Face matching for post-disaster family reunification

Eugene Borovikov, Szilárd Vajda, Girish Lingappa, Sameer Antani, George Thoma
Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institutes of Health
8600 Rockville Pike, Bethesda, MD USA
FaceMatch@NIH.gov

***Abstract*—The National Library of Medicine (NLM) has developed People Locator™(PL), a Web-based system for family reunification in cases of a natural or man-made disaster. PL accepts photos and brief text meta-data (name, age, etc.) of missing or found persons. Searchers may query PL with text information, but text data is often incomplete or inconsistent. Adding an image-based search capability, i.e., matching faces in query photos to those already stored in the system, would significantly benefit the user experience. We report on our face matching R&D that aims to provide robust face localization and matching on digital photos of variable quality. In this article, we review relevant research and present our approach to robust near-duplicate image detection as well as face matching. We describe the integration of our face matching system with PL, report on its performance, and compare it to other publicly available face recognition systems. In contrast to these systems that have many good quality well-illuminated sample images for each person, our algorithms are hampered by the lack of training examples for individual faces, as those are unlikely in a disaster setting.**

I. Introduction

Natural or man made disasters can cause massive casualties and separate loved ones from their families. For information assistance in post-disaster family reunification, the Lister Hill National Center for Biomedical Communications at the National Library of Medicine (NLM) has developed a Web-based based system called People Locator™(PL)[1], to which photos and brief text meta-data (e.g. name, age, last known location) of missing or found persons may be posted. Searchers may query PL via a Web browser or via the ReUnite™iOS application, using textual information, but this data is often incomplete or variable, e.g. due to variations in names or their spelling. It would therefore be of a significant benefit to provide an image-based search capability, i.e. by matching faces in query photos to those already stored in the system.

This system is distinct from a face recognition system that has prior multiple photos of an individual available to train on. Rather, this system is designed on an information retrieval framework which matches a query face image to those acquired in a hospital or recovery camp following a disaster.

The FaceMatch (FM) task objective: *given a person's picture (as a digital photo) find visually similar faces in the PL database of pictures toward minimizing the manual browsing time and effort.*

There are many challenges to achieving this goal:

- Size of the database can be quite large. For example, the collection of records from the Haiti earthquake of 2010 is about $100,000$ records with over $15,000$ images.

Fig. 1. Challenging images for the modern face detection/matching systems

- No constraints on uploaded (query) pictures: they may contain zero or more faces and face-like objects e.g. cats/dogs/cartoon faces, as shown in figure 1
- Both query and database images could be of sub-optimal quality due to:
  - low resolution, as taken by older cameras, or poorly scanned,
  - noise from digitizing, compression, water-marking,
  - under- or over-exposed, under- or over-lit,
  - partially occluded, turned-away, damaged faces.
- Due to multiple reports of the same individual, the database may contain many near-duplicate images.
- There may be inconsistency in face appearance between the database image and the query image, e.g. due to facial hair, glasses, jewelry, or injuries sustained in the disaster.

We have evaluated several commercial[1,2] and open source[3,4] solutions to face detection and recognition, but were unsatisfied with the results because most of them (to work well) tend to require multiple shots of the same person made with fairly high resolution cameras in controlled or predictable environments. The images posted to and tracked by PL database do not answer those requirements, and cause the modern face matching systems to produce sub-optimal results.

Our FaceMatch R&D effort addresses many of the mentioned challenges. The PL system can now support:

[1] PittPatt - recently acquired by Google
[2] FaceSDK - http://www.luxand.com/facesdk/
[3] FaceL - http://www.cs.colostate.edu/facel/index09.php
[4] OSSK - http://www.openu.ac.il/home/hassner/projects/Ossk/

- near-duplicate image detection, grouping and removal,
- semi-automatic data-set annotation for faces and skin,
- robust to noise skin detection,
- accurate face detection,
- ensemble based combination of features for improved face matching.

The resulting image retrieval system is geared towards face detection and matching. It is equipped with robust image processing techniques (helping normalize photographs and detect near-duplicates), modern computer vision methodology (for multiple face detection and matching), and information retrieval machinery (for efficient, adaptive indexing) to address the posed challenges.

II. Related work

In this section, we review some relevant research describing the methods we drew upon and utilized in the implementation of our FaceMatch system.

A. Image de-duplication

Image set de-duplication is one of the first stages in face matching systems, as it helps reduce the dataset by removing or grouping near-duplicate images. Following a disaster, people seeking to report on (or search for) loved ones tend to upload the same or a slightly modified (rotated, cropped, scaled, etc.) copies of the same photo to various Web sites and systems. These systems often collaborate to improve the chances of finding missing people. The resulting (shared) data collection is very likely to contain duplicate information which can result, not only in poor quality matches, but also can considerably slow down the matching process itself. Hence, it is desirable to discard the redundant information and select the most appropriate high-quality photos for the matching.

Jacobs et al.[2] present an fast and robust to scale image/sketch matching method by constructing a 2D Haar wavelet based image descriptor. The descriptor is fast to compute, requires little storage for each database image, and results in a similarity measure that improves significantly upon the L_1 or L_2 metrics in discriminating the targets of fuzzy image matches. It is also fast to compare and is robust to noise and scale. It uses 128x128 YIQ color images[3] from which only 40-60 most significant wavelet coefficients are needed for computation. We noted that the method's reliance on the Haar wavelet makes the descriptor sensitive to image translation, cropping, and rotation.

To address this sensitivity to affine transformations, one could utilize the Steerable Pyramid [4], [5], [6], a linear multi-scale, multi-orientation image decomposition that provides a useful tool for robust image matching. The steerable pyramid performs a polar-separable decomposition in the frequency domain, thus allowing independent representation of scale and orientation. More importantly, the representation is translation-invariant (i.e., the sub-bands are aliasing-free, or equivariant with respect to translation) and rotation-invariant (i.e., the sub-bands are steerable, or equivariant with respect to rotation). The primary drawback is that the representation is over-complete by a factor of $4k/3$, where k is the number of orientation bands.

