Fast Super-Resolution Algorithm Based on Dictionary Size Reduction Using *k*-Means Clustering

Shin-Cheol Jeong and Byung Cheol Song

This paper proposes a computationally efficient learning-based super-resolution algorithm using *k*-means clustering. Conventional learning-based super-resolution requires a huge dictionary for reliable performance, which brings about a tremendous memory cost as well as a burdensome matching computation. In order to overcome this problem, the proposed algorithm significantly reduces the size of the trained dictionary by properly clustering similar patches at the learning phase. Experimental results show that the proposed algorithm provides superior visual quality to the conventional algorithms, while needing much less computational complexity.

Keywords: Super-resolution, training, dictionary, clustering, image-based, learning.

I. Introduction

Image interpolation is a key technology to display high quality up-scaled images on cutting-edge digital consumer applications such as high-definition television (HDTV), digital still camera (DSC), and digital camcorder. For several decades, a lot of single image interpolation algorithms have been discussed in the literature. They can be classified into three categories: interpolation-based, reconstruction-based, and super-resolution approaches. Firstly, the interpolation-based methods [1], [2] are computationally light and have simple structures in comparison with the others. However, they suffer from blurring and jagging artifacts in diagonal edges. Even the edge preserving interpolation methods as in [3]-[6] have a difficulty in synthesizing fine details. Secondly, reconstruction-based methods [7], [8] produce a high-resolution image under the constraint that the smoothed and down-sampled version of the reconstructed high-resolution image is close to the input low-resolution image. For example, the back-projection algorithm iteratively minimizes the reconstruction error. But, those algorithms rarely avoid jagging and ringing artifacts along the strong edges. Finally, so-called super-resolution (SR) algorithms [9]-[13] have been developed as the most promising approach. A typical SR image reconstruction makes use of signal processing techniques to obtain a high-resolution (HR) image (or a sequence) from multiple low-resolution (LR) images [9]. In general, success of such SR schemes depends on existence of sub-pixel motion between adjacent LR images and accurate sub-pixel estimation. However, sub-pixel motion estimation among neighbor LR images requires not only huge computational cost, but also its accuracy is not guaranteed in certain environments. In order to solve the above-mentioned problem, a lot of single image-based SR methods have been

Manuscript received Nov. 3, 2009; revised Feb. 26, 2010; accepted May 3, 2010.

This research was supported by the Ministry of Knowledge Economy (MKE) and Korea Institute for Advancement of Technology (KIAT) through the Human Resource Training Project for Strategic Technology, and was supported by National Research Foundation of Korea Grant funded by the Korean Government (2009-0071385).

Shin-Cheol Jeong (phone: +82 32 860 7413, email: shinchul61@naver.com) and Byung Cheol Song (corresponding author, email:bcsong@inha.ac.kr) are with the School of Electronic Engineering, Inha University, Incheon, Rep. of Korea.

doi:10.4218/etrij.10.0109.0637

devised such as example-based or learning-based SR algorithms [10]-[13]. They exploit the prior knowledge between the HR examples and the corresponding LR examples through the so-called learning process. Most example-based SR algorithms usually employ a dictionary 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 input LR patch, either one best-matched patch or a set of the best-matched LR patches are selected from the dictionary. The corresponding HR patches are used to reconstruct the output HR image. However, most of the existing algorithms are so-called 'searching and pasting' approaches, and are therefore computationally intensive in finding the best match of LR-HR patch from a huge dictionary. Furthermore, best-matched but incorrect patches will seriously degrade the reconstruction results.

This paper achieves fast image super-resolution by reducing the size of trained dictionary. At the learning phase, the number of LR-HR patch pairs in dictionary is noticeably reduced by grouping similar LR patches using *k*-means clustering. At the synthesis phase, each input LR patch is compared with the candidate LR patches in the dictionary one-by-one. So, one of the best-matched HR patches is selected from the dictionary. Subsequently, an HR residue patch is chosen from a residue dictionary. Finally, the corresponding HR patch and residue patch are used to reconstruct the output HR patch. Thus, the reduced dictionary size makes it possible to significantly speed up SR processing and save the memory cost, while providing reasonable visual quality.

