Finding Lost Children

Ashley Eden   C. Mario Christoudias   Trevor Darrell
UC Berkeley EECS & ICSI
{eden, cmch, trevor}@eecs.berkeley.edu

Abstract

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 effectively expedites 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 and attribute search. In this paper 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.

1. Introduction

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 [7], the rapid pace of evacuation efforts [7], and 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 most risk for adverse consequences. They are often unable or unwilling to give their name, address, or phone number [10].

After Katrina hit in 2005, more than 5000 children were separated from their families for as long as 18 months. 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 [1].

Currently in the US, there is no system in the public health infrastructure that effectively expedites reunification when children can't be identified. In particular, there is no central database of children separated from their families. After Katrina, parents had to drive around, checking hospitals, shelters, etc in a large multi-state radius. If they were lucky, each hospital would have a book of photos of everyone admitted. Many hospitals did not. International Red Cross data state that current methods for reunification remain primitive around the globe [2].

Working with the Children's Hospital Boston, we have engineered an image-based browsing system and central database to quickly reunify children with their families, should they get separated in a disaster. The work is part of a Federally funded grant, and based on the hospital's findings that many children can not give their information.

Based on the findings in [2], 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. 3) Parents can go to a designated center, input facial characteristics of their child, and search.

For privacy and the reduction of mental anguish, the parent should look through as few images as possible. Also, based on information from previous events [10], the system must be easy to use, and must address the need for surge capacity. Similarly, based on information provided by hospital ER workers who have worked through several disasters, we must assume that the parents may not have 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.

In this paper 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. The drill was performed with hospital workers, social workers, parents and children.

2. Previous Work

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 subfield of the rich field of Content Based Image Retrieval, or CBIR [5]. CBIR can be divided into three main categories: open browsing, category browsing, and target search. 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

978-1-61284-035-2/10/$26.00 ©2010 IEEE

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 the 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.

Target CBIR searches can be further divided into query-by-example and mental image search. 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. 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*. 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, [4, 6, 9]. Navarrete *et al*. [9] 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*. [4] and Fang and Geman [6] are more closely related to the system described here. Cox *et al*. describe a Bayesian method that, at each iteration, updates the probability $P(\text{image is target} \mid \text{browser history})$. Here, the browser history includes all previously displayed images and user actions. Fang and Geman is a specific application 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. These methods do not exploit extracted or labeled attributes. Below, we describe a hybrid attribute and CBIR browsing system.

3. System Overview

Figure 1 shows the general overview of a parent's search. The parent first enters some semantic attributes about her child. We believe it is useful to have the parent enter attribute labels for the child she is trying to find, and for attribute labels to already be stored with the images in the database. Working with the Children's Hospital Boston, we came up with a list of distinguishing attributes, such as eye color, skin color, and age. Attributes have been used in the past as a natural way to perform mental image search. Kumar *et al*. [8] successfully used attributes in their FaceTracer system, which was able to perform a better facial image search using attribute text queries.

After entering attributes, the parent browses over the photos. The browsing GUI we use can be seen in Figure 2. 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". Based on feedback from our collaborators at the Children's Hospital Boston, we included three other buttons in addition to "Refine" and "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. The 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.

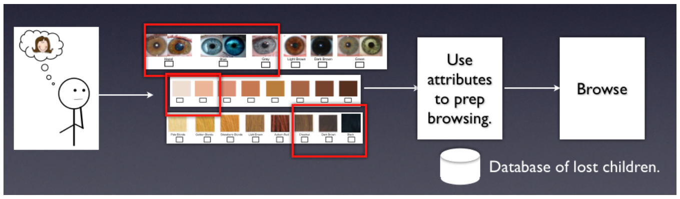
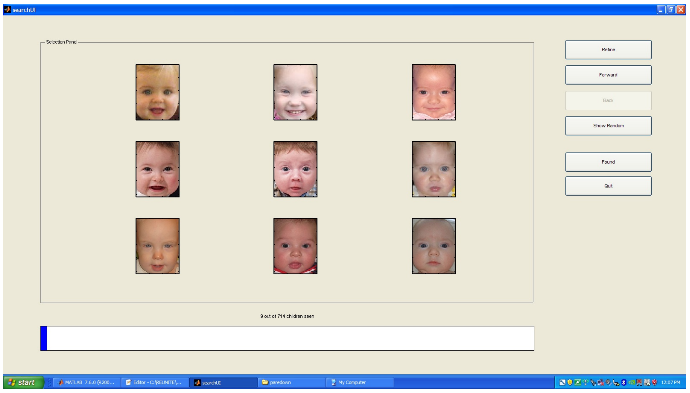


Figure 1. System overview.



9 out of 714 children seen

Figure 2. Screenshot of the search GUI.

4. Algorithm

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. In a real disaster, when children are admitted to hospitals, triages, etc, they will have their photos taken and their attributes labeled (see below), and this information will be uploaded to the central database. 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

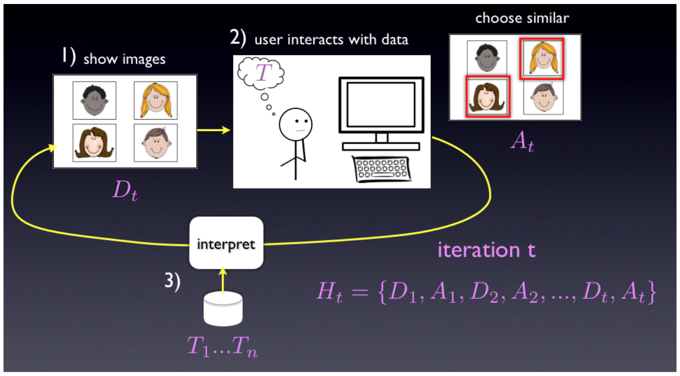


Figure 3. Overview of Cox *et al.*

database, the output of the algorithm will be the other meta-information stored with the image – *e.g.* the hospital name and room number the child is in.