Chum et al.[7] propose two image similarity measures for fast indexing via locality sensitive hashing. The proposed method uses a visual vocabulary of vector quantized local feature descriptors (SIFT) and for retrieval exploits enhanced min-Hash techniques. They propose a way of exploiting more sophisticated similarity measures that have shown to be essential in object image retrieval. They focus primarily on scalability to very large image and video databases. The method requires a small amount of data need be stored for each image. They show results on the TrecVid 2006 data set which contains approximately 146,000 key frames, and also on challenging the University of Kentucky image retrieval database.

B. Face detection

The problem of robust face detection naturally arises in many multimedia retrieval applications dealing with images of human faces. It has been studied quite extensively over the recent years and multiple on-line and off-line solutions to this problem have been proposed [8], [9], [10], [11], [12], [13], [14], [15].

Zhang and Zhang[13] provide a survey of recent advances in face detection for the past decade, referencing the seminal Viola-Jones face detector, as well as the various techniques according to how they extract features (e.g. Haar-like, pixel-based, binarized, generic linear, statistics-based, composite, and shape) and what learning algorithms are adopted (e.g. template matching, Bayesian, SVM, neural nets, and part-based).

Viola and Jones [16], [10] introduced an object detection process that can be robust and efficient, if it is based on the detection of features that capture some information about the class to be detected. This is the case of Haar-like features that encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts exhibited by a human face and their spatial relationships. Haar-like features are so called because they are computed similarly to the coefficients in a Haar wavelet transform. The OpenCV implementation of this method works quite well for the web-cam quality (or better) images making the real-time near-frontal up-right face detection possible. However, the method often fails to detect faces in unconstrained images partly because it is implemented to be color-blind, taking no advantage of such image clues as skin tone, which may be quite suggestive of the face presence or absence.

Chhaya and Oates[17] present one of the successful attempts to make unconstrained face detection possible for the case of hospital patient triage, where face images exhibit a great variety in pose, occlusion and skin tone. They noticed that the standard face detection algorithms perform poorly on triage images taken during disaster simulations, and proposed an ensemble-based face detection method that combines robust skin detection (in Lab color space) with simple pattern matching face detection techniques, and show that this considerably improves the face detection accuracy (e.g. compared to OpenCV) on an image set of about 100 patients. Our approach goes deeper with the skin color detection by working with the extended color space and using artificial neural net (ANN) to

classify skin pixels. We also handle larger datasets, all of which are publicly available.

Hoffmann et al.[12] introduce a hierarchical classification approach to face detection, where discrete Gabor jets (DGJ) are used for extracting brightness-related features (for preliminary classification), then a skin detection algorithm is employed, showing that the use of color efficiently reduces the number of false positives while maintaining a high true positive rate. In comparison with the Viola-Jones Face Detector (VJFD) this method shows higher correct classification rates. Our face detection strategy (although not using DGJ) goes further and attempts to recover some false negatives by locally enhancing the likely skin patches for the VJFD to re-iterate.

Zhu and Ramanan[18] present a unified model for face detection, pose estimation, and landmark estimation in real-world, cluttered images. Their model is based on a mixtures of trees with a shared pool of parts, showing that tree-structured models may be effective at capturing global elastic deformation, while being easy to optimize. The results on standard face benchmarks and on "in the wild" annotated dataset show how this system advances the state-of-the-art for all three tasks in good accuracy, but may be rather slow and admittedly adapted to high-resolution (80x80 or larger) faces. It may be more precise than our method, but it tends to miss many low resolution faces, that our system needs to work with.

C. Skin tone detection

Grayscale face detection can often be improved by robust human skin tone detection. The literature typically distinguishes between region-based and pixel-based color detection methods for skin detection. In contrast to region based methods[19] which consider both color information and location information, pixel based skin classification methods [20], [21], [22], [23] typically operate on individual pixels without considering its spacial context. As a result, pixel based color classification is more robust to the affine transforms and has much lower computational complexity than the region based alternative.

A number of well know color spaces (e.g. RGB, HSV, LUV, Lab, YCbCr, etc.)[24] were considered for skin tone classification in many experimental systems. More restrictive methods use explicit skin definitions [25], [26], while others estimate skin tone distributions from the available data. The non-parametric methods estimate the color distribution by histograms and build normalized lookup tables[21]. Some other methods consider parametric models where the data distribution is typically modeled by a Gaussian mixture[22]. We have also seen that machine learning solutions involving SVMs or neural networks[20] perform very well at the skin color detection task.

D. Face matching

Face matching in general conditions with some minimal (one, seldom two) reference images is an open problem and is an active area of research. Some promising methods have been proposed over the years. Here we list a few that are relevant to the problem we solve.

A SIFT-based energy minimization method described by Luo et al.[27] serves as a general object matching framework, but can be applied to face matching in general conditions as well. This method has shown to be fairly robust to scale, lighting and affine transforms. Although suitable for general purpose object identification, its face matching performance also appears quite promising. Our face matching approach is similar because we are also using SIFT key points and descriptors, but we are skipping the energy minimization routine, which can slow down the query response; instead, we are utilizing an ensemble of descriptors, which can be easily parallelized.

Bay et al.[28] present a scale- and rotation-invariant interest point detector and descriptor, coined as SURF (Speeded Up Robust Features). It can be utilized for object and face matching, and it appears to approximate or outperform previously proposed object matching schemes (e.g. based on SIFT) with respect to repeatability, distinctiveness and robustness, yet it can be computed and compared somewhat faster. The method relies on integral images for image convolutions, builds on the strengths of the leading existing detectors and descriptors, e.g. using a Hessian matrix-based measure for the detector, and a distribution-based descriptor. We are utilizing the quick SURF descriptor in our ensemble matching, also noticing the matching benefits of using SIFT descriptors on SURF key points.

Wolf et al.[29] presented an interesting approach to face matching called the one-shot similarity kernel, using a special similarity measure to produce some impressive face matching results on e.g. Labeled Faces in the Wild (LFW) collection. Given two vectors, their one-shot similarity score reflects the likelihood of each vector belonging in the same class as the other vector and not in a class defined by a fixed set of negative examples. They showed that: (1) when using a version of Linear Discriminant Analysis (LDA) as the underlying classifier, this score is a Conditionally Positive Definite kernel and may be used within kernel-methods (e.g. SVM), and (2) it can be efficiently computed. We are not utilizing this method in part because our approach is explicitly required not to use any training.

