

Image Super-Resolution as Sparse Representation of Raw Image Patches

Jianchao Yang[†], John Wright[‡], Thomas Huang[†], Yi Ma[‡]
ECE Department, University of Illinois at Urbana-Champaign, USA
Beckman Institute[†] and Coordinated Science Laboratory[‡]
{jyang29, jnwright, huang, yima@uiuc.edu}

Abstract

This paper addresses the problem of generating a super-resolution (SR) image from a single low-resolution input image. We approach this problem from the perspective of compressed sensing. The low-resolution image is viewed as downsampled version of a high-resolution image, whose patches are assumed to have a sparse representation with respect to an over-complete dictionary of prototype signal-atoms. The principle of compressed sensing ensures that under mild conditions, the sparse representation can be correctly recovered from the downsampled signal. We will demonstrate the effectiveness of sparsity as a prior for regularizing the otherwise ill-posed super-resolution problem. We further show that a small set of randomly chosen raw patches from training images of similar statistical nature to the input image generally serve as a good dictionary, in the sense that the computed representation is sparse and the recovered high-resolution image is competitive or even superior in quality to images produced by other SR methods.

1. Introduction

Conventional approaches to generating a super-resolution (SR) image require multiple low-resolution images of the same scene, typically aligned with sub-pixel accuracy. The SR task is cast as the inverse problem of recovering the original high-resolution image by fusing the low-resolution images, based on assumptions or prior knowledge about the generation model from the high-resolution image to the low-resolution images. The basic reconstruction constraint is that applying the image formation model to the recovered image should produce the same low-resolution images. However, because much information is lost in the high-to-low generation process, the reconstruction problem is severely underdetermined, and the solution is not unique. Various methods have been proposed to further regularize the problem. For instance, one can choose a MAP (maximum a-posteriori) solution

under generic image priors such as Huber MRF (Markov Random Field) and Bilateral Total Variation [14, 11, 25].

However, the performance of these reconstruction-based super-resolution algorithms degrades rapidly if the magnification factor is large or if there are not enough low-resolution images to constrain the solution, as in the extreme case of only a single low-resolution input image [2]. Another class of super-resolution methods that can overcome this difficulty are learning based approaches, which use a learned co-occurrence prior to predict the correspondence between low-resolution and high-resolution image patches [12, 26, 16, 5, 20].

In [12], the authors propose an example-based learning strategy that applies to generic images where the low-resolution to high-resolution prediction is learned via a Markov Random Field (MRF) solved by belief propagation. [23] extends this approach by using the Primal Sketch priors to enhance blurred edges, ridges and corners. Nevertheless, the above methods typically require enormous databases of millions of high-resolution and low-resolution patch pairs to make the databases expressive enough. In [5], the authors adopt the philosophy of LLE [22] from manifold learning, assuming similarity between the two manifolds in the high-resolution patch space and the low-resolution patch space. Their algorithm maps the local geometry of the low-resolution patch space to the high-resolution patch space, generating high-resolution patch as a linear combination of neighbors. Using this strategy, more patch patterns can be represented using a smaller training database. However, using a fixed number K neighbors for reconstruction often results in blurring effects, due to over- or under-fitting.

In this paper, we focus on the problem of recovering the super-resolution version of a given low-resolution image. Although our method can be readily extended to handle multiple input images, we mostly deal with a single input image. Like the aforementioned learning-based methods, we will rely on patches from example images. Our method does not require any learning on the high-resolution patches, instead working directly with the *low-resolution* training patches or their features. Our approach is motivated



Figure 1. Reconstruction of a raccoon face with magnification factor 2. Left: result by our method. Right: the original image. There is little noticeable difference.

by recent results in sparse signal representation, which ensure that linear relationships among high-resolution signals can be precisely recovered from their low-dimensional projections [3, 9].

