J. Vis. Commun. Image R. 20 (2009) 312–322

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Example-based image super-resolution with class-specific predictors

Xiaoguang Li [a,b], Kin Man Lam [b,*], Guoping Qiu [c], Lansun Shen [a], Suyu Wang [a]

[a] Signal and Information Processing Laboratory, Beijing University of Technology, Beijing 100124, China
[b] Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
[c] School of Computer Science, University of Nottingham, UK

ARTICLE INFO

Article history:
Received 22 May 2008
Accepted 31 March 2009
Available online 7 April 2009

Keywords:
Example-based super-resolution
Human face magnification
Content-based encoding
Class-specific predictor
Self-specific training set
Domain-specific training set
General-purpose training set
Vector quantization

ABSTRACT

Example-based super-resolution is a promising approach to solving the image super-resolution problem. However, the learning process can be slow and prediction can be inaccurate. In this paper, we present a novel learning-based algorithm for image super-resolution to improve the computational speed and prediction accuracy. Our new method classifies image patches into several classes, for each class, a class-specific predictor is designed. A class-specific predictor takes a low-resolution image patch as input and predicts a corresponding high-resolution patch as output. The performances of the class-specific predictors are evaluated using different datasets formed by face images and natural-scene images. We present experimental results which demonstrate that the new method provides improved performances over existing methods.

1. Introduction

Image super-resolution plays an important role in many multimedia applications. This term refers to the reconstruction of a high-resolution (HR) image from a single or a set of low-resolution images [1]. In this paper, we consider image super-resolution based on a single image; this is also called image magnification or image interpolation. The simplest method to enhance image resolution is by direct interpolation. However, this approach does not include any additional information for compensating the high-frequency content of the HR images to be constructed, which has been lost in the low-resolution (LR) images. A number of super-resolution algorithms [2–5] have employed regularization terms to solve the ill-posed image up-sampling problem. These algorithms usually incorporate smoothness priors as a constraint in reconstructing the HR images. However, using smoothness priors that are defined artificially has been found to lead to overly smoothed results [6,7]. Example-based or learning-based super-resolution algorithms [6–16] have been proposed recently as a very attractive approach for image super-resolution. Instead of defining a prior intuitively, this approach exploits the prior knowledge between the high-resolution and the corresponding low-resolution examples by learning algorithms.

Baker and Kanade [6,7] have demonstrated that the reconstruction constraint used in many regularization-based methods provides less and less useful information as the zooming factor increases. They proposed a "hallucination algorithm" to break the limit of the reconstruction constraint. To estimate the high-frequency components for a HR image, a multi-scale feature vector from a training set, which is composed of both LR details and the corresponding HR details, is searched as the best match, based on the LR patches from a LR image and the LR pixel values of the feature vector. Most example-based super-resolution algorithms [8–12] also involve a training set, which is usually composed of a large number of HR patches and their corresponding LR patches. The input LR image is split into either overlapping or non-overlapping patches. Then, for each LR patch from the input image, either one best-matched patch or a set of the best-matched LR patches is selected from the training set. The corresponding HR patches are used to reconstruct the output HR image. Freeman et al. [8,9] embedded two matching conditions into a Markov network. One is that the LR patch from the training set should be similar to the input observed patch, while the other condition is that the contents of the corresponding HR patch should be consistent with its neighbors. Wang et al. [10] extended the Markov network to handle the estimation of PSF parameters. Stephenson and Chen [11] presented a method in which the symmetry of a cropped human face is considered in the Markov network. Qiu [12] proposed an alternative method, based on vector quantization, to organize example patches. A survey of example-based super-resolution methods is available in [13].

The above-mentioned work has made significant contributions to the way we now exploit learning-based image super-resolution.

* Corresponding author. Fax: +852 23628439.
E-mail address: enkmlam@polyu.edu.hk (K.M. Lam).

1047-3203/$ - see front matter © 2009 Published by Elsevier Inc.
doi:10.1016/j.jvcir.2009.03.008

However, most of these existing algorithms involve only a kind of "searching and pasting" approach, and are therefore computationally intensive when searching for a LR–HR patch from a huge training set. Furthermore, best-matched but incorrect patches will seriously degrade the reconstruction results. To deal with these problems, usually the algorithms simply adopt the average of a set of the "best-matched" patches; the averaged high-frequency component is then pasted into the magnified image. For example, Qiu [12] employed the "classifying and averaging" scheme. However, the averaging will result in over-smoothing in the output HR image.

In this paper, we propose a new example-based super-resolution algorithm with a class-specific predictor so as to solve the above-mentioned problems in the existing algorithms. Inspired by Qiu's approach [12], we propose the class-specific predictor, which is a novel scheme to further improve the performance of the example-based super-resolution algorithms. In our algorithm, three questions are addressed: (1) How to generate a correlated and compact training set? Obviously, the training set is not simply a case of "the larger the better". Only those related training samples can provide useful information for high-frequency reconstruction. A large number of irrelevant examples will not only cost more searching time, but also disturb the search for the correct "best-matched" patches. In this paper, we will investigate the use of the self-example training set, a domain-specific training set, a general-purpose training set, and the combination of these two sets. (2) How to organize the set of training samples? A well-organized training set can speed up the searching or the training process, and therefore improve the efficiency of the algorithm. Inspired by the work of Qiu [12], a content-based encoding scheme is developed, which divides the space of training samples into several classes. (3) How to learn from the training set? We believe that "learning" should be more than just "searching and pasting". Therefore, a class-specific predictor is proposed for the reconstruction of high-frequency content for each class of LR patches.

In summary, the main contributions of this paper are: (1) a class-specific predictor is designed for each class in our example-based super-resolution algorithm – this can improve the performance in terms of visual quality and computational cost; and (2) different types of training set are investigated so that a more effective training set can be obtained.

2. Design and generation of training databases

The training set selected for use is important to the performance of the example-based super-resolution methods. Each record in the training set is an example patch-pair, viz. a HR image block and the corresponding LR block. Similar to the method proposed by Qiu [12], a multi-resolution representation of an input image is formed using a three-level Laplacian pyramid. As shown in Fig. 1(a), let I_0 represent a HR example image, which is blurred and down-sampled to produce I_1 by a zooming factor of z. Similarly, I_2 is generated from I_1 using the same zooming factor z. The up-sampled images from I_1 and I_2 are generated using bilinear interpolation with a factor z, and are then subtracted from I_0 and I_1, respectively, to compute the difference images L_0 and L_1. The example patch-pairs are then extracted from L_0 and L_1, which are then used to train up the corresponding class-specific predictors. For each block in L_0, there is a corresponding small block in the LR difference image L_1.