Bolme et al.[30] introduced a real-time system for face matching and labeling called FaceL that labels faces in live video from a webcam. FaceL presents a window with a few controls and annotations displayed over the live video feed. The annotations indicate detected faces, positions of eyes, and after training, the names of enrolled people. Enrollment is video based, capturing many images per person. FaceL distinguishes between a small set of people (e.g. 3-5) in fairly uncontrolled settings and incorporates a novel incremental training capability. The system is reported to run at about 10 frames per second on their hardware, but we are unable to utilize it, since our approach is required to be training-less.

Zhou et al.[31] describe how to combine perceptual features with diffusion distance for face recognition by incorporating spatially structured features into a histogram-based face-recognition framework. While diffusion distance is computed over a pair of human face images, the shape descriptions of these images are built using Gabor filters that consist of a number of scales and levels, which enables the system performance to be significantly improved, compared to several classical algorithms. The oriented Gabor filters lead to discriminative image representations that are then used to classify human

faces in the database. We are not utilizing this method because it requires training, and Gabor filters are rather slow.

III. Image Repository

Before any face/object detection or recognition takes place, an image collection needs to be turned into a consistent and annotated image repository. With our FaceMatch, we primarily use image collection from disaster areas, such as Haiti Earthquake (HEPL)[1]. The data-set consists of 15,000 mostly color, low quality images, some of which are shown in figure 1. We developed several image processing tools for it and use them to

- reduce the set by about 30% via identifying near-duplicates and no-face images,
- identify and label faces and profiles as rectangular regions, and
- localize skin and facial features (eyes, nose, mouth) within the face regions.

We manually annotated 4000 HEPL images and found that the majority of face regions are with the frontal views and about 900 are with the profile views. The distribution of face region dimensions is quite diverse. The average face diameter is 40 pixels with a standard deviation of 28 pixels. The average profile diameter was 50 pixels. For skin color, 7680 images were selected and 33,431 from faces, arms, legs were manually annotated resulting in a total of 13,113,994 pixels, of which 7,314,106 pixels were skin and 5,799,88 pixels were non-skin.

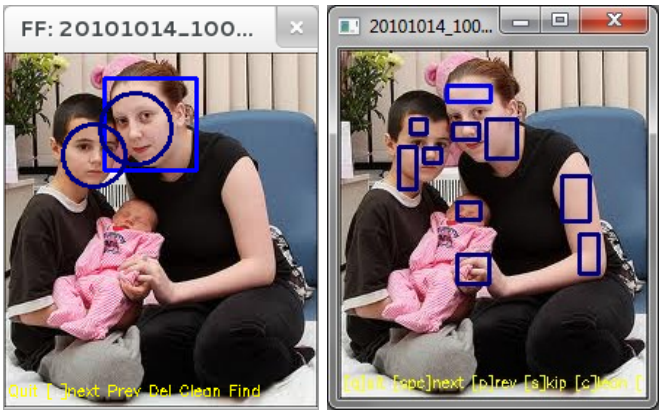


Fig. 2. Annotated images: face/profile (left) and skin (right) regions

Some additional annotation (e.g. ethnicity, age/group, gender) is also possible with the GUI developed by us, as shown in figure 2. This cross-platform desktop annotation tool aids the annotator with semi-automatic face/profile detection, correction and annotation. The annotations repository is tracked by a database, updated with the new information about existing and new images, and used for improving face detection and matching performance.

IV. Near-duplicate image detection

The raw image data-set typically contains many duplicate images due to multiple postings of the same photograph. Such duplicates (or near-duplicates) need to be identified and collected in bunches that would be represented by the highest quality images in the groups.

Strict duplicate images can be identified with high confidence by comparing their checksums or hash codes. However, some images depicting (nearly) the same scene posted to the image repository could come from the same source (e.g. digital photograph), but be cropped, resized or re-compressed. We define such images as near-duplicates and report on our efficient method of locating and grouping them. Each group of near duplicate images needs to be represented by its best member, e.g. image with the highest resolution and lowest noise.

We process the list of image files sequentially. For each input image, we compute its image descriptor and separate a candidate image from a group of similar processed images. A candidate image is one which meets the following criterion: the *lowest matching distance with its peers* (where the matching distance is below a specified tolerance) and has the highest resolution of the group of images. We could add other criteria (e.g. minimum noise) later. On completion of this process each group of near duplicates would be represented by its champion.

We use an efficient content-based image retrieval technique developed by Jacobs et al.[2] that provides fast image matching procedure that is robust to image noise, compression and scale. By this method, all images in the source data-set would be represented by their Haar wavelet based descriptors. In our implementation, we use 60 most significant wavelet coefficients.

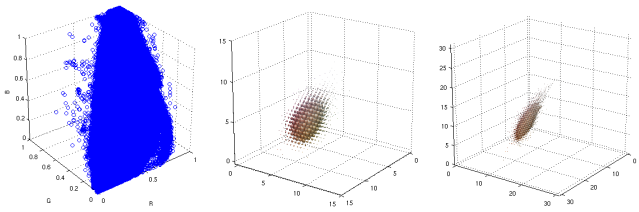
V. Skin localization

Skin localization is a strategy for improving the existing face detection methods due to the fact that human skin lies in a specific color range[21]. However, skin detection methods often face many more challenges, e.g. surfaces with skin-like tones, lighting conditions, make-up, specular reflection due to moisture, camera settings, etc.

Researchers have been studying various color spaces[24] for skin detection over the last couple of decades. However, the existing skin detectors [20], [21], [22] are typically limited to using a single color space. In our skin classification framework, we include attributes of several color spaces in a single composite feature vector, which is built by concatenating the color band values in a composite color vector.

