Group Embedding for Face Hallucination

Tao Lu, Kangli Zeng, Zixiang Xiong, Junjun Jiang and Yanduo Zhang

Abstract—Face hallucination refers an application-specific super-resolution (SR) which predicts high-resolution images from one or multiple low-resolution (LR) inputs. Learning-based SR algorithms infer latent HR images by the guidance of coexisted priors from training samples. Various regularization methods have been successfully applied in face hallucination to ameliorate its ill-posed nature. But most of them only consider the local manifold geometry of a single patch which results in unstable solution for SR reconstruction. In this work, we propose a novel face hallucination algorithm to embed group patches for accurate prior representation and reconstruction. Firstly, we select multiple recurrent self-similar patches to form a group embedding matrix. Then, a graph regularization term and another multiple-manifold regularization term are used to exploit accurate representation for SR performance. Our resulting ADMM algorithm gives a stable solution in an iterative manner. Furthermore, we use a two-step searching strategy for accelerated patch matching. Experimental results on the LFW database, FEI database, and some real-world images demonstrate the superiority of the proposed method, when compared with state-of-the-art face hallucination results both on subjective and objective qualities.

Index Terms—Face hallucination, Group embedding, Graph-regularization, Manifold regularization, Binary Hashing representation.

1 INTRODUCTION

In different applications of face images, such as video surveillance, authentication and entertainment, due to the limitation of imaging sensors and complex real-world conditions, the target facial images are often in low resolution or quality. These low-resolution (LR) facial images severely impact subsequent image processing and display effects, even reduce the recognition and searching performance. Therefore, it is necessary to enhance the resolution and quality of the LR input facial images by super-resolution (SR) algorithms [1]. Face hallucination is a domain-specific SR approach that can infer a HR facial image from one or multiple LR input images without changing the hardware environment.

Generally speaking, face hallucination can be divided into three categories: interpolation-based, reconstruction-based and learning-based methods. Interpolation-based and reconstruction-based SR methods reconstruct HR images by a large collection of data and mathematical models [2] thus they are efficient and intuitive. However, these methods can only achieve good results when the magnification factor is no more than two [3]. On the other hand, learning based SR algorithms use additional priors from training samples which have better adaptability for real applications. Recently, more and more learning-based SR approaches are proposed for different application purposes. This paper focuses on learning-based face hallucinations.

Baker et al. [4] first proposed learning-based face SR algorithm by multi-resolution pyramid matching. Freeman et al. [5], [6] proposed an example based approach that uses Markov networks to learn and predict the details of HR images from the training samples. Inspired by these works, various statistical models [7] are used to improve SR performance. The core problem of SR is to build the mapping function from LR images to their HR versions. From this perspective, some intuitive regression models [8], subspace learning models [9] are developed for face hallucinations. However, they cannot directly use regularization terms to constrain prior representation. Timofte et al. [8], [10] discussed an efficient face SR method, which learned the mapping relationship between HR and LR through anchored regression. Romano et al. [11] learned multiple regressors in advance for fast prediction. In order to ameliorate the over-fitting problem of least squares estimation, representation-based SR methods introduce various constraints to get accurate prior information. Yang et al. [12] proposed a coupled dictionary learning scheme for sparse representation. They assume the LR and HR images shared resolution invariance coefficients than to infer a coexisted HR image from input LR ones. Jiang et al. [13] claimed that locality-constrained representation had better performance than sparse representation. They extended locality regularization into a multi-layer version for boosting its reconstruction performance. Dong et al. [14] used nonlocal mean to enrich the prior source for considering nonlocal patches. Some works asserted that sparse representation [15], [16], [17], [18], low rank constraints [19], [20], bi-layered representation [21], regression method [22] and kernel extension [23] effectively perform good results for face SR. Representation-based SR provides a flexible framework of utilizing training priors. However, learning dictionary is an off-line sophisticated process which cannot cover new training samples.

Neighbor embedding assumes that low-dimensional manifold and its high-dimensional manifold are isometric and searches K nearest neighbors to recover the manifold structure. Chang et al. [24] first proposed a SR algorithm by neighbor embedding. This algorithm uses the locally linear embedding (LLE) algorithm to reconstruct the HR image. Ma et al. [25] proposed a position-patch based hallucination to efficiently super-resolve the LR inputs. Jung et al. [26] uses sparse regularization term to solve the over-fitting of embedding weights. Jiang et al. [27], [28] extended iteratively embedding scheme to multiple layers for accurate
prior. Contextual patches [29], nonlocal patches [30], super-pixel patches [31] are explored for enriching the resource of patch priors. Shi et al. [32] proposed a HR feature matching scheme to get better embedding quality. However, the bottleneck of neighbor embedding based SR is that it searches the neighbors online, if the amount of data is too large, the required long waiting time severely limits their application range.

