

Random Sampling for Fast Face Sketch Synthesis

Nannan Wang, and Xinbo Gao, and Jie Li

Abstract

Exemplar-based face sketch synthesis plays an important role in both digital entertainment and law enforcement. It generally consists of two parts: neighbor selection and reconstruction weight representation. The most time-consuming or main computation complexity for exemplar-based face sketch synthesis methods lies in the neighbor selection process. State-of-the-art face sketch synthesis methods perform neighbor selection online in a data-driven manner by K nearest neighbor (K -NN) searching. Actually, the online search increases the time consuming for synthesis. Moreover, since these methods need to traverse the whole training dataset for neighbor selection, the computational complexity increases with the scale of the training database and hence these methods have limited scalability. In this paper, we proposed a simple but effective offline random sampling in place of online K -NN search to improve the synthesis efficiency. Extensive experiments on public face sketch databases demonstrate the superiority of the proposed method in comparison to state-of-the-art methods, in terms of both synthesis quality and time consumption. The proposed method could be extended to other heterogeneous face image transformation problems such as face hallucination. We release the source codes of our proposed methods and the evaluation metrics for future study online: <http://www.ihitworld.com/RSLCR.html>.

Index Terms

Face sketch synthesis, locality constraint, neighbor selection, random sampling, weight computation

N. Wang is with the State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an 710071, Shaanxi, P. R. China (e-mail: nnwang@xidian.edu.cn).

X. Gao and J. Li are with the State Key Laboratory of Integrated Services Networks, School of Electronic Engineering, Xidian University, Xi'an 710071, Shaanxi, P. R. China (e-mail: xbgao@mail.xidian.edu.cn;leejie@mail.xidian.edu.cn).

I. INTRODUCTION

Face sketch synthesis mainly refers to generating a sketch given one input photo and some face sketch-photo pairs as the training dataset. It has achieved wide applications in both digital entertainment and law enforcement [1]. For example, since limited information about the suspect is available due to the low quality of surveillance videos or even no video/image clues, a sketch drawn by the artist is usually taken as the substitute for suspect identification. Then, face sketch synthesis bridges the great texture discrepancy between face photos and sketches.

Exemplar-based face sketch synthesis generally proceeds in two steps: neighbor selection and reconstruction weight representation. Given an input test photo, it is divided into some patches with even size and adjacent patches have some overlap to guarantee the compatibility. Then for each test patch, some number (*e.g.* K) of nearest photo patches are selected from the training photos. Sketch patches corresponding to these nearest photo patches are taken as the candidates for sketch patch synthesis. The prevalent way to represent the target sketch patch is the linear combination of selected candidate sketch patches. The linear combination coefficients are usually calculated under the assumption that a photo patch and its corresponding sketch patch share similar geometric manifold structure, *i.e.* if two photo patches are similar, then their sketch patch counterparts are also similar.

Exemplar-based face sketch synthesis started from the work of Eigen-transformation of Tang and Wang [2], [3]. In their work, there is no special neighbor selection process but all training images are utilized. The linear combination coefficients are learned by projecting the input photo onto the training photos through principal component analysis.

Considering that only learning one holistic reconstruction model is difficult to represent the nonlinear mapping between face photos and sketches, Liu *et al.* [4] proposed to estimate the holistic nonlinear mapping relationship with many piece-wise linear mappings, which are generally followed in subsequent methods. This method works on the image patch level. K nearest photo patches are searched from the training set in terms of Euclidean distance. Then the reconstruction weight is calculated in the spirit of locally linear embedding [5]:

$$\min_{\mathbf{w}} \|\mathbf{x} - \mathbf{X}\mathbf{w}\|_2^2, s.t. \sum_{i=1}^K w_i = 1 \quad (1)$$

where $\mathbf{w} = (w_1, w_2, \dots, w_K)^T$ is the representation weight vector, \mathbf{x} is the test photo patch in the form of column vector and \mathbf{X} is the matrix of column-concatenation of K selected training photo patches. The target sketch patch corresponding to the test photo patch is reconstructed from the linear combination of K training sketch patches weighted by \mathbf{w} . Song *et al.* [6] casted the face sketch synthesis problem into

a spatial sketch denoising (SSD) problem and calculated the reconstruction weight through conjugate gradient solver. Gao *et al.* [7] proposed to adaptively determine the number of nearest neighbors by sparse representation [8] rather than the fixed number (*e.g.* K) of nearest neighbors. Instead of using sparse representation for neighbor selection, some dictionaries are learned through sparse coding and sparse representation to substitute the role of nearest neighbors in the work [9].

Wang *et al.* [10] employed Markov random field (MRF) to model the dependency from two aspects: the dependency between test photo patches and nearest photo patches and the dependency between adjacent synthesized sketch patches which are neglected in the above methods. In their method, K nearest photo patches and their corresponding sketch patches are selected from the training dataset. Only one single nearest sketch patch is finally selected through MRF networks which is taken as the target synthesized sketch patch. In other words, the weight reconstruction representation for this method can be seemed as finding one most appropriate sketch patch and its weight is set to 1.

Zhou *et al.* [11] proposed to introduce the linear combination into the MRF model (namely Markov weight field, MWF) to overcome the face deformation problem due to single sketch patch search strategy in [10]. The difference between the MWF method and the LLE method [4] is the consideration of the dependency between adjacent synthesized sketch patches as follows:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|\mathbf{x} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda \|\mathbf{O}^i \mathbf{w} - \mathbf{O}^{(i)} \mathbf{w}^i\|, \\ \text{s.t.} \quad & \sum_{l=1}^K w_l = 1, w_l \geq 0, \end{aligned} \quad (2)$$

where the second term $\sum_{i=1}^4 \|\mathbf{O}^i \mathbf{w} - \mathbf{O}^{(i)} \mathbf{w}^i\|$ represents the dependency constraint between the synthesized sketch corresponding to the test photo patch \mathbf{x} and its four adjacent synthesized sketch patches. Here the constraint is modeled by the distance of pixel intensity vector extracted from the overlapping area between adjacent sketch patches. In equation (2), the column vector \mathbf{w}^i is the reconstruction weight corresponding to the i -th adjacent sketch, $\mathbf{O}^{(i)}$ denotes the pixel intensity vector extracted from the overlapping area of adjacent sketch patch i , and \mathbf{O}^i denotes the pixel intensity vector extracted from the overlapping area of current target sketch patch. Wang *et al.* [12] further developed the MWF model from the perspective of transductive learning. Peng *et al.* [13] extended the MWF model to a multi-view version which improves the robustness against the cluttered background and lighting variations. Unlike the even patch employed in aforementioned methods, super-pixel segmentation of image patches is employed and the reconstruction representation model as in equation (2) is adopted in the work [14].

