Cao C, Ai HZ. Facial similarity learning with humans in the loop. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(3): 499–510 May 2015. DOI 10.1007/s11390-015-1540-3

Facial Similarity Learning with Humans in the Loop

Chong Cao (曹 翀), *Student Member, CCF*, and Hai-Zhou Ai (艾海舟), *Senior Member, IEEE, Member, CCF*

Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China

E-mail: caoc10@mails.tsinghua.edu.cn; ahz@mail.tsinghua.edu.cn

Received November 28, 2014; revised March 7, 2015.

Abstract Similarity learning has always been a popular topic in computer vision research. Among this, facial similarity is especially important and difficult due to its wide applications and the nonrigid nature of human faces. The large gap between feature representations and human perceptual descriptions makes the problem even harder. In this paper, we learn facial similarity through human-computer interactions. To learn perceptual similarities of faces in a gallery set, we ask users to label some candidate images with their similarities to a probe image. Based on users' responses, a sampling algorithm actively generates a probe image and a set of candidates for the next query. Assisted with human efforts, the algorithm embeds all the images into a space where the distance between two subjects conforms to their dissimilarity in human perception. We apply the learned embedding to face retrieval and compare our method with some feature-based methods on a dataset we collect from social network sites (SNS). Experimental results demonstrate that incorporating human efforts can ensure retrieval accuracy. At the same time, the active sampling algorithm reduces human efforts.

Keywords face retrieval, facial similarity, active learning

1 Introduction

User-based face retrieval has become a popular topic in the field of computer vision in recent years. Especially in criminal investigations, the police wants to find the suspect in a large database according to the witness's description. Asking the witness to look at each image in the database is time consuming and thus unrealistic. In some retrieval systems[①], the witness is asked to describe some features or attributes of the suspect (e.g., male, red hair) and accordingly the suspect is found. However, people may not agree on the same definition of an attribute. Short hair in one person's eyes might be regarded as long hair by another person. Furthermore, sometimes people come out of words when describing some features. For example, human eyes have lots of variations in shape and color, and it is hard to classify them into a few semantic categories. In some cases, the police draws a sketch according to the witness's description and asks the witness to help revise the sketch so that the sketch depicts the suspect more accurately. After the witness confirms that the sketch well describes the suspect, the police uses the sketch to retrieve the suspect[1]. This is to some extent more reliable than the former method since with the process of revision, the sketch gives more accurate descriptions of the suspect than semantic descriptions. However, sketches are quite different with photos in terms of color and texture, and sometimes do not contain enough information for face retrieval. Besides, drawing a sketch takes much time and needs special techniques, which does not fit for frequent use.

In computer vision, a typical face retrieval system requires an actual probe image. The system defines a set of features, and learns a similarity metric or a binary classifier from massive training data. A main problem with this scheme is the gap between pre-defined features and the high-level human descriptions. Although Kumar *et al.* defined some high-level facial attribute classifiers[2], these attributes still cannot compare to abundant human descriptions. Furthermore,

Regular Paper
Special Section on Computational Visual Media
This work was partly supported by the National Basic Research 973 Program of China under Grant No. 2011CB302203.

[①] M. Inc. Mughunt online face search engine. http://mughunt.securics.com, Dec. 2014.

under some circumstances (e.g., criminal investigations mentioned above), there is no real image of the suspect and we cannot directly extract image features. How to find a subject similar to the probe implicitly from a few interactions with the user is the main concern of this paper.

To solve the problems raised above, we propose an active learning method to embed subjects in a large database into a face map based on annotated similarities. At the beginning, we ask users to label relative similarities among sets of subjects and build a face space containing these subjects that conforms to human perceptual similarity. Without any pre-defined image features, we can locate a subject by collecting relative similarities between the subject and some computer selected candidates in the database. User responses can guide sampling candidates in afterward queries. Since we do not extract any visual features, the image can be a mental image that only exists in the user's mind. In that way, we combine human annotated similarity and active selection algorithm together. Fig.1 gives an overview of the retrieval system.

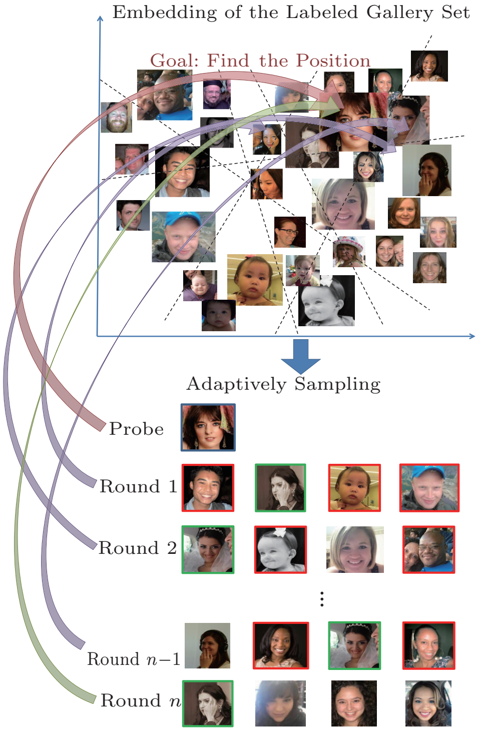


Fig.1. Overview of the retrieval system. First we embed all the subjects in an offline gallery set into a face map based on user annotations. When a user queries an image, we ask the user to iteratively label similarities between the query image and candidate sets. During each iteration, the sampling algorithm chooses a candidate set for the next iteration based on existing user annotations to accelerate the retrieval process. The algorithm ends when the user hits a satisfying subject in the gallery.

The contributions of this paper are: 1) We expand the traditional embedding learning algorithm based on triplet constraints to setwise active learning so that it generates triplet constraints from image sets more efficiently. 2) We combine user annotations and computer generated queries together to obtain high accuracy in retrieval and shorten the time consumed. 3) We collect a new SNS database and apply our methods to the problem of face retrieval on the new dataset.

The rest of the paper is organized as follows. In Section 2, we discuss relevant work in face retrieval, learning the embedding and human-in-the-loop active learning. In Section 3, we describe details of building a face map for the gallery set and setwise active selection algorithm for incremental learning. In Section 4, we show how to retrieve a face with humans in the loop. Finally in Section 5, we demonstrate our experiments and compare our approach with existing methods.

2 Related Work

The common way to do image retrieval, especially face retrieval, is to define a set of features, and use distance norms (e.g., Euclidean norm) to evaluate how similar a pair of images are[3-4]. However, the Euclidean distances between low-level feature vectors used in these studies do not always conform to human perception of dissimilarity. Besides, most of these methods are very sensitive to environmental and other variations (e.g., illumination, occlusion and facial expression). After that, more and more studies focus on face variations. Wiskott *et al.* described a face with an image graph of fiducial points using sets of wavelet components[5], which is locally invariant only to a set of known transformations. Berg and Belhumeur used an identity-preserving alignment to normalize a face into fixed size and position so that low-level features can work on the aligned images[6]. Chopra *et al.* learned a similarity metric that maps input images into a target space where a simple L2 norm conforms to the semantic similarity among images[7]. Different from these studies, some methods[8-9] compare images under similar conditions with the help of an additional image library.