To be more precise, let $D \in \mathbb{R}^{n \times K}$ be an overcomplete dictionary of K prototype signal-atoms, and suppose a signal $x \in \mathbb{R}^n$ can be represented as a sparse linear combination of these atoms. That is, the signal vector x can be written as $x = D\alpha_0$ where $\alpha_0 \in \mathbb{R}^K$ is a vector with very few ($\ll K$) nonzero entries. In practice, we might observe only a small set of measurements y of x :

$$y \doteq Lx = LD\alpha_0, \quad (1)$$

where $L \in \mathbb{R}^{k \times n}$ with $k < n$. In the super-resolution context, x is a high-resolution image (patch), while y is its low-resolution version (or features extracted from it). If the dictionary D is overcomplete, the equation $x = D\alpha$ is underdetermined for the unknown coefficients α . The equation $y = LD\alpha$ is even more dramatically underdetermined. Nevertheless, under mild conditions, the sparsest solution α_0 to this equation is unique. Furthermore, if D satisfies an appropriate near-isometry condition, then for a wide variety of matrices L , any sufficiently sparse linear representation of a high-resolution image x in terms of the D can be recovered (almost) perfectly from the low-resolution image [9, 21]. Figure 1 shows an example that demonstrates the capabilities of our method derived from this principle. Even for this complicated texture, sparse representation recovers a visually appealing reconstruction of the original signal.

Recently sparse representation has been applied to many other related inverse problems in image processing, such as compression, denoising [10], and restoration [17], often improving on the state-of-the-art. For example in [10], the authors use the K-SVD algorithm [1] to learn an overcomplete dictionary from natural image patches and successfully apply it to the image denoising problem. In our setting, we do not directly compute the sparse representation of the high-resolution patch. Instead, we will work with two coupled dictionaries, D_h for high-resolution patches, and $D_\ell = LD_h$ for low-resolution patches. The sparse representation of a low-resolution patch in terms of D_ℓ will be

directly used to recover the corresponding high-resolution patch from D_h . We obtain a locally consistent solution by allowing patches to overlap and demanding that the reconstructed high-resolution patches agree on the overlapped areas. Finally, we apply global optimization to eliminate the reconstruction errors in the recovered high-resolution image from local sparse representation, suppressing noise and ensuring consistency with the low-resolution input.

Compared to the aforementioned learning-based methods, our algorithm requires a much smaller database. The online recovery of the sparse representation uses the low-resolution dictionary only – the high-resolution dictionary is used only to calculate the final high-resolution image. The computation, mainly based on linear programming, is reasonably efficient and scalable. In addition, the computed sparse representation adaptively selects the most relevant patches in the dictionary to best represent each patch of the given low-resolution image. This leads to superior performance, both qualitatively and quantitatively, compared to methods [5] that use a fixed number of nearest neighbors, generating sharper edges and clearer textures.

The remainder of this paper is organized as follows. Section 2 details our formulation and solution to the image super-resolution problem based on sparse representation. In Section 3, we discuss how to prepare a dictionary from sample images and what features to use. Various experimental results in Section 4 demonstrate the efficacy of sparsity as a prior for image super-resolution.

2. Super-resolution from Sparsity

The single-image super-resolution problem asks: given a low-resolution image Y , recover a higher-resolution image X of the same scene. The fundamental constraint is that the recovered X should be consistent with the input, Y :

Reconstruction constraint. The observed low-resolution image Y is a blurred and downsampled version of the solution X :

$$Y = DHX \quad (2)$$

Here, H represents a blurring filter, and D the downsampling operator.

Super-resolution remains extremely ill-posed, since for a given low-resolution input Y , infinitely many high-resolution images X satisfy the above reconstruction constraint. We regularize the problem via the following prior on small patches x of X :

Sparse representation prior. The patches x of the high-resolution image X can be represented as a sparse linear combination in a dictionary D_h of high-resolution patches sampled from training images:¹

$$x \approx D_h \alpha \quad \text{for some } \alpha \in \mathbb{R}^K \text{ with } \|\alpha\|_0 \ll K. \quad (3)$$

¹Similar mechanisms – sparse coding with an overcomplete dictionary – are also believed to be employed by the human visual system [9].

To address the super-resolution problem using the sparse representation prior, we divide the problem into two steps. First, using the sparse prior (3), we find the sparse representation for each local patch, respecting spatial compatibility between neighbors. Next, using the result from this local sparse representation, we further regularize and refine the entire image using the reconstruction constraint (2). In this strategy, a local model from the sparse prior is used to recover lost high-frequency for local details. The global model from the reconstruction constraint is then applied to remove possible artifacts from the first step and make the image more consistent and natural.

2.1. Local Model from Sparse Representation

As in the patch-based methods mentioned previously, we try to infer the high-resolution patch for each low-resolution patch from the input. For this local model, we have two dictionaries D_ℓ and D_h : D_h is composed of high-resolution patches and D_ℓ is composed of corresponding low-resolution patches. We subtract the mean pixel value for each patch, so that the dictionary represents image textures rather than absolute intensities.