All aforementioned methods perform K nearest neighbor (K -NN) selection online, which heavily

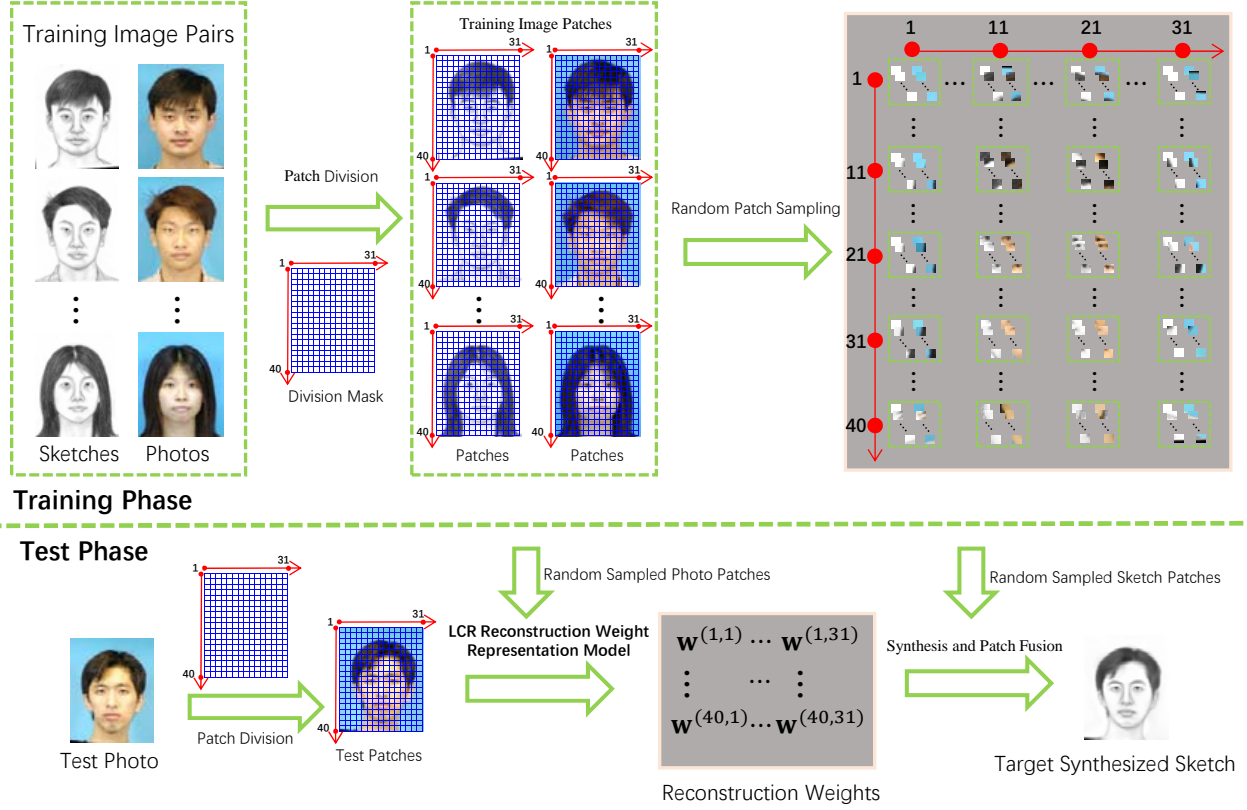


Fig. 1. A graphical illustration of the proposed RSLCR framework.

increases the time consuming for test. Moreover, with the increase of the scale of the database, the computation complexity would also increase linearly. In addition, the reconstruction weight representation model either in (1) or in (2) consider that all selected nearest neighbors contribute equally to the reconstruction weight computation process, while the distinct distance between these neighbors and the test patch are neglected.

In this paper, instead of online searching neighbors, we randomly sample some patches offline and then these patches are used to reconstruction the target sketch patch. This random sampling strategy greatly speeds up the synthesis process, which is much faster than K -NN based methods (*e.g.* the LLE method [4]) under the same experimental settings. In addition, state-of-the-art methods consider that all selected neighbors contribute equally to the reconstruction weight computation process while the distinct similarity between the test patch and these neighbors are neglected. Since these random sampled patches have distinct similarities with the test photo patch, we impose the locality constraint [15] to regularize their

corresponding reconstruction weights. The locality constraint would restrain the contribution of patches which distribute far from the test patch and excite the contribution of patches which distribute around the test patch. Similar techniques appeared in image restoration tasks such as image super-resolution [16], [17] and image denoising [18]. To further accelerate the synthesis process, we employ principal component analysis (PCA) [19] to reduce the dimension of each patch vector. A graphical outline of the proposed random sampling with locality constraint for face sketch synthesis method (RSLCR) is shown in Fig. 1. In addition, we proposed a fast version of the proposed method by dropping out some random sampled patches, namely Fast-RSLCR.

The contributions of this paper are twofold. Firstly, an offline random sampling strategy is employed to reduce the time consuming for online neighbor selection. In addition, the proposed strategy has stronger scalability than state-of-the-art methods due to the fact that the time-consuming does not depend on the scale of training dataset for our proposed strategy while not the case for other methods. We further imposed locality constraint to the reconstruction weight representation which takes the distinct similarities between the test patch and random sampled patches into consideration. This improves the quality of synthesized sketches. **Secondly, both our proposed RSLCR method and its fast version Fast-RSLCR achieve superior performance than state-of-the-art methods in terms of both synthesis performance and synthesis efficiency.** Specially, our proposed Fast-RSLCR could synthesize a sketch using no more than 1.5 seconds on the Chinese University of Hong Kong (CUHK) face sketch FERET database (CUFSF) under the MATLAB environment, which is the fastest exemplar-based face sketch synthesis method.

In this paper, excepted when noted, a bold lowercase letter represents a column vector, a bold uppercase letter denotes a matrix and regular lowercase and uppercase letters denotes scalars. The rest of this paper is organized as follows. Section II introduces the proposed RSLCR method and Fast-RSLCR method. Experimental results and analysis are given in section III and section IV concludes this paper.

II. RANDOM SAMPLING FOR FACE SKETCH SYNTHESIS

In this section, we would introduce how to sample "neighbors" in an offline manner, *i.e.* random sampling training image patches, and then how to represent the test photo patch using these random sampled training photo patches, *i.e.* locality constraint (LCR) based weight representation model.

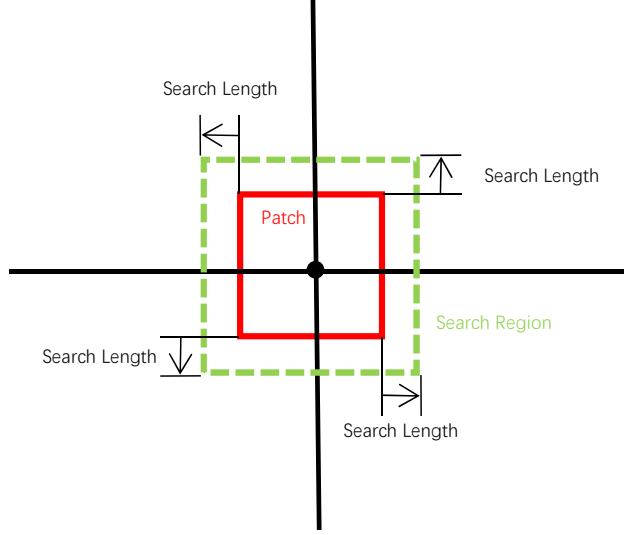


Fig. 2. Illustration of search region

A. Random Sampling Image Patches

Supposing there are M pairs of training sketches and training photos which are geometrically aligned according to three points: two eye centers and the mouth center. Each image is cropped to the size of 250×200 . We first divide these photos and sketches into some patches with even size. There is some overlapping (denoted as o) between adjacent patches. As shown in Fig. 1, each image is divided into $N = 40 \times 31$ patches where there are $r = 40$ patches in each column and $c = 31$ patches in each row (the patch size p is set to 20 with $o = 14$ pixels overlapped between adjacent patches). We reshape each image patch as a column vector. (i, j) denotes the location of the patch at the i -th row and the j -th column, $i \in \{1, \dots, r\}, j \in \{1, \dots, c\}$.