In some cases such as criminal investigations, we do not have an image of the suspect and cannot extract features for retrieval. What we have is only a mental image in the mind of the witness. Thus all the above mentioned feature-based methods do not fit any more. A simple solution is to train a set of attributes and simile classifiers that semantically describes a face image with binary attributes such as male, white, long-hair and so on[2]. Here attributes are high-level descriptions of an image invariant to environmental and pose changes. Some methods build multi-attribute spaces via data fusion and apply attribute classifiers to image ranking and retrieval[10-11]. On the one hand, these attributes do not have a generalized definition adopted by all users, and hence bring inconsistency during labeling and retrieval process. On the other hand, they only learn a limited number of simile classifiers. For example, they define an Angelina-Jolie-like lip classifier, but in reality, there are far more types of lips and such facial attributes are hard to list by words. To solve this problem, some studies learn relative attributes which are more consistent among users than traditional binary attributes[12-13]. A later work extends this work by actively learning to rank a set of images using set-wise margin criterion[14]. These studies to some extent solve the problem of user inconsistency, but are still limited by semantic definitions of attributes. To solve these problems, people try to directly learn from human-provided similarities between images rather than to define a domain-specific attribute space. Some studies model up the relative similarity information collected from crowd-sourced data and learn an embedding from human-labeled triplets[15-17]. These methods break up the limitation of pre-defined features and implicitly learn from human annotations. Garces *et al.*[18] applied the triplet constraints learning method[17] and learned an embedding for clip arts, but during each query, they merely randomly sampled a triple from the gallery set. Therefore the system relies on heavy annotations and is hard to promote to a large scale. A recent work introduces a bubble game that asks the user to reveal as few bubbles to recognize a blurred image as possible[19]. In that way, a bubble bank is constructed, which can help computers select the most discriminative features human uses for recognition. Holub *et al.*[20] asked users to assign a distance to close and far images respectively and built a low-dimensional space for human faces using multi-dimensional scaling (MDS)[21]. Then they learned a functional mapping from visual feature vectors to the face space, which combines visual features and human labeled similarities together. However, the absolute distance score is not consistent among different users, which brings noises in user responses. A similar work proposed a relevance feedback system for face retrieval[22]. At each iteration, the user declares which of the several displayed faces is "closest" to the target in his/her mind. A Bayesian, information-theoretic method models user responses and chooses which images to display next. Some studies train self-organizing maps (SOMs) to organize the database[23-24] in which the semantic classes are densely spread in separate areas. The SOMs classify images into accurate sub-classes by iteratively asking users to pick up similar subjects in the candidate set. The main idea of our algorithm is quite similar to these two studies, but we use sampling probability instead of semantic classes to choose candidates. Our method can recover even if the user gives an ambiguous response, and thus is more consistent to noises.

3 Learning a Perceptual Embedding

In this section, we introduce our approach for building a perceptual face map for a gallery database. During each iteration, the algorithm samples candidates and poses queries about their relative similarities. Each query contains a query image and a set of candidate images. Users are asked to annotate each candidate whether it is similar or dissimilar to the query image. The output of the algorithm is an embedding of all the images in the gallery where the Euclidean distances between image pairs in the new space accord with human-labeled dissimilarities. The algorithm starts with some randomly sampled query images and builds a basic embedding of the gallery. By calculating the confidence of each subject in the embedding, our algorithm automatically finds out these unsatisfying queries and re-sample candidates to collect more annotations. The pipeline of our approach is shown in Fig.2.

3.1 Stochastic Triplet Embedding

Given a set of subjects $Z = \{z_1, ..., z_N\}$, we want to learn an embedding $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\} \subset \mathbb{R}^r$, where the Euclidean distance between a pair of subjects (i, j) accords with human perception of dissimilarities d_{ij}, i.e.,

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 < \|\boldsymbol{x}_i - \boldsymbol{x}_l\|_2 \Longleftrightarrow d_{ij} < d_{il},$$

$$\begin{cases} d_{ij} = 0, & \text{if } i = j, \\ d_{ij} > 0, & \text{if } i \neq j. \end{cases}$$

In reality, such absolute dissimilarities are hard to collect and inconsistent among users[20,25]. Instead, we use relative dissimilarities in the form of triplets. A triplet set is defined as $\mathcal{T} = \{(i, j, l) | z_i$ is more similar to z_j than $z_l\}$, where i is a query subject and (j, l) is a candidate pair. After sampling query subjects and unordered candidate pairs from the gallery set, we ask

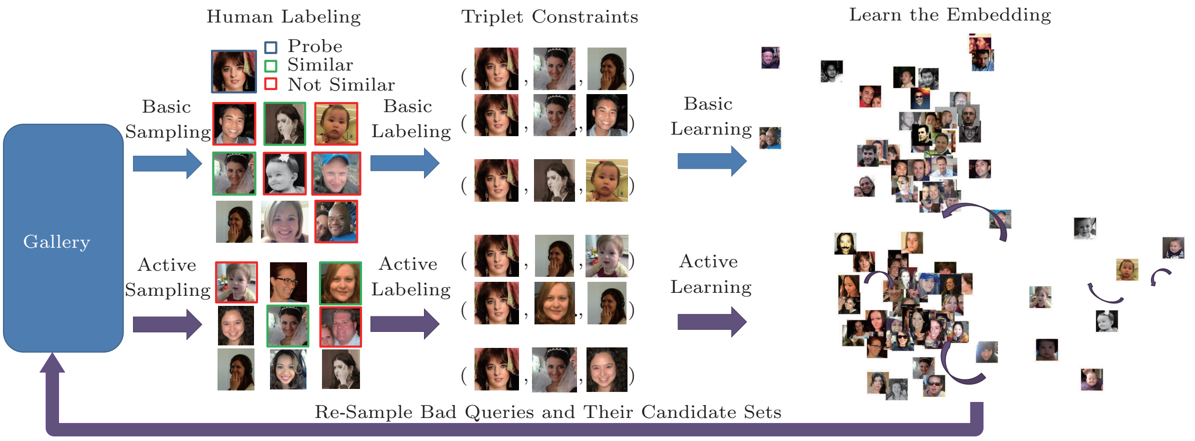


Fig.2. Pipeline for building the gallery face map. At the beginning, we sample some query images and candidates for labeling. With the triplet constraints generated from user responses, we can learn a perceptual embedding of the whole gallery set. The setwise criterion detects "unsatisfactory" queries and re-samples new candidate sets with less redundant and ambiguous candidate pairs.

users to order each candidate pair according to their similarity to the query subject and learn an embedding $\{\widetilde{\boldsymbol{x}}_i\}$ using the t-STE[17] algorithm. Define probability p_{ijl} as how well a triplet $(i,j,l) \in \mathcal{T}$ is modeled:

$$\begin{aligned} p_{ijl} &= \frac{\delta(i,j)}{\delta(i,j)+\delta(i,l)}, \\ \delta(i,j) &= \left(1+\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}. \end{aligned} \tag{1}$$

The goal of t-STE algorithm is to maximize the sum of the log-probabilities over all triplets:

$$\max_X \sum_{\forall(i,j,l)\in\mathcal{T}} \log p_{ijl}. \tag{2}$$

There are other definitions of $\delta(i,j)$. For example, crowd kernel learner (CKL) uses kernel function $\delta(i,j) = k_{ii} + k_{jj} - 2k_{ij}$ and stochastic triplet embedding (STE) uses exponential function $\delta(i,j) = exp^{-(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)}$. Here we use the t-STE rather than other functions because it decays to zero when a triplet constraint is very strongly violated, and thus handles noises in $\mathcal{T}$ by not trying to satisfy constraints that contradict the consensus. Projected gradient descent is used to solve the optimization problem. More details are introduced in the related work[17].

