

Research Article

An Efficient Image Enlargement Method for Image Sensors of Mobile in Embedded Systems

Hua Hua,¹ Xiaomin Yang,¹ Binyu Yan,¹ Kai Zhou,² and Wei Lu³

¹College of Electronics and Information Engineering, Sichuan University, Chengdu, Sichuan 610064, China

²College of Electrical and Engineering Information, Sichuan University, Chengdu, Sichuan 610064, China

³School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Xiaomin Yang; yangxiaomin_2001@163.com

Received 20 November 2015; Revised 5 March 2016; Accepted 21 March 2016

Academic Editor: Marco Anisetti

Copyright © 2016 Hua Hua et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Main challenges for image enlargement methods in embedded systems come from the requirements of good performance, low computational cost, and low memory usage. This paper proposes an efficient image enlargement method which can meet these requirements in embedded system. Firstly, to improve the performance of enlargement methods, this method extracts different kind of features for different morphologies with different approaches. Then, various dictionaries based on different kind of features are learned, which represent the image in a more efficient manner. Secondly, to accelerate the enlargement speed and reduce the memory usage, this method divides the atoms of each dictionary into several clusters. For each cluster, separate projection matrix is calculated. This method reformulates the problem as a least squares regression. The high-resolution (HR) images can be reconstructed based on a few projection matrixes. Numerous experiment results show that this method has advantages such as being efficient and real-time and having less memory cost. These advantages make this method easy to implement in mobile embedded system.

1. Introduction

Over the last few decades, people have widely adopted mobile phones to life. For 2017, the number of mobile phone users will reach almost 5.3 billion. For many mobile phone users, mobile phone is used not only for spoken communication but also as a tool to capture images. Mobile phones offer great benefits to the users by enabling photography and video recording always and everywhere. Unfortunately, many of the images being taken with mobile phone are low in resolution since the low quality image sensor. There are two ways to obtain high-resolution images: (1) replace the mobile phone with a more powerful mobile phone; (2) use some methods to enlarge the images. Most of the mobile phone users prefer to use a method to enlarge the image rather than replacing the mobile phone with a more powerful mobile phone. Many efforts have been devoted to image enlargement methods in the past decade. However, the enlargement methods face three challenges when applied in embedded systems: (1) performance requirement, (2) real-time requirement, and (3) constraint on memory consumption.

Superresolution (SR) is one of the most prospective image enlargement methods. Existing SR methods can be divided into three categories: interpolation-based methods [1, 2], reconstruction-based methods [3–5], and example-based methods [6–10].

The interpolation-based methods [1, 2] apply the correlation of neighboring image pixels to approximate the fundamental HR pixels. These types of methods have lower computation complexities. However, the interpolation does not add any new detailed information into the enlarged image. The quality of the enlarged image is still unsatisfying and it may cause the aliasing to the enlarged LR image. Although the interpolation-based methods run fast and need little memory, the poor performance limits the application of interpolation-based methods for the implementation in embedded system.

Reconstruction-based methods [3–7] require different LR images of the same scene taken from slightly moved viewpoints, and those LR images have different subpixel shifts from each other. This category of methods tries to exploit additional information from a sequence of successive LR images of the same scene to synthesize HR images. Compared

with interpolation-based methods, the reconstruction-based methods obtain better performance with a small desired magnification factor. However, the performance of this kind of methods degrades rapidly when the desired magnification factor becomes large. The reconstruction-based methods need to store the information of all the sequence LR images, which is high memory requirement. Due to the above reasons, reconstruction-based methods are not ready for embedded system.

Single image SR methods such as neighbor embedding-based methods [7, 8], regression-based methods [9, 10], and sparse representation-based methods [11–15] have been explored in recent years. These methods presume that the high-frequency details lost in the LR images can be predicted through learning the cooccurrence relationship between LR training patches and their corresponding HR patches. Recently, sparse representation-based methods have proven to be effective towards solving image superresolution problems. Yang et al. [16] proposed an approach based on sparse representation, with the assumption that the HR and LR images share the same set of sparse coefficients. Therefore, the HR image can be reconstructed by combining the trained HR dictionary and the sparse coefficients of the corresponding LR image. Although the sparse representation-based methods offer a good performance, the optimization of dictionary learning and image reconstruction has a problem of highly intensive computation. Besides, sparse representation-based SR methods reserve memory to store the information of HR dictionary and LR dictionary. The size of dictionary impacts the memory usage. Sparse representation-based SR methods require intensive large memory, especially with increasing size of dictionary. Both the time-complexity and memory usage are key limit factors in the embedded system applications of these methods. Zhan et al. [17] proposed a fast multiclass dictionaries learning method in MRI reconstruction. Timofte et al. [18] constructed a set of mapping relationships between the LR and HR patches using a learned LR-HR dictionary. Anchored Neighborhood Regression method [18] reformulates the problem as a least squares regression, which leads to a vast computational speedup while keeping the same accuracy as previous methods. Anchored Neighborhood Regression method calculates the mapping matrix based on a universal dictionary. However, a large number of different structural patterns exist in an image, whereas one dictionary is not capable of capturing all of the different morphologies. Besides, Anchored Neighborhood Regression method still needs to store separate projection matrix for each dictionary atom which is high memory usage.

The existing sparse representation-based SR methods always suffer from three main problems for embedded system. First, the performance of these methods is limited, since these methods only use one approach to extract the features of the image for presenting the LR image generally. However, the morphologies vary significantly across images. Different patches prefer different features for accurately representing different morphologies. A single feature extraction approach cannot represent the image accurately. Therefore, jointly representing an image with different kind of features is important. Furthermore, time-complexity and memory usage are

key limit factors in the embedded system applications of these methods. The optimization of dictionary learning and image reconstruction leads to highly intensive computation. Sparse representation-based methods need to reserve memory to store the information of HR dictionary and LR dictionary. The size of dictionary impacts the memory usage.

Above all, this study makes the following three main contributions. (1) Jointly representing an image with different types of features is proposed in feature extraction stage. For accurately representing different morphologies, images (or patches) prefer different types of features extracted by different approaches, since one single feature extraction approach cannot accurately capture the essential features of the image. (2) Multiple dictionaries are learned based on different types of features in sparse representation stage, since one dictionary with single type of features is inadequate in capturing all of the different morphologies of the image. To capture the different morphologies of the image more accurately, multifeature dictionaries, which consist of different dictionaries with different features, are learned. (3) To reduce the computational cost and memory usage, we propose an Anchored Cluster Regression method. Anchored Cluster Regression method divides the dictionary atoms into several clusters. Then, the projection matrix for each cluster is calculated. In Anchored Cluster Regression method, each HR patch can be reconstructed by the projection matrix of its corresponding cluster. Anchored Cluster Regression method reformulates the problem as a least squares regression. It only needs to store the projection matrix of each cluster. Anchored Cluster Regression leads to a vast computational speedup and needs less memory.

2. Sparse Representation-Based SR Method

Superresolution aims to reconstruct the HR image from the LR image, which can be formulated as follows:

$$Y = HBX, \quad (1)$$

where $Y \in R^N$ is the observed low-resolution (LR) image. $X \in R^{NL}$ is its corresponding high-resolution (HR) image of the same scene. Y is a downsampled and blurred version of X . H denotes a downsampling operator and B is the blur operator.

Let Patch_-y^i be LR patch of the LR image with the size $\sqrt{n} \times \sqrt{n}$ at the location i , $i = 1, 2, \dots, C$. Then, we have

$$\text{Patch}_-y^i = R_i(Y), \quad (2)$$

where $R_i(\cdot)$ is an operator that extracts a patch at position i from the LR image Y .

