Face Photo-Sketch Synthesis and Recognition

Xiaogang Wang and Xiaoou Tang, *Fellow*, *IEEE*

Abstract—In this paper, we propose a novel face photo-sketch synthesis and recognition method using a multiscale Markov Random Fields (MRF) model. Our system has three components: 1) given a face photo, synthesizing a sketch drawing; 2) given a face sketch drawing, synthesizing a photo; and 3) searching for face photos in the database based on a query sketch drawn by an artist. It has useful applications for both digital entertainment and law enforcement. We assume that faces to be studied are in a frontal pose, with normal lighting and neutral expression, and have no occlusions. To synthesize sketch/photo images, the face region is divided into overlapping patches for learning. The size of the patches decides the scale of local face structures to be learned. From a training set which contains photo-sketch pairs, the joint photo-sketch model is learned at multiple scales using a multiscale MRF model. By transforming a face photo to a sketch (or transforming a sketch to a photo), the difference between photos and sketches is significantly reduced, thus allowing effective matching between the two in face sketch recognition. After the photo-sketch transformation, in principle, most of the proposed face photo recognition approaches can be applied to face sketch recognition in a straightforward way. Extensive experiments are conducted on a face sketch database including 606 faces, which can be downloaded from our Web site (http://mmlab.ie.cuhk.edu.hk/facesketch.html).

Index Terms—Face recognition, face sketch synthesis, face sketch recognition, multiscale Markov random field.

1 Introduction

An important application of face recognition is to assist law enforcement. Automatic retrieval of photos of suspects from the police mug shot database can help the police narrow down potential suspects quickly. However, in most cases, the photo image of a suspect is not available. The best substitute is often a sketch drawing based on the recollection of an eyewitness. Therefore, automatically searching through a photo database using a sketch drawing becomes important. It can not only help police locate a group of potential suspects, but also help the witness and the artist modify the sketch drawing of the suspect interactively based on similar photos retrieved [1], [2], [3], [4], [5], [6], [7]. However, due to the great difference between sketches and photos and the unknown psychological mechanism of sketch generation, face sketch recognition is much harder than normal face recognition based on photo images. It is difficult to match photos and sketches in two different modalities. One way to solve this problem is to first transform face photos into sketch drawings and then match a query sketch with the synthesized sketches in the same modality, or first transform a query sketch into a photo image and then match the synthesized photo with real photos in the gallery. Face sketch/photo synthesis not only helps face sketch recognition, but also has many other useful applications for digital entertainment [8], [9]. In this paper, we will study these two interesting and related problems: face sketch/photo synthesis and face sketch recognition.

Artists have a fascinating ability to capture the most distinctive characteristics of human faces and depict them on sketches. Although sketches are very different from photos in style and appearance, we often can easily recognize a person from his sketch. How to synthesize face sketches from photos by a computer is an interesting problem. The psychological mechanism of sketch generation is difficult to be expressed precisely by rules or grammar. The difference between sketches and photos mainly exists in two aspects: texture and shape. An example is shown in Fig. 1. The patches drawn by pencil on paper have different texture compared to human skin captured on a photo. In order to convey the 3D shading information, some shadow texture is often added to sketches by artists. For shape, a sketch exaggerates some distinctive facial features just like a caricature, and thus involves shape deformation. For example, if a face has a big nose in a photo, the nose drawn in the sketch will be even bigger.

1.1 Related Work

In psychology study, researchers have long been using various face drawings, especially line drawings of faces, to investigate face recognition by the human visual system [10], [11], [12], [13], [14]. Human beings can recognize caricatures quite well, which is a special kind of line drawings of faces, with particular details of a face accentuated, compared with the ability to recognize face photos. Presumably, the details which get accentuated in caricaturing are those which are characteristics of that individual. Someone even question whether caricatures are in any way better representations than natural images, since caricatures may contain not only the essential minimum of information but also some kind of "super-fidelity" due to the accentuated structures [10]. Bruce

- *X. Wang is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong. E-mail: xgwang@ee.cuhk.edu.hk.*
- *X. Tang is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong and the Shenzhen Institute of Advanced Technology, Shenzhen, China. E-mail: xtang@ie.cuhk.edu.hk.*

Manuscript received 8 Oct. 2007; revised 14 Apr. 2008; accepted 9 July 2008; published online 4 Sept. 2008.
Recommended for acceptance by L. O'Gorman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-10-0683.
Digital Object Identifier no. 10.1109/TPAMI.2008.222.

Fig. 1. Examples of a face photo and a sketch.

et al. [11] have also shown that computer-drawn "cartoons" with edges, pigmentation, and shading of the original image can be well recognized by human beings.

Some computer-based sketch synthesis systems have been proposed in recent years. Most of them have the line drawing output without much sketch texture which is useful to convey 3D shading information. In [8], [9], face shape was extracted from a photo and exaggerated by some rules to make the result more similar to a sketch in shape. They were not based on learning. Freeman et al. [15] proposed an example-based system which translated a line drawing into different styles. Chen et al. [16] proposed an example-based face cartoon generation system. It was also limited to the line drawings and required the perfect match between photos and line drawings in shape. These systems relied on the extraction of face shape using face alignment algorithms such as Active Appearance Model (AAM) [17]. These line drawings are less expressive than the sketches with shading texture. In this paper, we work on sketches with shading texture. It requires modeling both face shape and texture.

There was only limited research work on face sketch recognition because this problem is more difficult than photo-based face recognition and no large face sketch database is available for experimental study. Methods directly using traditional photo-based face recognition techniques such as the eigenface method [1] and the elastic graph matching method [2] were tested on two very small sketch data sets with only 7 and 13 sketches, respectively.

In our previous work [3], [4], a face sketch synthesis and recognition system using eigentransformation was proposed. It was not limited to line drawing and could synthesize sketches with more texture. The transformation was directly applied to the whole face image. In [4], it was shown that a synthesized sketch by eigentransformation would be a good approximation to a sketch drawn by an artist only if two conditions are satisfied: 1) A face photo can be well reconstructed by PCA from training samples and 2) the photo-sketch transformation procedure can be approximated as linear. In some cases, especially when the hair region is included, these conditions are hard to be satisfied. Human hair varies greatly over different people and cannot be well reconstructed by PCA from training samples. PCA and Bayesian classifiers were used to match the sketches drawn by the artist with the pseudosketches synthesized from photos. Liu et al. [5] proposed a nonlinear face sketch synthesis and recognition method. It followed the similar framework as in [3], [4]. However, it did eigentransformation on local patches instead of the global face images. It used a kernel-based nonlinear LDA classifier for recognition. The drawback of this approach is that the local patches are synthesized independently at a fixed scale and face structures in large scale, especially the face shape, cannot be well learned. Zhong et al. [6] and Gao et al. [7] proposed an approach using an embedded hidden Markov model and a selective ensemble strategy to synthesize sketches from photos. The transformation was also applied to the whole face images and the hair region was excluded.

