process, there are a few areas of future work that would be interesting to pursue. In this chapter we will go over the main areas of future work that we hope to focus on in future iterations of the system. The ideas span additional functionality, as well as algorithmic, GUI, feature, and testing changes that could possibly improve performance.

7.1 Adding Photos

During the disaster drill and dyad studies, Children's Hospital took a few photos of each child, hand-chose which photo to use, noted the selected photo on the REUNITE form, then uploaded it later. They are hoping to streamline the process in future versions of the system. First, they would like to simply upload all photos of a child and let the system choose which image is the best. (The reason why multiple images were taken in the first place was that sometimes the angle or expression of the child wasn't ideal – *e.g.* the child began turning away or blinked.) One possible solution is to add all photos of the child. This would take more time at upload, the automatic attributes could be different, and there would be a larger number of images the parents would have to browse. However, having possibly different attributes could raise the chances that one of the photos matches the attributes that the parent enters, and even though there is a larger number of images to browse, there would still be the same number of unique children. Further tests would be required to see whether this process was helpful. In such tests, it would also be important to take into account the possible stress of seeing additional photos, especially if many of the photos showed injuries.

Additionally, future systems could wirelessly send the images to the computer as they were taken and automatically upload images through a script at set times during the day. One pitfall to this is the readiness of network access during a disaster. Of course if there weren't network access, one couldn't access the central database of photos either. Future col-

laboration with networking researchers may help solve these problems, and enable a quicker set up/synchronization of the multiple stations used for the parents' searches. Some form of centralized database will also be necessary for wide-scale deployment.

7.2   General System

In the tests described in this thesis, parents sometimes missed seeing their child on a screen. One question to ask is why were children being missed? With such a small number of missed children, it's hard to do an analysis of the current data. After more runs, it would be interesting to see, *e.g.* , whether parents completely missed their child, or if they selected their child as similar and refined (instead of clicking 'Found'). We could also use this information in making a subsequent version of the system. In general, if parents sometimes miss their children, it might be useful to show browsed children again. Currently, once a child's photo is seen, that photo would not be seen again.

Also, once a child is found, there is the question of whether he should be removed from the database. The protocol with which to do this (*e.g.* should this happen after the social worker confirms that the parent and child reunification is actually correct) needs to be studied.

The prototype system itself was written in MATLAB, and was not optimized for speed; each iteration of refinement in a parent's search was fairly instantaneous, but there was a bit of lag starting up the system and automatically classifying attributes. These issues did not affect any of the search results, but should be addressed for the actual deployment of the system.

7.3   Browsing Algorithm

If would be interesting to try a different method of choosing images for each screen. Currently, we choose images based on the most probable posterior value. Cox *et al.* [29] also tried choosing based on highest entropy, which they found performed better than most probable. Additionally, it could be useful to occasionally show a uniformly random image as one of the images on the screen. That way, it would have a similar effect to 'Show Random', which would be useful if the parent never uses the button (but should). It could also prevent perceptual adaptation.

There is also the question of what to do if the parent does have a photo of the child. It might be useful to allow the parent to opt for a query-based search, instead of a mental search. One way to incorporate the current method of relevance feedback could be to let the query image affect the initial screen shown, and to treat it like a perpetual similar image. We could also use more features from face recognition and matching literature. One potential problem, aside from general face recognition problems relating to image quality, lighting, pose, expression, etc, is that the image may have been taken when the child was much

younger.

As mentioned in the **Interactivity** subsection of Section 6.3.2, some parents were not very interactive. Future work could include further investigation of why a parent may or may not be interactive. It could also include learning more about the perceptual metrics parents use when judging similarity for this task (be it metrics general to most parents, or trying to learn individual metrics per person). Orthogonal to those questions, however, is how we may encourage a parent to choose more similar images. Because we are looking for user feedback that correctly reflects the parents' judgments, there is a fine line between forcing parents to choose similar images they completely disagree with, and perhaps training them to have more lenient standards for similarity. One possible way could be to sprinkle random images throughout the screens. This goes along with possibly enforcing the use of the "Show Random" button when appropriate (also discussed in Subsection **Interactivity** of Section 6.3.2). Some of the extra GUI functionality discussed in Section 7.4 may also help with this question.