Similarly, the corresponding HR patch Patch_-x^i is with the size $\sqrt{nL} \times \sqrt{nL}$ at the location i , $i = 1, 2, \dots, C$. And we have

$$\text{Patch}_-x^i = R_i(X). \quad (3)$$

With LR patch Patch_-y^i , y^i is the feature extracted from Patch_-y^i . The feature can be expressed as

$$y^i = \text{Features}_L(\text{Patch}_-y^i), \quad (4)$$

where $\text{Features}_L(\cdot)$ refers to extracting LR feature operator.

Subsequently, the corresponding HR feature x^i is extracted from HR patch Patch_{x^i} :

$$x^i = \text{Features}_{H_H}(\text{Patch}_{x^i}), \quad (5)$$

where $\text{Features}_{H_H}(\cdot)$ refers to extracting HR feature operator, which is usually the differences between the LR image and its corresponding HR image.

With the sparse generating model, each LR patch feature y^i ($y^i \in \mathfrak{R}^n$) can be projected over the LR dictionary D_l , which characterizes the LR patches. This projection produces a sparse representation of y^i via α_l^i :

$$y^i = D_l \alpha_l^i \quad y^i \in \mathfrak{R}^n, \quad \|\alpha_l^i\|_0 \ll k, \quad (6)$$

where D_l and α_l^i are the LR dictionary and the sparse representation of y^i , respectively. Generally, in order to obtain an optimal α_l^i that has the fewest nonzero elements, we should solve the following optimization problem:

$$\alpha_l^i = \min_{\alpha} \|D_l \alpha - y^i\|_2 + \lambda \|\alpha\|_0, \quad (7)$$

where λ is a constant.

Similarly, we have the sparse representation of the HR patch:

$$x^i \approx D_h \alpha_h^i, \quad (8)$$

where D_h is the HR dictionary. Conventional sparse representation-based methods assume that the LR patch and its corresponding HR version share the same sparse coefficients in relation to their own dictionaries; namely, $\alpha_l^i = \alpha_h^i$. Therefore,

$$x^i \approx D_h \alpha_l^i. \quad (9)$$

HR dictionary is defined as

$$D_h = \arg \min_{D_h} \sum_i \|x_i - D_h \alpha_l^i\|_2. \quad (10)$$

The sizes of the dictionaries D_l and D_h are $D_l \in \mathfrak{R}^{n \times w}$ and $D_h \in \mathfrak{R}^{m \times w}$, respectively, where w is the number of atoms in the dictionary. n is the dimension of each atom in LR dictionary while m is the dimension of each atom in HR dictionary.

It is clear that the sparse representation is a bridge between low-resolution and high-resolution patches. To generate such sparse representation, both LR dictionary D_l and HR dictionary D_h play a key role. The dictionaries D_l and D_h can be easily generated from a set of samples by the methods such as OMP [13].

Once sparse coefficients for each LR patch are learned, we can use this sparse representation to recover its corresponding HR patch. If we have obtained all the reconstructed HR patches, the HR image is recovered by averaging the overlapping reconstructed patches on their overlaps.

3. The Proposed SR Method

The proposed method can be divided into three steps: (a) learning different dictionary based on different morphologies, (b) calculating the projection matrixes, and (c) reconstructing the HR mobile sensor image.

3.1. Learning the Dictionaries Based on Different Features. Most existing sparse representation-based SR methods use only derivative features to represent the morphologies of LR image. However, the artifacts would occur when using inappropriate features. An explanation for this phenomenon is that dictionary learning from only one kind of features cannot represent essential morphologies of the images. Since the morphologies can vary significantly across images, different patches prefer different features for representation of their morphology accurately. As such, multifeature treatment can help represent the image in a more efficient manner. We propose a method which can present the image with different dictionaries based on different features.

For LR patch Patch_{y^i} , K different types of features can be adopted to represent it:

$$y^{i,k} = \text{Features}_L^k(\text{Patch}_{y^i}), \quad k = 1, 2, \dots, K, \quad (11)$$

where $y^{i,k}$ ($k = 1, 2, \dots, K$) is the k th kind of features of Patch_{y^i} . $\text{Features}_L^k(\cdot)$ denotes extracting k th kind of features.

Similarly, for the HR patch Patch_{x^i} , x^i is the feature of it.

Given HR patch Patch_{x^i} and LR patch Patch_{y^i} , we can obtain K kinds of LR and HR patch pairs $\{y^{i,k}, x^i\}$ ($k = 1, 2, \dots, K$) for training.

Based on the K kinds of LR and HR training sets $\{y^{i,k}, x^i\}$ prepared above, the LR and HR dictionaries of these training sets are learned from the following models.

The K -SVD dictionary training is applied to the set of patches $\{y^{i,k}\}$:

$$\begin{aligned} D_l^k &= \arg \min_{D_l^k, \alpha_l^{i,k}} \|y^{i,k} - D_l^k \alpha_l^{i,k}\|_2^2 \\ \text{s.t.} \quad &\|\alpha_l^{i,k}\|_0 \leq L \quad \forall i, \end{aligned} \quad (12)$$

where $\alpha_l^{i,k}$ are sparse coefficient vectors of $y^{i,k}$ and $\|\cdot\|_0$ is the l^0 norm counting the nonzero entries of a vector. Most sparse representation-based SR methods rely on the assumption that the HR and LR images share the same set of sparse coefficients. Therefore, the HR image can be reconstructed by combining the HR dictionary and the sparse coefficients of the corresponding LR image. Thus, the HR patch x^i can be recovered by approximation as $x^i \approx D_h^k \alpha_l^{i,k}$. D_h^k can be calculated by minimizing the following mean approximation error; that is,