Fig. 1 (b) shows a HR difference image L_0 and its corresponding LR difference image L_1. If $z = 2$, each 4×4 HR block in L_0, e.g. the gray block, has a corresponding 2×2 LR block in L_1, viz. the black block. In order to maintain the continuity of a HR block with its neighbors, we extend the boundary of the corresponding LR block by 1 pixel to form a LR sampling block, i.e. the LR block in black and the neighboring pixels in gray in L_1, as shown in Fig. 1(b). This HR block and the corresponding LR sampling block thus form a patch-pair. By considering all the possible HR blocks in L_0 and the corresponding LR sampling blocks, a training set of patch-pairs is generated.

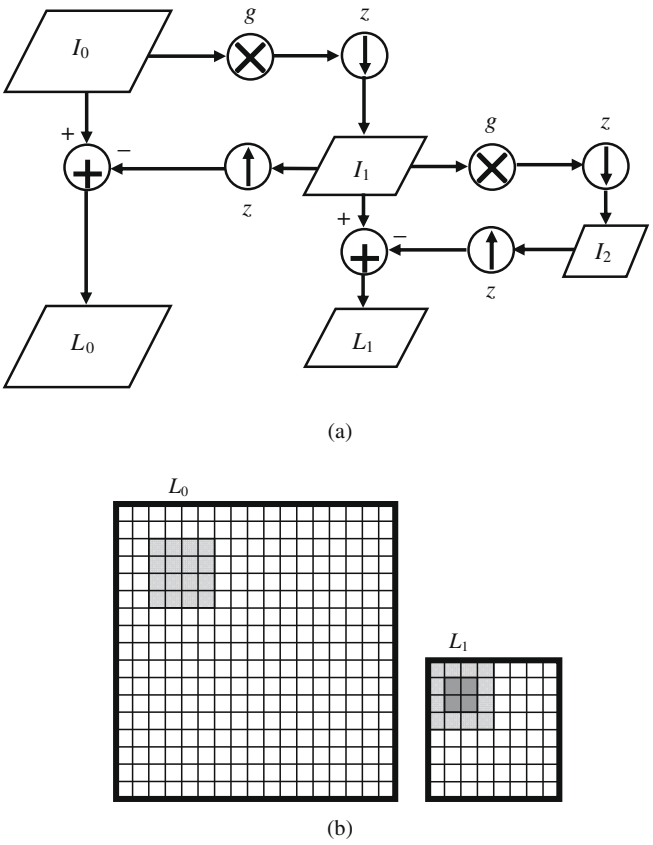

Fig. 1. Generation of the HR difference image L_0 and the LR difference image L_1 for the construction of HR–LR patches, and (b) a 4×4 HR block in L_0 and its corresponding 2×2 LR block in L_1.

In this paper, we will consider four types of training set for example-based image super-resolution:

(A) *Self-example training set (Set A):* An input LR image is taken as the image I_0 in Fig. 1(a), which is down-sampled to form images I_1 and I_2, so as to extract the training examples to be used. The contents obtained from self-examples should be more relevant to the input image itself, and so the number of required training examples should be much smaller than that based on other images. However, the generation of the training set and the training of specific-class predictors have to be performed on-line.

(B) *Domain-specific training set (Set B):* Images from a specific domain can be used to construct the training set. In this paper, we particularly consider facial images. Hence, the super-resolution of facial images based on our proposed algorithm will be evaluated. A face database is used for training, which can be performed off-line.

(C) *General-purpose training set (Set C):* Images of different types will be used to form a general-purpose database, which will be employed for training and used for super-resolution imaging. The training can be done off-line.

(D) *A combined training set (Set D):* This training set contains the examples from both Set A and either Set B or Set C.

3. Our proposed algorithm

Although a scene from the real world contains an abundance of varied content, a small local block in an image can be classified into just a few categories, such as flat, edge, corner, and so on. In our

algorithm, the classification is performed based on vector quantization (VQ), and then a simple and accurate predictor for each category, i.e. a class-specific predictor, can be trained easily using the example patch-pairs of that particular category. These class-specific predictors are used to estimate, and then to reconstruct, the high-frequency components of a HR image. Hence, having classified a LR patch into one of the categories, the high-frequency content can be predicted without searching a large set of LR–HR patch-pairs. The details of our algorithm are described in the following.

3.1. Content-based encoding/classification

To infer the high-frequency information of an estimated HR image effectively, the original LR image is divided into patches, which are classified into different categories. Those patches belonging to the same category have similar texture characteristics. A predictor can be designed for each category in order to estimate the high-frequency content of the patches.

In our algorithm, VQ is used to encode an input patch. The number of levels or codevectors in the codebook is the number of categories to be used. In other words, each category is represented by a codevector. Hence, a codebook must first be trained based on either the input image for self-example training or a number of images. Each training image is taken as I_0 in Fig. 1(a), which is converted into images I_1 and I_2 by means of Laplacian decomposition, as follows:

$$I_1 = s_z(g(I_0)) \quad \text{and} \quad I_2 = sz(g(I_1)) \tag{1}$$

where $g()$ and $s_z()$ represent the Gaussian operation and the subsampling operation with a factor of z, respectively. Then, the difference image L_1 is the difference between I_1 and I_2. This difference image is divided into a number of overlapping or non-overlapping blocks, and the corresponding HR blocks are then predicted. Following the work in [14], the block size is set at 4×4 in our implementation. The 16 elements of a block in L_1 are denoted as a vector, $\boldsymbol{b} = [b_0 \;\; b_1 \;\; \cdots \;\; b_{15}]^T$, which is transformed to have zero mean and unit variance, as follows:

$$\boldsymbol{x} = [x_0 \;\; x_1 \;\; \cdots \;\; x_{15}]^T, \quad \text{where } x_i = \frac{b_i - \mu}{\sigma^2}. \tag{2}$$

μ and σ^2 are the mean and the variance, respectively, of the 16 elements b_i. With this normalization, the encoding or the classification of the blocks will become more efficient. In our experiments, we will implement and evaluate our algorithms with and without performing the transformation (2). Assume that all of the training vectors are classified into N different categories. Then, a codebook containing N codevectors has to be constructed. The codebook is denoted as

$$CB = \{\boldsymbol{c}_i | \boldsymbol{c}_i \in R^{16}, i = 0, 1, \ldots, N-1\}. \tag{3}$$

Vector quantization is employed for implementing content-based encoding, whereby the LBG algorithm [17] can be used for constructing the codebook. This codebook can be determined in advance or off-line, except in the case of training based on self-examples. In the encoding process, the best-matched codevector $\boldsymbol{c}_j$ to an input LR block is determined, and the index j represents the category of the input block. The corresponding jth class-specific predictor will then be used to infer the high-frequency information.

