

Bayesian region selection for adaptive dictionary-based Super-Resolution

Eduardo Pérez-Pellitero^{1, 2}
 eduardo.perezpellitero@technicolor.com

Jordi Salvador¹
 jordi.salvador@technicolor.com

Javier Ruiz-Hidalgo³
 j.ruiz@upc.edu

Bodo Rosenhahn²
 rosenhahn@tnt.uni-hannover.de

¹ Image Processing Lab
 Technicolor R&I
 Hannover, Germany

² Institute for Information Processing
 Leibniz Universität Hannover
 Hannover, Germany

³ Image Processing Group
 Universitat Politècnica de Catalunya
 Barcelona, Spain

Abstract

The performance of dictionary-based super-resolution (SR) strongly depends on the contents of the training dataset. Nevertheless, many dictionary-based SR methods randomly select patches from of a larger set of training images to build their dictionaries [8, 24, 19, 20], thus relying on patches being diverse enough. This paper describes an external-dictionary SR algorithm based on adaptively selecting an optimal subset of patches out of the training images. Each training image is divided into sub-image entities, named regions, of such size that texture consistency is preserved. For each input patch to super-resolve, the best-fitting region (with enough high-frequency energy) is found through a Bayesian selection. In order to handle the high number of regions in the training dataset, a local Naive Bayes Nearest Neighbor (NBNN) approach is used. Trained with this adapted subset of patches, sparse coding SR is applied to recover the high-resolution image. Experimental results demonstrate that using our adaptive algorithm produces an improvement in SR performance with respect to non-adaptive training.

1 Introduction

The fast development and popularization of low-cost capturing devices contrasts with the proliferation of high-definition displays. Applications delivering low resolution images are diverse (e.g. surveillance, satellite) and there is also abundant multimedia content whose resolution is not up-to-date with current display's capabilities. To fill in this gap, Super-resolution (SR) techniques are used. SR deals with an inverse ill-posed problem which aims to obtain a high-resolution (HR) image from its low-resolution (LR) version by restoring the potential available hidden information. SR approaches can be broadly divided into three different categories: reconstruction-based, interpolation-based and learning-based methods.

Early approaches were based on multi-image reconstruction SR [5, 9]. These reconstruction techniques need as input several images of the same scene with sub-pixel displacements,

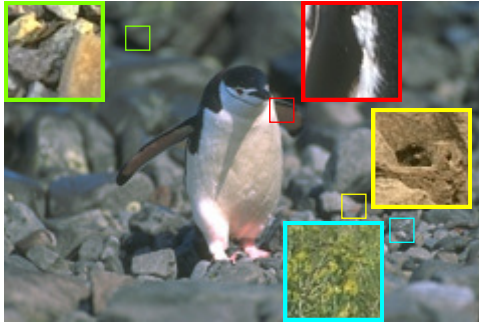


Figure 1: An example of how our method selects training regions based on Bayes rule. For every patch in the input image, a highly correlated region from the training images is found. The resulting set of regions is used to build a texture-adapted dictionary which will be used to recover the HR image.

which are used to build a set of linear constraints for the new high-resolution pixel intensities. If enough images are provided, the set of equations is determined and can be solved to obtain the HR image. This approach, however, depends on the accuracy of the required registration process and is limited to small magnification factors [10, 12].

First efforts in interpolation methods used well-known interpolation kernels such as bilinear or bicubic [13], which are fast and non-complex but tend to produce overly smooth edges. Further research in interpolation methods exploit natural image priors [9, 14], yielding improved results but still having a limited performance when dealing with complex textures.