The rest of the paper is organized as follows. Section II gives a brief overview of the existing learning-based SR. Section III describes the details of the proposed algorithm. Some experimental results are presented in section IV. Finally, we conclude in section V.

II. Learning-Based Super-Resolution

Figure 1 describes the basic concept of learning-based SR that is generally composed of two phases: an offline learning phase and an online synthesis phase. At the learning phase, the training data, that is, a dictionary consisting of LR and HR patches, is constructed. The LR and HR patch pairs are obtained from various training images. During the synthesis phase, the input LR image is super-resolved by using the dictionary. For each LR patch in the input image, its nearest neighbor LR patches are explored from the dictionary. The high-frequency components of the input LR patch are synthesized using the best matched LR patches.

Freeman and others [10] embedded two matching criteria

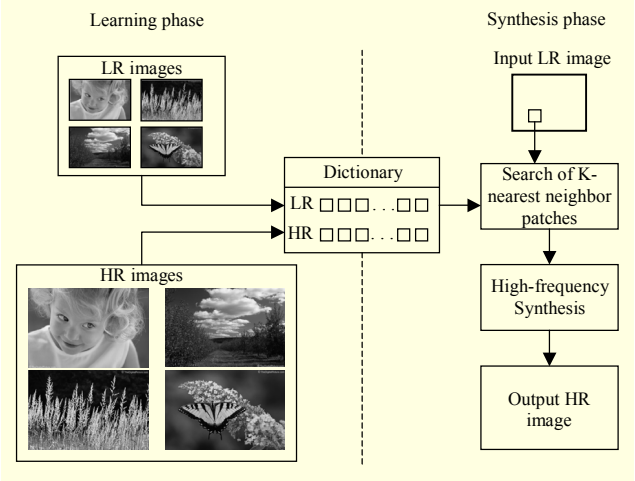

Fig. 1. Basic concept of learning-based super-resolution.

into a Markov network. One is that the LR patch from the dictionary should be similar to the input observed patch, while the other criterion is that the contents of the corresponding HR patch should be consistent with its neighbors. Chang and others [12] presented a neighbor-embedding-based SR algorithm which assumes that generation of a high-resolution image patch depends on multiple nearest neighbors in the dictionary. The algorithm finds the optimal reconstruction weights of the nearest neighbor patches and then estimates a proper HR patch by applying the weight to the corresponding HR patches.

The performance of those learning-based SR algorithms highly rely on matching accuracy of an input LR patch with candidate LR patches in the dictionary. In order to improve the accuracy of matching, a sufficient number of LR-HR patch pairs must be included in the dictionary. Usually, existing learning-based SR methods require hundreds of thousands of training examples for reliable performance. However, such a dictionary size causes tremendous memory cost for storing the training samples as well as awfully large computational complexity in the matching process. Therefore, it makes conventional learning-based SR impractical in implementation and restrictive in applications. In order to overcome this problem, we propose a fast learning-based SR algorithm with reduced dictionary based on *k*-means clustering.

III. Proposed Algorithm

The proposed algorithm accomplishes fast HR image reconstruction without degradation by reducing the dictionary size at the learning phase. Figure 2 describes the overall flow of our algorithm, which consists of a learning phase and a synthesis phase similar to conventional learning-based schemes. The learning phase includes preprocessing, patch extraction, dictionary size reduction, and ordinary dictionary construction

