Single Image Super-Resolution via Self-Similarity and Low-Rank Matrix Recovery

Hong Wang[1], Jianwu Li[2], Zhengchao Dong[3,4]

1. School of Science, Tianjin University, Tianjin 300072, China

2. Beijing Key Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

3. Department of Psychiatry, Columbia University, New York, NY10032, United States

4. New York State Psychiatric Institute, New York, NY 10032, United States

Abstract: We propose a novel single-image super resolution (SISR) approach using self-similarity of image and the low-rank matrix recovery (LRMR). The method performs multiple upsampling steps with relatively small magnification factors to recover a desired high resolution image. Each upsampling process includes the following steps: First, a set of low/high resolution (LR/HR) patch pairs is generated from the pyramid of the input low resolution image. Next, for each patch of the unknown HR images, similar HR patches are found from the set of HR/LR patch pairs by the corresponding LR patch and are stacked into a matrix with approximately low rank. Then, the LRMR technique is exploited to estimate the unknown HR image patch. Finally, the back-projection technique is used to perform the global reconstruction. We tested the proposed method on fifteen images including humans, animals, plants, text, and medical images. Experimental results demonstrate the effectiveness of the proposed method compared with several representative methods for SISR in terms of quantitative metrics and visual effect.

Keywords: single-image super-resolution, self-similarity, image pyramid, low-rank matrix recovery

1. Introduction

Super- resolution (SR) is to construct a high-resolution (HR) image from one or multiple low-resolution (LR) images by finding the missing high frequency information caused by the limitation of hardware [1]. This problem arises in many application domains, such as computer vision, medical imaging, video surveillance, and entertainment.

SR image reconstruction is an inherently ill-posed problem because an LR image can be generated from different HR images. Conventional approaches to SR require multiple LR images of the same scene taken at sub-pixel misalignments (multiple-frame image super-resolution, MISR). For these methods, prior knowledge is usually used to predict a large

number of unknown pixel values according to the input pixels. The performance of these approaches degrades rapidly when the number of input LR images is insufficient and the magnification factor of the image is large [2].

Another class of SR approaches only uses one input LR image (single-frame image super-resolution, SISR). The simplest methods for SISR assume that the images are spatially smooth, and analytical interpolation formulae, e.g., the bicubic function [3], are used to predict missing image details. However, natural images contain strong discontinuities such as texture and edges. The analytical interpolation methods often suffer from over-smoothed edges and the loss of details in textured regions.

Example-based SR methods for SISR are based on the assumption that the missing HR details can be learned and inferred from a representative training set or the input LR image [4]. Some papers propose to form a training set from some extra HR images [5-12] and utilize the high-frequency details of HR images in the training set to predict the HR image in terms of the input LR image. Freeman et al. pointed out that it was unreasonable to generate the correct high-frequency information without restriction to a specific class of training images [5].To ensure that the training set is compatible with the missing high-frequency details, self-similarity is utilized according to the observation that a smaller image patch tends to redundantly recur many times both within the same scale and across scales [13]. Some approaches, which do not depend on an extra database of training images, have been proposed to use the input LR image as the source for example patches (e.g., [14-17]). For the approaches in these papers, the LR/HR patch pairs are generated from the pyramid of the input LR image, and then the similar patches are searched from these LR/HR patch pairs by exploiting the multi-scale self-similarity. Last, an HR image is synthesized from those similar patches incorporating prior knowledge of image.

The performance of these aforementioned SISR algorithms also depends on the correct mapping from LR to HR patches. In fact, some existing approaches, such as linear function [18], multiple linear mappings [19], support vector regression[20], kernel ridge regression[21], anchored neighborhood regression (ANR) [10], and deep neural network [11, 12], emphasize the special mapping relationship between the LR and HR patches.

An example-based method, called RAISR, differs from the aforementioned methods and does not need to learn the mapping from LR to HR patches [22]. Instead, RAISR applied a set of pre-learned filters on the image patches to enhance the quality of the computationally very cheap interpolation method. The filters have been previously learned based on pairs of LR and HR training image patches, and hashing is done by estimating the local gradients' statistics.

The theory of compressed sensing has motivated the research on the sparse representation-based image restoration [8, 23, 24]. The compressed sensing-based methods depend on the fact that the image patches can be sparsely represented under common basis and overcomplete dictionary and assume that these representation coefficients are identically and independently distributed, and the correlation between the sparse coefficients can be ignored. Recently, simultaneous sparse coding (SSC) [25] has considered the structural correlation in the sparse model. Furthermore, several state-of-the-art image restoration techniques were proposed, including spatially adaptive iterative singular-value thresholding (SAISVT) [26], low-rank matrix recovery and neighbor embedding algorithm (LRNE)[27], and block-matching 3D filtering (BM3D) [28].

Dong et al. presented an image restoration algorithm toward modeling nonlocal similarity in natural images by the singular-value thresholding (SVT) technique and discussed its connection with simultaneous sparse coding [26]. Instead of coding each patch individually, they simultaneously coded a set of nonlocal similar patches under the dictionary of local principle component analysis from the input image.

Chen et al. proposed an LRNE method based on the low-rank matrix recovery (LRMR) and neighbor embedding (NE) algorithm for SISR [27]. Each training patch is put into a group with its K nearest neighbors. LRMR is utilized to learn the latent low dimensional subspace for LR patches and HR patches of each group, respectively. For the input LR patch, the weights over the low dimensional subspace of the nearest LR patches group to this input LR patch are computed. According to the structural geometries similarity between the LR images and HR images, the NE algorithm is used to reconstruct the missing pixels of HR images from the obtained weights and the low dimensional subspace of the corresponding HR patches group. The LRNE approach is similar to the one in [8]. Both approaches assume that LR images have similar local geometry with HR images. The difference is that the former uses the two low-rank structure subspaces instead of the two overcomplete dictionaries of the latter.

In this paper, we propose a method on SISR, called LRSE, which performs multiple upsampling steps of relatively small magnification factors gradually to achieve the desired SR image without any external database. Our method is inspired by the success of the low-rank matrix recovery and the self-similarity of images for SISR. In our approach, a set of LR/HR patch pairs is first generated from the pyramid composed of different scale images of the input LR image. Based on the self-similarity of images and the scale invariance property of image singularities and for each unknown HR patch, we search its similar HR patches from the set of LR/HR patch pairs. Because these similar HR patches form an approximate low-dimensional

subspace, we utilize the LRMR technique to get the true low-dimensional subspace of these similar HR patches. Finally, the low-rank component of this subspace is exploited to recover the unknown HR patch.

The core idea of the proposed method is to find the low-dimensional (LD) subspace of some similar HR patches to an unknown HR patch using the LRMR algorithm, then utilize this LD subspace to recovery the unknown HR patch. Our method shares the same strategy with LRNE [27] where the LRMR technique is also used to gain the LD subspace of similar image patches. But there are three differences between LRNE and our method. First, we utilized the *k* nearest neighbor algorithm to find the similar LR/HR patch pairs inside image, both within the same scale and across different scales, in terms of the LR patch corresponding to the unknown HR one. Our method does not need any extra training database. Second, only the matrix made up of similar HR patches is decomposed by LRMR in our method, whereas the matrix made up of similar LR patches also need to be decomposed in LRNE. Third, the weighted coefficients of the low-rank component of matrix to recover the unknown HR patch are computed by the similarity degree of the corresponding LR version.