It would also be worth exploring an explorative versus an exploitative approach of choosing the images to display per screen. (See Section 4.1 for an explanation of exploration and exploitation.) Different algorithms for choosing the display could also help with interactivity.

7.4   GUI

Throughout the process of iteratively making the system with the Children's Hospital, there were several ideas for future additions to the GUI that could potentially decrease the number of screens seen:

- Choosing favorites: In addition to the n images shown to the user based on the posterior, there could be an additional set of 'favorites'. The set of favorites would start out empty, but when choosing similar images, the user would have the additional option to add an image to the list of favorites. Then, at each iteration, any image currently chosen as a favorite would be treated as a similar image. Even if the parent gets to a screen where she doesn't choose any similar images, as long as she has a list of favorites, the user model will still change.

- Choosing bookmarks: This is very similar to favorites, but with the added functionality that the parent would be able to revisit the bookmarked images later. It could be rolled into the favorites GUI, or made separate.

- Choose parts of the face: Sometimes a parent may feel uncomfortable saying an entire image is similar, but she may want to convey that a specific part of the image is similar. She could be allowed to choose regions of the face, or select from a list of non-regional features, *e.g.* attributes (such as skin color, eye color, and age) or other semantically describable features (such as 'general face shape'). This functionality is also mentioned in the future work of Cox *et al.* [29].

- Amount of similarity: The parent could click on images multiple times to convey the degree of similarity. This is also mentioned in the future work of Cox *et al.* .

Another area for enhancement is GUI aesthetics. Our placement of buttons, images, etc, could be studied in future iterations in order to improve the parent's experience. The layout of the GUI could even affect how the parent approaches the search, and could thus affect search time or number of screens.

7.5 Better Attributes/ Attribute Labels

In our tests, we used skin color, eye color, and age as attributes. Some possible additional attributes are gender, hair color, and 'syndromic'. It would also be interesting to look into using a different set of labels for skin color. Because the skin color labels we used varied both in color and saturation, raters were sometimes divided between two far-apart labels. We could consider instead using a range of labels that does not refer to specific colors at all, for example the labels used in classifying sun damage risk. These labels include both a photo of an exemplar color, face, hand, etc, and a quick blurb that describes the types of skin the photo represents. The blurbs describe more about 'skin type' than color, and may yield a better inter-rater agreement so that a finer level of attribute labels might be used in automatic classification. Adding attributes and attribute labels, however, could lead to trade-offs in terms of error (see Section 4.3.4), and tests would have to be conducted. As mentioned in Section 4.3.3, the attribute labels could also be integrated continuously into the prior.

It might also be useful to allow the parents to rate their level of certainty when they select attribute labels.

7.6 Other Features to Try

Even though distance and ratio features were inconclusive in the triples experiments (described in Chapter 3), it would still be interesting to try using them in the CBIR system. Other features to try could be: PCA on the landmarks, PCA on the image warped by the landmarks, color, texture, or any combination. Additionally, we could integrate the attribute information into the feature vector, not just the initial prior.

7.7 User Studies

With regards to parents missing images of their children, it might be important to revise the instructions. If parents are choosing their child as similar but not saying that the child is "Found", they should know that if they choose "Found", they will see the full photo of the child and be asked whether they are sure before the system quits.

The volunteers should also perhaps be told to direct the parents more. In the **Interactivity** subsection of Section 6.3.2, it was noted that the parents may have been unaware of the functionality of all of the buttons, or what they were allowed to do. It will be important to refine the instructions read to the parent for subsequent tests, and to possibly reword parts if the parent doesn't understand.

In general, future tests should be performed on a much larger dataset. It is important to test the system for use in a much larger scale disaster, or for much larger subsets of homogeneous populations. This will also help demonstrate the full potential of CBIR based browsing.