$$D_h^k = \arg \min_{D_h^k} \|x^i - D_h^k \alpha_l^{i,k}\|_2^2. \quad (13)$$

3.2. Calculating the Projection Matrixes. Although sparse representation-based methods offer a good performance, the

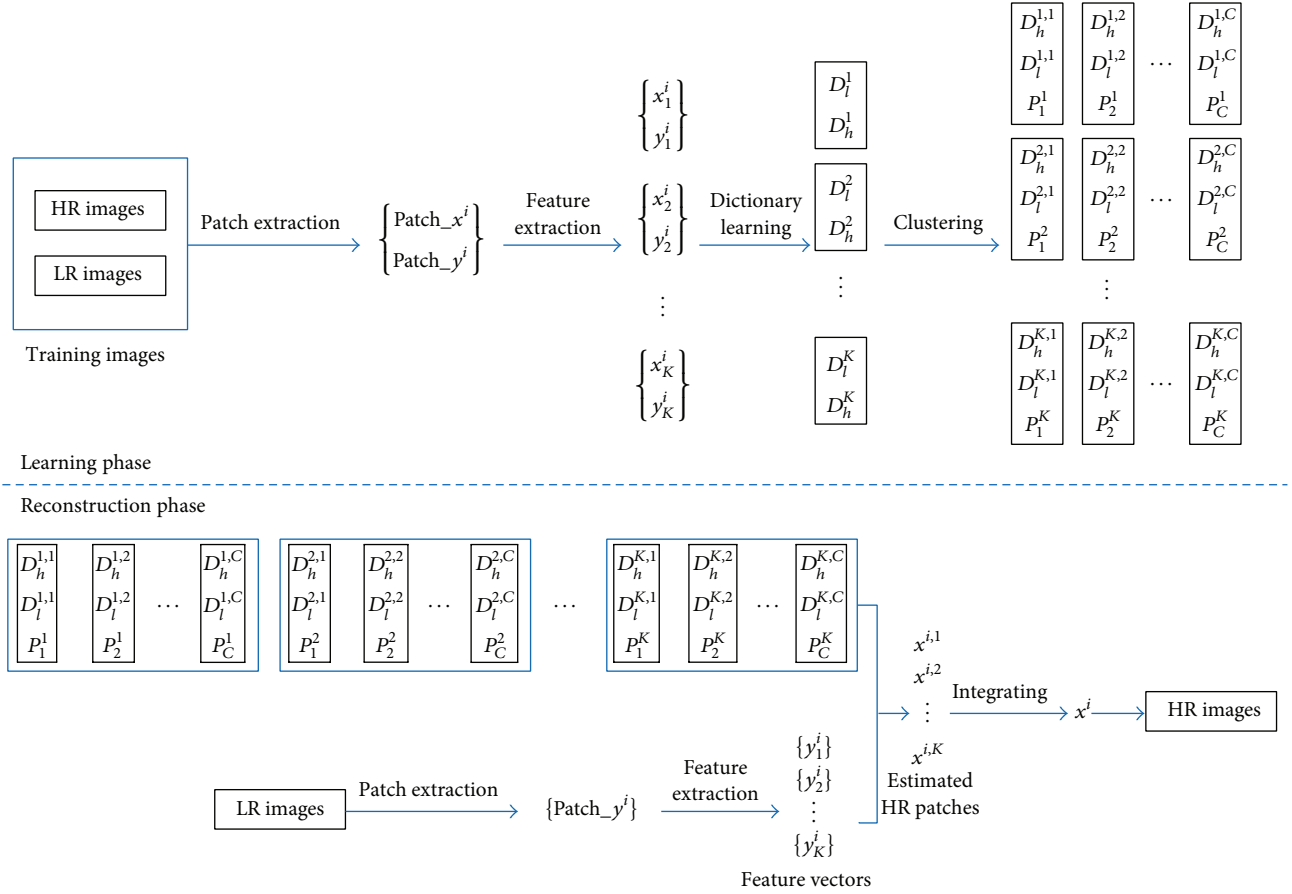


FIGURE 1: Framework of sparse representation-based SR method.



FIGURE 2: Test images in our experiments.

optimization of dictionary learning and image reconstruction has a problem of highly intensive computation. Besides, sparse representation-based SR methods need to reserve memory to store the information of HR dictionary and LR dictionary. The size of dictionary impacts the memory usage. Sparse representation-based SR methods require intensive large memory, especially with increasing size of dictionary. Both time-complexity and memory usage are key limit factors in the embedded system applications of these methods.

Timofte et al. [19] proposed an Anchored Neighborhood Regression method, which constructed a set of mapping matrixes between the LR and HR patches using learned LR and HR dictionaries.

Based on multiple dictionaries obtained in Section 3.2, Anchored Neighborhood Regression method solves this

problem as follows: for each dictionary, to calculate the sparse representation of y^i , problem (7) is reformulated as a least squares regression regularized by the l_2 -norm of the coefficients [19]:

$$\alpha_l^k = \min_{\alpha^k} \|D_l^k \alpha^k - y^i\|_2 + \lambda \|\alpha^k\|_2, \quad (14)$$

where D_l^k is the LR dictionary of the k th type of feature. D_h^k is the corresponding HR dictionary of D_l^k . α_l^k is the sparse vector of y^i .

Then, Ridge Regression is employed to solve the problem. The algebraic solution [19] is given as

$$\alpha_l^k = \left((D_l^k)^T D_l^k + \lambda I \right)^{-1} (D_l^k)^T y^i. \quad (15)$$

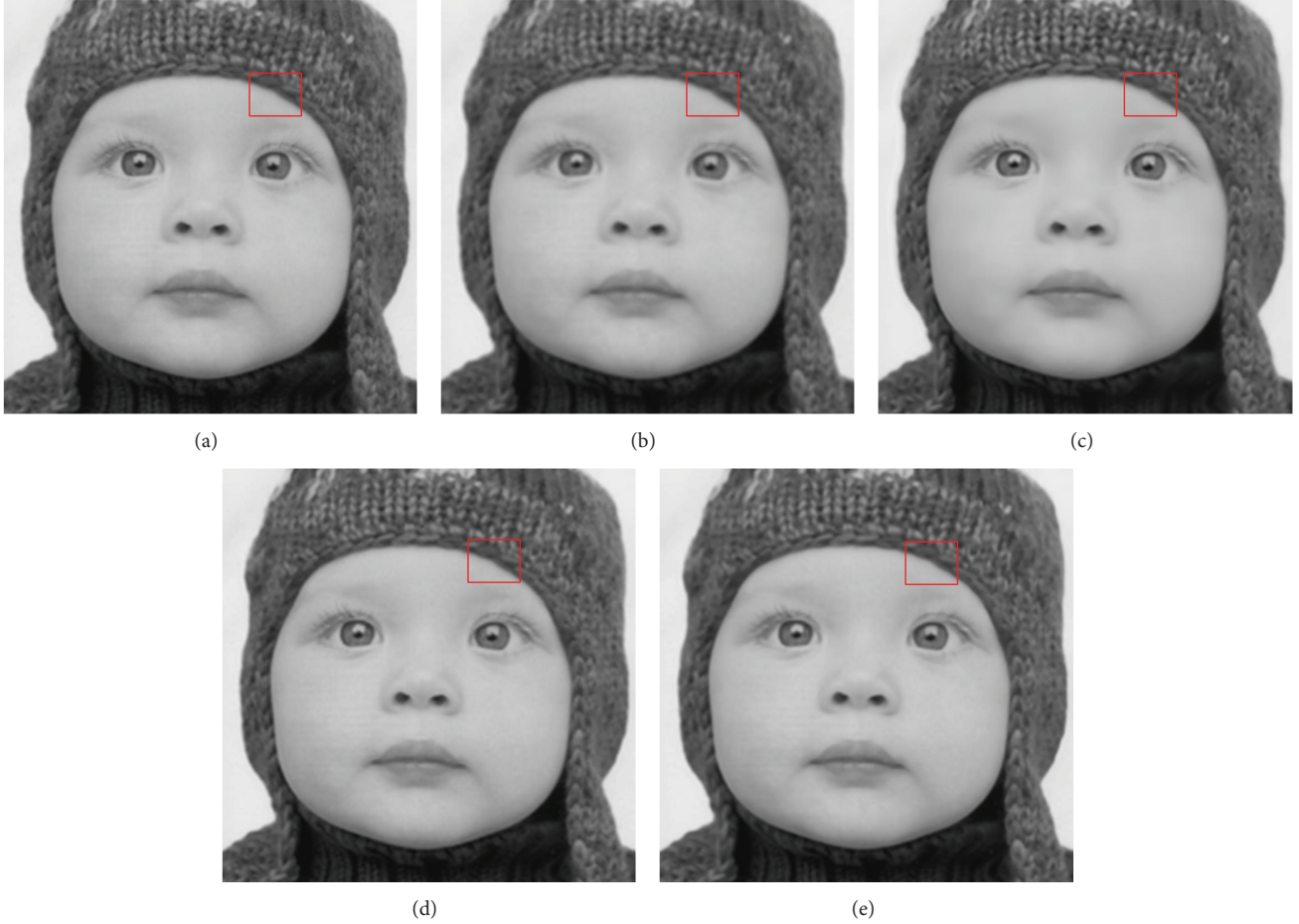


FIGURE 3: Visual comparison of baby: (a) original HR image, (b) results obtained using Cubic B-spline method, (c) results obtained using Yang's method, (d) results obtained using the ANR method, and (e) results obtained using our method.