For each input low-resolution patch \mathbf{y} , we find a sparse representation with respect to D_ℓ . The corresponding high-resolution patches D_h will be combined according to these coefficients to generate the output high-resolution patch \mathbf{x} . The problem of finding the sparsest representation of \mathbf{y} can be formulated as:

$$\min \|\alpha\|_0 \quad \text{s.t.} \quad \|FD_\ell\alpha - F\mathbf{y}\|_2^2 \leq \epsilon, \quad (4)$$

where F is a (linear) feature extraction operator. The main role of F in (4) is to provide a perceptually meaningful constraint² on how closely the coefficients α must approximate \mathbf{y} . We will discuss the choice of F in Section 3.

Although the optimization problem (4) is NP-hard in general, recent results [7, 8] indicate that as long as the desired coefficients α are sufficiently sparse, they can be efficiently recovered by instead minimizing the ℓ^1 -norm, as follows:

$$\min \|\alpha\|_1 \quad \text{s.t.} \quad \|FD_\ell\alpha - F\mathbf{y}\|_2^2 \leq \epsilon. \quad (5)$$

Lagrange multipliers offer an equivalent formulation

$$\min \lambda \|\alpha\|_1 + \frac{1}{2} \|FD_\ell\alpha - F\mathbf{y}\|_2^2, \quad (6)$$

where the parameter λ balances sparsity of the solution and fidelity of the approximation to \mathbf{y} . Notice that this is essentially a linear regression regularized with ℓ^1 -norm on the coefficients, known in statistical literature as the Lasso [24].

²Traditionally, one would seek the sparsest α s.t. $\|D_\ell\alpha - \mathbf{y}\|_2 \leq \epsilon$. For super-resolution, it is more appropriate to replace this 2-norm with a quadratic norm $\|\cdot\|_{FTF}$ that penalizes visually salient high-frequency errors.

Solving (6) individually for each patch does not guarantee compatibility between adjacent patches. We enforce compatibility between adjacent patches using a one-pass algorithm similar to that of [13].³ The patches are processed in raster-scan order in the image, from left to right and top to bottom. We modify (5) so that the super-resolution reconstruction $D_h\alpha$ of patch \mathbf{y} is constrained to closely agree with the previously computed adjacent high-resolution patches. The resulting optimization problem is

$$\min \|\alpha\|_1 \quad \text{s.t.} \quad \|FD_\ell\alpha - F\mathbf{y}\|_2^2 \leq \epsilon_1 \\ \|PD_h\alpha - \mathbf{w}\|_2^2 \leq \epsilon_2, \quad (7)$$

where the matrix P extracts the region of overlap between current target patch and previously reconstructed high-resolution image, and \mathbf{w} contains the values of the previously reconstructed high-resolution image on the overlap. The constrained optimization (7) can be similarly reformulated as:

$$\min \lambda \|\alpha\|_1 + \frac{1}{2} \|\tilde{D}\alpha - \tilde{\mathbf{y}}\|_2^2, \quad (8)$$

where $\tilde{D} = \begin{bmatrix} FD_\ell \\ \beta PD_h \end{bmatrix}$ and $\tilde{\mathbf{y}} = \begin{bmatrix} F\mathbf{y} \\ \beta \mathbf{w} \end{bmatrix}$. The parameter β controls the tradeoff between matching the low-resolution input and finding a high-resolution patch that is compatible with its neighbors. In all our experiments, we simply set $\beta = 1$. Given the optimal solution α^* to (8), the high-resolution patch can be reconstructed as $\mathbf{x} = D_h\alpha^*$.

2.2. Enforcing Global Reconstruction Constraint

Notice that (5) and (7) do not demand exact equality between the low-resolution patch \mathbf{y} and its reconstruction $D_\ell\alpha$. Because of this, and also because of noise, the high-resolution image X_0 produced by the sparse representation approach of the previous section may not satisfy the reconstruction constraint (2) exactly. We eliminate this discrepancy by projecting X_0 onto the solution space of $DHX = Y$, computing

$$X^* = \arg \min_X \|X - X_0\| \quad \text{s.t.} \quad DHX = Y. \quad (9)$$

The solution to this optimization problem can be efficiently computed using the back-projection method, originally developed in computer tomography and applied to super-resolution in [15, 4]. The update equation for this iterative method is

$$X_{t+1} = X_t + ((Y - DHX_t) \uparrow s) * p, \quad (10)$$

where X_t is the estimate of the high-resolution image after the t -th iteration, p is a ‘‘backprojection’’ filter, and $\uparrow s$ denotes upsampling by a factor of s .