Chapter 8

Contributions and Conclusions

In this thesis, we have presented a novel Content-Based Image Retrieval system to assist in reuniting missing children with their parents after a disaster. In particular, we have novelly combined an attribute search with user feedback in a Bayesian CBIR mental search framework. We also implemented functionality to automatically classify skin color, eye color, and age attributes, and tested the entire system, including using the automatic attribute classification, in a series of realistic experiments. In the experiments, we showed that our system significantly outperforms a random, non-CBIR search.

Overall, we ran three sets of experiments with the Children's Hospital Boston: 1) a disaster drill to test the overall 'attributes plus browsing' mental image search system on actual parents, 2) a series of tests on 'synthetic' parents to evaluate the effectiveness of browsing, and 3) a set of dyad tests, run on actual parents, designed to evaluate the 'attributes plus browsing' system with automatic attributes, and to demonstrate the merit of browsing in a realistic setup.

The disaster drill was set up as a dry-run of the entire reunification process. Social workers observed the volunteers and parents, evaluated the parents' mental state, and helped the volunteers work with parents who had been instructed to act difficult. The parents who were enrolled in the drill ran a version of the full system, entering skin color, eye color and age attributes of their "missing" child, which were then used to prep the Bayesian CBIR mental search. On average they saw 5.7 times fewer screens than they would have with a random presentation of the photos, demonstrating that the overall system was extremely effective. Because the purpose of this drill was to demonstrate overall effectiveness, the children added to the the database during the drill had their attributes manually labeled.

The distribution and size of the population in the disaster drill database was such that using attributes dominated the performance of the system to a degree that it wasn't possible to draw any conclusions about the benefits of using browsing in addition to attributes, as opposed to just using attributes alone. (The huge increase in performance over random, however, did suggest that browsing was not, at least, having a hugely negative effect.) In order to test the effectiveness of browsing in general, we ran a series of tests on 'synthetic'

parents, *i.e.* people who weren't actually parents. In these tests, we showed that browsing alone yielded, on average, 2.9 times fewer screens than random. This clearly showed that browsing is beneficial. When 'synthetic' parents used attributes with browsing, the performance was, on average, 4.1 times better than random. Because the parents were 'synthetic', they had to refer to an image of the child they were trying to find. While this means we needed to run another test to demonstrate the effectiveness of browsing in a mental search framework, these tests clearly showed the benefits of browsing in general.

The third set of experiments, the dyad tests, were designed to evaluate automatic attributes with a Bayesian CBIR mental image search, using real parents. The tests were also designed to demonstrate the merits of browsing in scenarios where browsing would be most beneficial. In the main series of dyad tests, the enrolled children had their attributes automatically classified, and the parents ran the system with two methods: 'attributes plus browsing' and 'just browsing'. When parents ran the full 'attributes plus browsing' system, they looked at, on average, 2.5 times fewer screens than random presentation. The distribution of the number of screens also had a p-value of $3.51e^{-7}$, when compared in a T-test to chance, showing strong statistical significance. Using automatic attributes in addition to browsing had better performance than 'just browsing', even though 'just browsing' was still 1.6 times better than chance with a p-value of $3.5e^{-3}$. 'Attributes plus browsing' also performed slightly better overall than just using attributes. Again, however, the population sub-group sizes were too small to expect a strong effect. Looking only at the runs where there was attribute classification error, however, and the the runs where the parent was the most interactive, 'attributes plus browsing' had a stronger effect over 'just attributes'. Even 'just browsing' was better than 'just attributes', looking at the runs where the parents were more interactive during the 'just browsing' method.

In order to simulate a very large dataset, or a homogeneous population (two cases where browsing would be extremely important), we also ran a dyad test over a homogeneous population. Here, browsing yielded 1.6 times fewer screens than random presentation which, in this case, was equivalent to just using attributes. This scenario is equivalent to either a population where all of the attributes were the same, or an extremely large population that was still large enough to benefit from browsing, even after the attributes had been used.