1.2 Our Approach

In this paper, we develop a new approach to synthesize local face structures at different scales using a Markov Random Fields model. It requires a training set containing photo-sketch pairs. We assume that faces to be studied are in a frontal pose, with normal lighting and neutral expression, and have no occlusions. Instead of directly learning the global face structure, which might be too complicated to estimate, we target at local patches, which are much simpler in structure. The face region is divided into overlapping patches. During sketch synthesis, for a photo patch from the face to be synthesized, we find a similar photo patch from the training set and use its corresponding sketch patch in the training set to estimate the sketch patch to be synthesized. The underlying assumption is that, if two photo patches are similar, their sketch patches should also be similar. In addition, we have a smoothness requirement that neighboring patches on a synthesized sketch should match well. The size of patches decides the scales of the face structures which can be learned. We use a multiscale Markov Random Fields model to learn face structures at different scales. Thus, local patches in different regions and scales are learned jointly instead of independently as in [5]. This approach can also be used to synthesize face photos given sketches. Our sketch/photo algorithm is relevant to [18], which used MRF to estimate *scenes*, such as motion and range map, from *images*.

During the face sketch recognition stage, there are two options to reduce the modality difference between photos and sketches: 1) All of the face photos in the gallery are first transformed to sketches using the sketch synthesis algorithm and a query sketch is matched with the synthesized sketches, and 2) a query sketch is transformed to a photo and the synthesized photo is matched with real photos in the gallery. We will evaluate both options in Section 3. After the photos and sketches are transformed into the same modality, in principle, most of the proposed face photo recognition approaches can be applied to face sketch recognition in a straightforward way. In this paper, we will evaluate the performance of several appearance-based face recognition approaches.

2 Face Sketch Synthesis Using the Multiscale Markov Random Fields Model

In this section, we describe our sketch synthesis approach based on local patches. This approach can be easily extended to face photo synthesis by simply exchanging the roles of photos and sketches. The steps of our sketch synthesis algorithm can be outlined as follows. The input is a face photo and the output is a synthesized sketch:

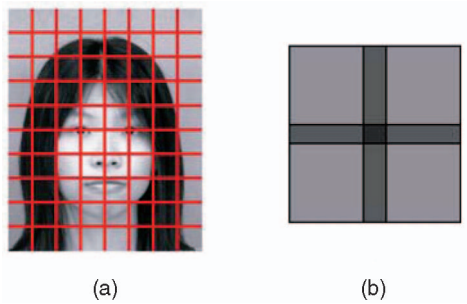


Fig. 2. (a) The face region is divided into patches. (b) The neighboring patches overlap.

1. Perform geometry alignment and transform color space in the preprocessing step.
2. Patch matching: For each patch y_j on the input photo, find K photo patches $\{\tilde{y}_j^l\}_{l=1}^K$ from the training set best matching y_j in appearance, and use their corresponding sketch patches $\{\tilde{x}_j^l\}_{l=1}^K$ as candidates for the estimation of the synthesized sketch patch x_j corresponding to y_j.
3. Build a multiscale Markov network and conduct belief propagation to estimate the sketch patches $\{\hat{x}_j\}$ of the input photo.
4. Synthesize the sketch image by stitching the estimated sketch patches $\{\hat{x}_j\}$.

Each of the above steps will be explained in the following sections.

2.1 Preprocessing

In the preprocessing step, all the photos and sketches are translated, rotated, and scaled such that the two eye centers of all the face images are at fixed position. This simple geometric normalization step aligns the same face components in different images roughly to the same region. Face photos can be gray or color images. When photos are in color, we fist convert the *RGB* color space to the *Luv* color space, since euclidean distance in *Luv* space better correlates to the perceived change in color.

2.2 Patch Matching

The face region is divided into patches and the neighboring patches overlap, as shown in Fig. 2. For each patch on the input photo image, we try to estimate its sketch patch. A smoothness constraint requires that two neighboring synthesized sketch patches have similar intensities or colors at the pixels inside their overlapping region. How to measure the compatibility between two neighboring synthesized sketch patches is formulated in Section 2.3.

In order to estimate the sketch patch x_j of the input photo patch y_j, K candidate sketch patches $\{\tilde{x}_j^l\}_{l=1}^K$ are collected from the training set. We assume that if a patch $\tilde{y}_j^l$ found on a training photo is similar to the patch y_j on the input photo in appearance, the sketch patch $\tilde{x}_j^l$ corresponding to $\tilde{y}_j^l$ is considered as one of the good candidates for x_j. The procedure for searching candidate sketch patches is described in Fig. 3. For each local patch y_j on the input photo, we find its corresponding position on a training photo. Since face images are not exactly aligned in shape,

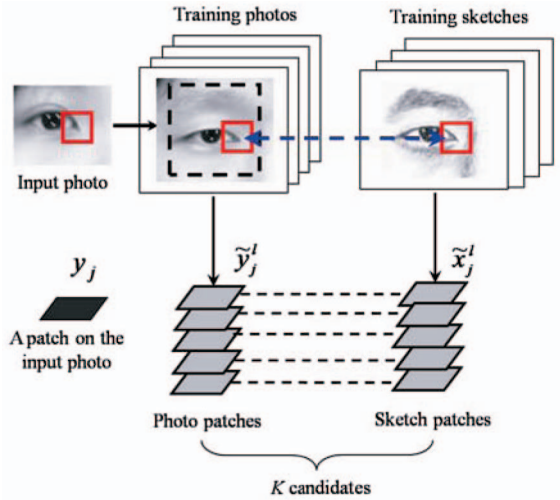


Fig. 3. Procedure for searching candidate sketch patches for a patch on the input photo. For a patch on the input photo, the dash black window is the searching range on the training photos.

the same face components on different images may not locate exactly at the same position. We cannot directly sample the patch on the training photo at the same position as on the input photo. Instead, we set a searching range around this position indicated by the black dash window in Fig. 3. Searching inside this range, we find the patch best matching y_j as the sampled patch from this training photo. Here, we use the euclidean distance between intensities or colors of two photo patches as the matching measure. Let I be a photo in the training set and R be a patch inside the searching range, then the distance is

$$\begin{aligned} D_R &= \sum_{i \in R} \big(y_j(i) - I_R(i)\big)^2 \\ &= \sum_i y_j^2(i) + \sum_i I_R^2(i) - 2\sum_i y_j(i) I_R(i), \end{aligned} \tag{1}$$

where $y_j(i)$ and $I_R(i)$ are the intensities or color vectors at pixel i on the input photo patch and the patch R, respectively, on the training photo. After searching through the entire training set, for each input photo patch y_j, we have a patch sampled from each training photo. Suppose there are M photo-sketch pairs in the training set. We select K photo patches best matching y_j in appearance from the M training photos. Each patch on a training photo has a corresponding patch on its training sketch. We use the K sketch patches corresponding to the K selected photo patches from the training set as candidates for the possible states of x_j. An example is shown in Fig. 4.