Recently, a variety of deep learning methods are used for face SR. They offer an end-to-end approach to map LR images to a HR one. Many methods [33], [34], [35] have achieved good visualization results. SDNE [33] used auto-encoders to embed graph nodes and capture highly non-linear dependencies. Dong et al. [34] proposed a three-layers convolutional neural network named “SRCNN”, to learn the mapping functions. Kim et al. [35] further developed a very deep residual convolutional network for face hallucination. Although deep-learning based approaches give good performance for SR tasks, how to design an efficient network is still an open question. Meanwhile, computing resources such as GPU devices are required in the training phase, which limits their applications.

Although the above methods have been successful in SR tasks, most of them do not consider the relationship of inputs patches, that is: inputs patches are sampled from image, thus there exists correlation among these patches as face images contain various local redundant information and structural self-similarity. Combined with graph embedding methods [36], [37], patch prior representation can be optimized effectively. Considering manifold discriminative nature, Harandi [38] used the Grassman manifold diagrammatic embedding discriminant analysis to improve the performance of image matching. Wan et al. [39], [40] developed a multi-manifold local graph embedding method in face recognition based on the maximum boundary criterion (MLGE / MMC) and achieved good results. Wang et al. [41] proposed Laplace graph embedding specific dictionary (LGEC) algorithm which trained a weight matrix and embedded Laplace diagram to rebuild the dictionary. Although methods in [38], [39], [40], [41] showed promising results using multiple manifold structure, the traditional single patch input formalism cannot take into account self-similarity of the multiple manifold geometric image structure. Thus without graph multiple manifold constraints, performance of SR degrades due to unstable estimation from traditional regularized least squares estimation.

In order to alleviate these problems, we propose a method of face hallucination by group embedding to explore the intrinsic multiple manifolds for better prior representation and reconstruction. We explain the conceptual illustration of group embedding in Fig. 1. Given a candidate feature/patch space $M$, the basic idea of embedding is giving a query feature/patch to find $K$ nearest points which can represent the query point as a linear combination of the selected neighbors. We see clearly that subfig (a) find $K$ neighbors easily in the candidate manifold $M$. In subfig (b), two query points which connect into a line, can find two $K$ neighbors separately, however some of the selected points are the same because two query points are very similar. Subfig (c) embeds a group of query points to select their neighbors separately. Because points from one group usually have intrinsical structure, the selected points hold their relationship in the select new sub-manifold. On the other hand, the introduced graph constraint further makes the representation coefficients of similar input patches consistent and stable.

The specific implementation of our method is as follows. First, we use the input image itself to construct the image patch group, and construct the graph structure to embed the group patches to learn the prior knowledge of the image representation and obtain the additional information needed for reconstruction. Inspired by [42], [43], we use a two-step strategy to achieve image matching to improve efficiency. At the same time, a new ADMM algorithm is introduced to iteratively optimize the objective function and solve the exact representation coefficient. Experiments show that the face image reconstructed by the proposed algorithm is improved both in visual effects and image quality evaluation indicators. The main contributions are listed in three aspects.

- We propose a group embedding framework for face hallucination. Different from the single patch input method, the proposed method uses group patches as inputs. According to the self-similarity of the face image, the graph constraints are introduced. It maintains the intrinsic local geometric feature of the samples by the group embedding, and the representation ability of the image is enhanced to improve the reconstruction performance.

- Our method adopts a two-step strategy to match the image patches, not only to get the accurate representation coefficient, but also to improve the efficiency greatly. First, coarse matching is done in hash space to achieve the purpose of dimensionality reduction, and then exact matching in the Euclidean space are used to match the exact nearest neighbors.

- We introduced the ADMM algorithm to optimize the objective function, and adopted iteratively manner to solve the regularization problem of the group embedding to obtain a stable solution.

The rest of this paper is organized as follows. Section 2 gives the image degradation model and patch-based face hallucination. Section 3 introduces our proposed GE based hallucination as well as its optimization. Experimental results are presented in Section 4 and conclusion is drawn in Section 5.
2 RELATED WORK

2.1 Image degradation model

In the perspective of physical imaging, supposing that the desired high-resolution (HR) image $Y \in \mathbb{R}^{m \times n}$, and its down-sampled version as $X \in \mathbb{R}^{t \times t \times n}$ in low-resolution (LR), here $t$ is scale factor, which indicates down-sample multiple. Mathematically, the LR image is generated by following image degradation model:

$$X = ECY + \varphi,$$  

(1)

where $E$ is the down sampling matrix, $C$ is a blur function to describe the diffraction effect of imaging system and $\varphi$ represents system noise which usually regarded as additive Gaussian white noise. Usually, we ignore noise items and set $D = EC$ as image degrading operation, then above image degradation model becomes: $X = DY$.

However, in real application scenario, we only observe the LR image $X$ from imaging system. Intuitively, if the degrading operation $D$ can be measured in advance, thus the desired HR image $Y$ can be inferred by following objective function:

$$\min_Y \left\{ \|DY - X\|_2^2 + \lambda \Gamma(Y) \right\},$$

(2)

where, $\lambda$ is a balance parameter, and $\Gamma(Y)$ is a regularization term to provide prior from natural image, such as smooth prior.