Learning-based SR methods overcame some of the already mentioned limitations [9, 15, 16]. These methods aim to learn the relation from LR to HR from a training dataset, usually at a patch level. In [9] the prediction from LR to HR patches is learned through a Markov Random Field and solved by belief propagation. This idea was extended by [16] using primal sketch priors (e.g. edges, ridges and corners). However, these approaches require large training datasets, in the order of millions of patch pairs, thus being computationally costly. [9] proposed a manifold assumption where LR and HR manifolds have similar local geometry in two distinct feature spaces. Following this assumption, locally linear embedding (LLE) is used to estimate HR patches by combining the respective HR counterparts of the LR patches found in the training dataset. Recent SR research explored the sparsity prior, where LR patches are coded with respect to an over-complete dictionary in order to linearly combine their HR counterparts with the same sparse codes [17].

The performance of learning-based SR methods depends strongly on the content of the training dataset. In [18] the dictionary is built by randomly sampling raw patches from a large set of images regardless of the image to be recovered, hence relying on gathering sufficiently diverse patches so that they can generalize for any patch to be super-resolved. More recent follow-up works [19, 20] keep using the same strategy for the training, although these raw patches are compressed in a smaller number of patches through sparse coding techniques. In the Neighbor Embedding SR work of [8], a clustering in the training set is performed based on geometrical structure of patches. The k -NN query of the input LR patch is then carried out within the closest cluster, thus showing some adaptive behavior. Nevertheless, the patches to be included in the clustering are also randomly selected out of a larger set of training patches.

This paper proposes a novel sparse SR method, which focuses on adaptively selecting optimal patches for the dictionary training. Observing the fact that images usually contain a non-predictable group of different elements with their characteristic textures and edges (e.g. grass, rocks, fur, sand), we divide our training dataset into sub-image entities which we call regions, and extract descriptors in order to characterize them. The key idea is that, being these regions smaller, they have more consistent texture or edge content. For every patch to be super-resolved we find its best-fitting texture region from the training images by using the efficient local *Naive Bayes Nearest Neighbor* (NBNN), thus ensuring the obtained example pairs are highly correlated with the input LR patches. Furthermore, our method is not only applicable to the original sparse SR [19], but to all other SR methods using a reduced patch pair subset from the larger training image dataset, hence including dictionary optimization processes as the one discussed in Section 2.1 [24].

The rest of the paper is organized as follows. Section 2 reviews the sparse representation approach to SR. In Section 3, details about the proposed work and its formulation are given. The experimental results obtained and the comparison with other methods are shown in Section 4. Finally, Section 5 concludes this paper.

2 Sparse representation approach to super-resolution

The core idea of sparse signal representation is that linear relationships between signals can be precisely reconstructed from their low-dimensional projections [25]. The application of sparse signal representation to SR was firstly proposed by [19] and has been a very active research topic [24, 20, 21]. In this section we provide a brief presentation of the method.

Let X be a HR image from which we extract patches x and Y a LR image from which we extract patches y fulfilling the following reconstruction constraint:

$$Y = (H(X)) \downarrow, \quad (1)$$

being $(\cdot) \downarrow$ a decimator operator and $H(\cdot)$ a transfer function modeling blur. For a given Y , recovering X stays an ill-posed problem since many HR images comply with (1). Sparse SR approaches solve this ill-posed problem by assuming the sparsity prior

$$x \approx D_h \alpha \quad \text{for some } \alpha \in \mathbb{R}^K \text{ with } \|\alpha\|_0 \ll K, \quad (2)$$

where α is the sparse representation with reduced non-zero entries ($\ll K$) and D_h an over-complete dictionary containing HR patches. To recover x , the sparse representation α will be calculated from LR patches y with respect to a dictionary containing the correspondent LR patches D_l and using a feature transformation F :

$$\min \|\alpha\|_0 \quad s.t. \quad \|FD_l \alpha - Fy\|_2^2 \leq \varepsilon. \quad (3)$$