All the training examples are encoded using the codebook. With the codebook for content-based encoding, each example patch-pair can be classified into one of the N categories. In other words, given a LR block of an example patch-pair after demeaning and normalization by (2), the closest vector is searched in the codebook. Then, the corresponding codevector is assigned to this patch-pair, where each codevector represents a category. Consequently, the training set is well structured with example pairs.

3.2. The class-specific predictors

As described in Section 2, different training sets are generated, which are in the form of HR–LR patch-pairs. Based on the LR part of the patch-pairs, a codebook is trained so that each patch from a LR image can be encoded, and hence identified to belong to one of the N categories. In other words, with a given training set, the LR part of each training patch is classified by content-based encoding. Hence, each category contains a number of HR–LR training patches. Now, the remaining question is how to learn from these training patches to help the reconstruction of high-frequency information? In our algorithm, a class-specific predictor will be trained for each category. Upon training up the predictor for a category, the prior knowledge of HR–LR relations is stored in the weights of the predictor. This scheme achieves the goal of "learning" from the training examples, rather than just performing "searching and pasting".

Fig. 2 shows the implementation of our algorithm, which is composed of a content-based encoder to classify the input LR patches, and a set of N class-specific predictors. The well-known least-mean-squares (LMS) algorithm is used [18] to train up the predictors. The input to a predictor is the 4×4 blocks of the difference images L_1, while the output is the corresponding predicted HR blocks of the central 2×2 patches of the input blocks, as described in Section 2. Therefore, the predictor output is given as follows:

$$y_i = \sum x'_j w_{ij}, \quad i = 0, 1, \ldots, 15, \tag{4}$$

where w_{ij} represents the weights for the ith predictor output value, and x' is an augmented representation of the input, as shown below:

$$\boldsymbol{x}' = [x_0 \;\; x_1 \;\; \cdots \;\; x_{15} 1]^T. \tag{5}$$

The weights are updated as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \boldsymbol{x}'_j (\boldsymbol{d}_i - y_i), \quad i = 0, 1, \ldots, 15, \;\; j = 0, 1, \ldots, 16, \tag{6}$$

where d is the vector of the targeted HR block, which comprises the corresponding HR patches of the patch-pairs in the training set, and η is the training rate ($0 < \eta < 1$). The training rate should be a small positive number, to ensure the convergence of the training of the weights in (6). The value of η will only affect the training speed, rather than predict quality, if the training converges. There is no systematic method for choosing η. In our experiments, we set

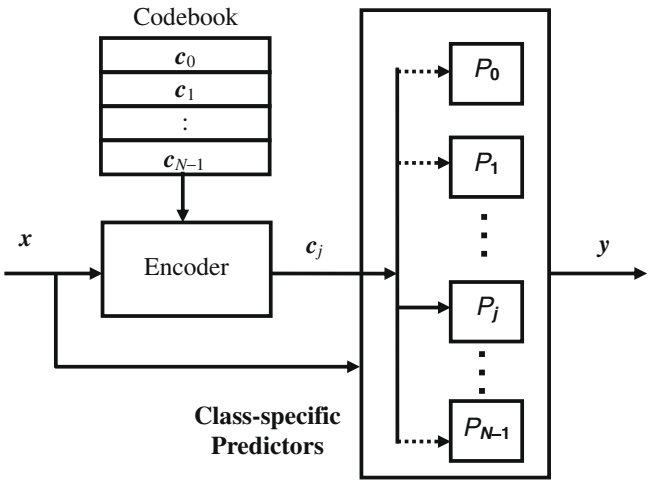

Fig. 2. A block diagram of our example-based image super-resolution algorithm, which is composed of a content-based encoder in the form of a vector quantizer, and a group of class-specific predictors to infer the high-frequency details.

$\eta = 0.005$. At the beginning of the training, all the weights are set at 0. Note that the N class-specific predictors can be trained simultaneously. In the case of using the self-example training set, the training must be performed online. Using the multi-threading programming technique can improve the efficiency of the training.

3.3. High-resolution image reconstruction

Having trained the content-based encoder and the class-specific predictors, the HR version of a LR image can be constructed. The input LR image is first magnified using the bilinear interpolation to form an initial estimation of its HR version, denoted as $\widehat{I}_0$. The high-frequency layer L_0 is estimated using one of the N class-specific predictors, and is then added to the initial estimated image to construct a HR image with high visual quality, i.e.

$$I_0 = \widehat{I}_0 + L_0. \tag{7}$$

Each 4×4 block B_h in the HR image has a corresponding 4×4 LR block B_l in the difference image L_1 of the input LR image. The central 2×2 patch of B_l is the low-resolution version of B_h. In our implementation, in order to handle those blocks at the boundary of L_1, all of the pixels at the boundary are extended and duplicated by one pixel. The block B_l is then encoded and classified to one of the categories, and the corresponding class-specific classifier is employed to infer the high-frequency information about B_h. Note that the reconstructed HR blocks have zero mean and unit variance, so they are transformed to have the original means and variances. In our algorithm, the HR block B_h is shifted by a step of 2 in the horizontal and vertical directions, and the corresponding LR block B_l is shifted by a step of 1 accordingly. At each position of the blocks, the high-frequency information is predicted using an appropriate class-specific predictor. Then, the overlapped high-frequency information is averaged to produce an estimation of the high-frequency layer.

Finally, the high-frequency layer is added to the initial estimated image, as in (7), and a LR constraint is also applied to the resulting image. We assume that the reconstructed HR image can produce the input LR image by smoothing and sub-sampling. The image I_0 is blurred and down-sampled to form the LR image I_1, as shown in Fig. 1(a). The average of a $z \times z$ block in I_0 will correspond to a single pixel in I_1. Suppose that the average value in I_0 and the corresponding single pixel values in I_1 are p_i and q_i, respectively. Then, the error is computed as follows:

$$e_i = q_i - p_i. \tag{8}$$

This error value is added to each pixel in the $z \times z$ block to reconstruct the final HR image.