Patch matching is the most time-consuming part of our algorithm. This part can be speeded up using integral computation [19] and 2D fast Fourier transform. In order to find the patch on a training photo I best matching the input photo patch y_j, the distance in (1) has to be computed for all possible patches R. For each input photo, $\sum_i y_j^2(i)$ only need to be computed once. $\sum_i I_R^2(i)$ can be computed efficiently using the trick of integral computation which first computes an integral image once and then is able to compute statistics over any rectangle regions over the image very fast. More importantly, this term can be computed offline and saved

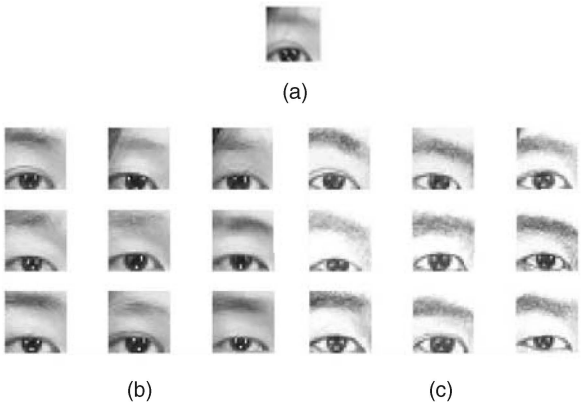


Fig. 4. Example of collecting candidate sketch patches from the training set. (a) A patch on the input photo. (b) Photo patches selected from the training photos best matching the patch on the input photo. (c) Sketch patches corresponding to the selected photo patches from the training set.

for each training photo. The correlation term $\sum_i y_j(i)I_R(i)$ costs the most computation since it has to be computed for each pair of input photo and training photo online. Fortunately, it is well known that correlation can be speeded up using fast Fourier transform.

If we simply choose a single training sketch patch whose photo patch best matches the input photo patch y_j in appearance as an estimate of sketch patch x_j, the synthesized sketch image is not smooth with mosaic effect. Also, because sketch patches are estimated independently, when estimating a sketch patch, information from the remaining face region is not considered. This is quite different from the process of an artist drawing a sketch. An artist often considers the whole face structure when drawing a small patch. In our approach, to estimate a sketch patch, we keep K candidates as its possible values, and also require that neighboring synthesized sketch patches match well. Thus, all the sketch patches need to be jointly modeled. The synthesized sketch image should closely match the input photo in appearance and be smooth in the meanwhile. To reach this goal, a Markov network is used to model the process of sketch synthesis. It will be explained in Sections 2.3 and 2.5.

2.3 Markov Network at a Single Scale

The graphical model representation of the Markov network is shown in Fig. 5. The whole face region is divided into N patches. Each node on the network is a sketch patch or a photo patch. Let y_j and x_j be the input photo patch and the sketch patch to be estimated at face patch j. The dependency between y_j and x_j, written as $\Phi(x_j, y_j)$, provides the local evidence for x_j. The variable x_j is connected to other sketch nodes in its neighborhood by the compatibility function $\Psi(x_j, x_{j'})$. The joint probability of the input photo and its sketch can be written as

$$p(x_1, \ldots, x_N, y_1, \ldots, y_N) = \prod_{j_1 j_2} \Psi(x_{j_1}, x_{j_2}) \prod_j \Phi(x_j, y_j), \quad (2)$$

where x_i has a discrete representation taking values only on K possible states, which are candidate sketch patches

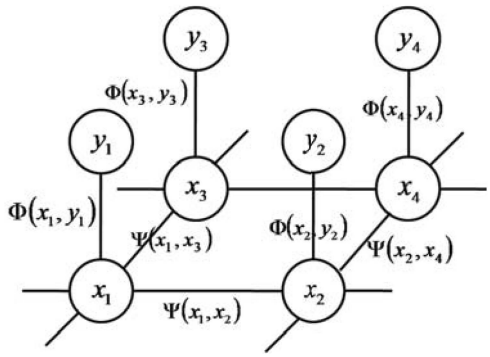


Fig. 5. The graphical model of Markov network.

$\{\tilde{x}_j^l\}_{l=1}^K$ collected through patch matching, as described in Section 2.2.

The local evidence is computed as

$$\Phi(\tilde{x}_j^l, y_j) = exp\{-\|\tilde{y}_j - y_j\|^2 / 2\sigma_e^2\}, \quad (3)$$

where $\tilde{y}_j^l$ is the corresponding photo patch of the candidate sketch patch $\tilde{x}_j^l$ from the training set. If $\tilde{x}_j^l$ is a good estimation of x_j, $\tilde{y}_j^l$ should be similar to y_j.

Let j_1 and j_2 be two neighboring patches with overlapping region A. Let $d_{j_1 j_2}^l$ be the intensity or color vector of the lth candidate sketch patch $\tilde{x}_{j_1}^l$ at j_1 inside A. Let $d_{j_1 j_2}^m$ be the intensity or color vector of the mth candidate sketch patch $\tilde{x}_{j_2}^l$ at j_2 inside A. The compatibility function is computed as

$$\Psi(\tilde{x}_{j_1}^l, \tilde{x}_{j_2}^m) = exp\{-\|d_{j_1 j_2}^l - d_{j_1 j_2}^m\| / 2\sigma_c^2\}. \quad (4)$$

If both $\tilde{x}_{j_1}^l$ and $\tilde{x}_{j_2^m}$ are estimations of synthesized patches, they should have consistent intensities or colors in their overlapping region.

Given the Markov network, the sketch patches can be estimated by taking maximum a posteriori (MAP) estimator $\hat{x}_{jMAP}$ or minimum mean-square error (MMSE) estimator $\hat{x}_{jMMSE}$

$$\hat{x}_{jMAP} = \arg\max_{[x_j]} \max_{[x_i, i \neq j]} P(x_1, \ldots, x_N | y_1, \ldots, y_N), \quad (5)$$

$$\hat{x}_{jMMSE} = \sum_{x_j} x_j \sum_{[x_i, i \neq j]} P(x_1, \ldots, x_N | y_1, \ldots, y_N). \quad (6)$$

We use belief propagation [20] to do inference. Messages from local regions propagate along the Markov network to reach optimal solution. When the network has no loops, (5) and (6) can be exactly computed using the "message-passing" rules [18]. The MAP estimate at node j is

$$\hat{x}_{jMAP} = \arg\max_{[x_j]} \Phi(x_j, y_j) \Pi_k M_j^k(x_j), \quad (7)$$

where M_j^k is the message from the neighbor node k to node j, and is computed as

$$M_j^k = \max_{[x_k]} \Psi(x_j, x_k) \Phi(x_k, y_k) \prod_{l \neq j} \tilde{M}_k^l(x_k), \quad (8)$$

where $\tilde{M}_j^k$ is M_j^k computed from the previous iteration. The MMSE estimate at node j is

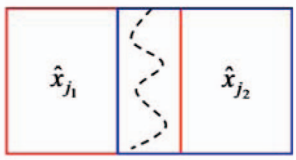


Fig. 6. Minimum error boundary cut between two overlapping estimated sketch patches $\hat{x}_{j_1}$ and $\hat{x}_{j_2}$.

$$\hat{x}_{jMSSE} = \sum_{x_j} x_j \Phi(x_j, y_j) \prod_k M_j^k(x_j), \tag{9}$$

$$M_j^k = \sum_{x_k} \Psi(x_j, x_k) \Phi(x_k, y_k) \prod_{l \neq j} \tilde{M}_k^l(x_k). \tag{10}$$

When messages pass along the network, a sketch patch receives information not only from neighboring patches but also from other patches far away. A detailed description of belief propagation can be found in [18], [20].

Computing the MAP and MMSE values for a Markov network with loops is prohibitive. But the above propagation rules can still be applied to get the approximated solution. From our experimental results, the MAP estimate has a better performance while the MMSE estimate often brings blurring effect. Besides belief propagation, there are also other approaches, such as graph cut [21], to approximate the optimal solution of Markov Random Fields.