2.2 Patch-based face hallucination

In order to pursuit better prior, image is always segmented into small patches manner to reconstruct the whole image. Let $\{A_i\}_{i=1}^N$ and $\{B_i\}_{i=1}^N$ denote the LR and HR training samples, where $N$ is the number of training samples, $A_i \in \mathbb{R}^{m \times n}$ and $B_i \in \mathbb{R}^{m \times n}$, here $t$ is the scale factor. Given an input image $X \in \mathbb{R}^{m \times n}$, the purpose of SR is to render a high resolution image $Y \in \mathbb{R}^{mt \times nt}$. We operate the image super-resolution in patch level, for every LR patch $x_i \in \mathbb{R}^{t \times t}$ (patches are with size of $\sqrt{t} \times \sqrt{t}$ and transformed into column vectors) to infer the potential HR patch $y_i \in \mathbb{R}^{(t^2 \times 1) \times 1}$. The first step is to learn dictionary pairs $H_i \in \mathbb{R}^{(t^2 \times 1) \times k}$ and $L_i \in \mathbb{R}^{t \times k}$, $k$ is the number of dictionary atoms. Hereby, the objective function of the LR representation weight is as following:

$$\min_{L_i, \alpha_i} \left\{ \|x_i - L_i \alpha_i\|_2^2 + \lambda \Gamma(\alpha_i) \right\},$$

(3)

where, $\alpha_i$ is the $i$-th representation the LR coefficient vector, $\Gamma(\alpha_i)$ is a regularization constraint term to provide prior from patch and $\lambda$ is a balance parameter. $\alpha_i$ is assumed as resolution-invariance feature which satisfies HR and LR manifold consistent assumption, therefore, the reconstructed high-resolution patch can be expressed as following:

$$y_i = H_i \alpha_i.$$  

(4)

As a result, the desired HR image $Y$ can be stitched by all patches.

3 FACE HALLUCINATION BY GROUP EMBEDDING

As shown in Fig. 2, the whole framework of face hallucination by group embedding consists of two stages: dictionary learning and SR reconstruction. First, a LR image is divided into patches and a few of similar patches are selected to form a patch group. Then we use hash binary search engine to find the group embedding dictionary. The HR group patches can be reconstructed by the embedding weights. We introduce the group embedding as the following subsections.

3.1 Group embedding analysis

Different from neighbor embedding, group embedding not only extends the selected patches for prior involving reconstruction but also provides stable solution of representation objective function. From this point, we wish to fully investigate the reason why group embedding outperform neighbor embedding. Generally, we use entropy to measure the prior information involving image reconstruction. As we know, in information theory, entropy( also known as self-information) denotes the average amount of information and mutual information represents the measure of similarity between two variables. Because many methods have applied entropy and mutual information to image quality assessment [44], [45]. We use average entropy and average mutual information to show the quality of embedding. Given a random patch $x_1$ in the input image, then we set this point as processing object and at the center of a circle with radius of $t$. The selected patches are assume to satisfy the following condition:

$$\max_{x_1, x_j \in G} \|x_1 - x_j\|_i \leq t, s.t. G = \{x_1, \cdots, x_p\},$$

(5)

where $t$ is a very small threshold to maintain self similarity, $p$ is the number of similar patches in a group. The purpose of embedding is to find most similar $K$ patches for every input patch $x_j$. Here, we define $→$ as an operation to select $K$ nearest neighbors. Thus, for input $x_1$, we have $x_1 \rightarrow L_1 = \{l_1, l_2, \ldots l_K\}$, for input $G = \{x_1, x_2, \ldots x_p\}$, we have $G \rightarrow L_g = \{l_1, l_2, \ldots l_g\}$, here $1 \leq g \leq pK$ is the number of embedding patches. From this point, we can use average entropy of embedding set to measure the prior which involves in reconstruction. Average entropy (AE) is calculated as:

$$AE = \frac{1}{g} \sum_{j=1}^{g} H(L_j),$$

(6)

here, $H(x)$ function is the entropy of $x$ which is defined as:

$$H(L) = \sum_{i=1}^{c} p(l_i) \log p(l_i), s.t. \sum_{i=1}^{c} p(l_i) = 1,$$

(7)

where $L \in \mathbb{R}^c$ and $p(l_i)$ is the probability function of $l_i$ in $L_g$. In this paper, $p(l_i)$ represents the probability of each gray level pixel point. In the other hand, we define average mutual information(AMI) to measure the correlation between testing patch $x_1$ and the embedding set $L_g$:

$$AMI = \frac{1}{g} \sum_{j=1}^{g} I(x, l_j) = \frac{1}{g} \sum_{j=1}^{g} (H(x) + H(l_j) - H(x, l_j)),$$

(8)

here, function $I(x, y)$ is the mutual information which is calculated by two variable $x$ and $y$, the details can be found in [44].
Fig. 2. Flowchart of group embedding for face hallucination. It provides a multi manifold online learning method, which is mainly divided into group dictionary learning and SR reconstruction process.