3.2 Setwise Criterion for Active Learning

Collecting triplets is a huge task. For N subjects, we can generate $N \times \binom{N-1}{2}$ triplets for user annotations. For example, there are 500 million triplets in 1 000 images. Apparently, it is hard and unnecessary to label them all as some will be redundant or ambiguous. But if we label too few triplets for each subject, they might be insufficient to learn a good embedding. Instead of an exhaustive enumeration, we generate triplets in a more efficient way.

We query users in the form of image sets $\{(q^t, C^t)|t = 1,\ldots,n\}$. Here n is the number of queries posed. $q^t \in Z$ denotes a query image and $C^t = \{c_k^t|k = 1,\ldots,K, c_k^t \in Z\}$ is a candidate set. In a query, the user is asked to select candidates that are most similar to the probe image and candidates that are definitely not similar to the probe. We do not limit the number of similar and dissimilar images a user picks up. Denoting user responses $U^t = \{u_k^t|i = 1,\ldots,K\}$. u_k^t is set to 1 if a subject c_k^t is labeled as "similar" to the probe image q^t, -1 if "dissimilar" and 0 if "neither similar nor dissimilar". In this way, we can generate tens of triplets from a query with ten candidates, denoting the sets of triplets generated in the i-th query $\mathcal{T}^i = \{(q^i, c_j^i, c_l^i) \mid u_j^i > u_l^i\}$. The set-wise triplet generating method is more efficient than the one-at-a-time triplet generating methods[15-17].

After running the t-STE algorithm using the set of triplets $\mathcal{T}$ generated from set-wise queries, we learn an estimated embedding $\{\widetilde{\boldsymbol{x}}_i\}$. However, with limited annotation, this result is not good enough. The small candidate set for each query directly results in that some of the query subjects are dissimilar to all of the images in their corresponding candidate sets. Therefore, we need to find these "bad" queries and re-sample some candidates for them. Intuitively, the objective function defined in (1) favors large margin triplets, i.e., one of the candidates is clearly more similar to the probe than the other candidates. Still, we use p_{ijl} defined in (1) to represent how well the triplet (i, j, l) is modeled. For an ideal candidate set, there should be some candidates that are very similar to the query while some are relatively dissimilar. We define the confidence of a query (q^t, C^t) as:

$$conf(q^t, C^t) = \prod_{u_j^t \neq u_l^t} \frac{\delta(q^t, c_j^t)}{\delta(q^t, c_j^t) + \delta(q^t, c_l^t)}, \quad j, l \in \{1, \ldots, K\}. \tag{3}$$

Here we use the estimated embedding of the subjects $\widetilde{\boldsymbol{x}}_i$ to calculate δ. The bigger the confidence is, the better the query is. For those queries with small confidence, we need to re-sample a candidate set to improve the embedding we constructed.

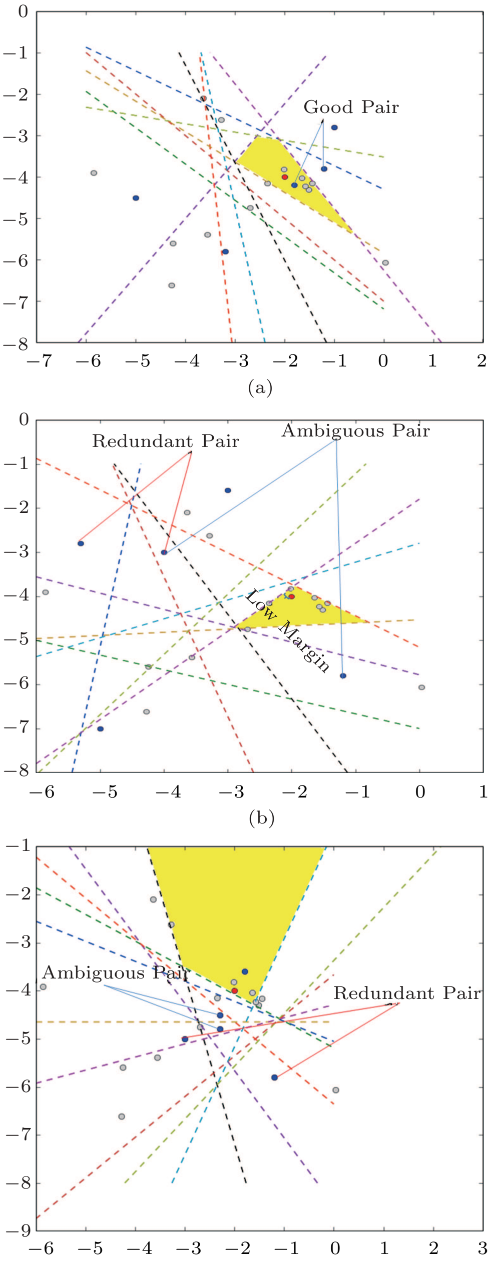
A main cause for a "bad" query is that some of the triplet constraints we generated are ambiguous or redundant. For example, if both the subjects in a candidate pair are very dissimilar to the probe, it is very likely that the user is hard to decide their relative similarity. Thus the response provided is ambiguous. Fig.3 shows some examples of good, ambiguous, and redundant candidate pairs. We desire a candidate set that can produce informative and consistent triplet constraints. And intuitively, the learning method favors candidate pairs with one near to the query image and the other relatively far. Similar to the objective function defined in (1), we naturally select the best candidate set C^* for query subject z_i as:

$$C^* = \arg\max_{C \subset Z} \sum_{j,l \in C} \frac{\delta(i, j)}{\delta(i, j) + \delta(i, l)}. \tag{4}$$

The search space of optimizing (4) is $O(N^2)$ where N is the number of images in the gallery set. For efficiency, we simply sample some possible candidate sets and choose the one with the highest score.

4 Interactive Mental Image Retrieval

Mental image retrieval does not use any visual representation of the probe image during retrieval. As mentioned in Section 1, due to the limitation of pre-defined visual features, simple Euclidean distance may not accord with the dissimilarity between a pair of images. In Section 3, we learn a face map from user labeled relative similarities, where the Euclidean distance between a pair of images conforms to their perceptual dissimilarity. In this section, we focus on face retrieval with humans in the loop with the help of the face map.



