Single Image Super-resolution from Transformed Self-Exemplars

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja
University of Illinois, Urbana-Champaign
{jbhuang1,asingh18,n-ahuja}@illinois.edu

Abstract

Self-similarity based super-resolution (SR) algorithms are able to produce visually pleasing results without extensive training on external databases. Such algorithms exploit the statistical prior that patches in a natural image tend to recur within and across scales of the same image. However, the internal dictionary obtained from the given image may not always be sufficiently expressive to cover the textural appearance variations in the scene. In this paper, we extend self-similarity based SR to overcome this drawback. We expand the internal patch search space by allowing geometric variations. We do so by explicitly localizing planes in the scene and using the detected perspective geometry to guide the patch search process. We also incorporate additional affine transformations to accommodate local shape variations. We propose a compositional model to simultaneously handle both types of transformations. We extensively evaluate the performance in both urban and natural scenes. Even without using any external training databases, we achieve significantly superior results on urban scenes, while maintaining comparable performance on natural scenes as other state-of-the-art SR algorithms.

1. Introduction

Most modern single image super-resolution (SR) methods rely on machine learning techniques. These methods focus on learning the relationship between low-resolution (LR) and high-resolution (HR) image patches. A popular class of such algorithms uses an external database of natural images as a source of LR-HR training patch pairs. Existing methods have employed various learning algorithms for learning this LR to HR mapping, including nearest neighbor approaches [14], manifold learning [6], dictionary learning [41], locally linear regression [38, 33, 34], and convolutional networks [9].

However, methods that learn LR-HR mapping from external databases have certain shortcomings. The number and type of training images required for satisfactory levels of performance are not clear. Large scale training sets are often required to learn a sufficiently expressive LR-HR dictionary. For every new scale factor by which the resolution has to be increased, or SR factor, these methods need to retrain the model using sophisticated learning algorithms on large external datasets.

Figure 1. Examples of self-similar patterns deformed due to local shape variation, orientation change, or perspective distortion.

To avoid using external databases and their associated problems, several approaches exploit internal patch redundancy for SR [10, 15, 13, 28]. These methods are based on the fractal nature of images [3], which suggests that patches of a natural image recur within and across scales of the same image. An internal LR-HR patch database can be built using the scale-space pyramid of the given image itself. Internal dictionaries have been shown to contain more *relevant* training patches, as compared to external dictionaries [44].

While internal statistics have been successfully exploited for SR, in most algorithms the LR-HR patch pairs are found by searching only for "translated" versions of patches in the scaled down images. This effectively assumes that an HR version of a patch appears in the same image at the desired scale, orientation and illumination. This amounts to assuming that the patch is planar and the images of the different assumed occurences of the patch are taken by a camera translating parallel to the plane of the patch. This fronto-parallel imaging assumption is often violated due to the non-planar shape of the patch surface, common in both natural and man-made scenes, as well as perspective distortion. Fig. 1 shows three examples of such violations, where self-similarity across scales will hold better if suitable geometric transformation of patches is allowed

In this paper, we propose a self-similarity driven SR algorithm that expands the internal patch search space. First, we explicitly incorporate the 3D scene geometry by localizing planes, and use the plane parameters to estimate the perspective deformation of recurring patches. Second, we expand the patch search space to include affine transfor-

mation to accommodate potential patch deformation due to local shape variations. We propose a compositional transformation model to simultaneously handle these two types of transformations. We modify the PatchMatch algorithm [1] to efficiently solve the nearest neighbor field estimation problem. We validate our algorithm through a large number of qualitative and quantitative comparisons against state-of-the-art SR algorithms on a variety of scenes. We achieve significantly better results for man-made scenes containing regular structures. For natural scenes, our results are comparable with current state-of-the-art algorithms.

Our Contributions:

1. Our method effectively increases the size of the limited internal dictionary by allowing geometric transformation of patches. We achieve state-of-the-art results without using any external training images.
2. We propose a decomposition of the geometric patch transformation model into (i) perspective distortion for handling structured scenes and (ii) additional affine transformation for modeling local shape deformation.
3. We use and make available a new dataset of urban images containing structured scenes as a benchmark for SR evaluation.

2. Related Work

The core of image SR algorithms has shifted from interpolation and reconstruction [22] to learning and searching for best matching existing image(s) as the HR map of the given LR image. We limit our discussion here to these more current learning-based approaches and classify the corresponding algorithms into two main categories: external and internal, depending on the source of training patches.

External database driven SR: These methods use a variety of learning algorithms to learn the LR-HR mapping from a large database of LR-HR image pairs. These include nearest neighbor [14], kernel ridge regression [23], sparse coding [41, 40, 42, 36], manifold learning [6] and convolutional neural networks [9]. The main challenges lie in how to effectively model the patch space. As opposed to learning a global mapping over the entire dataset, several methods alleviate the complexity of data modeling by partitioning or pre-clustering the external training database, so that relatively simpler prediction functions could be used for performing the LR-HR mapping in each training cluster [38, 33, 34]. Instead of learning in the 2D patch domain, some methods learn how 1D edge profiles transform across resolutions [30, 11]. Higher-level features have also been used in [16, 31, 32] for learning the LR-HR mapping. In contrast, our algorithm has the advantage of neither requiring external training databases, nor using sophisticated learning algorithms.

Internal database driven SR: Among internal database driven SR methods, Ebrahimi and Vrscay [10] combined ideas from fractal coding [3] with example-based algorithms such as non-local means filtering [5], to propose a self-similarity based SR algorithm. Glasner *et al*. [15] unified the classical and example-based SR by exploiting the patch recurrence within and across image scales. Freedman and Fattal [13] showed that self-similar patches can often be found in limited spatial neighborhoods, thereby gaining computational speed-up. Yang *et al*. [39] refined this notion further to seek self-similar patches in extermely localized neighborhoods (in-place examples), and performed first-order regression on them. Michaeli and Irani [26] used self-similarity to jointly recover the blur kernel and the HR image. Singh *et al*. [29] used the self-similarity principle for super-resolving noisy images.