³There are different ways to enforce compatibility. In [5], the values in the overlapped regions are simply averaged, which will result in blurring effects. The one-pass algorithm [13] is shown to work almost as well as the use of a full MRF model [12].

Algorithm 1 (Super-resolution via Sparse Representation).

- 1: **Input:** training dictionaries D_h and D_ℓ , a low-resolution image Y .
 - 2: **for** each 3×3 patch y of Y , taken starting from the upper-left corner with 1 pixel overlap in each direction,
 - Solve the optimization problem with \tilde{D} and \tilde{y} defined in (8): $\min \lambda \|\alpha\|_1 + \frac{1}{2} \|\tilde{D}\alpha - \tilde{y}\|_2^2$.
 - Generate the high-resolution patch $x = D_h \alpha^*$. Put the patch x into a high-resolution image X_0 .
 - 3: **end**
 - 4: Using back-projection, find the closest image to X_0 which satisfies the reconstruction constraint:
$$X^* = \arg \min_X \|X - X_0\| \quad \text{s.t.} \quad DHX = Y.$$
 - 5: **Output:** super-resolution image X^* .
-

We take result X^* from backprojection as our final estimate of the high-resolution image. This image is as close as possible to the initial super-resolution X_0 given by sparsity, while satisfying the reconstruction constraint. The entire super-resolution process is summarized as Algorithm 1.

2.3. Global Optimization Interpretation

The simple SR algorithm outlined above can be viewed as a special case of a general sparse representation framework for inverse problems in image processing. Related ideas have been profitably applied in image compression, denoising [10], and restoration [17]. These connections provide context for understanding our work, and also suggest means of further improving the performance, at the cost of increased computational complexity.

Given sufficient computational resources, one could in principle solve for the coefficients associated with all patches *simultaneously*. Moreover, the entire high-resolution image X itself can be treated as a variable. Rather than demanding that X be perfectly reproduced by the sparse coefficients α , we can penalize the difference between X and the high-resolution image given by these coefficients, allowing solutions that are not perfectly sparse, but better satisfy the reconstruction constraints. This leads to a large optimization problem:

$$X^* = \arg \min_{X, \{\alpha_{ij}\}} \left\{ \|DHX - Y\|_2^2 + \eta \sum_{i,j} \|\alpha_{ij}\|_0 + \gamma \sum_{i,j} \|D_h \alpha_{ij} - P_{ij} X\|_2^2 + \tau \rho(X) \right\}. \quad (11)$$

Here, α_{ij} denotes the representation coefficients for the $(i, j)_{th}$ patch of X , and P_{ij} is a projection matrix that selects the $(i, j)_{th}$ patch from X . $\rho(X)$ is a penalty function that encodes prior knowledge about the high-resolution image. This function may depend on the image category, or

may take the form of a generic regularization term (e.g., Huber MRF, Total Variation, Bilateral Total Variation).

Algorithm 1 can be interpreted as a computationally efficient approximation to (11). The sparse representation step recovers the coefficients α by approximately minimizing the sum of the second and third terms of (11). The sparsity term $\|\alpha_{ij}\|_0$ is relaxed to $\|\alpha_{ij}\|_1$, while the high-resolution fidelity term $\|D_h \alpha_{ij} - P_{ij} X\|_2$ is approximated by its low-resolution version $\|F D_\ell \alpha_{ij} - F y_{ij}\|_2$.

Notice, that if the sparse coefficients α are fixed, the third term of (11) essentially penalizes the difference between the super-resolution image X and the reconstruction given by the coefficients: $\sum_{i,j} \|D_h \alpha_{ij} - P_{ij} X\|_2^2 \approx \|X_0 - X\|_2^2$. Hence, for small γ , the back-projection step of Algorithm 1 approximately minimizes the sum of the first and third terms of (11).