In order to testify the AE and AMI index for group embedding, we use LFW face database as in following experimental section. We list 250 patches from 2 images to show the results in Table 1.

<table>
<thead>
<tr>
<th>The number of similar patches in a group(p)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>0.1666</td>
<td>0.1813</td>
<td>0.1822</td>
<td>0.1864</td>
<td>0.1918</td>
</tr>
<tr>
<td>AMI</td>
<td>0.1587</td>
<td>0.1693</td>
<td>0.1722</td>
<td>0.1744</td>
<td>0.1788</td>
</tr>
</tbody>
</table>

The higher the AE and AMI score, the more prior information that involves in the reconstruction. Therefore, when \( p = 3 \), AE and AMI are the biggest, and the reconstruction performance is the best. In Section 4.2.1, we will further illustrate the performance of group embedding from this point.

### 3.2 Group embedding

For group embedding, we introduce graph constraints to improve the performance of our proposed method. Therefore, formula (3) can be rewritten into the following form:

\[
\begin{align*}
\min_{D^i, S_i} & \left\{ \left\| G_i - D^i S_i \right\|_F^2 + \gamma T_r \left( S_i \Omega_i (S_i)^T \right) + \beta \left\| \Theta_i \circ S_i \right\|_F^2 \right\} \\
\text{s.t.} & \quad 1^T s_j = 1
\end{align*}
\]

The first term measures the reconstruction error. Here, \( G_i = [x_1, \ldots, x_p] \) is the p similar patch group of i-th input patch \( x_i \), \( D^i = [D_1, \ldots, D_p] \) is the corresponding group dictionary. Here, superscript \( l \) represents LR space. \( S_i = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_p \end{bmatrix} \) is a group coefficient matrix composed of expression coefficients corresponding to each similar patch \( x_j \) in \( G_i \).

The second term of the objective function (9) is the graph constraint and \( \gamma \) is a graph constraint parameter. For \( G_i \), we construct a nearest neighbor graph \( T \) with \( p \) vertices, where each vertex represents a data point, and the edges connect neighboring points to form a graph. Because we process every patch in order, so the subscript \( i \) can be omitted for convenience.

Let \( W \) be the weight matrix of \( T \). For each point \( x_i \) in \( T \), we connect it with the points in its \( p \)-nearest neighborhood set as follows:

\[
C(x_i) = \{ x_i | i \in N_p(x_i) \},
\]

where \( N_p(x_i) \) is the index of the \( p \)-nearest neighbor. If \( x_j \) is among the \( p \)-nearest neighbors of \( C(x_i) \),

\[
w_{ij} = \left\{ \begin{array}{c}
\arg \min_{w_{ij}} \left\| x_i - \sum_{j \in N_p(x_i)} w_{ij} x_j \right\|_2^2, \quad s.t. \sum_{j=1}^{p-1} w_{ij} = 1.
\end{array} \right.
\]

The degree of \( x_i \) is defined as following:

\[
V_i = \sum_{j=1}^{p} w_{ij},
\]

Thus the graph minimization
function can be wrote as follows:
\[
\frac{1}{2} \sum_{i,j} (s_i - s_j)^2 w_{ij} = \frac{1}{2} \sum_{i,j} (s_i^2 + s_j^2 - 2s_is_j) w_{ij} = \frac{1}{2} \left( \sum_i s_i^2 w_{i1} + \sum_j s_j^2 w_{j2} - 2 \sum_{i,j} s_is_j w_{ij} \right),
\]
(13)
where \(Tr(\bullet)\) represents the trace of the matrix and \(\Omega = V - W\) is the Laplacian matrix.

And the third is the local constraint and \(\beta\) is the balance parameters. In formulate (9), we define \(\|\Theta_i \odot S_i\|_F\) to represent locality-constrain term. Here, \(\odot\) denotes matrix dot product. Locality constraint matrix is defined as
\[
\Theta_i = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_p \end{bmatrix},
\]
where the \(j\)-th element in vector \(\sigma_i\) is calculated by
\[
\sigma_j = \exp \left( \frac{\text{dist}(x_i, d_j)}{\pi} \right),
\]
(14)
where \(\sigma_j\) is the locality adaptor and function \(\text{dist}\) indicates the Euclidean distance. \(d_j\) is the atoms in sub-dictionary \(D_i\) according input patch \(x_i\). \(\pi\) is used for adjusting the weight decay speed for the locality adaptor.

### 3.3 Optimization
As we all know, function (9) has a solution, which has been proved in [46], [47]. In order to improve the stability of the solution, in this paper, we use a different solution based on ADMM [48], which can update all the columns of \(S\) at the same time. We use the following fully separated objective functions to optimize group embedding:
\[
Q(S, Z, U) = \|G - D^l S\|_F^2 + \gamma Tr(\Omega S\Omega^T) + \beta \|\Theta \odot Z\|_F^2 + \rho \|S + U - Z\|_F^2,
\]
(15)
where, \(U\) is the scaled dual form variable.