(a)

(b)

(c)

Fig.3. Examples of good and bad candidates. The red dot is the query subject, and the blue dots are candidates. The dash lines are constraints of a candidate pair. The yellow region is the possible area of the query image determined by all constraints. In (a), the blue lines mark a good pair. One is near to the query image and the other is relatively far. Constraints provided by the pair cut off most of the space and restrict the possible embedding of the query image to a small area. In (b), both of the two subjects of the redundant pair are very far from the query image and the constraint they provide is almost useless. The two subjects in the ambiguous pair are also very far from the query image. Although their constraint seems to play an important role in restricting the possible embeddings of the query image, it is very likely that the user gives a contradictory response. The condition in (c) is similar to that in (b), and most of the candidates are quite far from the query image resulting in an open infinite area for the possible embedding of the query image.

Given a query image q, our goal is to locate it or find the most similar image in a gallery dataset $Z = \{z_1, ..., z_N\}$. Here for $i = 1, ..., N$, each image z_i is embedded in a face map at $\boldsymbol{x}_i$. The probe image can be any kind of image (e.g., a sketch or a mental image that exists in the user's mind). We denote π as the distribution of images in the gallery being the target image. At step 0, we know nothing about q, and it can be located anywhere in the face map. Every image in the gallery has equal chance to be the target image, i.e., $p_i^0 = \frac{1}{N}$, $i = 1, ..., N$, and N is the number of images in the gallery.

During each iteration, we sample a candidate set $C^t = \{c_k^t | k = 1, ..., K\}$ according to the probability distribution π^{t-1}, where K is the number of subjects in a candidate set. We ask the user to label relative similarities between the query image and the candidate set with "similar", "dissimilar" and "neither similar nor dissimilar". Same as in Section 3, we denote the user responses as $\{u_k^t | k = 1, ..., K\}$. Hence to embed the query q into the learned face map, we maximize the objective function *w.r.t.* $\boldsymbol{x}_q^t$, which is the estimated embedding of the query image during iteration t :

$$\max_{\boldsymbol{x}_q^t} \sum_{\forall (z_i, z_j) \in C^t} \log \frac{\delta(q^t, i)}{\delta(q^t, i) + \delta(q^t, j)}. \tag{5}$$

To sample candidates for the next round annotation, we update the probability with the criterion:

$$p^{t+1}(z_i | \boldsymbol{x}_q^t) = \prod_{\forall j} \frac{\delta(q^t, i)}{\delta(q^t, i) + \delta(q^t, j)},$$

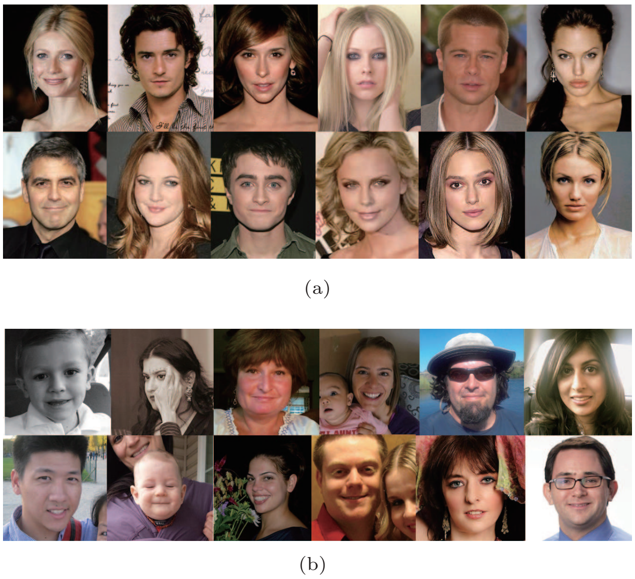
where the position of the query image q at round t is estimated through a maximization defined in (5).

With this iterative sample-label-estimate process, we adaptively learn the probability distribution of each image in the gallery set to be sampled in the next query and automatically sample candidate sets that are similar to query image according to the distribution. The advantage of this sampling process over simply sampling around the "similar" subjects is that even if there are some deficiencies of the face map we learned before, or the user gives back a noisy response, the algorithm is still likely to get back on track after a few iterations.

5 Experiment

5.1 Dataset and Data Collection

We need a dataset that contains a massive number of faces that cover subjects of different genders, races, ages and so on for our experiments. Although there are a few existing datasets for face recognition, none of them meet our demands perfectly. For example, the Public Figures (PubFig) dataset[2] contains only 200 people, and most of them are celebrities aging from 20 to 50. Although the most widely used Labeled Faces in the Wild (LFW) dataset[26] contains 5 749 people, it also suffers severe bias on distributions in age and race. To conduct our experiments, we collect a large number of portrait images from an SNS where each image represents a unique subject (regardless of that some users use celebrity photos or popular images on the Internet as their portraits) including babies and seniors. We run a face detector[27], cut out faces in the dataset, and filter out those images that do not have any faces in them. Limited by annotation cost, we use only $N = 500$ randomly selected subjects in the following experiments. We call the dataset 500-SNS. As shown in Fig.4, faces in our 500-SNS dataset shown in Fig.4(b) vary in age, gender and race, etc., while the PubFig dataset shown in Fig.4(a) biases heavily on these aspects.



(a)

(b)

Fig.4. Examples in (a) PubFig[2] and (b) 500-SNS dataset respectively.

5.2 Learning the Embedding

5.2.1 *Absolute Score vs Relative Comparisons*

First we perform experiments on a small dataset with 20 subjects to verify the advantage of relative comparisons over absolute scores. We randomly select 20 images from the PubFig dataset[2] and ask a volunteer to label each image with its similarity to the other images with one of the labels “very similar”, “definitely not similar”, “neither similar nor dissimilar”. We ask the user to do this twice to measure the user’s ability to replicate his or her original annotation. Denote the user’s response is $U = \{u_{k,t} | k = 1, \ldots, K, t = 1, \ldots, N\}$, where N is the number of queries we posed and K is the number of candidates in each query. In this experiment, $N = 20$ and $K = 19$. For simplicity, we set $u_{k,t}$ as 1 if it is labeled as “similar” to the query, -1 as “dissimilar” and 0 as “neither similar nor dissimilar”. The absolute and the relative difference between the two times of annotation, U^1, U^2, are defined as:

$$\begin{aligned}
&Ascore(U^1, U^2) = \frac{\sum\limits_{k,t} I(u^1_{k,t} \neq u^2_{k,t})}{K \times N},\\
&Rscore(U^1, U^2) = \frac{\sum\limits_{u^1_{i,t} < u^1_{j,t}} I(u^2_{i,t} > u^2_{j,t})}{\sum\limits_{u^1_{i,t} < u^1_{j,t}} I(u^2_{i,t} \neq u^2_{j,t})},\\
&i, j, k = 1, \ldots, K, t = 1, \ldots, N.
\end{aligned}$$

The *Ascore* of the user is 0.1895 and the *Rscore* is 0.0098. We can see that although the user might change his/her absolute score for similarity during the two times of annotation, the relative similarity is consistent. We also ask another user to label the same set of images and calculate the difference between the two users’ annotation. The *Ascore* is 0.3111 and the *Rscore* is 0.0945. Although different users are supposed to have different judgments for whether two subjects are similar, the consistency of relative similarity is acceptable. In the following experiment, we assume all users behave the same in annotations and ignore the difference among users.