Our target is to generate N clusters of photo-sketch patch pairs corresponding to the N locations $\{(i, j) | i \in \{1, \dots, r\}, j \in \{1, \dots, c\}\}$. The most intuitive way is to put patches located at the same position together. However, since images are aligned relying on only three points, there exist misalignments between test photos and training photos, which may result in mismatch during the reconstruction process. To alleviate the influence of misalignment, we enlarge the sampling area to allow more candidate patches to be sampled as shown in Fig. 2. Let s denote the search length and then there are $(2s + 1)^2$ patches in the search region. Therefore, for each location, we have $(2s + 1)^2 M$ pairs of patches for sampling. Let n_{rs} denote the number of random sampled patches. In our implementations, we employ the MATLAB function *randperm()* to sample training sketch-photo patch pairs. $\mathbf{X}^{(i,j)} \in \mathbb{R}^{3p^2 \times n_{rs}}$ and

Algorithm 1 Random Sampling Image Patches

Input: p, o, s, n_{rs}

Step 1: According to the patch size p and overlap size o , compute

r, c , and the positions of all patches in an image;

Step 2: Within the search region of each patch position (i, j) ,

$i \in \{1, \dots, r\}, j \in \{1, \dots, c\}$, random sample n_{rs} pairs
of photo patches $\mathbf{X}^{(i,j)}$ and sketch patches $\mathbf{Y}^{(i,j)}$ in all
training image pairs;

Step 3: Compute the PCA projection matrix $\mathbf{E}^{(i,j)}$ for each cluster

of training photo patches and project the training photo
patches to the subspace spanned by $\mathbf{E}^{(i,j)}$ as in equation

(3).

Output: $\mathbf{X}^{(i,j)}, \mathbf{Y}^{(i,j)}, \mathbf{E}^{(i,j)}, i \in \{1, \dots, r\}, j \in \{1, \dots, c\}$.

$\mathbf{Y}^{(i,j)} \in \mathbb{R}^{p^2 \times n_{rs}}$ denote the sampled training photo patches and sketch patches in (i, j) -th cluster respectively, $i \in \{1, \dots, r\}, j \in \{1, \dots, c\}$.

In order to improve the computation efficiency, we employ PCA to reduce the dimension of training photo patches. 99% energy is preserved in the projection process. Let $\mathbf{E}^{(i,j)} \in \mathbb{R}^{3p^2 \times D^{(i,j)}}$ represent the projection matrix and $D^{(i,j)}$ is the reduced dimension. The training photo patches are projected onto the subspace spanned by the column vectors of $\mathbf{E}^{(i,j)}$:

$$\mathbf{X}'^{(i,j)} = \mathbf{E}^{(i,j)T} \mathbf{X}^{(i,j)} \quad (3)$$

where $\mathbf{X}'^{(i,j)} \in \mathbb{R}^{D^{(i,j)} \times n_{rs}}$ is the newly projected training photo patches. For easy of notation, we still use $\mathbf{X}^{(i,j)}$ to denote the projected training photo patches in the following text. Algorithm 1 summarizes the proposed random sampling method.

B. Reconstruction Weight Representation

Given a test photo $\mathbf{P} \in \mathbb{R}^{250 \times 200}$, it is divided into some patches $\mathbf{x}^{(i,j)} \in \mathbb{R}^{3p^2 \times 1}$ according to the same way for training images, $i \in \{1, \dots, r\}, j \in \{1, \dots, c\}$. These patches are projected to the respective subspace obtained in the training phase:

$$\mathbf{x}'^{(i,j)} = \mathbf{E}^{(i,j)T} \mathbf{x}^{(i,j)} \quad (4)$$

where $\mathbf{x}^{(i,j)} \in \mathbb{R}^{D^{(i,j)} \times 1}$ is the projected training photo patch and for easy of notation, we still use $\mathbf{x}^{(i,j)}$ to represent the test photo patch. In order to take the correlations between different random sampled patches into considerations, we impose a weight to the distances of the test photo patch and random sampled photo patches. Then the reconstruction weight representation model is written as follows:

$$\begin{aligned} \min_{\mathbf{w}^{(i,j)}} & \|\mathbf{x}^{(i,j)} - \mathbf{X}^{(i,j)} \mathbf{w}^{(i,j)}\|_2^2 + \lambda \|\mathbf{d}^{(i,j)} \odot \mathbf{w}^{(i,j)}\|, \\ s.t. & \mathbf{1}^T \mathbf{w}^{(i,j)} = 1, \forall i \in \{1, \dots, r\}, \forall j \in \{1, \dots, c\}, \end{aligned} \quad (5)$$

where \odot denotes the element-wise multiplication, $\mathbf{w}^{(i,j)} \in \mathbb{R}^{n_{rs} \times 1}$ is the weight representation for the test photo patch $\mathbf{x}^{(i,j)}$, λ balances the reconstruction error and the locality constraint, and $\mathbf{d}^{(i,j)} \in \mathbb{R}^{n_{rs} \times 1}$ is the Euclidean distance vector between the test photo patch $\mathbf{x}^{(i,j)}$ and sampled training photo patches $\mathbf{X}^{(i,j)}$. It can be derived that the problem (5) has analytical solution:

$$\begin{aligned} \mathbf{w}'^{(i,j)} &= (\mathbf{C}^{(i,j)} + \lambda \text{diag}(\mathbf{d}^{(i,j)})) \setminus \mathbf{1}, \\ \mathbf{w}^{(i,j)} &= \mathbf{w}'^{(i,j)} / \mathbf{1}^T \mathbf{w}'^{(i,j)}, \end{aligned} \quad (6)$$

where $\mathbf{1}$ is a column vector of all 1s and its dimension can be determined in the context. $\mathbf{C}^{(i,j)} = (\mathbf{X}^{(i,j)} - \mathbf{1}\mathbf{x}^{(i,j)T})(\mathbf{X}^{(i,j)} - \mathbf{1}\mathbf{x}^{(i,j)T})^T$ denotes the data covariance matrix and $\text{diag}(\mathbf{d})$ extends the vector \mathbf{d} into a diagonal matrix. The target sketch patch $\mathbf{y}^{(i,j)}$ is generated from the linear combination of random sampled training sketches weight by the obtained representation vector $\mathbf{w}^{(i,j)}$:

$$\mathbf{y}^{(i,j)} = \mathbf{Y}^{(i,j)} \mathbf{w}^{(i,j)}. \quad (7)$$

After obtaining all target sketch patches, they are arranged into a whole sketch with overlapping area averaged.

Since the computation complexity in equation (6) mainly depends on the number of random sampled patches, we could further accelerate the proposed RSLCR method by dropping out some random sampled patches in the training phase. Actually we have already computed the distance between the test photo patch and the random sampled training photo patches, we could drop out sampled patches whose distance to the test photo patch are larger. In other words, we could retain K sampled patches whose distance to the test photo patch are among the first K smallest. In comparison to equation (6), we only need to update the data matrix $\mathbf{X}^{(i,j)}$ with its subset, i.e. $\mathbf{X}_s^{(i,j)} = \mathbf{X}^{(i,j)}(:, \text{idx}(1 : K))$ and $\mathbf{Y}_s^{(i,j)} = \mathbf{Y}^{(i,j)}(:, \text{idx}(1 : K))$ where idx stores the index to distances between the test photo patch and n_{rs} sampled training photo patches in an ascending order. We call this fast version as Fast-RSLCR. We summarize the proposed RSLCR and Fast-RSLCR algorithm in Algorithm 2.