This optimization problem can be rephrased, using Lagrange multipliers, with the equivalent formulation proposed by [24]:

$$\min_{\alpha} \|FD_l \alpha - Fy\|_2^2 + \lambda \|\alpha\|_0, \quad (4)$$

where the regularization parameter λ provides a trade-off between the sparsity and accuracy of the found solution. Since patches extracted from an image are commonly sampled with

a certain overlap, the optimization problem in (4) needs to be redefined in such a way that adjacent patches are compatible:

$$\min_{\alpha} \|\tilde{D}\alpha - \tilde{y}\|_2^2 + \lambda \|\alpha\|_0, \quad (5)$$

where $\tilde{D} = \begin{bmatrix} FD_l \\ PD_h \end{bmatrix}$ and $\tilde{y} = \begin{bmatrix} Fy \\ w \end{bmatrix}$, being P the overlap region between the current patch and the previously reconstructed HR image, and w the previously reconstructed values for the overlap region. After solving the optimization problem of (5) we can reconstruct the HR patches by applying the found optimal solution as $x = D_h \alpha^*$.

2.1 Dictionary Training

The simplest way to obtain the dictionaries needed for sparse SR is to directly extract raw patches from the training images [19]. By doing so, we will need large dictionaries which will substantially increase the computational cost of the optimization problem in (5) ([20] shows computational time grows linearly to dictionary size). Later work of [21] tackles this problem by using sparse coding techniques. The goal of sparse coding is to represent input vectors as a weighted linear combination of a smaller set of *basis vectors*, extracting high level patterns of the input unlabeled data [22], hence having a more compact dictionary.

Let $T = [t_1, t_2, \dots, t_n] \in \mathbb{R}^{m \times n}$ be the training patches and $D = [d_1, d_2, \dots, d_k] \in \mathbb{R}^{m \times k}$ an overcomplete dictionary containing all the patch basis. Let $S = [s_1, s_2, \dots, s_n] \in \mathbb{R}^{k \times n}$ be the coefficients for all training patches in T . These coefficient and the dictionary should best approximate T , thus we use the objective function

$$\min_{D, S} \|T - DS\|_2^2 + \lambda \|S\|_1 \quad s.t. \quad \|d_i\|_2^2 \leq c, \quad i = 1, 2, \dots, b, \quad (6)$$

where λ is the regularization parameter, $\|S\|_1$ enforces sparsity of the solution and $\|d_i\|_2^2$ removes scale ambiguity. The formulation in (6) has been broadly studied, and although not being convex in both D and S , it is convex in one of them while the other is fixed [23]. Consequently, in order to minimize (6), an alternate iterative scheme is used, i.e. updating one variable while the other is fixed until convergence.

In SR it is usually needed a pair training data matrices T_h and T_l with HR and LR data points respectively. To jointly train two dictionaries so that the sparse representation of the HR datapoints is the same as the sparse representation of the correspondent LR datapoints the following minimization is performed:

$$\min_{\{D_h, D_l, S\}} \frac{1}{L_h} \|T_h - D_h S\|_2^2 + \frac{1}{L_l} \|T_l - D_l S\|_2^2 + \lambda \left(\frac{1}{L_h} + \frac{1}{L_l} \right) \|S\|_1, \quad (7)$$

where L_h and L_l are the dimensions of HR and LR patches, respectively.

3 Proposed Method

The performance of sparse SR methods highly depends on the content of D_h and D_l and those in turn are determined by the contents of the training examples T_h and T_l , thus being these subsets of capital importance for the whole SR process. In contrast to previous methods that

build dictionaries selecting randomly patches from the training images [8, 12, 19, 20], in this section we introduce a novel adaptive SR method that builds T_h and T_l from the training images based on the input LR image Y .

Our approach includes a stage which adaptively selects the regions of the training images which better represent each of the input image patches without doing any manual image pre-selection. The key idea is to extract training pair patches only from the regions likely to contain similar textures to the ones present in the image. By doing so, the training of the dictionary in Section 2.1 will have a texture-adapted set of training pairs T_h and T_l .