Expanding patch search space: Since internal dictionaries are constructed using only the given LR image, they tend to contain a much smaller number of LR-HR patch pairs compared to external dictionaries which can be as large as desired. Singh and Ahuja used orientation selective sub-band energies for better matching textural patterns [27] and later reduced the self-similarity based SR into a set of problems of matching simpler sub-bands of the image, amounting to an exponential increase in the effective size of the internal dictionary [28]. Zhu *et al*. [43] proposed to enhance the expressiveness of the dictionary by optical flow based patch deformation during searching, to match the deformed patch with images in external databases. We use projective transformation to model the deformation common in urban scenes to better exploit internal self-similarity. Fernandez-Granda and Candes [12] super-resolved planar regions by factoring out perspective distortion and imposing group-sparse regularization over image gradients. Our method also incorporates 3D scene geometry for SR, but we can handle multiple planes and recover regular textural patterns beyond orthogonal edges through self-similarity matching. In addition, our method is a generic SR algorithm that handles both man-made and natural scenes in one framework. In the absense of any detected planar structures, our algorithm automatically falls back to searching only affine transformed self-exemplars for SR.

Our work is also related to several recent approaches that solve other low-level vision problems using over-parameterized (expanded) patch search spaces. Although more difficult to optimize than 2D translation, such over-parametrization often better utilizes the available patch samples by allowing transformations. Examples include stereo [4], depth upsampling [19], optical flow [18], image completion [21, 20], and patch-based synthesis [8]. Such expansion of the search space is particularly suited for the SR problem due to the limited size of internal dictionaries.

3. Overview

Super-resolution scheme: Given a LR image I, we first blur and subsample it to obtain its downsampled version $I_{\downarrow}$.

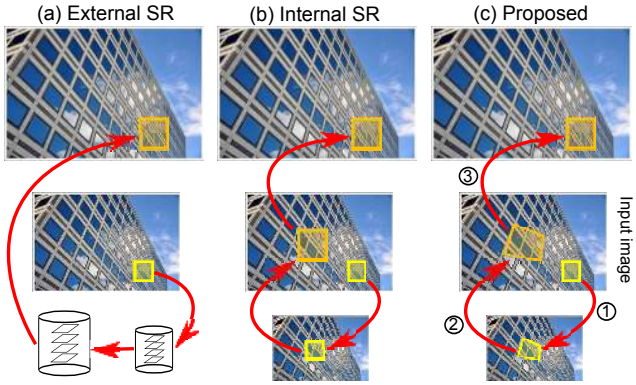


Figure 2. Comparison with external dictionary and internal dictionary (self-similarity) approaches. Middle row: Given LR image I. Our method allows for geometrically transforming the target patch from the input image, while searching for its nearest neighbor in the downsampled image. The HR version of the best match found is then pasted on to the HR image. This is repeated for all patches in the input image I.

Using I and I_{D}, our algorithm to obtain an HR image I_{H} consists of the following steps:

1) For each patch P (target patch) in the LR image I, we compute a transformation matrix $\mathbf{T}$ (homography) that warps P to its best matching patch Q (source patch) in the downsampled image I_{D}, as illustrated in Fig. 2 (c). To obtain the parameters of such a transformation, we estimate a nearest neighbor field between I and I_{D} using a modified PatchMatch algorithm [1] (details given in Section 4).

2) We then extract Q_{H} from the image I, which is the HR version of the source patch Q.

3) We use the inverse of the computed transformation matrix $\mathbf{T}$ to 'unwarp' the HR patch Q_{H}, to obtain the self-exemplar P_{H}, which is our estimated HR version of the target patch P. We paste P_{H} in the HR image I_{H} at the location corresponding to the LR patch P.

4) We repeat the above steps for all target patches to obtain an estimate of the HR image I_{H}.

5) We run the iterative backprojection algorithm [22] to ensure that the estimated I_{H} satisfies the reconstruction constraint with the given LR observation I.

Fig. 2 schematically illustrates the important steps in our algorithm, and compares it with other frameworks.

Motivation for using transformed self-exemplars: The key step in our algorithm is the use of the transformation matrix $\mathbf{T}$ that allows for geometric deformation of patches, instead of simply searching for the best patches under translation. We justify the use of transformed self-exemplars with two illustrative examples in Fig. 3. Matching using the affine transformation and and planar perspective transformation achieves both lower matching errors and more accurate prediction of the HR content than that from matching patches under translation.

4. Nearest Neighbor Field Estimation

4.1. Objective function

Let Ω be the set of pixel indices of the input LR image I. For each target patch $P(\mathbf{t}_i)$ centered at position $\mathbf{t}_i = (t_i^x, t_i^y)^\top$ in I, our goal is to estimate a transformation matrix $\mathbf{T}_i$ that maps the target patch $P(\mathbf{t}_i)$ to its nearest neighbor in the downsampled image I_{D}. A dense nearest neighbor patch search forms a nearest-neighbor field (NNF) estimation problem. In contrast to the conventional 2D translation (or offsets) field, here we have a field of *transformations* parametrized by θ_i for i^{th} pixel in the input LR image. Our objective function for this NNF estimation problem takes the form

$$\min_{\{\theta_i\}} \sum_{i \in \Omega} \mathbf{E}_{\mathrm{app}}(\mathbf{t}_i, \theta_i) + \mathbf{E}_{\mathrm{plane}}(\mathbf{t}_i, \theta_i) + \mathbf{E}_{\mathrm{scale}}(\mathbf{t}_i, \theta_i), \qquad (1)$$

where θ_i is the unknown set of parameters for constructing the transformation matrix $\mathbf{T}_i$ that we need to estimate (in a way explained later). Our objective function includes three costs: (1) appearance cost, (2) plane cost, and (3) scale cost. Below we first describe each of these costs.