2.4 Stitching Sketch Patches

Since neighboring sketch patches have overlap regions, to synthesize the whole sketch image, one could average the neighboring patches. However, this will lead to blurring effect. Instead, we make a minimum error boundary cut between two overlapping patches on the pixels where the two patches match best [22], as shown in Fig. 6. The minimum cost path through the error surface is computed with dynamic programming.

2.5 Markov Network at Multiple Scales

In Sections 2.2 and 2.3, we assume that all the image patches have the same size. A drawback of using a uniform scale of Markov random field is that it cannot address the long-range dependency among local patches. When an artist draws a patch, he usually refers to the larger structure around that patch. Sometimes even though two photo patches are very similar, their corresponding sketch patches might be very different. One example is shown in Fig. 7. The size of the patch decides the scale of the face structures to be learned. When the patch is small, some shadow added by the artist to convey 3D shading information and some face structures such as the face contour, eyebrows, and the bridge of the nose might be missed. It seems that these structures have to be learned using larger patches. However, patches in large size will lead to distortions and mosaic effect on the synthesized sketch. To overcome this conflict, we develop a multiscale Markov Random Fields model which learns face structures across different scales.

As shown in Fig. 8, a multiscale Markov random fields model is composed of L layers of random fields, $x^{(1)}, x^{(2)}, \ldots, x^{(L)}$, with different resolutions. $x^{(1)}$ is the finest

Fig. 7. When an artist draws a sketch patch, he often refers to the larger structure around that patch. In this figure, the two local patches from the bridge of the nose and the cheek have similar appearance in photos, but the corresponding sketch patches drawn by the artist are very different.

scale random fields with the smallest patch size. $x^{(L)}$ is the coarsest scale random fields with the largest patch size. A node at layer n is decomposed into s^2 nodes at layer $n-1$, where s is the resolution reduction rate. They are defined as neighboring nodes in different scales. It is assumed that the distribution of $x^{(n)}$ only depends on the neighboring layers:

$$\begin{aligned} &P(x^{(1)}, \ldots, x^{(L)}, y^{(1)}, \ldots, y^{(L)}) \\ &\quad = \prod_{n=1}^{L} \Omega(x^{(n)}, y^{(n)}) \prod_{n=1}^{L-1} \Theta(x^{(n)}, x^{(n+1)}), \end{aligned} \tag{11}$$

where $y^{(1)}, \ldots, y^{(L)}$ are the photo images on different layers. Their only difference is the patch size. Thus, the joint probability distribution (2) can be extended by adding the connection between the hidden variable $x_j^{(n)}$ and its neighboring nodes in adjacent scale layers $n-1$ and $n+1$:

$$\begin{aligned} &P(x^{(1)}, \ldots, x^{(L)}, y) \\ &\quad = P(x^{(1)}, \ldots, x^{(L)}, y^{(1)}, \ldots, y^{(L)}) \\ &\quad = P(x_1^{(1)}, \ldots, x_{N_1}^{(1)}, x_1^{(L)}, \ldots, x_{N_L}^{(L)}, y_1^{(1)}, \ldots, y_{N_L}^{(L)}) \\ &\quad = \prod_{n=1}^{L} \prod_i \Phi(x_i^{(n)}, y_i^{(n)}) \prod_{i,j,m,n} \Psi(x_i^{(m)}, x_j^{(n)}). \end{aligned} \tag{12}$$

When $m = n, x_i^{(m)}$ and $x_j^{(n)}$ are of the same size and are neighbors in space. When $m = n-1$, the region covered by $x_i^{(m)}$ is part of that covered by $x_j^{(n)}$, and $\Psi(x_i^{(m)}, x_j^{(n)})$ is defined in the same way as (4) by comparing the intensity or color difference in the overlapping region of two patches. We use the same belief propagation rules as described in Section 2.3, except that messages pass between scale layers. We take the finest resolution layer $x^{(1)}$ as the synthesis result.

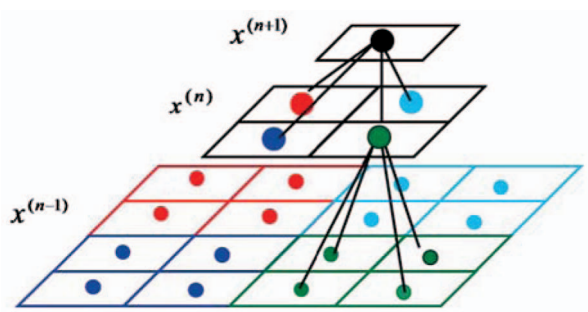


Fig. 8. Pyramid structure of the multiscale Markov Random Field model.

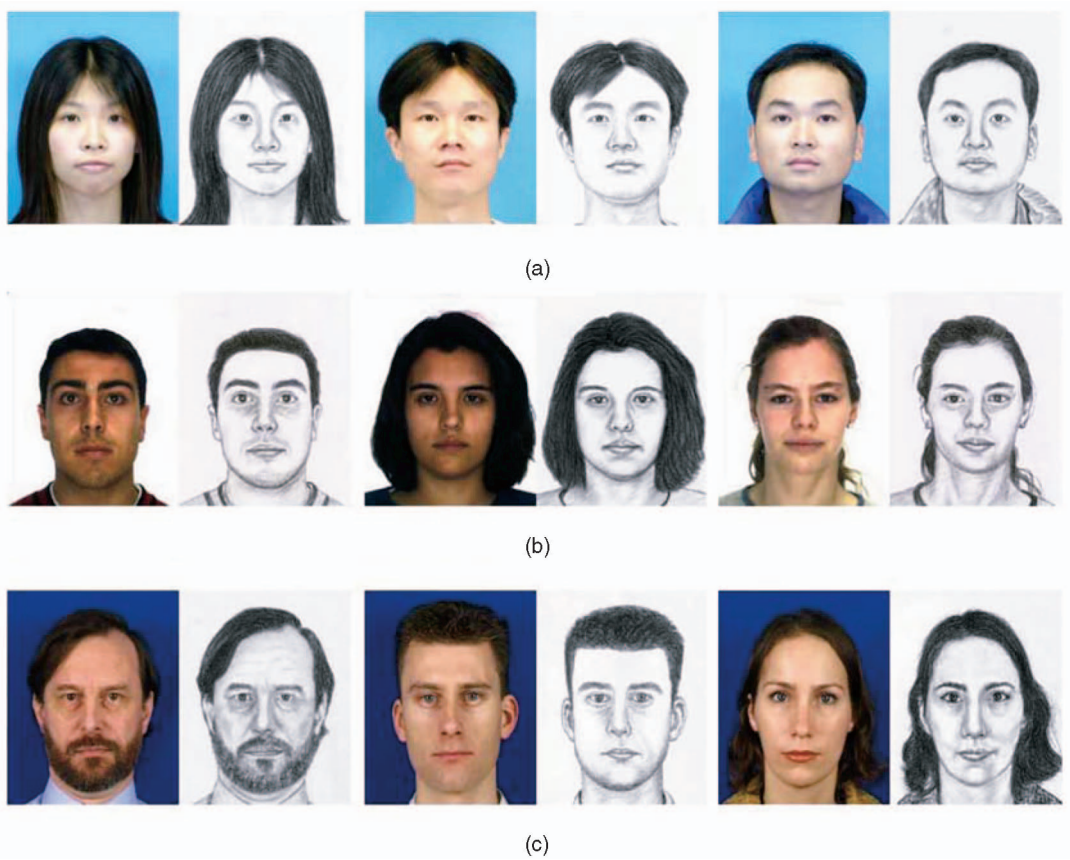


Fig. 9. Examples of face photos and sketches from (a) the CUHK student database, (b) the AR database, and (c) the XM2VTS database.