Algorithm 1 does not, however, incorporate any prior besides sparsity of the representation coefficients – the term $\rho(X)$ is absent in our approximation. In Section 4 we will see that sparsity in a relevant dictionary is a strong enough prior that we can already achieve good super-resolution performance. Nevertheless, in settings where further assumptions on the high-resolution signal are available, these priors can be incorporated into the global reconstruction step of our algorithm.

3. Dictionary Preparation

3.1. Random Raw Patches from Training Images

Learning an over-complete dictionary capable of optimally representing broad classes of image patches is a difficult problem. Rather than trying to learn such a dictionary [19, 1] or using a generic set of basis vectors [21] (e.g., Fourier, Haar, curvelets etc.), we generate dictionaries by simply *randomly sampling raw patches* from training images of similar statistical nature. We will demonstrate that so simply prepared dictionaries are already capable of generating high-quality reconstructions,⁴ when used together with the sparse representation prior.

Figure 2 shows several training images and the patches sampled from them. For our experiments, we prepared two dictionaries: one sampled from flowers (Figure 2 top), which will be applied to generic images with relative simple textures, and one sampled from animal images (Figure 2 bottom), with fine furry or fractal textures. For each high-resolution training image X , we generate the corresponding low-resolution image Y by blurring and downsampling. For each category of images, we sample only about 100,000 patches from about 30 training images to form each dictionary, which is considerably smaller than that needed by

⁴The competitiveness of such random patches has also been noticed empirically in the context of content-based image classification [8].



Figure 2. Left: three out of the 30 training images we use in our experiments. Right: the training patches extracted from them.

other learning-based methods [12, 23]. Empirically, we find such a small dictionary is more than sufficient.

3.2. Derivative Features

In (4), we use a feature transformation F to ensure that the computed coefficients fit the most relevant part of the low-resolution signal. Typically, F is chosen as some kind of high-pass filter. This is reasonable from a perceptual viewpoint, since people are more sensitive to the high-frequency content of the image. The high-frequency components of the low-resolution image are also arguably the most important for predicting the lost high-frequency content in the target high-resolution image.

Freeman et al. [12] use a high-pass filter to extract the edge information from the low-resolution input patches as the feature. Sun et al. [23] use a set of Gaussian derivative filters to extract the contours in the low-resolution patches. Chang et al. [5] use the first-order and second-order gradients of the patches as the representation. For our algorithm, we also use the first-order and second-order derivatives as the feature for the low-resolution patch. While simple, these features turn out to work very well. To be precise, the four 1-D filters used to extract the derivatives are:

$$\begin{aligned} \mathbf{f}_1 &= [-1, 0, 1], & \mathbf{f}_2 &= \mathbf{f}_1^T, \\ \mathbf{f}_3 &= [1, 0, -2, 0, 1], & \mathbf{f}_4 &= \mathbf{f}_3^T, \end{aligned} \quad (12)$$

where the superscript “ T ” means transpose. Applying these four filters, we get four description feature vectors for each patch, which are concatenated as one vector as the final representation of the low-resolution patch.

4. Experiments

Experimental settings: In our experiments, we will mostly magnify the input image by a factor of 3. 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. The features are not extracted directly from the 3×3 low-resolution patch, but rather from an upsampled version produced by bicubic interpolation. For color images, we apply our algorithm to the illuminance

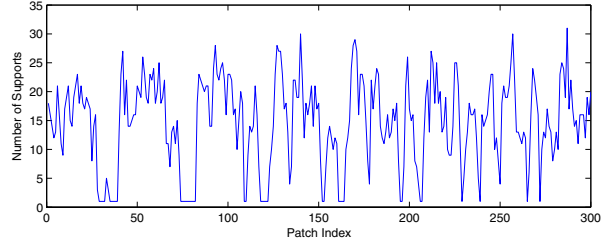


Figure 3. Number of nonzero coefficients in the sparse representation computed for 300 typical patches in a test image.

component only, since humans are more sensitive to illuminance changes. Our algorithm has only one free parameter λ , which balances sparsity of the solution with fidelity to the reconstruction constraint. In our experience, the reconstruction quality is stable over a large range of λ . The rule of thumb, $\lambda = 50 \times \dim(\text{patch feature})$, gives good results for all the test cases in this paper.