Algorithm 2 RSLCR & Fast-RSLCR

Input: \mathbf{P} , p , o , K

Step 1: According to the patch size p and overlap size o , divide

\mathbf{P} into patches $\mathbf{x}^{(i,j)}, i \in \{1, \dots, r\}, j \in \{1, \dots, c\}$;

Step 2: For $i \in \{1, \dots, r\}$

For $j \in \{1, \dots, c\}$

Step 3 RSLCR: Compute $\mathbf{C}^{(i,j)}$ with $\mathbf{X}^{(i,j)}$;

Fast-RSLCR: Compute $\mathbf{C}^{(i,j)}$ with $\mathbf{X}_s^{(i,j)}$;

Step 4: Compute $\mathbf{w}^{(i,j)}$ as in equation (6);

Step 5: RSLCR: $\mathbf{y}^{(i,j)} = \mathbf{Y}^{(i,j)} \mathbf{w}^{(i,j)}$;

Fast-RSLCR: $\mathbf{y}^{(i,j)} = \mathbf{Y}_s^{(i,j)} \mathbf{w}^{(i,j)}$;

Step 6: Arrange all target sketch patches into a whole sketch with overlapping area averaged.

Output: the target sketch \mathbf{S} .

III. EXPERIMENTAL RESULTS AND ANALYSIS

Experimental results are conducted to illustrate the efficiency and effectiveness of the proposed RSLCR method and Fast-RSLCR method. Two public available database are used: the CUHK face sketch database (CUFS) [10] and the CUFSF database [20]. The CUFS database consists of face photos from three databases: the CUHK student database [21] (188 persons), the AR database [22] (123 persons) and the XM2VTS database [23] (295 persons). Persons in the XM2VTS database are different in ages, skins (races) and hair styles. The CUFSF database includes 1194 persons from the FERET database [24]. There are one face photos and one face sketch drawn by the artist for each person in both CUFS and CUFSF databases. Face photos in the CUFSF database are with lighting variation and sketches are with shape exaggeration. All these face photos and sketches are geometrically aligned relying on three points: two eye centers and the mouth center and they are cropped to the size of 250×200 . Fig. 3 gives some examples from these two databases.

In the following context, we would first discuss the experimental settings (parameter settings) for our proposed method on the CUHK student database. Afterwards, under the experimental settings, we perform face sketch synthesis on the CUFS database and the CUFSF database to subjectively illustrate the superiority of the proposed RSLCR method and the Fast-RSLCR method compared with state-of-



Fig. 3. Example face sketch-photo pairs in the CUFS database (the first two rows) and the CUFSF database (the last two rows). The first and the third row are face photos and the second and the last rows are corresponding face sketches drawn by the artist. The first person, second person and the last three persons at the first two rows are from the CUHK student database, the AR database and the XM2VTS database respectively.

the-arts. Then time consumption has discussed. Subsequently, objective statistic experiments (objective image quality assessment and face recognition) are conducted to indirectly validate the superiority of proposed methods.

A. Discussion on Experimental Settings

We employ the CUHK student database [21] to perform parameter adjusting in this sub-section. 88 pairs of face photo-sketch are taken as the training set and the rest 100 pairs of face photo-sketch are taken for validation. To objectively assess the quality of synthesized sketches under different experimental settings, structural similarity index metric (SSIM) [25] is adopted as the evaluation criterion. The 100 sketches drawn by the artist in the validation set are taken as the reference image and 100 photos in the validation set are taken as the test image for face sketch synthesis. Under each experimental setting, the average SSIM score of 100 synthesized sketches are taken as the final evaluation value.

There are five parameters (*i.e.* patch size p , overlap size o , search length s , the number of random sampled patches $r_{n.s}$ and the trade-off parameter λ) for the proposed RSLCR method and one additional

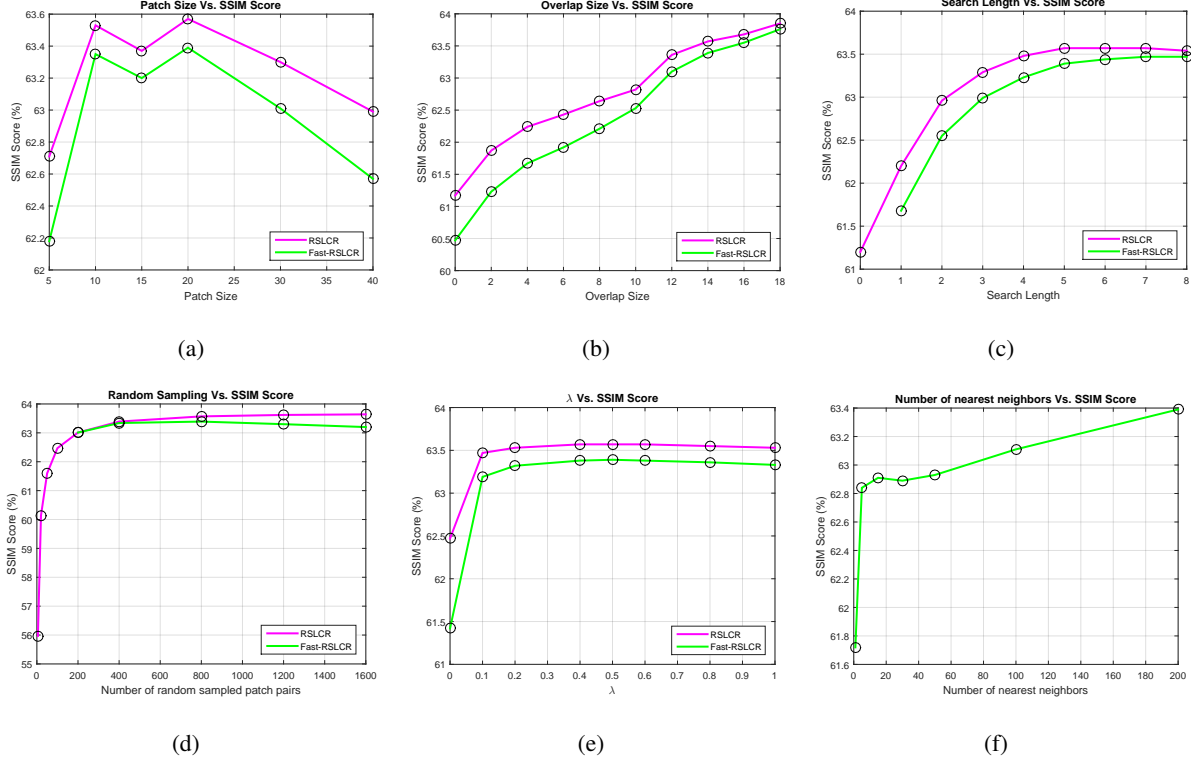


Fig. 4. Statistics of SSIM scores under different parameter settings: (a) patch size, (b) overlap size, (c) search length, (d) number of random sampled patch pairs, (e) λ , (f) number of nearest neighbors for Fast-RSLCR.

parameter (*i.e.* the number of selected neighbors) for Fast-RSLCR. All experiments are conducted using MATLAB R2015a on Windows 7 system with i7-4790 3.6G CPU. Fig. 4 presents the SSIM scores against different parameter settings.