2.6 Discussion

Our sketch synthesis approach is based on local patches. It does not require that a face photo be well reconstructed by PCA from the training set, and the photo-sketch transform can be approximated as a linear procedure as in the global eigentransformation approach [4]. So it can synthesize more complicated face structure such as hair which is hard for global eigentransformation. Hair is an important feature for entertainment applications of the sketch synthesis algorithm. For the recognition task, in some cases, especially when two images of the same person are captured long time apart, e.g., several months or years, hair may not be a stable feature for recognition since its style may change. However, under some situations when this interval is not long, hair is still a distinctive feature for recognition. When the police ask the witness to generate the sketch of a suspect, hair feature is often required.

The approach proposed in [5] was also based on local patches. However, it has several key differences with our method. First, in [5], sketch patches were synthesized independently, while in our approach, sketch patches are jointly modeled using MRF. In our approach, a sketch patch receives information not only from neighboring patches but also from other patches far away by belief propagation. Second, in [5], the size of patches is fixed at one scale. Experimental results in [5] showed that small and large patch sizes led to different problems in the synthesis results. Our approach synthesizes sketch patches over different scales. Because of these two reasons, our approach can better learn the long-range face structure and global shape feature, and generate smoother results. Third, method in [5] synthesized a local patch through the linear combination of candidate patches. This brought blurring effect. Alternatively, our approach finally chooses only one candidate sketch patch as an estimate. We will compare these approaches through experimental evaluation.

3 Experimental Results

We build a face photo-sketch database for experimental study. It includes 188 faces from the CUHK student database, 123 faces from the AR database [23], and 295 faces from the XM2VTS database [24]. For each face, there is a sketch drawn by the artist and a photo taken in a frontal pose, under normal lighting condition, and with a neutral expression. Some examples are shown in Fig. 9. The data can be downloaded from our Web site.[1]

3.1 Face Sketch Synthesis

We conduct the face sketch synthesis experiments on the three face databases. In the CUHK student database, 88 faces are selected for training and the remaining 100 faces are selected for testing. In the XM2VTS database, 100 faces are selected for training and the remaining 195 faces for testing. In the AR database, we use the leave-one-out strategy, i.e., each time one face is selected for testing and the remaining 122 faces are used for training. In Fig. 10, we show some examples of sketch synthesis results. We choose examples from each of the three databases. Please see more results

1. http://mmlab.ie.cuhk.edu.hk/facesketch.html.

Fig. 10. Face sketch synthesis results: (a) photo; (b) sketch drawn by the artist; and (c) synthesized sketch.

from our Web site. We use the MAP estimate. Our multiscale Markov Random Fields model has two layers. At the first layer, the local patch size is 10×10. At the second layer, the local patch size is 20×20. The face region is in size of 200×250. We set $\sigma_e = 0.5$, $\sigma_c = 1$ in (3) and (4) throughout the experiments. From (2), (3), and (4), only the ratio σ_e/σ_c actually matters. From our empirical study, good performance can be achieved when σ_e/σ_c takes value between 0.3 and 0.8.

In Fig. 11, we compare the synthesized sketches after different numbers of iterations of belief propagation. At the beginning (zero iteration), a sketch is synthesized from the sketch patches best matching input photo patches without considering smoothness constraint. The result is noisy and has mosaic effect. Based on our statistic, more than 80 percent of these estimated sketch patches are subsequently corrected by Markov analysis. Belief propagation quickly converges after five iterations and the quality of the synthesized sketch is greatly improved after belief propagation.

In Fig. 12, we compare two different estimation methods: MMSE and MAP. MMSE estimation has a blurring effect. The results of MAP have sharper edges and more clear contours, and are more similar to the sketches drawn by the artist.

In Fig. 13, we compare the sketch synthesis performance using the one-scale MRF model and the multiscale MRF

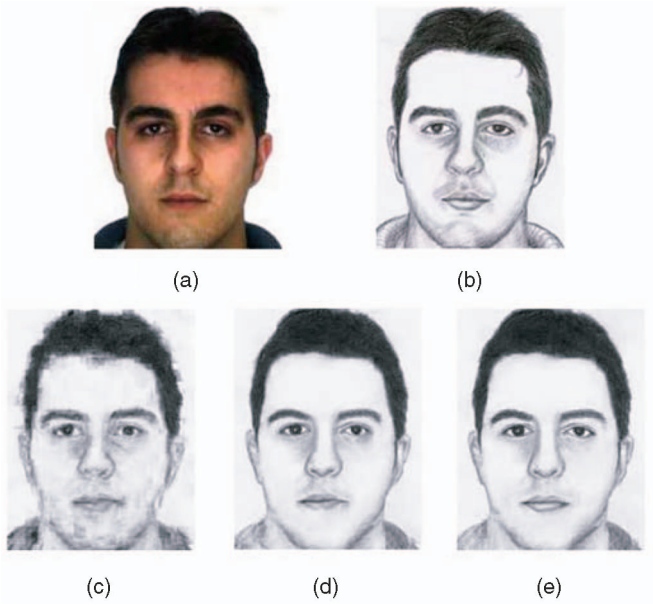


Fig. 11. Synthesized sketches after different numbers of iterations of belief propagation. (a) photo, (b) sketch drawn by the artist, (c) after 0 iterations, (d) after 5 iterations, (e) after 40 iterations.

model. Under the one-scale MRF model, when the patch size is small (10×10), some shadow texture and face structures, such as the lower part of the face contour and ear, are missing. These structures can be learned using patches of larger size. However, there is distortion and mosaic effect when the patch size is large (20×20). Using the multiscale MRF model, the result has less distortion and mosaic effect compared with the results learned only at the coarse resolution layer, and more face structures are synthesized compared with the results leant only at the fine resolution layer. Based on our empirical study, there is no significant improvement in the performance of sketch synthesis when increasing the numbers of layers to three or more.

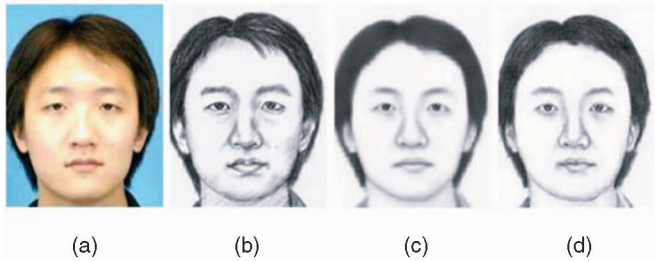


Fig. 12. Comparison of sketch synthesis results using MMSE estimate and MAP estimate: (a) photo; (b) sketch drawn by the artist; (c) synthesized sketch using MMSE estimate; and (d) synthesized sketch using MAP estimate.