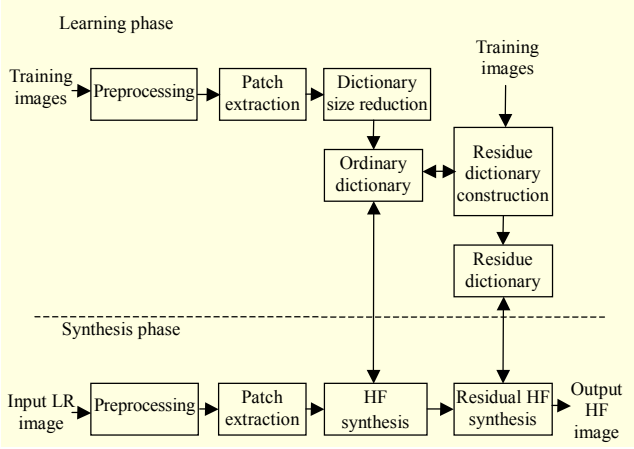

Fig. 2. Proposed learning-based SR algorithm.

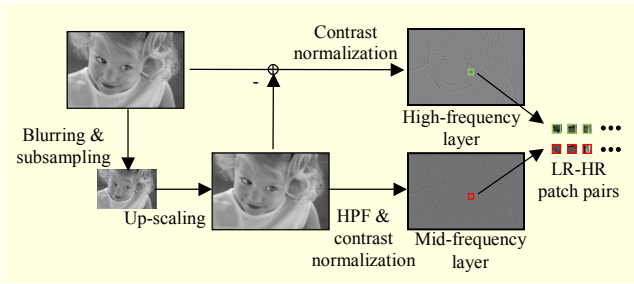

Fig. 3. Preprocessing for dictionary construction.

steps. In addition, the residue dictionary is designed to compensate for some high-frequency (HF) information lost during dictionary size reduction. The synthesis phase is composed of preprocessing, patch extraction, HF synthesis using an ordinary dictionary, and residual HF synthesis using a residue dictionary. The details of the proposed algorithm are as follows.

1. Learning Phase 1: Preprocessing and Patch Extraction

Prior to the learning process, training images should be appropriately preprocessed to achieve effective dictionary construction (see Fig. 3). Each HR image I_H is blurred and subsampled to generate an LR image I_L. Then, I_L is again up-scaled using simple linear interpolation such as bilinear interpolation or cubic convolution to produce an image I_{UP} having the same resolution as I_H.

The dictionary should possess various HF details lost by image degradation process and specific features to index them. The HF image I_{HF} is obtained by subtracting I_{UP} from I_H, and mid-frequency (MF) image I_{MF} stands for a high-pass filtered version of I_{UP}. I_{MF} is employed as the features for indexing. Note that the HF layer I_{HF} is the target information to be recovered by the proposed algorithm. They indicate lost HF and MF layers for predicting them, respectively. As a result, we

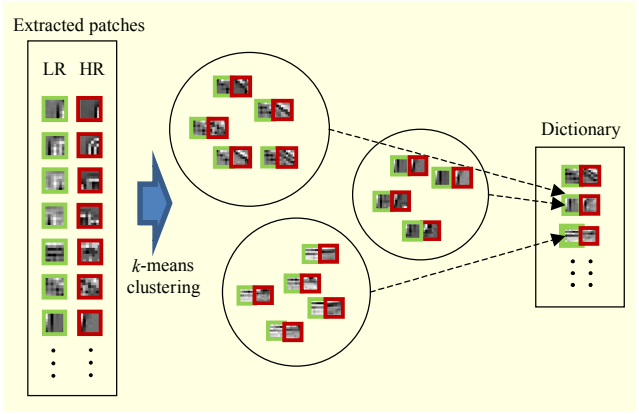

Fig. 4. Dictionary reduction using k-means clustering.

extract and store salient HR and LR patches from I_{HF} and I_{MF}, respectively. Those patches are properly overlapped with neighboring patches for local smoothness. Without loss of generality, we assume that the relationship between I_{HF} and I_{MF} is independent of the local image contrast. So, we normalize the contrasts of LR and HR patches by dividing them by the energy of the LR patch. Here, the energy stands for the L_1-norm. Finally, so-called primitive patches including edges or textures are chosen and they are exclusive to the dictionary. In other words, the proposed synthesis may be applied only for the primitive regions. A maximum response filter [14] is used to extract primitives as in [13].