1) *Patch Size*: We set patch size to 5, 10, 15, 20, 30 and 40 respectively and keep the overlap size as 70% of the patch size. It can be seen from Fig. 4(a) that both RSLCR and Fast-RSLCR achieves the highest SSIM score when the patch size is 20. When the patch size is 10, it has very close performance with the patch size 20. However, smaller patch size means more patches to be synthesized and hence it consumes much more time (78.84 seconds vs. 18.79 seconds for RSLCR and 21.07 seconds vs. 1.82 seconds for Fast-RSLCR).

2) *Overlap Size*: Given the patch size being 20, we set the overlap size to different values: 0, 2, 4, 6, 8, 10, 12, 14, 16 and 18. From Fig. 4(b) it can be seen that with the increase of the overlap size, SSIM scores for both RSLCR and Fast-RSLCR are also increasing. However, the time-consuming is also growing rapidly. In our following experiments, we set the overlap size as the trade-off value 14.

3) *Search Length*: The search length is used in the training stage and hence it does not affect the test phase. From Fig. 4(c) it can be seen that though the SSIM score of Fast-RSLCR always grows with the increase of the search length, the SSIM score of RSLCR grows at first and then goes down. This is because with the increase of the search length, some outliers may be sampled and these outliers would bring noise to the RSLCR method. However, Fast-RSLCR does not subject to these outliers because it selects the K nearest neighbors among random sampled patches which could filter these outliers. The search length is set to 5 in our experiments.

4) *Random Sampling*: Fig. 4(d) presents the SSIM score corresponding to different number of random sampled training face photo-sketch patches. It can be seen that generally the SSIM score grows with the increase of random sampling number for the RSLCR method. However, it begins to decrease when the number is bigger than 800 for the Fast-RSLCR method. We set it to 800 in our experiments.

5) *Regularization Parameter*: λ It should be noted that when $\lambda = 0$ (refer to equation (5)), the locality constraint has no contribution to the reconstruction weight computation. Then equation (5) reduces to the LLE model in equation (1). The difference is that the entries of the data matrix \mathbf{X} for the LLE method [4] is selected through K -NN while they are random sampled for RSLCR and Fast-RSLCR. From Fig. 4(e), on the one hand, it can be seen that when $\lambda = 0$, SSIM scores for RSLCR and Fast-RSLCR are 0.6250 and 0.6150 respectively compared with 0.5990 of the LLE method [4]. In addition, RSLCR and Fast-RSLCR run much faster than the LLE method (18.79 seconds for RSLCR, 1.82 seconds for Fast-RSLCR, and 536.34 seconds for LLE). This illustrates the effectiveness of the proposed random sampling strategy. On the other hand, when $\lambda = 0.5$, RSLCR and Fast-RSLCR achieve the best performance among all λ values, which is much larger than SSIM socres for $\lambda = 0$. This validates that the locality constraint does help to improve the performance. λ is set to 0.5 in this paper.

6) *Number of Nearest Neighbors for Fast-RSLCR*: K -NN is conducted in Fast-RSLCR to improve the computation efficiency. Fig. 4(f) presents the SSIM score against different number of nearest neighbors. Generally it grows with the increase of the number. To comprehensively comprise the time consuming and the SSIM socre, it is set to 200 in our experiments. From Fig. 4(d) it can be seen that directly random sampling 200 patches achieves an SSIM score of 0.6301 (at the time cost of 1.82s) while our proposed Fast-RSLCR (also use 200 sampled patches) could achieve an SSIM score of 0.6339 (at the time cost of 1.89s). It demonstrates the effectiveness of the proposed Fast-RSLCR method.

7) *Random Sampling Searching Vs. Accelerated Nearest Neighbor Searching*: In order to illustrate the effectiveness of the proposed the offline random sampling searching strategy in comparison to accelerated nearest neighbor (ANN) searching strategies, we utilize two kinds of ANN approaches for online neighbor

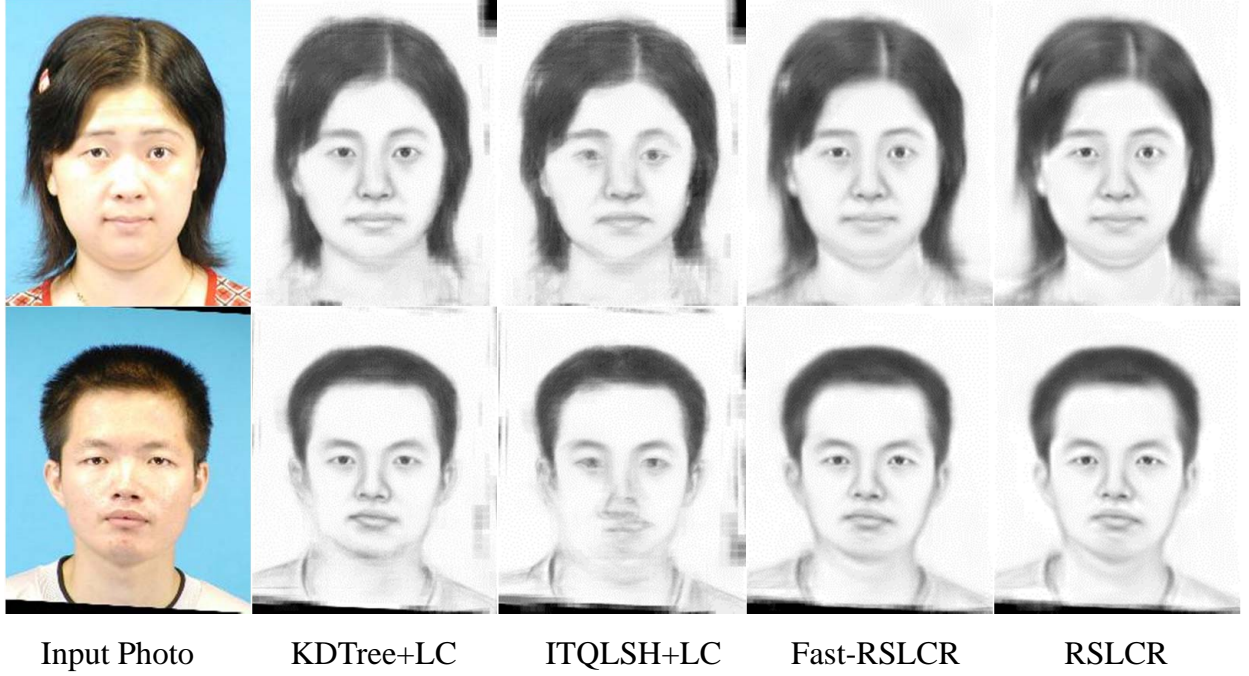


Fig. 5. Synthesized sketches by KDTree-LC, ITQLSH-LC, Fast-RSLCR, and RSLCR method respectively.

searching: iterative quantization based locality-sensitive hashing (ITQLSH)¹ [26] and the KDTree method² [27]. We substitute the random sampling strategy in our proposed method with these two KNN searching method and they are denoted as ITQLSH-LC and KDTree-LC respectively. Table I gives the effect of the number of nearest neighbors on these two methods. It can be seen that when the number of neighbors is 20, these two methods achieve the best performance. Table II shows the comparison between the proposed methods with these two ANN based methods. Fig. 5 presents the synthesized sketches by four different methods. Table II and Fig. 5 illustrate that the random sampling strategy outperforms ITQLSH and KDTree in terms of both time consumption and image quality.