One advantage of our approach over methods such as neighbor embedding [5] is that it selects the number of relevant dictionary elements *adaptively* for each patch. Figure 3 demonstrates this for 300 typical patches in one test image. Notice that the recovered coefficients are always sparse (< 35 nonzero entries), but the level of sparsity varies depending on the complexity of each test patch. However, empirically, we find the support of the recovered coefficients typically is neither a superset nor subset of the K nearest neighbors [5]. The chosen patches are more informative for recovering the high-resolution patch, leading to more faithful texture reconstruction in the experiments below.

Experimental results: We first apply our algorithm to generic images including flower, human face, and architecture, *all* using the same dictionary sampled from training images of flowers (first row of Figure 2). We will further demonstrate our algorithm’s ability to handle complicated textures in animal images, with the second dictionary sampled from training animal images (second row of Figure 2).

Figure 4 compares our results with neighbor embedding [5]⁵ on two test images of a flower and a girl. In both cases, our method gives sharper edges and reconstructs more clearly the details of the scene. There are noticeable differences in the texture of the leaves, the fuzz on the leaf-stalk, and also the freckles on the face of the girl.

In Figure 5, we compare our method with several other methods on an image of the Parthenon used in [6], including back projection, neighbor embedding [5], and the recently

⁵Our implementation of the neighbor embedding method [5] differs slightly from the original. The feature for the low-resolution patch is not extracted from the original 3×3 patch, which will give smoother results, but on the upsampled low-resolution patch. We find that setting $K = 15$ gives the best performance. This is approximately the average number of coefficients recovered by sparse representation (see Figure 3).



Figure 4. The flower and girl image magnified by a factor of 3. Left to right: input, bicubic interpolation, neighbor embedding [5], our method, and the original. (Also see Figure 8 for the same girl image magnified by a factor of 4).

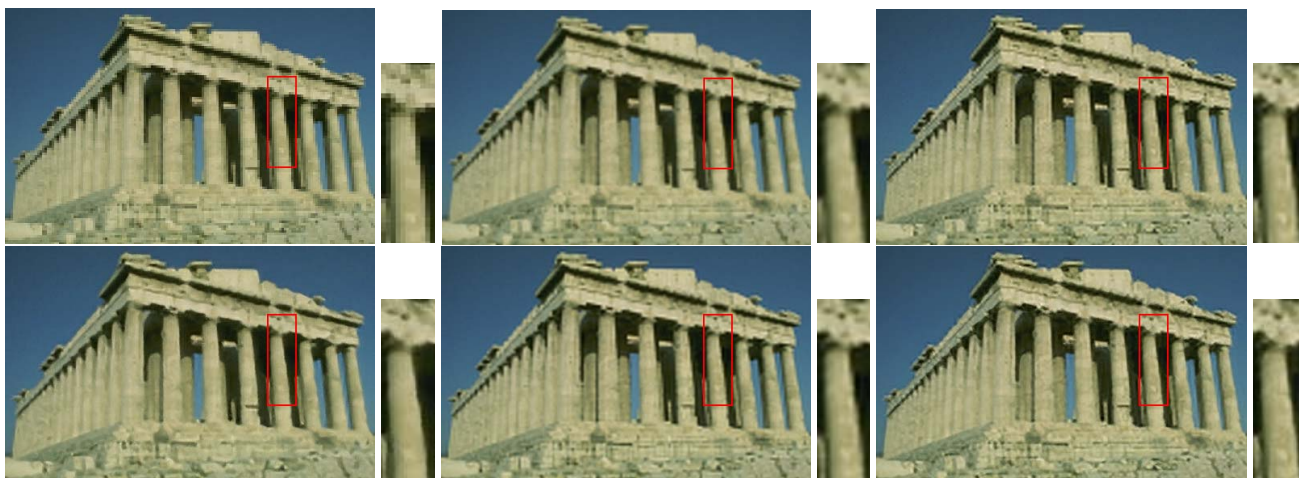


Figure 5. Results on an image of the Parthenon with magnification factor 3. Top row: low-resolution input, bicubic interpolation, back projection. Bottom row: neighbor embedding[5], soft edge prior [6], and our method.

proposed method based on a learned soft edge prior [6]. The result from back projection has many jagged effects along the edges. Neighbor embedding generates sharp edges in places, but blurs the texture on the temple’s facade. The soft edge prior method gives a decent reconstruction, but introduces undesired smoothing that is not present in our result. Additional results on generic images using this dictionary are shown in Figure 7 left and center. Notice that in both cases, the algorithm significantly improves the image resolution by sharpening edges and textures.