This problem can be divided into three sub-problems by the following iterative and alternating steps:

**S sub-problem:**
\[
S^{(k)} = \arg \min_S \left\{ \|G - D^l S\|_F^2 + \gamma Tr(\Omega S\Omega^T) + \beta \|\Theta \odot Z\|_F^2 + \rho \|S + U^{(k-1)} - Z^{(k-1)}\|_F^2 \right\}.
\]
(16)

**Z sub-problem:**
\[
Z^{(k)} = \eta \left( S^{(k)} + U^{(k-1)} \right).
\]
(18)

**U sub-problem:**
\[
U^{(k)} = U^{(k-1)} + S^{(k)} - Z^{(k)}.
\]
(19)

After several iterations, the \(i\)-th representation coefficient matrix \(Z_i\) is ready. Thus reconstructed HR group patches can be expressed as following:
\[
G_i^h = D_i^h Z_i,
\]
(20)
where \(D_i^h\) is the HR space group dictionary. Finally, the reconstructed patch \(y_i\) is the first column of \(G_i^h\) and then the desired HR image \(Y\) can be stitched by all patches.

The entire procedure of GE is depicted in Algorithm 1 to speed the convergence, \(S\) is initialized with the standard locality-constrained representation.

**Algorithm 1** Face hallucination by group embedding

**Input:**
1. The group patch matrix \(G\) of input patches, the group dictionary matrix \(D\) corresponding positions and regularization parameter \(\gamma\) and \(\beta\).

**Initialize:**
2. \(S^{(0)} = \arg \min_S \left\{ \|G - D S\|_F^2 + \beta \|\Theta \odot Z\|_F^2 \right\}
3. \(Z^{(0)} = S^{(0)}
4. \(U^{(0)} = 0\)

**Iterate:** for \(k = 1, 2, \ldots\)

**Update \(S^{(k)}\) as the solution of:**
5. \(\left( (D^l)^T D^l + \rho I \right) S + \gamma S \Omega = (D^l)^T G + \rho (Z^{(k-1)} - U^{(k-1)})\)

**Update \(Z^{(k)}\):**
6. \(Z^{(k)} = \eta \left( S^{(k)} + U^{(k-1)} \right)\)

**Update \(U^{(k)}\):**
7. \(U^{(k)} = U^{(k-1)} + S^{(k)} - Z^{(k)}\)

**Output:** The desired result is \(Z^{(k)}\).

### 3.4 Group based dictionary learning
In order to learn the dictionary which supports the group-patch matrix, we combine every single dictionary together. Keep in aware that LR images are interpolated into HR image size, and every image owns same patch extracting order. If there are \(q\) patches in one image, then we extract HR and LR training patch-pools as \(H^p\) and \(L^p\) respectively, both of them have \(N \times q\) patches. Here, different from traditional patch-based approaches [13], [25], we learn the dictionary from feature space not in pixel domains. Four gradient features (first- and second-order gradients) in the horizontal and vertical directions are used for searching the nearest patches in patch pools. Thus for image \(X_j\), its four-layer feature-maps \(X_j^l\) are defined as:
\[
X_j^l = [g_{1(i)}(X_j); g_{2(i)}(X_j); g_{3(i)}(X_j); g_{4(i)}(X_j)],
\]
(21)
where, \(g_{m(i)}\) is the \(m\)-th filter \((m = 1, 2, 3, 4)\) on the \(i\)-th patch over image \(X_j\). Then, we transfer LR and HR training databases into feature spaces as \(H^l\) and \(L^l\).
For one LR patch $x_i$, the purpose of dictionary learning is to find the most similar patches from patch-pool $L^p$ using the feature measure spaces. We use LR input patch feature $x_i^f$ to cluster $k$ nearest feature patches in $L^f$. Suppose the index of $k$ nearest neighbors is $C_k(x_i)$, then,

$$C_k(x_i) = \text{support}(\text{dist}_i, \text{dist}_i = ||x_i - t_j||_2, \ (22)$$

here, $t_j$ is the $j$th atom in feature pool $L^f$, operation support means index selection. With $C_k(x_i)$, we can select $k$ coupled patches from HR and LR patch-pools to form $L_i$ and $H_i$. Thus the group-dictionaries are $D_i^L = [L_1,...,L_p]$, $D_i^h = [H_1,...,H_p]$.

3.5 A naive two-step searching method

In the 3.3 section, we use the K-NN method to find the most similar patches from patch-pool $L^p$ using the feature measure spaces. Because the feature pool $L^f$ is a set of high dimensional data, a problem encountered when searching for the most similar patches is the “dimensionality disaster”. That is to say, when the data dimension is too large, the previous indexing methods (such as K-NN) can not play a role. In this case, new methods are necessary, and the hash algorithm satisfies this requirement. Although Hash algorithm (Hamming distance) can search effectively, the Euclidean distance is more accurate than the Hamming distance. Therefore, we will use the Hamming space and the Euclidean space to search the nearest neighbour one step by step in our algorithm.