Since sparse representation-based SR methods assume that the HR and LR images share the same set of sparse coefficients, therefore, the HR patches can be reconstructed by the sparse coefficients of the LR image α_l^k and the corresponding HR dictionary D_h^k :

$$x^i = D_h^k \alpha_l^k = D_h^k \left((D_l^k)^T D_l^k + \lambda I \right)^{-1} (D_l^k)^T y^i. \quad (16)$$

We can obtain mapping matrixes between the LR and HR patches:

$$\begin{aligned} x^i &= P_G^k y^i, \\ P_G^k &= D_h^k \left((D_l^k)^T D_l^k + \lambda I \right)^{-1} (D_l^k)^T. \end{aligned} \quad (17)$$

Equation (17) means that we can precalculate a mapping matrix for each dictionary. Inferring the HR patch becomes a multiplication for each input patch. The mapping matrix P_G^k can be computed offline and saved as a simple matrix to be applied to new image patches, which makes vast computational speedup while keeping the same accuracy as previous methods.

Timoft et al. [18] group the dictionary atoms into neighborhoods. More specifically, for each atom in the dictionary, they compute its K nearest neighbors, which will represent its neighborhood. Once the neighborhoods are defined, Anchored Neighborhood Regression method calculates a separate projection matrix P_i^k for each dictionary atom d_i^k based on its own neighborhood. The SR problem can then be solved by calculating the nearest atom in the dictionary for each input patch feature. Then, the HR patch can be reconstructed using the projection matrix of the nearest atom:

$$\begin{aligned} x^i &= P_i^k y^i, \\ P_i^k &= D_h^{k,i} \left((D_l^{k,i})^T D_l^{k,i} + \lambda I \right)^{-1} (D_l^{k,i})^T, \end{aligned} \quad (18)$$

where P_i^k is the projection matrix of the atom d_i^k and d_i^k is the nearest atom of y^i in the LR dictionary D_l^k . $D_l^{k,i}$ is the neighborhoods set of atoms d_i^k . $D_h^{k,i}$ is the corresponding set of the HR dictionary D_h^k .

Anchored Neighborhood Regression method reformulates the SR problem as a least squares regression, which

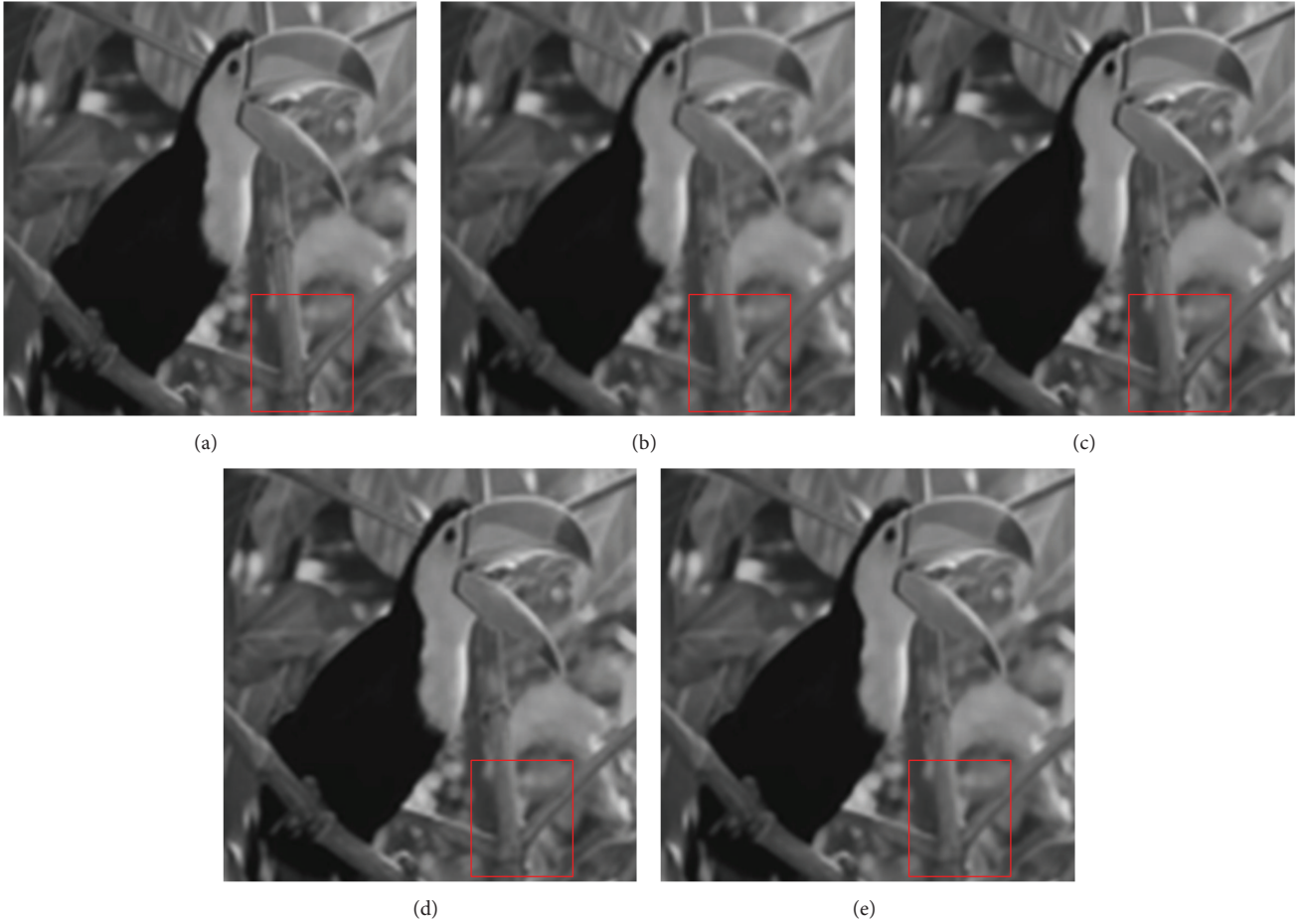


FIGURE 4: Visual comparison of bird: (a) original HR image, (b) results obtained using Cubic B-spline method, (c) results obtained using Yang's method, (d) results obtained using the ANR method, and (e) results obtained using our method.

leads to a vast computational speedup. However, Anchored Neighborhood Regression method still needs to store separate projection matrix for each dictionary atom which is high memory usage. Memory usage is key limit factor in the embedded system application for Anchored Neighborhood Regression methods.

To reduce the memory usage, we propose an Anchored Cluster Regression method. This method divides the atoms into several clusters for each dictionary by K -means clustering. Then, separate projection matrix P_c^k of each cluster is calculated. Then, use the projection matrix of the nearest cluster to reconstruct the HR patch:

$$x^{i,k} = P_c^k y^i, \quad (19)$$

$$P_c^k = D_h^{k,c} \left((D_l^{k,c})^T D_l^{k,c} + \lambda I \right)^{-1} (D_l^{k,c})^T, \quad (20)$$

$$\alpha_l^k = \left((D_l^{k,c})^T D_l^{k,c} + \lambda I \right)^{-1} (D_l^{k,c})^T y^i, \quad (21)$$

where $D_l^{k,c}$ is the set of atoms in the cluster c^k of the LR dictionary D_l^k . $D_h^{k,c}$ is the corresponding set of the HR dictionary D_h^k .

Anchored Cluster Regression method only needs to store the projection matrix of each cluster rather than the projection matrix of each atom. If N atoms are divided into K clusters, Anchored Cluster Regression method only needs to store K projection matrix of each cluster, while Anchored Neighborhood Regression needs to store N projection matrix of each atom. Anchored Cluster Regression significantly reduces the memory. Furthermore, the computational complexity of Anchored Cluster Regression method is $O(K)$, while complexity of Anchored Neighborhood Regression is $O(N)$, where N is the number of atoms and K is the number of clusters. Anchored Cluster Regression significantly reduces the computation.