In our method, we compute pixel color values over several commonly used color spaces, e.g. XYZ, RGB, Lab, CYMK, NTSC, HSV, Luv, and YCbCr [20], [21], [22]. We apply non-linear injection of the sampled RGB color into a high-dimensional extended color space, which provides for a more robust skin tone clustering, highly effective feature analysis, and subsequent skin tone classification. We studied concatenating color representations for our face detection needs on unconstrained mixed quality images. The optimal combination was determined to be [RGB, HSV, Lab], spanning over 9-dimensional extended color space (ECS).

To classify the pixels into $skin$ and $\neg skin$ (non-skin), we studied several methods.



(a) (b) (c)

Fig. 3. Representation of skin color in (a) RGB space, (b) BKS space $\in [1:16]^3$ and (c) BKS space $\in [1:32]^3$.

A. Color histograms

Color histograms represent the underlying skin tone distribution in a specific color space [22], [21]. The color space is quantized and the histogram provides a probability distribution of the color in the image as

$$P(c_i) = \frac{count(c_i)}{N} \tag{1}$$

where c_i gives the count of the i^{th} component of color c, while N is the total number of color data points used. These probabilities correspond to the likelihood that a given color belongs to skin. The classification into skin and ¬skin is based on a comparison of the likelihood to an empirically defined threshold. The main advantage of this method is its computational speed, as only some indexing operations and a comparison is necessary. By its nature, the method does not presume any type of predefined distribution for the skin colors, as may be the case for the parametric statistical classification. However, this approach requires a substantial amount of training data for realistic look-up tables (LUT).

To overcome the drawback related to the thresholds for each dimension of the input space, a more sophisticated histogram was built, similar to the Behavior Knowledge Space[32], but instead of counting the votes coming from different classifiers, we count the frequency of a given input to fall into a specific bin of the space. In our case, the classifiers are represented by the different bins where the color values should be distributed, as shown in Figure 3. The threshold for the decision was reduced to one single value, selected based on many trial runs. The size of the spheres in Figure 3b and Figure 3c represent the likelihood of a given input to be skin.

B. Parametric distribution modeling

Let $x = [x_1, x_2, \ldots, x_m]^T$ denote the input vector describing an input pixel in the extended color space (ECS). Assuming that the skin tones are approximately normally distributed in ECS, we can write the likelihood of a pixel color coming from a skin region as

$$d(x) = exp\{-a(x-\mu)^T \Sigma^{-1}(x-\mu)\} \tag{2}$$

where μ denotes the mean vector and Σ stands for the covariance matrix estimated from the training skin pixels, and a is a positive distribution bandwidth factor chosen experimentally. Thresholding $d(x)$ at some level $\theta \in [0,1]$ classifies the pixel as skin or non-skin. For our choice of $a = 0.25$, we set $\theta = 0.5$.

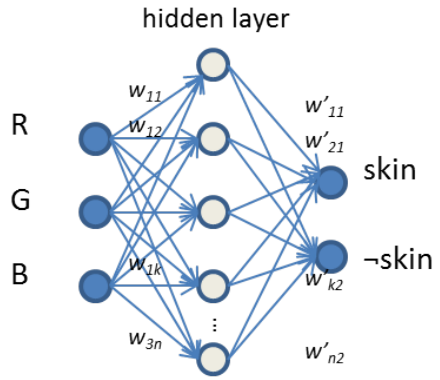


Fig. 4. The artificial neural network architecture for skin classification in RGB color space.

The main advantage of this method is its low computational complexity, and good computing speed. The main limitation of the method is the assumption of normal distribution of skin tone potentially limiting its generalization power, but given our skin color distribution (as in Figure 3), we can assume the skin tone normality in our experiments.

C. Artificial neural net (ANN) classifier

A more sophisticated method for skin color detection involves a neural network that can learn a skin likelihood map considering skin and non-skin pixels. A fully connected multilayer perceptron (MLP) based artificial neural network (ANN) with one hidden layer is proposed to model the decision surface in the color space, as shown in Figure 4. The input layer corresponds to the number of color components describing the analyzed pixel, while the output layer contains two neurons (units) to distinguish between skin and non-skin.

The number of units in the hidden layer was set experimentally to 16. Based on our experiments it was found that doubling the number of neurons in the hidden layer compared to the input layer provides the most accurate scores. A speed constraint was also considered in order to provide a fairly fast decision - an absolute necessity for an interactive system.

While a statistical decision (by parametric or non-parametric models) is a fairly simple (and fast) procedure, the neural net solution adapts its weights during each epoch[5] invoking back-propagation learning strategy, which can assure a certain optimum. In our experiments, $\alpha = 0.02$ (learning rate) and $\beta = 0.08$ (momentum) were considered.

VI. Face Localization

Robust and reliable face detection and localization capability in digital photographs is the function of central importance to the described system. It is one of the first critical steps in toward the end goal of face matching. It is used in many modules including spurious image removal, localizing faces for annotations and matching, presentation of the results. Face detection and localization is one of the first critical steps of our face matching system. Its key challenges are the low resolution,

[5] a step in the training process of an artificial neural network

low quality images that are likely in an unconstrained image acquisition situation.

Given a set of images, the system needs to inspect each of them for the presence of human faces in near-frontal and partially turned (profile) views. The detected face regions need to be associated with their images and recorded to the database for the subsequent face matching task. It is imperative that the face detection subsystem should allow for effective visual analysis of the detected faces and profiles via a simple GUI that displays both, the image being analyzed and the detected face/profile regions. The GUI needs to allow the user to correct the automatic face detection mistakes by removing the erroneous face/profile regions and providing the correct ones, possibly with some meta-information about the subjects, such as age, gender, ethnicity, or name, if available. The resulting system should be robust to the scale, shift and rotation of the presented face regions.