In general, there is a dire need in the community for a post-disaster pediatric reunification system that helps the most at-risk subgroup of children – those who are unable to identify themselves. There have been some recent advances in disaster reunification in general, but these methods for the most part require the children to identify themselves verbally or in writing, and thus can not be utilized by at-risk children. When children are unable to personally provide identification, and there is no other identification on their person, photographs are a good way for others to positively identify them. There is one existing system for disaster reunification currently in creation that uses photographs, but it does not focus on reducing the number of images seen using Content-Based Image Retrieval methods, and the only visually semantic attributes it uses are gender and age. These attributes are also not automatic, and in a surge capacity situation there might not be personnel resources

to hand-classify them. Without these attributes, parents would have to randomly browse through potentially thousands of images.

We designed, engineered, and tested our system specifically with the goal of at-risk pediatric reunification in mind. We also addressed the issues of surge capacity, and attempted to reduce the parents' stress and mental anguish by showing as few images as possible. We have shown through our realistic tests that our system does significantly reduce the number of images the parent would have to see. There are also several promising avenues for improvement of the system. Ideally, such a system will be integrated into the disaster response framework, and parents will no longer have the added trauma of long separations from their children after a disaster.

Bibliography

[1] Amazon.com, Inc, or its Affiliates. Mechanical turk. www.mturk.com.

[2] American Red Cross. Safe and well. https://safeandwell.communityos.org/cms/.

[3] J. Assfalg, A. Del Bimbo, and P. Pala. Three-dimensional interfaces for querying by example in content-based image retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 8(4):305–318, 2002.

[4] P. Auer and A.P. Leung. Relevance feedback models for content-based image retrieval. *Multimedia Analysis, Processing and Communications*, 2009.

[5] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[6] M.S. Bartlett, M.H. Lades, and T.J. Sejnowski. Independent component representations for face recognition. In *Proceedings of SPIE*, volume 3299, page 528, 1998.

[7] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[8] A. Bell. The Definition of Beauty. *Nature, October/November Issue*, 1997.

[9] Philip J Benson and David I Perrett. Synthesising continuous-tone caricatures. *Image and Vision Computing*, 9(2):123 – 129, 1991.

[10] F.L. Bookstein. *Morphometric tools for landmark data: geometry and biology*. Cambridge Univ Pr, 1997.

[11] Susan E. Brennan. Caricature generator: the dynamic exaggeration of faces by computer. *Leonardo*, 18(3):170–178, 1985.

[12] D.D. Broughton, E.E. Allen, and Hannemann R.E. Reuniting fractured families after a disaster: The role of the national center for missing and exploited children. *Pediatrics*, 117:5442–5455, 2006.

[13] V Bruce, A Cowey, A W Ellis, and D I Perrett. Eds) processing the facial image, 1992.

[14] V Bruce, P J B Hancock, and A M Burton. Human face perception and identification. in face recognition: From theory to applications. In H. Wechsler *et al.* , editor, *NATO ASI Series*, pages 51–72. Springer-Verlag, 1998.

[15] V. Bruce, P.J.B. Hancock, and A.M. Burton. Comparisons between human and computer recognition of faces. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 408–413. IEEE, 2002.

[16] V. Bruce, A.W. Young, and Scottish National Portrait Gallery (Edinburgh). *In the eye of the beholder: The science of face perception*, volume 569. Oxford University Press Oxford:, 1998.

[17] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(10):1042–1052, 2002.

[18] Yair Al Censor and Stavros A. Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press, 1997.

[19] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-Based Image Matching Using Angular Partitioning. *IEEE Transactions on Systems, Man, and CyberneticsPart A: Systems and Humans*, 35(1), 2005.

[20] Y. Chen, J.Z. Wang, and R. Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *Image Processing, IEEE Transactions on*, 14(8):1187–1201, 2005.

[21] S. Chung and M. Shannon. Reuniting children with their families during disasters: A proposed plan for greater success. *American Journal of Disaster Medicine*, 2, 2007.