3.3. Reconstructing the HR Image. Given a LR patch, we can get different HR patches based on different projection matrixes. These different HR patches are integrated to generate the final reconstructed HR image.

For a LR patch $\text{Patch}_l y^i$, we get the K kinds of features $\{y^{i,k}\}$ ($k = 1, 2, \dots, K$). For the features $y^{i,k}$, we can find the nearest cluster c^k of the k th LR dictionary; then, we can obtain k th estimated HR patch $x^{i,k}$ of the HR patch x^i based on the projection matrix by (19).

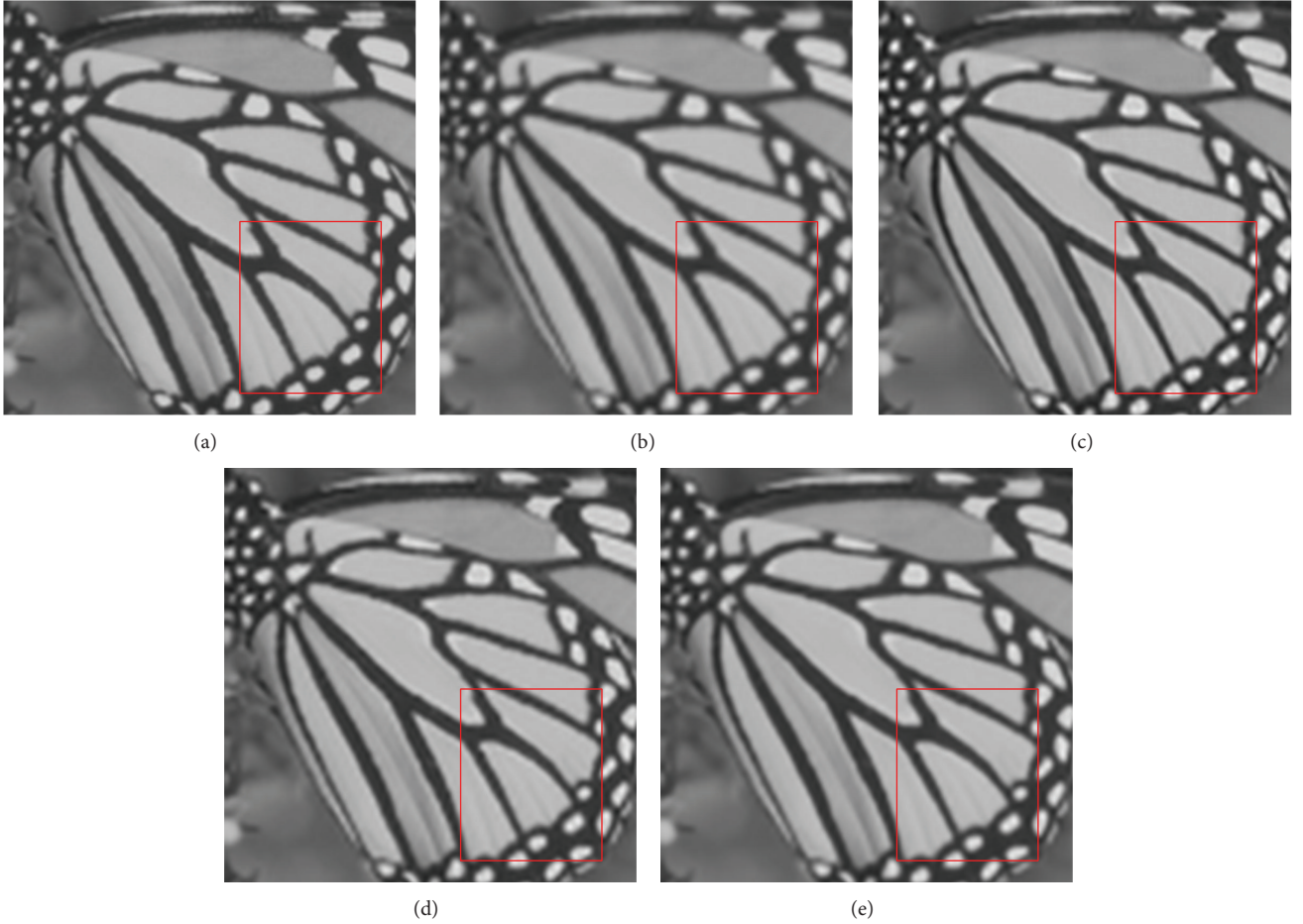


FIGURE 5: Visual comparison of butterfly: (a) original HR image, (b) results obtained using Cubic B-spline method, (c) results obtained using Yang's method, (d) results obtained using the ANR method, and (e) results obtained using our method.

Those K different estimated HR patches are fused together to get a final reconstructed HR image \hat{x}^i of HR patch x^i [20]:

$$\hat{x}^i = \frac{\sum_{k=1}^K w^k x^{i,k}}{\sum_{k=1}^K w^k}, \quad (22)$$

where w^k is important. According to the weight w^k , this study fuses the K different estimated HR patches $\{x^{i,k}\}$ ($k = 1, 2, \dots, K$) together to get the final reconstructed HR image \hat{x}^i :

$$w^k = \exp\left(-\frac{(e_l^k)^2}{2\sigma^2}\right), \quad (23)$$

where e_l is representation error function. e_l reflects the accurateness of the sparse representation:

$$e_l = y^{i,k} - D_l^{k,c} \alpha_l^{k,c}, \quad (24)$$

where e_l is smaller, $D_l^{k,c} \alpha_l^{k,c}$ is more similar to $y^{i,k}$.

3.4. Summary of Proposed Algorithm. The proposed method contains two phases, that is, learning phase and reconstruction phase. For the learning phase, features of different morphologies are extracted from training images. Then, valid multiple dictionaries are learned based on different morphologies. For each dictionary, the atoms are divided into multiple clusters. The projection matrix of each cluster is calculated by (19)–(21).

In the reconstruction phase, for each LR patch, features of different morphologies are first extracted. Then, for each type of features, the nearest clusters in its corresponding morphology dictionaries are found. Based on the projection matrixes of these clusters, multiple estimated HR patches are reconstructed in the final stage. Then, the final HR patches are generated by using weighting average to process all estimated HR patches. Ultimately, the HR image is composed through averaging the overlapping reconstructed patches. The algorithm is illustrated in Figure 1.

4. Experimental Results

In our experiments, we use the same training set as [18], which contains 91 images. For test, we use Set 5 datasets from