2. The Proposed Method

This section describes the details of our method based on the self-similarity of natural images and the low-rank matrix recovery.

2.1 Generating the Set of LR/HR Patch Pairs

Assuming that an observed LR image L is a blurred and downsampled version of the unknown HR image H ,

$$L = D_S(H * B) \qquad (1)$$

where B is an isotropic Gaussian kernel with the variance σ^2 , D_S denotes a downsampling operator with scale factor S , and *is a convolution operator. The observed LR image and the unknown HR image are viewed as our input LR image and the corresponding target HR image, respectively.

Let I_i denote an unknown HR image with the small magnification factors $R_i = s^i$ from the image L using Gaussian kernel B_i with variance σ_i^2 ($i = 1,2,...,n$), where s is a constant, $n = \log(S)/\log(s)$ that denotes the number of HR images to be reconstructed for obtaining the target HR image H . Thus I_i satisfies:

$$L = D_{R_i}(I_i * B_i) \quad (i = 1,2,...,n) \qquad (2)$$

where D_{R_i} denotes a downsampling operator with scale factor R_i. Yang et al. [16] proved the proposition that for any two downsampled images $I_i = D_{R_{n-i}}(H * B_{n-i})$ and $I_j = D_{R_{n-j}}(H * B_{n-j})$ of the image pyramid, the variances of their Gaussian kernels are related by $\sigma_i^2 = \sigma_j^2 \cdot \log(R_{n-i}) / \log(R_{n-j})$. According the aforementioned proposition, we can know that $\sigma_i^2 = \sigma^2 \times (n-i)/n$. The yellow images represent the sequence of unknown HR images in Fig. 1.

For obtaining the set of LR/HR patch pairs, a sequence of LR images $I_{-1}, I_{-2}, ..., I_{-m}$ is generated from L through the following degradation process using the same blur Gaussian kernel B_i:

$$I_{-i} = D_{R_i}(L * B_i) \quad (i = 1,2,...,m) \qquad (3)$$

For convenience, let $I_0 = L$. The blue images represent the sequence of LR images in Fig. 1.

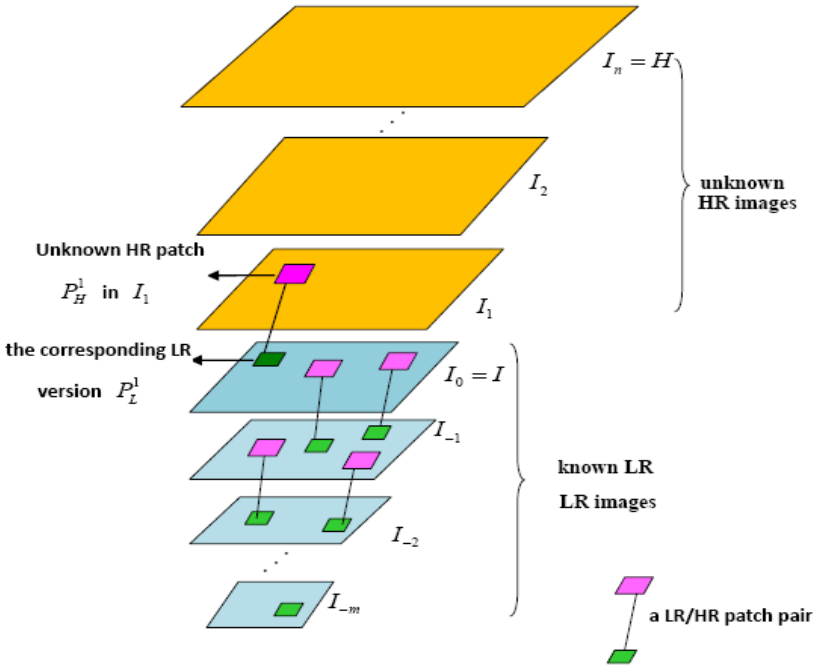

Fig.1. Pyramid of input image and similar LR/HR patch pairs. The light pink squares represent the similar LR patches of the dark pink square P_H^1, and the light green squares represent the similar patches of the dark green square P_L^1, respectively.

The above degraded images $I_{-1}, I_{-2}, ..., I_{-m+1}$ and I_0 are all subdivided into small patches in raster-scan order. For each patch P_{-i} in I_{-i} $(i = 0,1,2,...,m-1)$, its corresponding LR patch $P_{-(i+1)}$ is computed from the blurred image $I_0 * B_{i+1}$ with decreasing resolution by the scale factor s^{i+1}. The LR patch $P_{-(i+1)}$ and its HR version P_{-i} form a LR/HR patch pair. A green patch and a pink one connected by a line form a LR/HR patch pair in Fig. 1.

The set of LR/HR patch pairs includes all the LR/HR patch pairs. Note that the LR

patches in the HR/LR patch pair of our approach are different from those in [8, 16, 17] that are the bicubic interpolation-based upsampled versions of the LR patch. In [14] and our methods, the position of the corresponding LR version of each unknown HR patch in known LR image can be calculated by the blur kernel, and then the corresponding LR version can be obtained instead of using the bicubic interpolation-based upsampled versions. While the LR/HR patch pairs are different from those in [14], the LR/HR patch pairs in our method are from two adjacent scale images with the same blur kernel. Meanwhile, those patch pairs in [14] may come from two images with different scale gaps, so that LR/HR patch pairs with different scale gaps will have different supports.

Subsequently, these LR/HR patch pairs are used to estimate the unknown HR image I_1 by a reconstruction method that will be introduced in Section 2.2. Once the image I_1 is obtained, the set of LR/HR patch pairs is updated by combining the existing set of LR/HR patch pairs with all the LR/HR patch pairs obtained from the image I_1 and its corresponding LR version I_0. Then the unknown HR image I_2 is estimated by this new set of LR/HR patch pairs. This process is repeated until the target HR image H is obtained.

2.2 Estimating HR Patches Using Low-Rank Matrix Recovery Technique

Based on the idea of [26] that exploits the dependencies among similar patches to estimate their low-dimensional subspace by the SVT algorithm, we estimated the unknown HR patches from the set of generated LR/HR patch pairs for the input LR patch using the LRMR technique. The details are described as follows.