2. Learning Phase 2: Dictionary Size Reduction

Now, we need to effectively reduce the number of LR-HR patch pairs in the dictionary so as to mitigate memory cost and computational burden in synthesis. This process is very significant in that the number of training examples in the dictionary generally dominates the performance of learning-based SR. Most of all, the small number of the samples can improve the practicality of the proposed SR algorithm.

So, we group adjacent LR-HR patch pairs into a single patch pair. We adopt k-means clustering to gather similar patches. Figure 4 illustrates this clustering process. LR patches which are close to each other in terms of L_2-norm distances are clustered into a single group, and simultaneously, the corresponding HR patches are clustered in the same group. Note that LR and HR patches in an LR-HR patch pair are always assigned into the same cluster. Finally, the center points of each cluster become new LR and HR patches belonging to the ordinary dictionary. In practice, we can determine k by considering memory cost and computational complexity of the synthesis phase.

3. Learning Phase 3: Residue Dictionary Construction

The above-mentioned dictionary size reduction sometimes

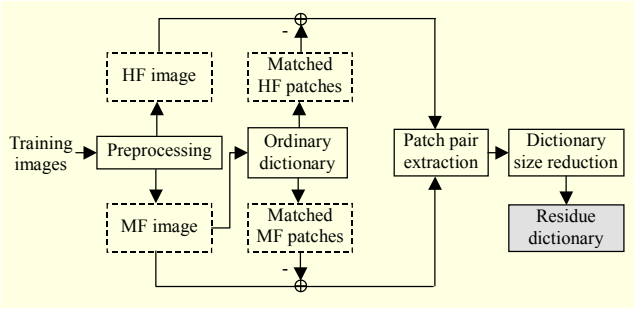

Fig. 5. Residue dictionary construction.

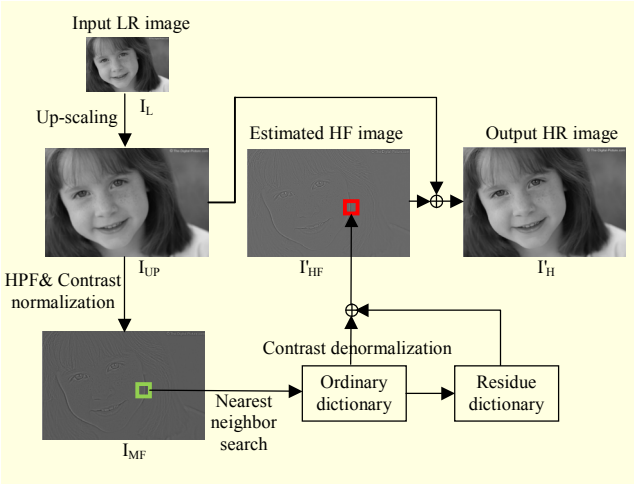

Fig. 6. Synthesis phase.

causes a blurred artifact because it can weaken HF components by averaging similar HR patches. If the residue patch, that is, the difference between the actual HR and estimated HR patches is learned well, the learned dictionary of the residue patches may further improve visual quality [13], [15]. So, we employ a residue dictionary to enrich weak HF information. Figure 5 describes the training procedure to construct the residue dictionary. Note that the training images for the residue dictionary are different from those for the ordinary dictionary. Firstly, the HF and MF images are generated from preprocessing. Next, the best matched MF patch to each training MF patch is searched with its corresponding HF patch. Then, the HF residue patches between the original HF patches and the estimated HF patches are produced. Similarly, the MF residue patches are computed. Finally, the MF and HF residue patches are clustered in the same fashion as in III.2, and the residue dictionary is finally obtained. Note that the cluster centers of the MF residue patches are employed for indexing.

