Fast Image Super-resolution Based on In-place Example Regression

Jianchao Yang, Zhe Lin, Scott Cohen
Adobe Research
345 Park Avenue, San Jose, CA 95110
{jiayang, zlin, scohen}@adobe.com

Abstract

We propose a fast regression model for practical single image super-resolution based on in-place examples, by leveraging two fundamental super-resolution approaches—learning from an external database and learning from self-examples. Our in-place self-similarity refines the recently proposed local self-similarity by proving that a patch in the upper scale image have good matches around its origin location in the lower scale image. Based on the in-place examples, a first-order approximation of the nonlinear mapping function from low- to high-resolution image patches is learned. Extensive experiments on benchmark and real-world images demonstrate that our algorithm can produce natural-looking results with sharp edges and preserved fine details, while the current state-of-the-art algorithms are prone to visual artifacts. Furthermore, our model can easily extend to deal with noise by combining the regression results on multiple in-place examples for robust estimation. The algorithm runs fast and is particularly useful for practical applications, where the input images typically contain diverse textures and they are potentially contaminated by noise or compression artifacts.

1. Introduction

Single image super-resolution aims at generating a high-resolution image from one low-resolution input. The problem is dramatically under-constrained, and thus it has to rely on some strong image priors for robust estimation. Such image priors range from simple analytical "smoothness" priors to more sophisticated statistical priors learned from natural images [16, 5, 7, 19, 10].

For image upscaling, the most popular methods are those based on analytical interpolations, e.g., bicubic interpolation, with the image "smoothness" assumption. As images contain strong discontinuities, such as edges and corners, the simple "smoothness" prior will result in ringing, jaggy and blurring artifacts. Therefore, more sophisticated statistical priors learned from natural images are explored [5, 3, 13, 15]. However, even though natural images are sparse signals, trying to capture their rich characteristics with only a few parameters is impossible. Alternatively, example-based nonparametric methods [7, 14, 2, 17] were used to predict the missing frequency band of the upsampled image, by relating the image pixels at two spatial scales using a universal set of training example patch pairs. Typically, a huge set of training patches are needed, resulting in excessively heavy computation cost. Taking one step further, Yang et al. [19, 18] proposed to use sparse linear combinations to recover the missing high-frequency band, allowing a much more compact dictionary model based on sparse coding.

Some recent studies show that images generally possess a great amount of self-similarities, i.e., local image structures tend to recur within and across different image scales [4, 1, 21], and image super-resolution can be regularized based on these self-similar examples instead of some external database [4, 9, 6]. In particular, Glasner et al. [9] use self-examples within and across multiple image scales to regularize the otherwise ill-posed classical super-resolution scheme. Freedman and Fattal [6] extend the example-based super-resolution framework with self-examples and iteratively upscale the image. They show that the local self-similarity assumption for natural images holds better for small upscaling factors and the patch search can be conducted in a restricted local region, allowing a very fast practical implementation.

In this paper, we refine the local self-similarity [6, 21] by *in-place self-similarity*, by proving that, for a query patch in the upper scale image, patch matching can be restricted to its origin location in the lower scale image. Based on these in-place examples, we learn a robust first-order approximation of the nonlinear mapping function from low- to high-resolution image patches. Compared with the state-of-the-art methods [9, 18, 6], our algorithm runs very fast and can produce more natural structures, thereby it is better at handling real applications, where the input images contain complex structures and diverse textures and they are potentially contaminated by sensor noise (e.g, cellphone cameras) or compression artifacts (e.g, internet images). In summary, this paper makes the following contributions.

1. We propose a new fast super-resolution algorithm based on regression on in-place examples, which, for the first time, leverages the two fundamental super-resolution approaches of learning from external-examples and learning from self-examples.
2. We prove that patch matching across different image scales with small scaling factors is in-place, which refines and validates the recently proposed local self-similarity theoretically.
3. We can easily extend our algorithm to handle noisy input images by combining regression results on multiple in-place examples. The algorithm runs very fast and is particularly useful for real applications.