To illustrate our proposed algorithm, Fig. 3 shows an example of the steps involved. In this example, all the training patch-pairs are clustered into four categories. Therefore, there are four codevectors, and four predictors are trained in the training stage, as described in Sections 3.1 and 3.2. The four codevectors, $\mathbf{c}_0$–$\mathbf{c}_3$, are also shown in Fig. 3. On the left of each codevector, a typical patch-pair of that class is provided. Given an input LR patch, the vector $\boldsymbol{x}$ is extracted using Eqs. (1) and (2). Then, $\boldsymbol{x}$ is encoded using the 4-level codebook, and the codevector $\boldsymbol{c}_0$ is the closest to $\boldsymbol{x}$ in this example. Therefore, the corresponding predictor P_0 is selected to predict the high-frequency layer. Finally, the high-frequency layer is added to the interpolated LR image to produce a good-quality HR patch.

4. Experiments and discussions

We will evaluate the performance of our proposed algorithm via the use of different training sets. Two different types of images will be considered in our experiments: face images and natural-scene images. For each type of training set, the optimal number of categories for content-based encoding will be determined, and both the visual qualities and the computational complexities of our algorithm, in combination with each of the different training sets, will be measured. The effect of the number of training samples used will also be investigated in relation to the domain-specific database and the general-purpose database. Furthermore, to evaluate

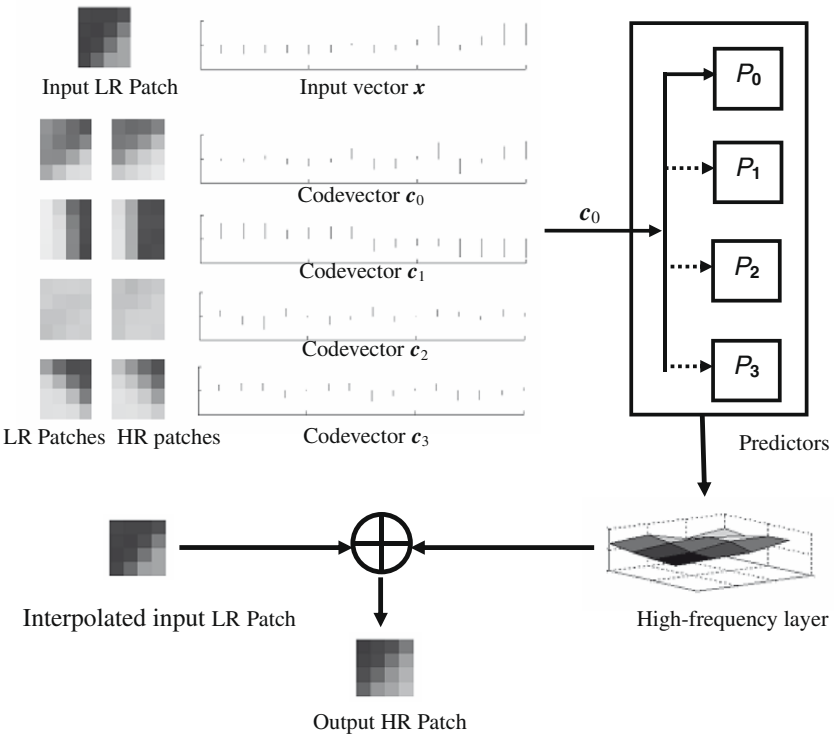

Fig. 3. An example showing the different steps of our proposed algorithm.

the performance of the class-specific predictors, we will also compare our proposed algorithm to Qiu's algorithm [12].

4.1. Training and testing images

For domain-specific applications, we consider the super-resolution of face images. Therefore, a number of face images and natural-scene images are used in the experiments. For the face images, the ORL database [19] is employed, which contains 40 distinct subjects, and each subject has 10 different images of size 92×112 pixels. Fig. 4 shows some of the face images from this database. In addition, to evaluate the performance of our algorithm for different types of images, a set of natural-scene images is used. The images have very different appearances to each other. For the self-example training set, the images themselves are used for training as well as for testing. Concerning the domain-specific training set and the general-purpose training set, a certain percentage of the face images and the natural-scene images, respectively, is selected for training, while the remainder will be used for testing. With different percentages of the images being selected for training, we can evaluate the effect of the number of training samples on the domain-specific training set and the general-purpose training set. In order to obtain a reliable measure of these performances, fivefold cross-validation will be employed.

Fig. 4. Some face images in the ORL database.

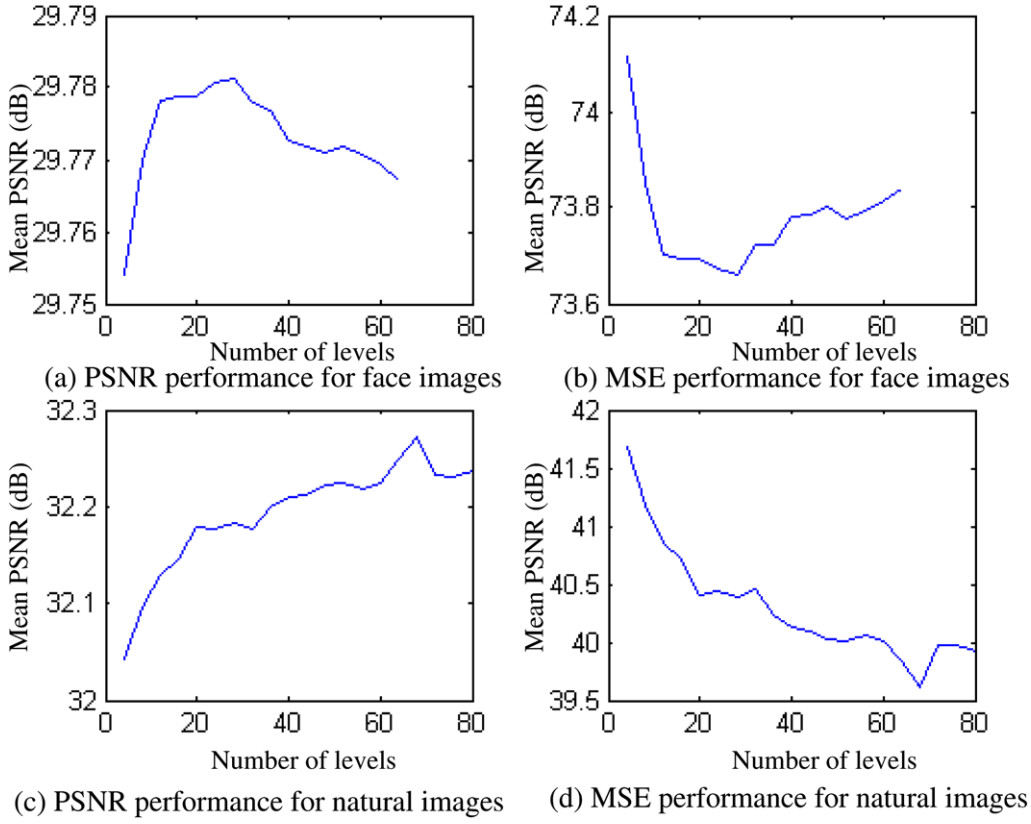

Fig. 5. PSNR and MSE with respect to different numbers of levels using the self-example training set.