4. Synthesis Phase

Figure 6 describes the synthesis phase. The input LR image is initially up-scaled using a linear scaler, and then LR patches are extracted from the MF layer of the input image as in the learning phase. Each input LR patch is compared with the candidate LR patches in the dictionary to find the best match in L_2-norm distance. Next, the HR patch corresponding to the best matched LR patch is denormalized by multiplying with the energy of the input LR patch. Subsequently, a proper residue HF patch for each LR patch is explored from the residue dictionary, and the final HF patch is obtained by adding the residue patch to the best-matched HF patch selected from the ordinary dictionary. Note that the input MF residue patch, that is, the difference between the input LR patch and the best-matched LR patch, is compared with candidate MF residues in the residue dictionary. This process is applied to all the input patches. Averaging is only performed for pixels in overlapped regions. Finally, we obtain a synthesized HR image by adding the HF image I'_{HF} to the initially up-scaled image I_{UP}.

Note that the proposed algorithm selects the single best-matched patch unlike the conventional learning-based SR algorithms using multiple nearest patches. The single best-matched patch of the proposed algorithm may correspond to the average of multiple nearest neighbor patches of [13] because adjacent LR/HR patches on Euclidean space are clustered in the training phase of the proposed algorithm. Therefore, even though we use a single best-matched patch for HF synthesis, we can obtain a similar effect to synthesis using multiple nearest neighbor patches.

IV. Experimental Results

In order to fairly evaluate the performance of our algorithm, we compared it with the bicubic algorithm and Fan's algorithm [13].

For training, we used 16 500×333 digital camera images downloaded from http://www.the-digital-picture.com/ (see Fig. 7). This paper considered a scaling ratio of 1/2. So, the LR images were produced from the corresponding HR images by using anti-aliasing filtering based on a 7×7 Gaussian filter with a standard deviation of 0.85 and down-sampling. The patch size was set to 7×7 and the patches were overlapped every 4 pixels.

The initial ordinary dictionary was constructed with 100,000 primitive patch pairs extracted from the upper eight training images of Fig. 7. Then, we reduced the size of the initial dictionary by 1/10 and 1/20, respectively, as proposed in III.2. We used bilinear interpolation and Laplacian filter for initial up-scaling and high pass filtering, respectively. However, the residue dictionary was learned using lower eight images in Fig. 7. The residue dictionary size was set to a half of the ordinary dictionary size so as to mitigate the memory cost. Also, we restricted the maximum number of iterations of *k*-means

Fig. 7. 16 training images (upper 16 images) and two test images (lower).

Table 1. PSNR comparison (dB). The numbers in parenthesis indicates the reduction ratios of the dictionary size.

Test images	Bicubic	Fan's (1/20)	Fan's (1/10)	Proposed (1/20)	Proposed (1/10)
Lena	33.15	34.35	34.46	34.89	34.86
Barbara	24.89	25.18	25.22	25.33	25.32
Baboon	24.25	24.64	24.69	24.91	24.90
Pirate	30.34	31.20	31.32	31.68	31.72
Flower	32.43	33.67	33.73	34.37	34.50
Bee	28.23	29.16	29.36	29.68	29.70

clustering to 10 as a termination condition.

For evaluation, we tested four well-known 512×512 still images: Lena, Pirate, Barbara, and Baboon as well as two 500×333 images: Flower and Bee (see Fig. 7).

Actually, in order to store 100,000 examples in the dictionary, a large memory size of about 15 megabytes (MB) is required. Note that the proposed algorithm reduces such a huge dictionary size up to 1.5 MB (1/10) or 0.75 MB (1/20).

On the other hand, we constructed the dictionary for Fan's algorithm by regularly sampling LR-HR patch pairs in the initial dictionary so that its dictionary size is equivalent to the total dictionary size of the proposed algorithm.