Let P_H^i denote a small patch of the unknown HR image I_i $(1 \le i \le n)$ (provided that the images $I_1,...,I_{i-1}$ have been obtained). First, the corresponding LR version P_L^i of P_H^i is computed according to the position of P_H^i. Then the approximate nearest neighbor search [29] is used to search K nearest LR patches $\{N_{Lk}^i\}_{k=1}^K$ of P_L^i from the set of LR/HR patch pairs. The corresponding HR patch N_{Hk}^i of the k-th nearest LR patch N_{Lk}^i is viewed as the k-th similar patches of P_H^i. Fig. 1 shows an example of the similar LR/HR patch pairs

These K similar HR patches in $\{N_{Hk}^i\}_{k=1}^K$ are stacked as columns of a matrix A in ascending order of the distance between the corresponding LR patch $\{N_{Lk}^i\}_{k=1}^K$ and the LR patch P_L^i. All columns in A are the K nearest neighbors of the unknown HR patch P_H^i and they have similar structures. Therefore, the matrix A has a low rank property and can be viewed as an approximate low rank matrix. The LRMR technique is used to decompose A into a low-rank matrix and a sparse matrix, where the low-rank matrix naturally corresponds to the true LD subspace of these similar HR patches and the sparse matrix represents the specifics of each HR patch and noise. The above decomposition can be formulated as the following convex

optimization problem,

$$\min_{A_l, A_s} \|A_l\|_* + \lambda \|A_s\|_{2,1} \quad (4)$$

$$\text{s.t. } A = A_l + A_s$$

where A_l is a low-rank matrix, A_s is a sparse matrix and λ is a positive weighting parameter, let $\lambda = 0.2$ in our method. $\|\cdot\|_*$ denotes the nuclear norm of a matrix (i.e., the sum of its singular values). The nuclear norm was adopted to enforce the low rank property of A_l. $\|\cdot\|_{2,1}$ denotes the sum of thel_2 norm of each row vector of a matrix. The $l_{2,1}$ norm, which is rotation invariant compared with the traditional l_1 norm [30], was used as regularization terms to impose the sparsity of matrix A_s.The advantages of $l_{2,1}$ norm regularization are that it can select features across all data points with joint sparsity, and have better performance effect in feature selection[31], classification [32], and so on. We used the inexact augmented Lagrange multipliers (ALM) algorithm [33] to solve the above optimization problem (4) which is referred to as Robust PCA (RPCA) [34].

The low-rank matrix decomposition does not change the identities of columns, so that the *k*-th column of matrix A_l, denoted by A_{lk}, is the low-rank component of N^i_{Hk} that is the *k*-th nearest neighbor of P^i_H.The weighted average of all the columns of A_l is reshaped to a small matrix as an estimate of pixels of the unknown HR image patch P^i_H. The weight of the *k*-th column is defined as $w_k = \frac{1}{Z} e^{-\frac{\|P^i_L - N^i_{Lk}\|^2_F}{h^2}}$, where Z is the normalizing constant $Z = \sum_k e^{-\frac{\|P^i_L - N^i_{Lk}\|^2_F}{h^2}}$, and the parameter h controls the decay of the exponential function. The closer the found similar LR patches N^i_{Lk} are to the corresponding LR version P^i_L of P^i_H, the large the weights of columns A_{lk} that correspond to N^i_{Lk} are. The basic step to generate the unknown HR patch is illustrated in Fig. 2.

After all patches of the unknown HR image I_i are obtained, the initial image of I_i is generated by simply averaging the pixels in the overlapped regions.

2.3 Enforcing Global Reconstruction Constraint

The target HR image is gradually synthesized with a relative magnification factor s. To ensure that the obtained HR image I_i in each upsampling process is consistent with the input LR image L, the back-projection technique [8] is used to project the initial image onto the input LR image L. Let I^0_i denote the initial image of I_i. The final reconstructed image is obtained from:

$$I_i^* = \arg\min_{I_i} \left\| D_{R_i} B_i I_i - L \right\|_F^2 + \lambda \left\| I_i - I_i^0 \right\|_F^2 \qquad (5)$$

where D_{R_i} is the downsampling matrix, B_i is the blurring matrix, and λ is a balancing parameter. The solution of the optimization problem (5) can be obtained by gradient descent with the following iterative update rule:

$$I_i^{t+1} = I_i^t + \tau \left[B_i^T D_{R_i}^T (L - D_{R_i} B_i I_i^t) + \lambda (I_i^0 - I_i^t) \right] \qquad (6)$$

where I_i^t is the estimate of the HR image I_i after the t-th iteration, and τ is the step size of the gradient descent. The optimal solution I_i^* is used as the final estimate of the HR image I_i. This image is not only as close as possible to the initial estimate image I_i^0 also satisfies the global reconstruction constraint.

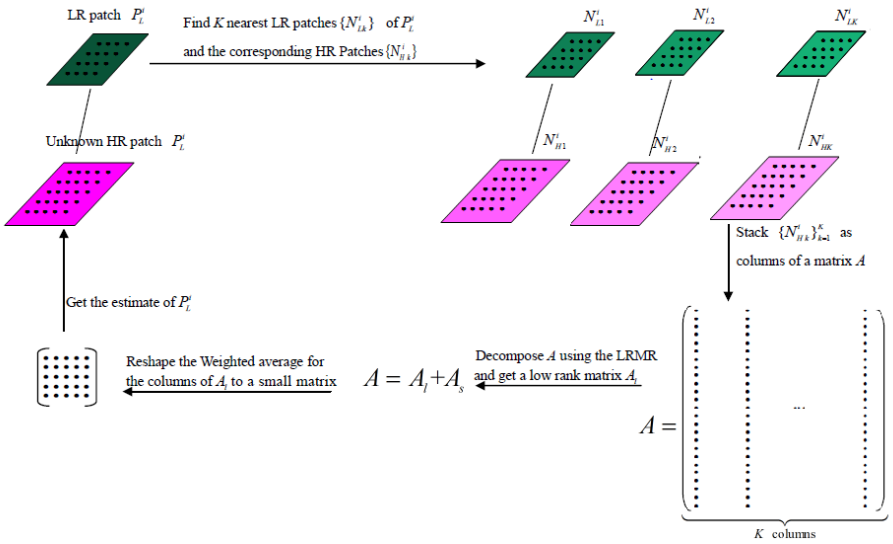

Fig. 2. The flowchart of our approach for generating the unknown HR patch. The constant scale factor s is set 1.25 in our algorithm and the sizes of HR image patch and LR one are 5×5 and 4×4, respectively.

After obtaining the n-th unknown HR image I_n, if the size of the HR image I_n is larger than the desired HR image, it will need to be downsampled to the desired size. The complete framework of our algorithm is summarized as Algorithm 1.

3 Numerical Experiments

3.1 Experimental Setup

We did three experiments to demonstrate the effectiveness of the proposed method. In the first two experiments, we selected ten images from the software package which is available on the website http://www.ifp.illinois.edu/~jyang29/ScSR.htm1 as original HR images (Fig. 3).