Appearance cost $\mathbf{E}_{\mathrm{app}}$: This cost measures similarity between the sampled target and source patches. We use Gaussian-weighted sum-of-squared distance in the RGB space as our metric:

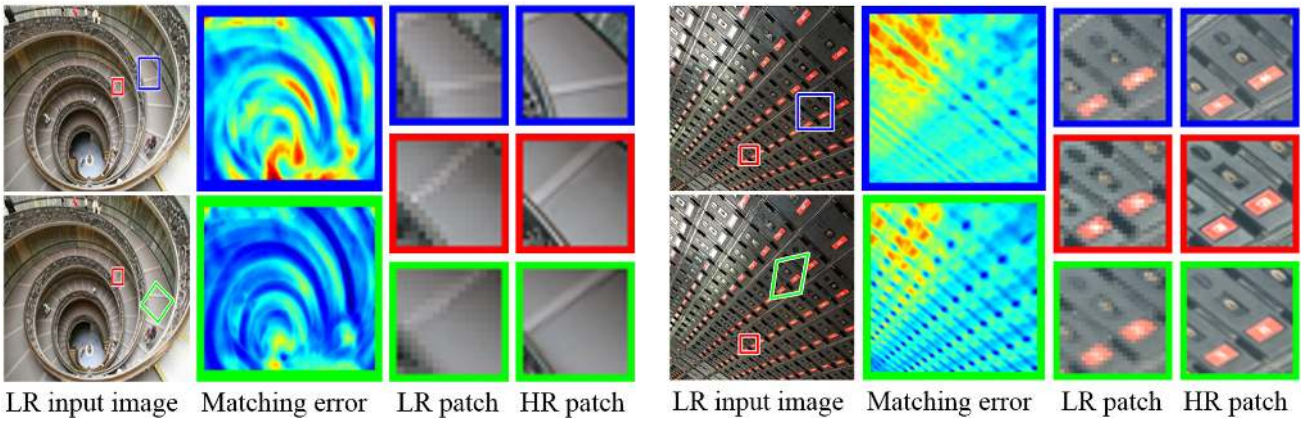
$$\mathbf{E}_{\mathrm{app}}(\mathbf{t}_i, \theta_i) = ||W_i\,(P(\mathbf{t}_i) - Q(\mathbf{t}_i, \theta_i))\,||_2^2, \qquad (2)$$

where the matrix W_i is the Gaussian weights with $\sigma^2 = 3$, $Q(\mathbf{t}_i, \theta_i)$ denotes the sampled patch from I_{D} using the transformation $\mathbf{T}_i$ with parameter θ_i.

We now present how we design and construct the transformation matrix $\mathbf{T}_i$ from estimated parameter θ_i for sampling the source patch $Q(\mathbf{t}_i, \theta_i)$. The geometric transformation of a patch in general can have up to 8 degrees of freedom (i.e., a projective transformation). One way to estimate the patch geometric transformation is to explicitly search in the additional patch space (e.g., scale, rotation) [2, 17, 8] beyond translation. However, perspective distortion can only be approximated by scaling, rotation and shearing of affine transformations. Therefore, affine transformations by themselves are less effective in modeling the appearance variations in man-made, structured scenes. Huang *et al*. [20] addressed this problem by detecting planes (and their parameters) and using them to determine the perspective transformation between the target and source patch. In Fig. 4, we show a visualization of vanishing point detection and posterior probability map for detection of planes, as yielded by [20].

In this paper, we combine the explicit search strategy of [2, 17, 8], along with the perspective deformation estimation approach of [20]. Using the algorithm of [20][1], we detect and localize planes and compute the planar parameters,

[1] Available at https://github.com/jbhuang0604/StructCompletion



(a) Affine transformation (b) Planar perspective transformation

Figure 3. Examples demonstrating the need for using transformed self-exemplars in our self-similarity based SR. Red boxes indicate a selected target patch (to be matched) in the input LR image I. We take the selected target patch, remove its mean, and find its nearest neighbor in the downsampled image I_{D}. We show the error found while matching patches in I_{D} in the second column. Blue boxes indicate the nearest neighbor (best matched) patch found among only translational patches, and green boxes indicate the nearest neighbor found under the proposed (a) affine transformation and (b) planar perspective transformation. In the third and fourth columns we show the matched patches Q in the downsampled images I_{D} and their HR version Q_{H} in the input image I.

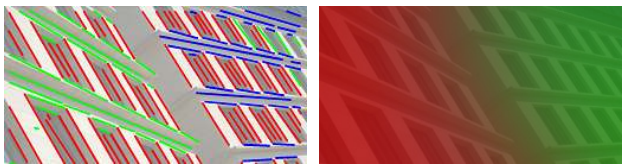


Figure 4. (a) Vanishing point detection. (b) Visualization of posterior plane probability.

as shown by the example in Fig. 4. We propose to parameterize $\mathbf{T}_i$ by $\theta_i = (\mathbf{s}_i, m_i)$, where $\mathbf{s}_i = (\mathbf{s}_i^x, \mathbf{s}_i^y, \mathbf{s}_i^s, \mathbf{s}_i^\theta, \mathbf{s}_i^\alpha, \mathbf{s}_i^\beta)$ is the 6-D affine motion parameter of the source patch and m_i is the index of detected plane (using [20]). We propose a factored geometric transformation model $\mathbf{T}_i(\theta_i)$ of the form:

$$\mathbf{T}_i(\theta_i) = \mathbf{H}\left(\mathbf{t}_i, \mathbf{s}_i^x, \mathbf{s}_i^y, m_i\right) \mathbf{S}\left(\mathbf{s}_i^s, \mathbf{s}_i^\theta\right) \mathbf{A}\left(\mathbf{s}_i^\alpha, \mathbf{s}_i^\beta\right), \quad (3)$$

where the matrix $\mathbf{H}$ captures the perspective deformation given the target and source patch positions and the planar parameters (as described in [20]). The matrix

$$\mathbf{S}\left(\mathbf{s}_i^s, \mathbf{s}_i^\theta\right) = \begin{bmatrix} \mathbf{s}_i^s \mathbf{R}(\mathbf{s}_i^\theta) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (4)$$

captures the similarity transformation through a scaling parameter $\mathbf{s}_i^s$ and a 2×2 rotation matrix $\mathbf{R}(\mathbf{s}_i^\theta)$, and the matrix

$$\mathbf{A}\left(\mathbf{s}_i^\alpha, \mathbf{s}_i^\beta\right) = \begin{bmatrix} 1 & \mathbf{s}_i^\alpha & 0 \\ \mathbf{s}_i^\beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

captures the shearing mapping in the affine transformation.