3.1 Building adaptive dictionaries through NBN

Each training image I_T is split in square regions R of size L_R . Given a patch y we find its training texture region R . Assuming a uniform region prior over R this can be achieved through a maximum likelihood (ML) decision rule:

$$\hat{R} = \underset{R}{\operatorname{argmax}} p(R | y) = \underset{R}{\operatorname{argmax}} p(y | R). \quad (8)$$

Let $\{f\} = f_1, f_2, \dots, f_n$ denote the descriptors extracted from patch y (further details about the used descriptors can be found in Section 3.3). We use the Naive Bayes assumption, i.e. descriptors are independent, identically distributed:

$$p(y | R) = p(f_1, f_2, \dots, f_n | R) = \prod_{i=1}^n p(f_i | R), \quad (9)$$

then, the log likelihood reads:

$$\hat{R} = \underset{R}{\operatorname{argmax}} \sum_{i=1}^n \log p(f_i | R). \quad (10)$$

This maximum-a-posteriori (MAP) decision requires computing the probability density $p(f | R)$, which can be obtained through a nearest-neighbor (NN) approximation of a Parzen density estimation $p_{NN}(f | R)$, as proposed by [2]. For that purpose, let then $\{f^R\} = f_1^R, f_2^R, \dots, f_L^R$ be all the descriptors of a region R , where f_j^R is the j th descriptor. The Parzen kernel $K(f_i - f_j^R) = \exp(-\frac{1}{2\sigma^2} \|f_i - f_j^R\|^2)$ yields negligible values for very distant descriptors since K exponentially decreases with distance. Therefore, using only the r NN of descriptor f will accurately approximate the Parzen estimation:

$$p_{NN}(f_i | R) = \frac{1}{L} \sum_{j=1}^r K(f_i - jNN_R(f_i)) \quad (11)$$

In [2] a minor decrease in performance is observed when using as little as $r = 1$ NN compared to the full Parzen window estimation, whereas this choice considerably simplifies (10):

$$\hat{R} = \underset{R}{\operatorname{argmin}} \sum_{i=1}^n \|f_i - NN_R(f_i)\|^2. \quad (12)$$

Solving (12) requires calculating the distance from the patch to all existing regions in the training dataset. This might be computationally prohibitive since usual training sets can

contain hundreds of images which translates in a number of regions in the order of thousands. Recent research in NBNN classifiers proposed local NBNN [15] which seizes this problem by only exploring the local neighborhood of each descriptor d_i . The runtime grows with the log of the number of categories rather than linearly as in [9], which results in sensitive speed-ups for large numbers of categories (results in [15] show a $\times 100$ speed-up for 256 categories) while still outperforming the original method [9] in classification accuracy.

Let R be some region and \bar{R} the set of all other regions. If we reformulate the NBNN updates as adjustments to the posterior log-odds, the alternative decision rule will be:

$$\hat{R} = \operatorname{argmax}_R \sum_{i=1}^n \log \frac{P(f_i | R)}{P(f_i | \bar{R})} + \log \frac{P(R)}{P(\bar{R})} \quad (13)$$

Again, the prior can be dropped if assumed uniform over R . The benefit of this alternative formulation as log-odds increments is that we can select the region posteriors which give a positive contribution on the sum in (13). The main contribution of local NBNN consists in (a) only using the closest member from the regions whose descriptors are within the k nearest neighbors of each f_i and (b) modeling the distance to the rest of the regions $P(f_i | \bar{R})$ as the distance to the $k + 1$ nearest neighbor.

Algorithm 1 ADAPTIVETRAINING(Y, R)

Require: A Nearest Neighbor index containing all descriptors from all regions, queried by $NN(d, \#neighbors)$.

Require: Region lookup function $REGION(descriptor)$ that retrieves the region to which $descriptor$ belongs to.