Table 1 shows PSNR comparison results for various test images. For example, the proposed algorithm provides 2 dB higher PSNR than the bicubic for the Flower image when using only 10,000 examples. For the same case, the proposed algorithm shows about 0.8 dB higher PSNR than Fan's algorithm. Note that even when the dictionary size becomes much smaller, for example, reduction ratio of 1/20, the proposed algorithm still maintains higher PSNRs than the bicubic algorithm or Fan's algorithm.

Figure 8 shows the interpolated images of Lena for the

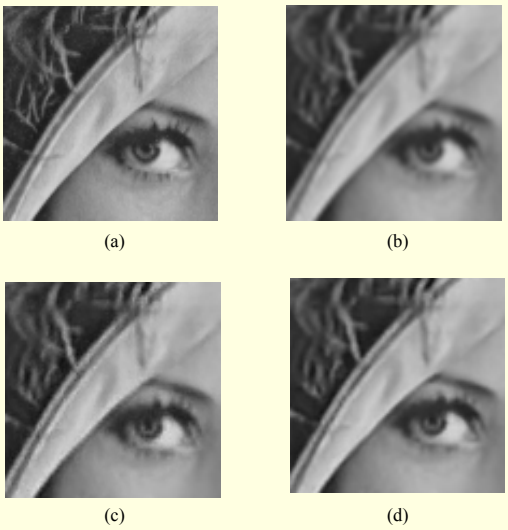

Fig. 8. Part of up-scaled images of Lena using 10,000 training examples: (a) original, (b) bicubic, (c) Fan's, and (d) the proposed algorithm.

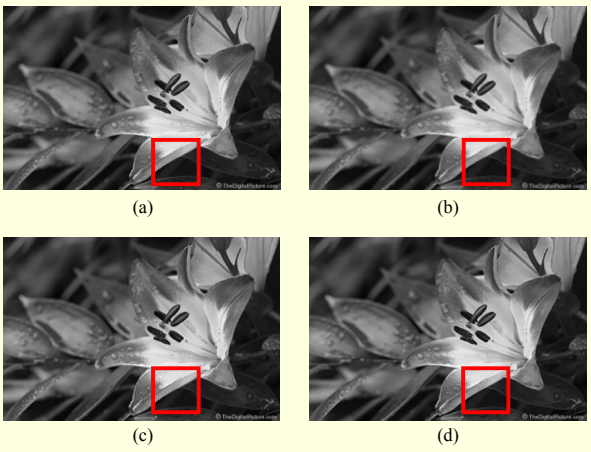

Fig. 9. Up-scaled images of Flower using 10,000 training examples: (a) original, (b) bicubic, (c) Fan's, and (d) the proposed algorithm.

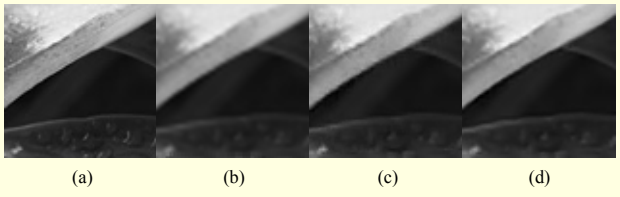

Fig. 10. Magnified images of the red box in Fig. 9: (a) original, (b) bicubic, (c) Fan's, and (d) the proposed algorithm.

reduction ratio of 1/10. We can observe that the proposed algorithm provides better visual quality than the bicubic algorithm and Fan's algorithm. Note that Fan's algorithm has some annoying artifacts in the diagonal edge of the hat due to