5.2.2 Offline Learning the Embedding

In this experiment, we test the offline learning accuracy using t-STE with different parameters. As introduced in Section 3, the objective of learning is to maximize the log probability defined in (2). Parameter α is set to be the number of degrees of freedom of the student-t distribution. Here we use $\alpha = r - 1$, where r is the desired dimension of the embedding.

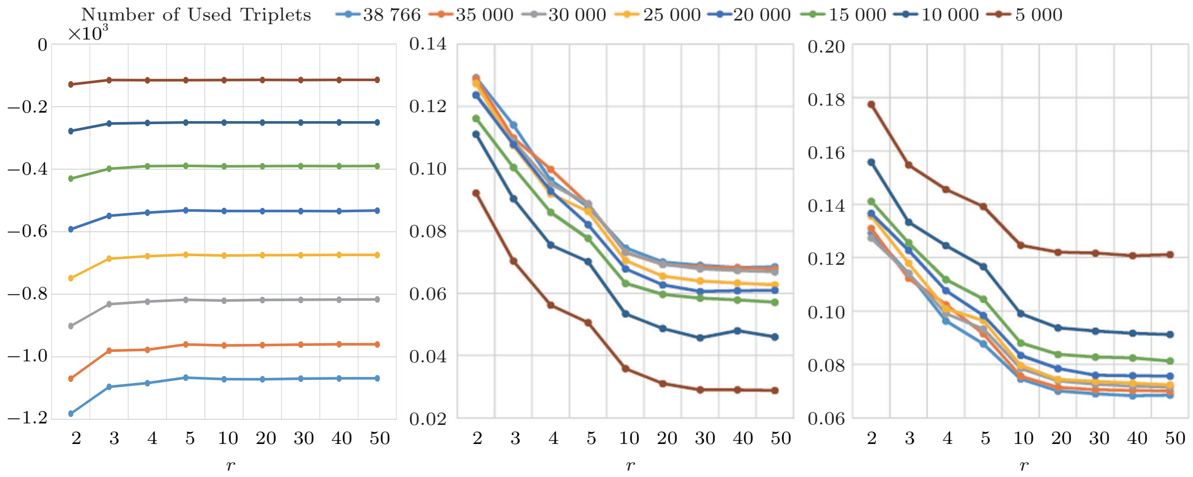
The gallery set we use is the 500-SNS dataset introduced in Subsection 5.1. In order to generate triplets, we pose a query for each subject in the gallery. Each query contains a unique query image and a set of 24 subjects in the gallery, namely $K = 24$. The number 24 is simply decided by the number of images that can be clearly shown on the screen at the same time. Since it is hard for users to distinguish the similarity of every candidate pair to the probe image, we simply ask the user to pick up these “most similar” and “definitely not similar” images. We call the rest of the candidate images “borderline” images. To avoid personal bias, we assign the annotations to 10 volunteers not affiliated with this project. Using user annotations, we generate 38766 triplets from the 500 queries we posed.

We test on different r and the number of triplets used in learning the embedding. The result is evaluated in three aspects: 1) the maximum log probability as defined in (2); 2) we generate triplets according to the learned embedding, compare that with triplets used in building the embedding, and calculate the ratio of violated triplets (RVT1); 3) we generate triplets according to the learned embedding, compare that with all the triplets we collected from users, and calculate the ratio of violated triplets (RVT2). We desire small RVT1 and RVT2. Especially RVT2 is considered as an important criterion to evaluate the embedding. The results are shown in Fig.5.

Generally speaking, the parameter r does not have a great influence on the log probability achieved in learning. But the log probability increases with the number of triplets used. Both RVT1 and RVT2 decrease as the dimension of the learned embedding increases, which accords with the intuition that a higher dimensional space can better embed these subjects. Although RVT1 increases as we use more triplets, RVT2 decreases. Namely, the more triplets we use, the less the learned embedding is likely to satisfy these triplets, and the nearer the learned embedding is to human perception.

5.2.3 Active Learning the Embedding

In this subsection, we illustrate the advantage of the proposed active sampling method on constructing the embedding. The experimental set-up is similar to that of offline learning. But in this experiment, we split the candidate set in each query into two subsets, each containing 12 subjects. The first subset is used to build a basic embedding, i.e., $\{\widetilde{\boldsymbol{x}}_i\}$ mentioned in Subsection 3.2. After that, we test our active learning algorithm in three ways. First, we pose queries randomly selected from the second subset and build an embedding as a baseline (i.e., random query + random candidates). Second, i.e., best query + random candidates, we use the estimated embedding to calculate a confidence score defined in (3) for each query subject. We



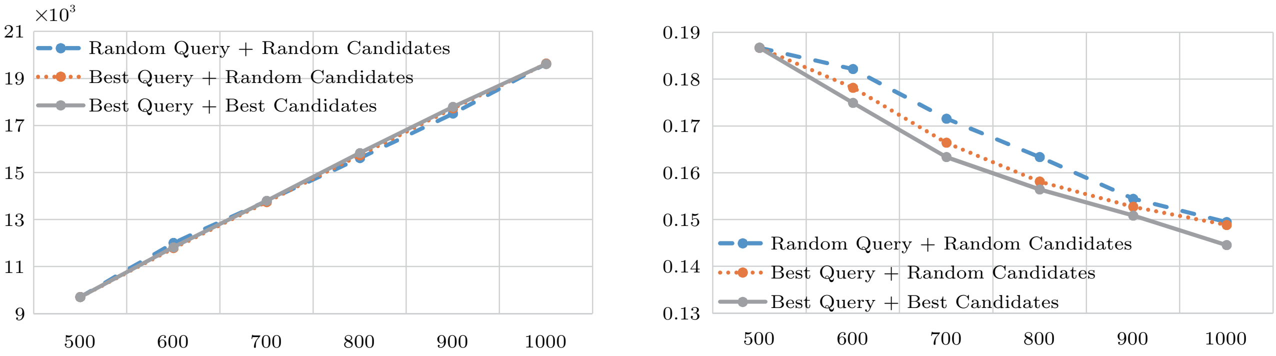
(a) (b) (c)

Fig.5. (a) Maximum log probability achieved in learning. (b) Ratio of violated triplets to all triplets used in building the embedding (RVT1). (c) Ratio of violated triplets in the learned embedding to all triplets we collected (RVT2). r ranges from 2 to 50 on the cross axis. The number of used triplets ranges from 5 000 to 38 766 marked in different line colors.

actively pose new queries with the second subset of candidates according to the order of the confidence score. The best query subject experiment validates the ability of our algorithm in finding "bad queries" in the existing embedding. In the last one, we sample a candidate set based on (4) with 12 candidate subjects and pose queries using these new candidates. All the three sampling methods sample in 24 candidate images for each face in the dataset, but the order of query images and candidate sets is different. We evaluate the improved embedding built using these three methods in terms of RVT2. Results are shown in Fig.6.