[22] Sarita Chung. Pediatric disaster readiness: How far have we come? *Pediatric Disaster Readiness*, 10(3), 2009.

[23] G. Ciocca and R. Schettini. A relevance feedback mechanism for content-based image retrieval. *Information Processing & Management*, 35(5):605–632, 1999.

[24] Committee on Environmental Health and Committee on Infectious Diseases. Chemical-biological terrorism and its impact on children. *Pediatrics*, 118, 2006.

[25] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, et al. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[26] Timothy Cootes. Xm2vts 68pt markup.

[27] Bonnier Corporation. www.parenting.com.

[28] C. Cortes. and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3), 1995.

[29] I. Cox, M. Miller, T. Minka, T. Papathomas, and P. Yianilos. The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1), Jan 2000.

[30] I. Cox, M. Miller, T. Minka, and P. Yianilos. An optimized interaction strategy for bayesian relevance feedback. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98)*, pages 553–558, 1998.

[31] IJ Cox, J. Ghosn, and P.N. Yianilos. Feature-based face recognition using mixture-distance. In *1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96*, pages 209–216, 1996.

[32] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), Apr 2008.

[33] Nicolas Davidenko. Silhouetted face profiles: A new methodology for face perception research. *Journal of Vision*, 7(4), 2007.

[34] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM.

[35] Douglas DeCarlo, Dimitris N. Metaxas, and Matthew Stone. An anthropometric face model using variational techniques. In *SIGGRAPH*, pages 67–74, 1998.

[36] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy - automatic naming of characters in tv video. In *In BMVC*, 2006.

[37] Y. Fang, D. Geman, and N. Boujemaa. An interactive system for mental face retrieval. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 193–200. ACM, 2005.

[38] Yunchun Fang and Donald Geman. Experiments in mental face retrieval. In *In Proc. of Audio- and Video-based Biometric Person Authentication*, pages 637–646, 2005.

[39] L.G. Farkas. *Anthropometry of the head and face in medicine*. Elsevier New York, 1981.

[40] L.G. Farkas, W. Bryson, and J. Klotz. Is photogrammetry of the face reliable? *Plastic and Reconstructive surgery*, 66(3):346, 1980.

[41] L.G. Farkas and I.R. Munro. *Anthropometric facial proportions in medicine.* Charles C. Thomas Publisher, 1987.

[42] Federal Emergency Management Agency. National emergency family registry and locator system. `http://www.fema.gov/media/fact_sheets/nefrls.shtm`.

[43] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, 28(9):23 –32, September 1995.

[44] Greet Frederix, Geert Caenen, and Eric J. Pauwels. Pariss: Panoramic, adaptive and reconfigurable interface for similarity search. In *ICIP*, 2000.

[45] Y. Freund and R. Shapire. Experiments with a new boosting algorithm. In *In ICML*, 1996.

[46] Andrea Frome, Fei Sha, Yoram Singer, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *In ICCV*, 2007.

[47] Dorota Glowacka, Alan Medlar, and John Shawe-Taylor. Sifting through images with multinomial relevance feedback. In *In the Machine Learning for Next Generation Computer Vision Challenges Workshop at NIPS*, 2010.

[48] V. Goffaux, B. Hault, C. Michel, Q.C. Vuong, and B. Rossion. The respective role of low and high spatial frequencies in supporting configural and featural processing of faces. *Perception-London*, 34(1):77–86, 2005.

[49] BA Golomb, DT Lawrence, and TJ Sejnowski. SexNet: A neural network identifies sex from human faces. In *Proceedings of the 1990 conference on Advances in neural information processing systems 3*, page 577. Morgan Kaufmann Publishers Inc., 1990.

[50] Waddy Gonzalez. National mass evacuation system (nmets), presentation to the evacuation, transportation, and housing subcommittee, November 9 2009.

[51] Google. Finder: Chile earthquake. http://chilepersonfinder.appspot.com/.

[52] Google. Finder: Haiti earthquake. http://haiticrisis.appspot.com/.