![Fig. 3. An example of the process of the proposed hashing encoding scheme.](Image)

Fig. 3 shows an example of the encoding process by employing the proposed hashing encoding scheme. The detailed steps are as follows: First, calculating the average gray average of all the pixels in the image (or image patch). Then, comparing the average value of each pixel to the average. The binary code for each pixel location is calculated as follows:

$$\psi(i,j) = \begin{cases} 1 & \Psi(i,j) \geq \text{means} \\ 0 & \Psi(i,j) < \text{means} \end{cases}, \ (23)$$

where term $(i,j)$ represents the pixel position of the data $\Psi$, means is the pixel average of data $\Psi$. Finally, encoding process, which combines the results of the previous step and composes the binary number of 9 bits, which is the encoding of the image (or image patch). After encoding, the original data hash code is $\Psi^* = [\Psi(1,1),...\Psi(3,3)] = [0,0,...1]$.

4 EXPERIMENTAL RESULTS

In this section, we will discuss various parameter settings and then demonstrate the SR results of the proposed method to compare with other SR methods in terms of visual quality and objective assessment values. In the proposed method, peak signal to noise ratio (PSNR) and structural similarity (SSIM) are adopted to objectively evaluate the quality of the estimated images. The higher the SSIM and PSNR are, the better is the image quality achieved.
In order to better verify the effectiveness of the proposed method, we have compared the proposed method with some state-of-the-art SR methods such as LSR [25] based method, LLE [24] based method, LCR [13] based method, LINE [27] based method, TLCR [29] based method, SRCNN [34] based method, VDSR [35] based method and RDCN [51] based method, respectively, on three different databases with LFW Face Database [52], FEI Face Database [53] and some real world images from LDHF face database [54]. All the parameters are set as their best performance according to their papers.

4.1 Database Description

The databases used in this experiment will be introduced in this section.

LFW Face Database: 303 facial images are randomly selected from LFW which are cropped into size of $64 \times 64$ pixels as HR images. Here, we use 270 images as training samples and the rest 33 images are testing samples. The scale factor is set to 4, and LR images are generated by average blur (blur kernel is 4 pixels) from their corresponding HR images with size of $16 \times 16$. Five representative HR and LR images are shown in Fig. 4.

FEI Face Database: The database contains 400 color images from 200 subjects (100 men and 100 women) and each subject has two frontal images, one with a neutral expression and the other with a smiling facial expression. All the images are cropped to $120 \times 100$ pixels, and we choose 360 images (180 subjects) as the training set, leaving the rest 40 images (20 subjects) for testing. Fig. 5. Samples from FEI Face Database (HR and LR images). The down-sampling factors for the three left and right three are 4 and 8, respectively.

4.2 Parameter Selection

In this paper, there are some parameters need to be determined: the patch size, overlap pixels between patches, nearest neighbors for constructing the graph of the original input image patch space, the number of iterations $m$, the number of dictionaries searched in Euclidean distance $K_1$, the number of dictionaries searched in Hamming space $K_1$, the number of hash bits $r$, the balance parameters $\beta$ and $\gamma$. In the experiment, the patch size on the LFW and FEI databases are set to $24 \times 24$ and $12 \times 12$, respectively. The sampling factor for all experiments is set to 4. Other parameters will be discussed below.

4.2.1 The quality of group embedding

In order to test the effect of $p$ on reconstruction performance, we will verify it by group embedding quality and PSNR. Embedding quality is closely related to image representation, so it is reasonable to use reconstruction errors to measure the embedding quality. In Section 3.1, AE and AMI are introduced to evaluate the performance. Unlike other methods [44], [55] to evaluate the embedding quality by dimensionality reduction, we use reconstruction error to measure the quality of embedding.

According to the variable assumptions in Section 3.1, supposing that $\alpha_g$ represents the representation coefficient corresponding to $x_1$ in the $p$-th subsection with $x_1$ centred, $1 \leq i \leq pK$, we can define the quality of embedding as

\[
Q = \|x_1 - L_\alpha \alpha_g\|^2_2.
\]

Since $p$ is important for group embedding, we use the above formula to find $Q$ according to different $p$. In fact, it represents the reconstruction error, which is closely related to the reconstruction performance. As shown in Fig. 6, we plotted the embedding quality $Q$ and PSNR of different $p$. Groups can provide self-similarity of geometric structures in an image, but $p$ cannot be set too large because it leads to loss of such similarity. In this paper, $p$ is set to 3.