The original HR images were degraded into LR versions by bicubic interpolation or

Algorithm 1: SR via Self-Similarity and Low-Rank Matrix Recovery
1. Input: LR image L, magnification factor S, Gaussian kernel variance σ^2, the number of nearest neighbors K, back-projection loop number T, the step size of the gradient descent τ, the number of downsampled LR images m, the number of upsampled unknown HR images n.
2. //Constructing the pyramid of image L // Set $I_0 = L$; For $i = 1,2,..,m$ do Compute convoluted image C_{-i} by $L * B_i$; Compute LR image I_{-i} by $D_{R_i}(C_{-i})$; End

3. //Generating the Set of LR/HR Patch Pairs//
For $i = 0,1,...,m-1$ do
For each 5×5 patch P_{-i} in I_{-i} do
Compute the corresponding region E_{-i} in $C_{-(i+1)}$;
Compute $P_{-(i+1)}$ by $D_{R_i}(E_{-i})$;
Save patch pair $(P_{-i}, P_{-(i+1)})$ into the set of HR/LR patch pairs Q;
End
End

4. //Estimating the unknown HR images //
For $i = 1,...,n$ do
//Estimating the unknown HR image I_i with magnification factor 1.25//
For each 5×5 patch P_H^i in I_i do
Compute the corresponding LR patch P_L^i in I_{i-1};
Search *K* nearest LR patches of P_L^i of in the set Q, the corresponding *K* HR patches construct a matrix *A*;
Decompose *A* into a low-rank matrix A_l and a sparse matrix A_s with ALM;
Reshape the weighted average of all the columns of A_l to 5×5 matrix as an estimate of pixels of P_H^i;
End
Generate I_i^0 by simply averaging the pixels in the overlapped regions;
Use the back-projection technique to reconstruct the unknown HR image I_i;
Update the set of HR/LR patch pairs Q;
End

5. If the size of HR image I_n is larger than the desired HR image, do Generate the desired HR image *H* by downsampling I_n; End
6. Output SR image *H*.

Gaussian blur kernels with standard deviation one which is introduced in [35]. In the third experiment, medical image *Chest*, text image *Eyetest*, animal image *Zebra*, plant image *Flowers* and human image *Baby* were used to test the performance of the proposed method from the view of visual effect. Because humans are more sensitive to luminance component, we only performed image super resolution on the luminance component of color images, and

the chromaticity fields were upsampled by bicubic interpolation.

For comparison, several representative SISR methods: bicubic interpolation (BI), sparse coding SR algorithms (SCSR) [8], self-exemplar SR algorithms(SESR) [14], anchored neighborhood regression (ANR) [10], deep convolutional neural network (SRCNN) [11], a rapid and accurate image super resolution (RAISR) [22] and LRNE[27] were implemented. The SCSR, ANR, SRCNN, RAISR and LRNE use external database for the set LR/HR patch pairs, whereas our method does not use any external database. We compared our approach with SESR, because both approaches gradually reconstruct HR image from only the input LR image without using external database and both exploit the multi-scale self-similarity property of images. The reason that we made comparisons between LRNE and our method is that both of them use the LRMR technique for SISR.

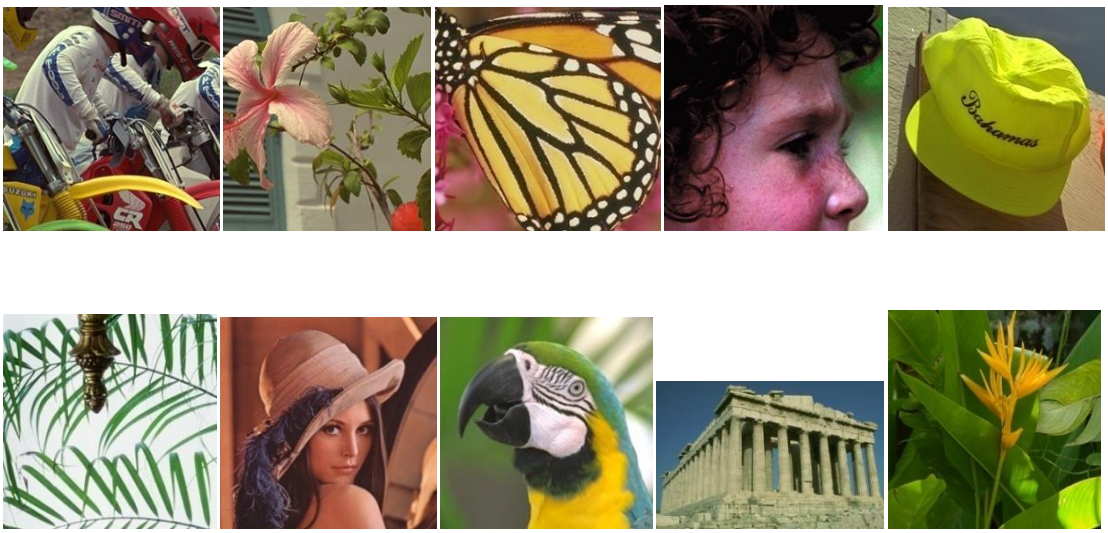

Fig.3. Original HR images in the first two experiments

3.2 The implementation of our proposed algorithm

We implemented our algorithm in MATLAB R2013, where the approximate nearest neighbor (ANN) search was implemented using the software developed by Mount et al.[36] and Bagon et al. [37] and the ALM algorithm's codes developed by Lin [33] were adopted.

In the proposed algorithm, the patch size was set to 5×5, the overlapping size was set to 4, the back-projection loop number $T=3$ and the step size of the gradient descent $\tau=1$. To avoid producing images with insufficient resolution, the number of the downsampled LR images from the input LR image was set to 3.The number of generated HR images was $n=\lceil \log(S)/\log(s) \rceil+1$, wherein S is the magnification factor for the target HR image, and s is set to 1.25 which is same as one in [14]. The parameter h that controls the decay of the exponential function in computing weights was set to 100.

The value of neighbor size *K* plays an important role in *k*-nearest neighbor. To investigate how the neighborhood size affects performance of our algorithm, our algorithm was implemented by tuning the neighborhood size from 8 to 24 with step of 4. It was observed that the reconstruction quality of the proposed algorithm is best when *K* is 16.

Table 1. The PSNR (dB) and SSIM values of the reconstructed HR images using different methods with a magnification factor of 3. The input images were downsampled from the initial HR image using bicubic interpolation. Results from [27] were directly used.

Images	BI	SCSR	SESR	ANR	SRCNN	RAISR	LRNE	Our method
Bike	22.808	24.008	24.360	23.978	**24.622**	23.906	23.121	24.479
	0.6832	0.7565	0.7737	0.7529	**0.7799**	0.7422	0.7499	0.7773
Butterfly	24.053	26.172	28.065	25.912	27.801	25.570	26.224	**28.101**
	0.7922	0.8753	0.8985	0.8696	0.9031	0.8186	0.8780	**0.8987**
Flower	27.456	28.708	29.137	28.671	**29.279**	28.628	28.291	29.176
	0.7753	0.8224	0.8361	0.8193	**0.8349**	0.8173	0.8177	0.8381
Girl	32.699	33.416	33.287	33.435	**33.517**	**33.457**	33.018	33.427
	0.7275	0.7275	0.7493	0.7175	0.7219	0.7529	0.8080	0.7532
Hat	29.197	30.492	31.221	30.365	30.965	30.145	30.419	**31.225**
	0.8003	0.8337	0.8432	0.8360	0.8406	0.8160	0.8305	**0.8433**
Leaves	23.452	25.346	**27.536**	25.398	26.604	25.216	25.161	27.502
	0.7654	0.8652	0.9157	0.8624	0.9034	0.8393	0.8597	**0.9184**
Lena	30.099	31.329	31.871	31.255	32.129	31.245	**32.714**	31.906
	0.8137	0.8320	0.8451	0.8264	0.9034	0.8338	**0.9618**	0.8471
Parrots	28.096	29.608	29.806	29.360	**30.447**	29.681	28.973	30.030
	0.8715	0.8986	0.8971	0.8957	**0.9065**	0.8885	0.8942	0.8999
Parthenon	26.040	26.722	27.143	26.595	27.142	26.519	26.517	**27.151**
	0.6739	0.7117	0.7274	0.7196	0.7273	0.7013	0.7157	**0.7286**
Plants	31.085	32.281	33.424	32.589	33.522	32.494	32.315	**33.616**
	0.8507	0.8531	0.8639	0.8700	0.8822	0.8471	**0.8906**	0.8640
Average	27.4985	28.8082	29.5850	28.7558	29.6028	28.6861	28.6753	**29.6613**
	0.7754	0.8176	0.8350	0.8169	0.8403	0.8057	**0.8406**	0.8369