[53] Google. Google image search. http://images.google.com.

[54] F. Grabler, M. Agrawala, W. Li, M. Dontcheva, and T. Igarashi. Generating photo manipulation tutorials by demonstration. *ACM Transactions on Graphics (TOG)*, 28(3):1–9, 2009.

[55] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision*, pages I: 413–426, 2008.

[56] H. Gunes and M. Piccardi. Assessing facial beauty through proportion analysis by image processing and supervised learning. *International Journal of Human-Computer Studies*, 64(12):1184–1199, 2006.

[57] H. Gunes, M. Piccardi, and T. Jan. Comparative beauty classification for pre-surgery planning. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2168–2174. IEEE, 2004.

[58] Peter J. B. Hancock, A. Mike Burton, and Vicki Bruce. Face processing: Human perception and principal components analysis. *Memory and Cognition*, 24:26–40, 1996.

[59] A. Holub, Y. Liu, and P. Perona. On constructing facial similarity maps. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[60] C.T. Hsu and C.Y. Li. Relevance feedback using generalized Bayesian framework with region-based optimization learning. *Image Processing, IEEE Transactions on*, 14(10):1617–1631, 2005.

[61] C. Johnston and I. Redlender. Summary of issues demanding solutions before the next one. *Pediatrics*, 117, 2006.

[62] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, 2007.

[63] T. Kanade. *Picture processing system by computer complex and recognition of human faces.* Dept. of Science, Kyoto University, 1973.

[64] Brian Kulis, Prateek Jain, and Kristen Grauman. Fast similarity search for learned metrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2143–2157, 2009.

[65] N. Kumar, P. N. Belhumeur, and S. K. Nayar. Facetracer: A search engine for large collections of images with faces. In *European Conference on Computer Vision*, pages 340–353, Oct 2008.

[66] ed. Leslie Farkas. *Anthropometry of the Head and Face.* Raven Press, 1994.

[67] Tommer Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. Data-driven enhancement of facial attractiveness. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2008)*, 27(3), August 2008.

[68] S.Z. Li and A.K. Jain. *Handbook of face recognition.* Springer, 2005.

[69] H. Liu, X. Xie, X. Tang, Z.W. Li, and W.Y. Ma. Effective browsing of web image search results. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 84–90. ACM, 2004.

[70] Y. Liu, D. Zhang, G. Lu, and W.Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[71] Y. Lu, H. Zhang, L. Wenyin, and C. Hu. Joint semantics and feature based image retrieval using relevance feedback. *Multimedia, IEEE Transactions on*, 5(3):339–347, 2003.

[72] Laurence T. Maloney and Maria F. Dal Martello. Kin recognition and the perceived facial similarity of children. *Journal of Vision*, 6(10), 2006.

[73] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, volume 964, pages 965–966. Springer Verlag, New York, 1999.

[74] B. Moghaddam and M.H. Yang. Learning gender with support faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):707–711, 2002.

[75] National Center for Missing and Exploited Children. Natural disasters: Is your family prepared? `http://www.missingkids.com/missingkids/servlet/PageServlet?LanguageCountry=en_US&PageId=3252`.

[76] *National Commission on Children and Disasters. 2010 Report to the President and Congress.* AHRQ Publication No. 10-M037, October 2010. Agency for Healthcare Research and Quality, Rockville, MD. http://www.ahrq.gov/prep/nccdreport.

[77] National Library of Medicine, U.S. Lost Person Finder (LPF). `http://archive.nlm.nih.gov/proj/lpf.php`.

[78] Pablo Navarrete and Javier Ruiz del Solar. Faceret: An interactive face retrieval system based on self-organizing maps. In *In Proc. of the CVIR*, 2002.

[79] Thomas V. Papathomas, Tiffany E. Conway, Ingemar J. Cox, Joumana Ghosn, Matt L. Miller, Thomas P. Minka, , and Peter N. Yianilos. Psychophysical studies of the performance of an image database retrieval system. In *Proc. SPIE*, 1998.