The remainder of the paper is organized as follows. Section 2 introduces the notations we use. Section 3 presents our robust regression model for super-resolution. Section 4 presents our algorithm implementation details and results on both synthetic and real-world images. Finally, Section 5 concludes our paper.

2. Preliminaries and Notations

This work focuses on upscaling an input image X_0 which contains some high-frequency content that we can borrow for image super-resolution, i.e., X_0 is a sharp image but with unsatisfactory pixel resolution.[1] In the following, we use X_0 and X to denote the input and output (by $s\times$) images, and Y_0 and Y their low-frequency bands, respectively. That is, Y_0 (Y) has the same spatial dimension as X_0 (X), but is missing the high-frequency content. We use bolded lower case $\boldsymbol{x}_0$ and $\boldsymbol{x}$ to denote $a \times a$ image patches sampled from X_0 and X, respectively, and $\boldsymbol{y}_0$ and $\boldsymbol{y}$ to denote $a \times a$ image patches sampled from Y_0 and Y, respectively. $\boldsymbol{y}_0$ and $\boldsymbol{y}$ are thus referred as low-resolution image patches because they are missing high-frequency components, while $\boldsymbol{x}_0$ and $\boldsymbol{x}$ are referred as their high-resolution counterparts. Plain lower case (x, y) and (p, q) denote coordinates in the 2D image plane.

3. Regression on In-place Examples

In this section, we present our super-resolution algorithm based on learning the in-place example regression by referring to an external database.

3.1. The Image Super-resolution Scheme

Similar to [6], we first describe our overall image upscaling scheme in Figure 1, which is based on *in-place examples* and *first-order regression* that will be discussed shortly.

[1]This is the implicit assumption of most previous example-based approaches.

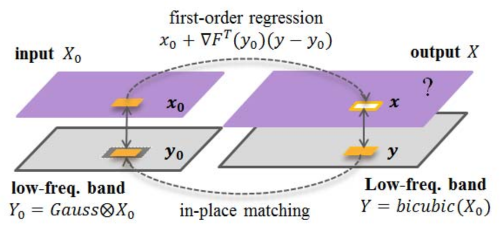

Figure 1. Image upscaling with first-order regression on in-place examples. For each patch $\boldsymbol{y}$ of the upsampled low-frequency band image Y, we find its in-place match $\boldsymbol{y}_0$ from the low-frequency band Y_0, and then perform a first-order regression on $\boldsymbol{x}_0$ to estimate the desired patch $\boldsymbol{x}$ for target X.

The input image is denoted as $X_0 \in \mathbb{R}^{K_1 \times K_2}$, from which we obtain its low-frequency band image $Y_0 \in \mathbb{R}^{K_1 \times K_2}$ by a low-pass Gaussian filtering. We upsample X_0 using bicubic interpolation by a factor of s to get $Y \in \mathbb{R}^{sK_1 \times sK_2}$. We use Y to approximate the low-frequency band of the unknown high-resolution image $X \in \mathbb{R}^{sK_1 \times sK_2}$. From X_0, Y_0, and Y, we aim to estimate the high-resolution image X.

For each image patch $\boldsymbol{y}$ ($a \times a$) from the image Y at location (x, y), we find its *in-place example* $\boldsymbol{y}_0$ ($a \times a$) around its origin coordinates (x_r, y_r) in image Y_0, where $x_r = \lfloor x/s + 0.5 \rfloor$ and $y_r = \lfloor y/s + 0.5 \rfloor$. Correspondingly, we can obtain the image patch $\boldsymbol{x}_0$ ($a \times a$) from X_0, which is a high-resolution version of $\boldsymbol{y}_0$. $\{\boldsymbol{y}_0, \boldsymbol{x}_0\}$ constitutes a low- and high-resolution *prior example pair* from which we apply a first-order regression model to estimate the high-resolution image patch $\boldsymbol{x}$ for $\boldsymbol{y}$. We repeat this procedure for overlapping patches of Y, and the final high-resolution image X is generated by aggregation all the recovered high-resolution image patches. For large upscaling factors, we iteratively repeat the above upscaling step by multiple times, each with a constant scaling factor of s. In the following, we will present details about in-place matching and regression.