4.2. Image super-resolution using self-example training set

In this section, we will measure the performance of our algorithm when applied to both face images and natural-scene images using the self-example training set, i.e. Set A, with different numbers of levels for content-based encoding. In the experiments, the original images are down-sampled horizontally and vertically by a factor of 2 to produce the LR images. These LR images are then processed using our algorithm to reconstruct the HR images. The PSNR (peak signal-to-noise ratio) and MSE (mean squared error) between the original images and the reconstructed HR images are measured.

Fig. 5 shows the PSNR and the MSE obtained by applying our algorithm to face images and natural-scene images, with different numbers of levels used in the content-based encoding. From the results, we can observe that our algorithm achieves the best performance with the face images when the number of levels used is about 28, while the natural-scene images require a greater number of levels, about 68. Figs. 6 and 7 illustrate some face images and natural-scene images, respectively, using different image super-resolution algorithms. Figs. 6 and 7(a) show the input LR images of size 46×56 for the face images, and 256×256 for the natural-scene images, which are down-sampled from the original HR images, shown in Figs. 6 and 7(b), respectively. The images in Figs. 6 and 7(c) are the results generated by bilinear interpolation. The results shown in Figs. 6 and 7(d) are based on Chen [14], and the results shown in Figs. 6 and 7(e) are based on Freeman [9], which uses the "searching and pasting" approach. We can see that the visual quality of these images is improved, to a certain extent, as compared to those achieved by bilinear interpolation. The mouth

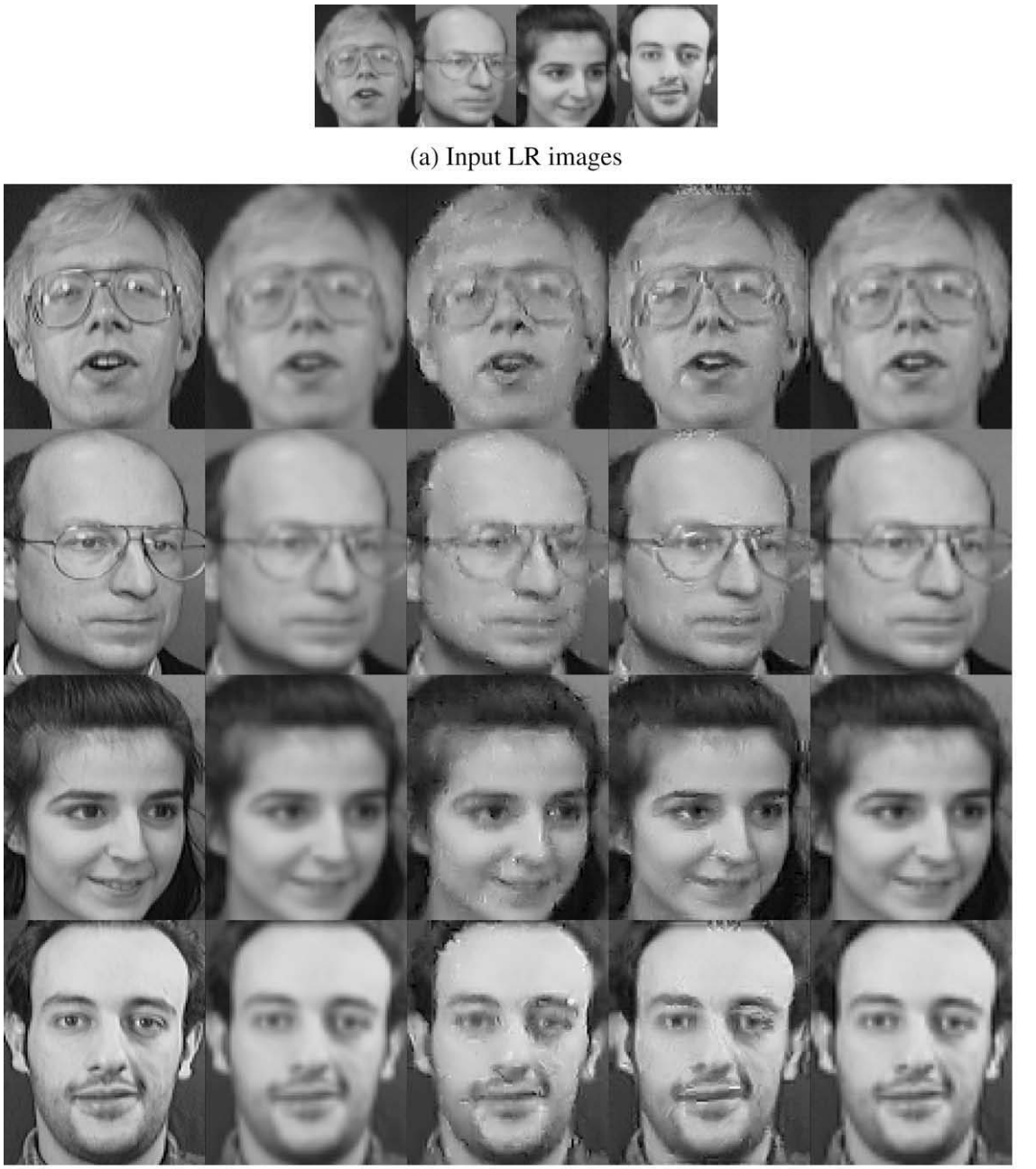

Fig. 6. Experimental results based on the ORL database.