[80] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 84–91, 1994.

[81] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[82] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

[83] Gillian Rhodes, Susan Brennan, and Susan Carey. Identification and ratings of caricatures: Implications for mental representations of faces. *Cognitive Psychology*, 19(4):473 – 497, 1987.

[84] R.M. Ricketts. Divine proportion in facial esthetics. *Clinics in Plastic Surgery*, 9(4):401, 1982.

[85] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), 1998.

[86] Simone Santini and Ramesh Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:871–883, 1999.

[87] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *In NIPS*. MIT Press, 2003.

[88] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, 2006.

[89] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1, 2000.

[90] State of Louisiana, Department of Social Services. DSS offers first look at hurricane evacuee-tracking system. `http://dss.louisiana.gov/assets/docs/searchable/pressReleases/2008/05/evacueetracking5-30-.pdf`.

[91] Mark Steyvers and Tom Busey. Predicting similarity ratings to faces using physical descriptions. In M. Wenger and J. Townsend, editors, *Computational, geometric, and process perspectives on facial cognition: Contexts and challenges.* Lawrence Erlbaum Associates, 2000.

[92] Zhong Su, Hongjiang Zhang, S. Li, and Shaoping Ma. Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. *Image Processing, IEEE Transactions on*, 12(8):924 – 937, August 2003.

[93] Texas Department of State Health Services. Medical special needs toolkit. `http://www.dshs.state.tx.us/comprep/msn/Tab_J_SNETS_Sheltering.pdf`.

[94] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3:71–86, January 1991.

[95] T. Valentine. A unified account of the effects of distinctiveness, inversion, and race in face recognition. *The Quarterly Journal of Experimental Psychology Section A*, 43(2):161–204, 1991.

[96] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *GRAPHICON03*, pages 85–92, 2003.

[97] J.Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on pattern analysis and machine intelligence*, pages 947–963, 2001.

[98] R. White, A. Eden, and M. Maire. Automatic prediction of human attractiveness, December 2003.

[99] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christopher von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:775–779, July 1997.

[100] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:34–58, January 2002.

[101] S.A. Yellin. Aesthetics for the next millennium. *Facial plastic surgery*, 13:231–240, 1997.

[102] A.W. Young and H.D. Ellis. *Handbook of Research on Face Processing*. Elsevier, 1989.

[103] W. Zhao, R. Chellappa, PJ Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[104] X.S. Zhou and T.S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6):536–544, 2003.

Appendix A

Instructions for Setting the Camera

How to set up the camera:

1) Put the mode switch to the top setting (the camera icon) and turn the camera on. (The mode switch is the black bar on the back of the camera.)

2) Hit FUNC. SET and look at the top icon on the left vertical bar. You can change which option it is by hitting left/right*. The choices you have are on the bar at the bottom, and hitting left/right will switch between them. Make sure that the camera is in manual mode. This is the camera icon with the 'M' on it.

* Use the outer circle around FUNC. SET as your up, down, left, right navigation.

3) Push the MENU button. At the top of the screen there are 3 tabs, each with a bunch of settings below.

Hit down to get to each of the settings. The setting you're on will be highlighted. To change the setting, hit left/right. Sometimes there are multiple options, so you may have to hit left/right multiple times before the option you want shows up. (If the setting has ... in the name, hit FUNC. SET to get to another screen with more options.)

To get to the settings for another tab, go back up to the top so that the tabs are highlighted and hit left/right to navigate to another tab.

3a) Set the following settings in the left tab (the camera icon):

If a setting is not mentioned, use the default. Some of the settings I mention may be the default anyway. That's fine. I'm just listing the ones that we should be synced on.

AF Frame: Face Detect
AF-Point Zoom: Off
Digital Zoom: Off
Flash Settings: Red-Eye Corr.: Off, Red-Eye Lamp: On

AF-assist Beam: On
Disp. Overlay: Off
IS Mode: Continuous

3b) Set the following settings in the middle tab (the tools icon):

File Numbering: Continuous

Don't worry about Format. This can be used before you take any photos to completely clear the memory card, but should not be used after you start taking photos as it will erase the card.