5.3 Facial Similarity Evaluation

As we claimed, the face map we build in Section 3 accords with the human perception of facial similarity. Fig.7 shows some of the nearest neighbors (NN) we get using different features. Columns 1∼7 orderly represent probe images, NN in the attribute space, NN in a color histogram space, NN in an LBP histogram space, NN in a combination of color and LBP histogram space, NN in the basic embedding we build (emb1), and NN in the actively learned embedding we build (emb2) respectively. The NNs in the two embedding spaces are the same in



(a) (b)

Fig.6. Active sampling results. We evaluate our method in terms of number of triplets generated and RVT2 using 500∼1 000 queries posed in three different ways. (a) shows the number of triplets generated with certain number of queries posed in the three different ways. With a certain number of queries, the number of triplets generated does not differ greatly. The horizontal axis starts at 500 since we use 500 queries each with 12 candidates to build a basic embedding $\{\tilde{\boldsymbol{x}}_i\}$. (b) is RVT2 with certain number of queries posed in the three different ways. Generally speaking, RVT2 decreases with the number of queries used. The queries generated using optimized query images and candidate sets proposed in Subsection 3.2 perform better than the baseline random sampling for they build an embedding which better accords with human perception.

Fig.7(a). We can see that the basic embedding does not conform to human perception especially when a face has few similar images in the gallery set (e.g., Asians and babies). Thus the random sampling in building the basic embedding might lose those similar images and lead to indistinguishable candidate sets. The improved active learning we proposed improves this case as shown in Fig.7(b). NNs in other feature spaces are similar to the probe image in some aspects (e.g., LBP counts for face pose and the color histogram can, to some extent, distinguish people of different races), but they are not so similar to the probe image on the whole as the embedding we build.

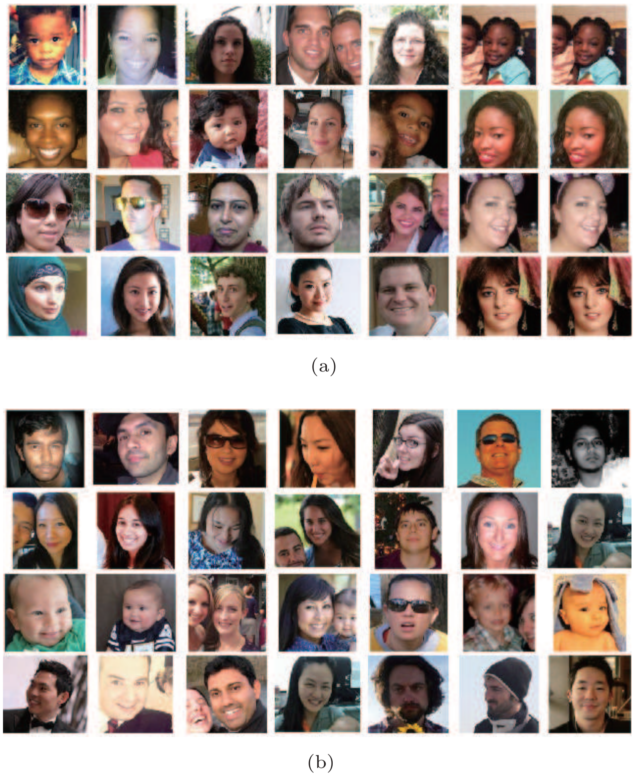


Fig.7. Columns 1∼7 represent probe image, NN in the attribute space, NN in the color histogram space, NN in the LBP histogram space, NN in a combination of color and LBP histogram space, NN in the basic embedding we build (emb1), and NN in the final embedding we build (emb2) respectively. Here the attributes we use are provided by a public software development kit (SDK)[2]. We select six attributes that we think are most important in facial similarity: age, gender, race, wearing glass, pose, and expression.

We ask two users to label the top 8 nearest neighbors generated by the five different methods (attri, color, lbp, color+lbp, emb) we mentioned above. They also use the "Similar, Borderline or Dissimilar" approach to label each set of 40 images. Fig.8 shows the CMC curve and a statistic of the number of pictures in a group of eight NNs selected as similar/dissimilar in each feature space. The meaning of different colors is shown in the legend.

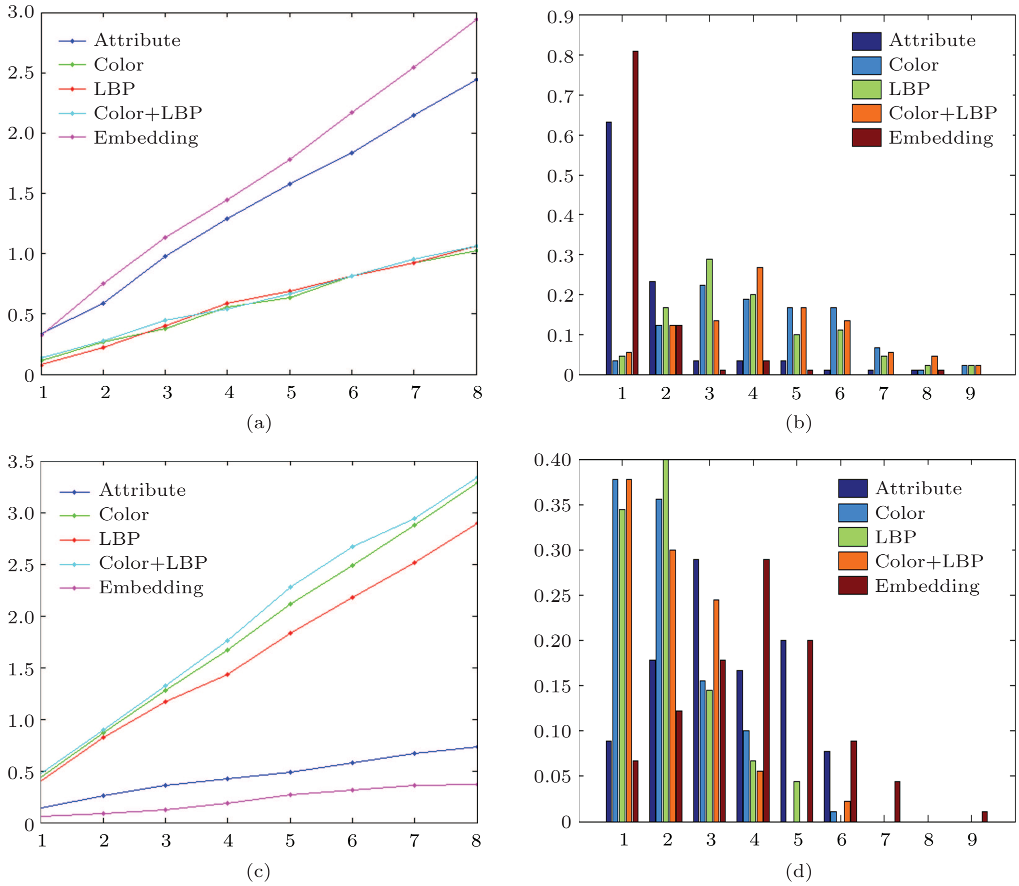
5.4 First Subject Hit in Face Retrieval

The goal of this experiment is to fast locate a facial image in the gallery set. We commit experiments on 20 images randomly selected from the dataset, and it takes an average of 5.25 rounds to find the image. Namely, users need to see 125 images on average to locate the probe image. Note that for a random retrieval method, this number is 250, which takes twice the time needed using the face map guided retrieval method. We believe that as the scale of the dataset increases, the advantage of our method can be more obvious. It takes $O(\log(n))$ iterations to locate the image compared with $O(n)$ using random sampling.