Require: Sampling patches function $SAMP_PATCHES(Region)$ which extracts patches with a certain overlap.

for all patches $y \in Y$ **do**

for all descriptors $d_i \in y$ **do**

$\{p_1, p_2, \dots, p_{k+1}\} \leftarrow NN(d_i, k+1)$

for all regions R found in the k nearest

neighbors **do**

$dist_R = \min_{\{p_j | REGION(p_j)\}} \|d_i - p_j\|^2$

end for

$totals[R] \leftarrow totals[R] + dist_R - dist_B$

end for

$Selected[y] \leftarrow \operatorname{argmin}_R totals[R]$

end for

for all Selected unique regions **do**

$T \leftarrow SAMP_PATCHES(Selected[R])$

end for

return T

After finding a region R for every patch y , we will sample patches of size L_p with a certain overlap inside the selected regions and include them in HR and LR training sets T_h and T_l , which will be used for training the sparse dictionaries and the sparse SR recovery as seen in Section 2. A summary including further implementation details can be found in Algorithm 1.

3.2 Rejecting regions

Some regions extracted from the training images might not be useful since they do not contain high frequency (e.g. blurry unfocused backgrounds, uniform colors). In order to reject these regions, we apply a high-pass filter whose frequency cut is related to the magnification factor MF intuitively requiring higher frequency content when a higher magnification factor is selected, according to $f_c = 1 - \frac{\beta}{MF}$, where β weights the impact of the second addend.

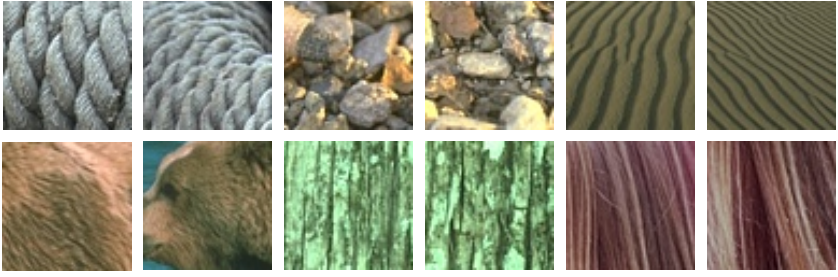


Figure 2: Appearance of 50x50 regions extracted from the training images. From left to right and top to bottom: rope, stones, sand, bear fur, tree bark, hair. Super-resolution performance can be improved by selecting a meaningful set of regions for every input image.

The energy per pixel is computed in the filtered region R' , defined as $E = \|R'\|_2^2 / L_R^2$. We reject a given region R when its energy E is lower than a given threshold ϵ . Some examples of selected regions are shown in Figure 2.

3.3 Feature space

Many feature representations have been proposed in the literature for boosting SR, according to the criteria that mid and high frequency are the most useful frequency bands when learning LR to HR patch correspondences. [6] used a simple high-pass filter in order to obtain the LR patch information closer in frequency to the one to be predicted in the HR patch. Later on, [16] used Gaussian derivative filters to also extract the high frequency information of the LR patches. In the same direction, [8] and [19] used concatenated first- and second-order gradients.

In this paper we use three different types of features for three different tasks.

1. *First- and second-order gradients of luminance* as a F_l feature transform for the patches in D_l .
2. *Centered luminance values*. In D_h we have the constraint for the descriptors to be easily reversible to pixel values. In this feature representation we subtract to every luminance value the mean of the patch's luminance (i.e. eliminating DC component from the patch). Although keeping the same descriptors as in D_l would be desirable to make a more consistent descriptor space, the reversion step is not feasible for gradient descriptors.
3. *Dense Scale Invariant Feature Transform (SIFT) descriptors* [13]. We use SIFT descriptors for our region selection stage. SIFT descriptors show improved resilience to changes in image scale and rotation, and they are robust to changes in illumination, noise and viewpoint. We use dense SIFT extraction instead of the original SIFT detector since we are dealing with small patches and we need to force a certain number of features per patch.

4 Results

In this section, we verify the performance of the proposed sparse method on image SR.

	Chang et al. [8]		Yang et al. [20]		Proposed method	
	2x	3x	2x	3x	2x	3x
<i>multiview</i>	38.060	34.195	40.772	36.089	41.276	36.487
	0.9667	0.9310	0.9774	0.9504	0.980	0.952
<i>Berkeley</i>	27.707	24.781	29.853	26.262	30.191	26.413
	0.8544	0.7430	0.9099	0.8121	0.9143	0.8149

Table 1: Average results from *multiview* and *Berkeley* testing scenarios. First row is PSNR in dB and second row is SSIM. Best results in bold.