We use the object detector in OpenCV for face detection has been initially proposed by Viola and Jones[10], [16], working with Haar-like features as in Figure 5. This classifier can be applied to a region of interest in an input image. A search window is internally used to analyze image subregions to find faces. The classifier is designed so that it can be easily

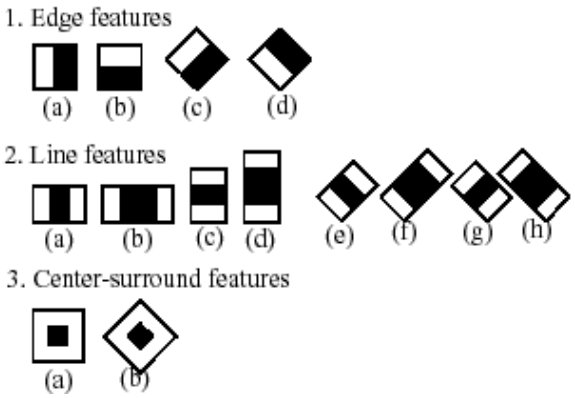


Fig. 5. Haar-like features [docs.opencv.org]: vertical (1.a, 2.a-b), horizontal (1.b, 2.c-d), diagonal (1.c-d, 2.e-h), central (3.a-b).

rescaled in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. Hence, to find an object of an unknown size in the given image, the scan procedure should be done several times at different scales.

Our face detector is equipped with two major cascade classifiers: one is for detecting frontal face views, and the other is trained to detect the profile views. We apply them sequentially and record the results of both, even if some detected regions overlap. The classifiers assume faces to be in close to vertical positions is limited to detect vertically oriented faces, which may not be always the case for some pictures in our data-set. To overcome this limitation we provide an option to automatically consider image $\pi k/2$ angle rotations, where $k = 0, 1, 2, 3$. In addition to the face/profile detectors, we also provide an option to detect commonly found facial elements, such as glasses, eyes and nose-mouth regions, which might be useful for computing features in the face matching stage.

The skin tone helper module, described earlier is a crucial part of the FaceFinder system. Only sufficiently large connected skin regions are considered for processing. The remaining image regions are enhanced for contrast based on the skin likelihood of each pixel (providing a lighter tone for the skin and a darker tone for the background) and cropped after padding from the original image and fed into the face localizer again to identify new possible faces not found originally by the system. The benefits of this approach are: a) it can discover new faces originally missed by the system, and b) it can overrule some previous false decisions. Use of the face detector has resulted in improved face localization accuracy, as our experiments show.

VII. Face matching

Face matching queries can be posed to the system after the face/profile regions in the image collection are localized and their descriptors are indexed. The face matching method needs to be robust to accommodate wide variations in the appearance, and it needs to be fairly exact to eliminate numerous false positives. When the system is provided a query image with a face, its goal is to localize that face and then match it against the repository of faces that were indexed by the face descriptors, and output a sorted list of most similar faces in the similarity descending order. The matching technique cannot assume that many faces of the same subject are present in the database, and it needs to be robust to illumination, scale and affine transformations.

We experimented with several promising object/face matching techniques, such as *One-Shot Similarity Kernel* by Wolf et al.[29], SIFT-based energy minimization method called *SoftCBIR* described by Luo et al.[27], SURF-based descriptors by Bay et al.[28], and ORB-based descriptors by Rublee at al.[33] as a compromise between SIFT and SURF. The most promising method for our needs turned out to be the SIFT-based descriptor. We adapted its OpenCV implementation by computing the SIFT descriptors on the cropped faces that were extracted by our face detector. Both stages (detection and matching) work with intensity images. We tested our method on HEPL and Labeled Faces in the Wild (LFW) data-sets and were sufficiently encouraged by the outcomes that we decided to integrate our adaptations of SIFT, SURF and ORB descriptors into the PL system for face matching needs.

Having several image descriptors per face (HAAR, SURF, SIFT, and ORB), we experimented with similarity distance-based and similarity rank-based feature combination strategies. The distance-based combinations used individual distances d_i and weights w_i:

weighted distance product $d = \prod {d_i}^{w_i}$ with the constituent distances and their weights typically in $[0, 1]$

decreasing confidence radical $d = \sqrt{d_1 \sqrt{d_2 \ldots \sqrt{d_n}}}$ with d_1 being the most confident (heaviest) distance

The rank-based combination procedure implemented a weighted variant of Borda count[34], so that the candidates rank-based points are multiplied by the descriptor weights.

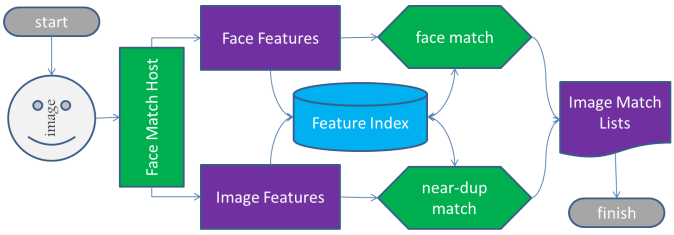


Fig. 6. FaceMatch module scheme

VIII. System

As mentioned before, the PL system allows searching record collections on missing persons using text queries. The FaceMatch (FM) extension of the system adds a visual modality to the search. With FaceMatch integrated into PL, information seekers are able to search the database by text and/or image queries. An image would typically start its journey through the FaceMatch module in the FaceMatch Host service that would extract the whole image features (for near-duplicate detection) as well as the facial features (for face matching). As schematically shown in figure 6, the extracted image and face descriptors could optionally be indexed by the FM DB Feature Index to answer subsequent visual queries that result in matching image lists, ordered by the descending similarity. The output of the Face Match module is then optionally fused with the text query results and the overall output list is presented to the user. The user is free to browse the PL database by inspecting the details of the retrieved records and optionally re-submitting queries using the retrieved faces as examples.

The core FaceMatch (FM) imaging code is written in portable C++, and for the web service integration purposes it is packaged as a shared library. This provides a suitable architecture to enable its use over the Web as a Web service, as this appeared an industry standard for exposing services in a platform agnostic way. The integration efforts exposed the native binary to a managed, garbage collected .NET environment by wrapping the core FaceMatch library with a C++ binary interface using the COM/ATL technology stack. A run-time callable wrapper now maps the types of the .NET run-time to the native types recognized by the C++ DLL and marshals the passed data and the returned results between the managed and the native environments.

A key focus during the integration was to ensure the top performance across all image descriptor basic operations, e.g. list, ingest, query and remove. We provided the design to take advantage of the industry trend toward multi-core architectures by exploiting task level and functional parallelism inside all critical modules. For instance, the Web service is capable of servicing multiple queries while performing an ingest or a deletion. The modules use the .NET task parallel library, as well as OpenMP and its synchronization constructs.