The proposed compositional transformation model resembles the classical decomposition of a projective transformation matrix into a concatenation of three unique matrices: similarity, affine, and pure perspective transformation [24]. Yet, our goal here is to "synthesize," rather than"analyze" the transformation $\mathbf{T}_i$ for sampling source patches. The proposed formulation allows us to effectively factor out the dependency of the positions of the target $\mathbf{t}_i$ and source patch $(\mathbf{s}_i^x, \mathbf{s}_i^y)$ for estimating the perspective deformation in $\mathbf{H}\left(\mathbf{t}_i, \mathbf{s}_i^x, \mathbf{s}_i^y, m_i\right)$ from estimating affine shape deformation parameters using $(\mathbf{s}_i^s, \mathbf{s}_i^\theta, \mathbf{s}_i^\alpha, \mathbf{s}_i^\beta)$ for matrices $\mathbf{S}$ and $\mathbf{A}$. This is crucial because we can then exploit piecewise smoothness characteristics in natural images for efficient nearest neighbor field estimation.

Plane compatibility cost $\mathbf{E}_{\mathbf{plane}}$: For man-made images, we can often reliably localize planes in the scene using standard vanishing point detection techniques. The detected 3D scene geometry can be used to guide the patch search space. We modify the plane localization code in [20] and add a plane compatibility cost to encourage the search over the more probable plane labels for source and target patches.

$$\mathbf{E}_{\mathrm{plane}} = -\lambda_{\mathrm{plane}} \log\left(\Pr[m_i|(\mathbf{s}_i^x, \mathbf{s}_i^y)] \times \Pr[m_i|(\mathbf{t}_i^x, \mathbf{t}_i^y)]\right), \quad (6)$$

where the $\Pr[m_i|(x,y)]$ is the posterior probability of assigning label m_i at pixel position (x,y) (see Fig 4 (b) for an example).

Scale cost $\mathbf{E}_{\mathbf{scale}}$: Since we allow continuous geometric transformations, we observed that the nearest neighbor field often converged to the trivial solution, i.e., matching target patches to itself in the downsampled image I_{D}. Such a match has small appearance cost. This trivial solution leads to the conventional bicubic interpolation for SR. We avoid such trivial solutions by introducing the scale cost $\mathbf{E}_{\mathrm{scale}}$:

$$\mathbf{E}_{\mathrm{scale}} = \lambda_{\mathrm{scale}} \min(0, \mathrm{SRF} - \mathrm{Scale}(\mathbf{T}_i)), \quad (7)$$

where SRF indicates the desired SR factor, e.g., 2x, 3x, or 4x, and the function $\mathrm{Scale}(\cdot)$ indicates the scale estimation of a projective transformation matrix. We approximately

estimate the scale of the source patch sampled using $\mathbf{T}_i$ with the first-order Taylor expansion [7]:

$$\text{Scale}(\mathbf{T}_i) = \sqrt{\det\left(\begin{bmatrix} \mathbf{T}_{1,1} - \mathbf{T}_{1,3}\mathbf{T}_{3,1} & \mathbf{T}_{1,2} - \mathbf{T}_{1,3}\mathbf{T}_{3,2} \\ \mathbf{T}_{2,1} - \mathbf{T}_{2,3}\mathbf{T}_{3,1} & \mathbf{T}_{3,1} - \mathbf{T}_{2,3}\mathbf{T}_{3,2} \end{bmatrix}\right)},$$

where $\mathbf{T}_{u,v}$ indicates the value of u^{th} row and v^{th} column in the transformation matrix $\mathbf{T}_i$ with $\mathbf{T}_{3,3}$ normalized to one. Intuitively, we penalize if the scale of the source patches is too small. Therefore, we encourage the algorithm to search for source patches that are similar to the target patch and at the same time to have larger scale in the input LR image space; and therefore we are able to provide more high-frequency details for SR. We soft-threshold the penalty to zero when the scale of the source patch is sufficiently large.

4.2. Inference

We need to estimate 7-dimensional ($\theta_i \in R^7$) nearest neighbor field solutions over all overlapping target patches. Unlike the conventional self-exemplar based methods [15, 13], where only a 2D translation field needs to be estimated, the solution space in our formulation is much more difficult to search. We modify the PatchMatch [1] algorithm for this task with the following detailed steps.

Initialization: Instead of the random initialization done in PatchMatch [1], We initialize the nearest neighbor field with zero displacements and scales equal to the desired SR factor. This is inspired by [13, 39], suggesting that good self-exemplars can often be found in a localized neighborhood. We found that this initialization strategy provides a good start for faster convergence.

Propagation: This step efficiently propagates good matches to neighbors. In contrast to propagating the transformation matrix $\mathbf{T}_i$ directly, we propagate the parameter $\theta_i = (\mathbf{s}_i, m_i)$ instead so that the affine shape transformation is invariant to the source patch position.

Randomization: After propagation in each iteration, we perform randomized search to refine the current solution. We simultaneously draw random samples of the plane index based on the posterior probability distribution, randomly perturb the affine transformation and randomly sample position (in a coarse-to-fine manner) to search for the optimal geometric transformation of source patches and reduce the matching errors.

5. Experiments

Datasets: Yang *et al*. [37] recently proposed a benchmark for evaluating single image SR methods. Most images therein consist of natural scenes such as landscapes, animals, and faces. Images that contain indoor, urban, architectural scenes, etc., rarely appear in this benchmark. However, such images feature prominently in consumer photographs. We therefore have created a new dataset *Urban 100* containing 100 HR images with a variety of real-world structures. We constructed this dataset using images from Flickr (under CC license) using keywords such as urban, city, architecture, and structure.

In addition, we also evaluate our algorithm on the *BSD 100* dataset, which consists of 100 test images of natural scenes taken from the Berkeley segmentation dataset [25]. For this dataset, we evaluate for SR factors of 2x, 3x, and 4x.