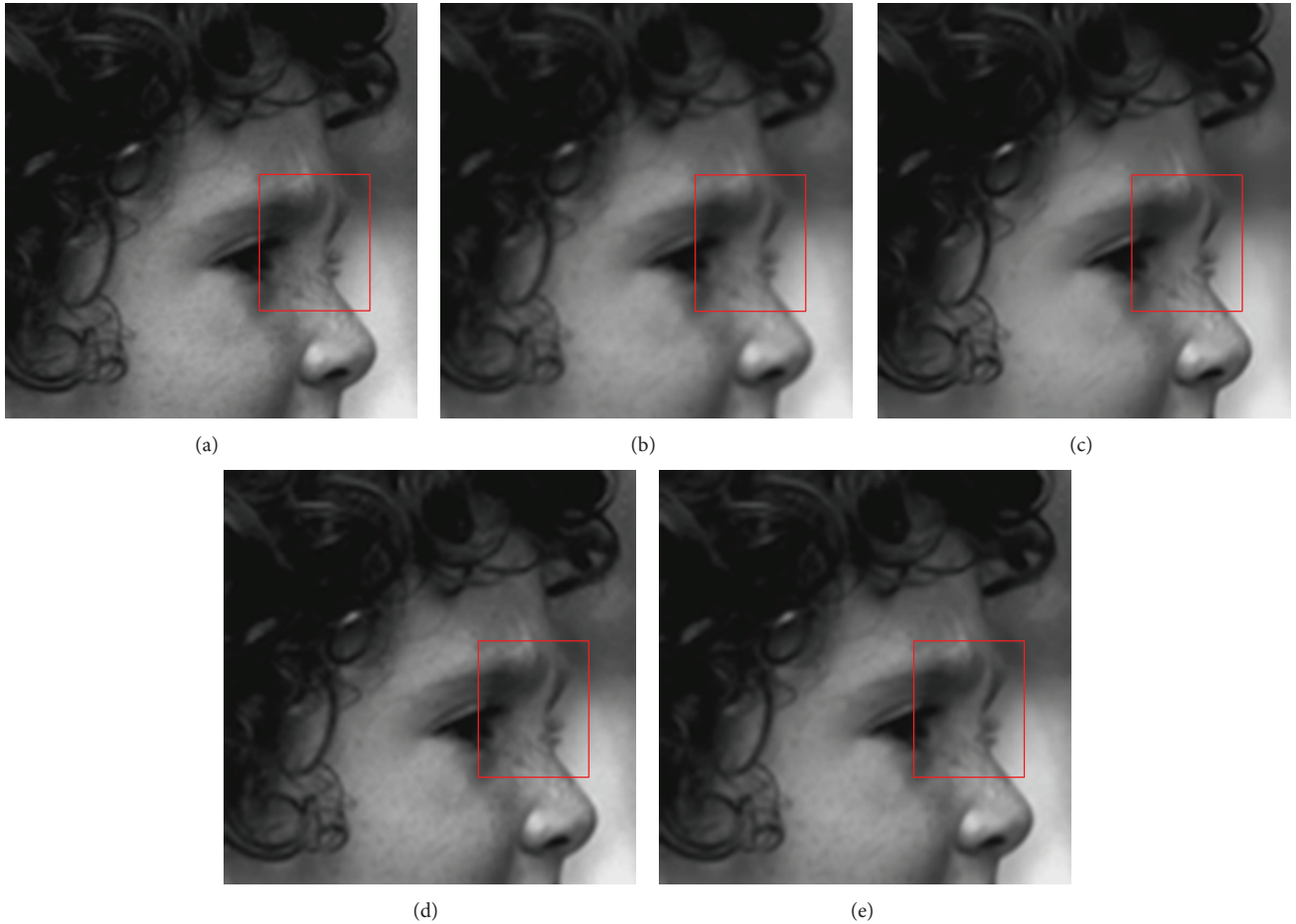


FIGURE 6: Visual comparison of head: (a) original HR image, (b) results obtained using Cubic B-spline method, (c) results obtained using Yang's method, (d) results obtained using the ANR method, and (e) results obtained using our method.

[18]. The Set 5 datasets contain 5 images. The test images are shown in Figure 2. The low-resolution images used in all experiments are downsampled from the high-resolution images. In our experiments, low-resolution images are generated by shrinking the corresponding high-resolution images with the scale factor of 3.

Gabor filters [21] have similar frequency and orientation representations to those of the human visual system. In the spatial domain, a 2D Gabor filter can be generated by Gaussian kernel function modulated with a sinusoidal plane wave. We employed Gabor features to represent the texture structure, since Gabor features can effectively characterize the texture representation and discrimination of LR patch. Gabor features of the image are extracted by convolving the normalized images with a family of Gabor filter (in this study, we use 2 scales and 4 orientations). Beside, we use derivative features representing high-frequency morphology structure of the image. $k = 1$ means that the features are Gabor features. $k = 2$ means that the features are derivative features.

In the low-resolution images, we always use 3×3 low-resolution patches, with overlap of 1 pixel between adjacent patches, corresponding to 9×9 patches with overlap of 3 pixels for the high-resolution patches. 10000 pairs of low- and

high-resolution patches are randomly chosen from the patch pairs generated by training images for dictionaries training.

Not only visual comparison but also quantitative comparisons are confirming the superiority of the proposed method. Peak signal-to-noise ratio (PSNR) and the structural similarity measurement (SSIM) have been implemented in order to obtain some quantitative results for comparison. The values of the PSNR and SSIM of all of the test images were used as the quality index. The PSNR evaluates the reconstruction quality based on the pixel intensity. The SSIM measures the similarity between two images based on their structural information. The SSIM metric needs a "perfect" reference image for comparison and provides a normalized value within $[0, 1]$, where "0" indicates that the two images are totally different, whereas "1" confirms that the two images are the same. Thus, higher values of PSNR and SSIM indicate a result with better quality.

4.1. The Effect of Dictionaries with Multifeatures. To validate the effect of dictionaries with multiple features, we compared our method with the derivative feature-based method and Gabor feature-based method. In our method, we use Gabor features to characterize the texture of the image and derivative