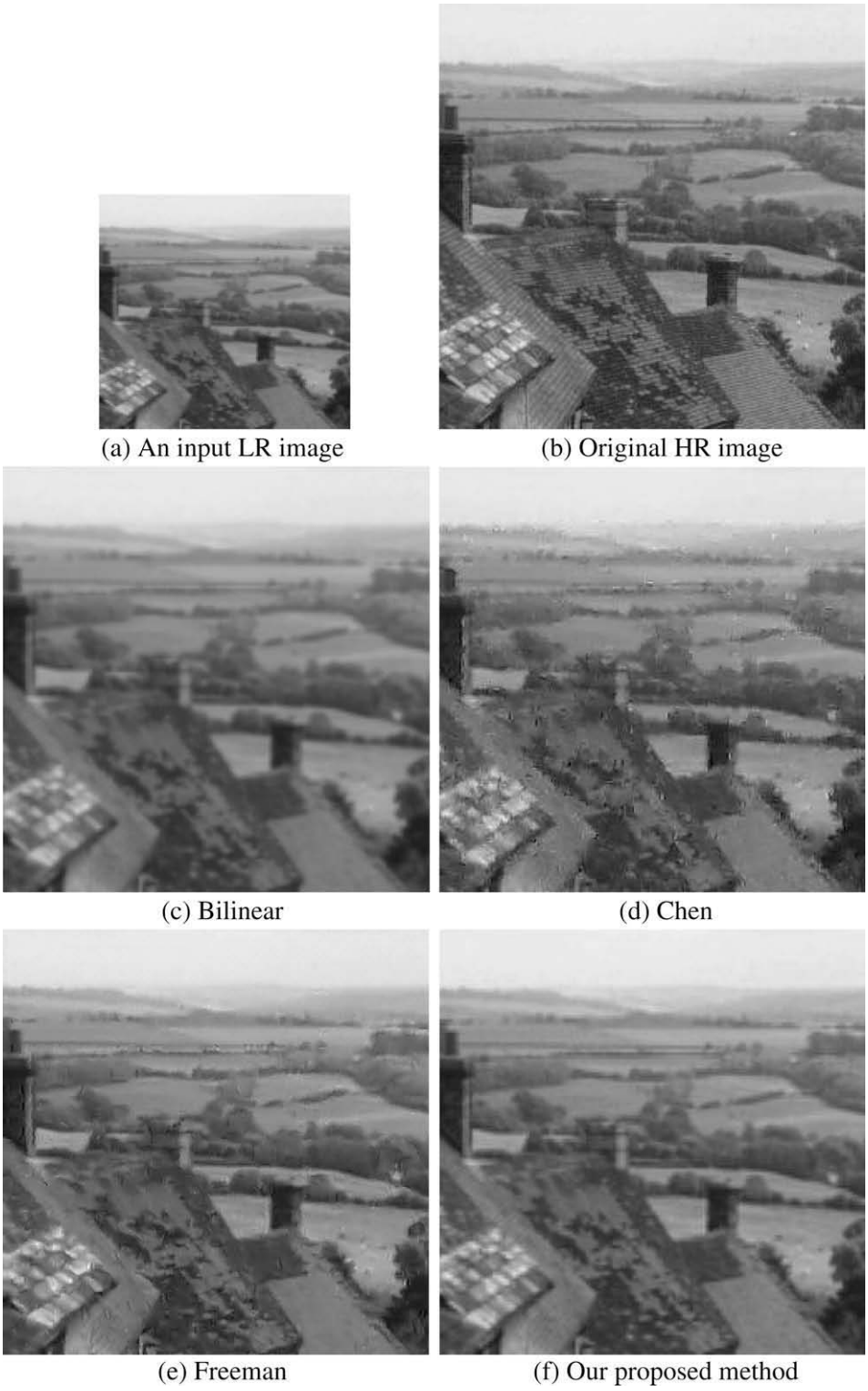

Fig. 7. Experimental results based on a natural-scene image.

region and the eye regions contain more high-frequency details. However, the unmatched patches for these regions will greatly degrade the reconstruction quality. The images in Figs. 6 and 7(f) are produced using our algorithm with the self-example training set. Because of the use of class-specific predictors in our algorithm, the unmatched problem can be avoided, and the image quality is improved. We have also noticed that the results based on our proposed method are blurrier than those shown in Figs. 6(e). This is due to the facts that the linear predictors may impose smoothing in the training patch space, and that the reconstructed high-frequency blocks which overlap are averaged. Therefore, more advanced predictors and post-processing techniques should be employed to overcome this drawback.

The PSNR and MSE are objective measurements of image quality, and they need not be consistent with subjective human visual perception. Table 1 tabulates the average PSNR, MSE, and runtime

Table 1
Performance of different algorithms.

Test images	Bilinear	Chen	Freeman	Our algorithm			
				Set A		Set B/C	Set D
				Without Pre-processing	With Pre-processing		
Face images							
PSNR(dB)	27.751	26.92	27.161	29.65	29.78	30.00	29.70
MSE	118.168	140.71	133.43	76.07	73.69	70.34	77.11
Runtime(s)	0.01	12.591	145	0.215	0.35	5.752	6.164
Natural images							
PSNR(dB)	30.70	28.61	30.35	31.96	32.25	32.34	32.02
MSE	64.64	90.18	61.212	42.45	39.84	39.14	41.83
Runtime(s)	0.015	82.021	439.943	5.693	7.203	6.146	6.614

of the different algorithms. The results obtained using our algorithms are based on the use of the optimal number of levels, i.e. 28 for face images and 68 for natural-scene images. The reason for this discrepancy is that the appearances of natural-scene images are quite different from each other, so a greater number of categories is needed to represent the variations. We can see that, with the self-example training set, our algorithm can achieve a smaller MSE, and therefore, a higher PSNR as compared to the other algorithms. The experiments were executed on an Inter® Core™ 2 CPU 6600 @2.40 GHz with 2 GB RAM system. Our algorithm can achieve a shorter runtime than other example-based algorithms for two reasons: the self-example training set is of a small size, with its content correlated; and the class-specific predictors can be designed in parallel by using multi-thread programming. As for the "searching and pasting" method, it requires searching a huge training set for each block of an input image, so it is more computationally intensive.

In order to show the improvement in performance when the image blocks are pre-processed by demeaning and unit-variance transformation before content-based encoding, our algorithm is also applied to the testing images – both with and without performing the pre-processing. These results are shown in Table 1. We can see that when demeaning and variance normalization is employed, the performance is improved. However, the runtime time will then increase slightly.

4.3. Image super-resolution using domain-specific training set (Set B) and general-purpose training set (Set C)

In this section, we will evaluate the performances of our algorithm with the use of a domain-specific training set and a general-purpose training set. In the experiments, we will change the number of levels for content-based encoding, and select different numbers of training images for the training of the codebook and the respective class-specific predictors.