In Fig. 14, we compare the sketch synthesis results using the multiscale MRF model and the global eigentransformation approach proposed in [4]. Since our synthesis is based on local patches, it works better for synthesizing local textures than global transformation. Our results are sharper with less noise. Global eigentransformation required that the face data have a Gaussian distribution in the high-dimensional space

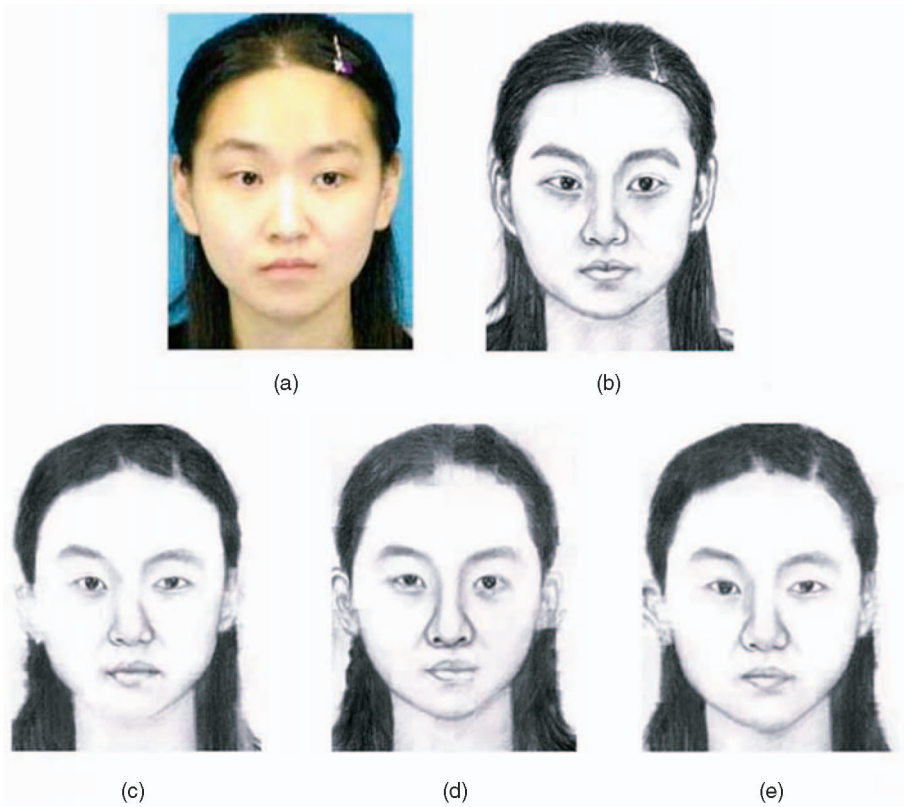


Fig. 13. Comparison of sketch synthesis results using the one-scale MRF model and multiscale MRF model. The size of the face region is 200×250: (a) face photo; (b) sketch drawn by the artist; (c) synthesized face sketch with patch size of 10×10; (d) synthesized face sketch with patch size of 20×20; and (e) sketch synthesized using two-level MRF. The patch size at the first level is 10×10 and the patch size at the second level is 20×20.

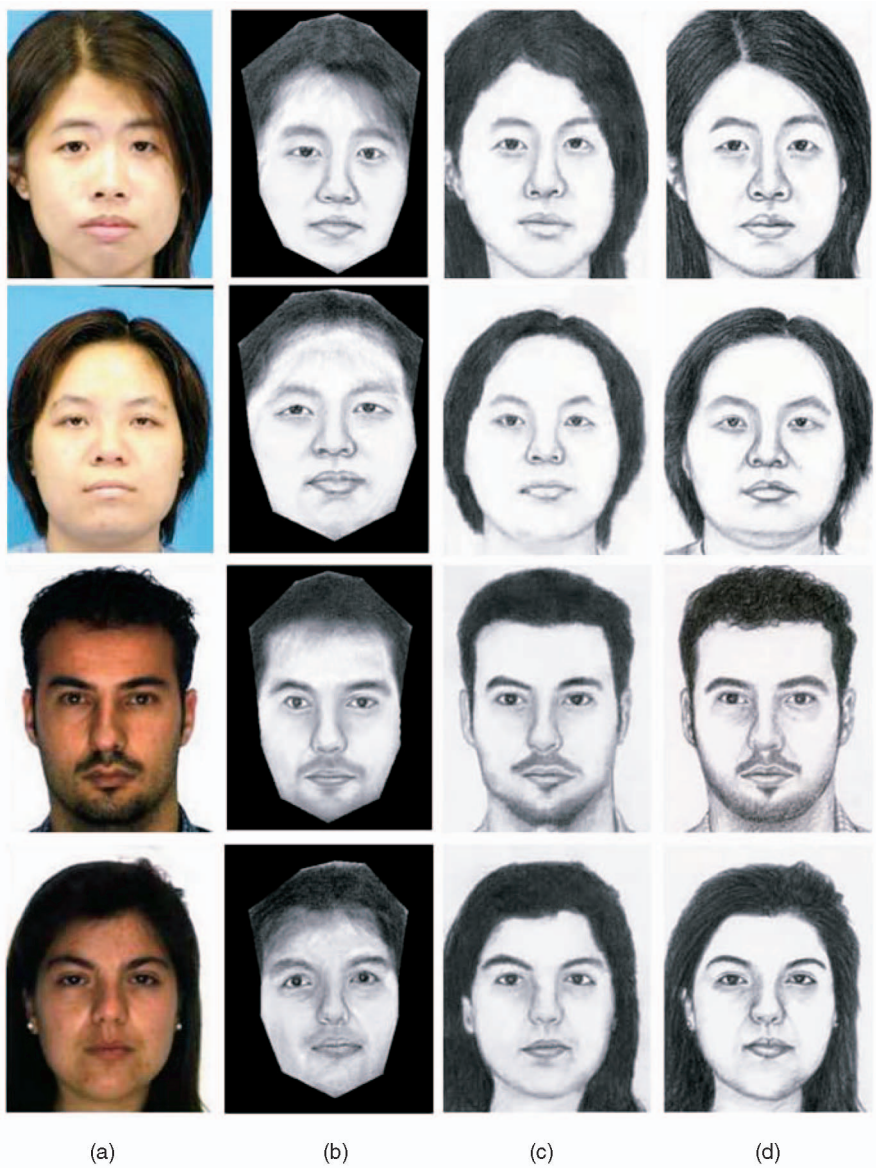


Fig. 14. Comparison of the sketch synthesis results of global eigentransformation and the multiscale MRF model when the hair region is included: (a) photos; (b) sketches synthesized by global eigentransformation; (c) sketches synthesized by the multiscale MRF model; and (d) sketches drawn by the artist.

and a testing face photo can be well reconstructed by PCA from the examples in the training set. However, since human hair has a large variety of styles, when the hair region is included, the distribution of face vectors cannot be estimated as Gaussian and face photos cannot be well reconstructed by PCA. In Fig. 14, eigentransformation has a much worse performance on synthesizing hair. The hair region also leads to errors on the synthesis of other regions in the face. Our approach has no such constraint. It synthesizes a variety of hair styles quite well.

In Fig. 15, we compare the face sketch synthesis results using our approach and the approach proposed in [5]. Liu et al. [5] used the same database as ours. We choose the examples which were published in [5] for comparison. Using our approach, the synthesized sketches are sharper and cleaner. Large face structures and shape features in sketches are well captured.