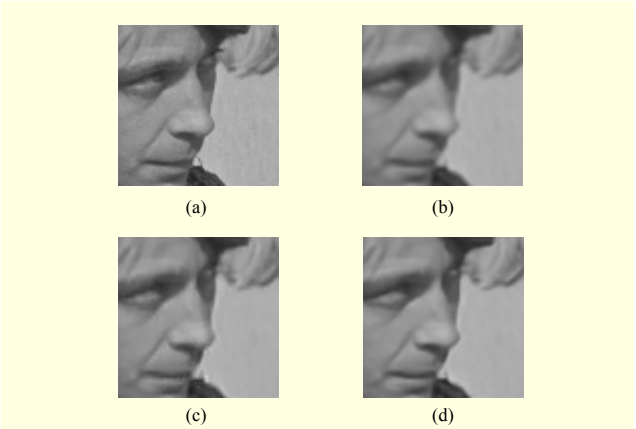

Fig. 11. Part of up-scaled images of Pirate using 5,000 training examples: (a) original, (b) bicubic, (c) Fan's, and (d) the proposed algorithm.

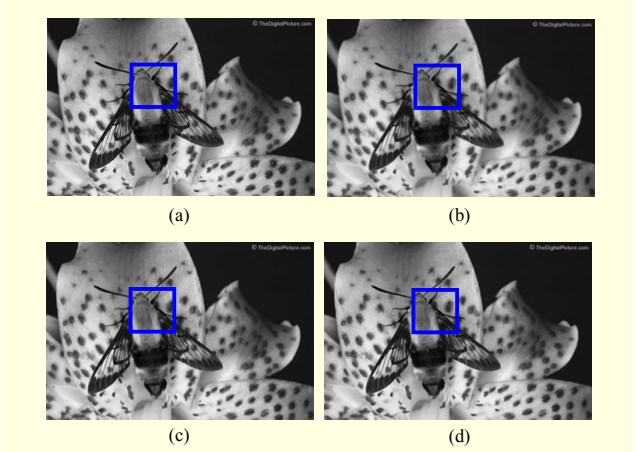

Fig. 12. Up-scaled images of Bee using 5,000 training examples: (a) original, (b) bicubic, (c) Fan's, and (d) the proposed algorithm.

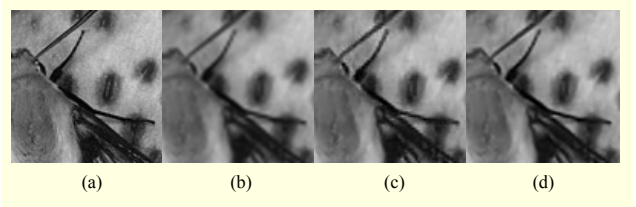

Fig. 13. Magnified images of the blue box in Fig. 12: (a) original, (b) bicubic, (c) Fan's, and (d) the proposed algorithm.

mismatching of LR patches as the dictionary size decreases. We can also see such artifacts in Figs. 9 and 10. In particular, Fig. 10 shows that the proposed algorithm provides clear details close to the original images in comparison with the other algorithms. Figures 11 to 13 prove that even when the number of patches is only 5,000, the proposed algorithm outperforms the other schemes. In a few parts of Figs. 12 and 13, for example, the legs of Bee, we can see that Fan's algorithm generates severe high-frequency artifacts.

Table 2. Running time comparison (s). The number in parenthesis indicates the number of patches in dictionary.

Test images	Bicubic	Fan's (100,000)	Proposed (10,000)	Proposed (5,000)
Lena	1.23	302.46	57.57	33.52
Barbara	1.23	311.98	56.69	34.44
Baboon	1.26	343.43	58.95	40.08
Pirate	1.23	332.96	65.24	35.40
Flower	1.03	207.25	38.09	23.51
Bee	1.00	194.67	36.38	20.50

The experiments were executed on an Intel Core™ 2 Duo CPU @ 2.5 GHz with 3 GB RAM. Table 2 compares the running time of the proposed algorithm with that of Fan's algorithm with 100,000 examples. We can see that the proposed algorithm is 6 to 10 times faster than Fan's algorithm, while providing reasonable performance without visible artifacts.