IX. Experiments

This section gives a description of the data sets we used in the different experiments, the evaluation protocols we followed and the results we obtained running different modules of the system.

TABLE I. Whole image matching accuracy results using Haar wavelet method utilized in near-duplicate detection

Distortion	Recall	Precision	F-score
rotation	0.69	0.62	0.65
crop	0.71	0.70	0.71
scale	0.99	0.99	0.99

A. Near duplicates detection

The near-duplicate detection capability of the system was tested first on our HEPL data set. The collection contains over 15,000 images gathered during different real disaster events. People tend to upload the same or very similar photos several times thus producing redundant data. Some 6,000 images were found as being duplicates or near-duplicates. This reduction of the collection to 65% of the original size allowed to speed-up our queries roughly by the factor of two. The evaluation was based on visual inspection as there was no annotation available for this image collection at the time.

For a more precise evaluation, we have also experimented with synthetically generated distortions of the original images introducing 792 near-duplicates from a set of 132 unique images by scaling (by the factors of 0.5 and 2), rotating (by the angles of $\pm\pi/12$) and cropping (by the factors of 0.9 and 0.75) the original images. Table I exhibits the results for full image matching using the Haar wavelet based method used in near-duplicate detection. We have discovered that our near-duplicate detector is most sensitive to rotations, then to cropping, missing quite a few of those, while detecting most of the scaled near-duplicates correctly. This behavior was expected, given the Haar wavelet's nature of the detector.

B. Face localization

For localization purpose we have considered a large variety of image collections. We selected from the previously mentioned HEPL data collection different subsets containing 500, 1881 and 4,000 images, respectively. These images contain a large variety of faces in unconstrained environments, containing small and large faces alike. Some of them are over-exposed, blurry or occluded as shown in Figure 1. For direct comparison purpose we also considered the well-known Caltech face collection[6] containing some 450 images of 27 different persons.

Comparing the results in Table II achieved on different image data collections, we can state that the Viola-Jones face detector (VJ) originally designed to deal with gray-scale images can radically be improved by the skin color information. Our skin detection step not only helps to reduce the number of unlikely faces, but it also can help recover faces which were originally missed by the baseline detector. For the Caltech dataset, the ANN based skin detector considerably improves the precision, while keeping the original high recall rate, hence significantly improving the F-score. This may be due to the fact that on Caltech image collection, the original VJ detector often gets confused by the cluttered background, and the ANN skin map drives it to the true faces. For the different HEPL collections there is always a certain gain either on the recall or precision which shows us that by using the appropriate

[6]http://www.vision.caltech.edu/html-files/archive.html

TABLE II. FACE DETECTION SCORES (PRECISION, RECALL, F-SCORE) ON DIFFERENT DATA SETS USING A BASELINE SYSTEM (VJ), DIFFERENT SKIN IMPROVEMENT BASED SYSTEMS (VJ+SKINSTAT, VJ+SKINANN AND VJ+SKINHIST), LEADING OPEN SOURCE SYSTEM (ZHU-RAMANAN[18]) AND COMMERCIAL SYSTEMS (IOS[35], FACESDK[36]).

Data	Method	Recall	Precision	F-score
HEPL-500	VJ	0.76	0.87	0.81
	VJ+SkinStat	0.77	0.89	**0.83**
	VJ+SkinANN	**0.81**	0.84	0.82
	VJ+SkinHist	0.71	0.87	0.78
	iOS	0.68	0.87	0.76
	FaceSDK	0.73	0.87	0.79
	Zhu-Ramanan	0.33	**0.92**	0.49
HEPL-4000	VJ	0.45	0.81	0.58
	VJ+SkinStat	0.47	0.82	0.60
	VJ+SkinANN	**0.51**	0.81	**0.63**
	VJ+SkinHist	0.44	0.82	0.57
	iOS	0.42	**0.88**	0.57
	FaceSDK	0.47	0.86	0.60
HEPL-1881	VJ	**0.39**	0.77	**0.51**
	VJ+SkinStat	0.35	**0.78**	0.49
	VJ+SkinANN	0.38	0.76	**0.51**
	VJ+SkinHist	0.33	**0.78**	0.47
	iOS	0.29	0.75	0.42
	FaceSDK	0.33	0.75	0.46
Caltech	VJ	0.95	0.88	0.91
	VJ+SkinStat	0.97	0.97	0.97
	VJ+SkinANN	**0.98**	**0.97**	**0.98**
	VJ+SkinHist	0.90	0.68	0.80
	iOS	0.97	**0.98**	0.97
	FaceSDK	0.96	0.94	0.95
	Zhu-Ramanan	0.97	0.97	0.97

skin map, we can recover originally not seen faces as well to discard non-faces.

When comparing our face localization to those of other face detection systems, such as iOS[35] and FaceSDK[36], it was found that on HEPL-1881 dataset FaceFinder has a higher accuracy than iOS or FaceSDK. On HEPL-1881 FaceFinder also had a better precision and recall than the other two face detection systems, as Table II illustrates. We noticed that on the high quality Caltech faces, VJ+SkinANN improves considerably recall and precision of the base VJ, and shows most accurate face detection performance compared to other open-source and commercial engines.

C. Face matching

For the face matching experiments, we considered a data post-disaster collection (HEPL-4000) and the Caltech faces data. Due to the nature and source of our target image collection (HEPL), we do not posses different pictures from the same person. A sample visual query result are shown in figure 7, and we can see how the system retrieves the faces similar to the query in the similarity descending order, observing the clear score gap after the self match. Since HEPL-4000 does not contain multiple photos of the same person (aside from the near-duplicates), the self-match accuracy score was naturally 1.