4.2.2 The number of dictionaries searched in Hamming space $K_1$

In order to test the effect of $K_1$, we test the performance with different $K_1$ and fixed other parameters. As shown in
Fig. 7, we plot the average of PSNR and SSIM values of LFW and FEI to different $K_1$. (a) and (b) represent the FEI and LFW databases, respectively. It is easy to find that as the $K_1$ increases, the performance of the proposed method also increases. However, if $K_1$ is too large, the performance of the algorithm will decrease and the time consumption will increase. From above, we use $K_1 = 1600$ ($K_1 = 1700$ in FEI) to achieve the best performance in LFW.

![Fig. 7. The average PSNR and SSIM values of different value $K_1$.](image)

4.2.3 The number of dictionaries searched in Euclidean distance $K_2$

As shown in Fig. 8, horizontal axis indicates the number of $K_2$, and vertical axis represents the averages of PSNR and SSIM values. It is easy to see that as $K_2$ increases, the performance of the proposed method increases too. However, large $K_2$ not always improves the performance and will increase the running times and drop the performance. It is reasonable that you select more local or nonlocal patches and more unrelated patches maybe involve in reconstruction which degrades the self-similarity prior. Therefore, we use $K_2 = 250$ ($K_2 = 230$ in FEI) to achieve a balance between performances and time-consuming in LFW.

![Fig. 8. Performance of face reconstruction with different $K_2$. (a) and (b) represent the FEI and LFW databases, respectively](image)

4.2.4 The number of hash bits $r$

We test the performance of our method with different $r$. As shown in Fig. 9, we plot the average of PSNR and SSIM values of all LFW and FEI test images with different $r$. It is easy to see that the number of hash bits $r$ have a significance benefit to the performance of the proposed method. As you can see from the Fig. 9, (a) and (b) represent the FEI and LFW databases, respectively and the performance of SR is also falling fast as the value of $r$ increases, so $r$ can't be set too high. Therefore, selecting a proper parameter $r = 64$ ($r = 64$ in FEI) in LFW, the proposed method gains best results.

![Fig. 9. PSNR (dB) and SSIM results of our method using different the number of hash bits $r$.](image)

4.2.5 The influence of different $\beta$ and $\gamma$

$\beta$ and $\gamma$ respectively represent the balancing parameters of locality-constraint and graph constraint, which are very important for the performance of our algorithm. We fixed one parameter and vary the other one to search the best performance of GE. As shown in Fig. 10 and Fig. 11, we plot the average of PSNR and SSIM values of all testing images from LFW and FEI with different $\beta$ and $\gamma$ values. When $\beta = 0.4$ and $\gamma = 350$ (set $\beta = 0.5$ and $\gamma = 400$ in FEI), GE achieves its best results in LFW.

![Fig. 10 and Fig. 11. Performance of face reconstruction with different $\beta$ and $\gamma$.](image)
4.2.6 The effect of iteration number $m$

As shown in Fig. 12, we plot the average MSE (Mean Square Error) values of all test face images according to the number of iteration, and find that: (i) as the iteration number increases, the gain of the proposed method becomes larger, which implies that this parameter have significant influence for the performance of the proposed method; (ii) the proposed method can be fast convergent and needs multiple iterations.

4.3 Comparison with state-of-the-arts methods

To the best of our knowledge, TLCR has the best performance among patch-based face hallucination and RDCN has the best performance among deep-learning based approaches. As the typical position-based method: LCR has best performance in single-layered face hallucinations, LINE has best performance on multi-layered position based face hallucinations. We compare our method with above state-of-the-art methods. PSNR and SSIM are listed in Table 2, we will discuss following sub-problems:

4.3.1 Patch prior vs group-patch prior

As shown in Table 2, LCR, LINE, TLCR, LSR and LLE based approaches are all on patch level, our method based on group-patch, which improves 4.37 dB, 2.55 dB, 1.81 dB, 1.62 dB, 1.34 dB and 0.35dB comparing with Bicubic, LSR, LLE, LCR, LINE and TLCR on LFW face database. As shown in Table 2, our method improves 6.47 dB, 1.86 dB, 1.50 dB, 1.22 dB, 0.78 dB and 0.29 dB comparing with Bicubic, LSR, LLE, LCR, LINE and TLCR on FEI face database. Meanwhile, SSIM is consistent with PSNR performance. This demonstrates that patch group always has better performance than single patch.

4.3.2 Deep learning or shallow learning

It is believed that with deep learning architecture always bring performance boosting in super-resolution. In this paper, we compare our method with two state-of-the-art deep learning models: SRCNN, VDSR and RDCN. As shown in Table 2, GE only owns two-layer: one for representation and the other one for reconstruction, but it has better performance than above two models on LFW face database. As shown in Table 2, the average PSNR(db) and SSIM of different SR methods on FEI database are obviously higher than deep learning models. This may indicates that if the samples are insufficient, shallow learning approach may has slightly better performance than deep learning approaches.

In order to visualize the results from different hallucinations, we list six representative testing images in Fig. 13 and Fig. 14. Comparing with all other approaches, GE has better details on the edges of facial organs, e.g., mouth, eyes. This further confirm the effectiveness of GE.