8) *Locality Constraint (LC)*: Locality constraint is firstly proposed for image classification and it shows comparable performance with sparse constraint [15]. The non-local similarity constraint as

¹The MATLAB/C++ mixed source codes are download from the website: <https://github.com/RSIA-LIESMARS-WHU/LSHBOX>

²We use the open source implementation (MATLAB/C++ mixed programming) of this method in VLFeat: <http://www.vlfeat.org/>

TABLE I
THE EFFECT OF THE NUMBER OF NEAREST NEIGHBORS ON ITQLSH-LC AND KDTree IN TERMS OF SSIM SCORE (%)
AND TIME CONSUMPTION (THE VALUE IN THE BRACKET IS IN SECONDS)

Number of Neighbors	5	20	50	100	200	400	800
KDTree-LC	60.72 (434.89)	61.27 (436.18)	61.13 (437.03)	60.72 (439.77)	60.36 (462.63)	60.34 (474.85)	60.50 (501.57)
ITQLSH-LC	58.80 (412.16)	59.30 (421.09)	58.96 (432.16)	58.58 (437.24)	58.53 (439.78)	58.65 (443.14)	58.76 (461.01)

TABLE II
COMPARISONS BETWEEN TWO ANN BASED METHODS AND THE PROPOSED METHODS IN TERMS OF SSIM SCORE (%)
AND TIME CONSUMPTION (SECONDS)

Method	KDTree-LC	ITQLSH-LC	RSLCR	Fast-RSLCR
SSIM	61.27	59.30	63.57	63.39
Time	436.18	421.09	18.79	1.82

a regularization term in [9] is also very similar with the locality constraint. Table III compares the performance of different face sketch synthesis methods with or without locality constraint. It can be found that locality constraint improves the performance of our proposed Fast-RSLCR method a lot and it also improves the LLE method and the RSLCR method. However, this is not the case for the MWF method. This is because the locality constraint is implicitly embedded in the neighboring constraint of the MWF model (see the second term of equation (2)). In this paper, locality constraint is utilized to distinguish the random sampled patches since these patches may be distributed in a scattered way, *i.e.* it is especially appropriate for random sampling based methods where the random sampled patches may distribute far away from each other.

TABLE III
THE EFFECT OF LOCALITY CONSTRAINT (IN TERMS OF SSIM SCORE (%)) ON DIFFERENT FACE SKETCH SYNTHESIS METHODS

Method	LLE	MWF	Fast-RSLCR	RSLCR
With LC	61.00	62.31	63.39	63.57
Without LC	59.97	62.31	61.42	62.47



Fig. 6. Synthesized sketches on the CUFS database by FCN [28], GAN [29], SSD [6], MRF [10], LLE [4], MWF [11], the proposed Fast-RSLCR and RSLCR respectively. Face photos in the first two rows are from the CUHK student database and the AR database respectively. The last four photos are from the XM2VTS database.

B. Face Sketch Synthesis

After the experimental setting for parameters, we set the patch size $p = 20$, overlap size $o = 14$, search length $s = 5$, the number for random sampling $n_{rs} = 800$, the regularization parameter $\lambda = 0.5$, the number of nearest neighbors for Fast-RSLCR $K = 200$. For the CUHK student database, 88 pairs of face photo-sketch are taken for training and the rest for testing (the data has been partitioned in this database).

For the AR database, we randomly choose 80 pairs for training and the rest 43 pairs for testing. For the XM2VTS database, we randomly choose 100 pairs for training and the rest 195 pairs for testing. Six state-of-the-art methods are compared: the FCN method [28], the GAN method [29]³, the LLE method [4], the SSD method [6], the MRF method [10], and the MWF method [11]. All synthesized sketches by the SSD method and the MWF method are generated from the source codes provided by the authors. For the MRF method, we use the codes from the implementation provided by authors of SSD [6]⁴. Results of the LLE method and the FCN method are based on our implementations⁵. The full list of synthesized sketches (both of our methods and all four compared methods) on these two databases is available on our project website.

Fig. 6 presents some synthesized face sketches from different methods on the CUFS database. It can be seen that the proposed RSLCR method and the Fast-RSLCR method could generate fine textures (*e.g.* hair region) and structures (*e.g.* glasses). This is because more candidate patches in our proposed methods are effectively incorporated through random sampling and locality constraint. Synthesized sketches on photos from the XM2VTS database generated by SSD, MRF, LLE, and MWF are less satisfying than photos from the CUHK student database and the AR database due to the fact that there are more variations such as aging, race, and hair styles on faces of the XM2VTS database. However, RSLCR and Fast-RSLCR achieve much better performance than these four compared methods and afford comparable performance on face photos from three different databases. This illustrates the robustness of the proposed methods.

We have also investigated the robustness of the proposed methods against shape exaggeration and illumination variations on the CUFSF database. We randomly choose 250 face photo-sketch pairs for training and the rest 944 pairs for test. Fig. 7 shows the synthesized results on this database by various methods. It is shown that there are some deformations on synthesized sketches by SSD and MRF, specially for the mouth area. In addition, our proposed Fast-RSLCR and RSLCR method could handle glasses with reflect light well while other methods cannot (see the third row of Fig. 7).

³ The source codes are available online: <https://github.com/phillipi/pix2pix>

⁴ The source codes for both the MRF method and the SSD method are available online: <http://www.cs.cityu.edu.hk/~yibisong/eccv14/index.html>

⁵ Available online: <http://www.ihitworld.com/RSLCR.html>. On this project website, we also release the source codes of both our proposed methods and the evaluation codes (objective image quality assessment codes and face recognition codes).