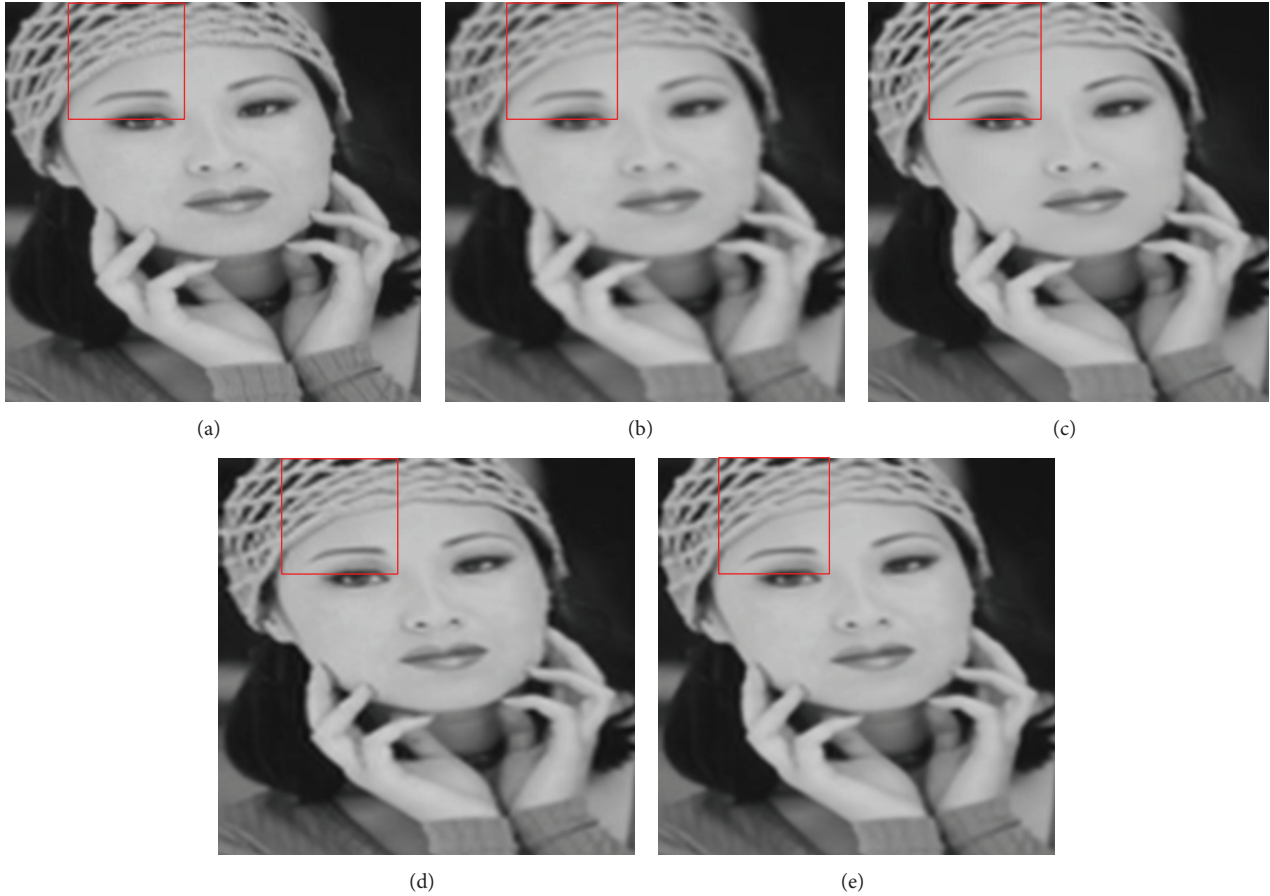


FIGURE 7: Visual comparison of woman: (a) original HR image, (b) results obtained using Cubic B-spline method, (c) results obtained using Yang's method, (d) results obtained using the ANR method, and (e) results obtained using our method.

features represent high-frequency morphology structure of the image. These two features are used together to present the image. The derivative feature-based method adopted derivative features to represent the LR image. And dictionaries with the derivative features were used to sparsely represent the LR patch. Similarly, Gabor feature-based method adopted Gabor features to represent the LR image. And dictionaries with the Gabor features were used to sparsely represent the LR patch. The PSNR and SSIM values of the SR results using various methods are listed in Table 1. We can observe that the proposed framework has a better performance than the single feature-based methods in terms of both PSNR and SSIM. This is due to the reason that one single feature extraction approach cannot accurately capture all the essential features of the image. Multitype features can jointly represent different morphologies in the image more accurately, so that different dictionaries with multitype features can represent the image in a more efficient manner and provide a more global look of the image, which would lead to the fact that the proposed framework has a better performance.

4.2. Reconstruction Results. To illustrate the effectiveness of the proposed framework in terms of visual fidelity and objective criterion, we compared the proposed framework with three well-known image SR algorithms, that is, Cubic

TABLE 1: Effects of multifeatures.

Method	Gabor features	Derivative features	Multiple features
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Baby	33.67/0.681	34.69/0.731	35.09/0.743
Bird	32.43/0.761	33.16/0.783	34.42/0.802
Butterfly	24.65/0.658	24.58/0.647	25.79/0.680
Head	29.46/0.673	33.19/0.712	33.56/0.728
Woman	24.94/0.578	29.86/0.683	30.16/0.693

B-spline interpolation method, Yang's method [16], and Anchored Neighborhood Regression method [18]. For ANR, in dictionary learning step, the number of atoms in the dictionary is 1,000. The neighborhood size is 40 when regressor is calculated.

Yang's method needs to store the information about the HR and LR dictionary. The sizes of HR patch and LR patch are $\sqrt{nL} \times \sqrt{nL}$ and $\sqrt{n} \times \sqrt{n}$, respectively. If the atom number in the dictionary is M , with t iterations needed, the computational complexity of Yang's method is roughly $O(kMnL)$. Anchored Neighborhood Regression needs to store M projection matrix of each atom in the dictionary. The size of projection matrix is $nL \times n$. The computational

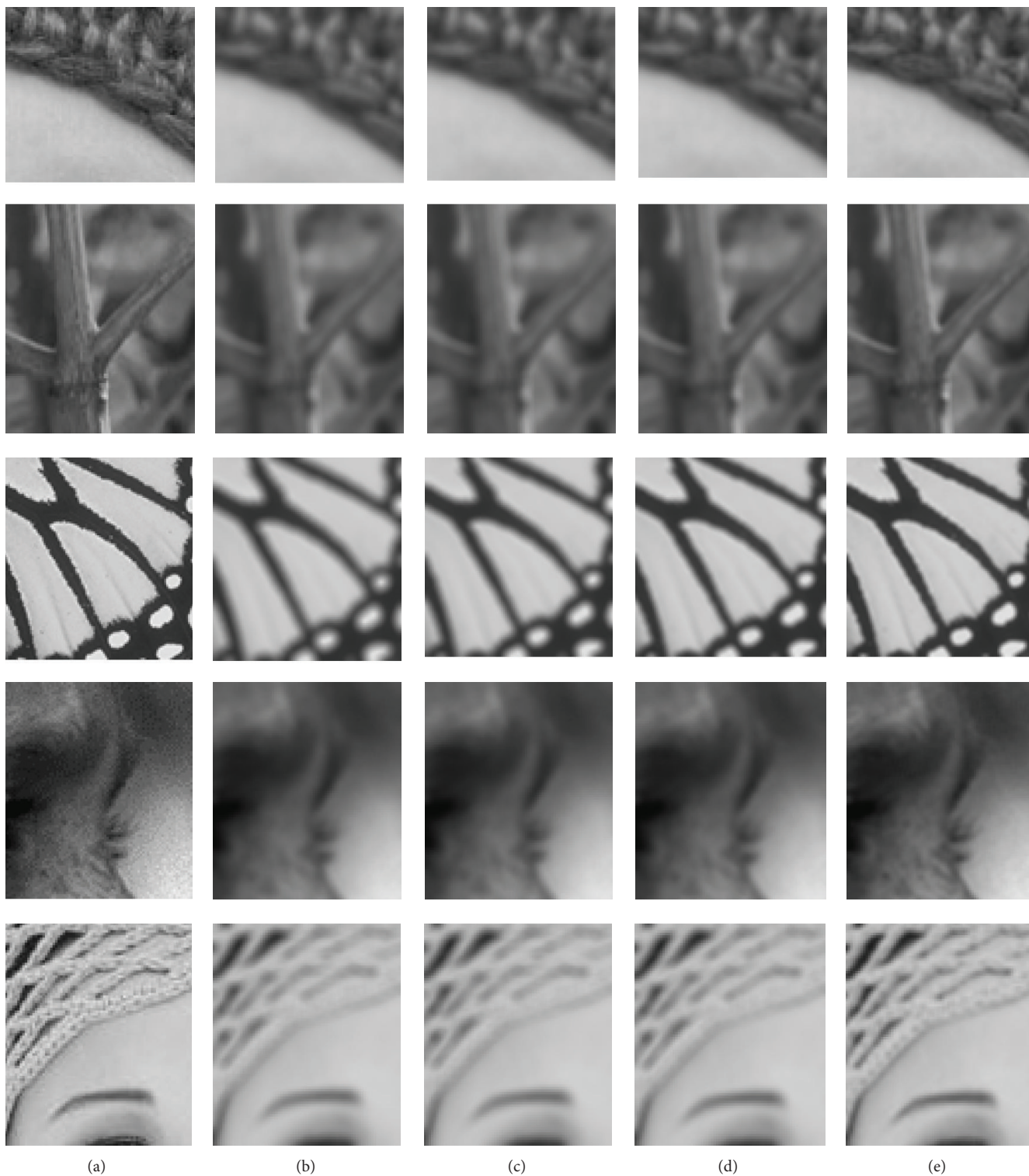


FIGURE 8: Zoomed version of the marked area of Figures 3–7: (a) original HR image, (b) results obtained using Cubic B-spline method, (c) results obtained using Yang's method, (d) results obtained using the ANR method, and (e) results obtained using our method.

complexity of Anchored Neighborhood Regression is $O(N)$; Anchored Cluster Regression method only needs to store the projection matrix of each cluster rather than the projection matrix of each atom. The size of projection matrix is $nL \times n$.