Methods evaluated: We compare our results against several state-of-the-art SR algorithms. Specifically, we choose four SR algorithms trained using a large number of external LR-HR patches for training. The algorithms we use are: Kernel rigid regression (Kim) [23], sparse coding (ScSR) [41], adjusted anchored neighbor neighbor regression (A+) [34], and convolutional neural networks (SRCNN) [9].[2] We also compare our results with those of the internal dictionary based approach (Glasner) [15] [3] and the sub-band self-similarity SR algorithm (Sub-Band) [28].[4] All our datasets, results, and the source code will be made publicly available.

Implementation details: We use 5×5 patches and perform SR in multiple steps. We achieve 2x, 3x, 4x SR factors in three, five and six upscaling steps, respectively. At the end of each step, we run 20 iterations of the backprojection algorithm [22] with a 5×5 Gaussian filter with $\sigma^2 = 1.2$. The NNF solution from a coarse level is upsampled and used as an initialization for the next finer level. We empirically set the parameters $\lambda_{\text{plane}} = 10^{-3}$ and $\lambda_{\text{scale}} = 10^{-3}$. The parameters are kept fixed for all our experiments.

Qualitative evaluation: In Figure 5, we show visual results on images from the Urban 100 dataset. We show only the cropped regions here. Full image results are available in the supplementary material. We find that our method is capable of recovering structured details that were missing in the LR image by properly exploiting the internal similarity in the LR input. Other approaches, using external images for training, often fail to recover these structured details. Our algorithm well exploits the detected 3D scene geometry and the internal natural image statistics to super-resolve the missing high-frequency contents. In Fig. 6 and 7, we demonstrate that our algorithm is not restricted to images of a single plane scene. We are able to automatically search for multiple planes and estimate their perspective and affine transformations to robustly predict the HR image.

In Fig. 8 and 9, we show two results on natural images where no regular structures can be detected. In such cases, our algorithm reduces to searching for affine transformations only in the nearest neighbor field, similar to [2]. On natural images without any particular geometric regularity, our method performs as well as the recent, state-of-the-art methods such as [9, 34], although, as can be seen in both examples, our results contain slightly sharper edges and fewer

[2] Implementations of [23, 41, 34, 9] are available on authors' websites.
[3] We implement this from the paper [15].
[4] Results were provided by the authors.

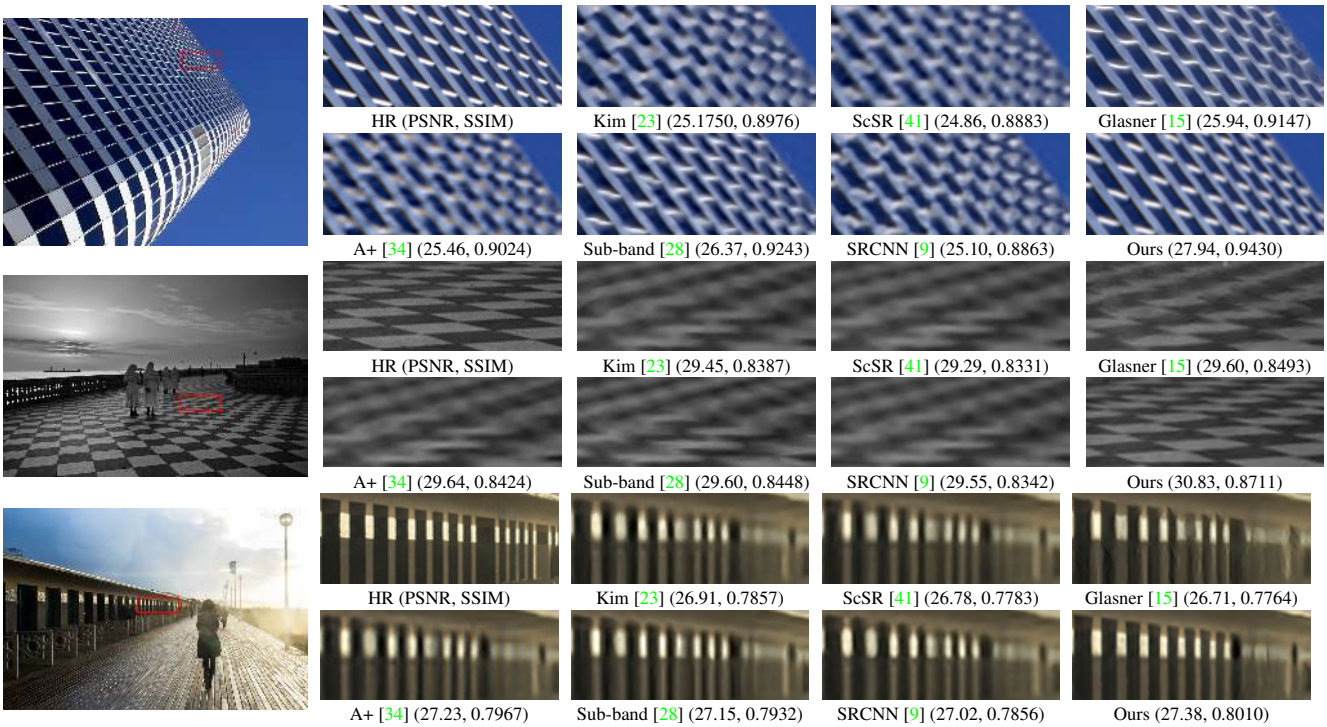


Figure 5. Visual comparison for 4x SR. Our method is able to explicitly identify perspective geometry to better super-resolve details of regular structures occuring in various urban scenes. Full images are provided in supplementary material.