Our sketch synthesis algorithm has a relatively high computational cost because of patch matching. After being speeded up using integral computation and 2D fast Fourier transform as mentioned in Section 2.2, it takes about 3 minutes to synthesize a sketch running on a computer with 3 GHz CPU. Note that if multiple CPUs are available, patch matching can be done in parallel, and thus, sketches can be synthesized faster.

3.2 Face Photo Synthesis

Our approach can also synthesize a face photo, given a sketch drawn by an artist, by simply switching roles of photos and sketches. In Fig. 16, we show the face photo synthesis results. The experiment settings and parameters are the same as for experiments in Section 3.1.

Fig. 15. Comparison of the sketch synthesis results using a nonlinear approach proposed in [5] and our multiscale MRF model: (a) photos; (b) sketches synthesized by the nonlinear approach proposed in [5]; (c) sketches synthesized by our multiscale MRF model; and (d) sketches drawn by the artist.

3.3 Face Sketch Recognition

At the face sketch recognition stage, there are two strategies to reduce the modality difference between photos and sketches: 1) All of the face photos in the gallery are first transformed to sketches using the sketch synthesis algorithm, and a query sketch is matched with the synthesized sketches; 2) a query sketch is transformed to a photo and the synthesized photo is matched with real photos in the gallery. We will evaluate both strategies in the face sketch recognition experiments. After the photos and sketches are transformed into the same modality, in principle, most of the proposed face photo recognition approaches can be applied to face sketch recognition in a straightforward way. In this paper, we will evaluate the performance of several appearance-based face recognition approaches, including PCA [25], Bayesianface (Bayes) [26], Fisherface [27], null-space LDA [28], dual-space LDA [29], and Random Sampling LDA (RS-LDA) [30], [31].

The 606 photo-sketch pairs are divided into three subsets. One hundred and fifty-three photo-sketch pairs in subset I are used for the training of photo/sketch synthesis. One hundred and fifty-three photo-sketch pairs in subset II are used for the training of subspace classifiers. When using strategy 1), the photos in subset II are first transformed to synthesized sketches using subset I as the training set. Then, the synthesized sketches and the sketches drawn by the artist in subset II are used to train subspace classifiers such as Fisherface and random sampling LDA. Strategy 2) is similar, except that sketches and photos switch roles. Three hundred photo-sketch pairs in subset III are used for testing. The division of the data set is the same as in [4].

In Table 1, we compare the rank-one recognition accuracy using different sketch/photo synthesis algorithms

Fig. 16. Sketch synthesis results: (a) sketch drawn by the artist; (b) photo; and (c) synthesized photo.

and face recognition methods. We evaluate three sketch/photo synthesis methods:

- sketch synthesis using global eigentransform: as described in [4], face photo texture and shape are first separated, transformed to sketch texture and shape, and finally combined for recognition;
- sketch synthesis using the multiscale MRF model (multiscale_MRF_SS) with strategy 1);
- photo synthesis using the multiscale MRF model (multiscale_MRF_SP) with strategy 2).

Sketch synthesis using the multiscale MRF model (multiscale_MRF_SS) achieves better results than global eigentransform. Photo synthesis using the multiscale MRF model (multiscale_MRF_SP) achieves even better results. These observations hold even when different face recognition methods are used. We evaluate six different appearance-based face recognition methods. Random Sampling LDA (RS-LDA) always has the best performance over different sketch/photo synthesis methods.

In Table 2, we compare the cumulative match scores of our methods with two conventional face recognition methods: Eigenface [25] and Elastic Graph Matching (EGM) [32] and a nonlinear face sketch recognition approach proposed in [5]. Eigenface and EGM have very poor recognition performance on our data set, with the first match accuracies of no more than 30 percent, which is consistent with results shown in [3],

TABLE 1
Rank-One Recognition Accuracy Using Different Face Sketch/Photo Synthesis Methods and Face Recognition Methods (in Percent)

	Eigentransfom	Multiscale_MRF_SS	Multiscale_MRF_SP
PCA	75.0	84.0	84.3
Bayes	81.3	89.0	93.7
Fisherface	79.7	89.3	93.3
Null−space LDA	84.0	90.7	94.7
Dual−space LDA	88.7	92.00	95.7
RS−LDA	90.0	93.3	96.3

See more detailed description in the text.

TABLE 2
Cumulative Match Scores Using Different Face Sketch Recognition Methods (in Percent)

	1	2	3	4	5	6	7	8	9	10
Eigenface	6.3	8.0	9.0	9.3	11.3	13.3	14.0	14.0	14.3	16.0
EGM	25.3	32.3	40.0	43.0	46.7	48.7	53.0	54.3	56.3	57.7
Nonlinear face sketch recognition [5]	87.7	92.0	95.0	97.3	97.7	98.3	98.7	99.0	99.0	99.0
Eigentransform RS−LDA	90.0	94.0	96.7	97.3	97.7	97.7	98.3	98.3	99.0	99.0
Multiscale_MRF_SS + RSLDA	93.3	94.6	97.3	98.3	98.3	98.3	98.3	99.0	99.0	99.0
Multiscale_MRF_PS + RSLDA	96.3	97.7	98.0	98.3	98.7	98.7	99.3	99.3	99.7	99.7

TABLE 3
Face Recognition Accuracies with Variations of Lighting, Expressions, and Occlusions (in Percent)

	Lighting	Expression	Occlusion
Direct matching by Euclidean distance (Sketch query)	48.9	64.4	37.8
Direct matching by Euclidean distance (Photo query)	51.1	66.7	42.2
Random Sampling LDA (Sketch query)	77.8	75.6	62.2
Random Sampling LDA (Photo query)	82.2	78.9	67.8

The recognition accuracies using sketches and photos taken under normal conditions as queries are compared. See more details in the text.

[4]. Liu et al. [5] proposed a face sketch recognition approach which synthesized sketches based on local patches and used kernel-based nonlinear LDA classifier for recognition. It had the first match rate of 86.7 percent and the tenth match rate of 99 percent. Our approaches significantly improve the first match to 96.3 percent and the tenth match to 99.7 percent.

In this work, we assume that all faces are in a frontal pose, with normal lighting and neutral expression, and have no occlusions. If the input photo is taken under a very different condition than the photos in our training set, our sketch synthesis algorithm may not work well with these significant variations. Solving this problem is a direction of future study. In order to match a normal sketch with photos with significant variations on poses, lightings, expressions, and occlusions, it is better to first transform the sketch to a pseudophoto. Since we can choose the photos in training set as those taken under a normal condition, there is no difficulty at this step. The synthesized photo from a sketch looks like a photo taken under a normal condition. Then, at the step of matching the pseudophoto with photos taken under different conditions, the difficulties caused by these intrapersonal variations have to be overcome. However, it becomes a traditional photo-to-photo face matching problem without special consideration on sketches. Many studies have been done to solve these problems for photo-to-photo face matching.