Fig. 8 shows the PSNR and MSE of our algorithm with different numbers of levels for the codebooks. Half of the images are used for training, while the remaining half is for testing. Similar to the self-example case, the optimal number of levels for the face images is smaller than that for the natural-scene images. The optimal number of levels for face images is 32, and the optimal number of levels for natural-scene images is 68. These optimal numbers are very similar to or the same as the self-example cases. Although the runtimes required to train up the codebooks and the

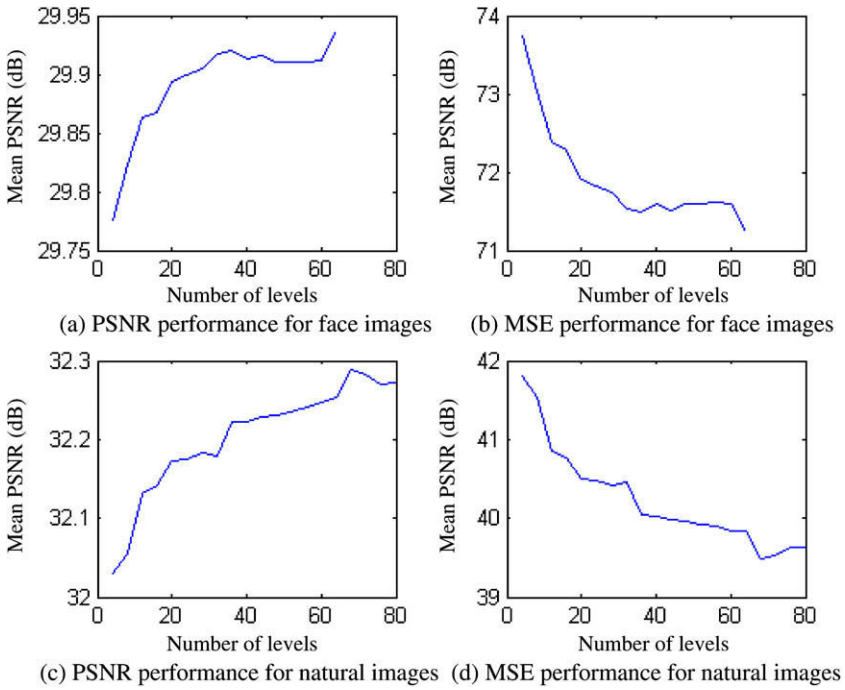

Fig. 8. PSNR and MSE with respect to different numbers of levels using the domain-specific training set and the general-purpose training set.

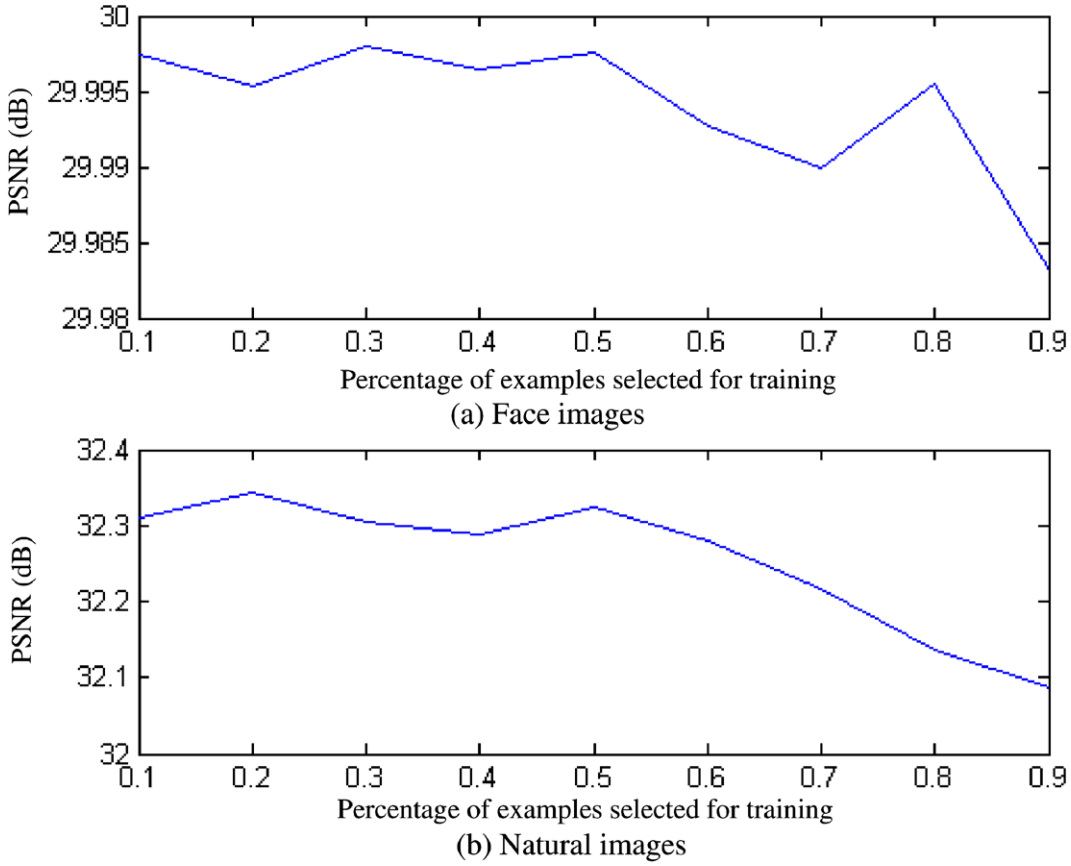

Fig. 9. PSNR performances of our algorithm with different numbers of training samples for (a) face images, and (b) natural-scene images.

class-specific predictors increase with the size of the training set, the training can be performed off-line.

Fig. 9 shows the effect of the size of the training sets on the performance of our algorithm. The percentage of images selected for training is changed from 10% to 90%. The respective remaining images are used for testing. When using the domain-specific training set, selecting 20% of the face images can produce a better result. For the general-purpose training set, selecting 50% of the images

Fig. 10. Other reconstructed HR images using our algorithm with combined training sets.

Table 2
Comparison of Qiu's algorithm and the proposed method using image "Lena".

	Bilinear	Qiu	Our algorithm
PSNR(dB)	26.5	27.4	33.77
MSE	39.8	32.4	27.28

Table 3
Performance of Qiu's algorithm and the proposed method under the same conditions.