A more realistic test scenario for the FaceMatch was to consider the Caltech data where different pictures of the same subjects can be found. For that purpose, we normalized the detected faces to 128x128 pixel patches and combined the matching power of different descriptors (HAAR, SIFT, SURF and ORB) using our three combination techniques described earlier. The results shown in Figure 8 provide a clear view on the quality of the different combination techniques. The distance radius based queries for threshold values $d \in [0, 1]$

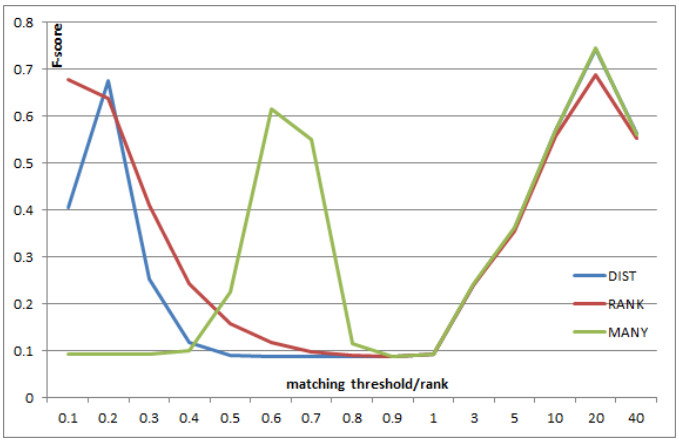


Fig. 8. FaceMatch result on the Caltech data considering different feature representations combined by weighted distances (DIST), Borda count re-ranking (RANK) and decreasing confidence radical (MANY).

(0 = exact match) appear to peak around 0.2 for RANK and DIST, which indicates the good ensemble re-ranking power, where individual descriptors would usually peak between 0.6 and 0.8, as MANY reflects. The peak matching accuracy, however, appears to be in the top-20 match queries for all combination strategies at about 0.74 for MANY. For the lower resolution 64x64 pixel patches, the highest top-20 F-score was 0.65, achieved by the DIST combination, which boldly suggests that the face resolution plays an important role in the matching phase, allowing to detect more key points and better match their descriptors in high resolution faces. However, in our current post-disaster datasets (e.g. HEPL), the majority of the pictures are low-resolution containing small faces.

To study individual face descriptor matching performance, we performed additional experiments with a 62 image annotated subset of HEPL images, considering the artificial variations: scale (factors 0.75 and 1.25), cropping (factors 0.75 and 1.25) and rotation (angles $\pm\pi/12$), ending up with a total of 372 variations of the original faces. The F-score accuracy figures shown in table III were obtained in leave-one-out experiments. This experiment showed the robustness

TABLE III. FACE MATCHING RESULTS ON SYNTHETIC HEPL-372

method	HAAR	SIFT	SURF	ORB	MANY	DIST	RANK
accuracy	0.67	**0.91**	0.88	0.71	**0.97**	0.96	0.86

of the SIFT descriptor to the different affine transformations. We also need to mention that the extraction of the SIFT key points and their descriptors is the slowest among the rest of the descriptors. HAAR was the fastest, but the least accurate out of them. ORB was also fast, but not as accurate as SIFT or SURF. Out of the three combination strategies, MANY was the champion, significantly more accurate than the best of the individual descriptors.

Considering the results for the HEPL-4000, the Caltech and HEPL-372 data set, we conclude that even though SIFT descriptors are the most accurate in our experiments, the matching based on SURF may be a realistic compromise between SIFT and ORB in terms of speed and accuracy. We also observed that with the correct descriptor ordering, the decreasing confidence radical (MANY) combination method

Fig. 7. FaceMatch sample visual query results on the HEPL-4000 data. Similarity computed as (1-distance).

can significantly outperform the most accurate individual descriptor in the radius-based retrieval. The weighted DISTance approach can compete with MANY without descriptor ordering constraints, while the re-RANK-ing was too much affected by the accuracy lacking HAAR and ORB descriptors.

X. Conclusion

With an aim to add the image based query capability in the People Locator™(PL) [1] system, we researched and developed several image matching and face recognition methods, evaluated a few state-of-the-art systems on existing datasets, developed a software library for: image near-duplicate detection (based on Jacobs et al.[2] work), face detection (based on Viola-Jones[10] work) and face matching (using a combination of SIFT, SURF, ORB and Haar descriptors), and developed Web services to allow PL to use the FaceFinder and FaceMatch software.

The major features that distinguish our face matching system from the majority of the available face recognition systems include:

- it works with *unconstrained images* with no prior knowledge of resolution, lighting or head pose,
- it is *training-less*, i.e. no training data available before face matching takes place, and
- it provides an *end-to-end image/face match solution* that can be deployed on desk-top and over the Web

We have developed the technology that exceeds in accuracy and speed many current open source and commercial solutions. Our 15,000 PL image dataset was reduced by about 30% by detecting image near-duplicates and no-face pictures, which helped dramatically reduce the overall query turn-around time. The PL image repository was also partially annotated with the correct face/profile locations and facial features (eyes, nose, mouth) using the desk-top and web-based annotation tools calling our FaceMatch library.

We have made several important improvements to the methods with regards to accuracy and speed, and our major technological contributions hence include:

Haar wavelet matching has been adapted to perform two functions: near-duplicate detection and image/face matching, using different thresholds and coefficient weights for each application. Besides fast and robust image set de-duplication, this method allows for quick whole-image and face-patch queries, but in its current form it is quite sensitive to cropping and rotation.

Face detection was improved by using human skin tone information, locating and enhancing skin tone regions that are fed as the input to the default (color-blind) face detection algorithm. The skin regions were located using the traditional statistical techniques (using mean and covariance of the skin colors in RGB, HSV and Lab), as well as via training an artificial neural network (ANN). The skin region enhancement was implemented similarly to the white point balancing algorithm. On several PL subsets, our face detector was more accurate than the available state-of-the-art engines, both commercial[36], [35] and open-source[10], [18].

Face matching utilized a descriptor ensemble approach to face matching to optimize the matching accuracy. Both distance- and rank-based weighted query result combination were utilized. Increasing the weight of the strong descriptors while diminishing the weaker ones typically improves the matching accuracy beyond any individual descriptor matching accuracy by 5-10% in our experiments on the PL data.

The developed face matching cross-platform tools are compiled into a portable FaceMatch library and are being integrated into the PL system through the Web services. We have annotated 4000 images in the PL dataset with face/profile locations as well as the facial features for large enough faces. This annotated dataset can be made available via the National Library of Medicine. Our team is actively engaged in research and development that lead to

- robust facial feature location,
- dense/coarse feature point based face/object matching,
- user-friendly web-based front-end for flexible image query/browsing.