We used the source codes online at http://www.ifp.illinois.edu/~jyang29/ScSR.htm1for SCSR, http://www.vision.ee.ethz.ch/-timofter/ for ANR, http://mmlab.ie.cuhk.edu.hk/projects /SRC.html for SRCNN and https://github.com /HerrHao/RAISR for RAISR.The code for SESR algorithms is available at website https://eng.ucmerced.edu/people/cyang35. We used the dictionaries or model parameters in the source code for SCSR, ANR and SRCNN. The filter matrix in RAISR was trained using the sixty-nine training images in software package that can be found on http://www.ifp.illinois.edu/~jyang29/ScSR.htm1.

3.3Experimental Results

Each algorithm was evaluated by computing peak signal to noise ratio (PSNR) and structural similarity (SSIM). The PSNR index is defined as $10\log_{10}(255^2 / MSE)$, where *MSE* is the mean square error between the original image and the generated SR image. The SSIM

index is a full reference metric that is a dimensionless score ranging from zero to one. Higher the SSIM value is, more similar the reconstructed image is to the original image in structure [38].

First, since the input LR images were downsampled by bicubic interpolation from ten color original HR images (Fig. 3) with scale factor of 3 in [27], we generated the input LR images in the first experiment as in [27].

Table 2. The PSNR and SSIM values ofthe reconstructed HR images by different methods with a magnification factor of 3. The input images were downsampled from the initial HR image using Gaussian blurring.

Images	BI	SCSR	SESR	ANR	SRCNN	RAISR	Ourmethod
Bike	22.660	23.723	23.866	23.831	**24.221**	23.735	24.069
	0.6778	0.7466	0.7622	0.7476	**0.7672**	0.7390	0.7664
Butterfly	23.750	25.723	27.067	25.678	27.052	25.443	**27.384**
	0.7906	0.8501	0.8842	0.8721	**0.8955**	0.8217	0.8874
Flower	27.261	28.421	28.581	28.473	**28.910**	28.473	28.782
	0.7710	0.8165	0.8294	0.8162	0.8278	0.8147	**0.8316**
Girl	32.544	33.238	32.935	33.311	**33.344**	33.341	33.201
	0.7238	0.7481	0.7472	0.7134	0.7165	0.7488	**0.7498**
Hat	29.076	30.192	30.806	30.159	30.598	29.934	**30.868**
	0.8004	0.8254	0.8364	0.8342	0.8359	0.8148	**0.8373**
Leaves	23.139	24.950	26.146	25.125	25.951	25.003	**26.425**
	0.7882	0.8539	0.9090	0.8615	0.8919	0.8279	**0.9101**
Lena	29.739	31.035	31.263	31.087	**31.729**	31.101	31.444
	0.8116	0.8338	0.8364	0.8252	0.8252	0.8325	**0.8394**
Parrots	27.873	29.239	29.310	29.229	**30.097**	29.571	29.666
	0.8691	0.8893	0.8900	0.8941	**0.9028**	0.8874	0.8951
Parthenon	25.917	26.471	26.681	26.406	**26.855**	26.334	26.762
	0.6724	0.7056	0.7232	0.7147	**0.7330**	0.6970	0.7224
Plants	30.865	32.281	32.709	32.389	32.839	32.368	**32.993**
	0.8485	0.8531	0.8469	0.8687	**0.8776**	0.8461	0.8525
Average	27.2824	28.5273	28.9364	28.5688	**29.1596**	28.5303	29.1594
	0.77534	0.81224	0.82649	0.81477	0.82734	0.80299	**0.82920**

Table 1 describes the PSNR and SSIM values of the reconstructed HR images using different methods. The qualities of the reconstruction using SESR, SRCNN and our method are significantly better than BI, SCSR, ANR and LRNE in terms of PSNR. Furthermore, the average PSNR improvements are 0.0585dB, 0.0763dB and 0.986dB for our method against SRCNN, SESR and LRNE, respectively. Particularly the maximum of PSNR improvements is up to 2.286 dB for LRNE. On the other hand, the results of SESR, SRCNN, LRNE and the proposed method outperform those of BI, SCSR, ANR and RAISR in terms of average SSIM values. The average SSIM improvement is 0.0037 for our method against LRNE. Based on the above analysis, it is showed that the proposed method is more effective than the other methods in terms of PSNR, and has similar restoration quality to SESR, SRCNN and LRNE in terms of

SSIM for the input images downsampled from the initial HR image using bicubic interpolation with a magnification factor of 3.

Secondly, we further demonstrated the utility of the proposed method on the LR images downsampled by Gaussian blurring. The LR images were generated from the test images by Gaussian kernel with a standard deviation 1. The code of downsampled image is available at website https://eng.ucmerced.edu/people/cyang35.

Table 3 The PSNR and SSIM values ofthe reconstructed HR images by different methods with a magnification factor of 4. The input images were downsampled from the initial HR image using Gaussian blurring.

Images	BI	SCSR	SESR	ANR	SRCNN	RAISR	Ourmethod
Bike	21.459	21.542	21.324	**21.870**	21.850	21.671	21.742
	0.5908	0.6188	0.6329	0.6373	0.6534	0.6201	**0.6549**
Butterfly	22.202	22.714	22.952	23.085	**23.793**	22.781	23.588
	0.7174	0.7287	0.7988	0.7777	**0.8163**	0.7152	0.8131
Flower	25.881	25.953	25.746	26.233	**26.314**	26.077	26.227
	0.6992	0.7124	0.7296	0.7232	0.7347	0.7143	**0.7451**
Girl	31.398	31.334	30.786	**31.628**	31.599	31.569	31.245
	0.6828	0.6905	0.6833	0.6559	0.6577	0.6929	**0.6948**
Hat	27.881	28.037	28.224	28.331	28.537	27.967	**28.554**
	0.7613	0.7571	0.7782	0.7796	0.7754	0.7485	**0.7859**
Leaves	21.275	21.671	21.317	**22.127**	21.879	21.820	21.844
	0.6861	0.7117	0.7923	0.734**1**	0.7864	0.7073	**0.8057**
Lena	28.533	28.624	28.435	28.983	**29.364**	28.905	29.004
	0.7625	0.7574	0.7626	0.7578	0.7710	0.7642	**0.7788**
Parrots	26.431	26.589	26.165	26.685	**26.840**	26.795	26.737
	0.8328	0.8347	0.8373	0.8478	**0.8512**	0.8362	0.8502
Parthenon	24.909	24.856	24.685	24.883	25.116	24.834	**25.120**
	0.6138	0.6256	0.6408	0.6395	**0.6569**	0.6202	0.6474
Plants	29.340	29.537	29.506	29.939	30.049	29.699	**30.124**
	0.7984	0.7667	0.7645	0.7930	**0.7979**	0.7650	0.7832
Average	25.9309	26.0857	25.914	26.3764	**26.5341**	26.2118	26.4185
	0.71451	0.72036	0.74203	0.73459	0.75009	0.71839	**0.75591**