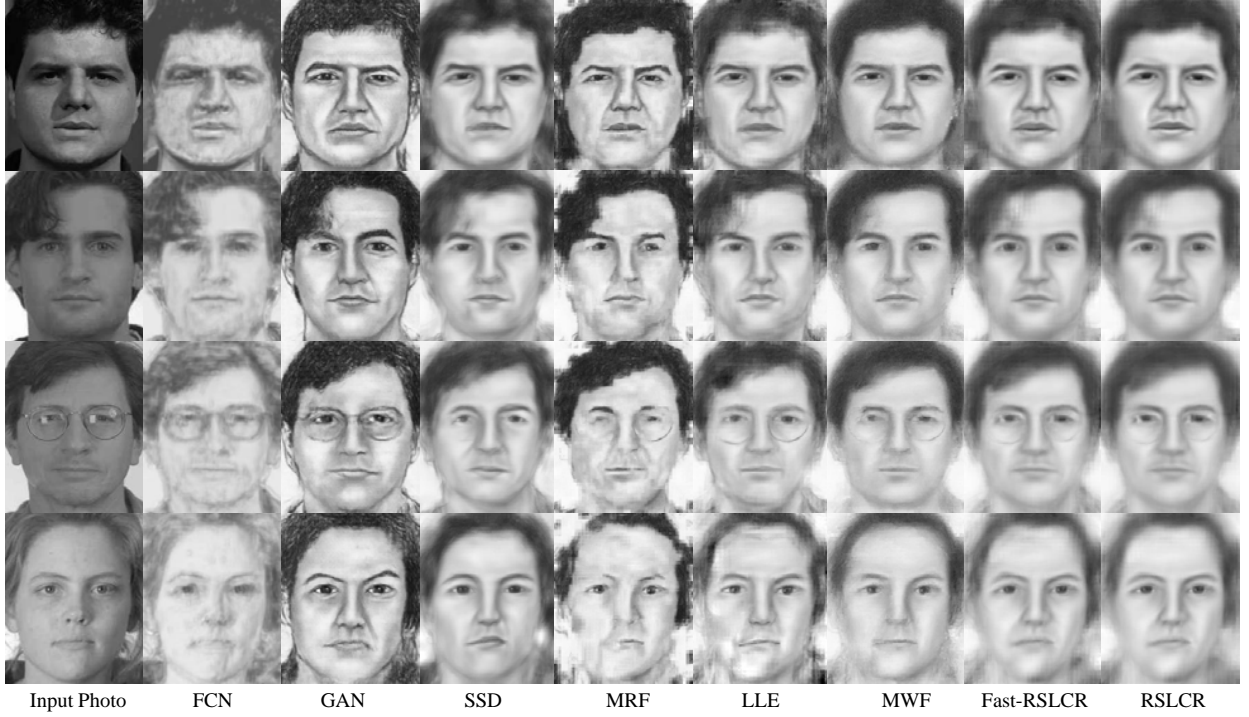


Fig. 7. Synthesized sketches on the CUFSF database by FCN [28], GAN [29], SSD [6], MRF [10], LLE [4], MWF [11], the proposed Fast-RSLCR and RSLCR respectively.

C. Time Consumption

Given the experimental settings in III-B, we count the time consumption for the proposed methods. Table IV compares the time consuming for different methods on different databases. There are 88, 80, 100, and 250 training photo-sketch pairs for the CUHK student, AR, XM2VTS and FERET database respectively. It can be seen from the table that time consumptions for SSD, MRF, LLE, and MWF are proportional to the scale of the training set because these methods search neighbors by traversing the whole training dataset. However, our proposed RSLCR and Fast-RSLCR method costs comparable time on these four databases. This validates the stronger scalability of the proposed RSLCR framework. Moreover, it can be seen that RSLCR has comparable or even less time consuming compared with state-of-the-art methods. Our proposed Fast-RSLCR is the most efficient method among all methods. It requires less than 1.5 seconds to synthesize a sketch on the CUHK FERET database, which is dozens of times faster than state-of-the-art methods.

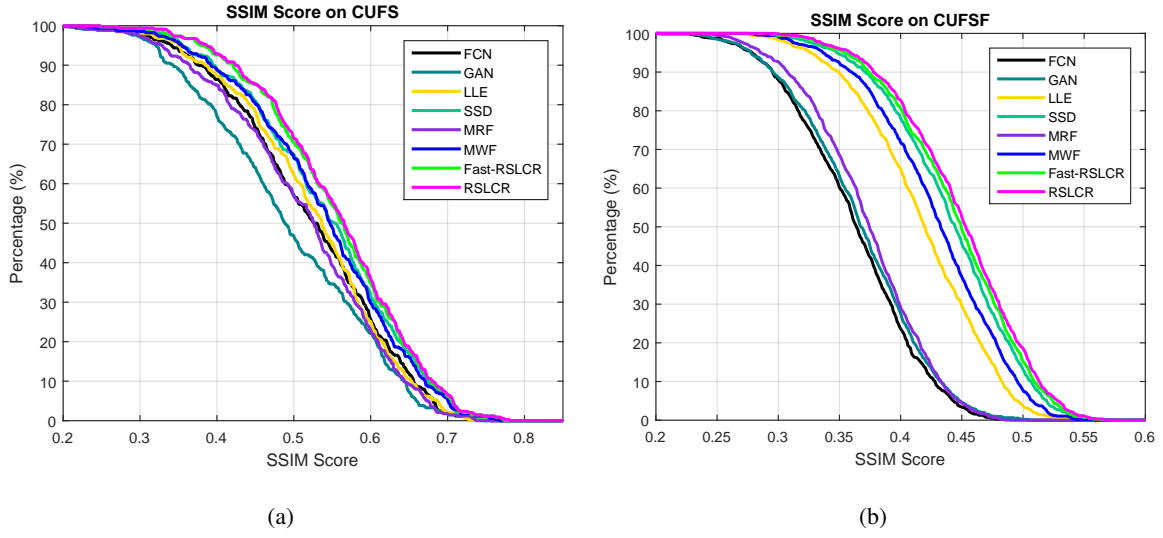


Fig. 8. Statistics of SSIM scores on (a) the CUFS database and (b) the CUFSF database.

TABLE IV
AVERAGE TIME CONSUMPTION (SECONDS) TO GENERATE ONE SKETCH ON DIFFERENT DATABASES

Methods	SSD	MRF	LLE	MWF	RSLCR	Fast-RSLCR
Programming language	C++	C++	MATLAB	C++	MATLAB	MATLAB
CUHK Student	4.50	8.60	536.34	16.10	18.79	1.82
AR	4.10	8.40	496.47	15.33	19.10	1.73
XM2VTS	5.10	10.4	642.50	18.80	18.14	2.36
CUHK FERET	11.60	24.25	1591.95	45.20	17.66	1.44

TABLE V
AVERAGE SSIM SCORE (%) ON THE CUFS DATABASE AND THE CUFSF DATABASE

	FCN [28]	GAN [29]	SSD[6]	MRF[10]	LLE[4]	MWF[11]	Fast-RSLCR	RSLCR
CUFS (%)	52.14	49.39	54.20	51.32	52.58	53.93	55.42	55.72
CUFSF (%)	36.22	36.65	44.09	37.34	41.76	42.99	44.56	44.96

TABLE VI
NLDA FACE RECOGNITION ACCURACY (%) BASED ON SYNTHESIZED SKETCHES FROM THE CUFS DATABASE AND THE CUFSF DATABASE

Methods	FCN	GAN	LLE	SSD	MRF	MWF	Fast-RSLCR	RSLCR
CUFS (%)	96.49 (131)	93.48 (139)	91.12 (148)	90.24 (149)	87.29 (149)	92.13 (149)	98.35 (121)	98.38 (133)
CUFSF (%)	69.80 (237)	71.44 (164)	61.76 (274)	70.92 (266)	46.03 (223)	74.15 (299)	73.41 (287)	75.94 (296)

D. Objective Image Quality Assessment

We utilize SSIM to evaluate the quality of synthesized sketches by different methods on CUFS and CUFSF. There are 338 ($100 + 43 + 195$) and 944 synthesized sketches for each method generated from the CUFS database and CUFSF database respectively. Fig. 8 gives the statistics of SSIM scores on these two databases respectively. The horizontal axis labels the SSIM score from 0 to 1. The vertical axis means the percentage of synthesized sketch whose SSIM scores are not smaller than the score marked on the horizontal axis. Table V presents the average SSIM score on the CUFS and CUFSF database respectively.

It can be seen from Fig. 8 and table V that both Fast-RSLCR and RSLCR outperform four other state-of-the-art methods. Comparable performance is achieved for SSD and MWF on the CUFS database but SSD outperforms MWF on the CUFSF database which illustrates SSD could handle faces with illumination variations better than the MWF method.

E. Face Sketch Recognition

Sketch based face recognition is always used to assist law enforcement. The sketch drawn by the artist is generally taken as the probe image and synthesized sketches play the role of images in the gallery. Null-space linear discriminant analysis (NLDA) [30] is employed to conduct the face recognition experiments. For the CUFS database, we randomly choose 150 synthesized sketches and corresponding ground-truth sketches drawn by the artist to train the classifier. The rest 188 sketches consists of the gallery. For the CUFSF database, we randomly choose 300 synthesized sketches and corresponding ground-truth sketches for training and the rest 644 synthesized sketches consist of the gallery. We repeat each face recognition experiment 20 times by randomly partition the data.