The algorithm used in this system is based on the browsing system by Cox *et al.* [4], but with the addition of semantic features. Because of this, we will first give an overview of the Cox *et al.* method. Then we will describe how attributes were added.

4.1. Cox *et al.* Method

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 3 shows the three steps. 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 $P(\text{image is target} \mid \text{browser history})$ as

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

$P(A_t|T = T_i, D_t, H_{t-1})$ 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 do they update the user model. First, we must answer step 2, how the user interacts with the data, 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.

In our system, "Refining" updates the user model (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. We plan to 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.

4.2. Attributes

As previously mentioned, we initialized 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 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.

Attributes with binary labels are assigned prior values of either 0 or S, a *softness value*. This value can be tuned per attribute type. 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. These softness values affect how much the attributes influence the overall search. If all the softness values are 0, then the attributes would make no difference and there is just browsing. If they are all 1, 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 1. However, as we will discuss shortly, error can happen even when using "ground truth" attribute labels in the image database.

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 burdonsome 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 judgement 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.

5. Dataset

Testing the system required collecting a dataset of front-facing children's faces, over a range of ethnicities, eye colors, and age. These images were used as the children separated during a disaster. We downloaded thousands of images from the Parenting.com website [3], 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 later report on) were using some subset of 1213 of these images.

Note that in the field, a standard camera, a Canon PowerShot SD1100 IS, will be used, with flash, preferably indoors, to try to standardize the lighting. We have written an instruction manual 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.

5.1. Attribute Labels

In order to get the ground truth attribute labels for each of the browsing sets, we ran Mechanical Turk experiments on all 1213 images from the Parenting.com dataset. We decided to use eye color, skin color and age as attributes. In the tests described later, we used these "ground truth" attributes (since it is reasonable for a hospital worker to mark the information after taking the child's photo), but one could also try to automatically label the attributes. Part of the problem with automatic labeling, however, is that even the "ground truth" labeling has a lot of noise.

For the Mechanical Turk task, each image was labeled by 5 different people. Out of the 1213 images, only 1027 had an interrater 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 $.14$. The automatic extraction of these attributes is future work.

6. Tests/Results

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 real parents.

To do this, we ran a complete disaster drill 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. Seventeen children from 8 familes 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. When the volunteer was done, the information and photos were uploaded to the system and onto three laptops.

The laptops were then taken to a separate room, where the parents filed in as there was space. The parents had been being prepped in a separate room by social workers who were evaluating 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. Because some families had more than one parent participate, there were 20 searches total. Figure 4 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.

For the drill, we used eye and skin color, and age attribtues. 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 interrater agreement $> 60\%$, the mode label was used, and then grouped into the appropriate binary class. Images with too low an interrater agreement were hand-fixed with the appropriate binary classes. When a parent was asked to enter skin and eye color, she chose labels from the finer-level as input, and these labels were then grouped into the corresponding binary value.

Figure 4. Photo taken during the disaster drill.

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 softness values used for the eye and skin attributes were .6. 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 system over random browsing. 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. 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 are currently running experiments on real parents to evaluate browsing in these scenarios.

6.2. Synthetic Parents

We have also run a series of experiments on "synthetic parents" – i.e. people who are not parents, but who can 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.

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.

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 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 looking 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.

Because there are 1213 images in the browsing set with 9 images per screen, chance performance is 67.4 screens. Browsing only yields 23.5 screens with a variance of 22.2, and browsing with attributes yields 16.3 screens with a variance of 15.7.

7. Conclusion

In this paper we described a system to quickly reunify children with their families should they get separated in a disaster. To our knowledge, this is the first system specifically designed for pediatric reunification using CBIR. We presented a series of evaluations, including a realistic disaster drill set up, run jointly with the Children's Hospital Boston, which reported quick retrieval times for the overall system. Synthetic experiments demonstrated the merit of browsing, in addition to the usefulness of attributes. We are currently performing further evaluations of our system to better show how browsing is helpful, in particular how using browsing and attributes is better than purely using attributes. Additionally, we are running an experiment using automatically extracted attributes to show how browsing aids in the presence of noisy attribute labels.

Acknowledgements

This work could not have been done without Sarita Chung, Stephen Monteiro, Valerie Deleon, Phuc Nguyen, Brian Kulis, the students and faculty at UC Berkeley and ICSI, and the people at Children's Hospital Boston. Special thanks to Michael Shannon, who was the force behind the entire project. This study was supported in part by a grant from the Health Resources and Services Administration, (EMS-C Targeted Issues) Grant #H34MC10575-01-01.

References

[1] S. Chung. Pediatric disaster readiness: How far have we come? *Pediatric Disaster Readiness*, 10(3), 2009. 1

[2] 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. 1

[3] B. Corporation. www.parenting.com. 4

[4] 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. 2, 3

[5] 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. 1

[6] Y. Fang and D. Geman. Experiments in mental face retrieval. In *In Proc. of Audio- and Video-based Biometric Person Authentication*, pages 637–646, 2005. 2

[7] C. Johnston and I. Redlender. Summary of issues demanding solutions before the next one. *Pediatrics*, 117, 2006. 1

[8] 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. 2

[9] P. Navarrete and J. R. del Solar. Faceret: An interactive face retrieval system based on self-organizing maps. In *In Proc. of the CVIR*, 2002. 2

[10] C. on Environmental Health and C. on Infectious Diseases. Chemical-biological terrorism and its impact on children. *Pediatrics*, 118, 2006. 1