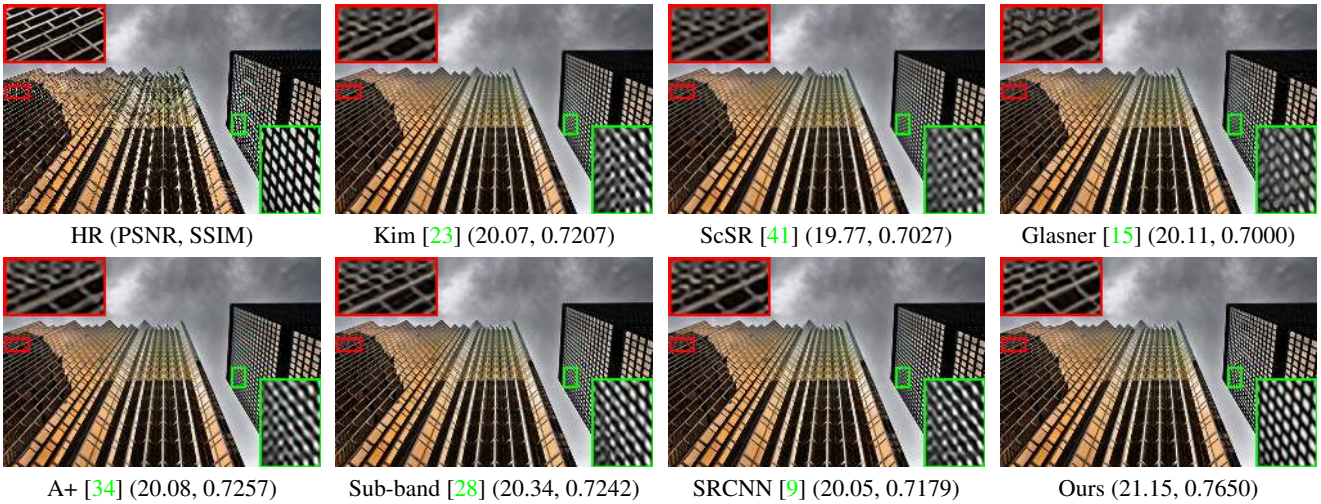
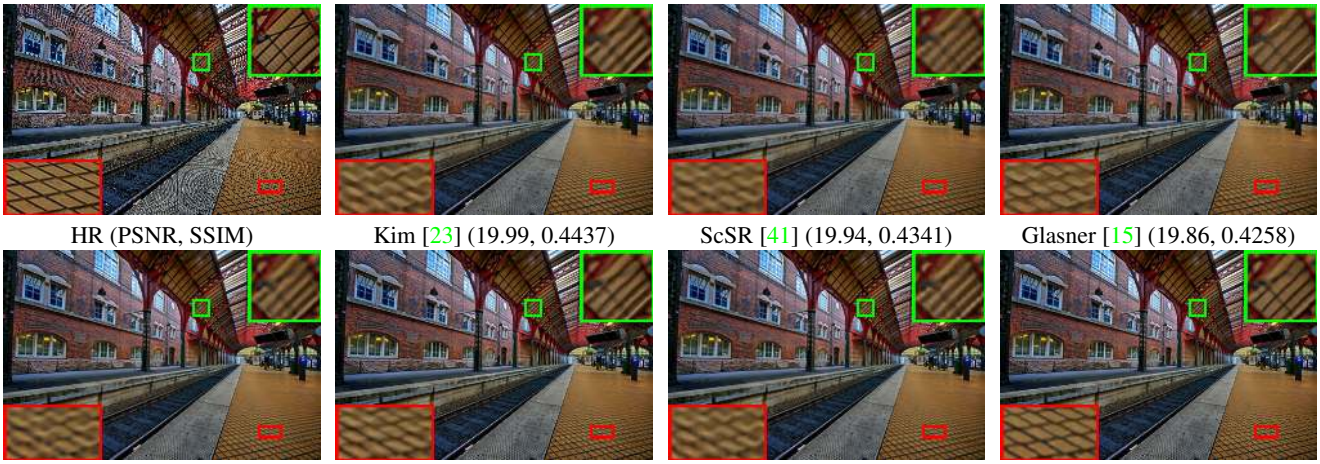


Figure 6. Visual comparison for 4x SR. Our algorithm is able to super-resolve images containing multiple planar structures.

artifacts such as ringing. We present more results for both Urban 100 and BSD 100 datasets in the supplementary material.

Quantitative evaluation: We also perform quantitative evaluation of our method in terms of PSNR (dB) and structural similarity (SSIM) index [35] (computed using luminance channel only). Since such quantitative metrics may not correlate well with visual perception, we invite the reader to examine the visual quality of our results for better evaluation of our method.

Table 1 shows the quantitative results on *Urban 100* and *BSD 100* dataset. Numbers in red indicate the best performance and those in blue indicate the second best performance. Our algorithm yields the best quantitative results for this dataset, 0.2-0.3 dB PSNR better than the second best method (A+) [34] and 0.4-0.5 dB better than the recently proposed SRCNN [9]. We are able to achieve these results *without* any training databases, while both [34] and [9] require millions of external training patches. Our method also outperforms the self-similarity approaches of [15] and [28], validating our claim of being able to extract better internal statistics through the expanded internal search space. In



HR (PSNR, SSIM) | Kim [23] (19.99, 0.4437) | ScSR [41] (19.94, 0.4341) | Glasner [15] (19.86, 0.4258)