Tables 2 and 3 list the PSNR and SSIM values of the reconstructed HR images using different methods with magnification factor of 3 and 4 (except for LRNE), respectively. We did not make comparison with LRNE in this experiment since there is no result with Gaussian blurring in [27].

The PSNR values of our approach are obviously higher than SCSR, ANR and RAISR algorithms only except image *Girl*, respectively. Our method slightly outperforms SESR on all ten test images in PSNR. While the average PSNR values of our method and SRCNN are almost equal to 29.159, and higher than other five algorithms. According to the SSIM index, the average SSIM value of our method is highest among all seven algorithms. Note that SESR,

SRCNN and our method obtain similar average SSIM values. Especially, the SSIM values of our approach are higher than RAISE on all test images. In addition, our approach is superior to SESR except *Parthenon* and to SCSR and ANR algorithms except *Plants* (see Table2).

Table 3 shows that our approach obtains better results than BI, SCSR, SESR, AIR and RAISR in terms of average PSNR, and our approach outperforms all other six methods in terms of average SSIM. The PSNR values obtained by our approach are higher than the other

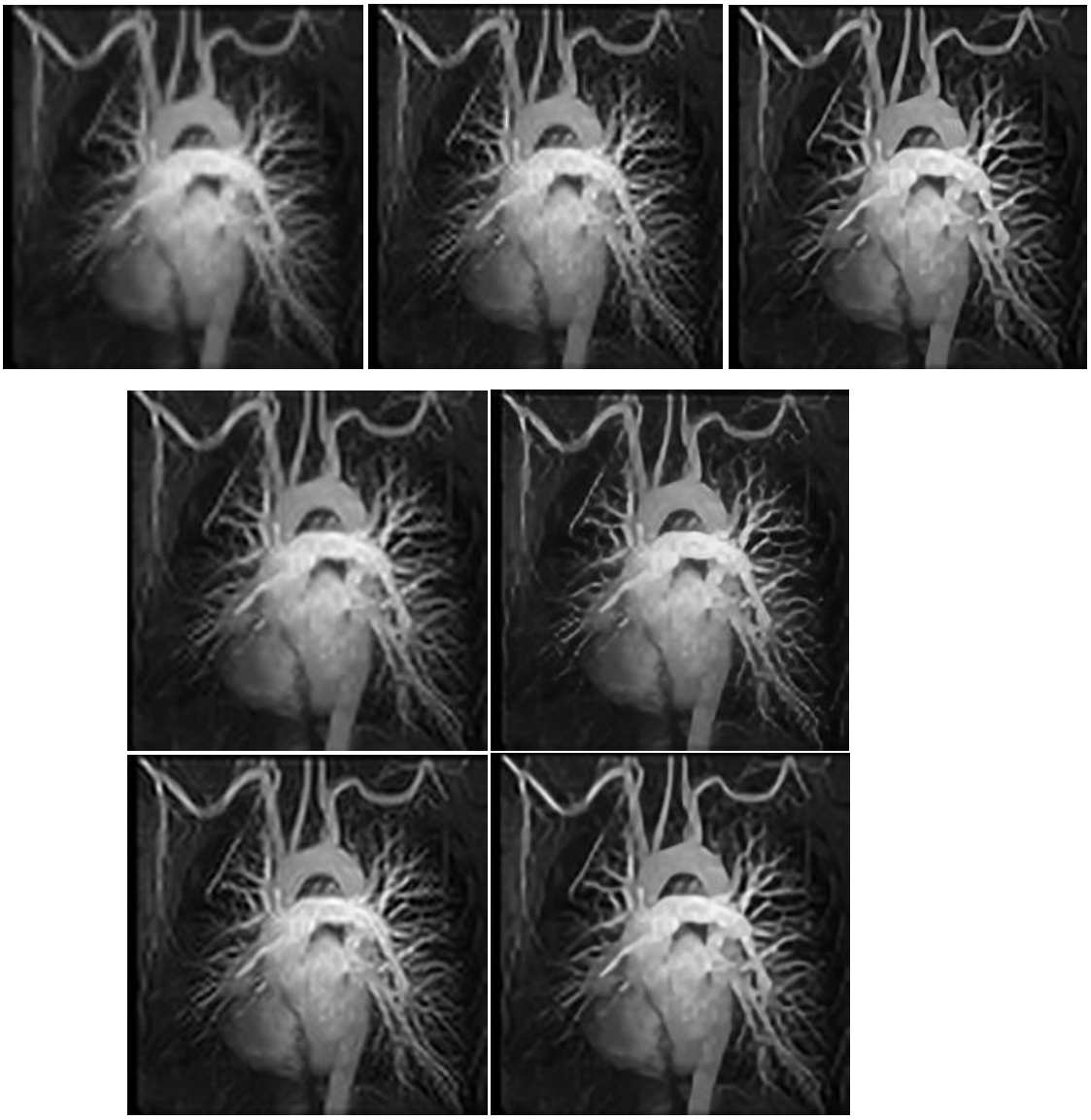

Fig.4. Comparison of the proposed method with the other methods on medical image *Chest* (magnified×3). Top row: BI (24.288/0.7108), SCSR(25.269/0.7719), SESR (25.285/0.7826); Middle row: ANR (25.190/0.7559), SRCNN (25.445/0.7786); Bottom row :RAISR (**25.544**/0.7789) and our method (25.391/**0.7832**).

algorithms on image *Hat*, *Parthenon* and *Plants*. SRCNN can obtain the highest PSNR values on image *Butterfly*, *Flower*, *Lena* and *Parrots*. ANR is superior to the other algorithms on image *Bike*, *Girl*, *Leaves* in terms of PSNR. As we can see, our approach can obtain the

highest SSIM values on image *Bike*, *Flower*, *Girl*, *Hat*, *Leaves* and *Lena*. SRCNN obtain the highest SSIM values on other four images.

The above analyses show that our algorithm and SRCNN are the most competitive among all seven methods on the input images downsampled by both bicubic interpolation and Gaussian blurring with a magnification factor of 3, according to quantitative measures in

Fig. 5. Comparison of our method with the other methods on text image *Eyetest* (magnified ×3). Top row: BI (18.961/0.7955), SCSR (20.309/0.8319), SESR (22.713/0.9179); Middle row: ANR (20.005/0.8235), SRCNN (22.1019/0.8836); Bottom row: RAISR (20.143/0.8194) and our method (**23.126/0.9225**).