4.1 Training and testing sets

In order to prove the effectivity of our adaptive training scheme, two different testing scenarios and a single training set have been used. Our training dataset comprises the training images provided in the Berkeley Segmentation dataset *BSD500*¹ composed by 200 generic images with a resolution of 380x420 and 60 images from the INRIA 4d repository² consisting in a capture of 4 different scenes (karate, stick, children, dog) from 15 different cameras. Our testing scenarios will be (a) 6 randomly selected generic images from the Berkeley testing dataset (referred as *Berkeley*) and (b) the 16th camera captures of the 4 different scenes plus 2 images from different time frames and cameras (referred as *multiview*). With these two testing sets we demonstrate that our method improves performance for generic images just by selecting optimal regions, but also that this method feature can effectively find the correlated regions within the training images when dealing with common scenarios as multiview cameras or video sequences.

For each testing image we set its full resolution version as ground truth, then we bicubically downscale it by the 2x and 3x magnification factors and finally apply the different SR algorithms to restore it to its original resolution, computing with the ground-truth image the luminance peak signal-to-noise-ratio (PSNR) and the Structural Similarity (SSIM) index, a quality measure based on the degradation of structural information [18]. In the case of the *multiview* testing images, we resize the original images to have a ground-truth resolution of 800x600, and we match the resolution of INRIA 4d repository training images to the one we are testing (i.e. different for each magnification factor) so that no training image from the INRIA 4d repository has higher resolution than the *multiview* images.

4.2 Comparisons with other methods

Our method is compared with two well-known SR methods: The original LLE-based method [8] and sparse coding SR method of [20]. The latter is the SR method which has been used as a baseline method in this paper, the direct comparison is therefore explicitly showing the improvement by using our algorithm.

When testing [8], the parameters and reduced training set recommended in their paper are used, adding an additional training image from the INRIA 4d repository in order to keep a fair ratio between generic images and correlated with *multiview* ones in the training set. For our proposed method, the square region size is fixed to $L_R = 50$ pixels, the patch size $L_p = 6$ pixels and the k -NN search of the local NBN to $k = 10$. For both our method and [20] the

¹<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>

²<http://4drepository.inrialpes.fr/pages/home>

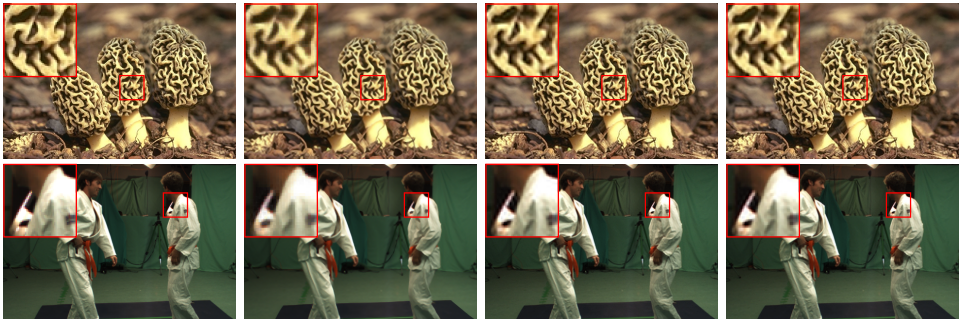


Figure 3: Reconstructed HR images (magnification factor 2x) from *Berkeley* (top) and *multiview* (bottom) testing sets. From left to right: Ground truth image, method in [9], method in [20] and proposed method. Better viewed when zoomed in.

same 260 training images have been used, selecting in both cases 100000 pair patches and compressing them into dictionaries D_h and D_l of 512 elements.

Table 1 summarizes the obtained results, reporting PSNR and SSIM average values for the different testing sets and magnification factors. Figure 3 offers visual results for one image of each testing set.