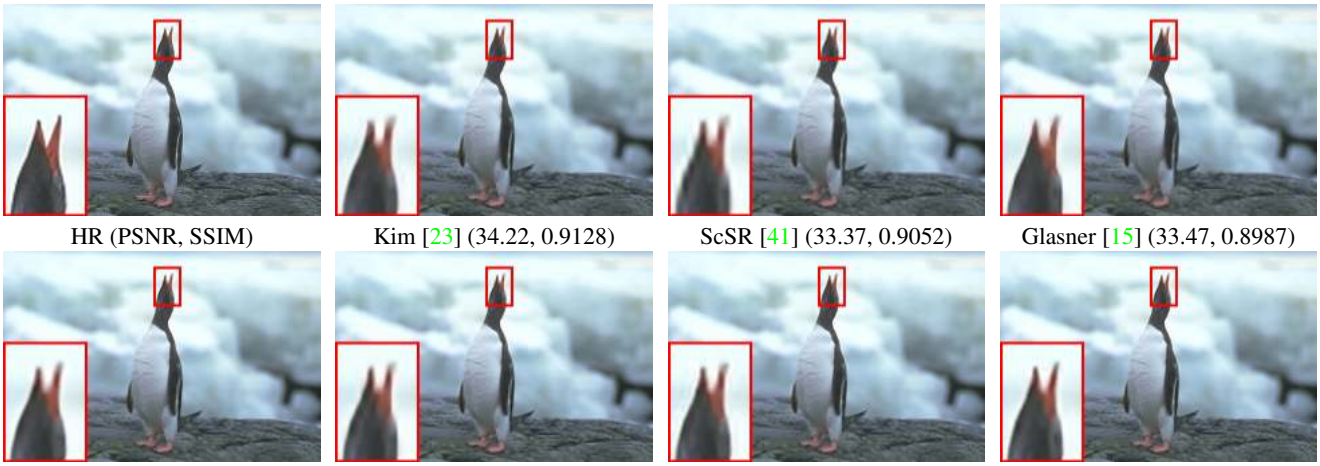
A+ [34] (20.03, 0.4523) | Sub-band [28] (20.00, 0.4573) | SRCNN [9] (19.98, 0.4386) | Ours (20.09, 0.4690)

Figure 7. Visual comparison for 4x SR. Our algorithm is able to better exploit the regularity present in urban scenes than other methods. .

HR (PSNR, SSIM) | Kim [23] (27.49, 0.6948) | ScSR [41] (27.42, 0.6908) | Glasner [15] (27.20, 0.6825)

A+ [34] (27.62, 0.7007) | Sub-band [28] (27.33, 0.6916) | SRCNN [9] (27.52, 0.6938) | Ours (27.60, 0.6966)

Figure 8. Visual comparison for 3x SR. Our result produces sharper edges than other methods. Shapes of fine structures (such as the horse's ears) are reproduced more faithfully in our result.



HR (PSNR, SSIM) | Kim [23] (34.22, 0.9128) | ScSR [41] (33.37, 0.9052) | Glasner [15] (33.47, 0.8987)

A+ [34] (34.27, 0.9144) | Sub-band [28] (33.55, 0.9053) | SRCNN [9] (34.18, 0.9107) | Ours (34.12, 0.9091)

Figure 9. Visual comparison for 3x SR. Our result shows slightly sharper reconstruction of the beaks.

BSD 100 dataset our results are comparable to those obtained by other approaches on this dataset, with ≈ 0.1 dB lower PSNR than the results of A+ [34]. Our quantitative results are slightly worse than the state-of-the-art in this dataset since it is difficult to find geometric regularity in such natural images, which our algorithm seeks to exploit. Also A+ [34] is trained on patches that contain natural textures quite suitable for super-resolving the BSD100 images.

Table 1. Quantitative evaluation on *Urban 100* and *BSD 100* datasets. Red indicates the best and blue indicates the second best performance.

Metric	Scale	Bicubic	ScSR [41]	Kim [23]	SRCNN [9]	A+ [34]	Sub-band [28]	Glasner [15]	Ours
PSNR (*Urban*)	2x	26.66	28.26	28.74	28.65	28.87	28.34	28.15	29.05
	4x	23.14	24.02	24.20	24.14	24.34	24.21	23.79	24.67
SSIM (*Urban*)	2x	0.8408	0.8828	0.8940	0.8909	0.8957	0.8820	0.8743	0.8980
	4x	0.6573	0.7024	0.7104	0.7047	0.7195	0.7115	0.6838	0.7314
PSNR (*BSD*)	2x	29.55	30.77	31.11	31.11	31.22	30.73	30.56	31.12
	3x	27.20	27.72	28.17	28.20	28.30	27.88	27.36	28.20
	4x	25.96	26.61	26.71	26.70	26.82	26.60	26.38	26.80
SSIM (*BSD*)	2x	0.8425	0.8744	0.8840	0.8835	0.8862	0.8774	0.8675	0.8835
	3x	0.7382	0.7647	0.7788	0.7794	0.7836	0.7714	0.7490	0.7778
	4x	0.6672	0.6983	0.7027	0.7018	0.7089	0.7021	0.6842	0.7064

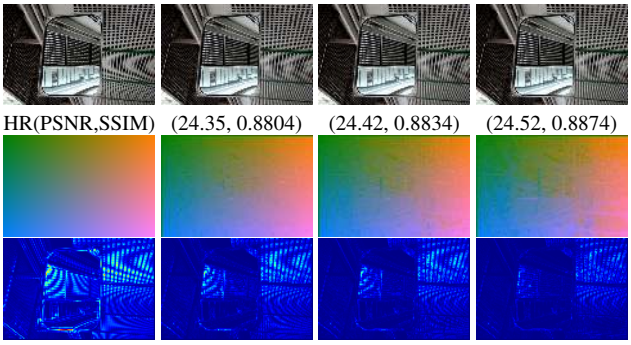


Figure 10. Effect of iterations. First row: HR and the SR results on 1, 2, and 5 iterations. Second row: the visualization of the nearest neighbor field. Third row: the patch matching cost.

While we achieve slightly worse quantitative performance on BSD100, our results are often visually more pleasing than others and do not have artifacts.

Convergence of NNF estimation: We investigate the effect of the number of iterations for nearest neighbor field estimation using our algorithm in Fig. 10, for one step 2x SR. We show the intermediate results after 1, 2, and 5 iterations. The second row shows a visualization of the source patch positions in the nearest neighbor field and the matching cost in each stage. The in-place initialization (zero iterations) already provides good matches for smooth regions. We can see a significant reduction in the matching cost even with one iteration. We use 10 iterations for generating all our results.

Effect of patch size: Patch size is an important parameter for example-based SR algorithms. Larger patches may be difficult to map to HR since they may contain complex structural details. Very small patches may not contain sufficient information to accurately predict their HR versions. In Fig. 11, we plot PSNR/SSIM for patch sizes ranging from 3×3 to 15×15. We obtain these plots by averaging over 25 images. We observe that there is a wide range of patch sizes for which our algorithm is able to perform consistently.