Test images		Qiu	Our algorithm
Face images	PSNR(dB)	27.52	30.00
	MSE	123.90	70.34
Natural images	PSNR(dB)	30.02	32.34
	MSE	65.40	39.14

can achieve the best result. Once again, our proposed algorithm can produce the best visual quality. In Table 1, the average PSNR, MSE, and runtime of our algorithm using the two training sets – with the optimal number of levels for the codebook and the optimal percentages of the training sets employed – are tabulated.

4.4. Image super-resolution using combined training sets

In this section, we will evaluate the respective performances of our algorithm when the self-example training set is combined with the domain-specific training set and the general-purpose training set. The optimal number of levels for the codebooks and the optimal number of training samples obtained in Sections 3.1 and 3.2, respectively, are also employed in our experiments. Table 1 also tabulates the PSNR, the MSE, and the runtimes for using the combined training sets. Fig. 10 illustrates some of the HR images generated.

4.5. Comparison of our class-specific predictor and the interresolution look-up table

To further evaluate our proposed algorithm using class-specific predictors, its performance is compared to that of the interresolution look-up table algorithm proposed by Qiu [12]. Like our algorithm, Qiu's algorithm classifies all the patch-pairs in the training set into several categories by means of vector quantization. Then, a codebook for the LR patches is produced. For each LR codevector, a corresponding HR codevector is computed by averaging all the HR patches belonging to the same category. This forms an interresolution look-up table for reconstructing HR images. The major difference between these two methods is mainly in the method of predicting high-frequency contents: one uses the class-specific predictor, and the other uses the interresolution look-up table.

Table 2 tabulates the PSNR and the MSE of the two algorithms. The image "Lena", of resolution 256×256, is magnified to 512×512 using different methods. The results based on bilinear interpolation and Qiu's method were reported in [12]. Form the results we can see the superior performance of our algorithm. Using the same training set and the same number of codevectors (32 for face images and 68 for natural images), we evaluate the PSNR and MSE performances of the two algorithms. The testing images are those used in Sections 4.1–4.3. The average PSNR and MSE are tabulated in Table 3. Some reconstructed HR images are illustrated in Fig. 11. The images in Fig. 11(a) are blurrier than those shown in Fig. 11(b), which may be due to the use of the averaging scheme in the construction of the interresolution look-up table. However, the class-specific predictor can alleviate this effect.

5. Conclusion

The example-based approach is a promising way to solve the image super-resolution problem, which can provide the high-frequency contents of a reconstructed HR image by learning. However, most of the existing algorithms interpret the "learning" as just a kind of "searching" the best-matched LR patch, and then "pasting" the corresponding HR component. In our algorithm, we improve the learning by using a set of class-specific predictors, where the prior high-resolution information is stored as the weights of the predictors. The content of a training set is more important than its size. In order to exploit the efficiency and effectiveness of training sets, a self-example set, a domain-specific training set, and a combined set have each been investigated in experiments. Experimental results show that our algorithm can achieve an excellent performance in terms of both quality and computational complexity.

(a) Qiu's interresolution look-up table

(b) Proposed method

Fig. 11. Some reconstructed HR images using (a) Qiu's interresolution look-up table, and (b) our proposed method.

Acknowledgments

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5199/06E), and by the National Nature Science Foundation of China (60472036, 60431020, 60402036), Ph.D. Foundation of Ministry of Education (20040005015).

References

[1] S.C. Park, M.K. Park, M.G. Kang, Super-resolution image reconstruction: a technical overview, IEEE Signal Processing Magazine 5 (2003) 21–36.
[2] H.A. Aly, E. Dubois, Image up-sampling using total-variation regularization with a new observation model, IEEE Transactions on Image Processing 14 (10) (2005) 1647–1659.
[3] S. Farsiu, M.D. Robinson, M. Elad, et al., Fast and robust multiframe super resolution, IEEE Transactions on Image Processing 14 (10) (2004) 1327–1343.
[4] H. He, L.P. Kondi, An image super-resolution algorithm for different error levels per frames, IEEE Transactions on Image Processing 15 (3) (2006) 592–603.
[5] K. Chantas, N.P. Galatsanos, N.A. Woods, Super-resolution based on fast registration and maximum a posteriori reconstruction, IEEE Transactions on Image Processing 16 (7) (2007) 1821–1830.
[6] S. Baker, T. Kanade, Limits on super-resolution and how to break them, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 372–379.
[7] S. Baker, T. Kanade, Limits on super-resolution and how to break them, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (9) (2002) 1167–1183.
[8] W.T. Freeman, E.C. Pasztor, Learning low-level vision, International Journal of Computer Vision 40 (1) (2000) 25–47.
[9] W.T. Freeman, T.R. Jones, E.C. Pasztor, Example-based super-resolution, IEEE Computer Graphics and Applications 22 (2) (2002) 56–65.
[10] Q. Wang, X. Tang, H. Shum, Patch based blind image super resolution, In: Proceedings of the Tenth IEEE International Conf. on Computer Vision, Beijing, China, Oct. 2005.
[11] T.A. Stephenson, T. Chen, Adaptive Markov random fields for example-based super-resolution of faces, Journal on Applied Signal Processing 2006 (2006) 1–11.
[12] G. Qiu, Interresolution look-up table for improved spatial magnification of image, Journal of Visual Communication and Image Representation 11 (2000) 360–373.
[13] M. Elad, D. Datsenko, Example-based regularization deployed to super-resolution reconstruction of single image, The Computer Journal Advance Access 20 (2007) (published online on April).
[14] M. Chen, G. Qiu, K.M. Lam, Example selective and order independent learning-based image super-resolution, in: Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, pp. 77–80.
[15] X. Zhang, K.M. Lam, L. Shen, Image magnification based on a blockwise adaptive Markov random field model, Image and Vision Computing 26 (9) (2008) 1277–1284.
[16] M. Ebrahimi, E.R. Vrscay, Solving the inverse problem of image zooming using self-examples, in: M.S. Kamel, A.C. Campilho (Eds.), International Conference on Image Analysis and Recognition Lecture Notes in Computer Science, 4633, Springer, Berlin, 2007, pp. 117–130.
[17] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Transactions on Communications 28 (1) (1980) 84–95.
[18] G. Qiu, A progressively predictive image pyramid for efficient lossless for coding, IEEE Transactions on Image Processing 8 (1) (1999) 109–115.
[19] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision, Sarasota, FL, Dec. 1994.