Comparing our method to [9], the visual improvements are easily noticeable: our method is clearly super-resolving finer details resulting in sharper images. The objective PSNR and SSIM results support this qualitative visual impression.

With respect to the baseline method [20], careful visual inspection shows that ringing artifacts along the edges are mostly suppressed and certain edges are sharper. In quantitative results, our method outperforms [20] in both PSNR and SSIM for the two magnification factors tested and in both testing set-ups. We observe that our method yields large improvement in performance when training and testing images are related (*multiview* test, 0.504 dB PSNR gain for 2x magnification factor), but there is also significant improvement for generic testing images (*Berkeley* test, 0.338 dB PSNR improvement for 2x magnification factor). We attribute the better performance to the fact that the image-adapted subsets T_h and T_l can represent the data more accurately, and that this adaptation is stronger whenever testing and some training image regions are correlated.

5 Conclusions and future work

This paper presents a novel sparse SR method, which focuses in adaptively selecting the optimal patches for the dictionary training. The method divides the training images into sub-image regions of sizes that preserve texture consistency, which are purged to reject those without high-frequency content. The best-representing region for each input LR patch is found through a Bayesian selection stage. In this selection process, SIFT descriptors are extracted densely from both input LR patches and regions and a local NBN approach is used in order to efficiently handle the high number of different regions in the training dataset. The resulting adapted subset of patches is compressed using sparse coding techniques and used to recover HR images by exploiting the sparsity prior. Experimental results show that our method improves performance for generic images, but also that this improvement is accentuated

ated when there exist training regions related to the testing images within the training dataset, therefore making our method specially adequate for applications where these conditions are met (e.g. video sequences, multiview scenarios, metadata-tagged images).

As for the future work, we believe that texture consistency can be further improved by using non-fixed shapes for the regions, allowing them to have a more flexible geometric model.

References

- [1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.
- [2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. pages 1–8, 2008. ISSN 1063-6919.
- [3] H. Chang, D. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 275–282, 2004.
- [4] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
- [5] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Trans. on Image Processing*, 13(10):1327–1344, 2004.
- [6] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int. J. Computer Vision*, 40(1):25–47, 2000.
- [7] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Comp. Graph. Appl.*, 22(2):56–65, 2002.
- [8] X. Gao, K. Zhang, D. Tao, and X. Li. Image super-resolution with sparse neighbor embedding. *IEEE Trans. on Image Processing*, 21(7):3194–3205, 2012. ISSN 1057-7149.
- [9] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graph. Models Image Processing*, 53(3):231–239, 1991.
- [10] R. Keys. Cubic convolution interpolation for digital image processing. *Acoustics, Speech and Signal Processing, IEEE Trans. on*, 29(6):1153–1160, 1981. ISSN 0096-3518.
- [11] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. *In NIPS*, pages 801–808, 2007.
- [12] Z. Lin and H.-Y. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):83–97, 2004.
- [13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. ISSN 0920-5691.

- [14] X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li. Geometry constrained sparse coding for single image super-resolution. *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1648–1655, 2012. ISSN 1063-6919.
- [15] S. McCann and D.G. Lowe. Local naive bayes nearest neighbor for image classification. *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 3650–3656, 2012. ISSN 1063-6919.
- [16] J. Sun, N. Zheng, H. Tao, and H. Shum. Image hallucination with primal sketch priors. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:II–729–36 vol.2, 2003. ISSN 1063-6919.
- [17] J. Sun, J. Sun, Z. Xu, and H. Shum. Image super-resolution using gradient profile prior. *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR.*, pages 1–8, 2008. ISSN 1063-6919.
- [18] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13(4): 600–612, 2004. ISSN 1057-7149.
- [19] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR.*, pages 1–8, 2008. ISSN 1063-6919.
- [20] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. on Image Processing*, 19(11):2861–2873, 2010.
- [21] K. Zhang, X. Gao, D. Tao, and X. Li. Multi-scale dictionary for single image super-resolution. *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1114–1121, 2012. ISSN 1063-6919.