4) Hit MENU again to get back to the normal LCD screen, then hit FUNC. SET.

Now you're going to set the other options just like you set auto camera mode. Hit down to get to the second to last setting on the vertical bar on the left. Use the left/right buttons to set this to S. (This means you're going to use the lowest amount of compression when taking the images.)

Hit down to get to the last setting on the vertical bar. Use the left/right buttons to get to the L option. (This means you're going to use the highest resolution when taking photos.)

5) Hit FUNC. SET again to get back to the normal LCD screen. Now hit the button you use for 'right'. This controls the flash.

Make sure that the flash is always on!

Appendix B

REUNITE Technician's Script/Talking Points

Technician Provides Instructions

Script- Ground Rules *(Technician MUST go through this section.)*
"Welcome and thank you for participating in this exercise. What is your name? My name is [BLANK] and I will be walking you through the REUNITE system today to search for your child.
<u>First</u> I am going to go over a few ground rules about the system before we begin.

- There are over 700 pictures of children in this database.

- Your child's picture has already been uploaded to the database.

- We have attempted to capture the best picture of your child, forward facing, but be aware in the picture, your child may not be smiling or he/she may have their eyes closed.

- Please be aware that if your child wears glasses, he/she will have their glasses off in the picture.

- Please do not look at your child when using the REUNITE system.

Any questions before we begin?"

Technician Begins REUNITE Process

System Talking Points

Technician MUST Say: First I am going to ask you some questions about your child. Please select the best age, eye and skin color of your child.

- If you dont think your child fits any of the choices given, choose the best answer or, if you need to, skip the question.

- These questions are designed to help you find your child faster. Don't worry about answering a question incorrectly. You are not eliminating any photos from being shown. You will be able to look through the entire database, if necessary.

Similarity *(Technician MUST Explain)*
Moving to the next screen is usually done by selecting similar children (if any) and then clicking refine, (to refine the search).

- In the next series of screens you will see photos of children. At each screen, I will ask you if you see your child.

- If you do not see your child, I will ask you if any of the children look similar.

What do I mean by similar: *(In disaster drill, was marked as: "Must Say All Bullets". In dyad tests, was changed to "As Needed", but technicians said they mentioned the bullet points. For consistency and to make sure the parents understand, will be changed back to "Must Say All Bullets")*

- Choose children than have a similar look as your child.

- You do not have to justify or explain why similar.

- Ok to choose images even if they have different

  - Eye, skin hair color
  - Age
  - Gender
  - Etc.

- Don't over-think – go with your gut feeling.

- You may choose any number of similar looking children per screen, including none of them.

- Be aware that after you choose an image, there may be photographs of children that you think do not look similar to your child.

Other choices besides similarity: *In disaster drill, was marked as: "Must explain all but italics. Italics can be explained as needed for further clarification." In dyad tests, was changed to "As Needed". For consistency and to make sure the parents understand, will be changed back to the disaster drill instructions.)*

- <u>BACK:</u> If you want to change your selection from a previous screen, you can ask to go Back to see the previous screen again.

- <u>FORWARD:</u> If you have gone back, but change your mind, you can go forward.

- <u>SHOW RANDOM:</u> If you feel like you're continually getting screens of children that look nothing like your child, you can ask to Select Random to mix things up.

[NOTE: Select Random is different than simply not choosing any similar looking children and refining. If you don't select any similar children and then refine, the next screen will still show you children similar to all the previous selections you made. Select Random will literally show you a random sample of images. Use this if you feel like you have encountered several screens in a row where no children look at all similar.]

Once Completed *(Talking Points, if needed.)*

- If the system forces QUIT in the middle, that means they have missed their child on the screen. Explain this to them.

- You have completed the process! Thank you again. *(If they have another child start the process over again.)*

- Please fill out the following survey.

- **If the parent does not find their child: Refer them to Social Work.**