Limitations: Our method has difficulty dealing with fine details when the planes are not accurately detected. We show one such case in Fig. 12 where we fail to recover the regular structures. Another limitation of our approach is

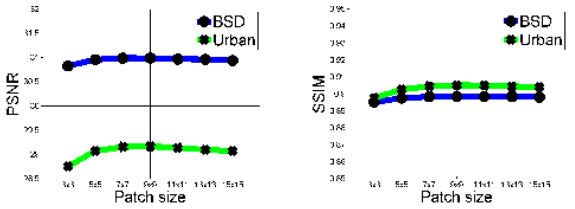


Figure 11. Quantitative performance as a function of patch size.

Figure 12. A failure case with SR factor 4x.

processing time. While external database driven SR methods require time-consuming training procedures, they run quite fast during test time [34, 33]. While our algorithm does not require an explicit training step, it is slow to super-resolve a test image. This drawback is associated with all self-similarity based approaches [15, 28]. On average, our Matlab implementation takes around 40 seconds to super-resolve an image in *BSD 100* by 2x with a 2.8 GHz Intel i7 CPU and 12 GB memory.

6. Concluding Remarks

We have presented a self-similarity based image SR algorithm that uses transformed self-exemplars. Our algorithm uses a factored patch transformation representation for simultaneously accounting for both planar perspective distortion and affine shape deformation of image patches. We exploit the 3D scene geometry and patch search space expansion for improving the self-examplar search. In the absense of regular structures, our algorithm reverts to searching affine transformed patches. We have demonstrated that even without using external training samples, our method outperforms state-of-the-art SR algorithms on a variety of man-made scenes while maintaining comparable performance on natural scenes.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. on Graphics*, 28(3):24, 2009. 2, 3, 5

[2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *ECCV*, 2010. 3, 5

[3] M. Barnsley. *Fractals Everywhere*. Academic Press Professional, Inc., 1988. 1, 2

[4] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, 2011. 2

[5] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 2

[6] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004. 1, 2

[7] O. Chum and J. Matas. Planar affine rectification from change of scale. In *ACCV*, 2010. 5

[8] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: combining inconsistent images using patch-based synthesis. *ACM Trans. on Graphics*, 31(4):82, 2012. 2, 3

[9] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 2, 5, 6, 7, 8

[10] M. Ebrahimi and E. R. Vrscay. Solving the inverse problem of image zooming using self-examples. In *Image analysis and Recognition*, 2007. 1, 2

[11] R. Fattal. Image upsampling via imposed edge statistics. *ACM Trans. on Graphics*, 26(3):95, 2007. 2

[12] C. Fernandez-Granda and E. J. Candes. Super-resolution via transform-invariant group-sparse regularization. In *ICCV*, 2013. 2

[13] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Trans. on Graphics*, 30(2):12, 2011. 1, 2, 5

[14] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002. 1, 2

[15] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 1, 2, 5, 6, 7, 8

[16] Y. HaCohen, R. Fattal, and D. Lischinski. Image upsampling via texture hallucination. In *ICCP*, 2010. 2

[17] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *ACM Trans. on Graphics*, volume 30, page 70, 2011. 3

[18] M. Hornáček, F. Besse, J. Kautz, A. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using patchmatch belief propagation. In *ECCV*, 2014. 2

[19] M. Hornáček, C. Rhemann, M. Gelautz, and C. Rother. Depth super resolution by rigid body self-similarity in 3d. In *CVPR*, 2013. 2

[20] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. Image completion using planar structure guidance. *ACM Trans. on Graphics*, 33(4):129, 2014. 2, 3, 4

[21] J.-B. Huang, J. Kopf, N. Ahuja, and S. B. Kang. Transformation guided image completion. In *IEEE International Conference on Computational Photography*, 2013. 2

[22] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991. 2, 3, 5

[23] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE TPAMI*, 32(6):1127–1133, 2010. 2, 5, 6, 7, 8

[24] J. J. Koenderink, A. J. Van Doorn, et al. Affine structure from motion. *JOSA A*, 8(2):377–385, 1991. 4

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 5

[26] T. Michaeli and M. Irani. Nonparametric blind super-resolution. In *ICCV*, 2013. 2

[27] A. Singh and N. Ahuja. Sub-band energy constraints for self-similarity based super-resolution. In *ICPR*, 2014. 2

[28] A. Singh and N. Ahuja. Super-resolution using sub-band self-similarity. In *ACCV*, 2014. 1, 2, 5, 6, 7, 8

[29] A. Singh, F. Porikli, and N. Ahuja. Super-resolving noisy images. In *CVPR*, 2014. 2

[30] J. Sun, Z. Xu, and H.-Y. Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *IEEE TIP*, 20(6):1529–1542, 2011. 2

[31] J. Sun, J. Zhu, and M. Tappen. Context-constrained hallucination for image super-resolution. In *CVPR*, 2010. 2

[32] L. Sun and J. Hays. Super-resolution from internet-scale scene matching. In *ICCP*, 2012. 2

[33] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, 2013. 1, 2, 8

[34] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014. 1, 2, 5, 6, 7, 8

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[36] C.-Y. Yang, J.-B. Huang, and M.-H. Yang. Exploiting self-similarities for single frame super-resolution. In *ACCV*. 2010. 2

[37] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-image super-resolution: A benchmark. In *ECCV*, 2014. 5

[38] C.-Y. Yang and M.-H. Yang. Fast direct super-resolution by simple functions. In *ICCV*, 2013. 1, 2

[39] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *CVPR*, 2013. 2, 5

[40] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE TIP*, 21(8):3467–3478, 2012. 2

[41] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE TIP*, 19(11):2861–2873, 2010. 1, 2, 5, 6, 7, 8

[42] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, 2012. 2

[43] Y. Zhu, Y. Zhang, and A. L. Yuille. Single image super-resolution using deformable patches. In *CVPR*, 2014. 2

[44] M. Zontak and M. Irani. Internal statistics of a single natural image. In *CVPR*, 2011. 1