V. Conclusion

This paper proposed a fast image super-resolution algorithm which reduced the size of the trained dictionary with minimal performance degradation. At the learning phase, the number of sample patch pairs in the dictionary is noticeably reduced by grouping similar patches using *k*-means clustering. At the synthesis phase, one best-matched patch for each input low-resolution patch is selected from the dictionary. Finally, the corresponding high-frequency patch is used to reconstruct the output high-resolution patch. Thus, the proposed algorithm realizes fast image super-resolution with the reduced dictionary size, while providing reasonable visual quality.

References

[1] R.G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 29, no. 6, Dec. 1981, pp. 1153-1160.

[2] H.S. Hou and H.C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 26, no. 6, Dec. 1978, pp. 508-517.

[3] J. Allebach and P.W. Wong, "Edge-Directed Interpolation," *Proc. Int. Conf. Image Process.*, vol. 3, Sept. 1996, pp. 707-710.

[4] X. Li and M. Orchard, "New Edge-Directed Interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, Oct. 2001, pp. 1521-1527.

[5] Q. Wang and R. Kreidieh, "A New Orientation-Adaptive Interpolation Method," *IEEE Trans. Image Process.*, vol. 16, no. 4, Apr. 2007, pp. 889-900.

[6] S.M. Kwak, J.H. Moon, and J.K. Han, "Modified Cubic Convolution Scaler for Edge-Directed Nonuniform Data," *Optical Eng.*, vol. 46, no. 10, 107001, Oct. 2007, doi:10.1117/12.782389.

[7] M. Irani and S. Peleg, "Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *J. Visual Commun. Image Representation*, vol. 4, no. 4, 1993, pp. 324-335.

[8] Z. Lin and H.Y. Shum, "Fundamental Limits of Reconstruction-Based Superresolution Algorithms Under Local Translation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 1, Jan. 2004, pp. 83-97.

[9] S.C. Park, M.K. Park, and M.G. Kang, "Super-Resolution Image Reconstruction: A Technical Overview," *IEEE Signal Process. Magazine*, vol. 20, no. 3, 2003, pp. 21-36.

[10] W.T. Freeman, T.R. Jones, and E.C. Pasztor, "Example-Based Super-Resolution," *IEEE Computer Graphics Appl.*, vol. 22, no. 2, Oct. 2002, pp. 56-65.

[11] J. Sun et al., "Image Hallucination with Primal Sketch Priors," *Proc. IEEE Comput. Soc. Conf. Comp. Vision Pattern Recog.*, vol. 2, 2003, pp. 729-736.

[12] H. Chang and D. Yeung, and Y. Xiong, "Super-Resolution through Neighbor Embedding," *Proc. IEEE Comput. Soc. Conf. Comp. Vision Pattern Recog.*, vol. 1, 2004, pp. 275-282.

[13] W. Fan and D. Yeung, "Image Hallucination Using Neighbor Embedding over Visual Primitive Manifolds," *Proc. IEEE Comput. Soc. Conf. Comp. Vision Pattern Recog.*, 2007, pp. 1-7.

[14] M. Varma and A. Zisserman, "A Statiscal Approach to Texture Classification from Single Images," *Int. J. Computer Vision*, vol. 62, no. 1-2, 2005, pp. 61-81.

[15] C. Kim, K. Choi, and J. Ra, "Improvement on Learning-Based Super-Resolution by Adopting Residual Information and Patch Reliability," *IEEE Int. Conf. Image Process.,* 2009, pp. 1197-1200.

Shin-Cheol Jeong received his BS in electronic engineering from Inha University, Incheon, Korea, in 2009. Currently, he is pursuing an MS in electronic engineering from Inha University. His research interests include image processing, super-resolution, and video coding.

Byung Cheol Song received the BS, MS, and PhD in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a senior engineer at Digital Media R&D Center, Samsung Electronics Co., Ltd., Suwon, Korea. In March 2008, he joined the School of Electronic Engineering, Inha University, Incheon, Korea, and is currently an assistant professor. His research interests are video coding, video processing, super-resolution, stereo vision, multimedia system design, image coding, content-based multimedia retrieval, and data mining.