We now conduct more challenging experiments on more intricate textures found in animal images, using the animal dictionary with merely 100,000 training patches (second row of Figure 2). As already shown in Figure 1, our method performs quite well in magnifying the image of a raccoon face by a factor of 2. When complex textures such as this one are down-sampled further, the SR task will be-

come more difficult than on images with simpler textures, such as flowers or faces. In Figure 6, we apply our method to the same raccoon face image with magnification factor 3. Since there are no explicit edges in most part of the image, methods proposed in [12], [23], and [6] would have tremendous difficulty here. Compared to neighbor embedding [5], our method gives clearer fur and sharper whiskers. Figure 7 shows an additional image of a cat face reconstructed using this dictionary. We compare several SR methods quantitatively in terms of their RMS errors for some of the images shown above. The results are shown in Table 1.

Finally, we test our algorithm on the girl image again, but with a more challenging magnification factor 4. The results are shown in Figure 8. Here, back-projection again yields jagged edges. Freeman et. al’s method [12] introduces many artifacts and fails to capture the facial texture, despite relying on a much larger database. Compared to the soft edge



Figure 6. A raccoon face magnified by a factor of 3. The input image, bicubic interpolation, neighbor embedding, and our method.

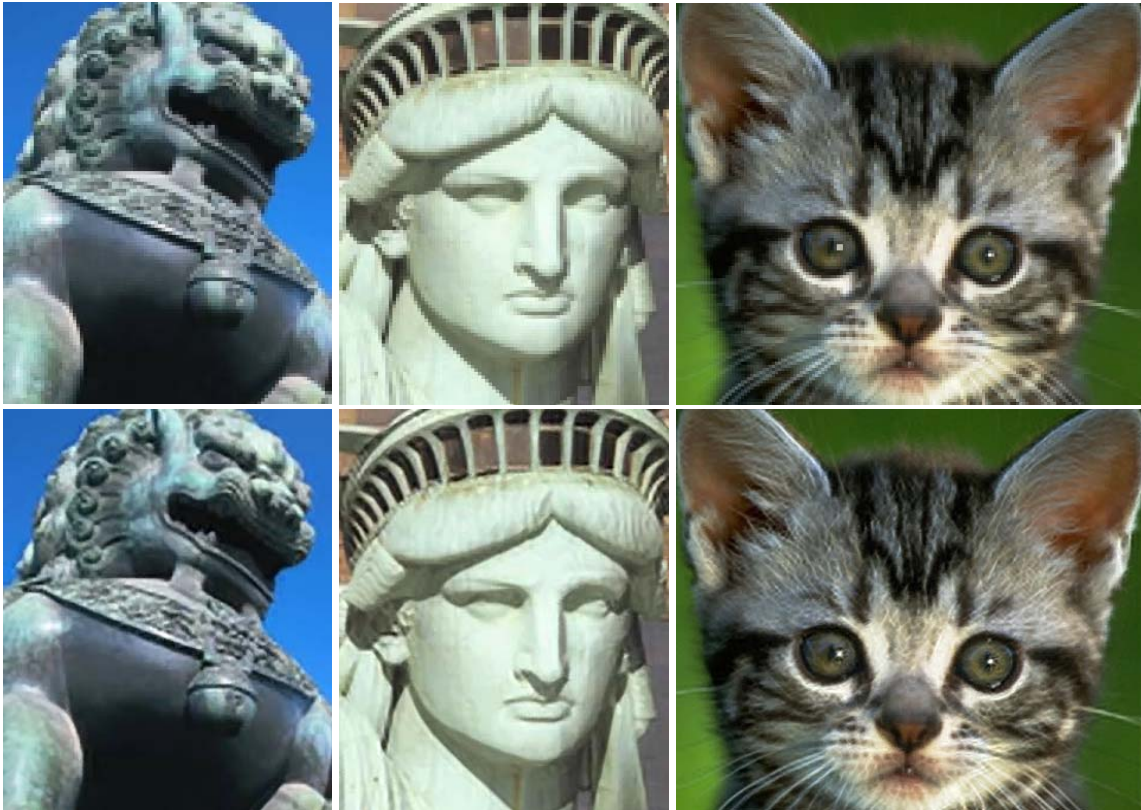


Figure 7. More results on a few more generic (left and center) and animal (right) images. Top: input images. Bottom: super-resolution images by our method, with magnification factor 3.