PSNR and SSIM. In the meanwhile, for the input images downsampled by Gaussian blurring with a magnification factor of 4, ANR, SRCNN and the proposed approach are better than other four ones in terms of PSRN. According to SSIM, SRCNN and our methods obviously outperform other five algorithms.

We assessed the visual quality of the proposed method based on medical images *Chest*, text image *Eyetest*, plant image *Flowers*, animal image *Zebra* and human image *Baby*. The

Fig. 6. Comparison of our method with the other methods on plant image *Flowers* (magnified ×3). Top row: BI (25.097/0.7480), SCSR (26.299/0.7795), SESR (26.458/0.7803); Middle row: ANR (26.373/0.7854), SRCNN (**26.922/0.7905**); Bottom row: RAISR (26.228/0.7717) and our method (26.697/0.7883).

original HR images were downsampled by Gaussian kernel with variance 1 under scale factor 3. The visual effects of our method are obviously better than BI, SCSR, ANR and RAISR, and

similar visual effect as SESR and SRCNN, although the PSNR and SSIM values of these three methods are different (Figs. 4-8). These three better algorithms (SESR, SRCNN and the proposed algorithms) can produce sharper edges, especially SESR. The reconstructed images by BI are all very blurred. SCSR results in block artifacts and strong ring in the end of smaller branches in *Chest* (Fig. 4), the leaves and small flowers in *Flowers* (Fig. 6) and the leg and head of *Zebra* (Fig. 7).

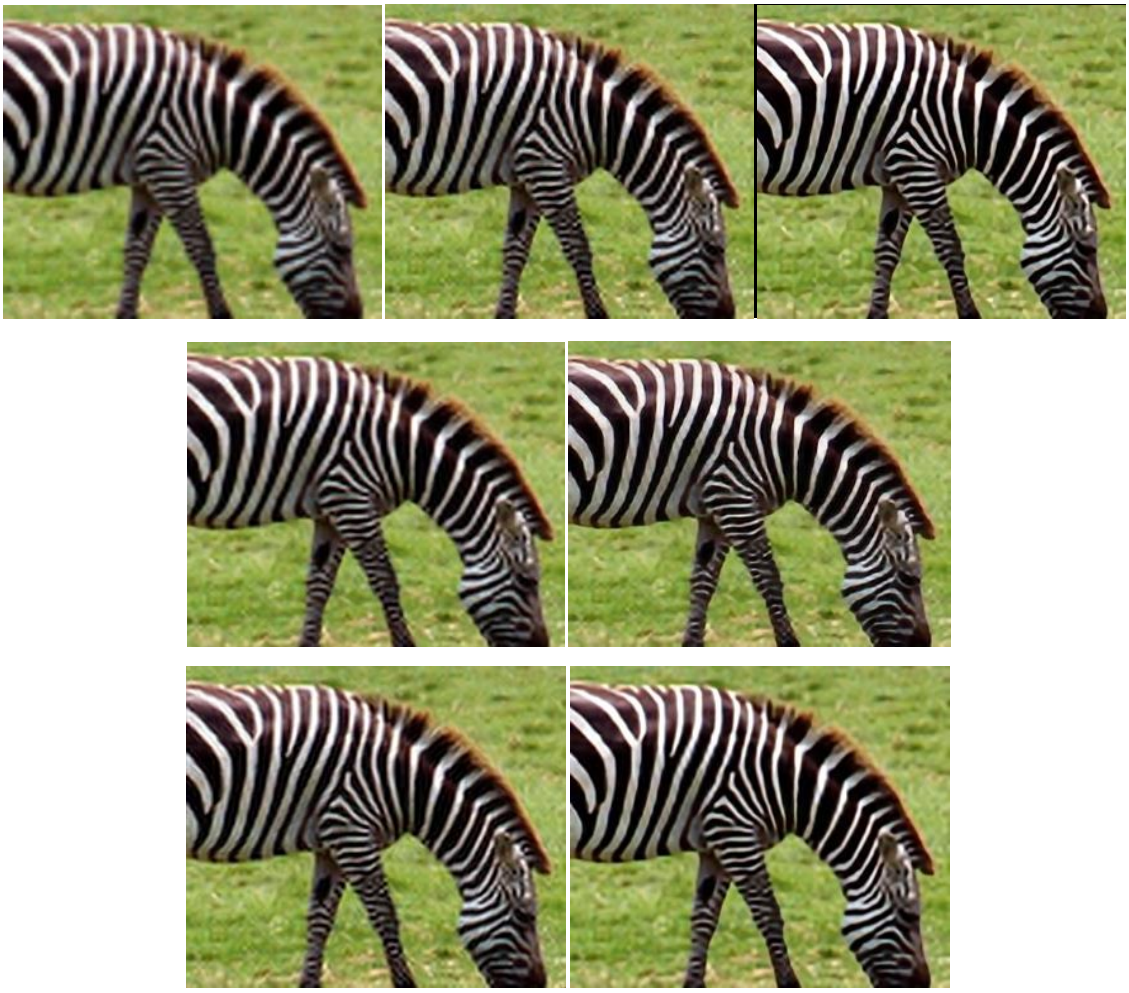

Fig. 7. Comparison of our method with the other methods on animal image *Zebra* (magnified ×3). Top row: BI (24.0301/0.7742), SCSR (26.115/0.8356), SESR (26.180/0.8473); Middle row: ANR (26.156/0.8405), SRCNN (26.567/0.8429); Bottom row: RAISR (26.464/0.8379) and our method (**26.587/0.8547**).

The images generated by ANR and RAISR are blurred in *Chest, Flowers and Zebra* (Figs. 4, 6, 7). For the text image *Eyetest*, the PSNR value of our method is significantly higher than the other six methods. The images reconstructed by SESR and our method are clearer than the other five methods on the letters of the last five lines (Fig. 5).

To summarize, we demonstrated that the proposed method, compared with SCSR, LRNE, ANR and RAISR, is more efficient, especially for the images with very similar image patches

(e.g. image *Butterfly*, *Leaves*, *Hat*, *Plants* and *Eyetest*). The reason lies in that there are some patches which recur within and across scales of image, so that self-similarity and SSC strategy play their part in our method. In contrast to SESR, the reason that our method outperforms SESR is that the LRMR based on similar patches can find the latent structure of subspace which the HR patches belong to and can remove the noise. In a word, the effectiveness of our method is attributed to the exploitation of both self-similarity of image and LRMR.

Fig. 8. Comparison of our method with the other methods on human image *Baby* (magnified ×3). Top row: BI (31.797/0.8295), SCSR (32.707/0.0.8569), SESR (32.125/0.8530);Middle row: ANR (32.8012/0.8559), SRCNN (32.746/0.8561); Bottom row: RAISR (**32.823**/0.8601) and our method (32.545/**0.860**2).