If N atoms are divided into K clusters, Anchored Cluster Regression method only needs to store K projection matrix of each cluster. The computational complexity of Anchored Cluster Regression method is $O(K)$.

TABLE 2: Numerical results of Figures 3–7.

	Cubic B-spline	Yang's	ANR	Our method
	PSNR/SSIM/time (second)	PSNR/SSIM/time (second)	PSNR/SSIM/time (second)	PSNR/SSIM/time (second)
Baby	33.91/0.697/0	34.29/0.726/56.06	34.90/0.738/0.98	35.09/0.743/0.8
Bird	32.58/0.768/0	34.11/0.789/22.33	34.38/0.794/0.29	34.42/0.802/0.24
Butterfly	24.04/0.632/0	25.58/0.674/20.22	25.82/0.687/0.24	25.79/0.680/0.20
Head	32.88/0.704/0	33.17/0.709/15.58	33.44/0.716/0.28	33.56/0.728/0.26
Woman	28.56/0.679/0	29.94/0.687/19.04	30.17/0.693/0.27	30.16/0.693/0.23

We first presented the superresolution results by different methods in Figures 3–7. Then, the zoomed version of the marked areas of Figures 3–7 is shown in Figure 8. We see that the Cubic B-spline interpolation methods blurred the edges and lost some dedicated details in the resultant images. Although Yang's method recovered plenty of details, it produced many jaggy and ringing artifacts along with the edges or details. The reason for this is that a single type of features is unable to completely represent various structures of the image. On the contrary, with the proposed framework, images were presented with sharp edges but fewer artifacts. Different dictionaries with multitype features were used to represent the LR image and the weights of different HR patches were also adaptively adjusted in the proposed framework. That is why the proposed framework obtained a better visual quality than all the other three methods.

Moreover, the PSNR and SSIM values of superresolution results using various algorithms are listed in Table 2. We can see the PSNR and SSIM gains of proposed framework over the other methods, which demonstrates that the superresolution results with the proposed framework have better objective quality than other methods in terms of PSNR and SSIM indexes. All this illustrates that the proposed framework is the best one among the compared methods in terms of visual perception and objective quality.

In addition, from Table 2, we see that the time consumption of Cubic B-spline interpolation methods is 0 seconds. That is because Cubic B-spline is an interpolation-based method which is the simplest and the fastest. We can find that Yang's method is consistently the most time-consuming, because it needs to solve a least squares optimization and an iterative convex optimization. Due to reformulating the SR problem as a least squares regression, our method runs a little faster than Anchored Neighborhood Regression method, that is, because it only needs to search the nearest cluster in our method rather than searching the nearest atom in Anchored Neighborhood Regression method. Besides, our method significantly reduces the memory usage, which is suitable for embedded system applications.

5. Conclusion

This paper introduces a new SR enlargement method for mobile sensor image. First, to represent mobile sensor images more accurately, complex information in natural images is optimally captured by different morphological components and multimorphology dictionaries learned from corresponding morphological training set. For each dictionary, the atoms

are divided into multiple clusters to calculate the projection matrix. Then, the weights of the HR patches obtained based on different projection matrix are adaptively controlled. Experiments have proved the improvement of the proposed framework in terms of both visual perception and quantitation comparisons with other compared methods. Since the main computation of this scheme is matrix multiplication and the memory usage is low, this method is easy to implement in mobile embedded system applications.

Competing Interests

The authors declare that they have no competing interests.

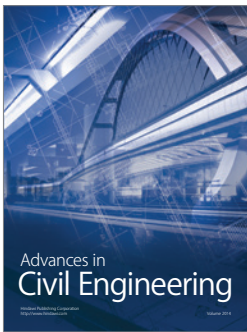
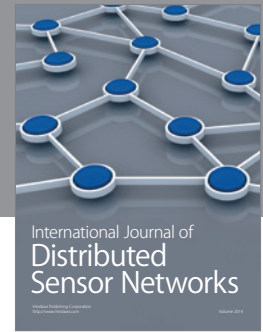
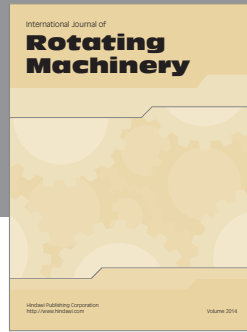
Acknowledgments

The research is sponsored by The National Natural Science Foundation of China (no. 61271330 and no. 6141101009), the Research Fund for the Doctoral Program of Higher Education (no. 20130181120005), the National Science Foundation for Postdoctoral Scientists of China (no. 2014M552357), the Science and Technology Plan of Sichuan Province (no. 2014GZ0005), and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

- [1] R. Kimmel, "Demosaicing: image reconstruction from color CCD samples," *IEEE Transactions on Image Processing*, vol. 8, no. 9, pp. 1221–1228, 1999.
- [2] X. Li, "Demosaicing by successive approximation," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.
- [3] H. Shen, L. Zhang, B. Huang, and P. Li, "A MAP approach to joint motion estimation, segmentation, and super resolution," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 479–490, 2007.
- [4] L. Zhang, H. Zhang, H. Shen, and P. Li, "A super-resolution reconstruction algorithm for surveillance images," *Signal Processing*, vol. 90, no. 3, pp. 848–859, 2010.
- [5] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.
- [6] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
- [7] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the IEEE*

- Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 275–282, July 2004.
- [8] K. Zhang, X. Gao, X. Li, and D. Tao, “Partially supervised neighbor embedding for example-based image super-resolution,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 2, pp. 230–239, 2011.
- [9] X. Wang and X. Tang, “Hallucinating face by eigen transformation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 35, no. 3, pp. 425–434, 2005.
- [10] W. Wu, Z. Liu, and X. He, “Learning-based super resolution using kernel partial least squares,” *Image and Vision Computing*, vol. 29, no. 6, pp. 394–406, 2011.
- [11] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [12] G. Monaci and P. Vandergheynst, “Learning structured dictionaries for image representation,” in *Proceedings of the International Conference on Image Processing (ICIP '04)*, pp. 2351–2354, IEEE, October 2004.
- [13] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Curves and Surfaces*, J.-D. Boissonnat, P. Chenin, A. Cohen et al., Eds., vol. 6920 of *Lecture Notes in Computer Science*, pp. 711–730, Springer, New York, NY, USA, 2012.
- [14] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with non-local means and steering kernel regression,” *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4544–4556, 2012.
- [15] X. Yang, K. Liu, Z. Gan, and B. Yan, “Multiscale and multitopic sparse representation for multisensor infrared image super-resolution,” *Journal of Sensors*, vol. 2016, Article ID 7036349, 14 pages, 2016.
- [16] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [17] Z. Zhan, J. Cai, D. Guo, Y. Liu, Z. Chen, and X. Qu, “Fast multi-class dictionaries learning with geometrical directions in MRI reconstruction,” *IEEE Transactions on Biomedical Engineering*, 2015.
- [18] R. Timofte, V. De, and L. V. Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 1920–1927, Sydney, Australia, December 2013.
- [19] W. Wu, X. Yang, Y. Pang, J. Peng, and G. Jeon, “A multifocus image fusion method by using hidden Markov model,” *Optics Communications*, vol. 287, pp. 63–72, 2013.
- [20] M. Perd'och, O. Chum, and J. Matas, “Efficient representation of local geometry for large scale object retrieval,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 9–16, Miami, Fla, USA, June 2009.
- [21] A. K. Jain, N. K. Ratha, and S. Lakshmanan, “Object detection using gabor filters,” *Pattern Recognition*, vol. 30, no. 2, pp. 295–309, 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