Two visualized examples of the retrieval process are shown in Figs. 9 and 10. The first rows in Figs. 9(a) and 10(a) show the probe image. In both experiments, the user hits the probe image in four iterations. The rest rows show the candidate set and user responses in each iteration. Candidates that are selected as "similar" to the probe image are with green boxes and "dissimilar" with red ones. Figs.9(b)∼9(d) and Figs.10(b)∼10(d) show the probe image and the candidates in the embedding space during each iteration. The big magenta dot is the location of the probe image. Green dots represent candidates that are labeled "similar" in current iteration and red dots "dissimilar". The rest colored in blue indicate "borderline" candidates. In Fig.9, we can see that the sampling tends to converge to the location of the probe image with human interactions. While in Fig.10, although our algorithm does not sample enough similar candidates in the first round due to the sparsity of children pictures in our dataset or results in a wrong sampling region, it can still recover from the misplacement of estimated embedding of the probe image in the second round and finally find the probe image after three rounds of annotation.

We also put four images of the other four methods (attribute, color, LBP, color+LBP) in the candidate set and see in which iteration images more similar to the probe image begin to appear (e.g., the NN in the attribute space is not labeled as similar to the probe and

[2] Face++ research toolkit. http://www.faceplusplus.com/face-research-toolkit/, Feb. 2015.



(a) (b) (c) (d)

Fig.8. Facial similarity measurement result. (a) CMC curve of similar images. (b) Statistic of the number of pictures in a group of eight NNs selected as similar in each feature space. (c) CMC curve of dissimilar images. (d) Statistic of the number of pictures in a group of eight NNs selected as dissimilar in each feature space.

some other candidates are labeled as similar. We call this the attribute NN is filtered out in this iteration). From the results shown in Table 1, we can see that most of the NNs are filtered out within two rounds, i.e., our retrieval approach can beat them in two iterations. The attribute space is better than the simple feature space such as color and LBP histograms, and nearest neighbors in the attribute space are filtered out in an average of 3.55 iterations.

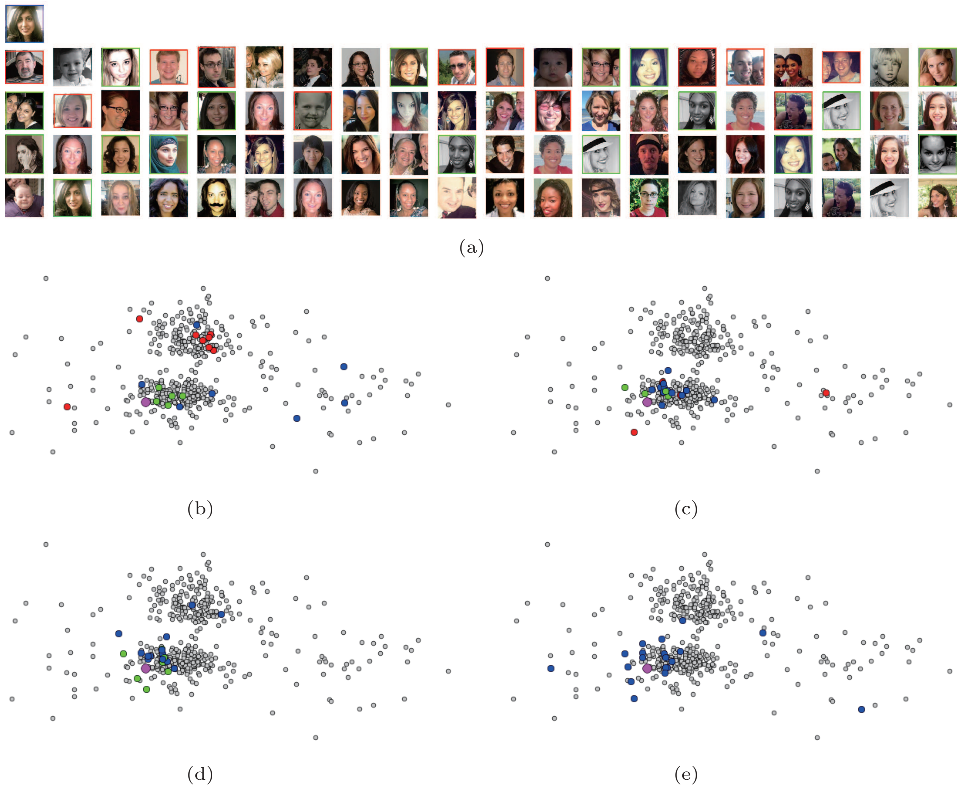
Table 1. Results for the First Subject Hit Experiment

Experiment		Number of Rounds
Filtered	Attribute	3.55
	Color	1.20
	LBP	1.25
	Color+LBP	1.45
Hit	Embedding	5.25

Note: Our methods hit the subject in an average of 5.25 rounds. The average number of iterations that nearest neighbors based on other features are filtered out is shown in the table.

6 Conclusions

In this paper, we learned an embedding of faces in a gallery from crowd-sourced set-wise relative similarities. With active sampling-labeling-learning process, the new approach learns an embedding that accords with human perception of facial similarity and needs less user interactions compared with random sampling. Based on the face embedding, we proposed a mental image retrieval framework which automatically samples candidate images according to user response history. Experimental results for building the embedding, facial similarity measurement, and face retrieval were demonstrated on an SNS dataset that contains 500 facial images. Due to annotation cost, we only experimented on a small part of the face dataset we collected. But we believe that a larger dataset can better present our idea. In the future, we are planning on reducing the computation complexity using methods such as spectral clustering[28] and carrying out experiments on larger datasets.