3.4Comparisons of Computation Times

We further compared the proposed method and other five methods (except BI) in terms of the CPU time. The SCSR, SESR, ANR, SRCNN, RAISR and our method require 400s, 187s, 91s, 19s, 38s, and 289s, respectively, to reconstruct an image of 256*256 pixels with a scaling factor of 3 on an Intel Core 2 Quad CPU with 2.67 GHz. The SCSR, ANR, SRCNN and RAISR algorithms need considerable extra time to train the LR/HR dictionaries or parameters of model. The above computation times of these four algorithms do not include their training times. The SESR and our method do not need to take any extra time to train the model. The proposed algorithm runs a longer time than other five algorithms, but the cost is tolerable in general, in particular compared to the gains in the effectivenessin the enhancement of resolution.

4. Conclusions

In this paper, we presented an algorithm exploiting the self-similarity of image and the low-rank matrix recovery for SISR. For obtaining an SR image, the scheme is developed to gradually magnify the LR input image to the desired size of HR image. Experimental results demonstrate the effectiveness of the proposed method compared with several representative SISR methods in terms of both quantitative metrics and visual effect. Our method works best on the image with more similar image patches within and across scales of image. One of the future research directions will be to improve the runtime of algorithm and apply our method to reconstruct SR-MRI from undersampled *k*-space data.

Acknowledgments

This work was supported by the National Science Foundation of China under Grant No. 61271374 and China Scholarship Council.

References

[1]S. Park, M. Park, and M. G. Kang. Super-resolution image reconstruction: A technical overview. IEEE Signal Processing Machine. 2003, 20(03): 21–36.

[2]Z. Lin and H. Shum. Fundamental limits of reconstruction-based super-resolution algorithms under local translation. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2004, 26(01) : 83-97.

[3]R. G. Keys. Cubic convolution interpolation for digital image processing. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1981, 29(6):1153–1160.

[4]W.T. Freeman, and E.C. Pasztor. Learning to estimate scenes from images. Adv. Neural Information Processing Systems, M.S. Kearns, S.A. Solla, and D.A. Cohn, eds., MIT Press, Cambridge, Mass., 1999, 11:775-781.

[5] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example learning-based super-resolution. IEEE Compuer Graphic Application. 2002, 22(02): 56–65.

[6] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2004, 1, 275-282.

[7] M. Bevilacqua, A. Roumy, C. Guillemot, and M.L. Alberi-Morel. Low complexity single image super-resolution based on nonnegative neighbor embedding. In: Proceedings British Machine Vision Conference. 2012, 135: 1-10.

[8] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. IEEE Transactions on Image Process. 2010, 19 (11): 2861–2873.

[9]R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse representations. In: JD. Boissonnat at al. (eds) Curves and Surfaces, LNCS. 2010, 6920: 711-730.

[10] R. Timofte, V. De Smet, and L. Van Gool. Anchored Neighborhood regression for fast example-based super-resolution. IEEE International Conference on Computer Vision. 2013, 1920-1927.

[11]C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In: Proceedings of European Conference on Computer Vision.2014, 184-199.

[12]D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. Huang. Robust single image super-resolution via deep networks with sparse prior. IEEE Transactions on Image Process. 2016, 25(7):1-14.

[13]M. Zontak, M. Irani. Internal statistics of a single natural image. IEEE Conference on Computer Vision & Pattern Recognition. 2011, 42(7):977-984.

[14]D. Glasner, S. Bagon and M. Irani. Super-resolution from a single image. IEEE International Conference on Computer Vision. 2009, 30(2):349–356.

[15]G. Freedman, and R. Fattal. Image and video upscaling from local self-examples. ACM Transactions on Graphics. 2011, 30(2): 1–11.

[16]C.Y. Yang, J.B. Huang, and M.H. Yang. Exploiting self-similarities for single frame super-resolution. Asian Conference on Computer Vision. 2010, 497–510.

[17]K. Zhang, X. Gao, D. Tao, and X. Li. Multiscale dictionary for single image super-resolution. IEEE Conference on Computer Vision & Pattern Recognition. 2012, 1114-1121.

[18]C. Y. Yang, and M. H. Yang. Fast direct super-resolution by simple functions. IEEE International Conference on Computer Vision. 2013, 561-568.

[19] K. Zhang, D. Tao, X. Gao, X. Li, and X. Xiong. Learning multiple linear mappings for efficient single image super-resolution. IEEE Transactions on Image Processing, 2015, 24(3):

846-861.

[20]M.C. Yang and Y.C. F. Wang. A self-learning approach to single image super-resolution. IEEE Transactions on Multimedia. 2013, 15(3): 498-508.

[21]F. Zhou, T. Yuan, W. Yang and Q. Liao. Single image super-resolution based on compact KPCA coding and kernel regression. IEEE Signal Processing Letters. 2015, 22(3): 336-340.

[22] Y. Romano, J. R. Isidoro, and P. Milanfar. RAISR: Rapid and Accurate Image Super Resolution. IEEE Transactions on Computational Imaging. 2017, 3(1):110-125.

[23]P. Sen, and S. Darabi. Compressive image super-resolution. Conference Record of the Forty-third Asilomar Conference on Signals, Systems and Computers. 2009, 1235-1242.

[24] N. Fan. Wavelet-based compressive super-resolution. Workshop on Applications of Computer Vision. 2009, 1–6.

[25]J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. IEEE International Conference Computer Vision. 2009, 2272-2279.

[26]W. Dong, G. Shi, and X. Li. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. IEEE Transactions on Image Processing. 2013, 22(2):700-711.

[27]X. Chen and C. Qi. Low-rank neighbor embedding for single image super resolution. IEEE Signal Processing Letters. 2014, 21(1):79-82.

[28] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions Image Processing. 2007, 16(8): 2080–2095.

[29]S. Arya, and D. M. Mount, Approximate nearest neighbor queries in fixed dimensions. Proceeding 4th Ann. ACM-SIAM Symposium on Discrete Algorithms, 1993, 271-280.

[30] C. Ding, D. Zhou, X. He, and H. Zha. R1-PCA: Rotational invariant *l*1-norm principal component analysis for robust subspace factorization. In: Proceedings of the International Conference Machine Learning. 2006, 281–288.

[31]C. X. Ren, D. Q. Dai, and H. Yan. Robust classification using *l*21-norm based regression model. Pattern Recognition. 2012, 45(7): 2708–2718.

[32]F. Nie, H. Huang, X. Cai, and C. H.Ding, Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In: Proceedings of the Advances in Neural Information Processing Systems. 2010, 1813–1821.

[33]Z. Lin, M. Chen, L. Wu, and Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC Technical Report UILU-ENG-09-2215, November, 2009.

[34]J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. Advances in Neural

Information Processing Systems.2009, 87(4):1-44.

[35]C. Y. Yang, C. Ma, and M. H. Yang, Single-image super-resolution: a benchmark. In: Proceedings of European Conference on Computer Vision, Zurich, Swiss, 2014.

[36]D.M.Mount, and S. Arya. ANN: a library for approximate nearest neighbor searching, 2006.

[37]S. Bagon. Matlab class for ANN, 2009.

[38]Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions Image Processing. 2004, 13(4): 600-612.