Fig. 9 gives the face recognition accuracy against variations of the number of reduced dimensions by NLDA on the CUFS database and CUFSF database respectively. Table VI presents the best face

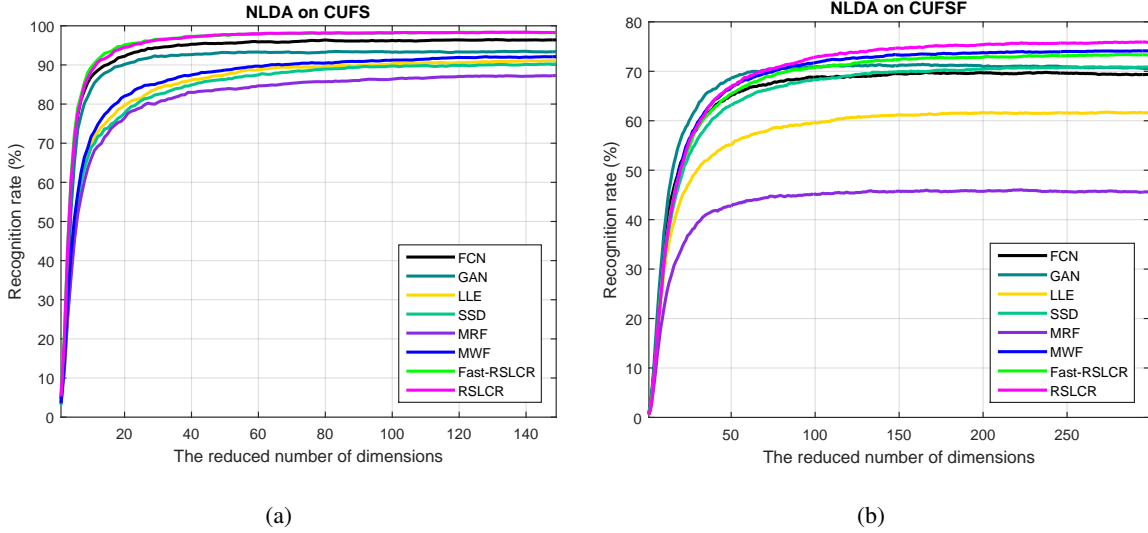


Fig. 9. Face recognition accuracy against variations of the number of reduced dimensions by NLDA on (a) the CUFS database and (b) the CUFSF database.

recognition accuracy at some dimension (the number in bracket). It can be seen that on the CUFS database, the proposed two methods outperform state-of-the-art methods a lot and on the more challenging CUFSF database, our proposed RSLCR method also obtain the best performance with an accuracy of 75.94%. The Fast-RSLCR method has comparable performance with MWF. As shown in table V, although SSD achieves higher SSIM score than MWF, it has lower face recognition accuracy than MWF. This is because though SSD could clear face sketches (much less noise than MWF) it generates face deformations (*e.g.* mouth area as shown in Fig. 6 and Fig. 7).

IV. CONCLUSION

In this paper, we presented a simple yet effective framework for face sketch synthesis based on random sampling and locality constraint. Random sampling in the offline stage could speed up the synthesis process since there is no need to search neighbors online as done in existing methods. The locality constraint could guarantee that similar sampled photo patches have similar reconstruction weights which is neglected in existing works. Through experiments including subjective (perception on the quality of synthesized sketches) and objective (image quality assessment and face recognition) evaluations illustrate the effectiveness of the proposed methods. In addition, discussion on time consumption demonstrates that the proposed Fast-RSLCR method is most efficient method in comparison to state-of-the-art methods. In the future, we would further improve the robustness of our proposed methods by incorporating more

robust features. The application of RSLCR framework on related fields is another mission on the schedule.

REFERENCES

- [1] N. Wang, D. Tao, X. Gao, and X. Li, “A comprehensive survey to face hallucination,” *International Journal of Computer Vision*, vol. 31, no. 1, pp. 9–30, 2014.
- [2] X. Tang and X. Wang, “Face sketch recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 1–7, 2004.
- [3] —, “Face sketch synthesis and recognition,” in *Proceedings of IEEE International Conference on Computer Vision*, 2003, pp. 687–694.
- [4] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, “A nonlinear approach for face sketch synthesis and recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 1005–1010.
- [5] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [6] Y. Song, L. Bao, Q. Yang, and M. Yang, “Real-time exemplar-based face sketch synthesis,” in *Proceedings of European Conference on Computer Vision*, 2014, pp. 800–813.
- [7] X. Gao, N. Wang, D. Tao, and X. Li, “Face sketch-photo synthesis and retrieval using sparse representation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 8, pp. 1213–1226, 2012.
- [8] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.
- [9] S. Wang, L. Zhang, Y. Liang, and Q. Pan, “Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2216–2223.
- [10] X. Wang and X. Tang, “Face photo-sketch synthesis and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 1955–1967, 2009.
- [11] H. Zhou, Z. Kuang, and K. Wong, “Markov weight fields for face sketch synthesis,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1091–1097.
- [12] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, “Transductive face sketch-photo synthesis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 9, pp. 1–13, 2013.
- [13] C. Peng, X. Gao, N. Wang, D. Tao, X. Li, and J. Li, “Multiple representations-based face sketch-photo synthesis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2201–2215, 2016.
- [14] C. Peng, X. Gao, N. Wang, and J. Li, “Superpixel-based face sketch-photo synthesis,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–12, DOI: 10.1109/TCSVT.2015.2502861, 2016.
- [15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.
- [16] H. Chang, D. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 1–8.
- [17] J. Yang, Z. Lin, and S. Cohen, “Fast image super-resolution based on in-place example regression,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1059–1066.
- [18] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

- [19] I. Jolliffe, *Principal component analysis*. New York: Springer, 2002.
- [20] W. Zhang, X. Wang, and X. Tang, “Coupled information-theoretic encoding for face photo-sketch recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 513–520.
- [21] X. Tang and X. Wang, “Face photo recognition using sketch,” in *Proceedings of IEEE International Conference on Image Processing*, 2002, pp. 257–260.
- [22] A. Martinez and R. Benavente, “The AR face database,” CVC, Barcelona, Spain, Tech. Rep. 24, Jun. 1998.
- [23] K. Messer, J. Matas, J. Kittler, J. Luetten, and G. Maitre, “XM2VTSDB: the extended M2VTS database,” in *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication*, Apr. 1999, pp. 72–77.
- [24] P. Phillips, H. Moon, P. Rauss, and S. Rizvi, “The feret evaluation methodology for face recognition algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [25] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [26] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.
- [27] M. Muja and D. Lowe, “Scalable nearest neighbor algorithms for high dimensional data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.
- [28] L. Zhang, L. Lin, X. Wu, S. Ding, and L. Zhang, “End-to-end photo-sketch generation via fully convolutional representation learning,” in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 627–634.
- [29] P. Isola, J. Zhu, T. Zhou, and A. Efros, “Image-to-image translation with conditional adversarial networks,” arXiv preprint arXiv: 1611.07004, Tech. Rep., 2016.
- [30] L. Chen, H. Liao, and M. Ko, “A new lda-based face recognition system which can solve the small sample size problem,” *Pattern Recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.