(a)

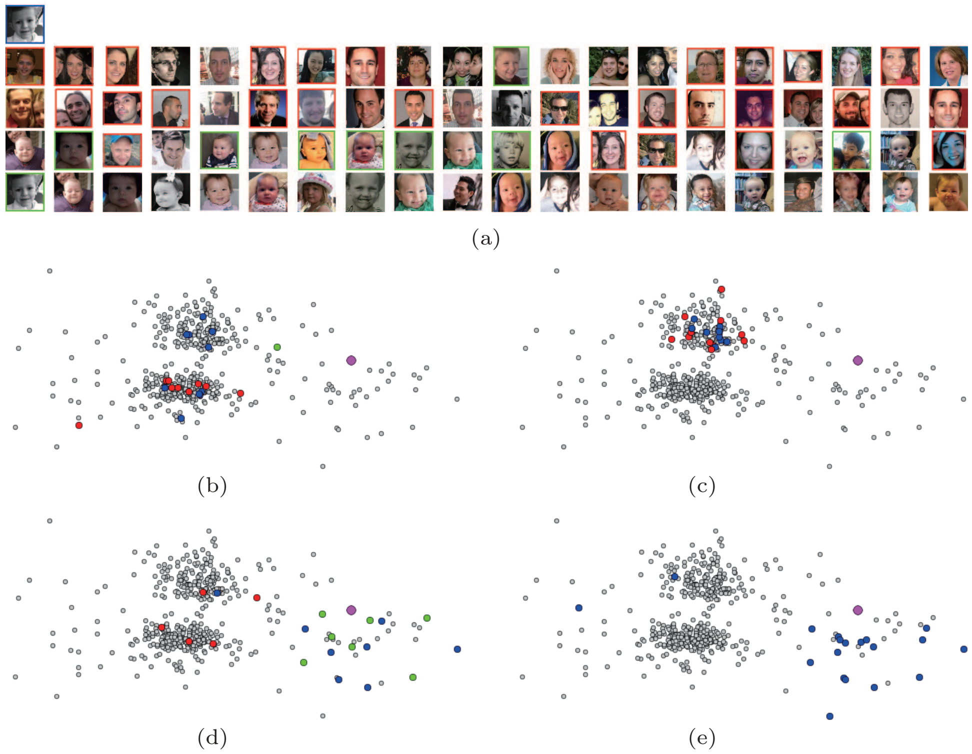
(b)

(c)

(d)

(e)

Fig.9. First subject hit result. The first row in (a) shows the probe image. In our experiment, the user hits the probe image in four iterations. Rows 2∼5 in (a) show the candidate set and the user labeling in each iteration.


(a)

(b)

(c)

(d)

(e)

Fig.10. Another first subject hit result.

References

[1] Yuen P C, Man C. Human face image searching system using sketches. *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, 2007, 37(4): 493-504.

[2] Kumar N, Berg A C, Belhumeur P N, Nayar S K. Attribute and simile classifiers for face verification. In *Proc. the 12th IEEE International Conference on Computer Vision* (*ICCV*), Sept. 29-Oct. 2, 2009, pp.365-372.

[3] Jain A K, Vailaya A. Image retrieval using color and shape. *Pattern Recognition*, 1996, 29(8): 1233-1244.

[4] Ahonen T, Hadid A, Pietikäinen M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2006, 28(12): 2037-2041.

[5] Wiskott L, Fellous J M, Krüger N *et al.* Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997, 19(7): 775-779.

[6] Berg T, Belhumeur P N. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *Proc. the British Machine Vision Conference*, September 2012.

[7] Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification. In *Proc. the IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), June 2005, pp.539-546.

[8] Schroff F, Treibitz T, Kriegman D, Belongie S. Pose, illumination and expression invariant pairwise face-similarity measure via doppelgänger list comparison. In *Proc. the 13th IEEE ICCV*, Nov. 2011, pp.2494-2501.

[9] Yin Q, Tang X, Sun J. An associate-predict model for face recognition. In *Proc. the IEEE CVPR*, June 2011, pp.497-504.

[10] Scheirer W J, Kumar N, Belhumeur P N, Boult T E. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *Proc. the IEEE CVPR*, June 2012, pp.2933-2940.

[11] Siddiquie B, Feris R S, Davis L S. Image ranking and retrieval based on multi-attribute queries. In *Proc. the IEEE CVPR*, June 2011, pp.801-808.

[12] Parikh D, Grauman K. Relative attributes. In *Proc. the 13th IEEE Int. Conf. Computer Vision*, Nov. 2011, pp.503-510.

[13] Biswas A, Parikh D. Simultaneous active learning of classifiers & attributes via relative feedback. In *Proc. the IEEE CVPR*, June 2013, pp.644-651.

[14] Liang L, Grauman K. Beyond comparing image pairs: Setwise active learning for relative attributes. In *Proc. the IEEE CVPR*, June 2014, pp.208-215.

[15] Agarwal S, Wills J, Cayton L *et al.* Generalized non-metric multidimensional scaling. In *Proc. the 11th Int. Conf. Artificial Intelligence and Statistics*, March 2007, pp.11-18.

[16] Tamuz O, Liu C, Belongie S, Shamir O, Kalai A T. Adaptively learning the crowd kernel. *arXiv:1105.1033*, 2011. http://arxiv.org/abs/1105.1033, March 2015.

[17] van der Maaten L, Weinberger K. Stochastic triplet embedding. In *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, September 2012.

[18] Garces E, Agarwala A, Gutierrez D, Hertzmann A. A similarity measure for illustration style. *ACM Transactions on Graphics*, 2014, 33(4): 93:1-93:9.

[19] Deng J, Krause J, Li F F. Fine-grained crowdsourcing for fine-grained recognition. In *Proc. IEEE CVPR*, June 2013, pp.580-587.

[20] Holub A, Liu Y H, Perona P. On constructing facial similarity maps. In *Proc. IEEE CVPR*, June 2007.

[21] Kruskal J B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 1964, 29(1): 1-27.

[22] Fang Y, Geman D. Experiments in mental face retrieval. In *Lecture Notes in Computer Science Volume 3546*, Kanade T, Jain A, Ratha N K (eds.), Springer, 2005, pp.637-646.

[23] Ruiz-del-Solar J, Navarrete P. FACERET: An interactive face retrieval system based on self-organizing maps. In *Proc. Int. Conf. Image and Video Retrieval*, July 2002, pp.157-164.

[24] Yang Z, Laaksonen J. Partial relevance in interactive facial image retrieval. In *Proc. the 3rd Int. Conf. Advances in Pattern Recognition*, Part II, August 2005, pp.216-225.

[25] Cao C, Kwak S, Belongie S, Kriegman D, Ai H. Adaptive ranking of facial attractiveness. In *Proc. the IEEE International Conference on Multimedia and Expo*, July 2014.

[26] Huang G B, Ramesh M, Berg T, Learned-Miller E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, Oct. 2007.

[27] Huang C, Ai H, Li Y, Lao S. High-performance rotation invariant multiview face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007, 29(4): 671-686.

[28] Yan D, Huang L, Jordan M I. Fast approximate spectral clustering. In *Proc. the 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, June 2009, pp.907-916.

Chong Cao received her B.S. degree in computer science and technology from Tsinghua University, Beijing, in 2010. She is currently a Ph.D. candidate at Tsinghua University. Her research interests include computer vision, pattern recognition and multimedia, with special focus on face retrieval and facial similarity learning.

Hai-Zhou Ai received his B.S., M.S. and Ph.D. degrees in computer applications from Tsinghua University, Beijing, in 1985, 1988 and 1991 respectively. He worked in the Flexible Production System Laboratory at the University of Brussels, Belgium, as a postdoctoral researcher from 1994 to 1996. He is currently a professor in the Department of Computer Science and Technology, Tsinghua University. His current research interests include image processing, computer vision and pattern recognition. He published more than 80 papers in peer-reviewed journals and international conferences. He is a senior member of IEEE and a member of CCF.