Figure 8. The girl image magnified by a factor of 4. From left to right: low-resolution input, back projection, learning-based method in [12], soft edge prior [6], and our method.

Images	Bicubic	NE [5]	Our method
Flower	3.5052	4.1972	3.2276
Girl	5.9033	6.6588	5.6175
Parthenon	12.7431	13.5562	12.2491
Raccoon	9.7399	9.8490	9.1874

Table 1. The RMS errors of different methods for super-resolution with magnification factor 3, respect to the original images.

prior method [6], our method generates shaper edges and is more faithful to the original facial texture.

5. Discussion

The experimental results of the previous section demonstrate the effectiveness of sparsity as a prior for patch-based super-resolution. However, one of the most important questions for future investigation is to determine, in terms of the within-category variation, the number of raw sample patches required to generate a dictionary satisfying the sparse representation prior. Tighter connections to the theory of compressed sensing may also yield conditions on the appropriate patch size or feature dimension.

Acknowledgement

This work was partially supported by the following grants: NSF EHS-0509151, NSF CCF-0514955, ONR YIP N00014-05-1-0633, NSF IIS 07-03756, and U.S. Government VACE Program.

References

- [1] 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, November 2006. **2, 4**
- [2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE TPAMI*, 24(9):1167-1183, 2002. **1**
- [3] E. Candes. Compressive sensing. *Proc. International Congress of Mathematicians*, 2006. **2**
- [4] D. Capel. Image mosaicing and super-resolution. Ph.D. Thesis, Department of Eng. Science, University of Oxford, 2001. **3**
- [5] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. *CVPR*, 2004. **1, 2, 3, 5, 6, 8**
- [6] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution *Proc. ICCV*, 2007. **5, 6, 7, 8**
- [7] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal ℓ^1 -norm solution is also the sparsest solution. *Comm. on Pure and Applied Math*, Vol. 59, No. 6, 2006. **3**
- [8] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal ℓ^1 -norm near-solution approximates the sparsest near-solution. Preprint, accessed at <http://www-stat.stanford.edu/~donoho/>. 2004. **3**
- [9] D. L. Donoho. Compressed sensing. Preprint, accessed at <http://www-stat.stanford.edu/~donoho/>. 2005. **2**
- [10] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE TIP*, Vol. 15, No. 12, 2006. **2, 4**
- [11] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super-resolution. *IEEE TIP*, 2004. **1**
- [12] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 2000. **1, 3, 5, 6, 7**
- [13] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, Vol. 22, Issue 2, 2002. **3**
- [14] R.C. Hardie, K.J. Barnard, and E.A. Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE TIP*, 1997. **1**
- [15] M. Irani and S. Peleg. Motion analysis for image enhancement: resolution, occlusion and transparency. *JVCI*, 1993. **3**
- [16] C. Liu, H. Y. Shum, and W. T. Freeman. Face hallucination: theory and practice. *IJCV*, Vol. 75, No. 1, pp. 115-134, October, 2007. **1**
- [17] J. Mairal, G. Sapiro, and M. Elad. Learning multi-scale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 2008. **2, 4**
- [18] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. *Proc. ECCV*, 2006. **4**
- [19] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311-3325, 1997. **2, 4**
- [20] L. C. Pickup, S. J. Roberts, and A. Zisserman. A sampled texture prior for image super-resolution. *Proc. NIPS*, 2003. **1**
- [21] H. Rauhut, K. Schnass, and P. Vandergheynst. Compressed sensing and redundant dictionaries. Preprint, accessed at <http://homepage.univie.ac.at/holger.rauhut/>. 2007. **2, 4**
- [22] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500): 2323-2326, 2000. **1**
- [23] J. Sun, N.-N. Zheng, H. Tao, and H. Shum. Image hallucination with primal sketch priors. *Proc. CVPR*, 2003. **1, 5, 6**
- [24] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal Statist. Soc B.*, Vol. 58, No. 1, pages 267-288, 1996. **3**
- [25] M. E. Tipping and C. M. Bishop. Bayesian image super-resolution. *Proc. NIPS*, 2003. **1**
- [26] Q. Wang, X. Tang, and H. Shum. Patch based blind image super resolution. *Proc. ICCV*, 2005. **1**