To illustrate this, we evaluate the performance of face sketch recognition on the AR database which includes face photos taken under different lighting conditions, with different expressions and occlusions. Each face has one photo taken under a normal condition, two photos taken with different occlusions (sun glasses and scarf), three photos taken under different lighting conditions, and three photos taken with different expressions. See more details in [23]. The 123 faces are divided into two subsets with 78 faces in the training set and 45 faces in the testing set. Random sampling LDA classifiers are trained to suppress the effect of these intrapersonal variations. For a face sketch in the testing set, its pseudophoto is synthesized using the sketches and photos taken under a normal condition in the training set. With the random sampling LDA classifiers learned from the training set, the pseudophoto synthesized from a sketch is used as a query image to match photos with different variations of lightings, expressions, and occlusions in the testing set. The recognition accuracies under different conditions are reported in Table 3. For comparison, instead of using a sketch, a photo taken under a normal condition is also used as query with the same classifiers. Its recognition accuracies are also reported in Table 3. Compared with the recognition accuracies reported in Table 2, the variations of lightings, expressions, and occlusions make the recognition task more difficult. Random sampling LDA significantly improves the recognition performance compared with directly matching images using correlation. However, the difference between using a pseudophoto synthesized from a sketch and a real photo taken under a normal condition as query is not very significant. This means that in order to solve the problems of variations caused by lighting, expressions, and occlusions in face sketch recognition, we can use the techniques which have been developed to solve these problems in photo-to-photo matching.

4 Conclusion

In this paper, we proposed a novel face photo-sketch synthesis and recognition system. Given a face photo (or a face sketch), its sketch (or photo) can be synthesized using a multiscale Markov Random Fields model, which learns the face structure across different scales. After the photos and the sketches have been transformed to the same modality, various face recognition methods are evaluated for the face sketch recognition task. Our approach is tested on a face sketch database including 606 faces. It outperforms existing face sketch synthesis and recognition approaches.

References

[1] R.G. Uhl and N.D.V. Lobo, "A Framework for Recognizing a Facial Image from a Police Sketch," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 1996.

[2] W. Konen, "Comparing Facial Line Drawings with Gray-Level Images: A Case Study on Phantomas," *Proc. Int'l Conf. Artificial Neural Networks,* 1996.

[3] X. Tang and X. Wang, "Face Sketch Recognition," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 14, no. 1, pp. 50-57, Jan. 2004.

[4] X. Tang and X. Wang, "Face Sketch Synthesis and Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[5] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, "A Nonlinear Approach for Face Sketch Synthesis and Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2005.

[6] J. Zhong, X. Gao, and C. Tian, "Face Sketch Synthesis Using a E-Hmm and Selective Ensemble," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* 2007.

[7] X. Gao, J. Zhong, and C. Tian, "Sketch Synthesis Algorithm Based on E-Hmm and Selective Ensemble," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 18, no. 4, pp. 487-496, Apr. 2008.

[8] H. Koshimizu, M. Tominaga, T. Fujiwara, and K. Murakami, "On Kansei Facial Processing for Computerized Facial Caricaturing System Picasso," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* 1999.

[9] S. Iwashita, Y. Takeda, and T. Onisawa, "Expressive Facial Caricature Drawing," *Proc. IEEE Int'l Conf. Fuzzy Systems,* 1999.

[10] J. Benson and D.I. Perrett, "Perception and Recognition of Photographic Quality Facial Caricatures: Implications for the Recognition of Natural Images," *European J. Cognitive Psychology,* vol. 3, pp. 105-135, 1991.

[11] V. Bruce, E. Hanna, N. Dench, P. Healy, and A.M. Burton, "The Importance of Mass in Line Drawings of Faces," *Applied Cognitive Psychology,* vol. 6, pp. 619-628, 1992.

[12] V. Bruce and G.W. Humphreys, "Recognizing Objects and Faces," *Visual Cognition,* vol. 1, pp. 141-180, 1994.

[13] G.M. Davies, H.D. Ellis, and J.W. Shepherd, "Face Recognition Accuracy As a Function of Mode of Representation," *J. Applied Psychology,* vol. 63, pp. 180-187, 1978.

[14] G. Rhodes and T. Tremewan, "Understanding Face Recognition: Caricature Effects, Inversion, and the Homogeneity Problem," *Visual Cognition,* vol. 1, pp. 275-311, 1994.

[15] W.T. Freeman, J.B. Tenenbaum, and E. Pasztor, "An Example-Based Approach to Style Translation for Line Drawings," technical report, MERL, 1999.

[16] H. Chen, Y. Xu, H. Shum, S. Zhu, and N. Zheng, "Example-Based Facial Sketch Generation with Non-Parametric Sampling," *Proc. IEEE Int'l Conf. Computer Vision,* 2001.

[17] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Model," *Proc. European Conf. Computer Vision,* 1998.

[18] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael, "Learning Low-Level Vision," *Int'l J. Computer Vision,* vol. 40, pp. 25-47, 2000.

[19] P. Viola and M. Jones, "Real-Time Object Detection," *Int'l J. Computer Vision,* vol. 52, pp. 137-154, 2004.

[20] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Understanding Belief Propagation and Its Generalizations," *Exploring Artificial Intelligence in the New Millennium,* Morgan Kaufmann, 2003.

[21] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization Via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[22] A.A. Efros and W.T. Freeman, "Quilting for Texture Synthesis and Transfer," *Proc. ACM Conf. Computer Graphics and Interactive Techniques,* 2001.

[23] A.M. Martinez and R. Benavente, "The AR Face Database," Technical Report 24, CVC, 1998.

[24] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "Xm2vtsdb: The Extended of m2vts Database," *Proc. Int'l Conf. Audio- and Video-Based Person Authentication,* 1999.

[25] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, pp. 71-86, 1991.

[26] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 696-710, July 1997.

[27] P.N. Belhumeur, J. Hespanda, and D. Kiregeman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.

[28] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, "A New Lda Based Face Recognition System which Can Solve the Small Sample Size Problem," *Pattern Recognition,* vol. 33, pp. 1713-1726, 2000.

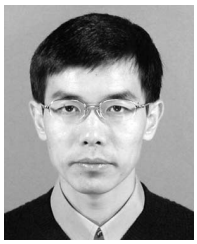
[29] X. Wang and X. Tang, "Dual-Space Linear Discriminant Analysis for Face Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2004.

[30] X. Wang and X. Tang, "Random Sampling Lda for Face Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2004.

[31] X. Wang and X. Tang, "Random Sampling for Subspace Face Recognition," *Int'l J. Computer Vision,* vol. 70, pp. 91-104, 2006.

[32] L. Wiskott, J. Fellous, N. Kruger, and C. Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 775-779, July 1997.

Xiaogang Wang received the BS degree in electrical engineering and information science from the University of Science and Technology of China in 2001, the MS degree in information engineering from Chinese University of Hong Kong in 2003, and the PhD degree from the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology in 2009. He is currently an assistant professor in the Department of Electronic Engineering at The Chinese University of Hong Kong. His research interests include computer vision and machine learning.



Xiaoou Tang received the BS degree from the University of Science and Technology of China, Hefei, in 1990, the MS degree from the University of Rochester, Rochester, New York, in 1991, and the PhD degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor in the Department of Information Engineering at the Chinese University of Hong Kong. He was the group manager of the Visual Computing Group at Microsoft Research Asia from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing. He received the Best Paper Award from the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2009. He is a program chair of the IEEE International Conference on Computer Vision 2009 and an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *International Journal of Computer Vision*. He is a fellow of the IEEE and a member of the IEEE Computer Society.