4.4 Complexity and running time

Speed can be a critical factor in practice. Therefore, in this section, we will comprehensively discuss the performance of SR algorithms and test run time issues. We employ Matlab scripts and we use the same system with Intel Core i5 CPU 720 @ 1.80GHz, 8 GBytes RAM for our experiments. As shown in Fig. 15, we list some state-of-the-arts methods compared with the proposed method in this paper. As can be seen from the Fig. 15, when GE uses only the single-step strategy (KNN), although the reconstructed image has the best performance, it consumes a large amount of computing time, which is obviously not desirable. However, when we use the two-step strategy, it provides a good trade-off between optimal performance and reasoning time. Compared with deep learning based methods (SRCNN, VDSR and RDCN), GE can not only achieve optimal performance, but also have almost the same time consumption. As we all
know, the method based on deep learning generally takes a long time to train, hours or days or even months, which is obviously not what we want. The experimental results prove the effectiveness of the algorithm. That is to say, learning binary hash code can save time effectively without affecting the performance of the algorithm.

4.5 Experimental results in the real world

In this section, we conduct experiments on some real world images from LDHF face database [54]. It contains some photos taken at different distances in the same scene. In our test experiment, we took the face image cropped from the 150m photo as input, and the face image cropped from the 60m photo was used as a contrasting HR image. Moreover, the FEI database will be used as a training sample. As shown in Fig. 16, We crop some representative face images of size $120 \times 100$ in the red frame of the original image, then, take these pictures as input for reconstructing.

We used several different methods of reconstruction to compare our approach, such as LSR, LLE, LCR, LINE, TLCR, SRCNN, VDSR, RDCN. The result of test images is shown in Fig. 17. It is obvious that our method has better visual details than all other methods which prove that our method has super performance even than some deep learning based approaches, especially the red boxes in Fig. 17, such as the eyes, noses, etc.

5 Conclusion

In this paper, we propose a novel group embedding (GE) SR method. Different from single-patch method, GE use the input image itself to construct similar patch group, and construct the graph structure to embed the group patches to...
learn the prior knowledge of the image representation and obtain the additional information needed for reconstruction. What’s more, it is important that an optimal solution can be obtained for reconstructing by ADMM. And we also match embedded patches with a two-step search strategy, which can effectively improve the efficiency of our proposed approach. The extensive experimental results have demonstrated the effectiveness of the proposed algorithm.

**APPENDIX A**

**Proof of the Sylvester Equation**

The general form of Sylvester equation is $PS + SQ = R$.

**Theorem 1.** Using the Kronecker product notation and the vectorization operator $vee$, we can rewrite Sylvester’s equation in the form

$$(I_m \otimes P + Q^T \otimes I_n) veeS = veeR$$

where $I_k$ is the $k \times k$ identity matrix. In this form, the equation can be seen as a linear system of dimension $mn \times mn$.

**Proposition.** Given complex matrices $P$ and $Q$, Sylvester’s equation has a unique solution $S$ for all $R$ if and only if $P$ and $−Q$ have no common eigenvalues.

**Proof.** Consider the linear transformation $S : M_n \to M_n$ given by $S \to PS + SQ$.

(i) Suppose that $P$ and $−Q$ have no common eigenvalues. Then their characteristic polynomials $f(z)$ and $g(z)$ have highest common factor 1. Hence there exist complex polynomials $p(z)$ and $q(z)$ such that $p(z)f(z) + q(z)g(z) = 1$. By the Cayley-Hamilton theorem, $f(P) = 0 = g(−Q)$; hence $g(P)q(P) = I$. Let $S$ be any solution of $\Gamma(S) = 0$; so $PS = −SQ$ and repeating this one sees that $S = q(P)g(P)S = q(P)g(P)(−Q) = 0$. Hence by the rank plus nullity theorem $L$ is invertible, so for all $R$ there exists a unique solution $S$.

(ii) Conversely, suppose that $l$ is a common eigenvalue of $P$ and $−Q$. Note that $l$ is also an eigenvalue of the transpose $P^T$. Then there exist non-zero vectors $v$ and $w$ such that $P^Tw = lw$ and $Qv = −lv$. Choose $R$ such that $Rv = \overline{w}$, the vector whose entries are the complex conjugates of $w$. Then $PS + SQ = R$ has no solution $S$, as is clear from the complex bilinear pairing $\langle (PS + SQ)v, w \rangle = \langle Rv, w \rangle = \langle \overline{w}, w \rangle$; the right-hand side is positive whereas the left is zero.

**ACKNOWLEDGMENTS**

This work was supported by the National Natural Science Foundation of China (61502354, 61501413, 61672174).
REFERENCES


Fig. 17. Visual results on LDHF database. From left to right: 150m photo, LSR [25], LLE [24], LCR [13], LINE [27], TLCR [29], SRCNN [34], VDSR [35], RDCN [51], ours and 60m photo.