More tests are needed to improve the visual face/profile matching performance, and more annotated data-sets are also

required to serve this goal. We are currently looking at efficient means of image annotation, which may include development of more convenient visual annotation tools, and use of crowdsourcing for developing more comprehensive testing and evaluations data sets.

ACKNOWLEDGMENT

The authors would like to acknowledge the valuable contributions of their colleagues: Chase Bonifant, Michael Bulgakov, Sonya Shooshan, Frank Flannery, Krittach Phichaphop, Ajay Kanduru, and Michael Gill. This research is supported by the Intramural Research Program of the National Library of Medicine and the National Institutes of Health.

REFERENCES

[1] G. Thoma, S. Antani, M. Gill, G. Pearson, and L. Neve, "People locator: A system for family reunification," *IT Professional*, vol. 14, pp. 13–21, 2012.

[2] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH'95. New York, NY, USA: ACM, 1995, pp. 277–286. [Online]. Available: http://doi.acm.org/10.1145/218380.218454

[3] W. H. Buchsbaum, *Color TV Servicing*. Englewood Cliffs, NJ: Prentice Hall, 1975.

[4] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *International Conference on Image Processing*, 1995, pp. 444–447.

[5] R. Manduchi, P. Perona, and D. Shy, "Efficient deformable filter banks," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1168 –1173, apr 1998.

[6] M. Unser, N. Chenouard, and D. Van De Ville, "Steerable pyramids and tight wavelet frames in $L_2(R^d)$," *Trans. Img. Proc.*, vol. 20, no. 10, pp. 2705–2721, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1109/TIP.2011.2138147

[7] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," in *British Machine Vision Conference*, 2008.

[8] P. Ho, *Rotation Invariant Real-time Face Detection and Recognition System*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2001.

[9] R. Lienhart, E. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," in *In DAGM 25th Pattern Recognition Symposium*, 2003, pp. 297–304.

[10] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

[11] D. Zhang, S. Z. Li, and D. Gatica-Perez, "Real-time face detection using boosting in hierarchical feature spaces," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02*, ser. ICPR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 411–414. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2004.736

[12] U. Hoffmann, J. Naruniec, A. Yazdani, and T. Ebrahimi, "Face detection using discrete gabor jets and color information." in *SIGMAP*, P. A. A. Assuno and S. M. M. de Faria, Eds. INSTICC Press, 2008, pp. 76–83.

[13] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Microsoft, Tech. Rep., 2010.

[14] L. Wan and P.-c. Chen, "Face detection method based on skin color and adaboost algorithm," in *Proceedings of the 2010 International Conference on Computational and Information Sciences*, ser. ICCIS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 144–147. [Online]. Available: http://dx.doi.org/10.1109/ICCIS.2010.41

[15] P. Deng and M. Pei, "Multi-view face detection based on adaboost and skin color," in *Intelligent Networks and Intelligent Systems, 2008. ICINIS '08. First International Conference on*, November 2008, pp. 457 –460.

[16] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, 2001, pp. 511–518.

[17] N. Chhaya and T. Oates, "Robust face detection in patient triage images," in *International Conference on Computer Vision Theory and Application (VISAPP,VISIGRPP)*, March 2011.

[18] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, 2012.

[19] H. Kruppa, M. A. Bauer, and B. Schiele, "Skin patch detection in real-world images," in *In Annual Symposium for Pattern Recognition of the DAGM, Springer LNCS 2449*, 2002, pp. 109–117.

[20] R. Khan, A. Hanbury, J. Stöttinger, and A. Bais, "Color based skin classification," *Pattern Recognition Letters*, vol. 33, no. 2, pp. 157–163, 2012.

[21] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *GRAPHICON*, 2003, pp. 85–92.

[22] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition*, vol. 40, no. 3, pp. 1106–1122, 2007.

[23] J. M. Chaves-González, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, "Detecting skin in face recognition systems: A colour spaces study," *Digit. Signal Process.*, vol. 20, no. 3, pp. 806–823, may 2010.

[24] D. Malacara, *Color vision and colorimetry: theory and application*. SPIE Press, 2002.

[25] P. Kuchi, P. Gabbur, P. S. Bhat, S. D. S., and S. Smieee, "Human face detection and tracking using skin color modeling and connected component operators," *IETE Journal of Research, Special issue on Visual Media Processing*, vol. 38, pp. 289–293, 2002.

[26] G. Kukharev and A. Nowosielski, "Visitor identification - elaborating real time face recognition system," in *WSCG (Short Papers)*, 2004, pp. 157–164.

[27] M. Luo, D. Dementhon, X. Yu, and D. Doermann, "SoftCBIR: Object searching in videos combining keypoint matching and graduated assignment," 2006.

[28] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[29] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *IEEE International Conference on Computer Vision (ICCV)*, September 2009. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/Ossk

[30] D. S. Bolme, J. R. Beveridge, and B. A. Draper, "Facel: Facile face labeling," in *ICVS*, 2009, pp. 21–32.

[31] H. Zhou and A. H. Sadka, "Combining perceptual features with diffusion distance for face recognition," *Trans. Sys. Man Cyber Part C*, vol. 41, no. 5, pp. 577–588, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2010.2051328

[32] v. Raudys and F. Roli, "The behavior knowledge space fusion method: analysis of generalization error and strategies for performance improvement," in *Proceedings of the 4th international conference on Multiple classifier systems*, ser. MCS'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 55–64. [Online]. Available: http://dl.acm.org/citation.cfm?id=1764295.1764304

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, nov. 2011, pp. 2564 –2571.

[34] M. van Erp and L. Schomaker, "Variants of the borda count method for combining ranked classifier hypotheses," in *7th International Workshop on Frontiers in Handwriting Recognition*, 2000, pp. 443–452.

[35] Apple, Inc, "Face Detector Library," 2012. [Online]. Available: https://developer.apple.com/library

[36] Luxand, Inc, "Face SDK," 2012. [Online]. Available: http://www.luxand.com/facesdk/