3.2. In-place Matching from Scale Invariance

Although the real-world scenes are complex, images are believed to be composed of much simpler local image structures, observed as a limited type of singular primitives, such as lines and arcs [1, 7]. These local singular primitives are more invariant to scale changes, i.e., an upper scale image contains singular structures similar to those in its lower spatial scale [9]. Furthermore, we expect a singular structure in the upper scale image will have a similar structure in its origin location on the lower spatial scale. To evaluate such local scale invariance, we compute the matching error between a 5×5 query patch $\boldsymbol{y}$ at (x, y) in Y and 49 patches centering around (x_r, y_r) in image Y_0 on the Berkeley Segmentation Dataset [11]. Figures 2 plots the average matching errors for different scaling factor s, where "blue" denotes small matching error and "red" denotes large matching error. As shown, the lowest matching error occurs at

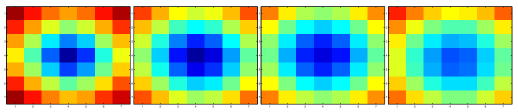

Figure 2. Matching errors of each patch $\boldsymbol{y}$ in an upper-scale image with its in-place neighbors in the lower-scale image for different scaling factors.

center (x_r, y_r) on average, and the smaller the scaling factor, the lower the matching error, and the more concentrated the area with lower matching error. Therefore, for a small scaling factor, finding a similar match for a query patch $\boldsymbol{y}$ could be extremely localized on Y_0. For image upscaling, we want to preserve the high-frequency information for the singular primitives. The local scale invariance indicates that we could efficiently find a similar example $\boldsymbol{y}_0$ for $\boldsymbol{y}$, and thus the corresponding high-resolution patch $\boldsymbol{x}_0$, from which we can infer the high-frequency information of $\boldsymbol{y}$. Formally, we refine this local self-similarity in the following proposition (proof in the appendix).

Proposition. *For each patch $\boldsymbol{y}$ of size $a \times a$ $(a > 2)$ from upsampled image Y at location (x, y), containing only one singular primitive, the location (x_0, y_0) of a close match $\boldsymbol{y}_0$ in Y_0 for $\boldsymbol{y}$ will be at most one pixel away from $\boldsymbol{y}$'s origin location $(x/s, y/s)$ on Y_0, i.e., $|x_0 - x/s| < 1$ and $|y_0 - y/s| < 1$, given the scaling factor $s < a/(a-2)$.*

Therefore, the search region for $\boldsymbol{y}$ on Y_0 is *in-place*, and $\boldsymbol{y}_0$ is called an in-place example for patch $\boldsymbol{y}$. Based on the in-place example pair $\{\boldsymbol{y}_0, \boldsymbol{x}_0\}$, we perform a first-order regression to estimate the high-resolution information for patch $\boldsymbol{y}$ in the following section.

3.3. In-place Example Regression

The patch-based single image super-resolution problem can be viewed as a regression problem, i.e., finding a mapping function f from the low-resolution patch space to the target high-resolution patch space. However, learning this regression function turns out to be extremely difficult due to the ill-posed nature of super-resolution; proper regularizations or good image priors are need to constrain the solution space. From the above analysis, the in-place example pair $\{\boldsymbol{y}_0, \boldsymbol{x}_0\}$ serves as a good *prior example pair* for inferring the high-resolution version of $\boldsymbol{y}$. Assuming some smoothness on the mapping function f[2], we have the following Taylor expansion:

$$\begin{aligned}\boldsymbol{x} =& f(\boldsymbol{y}) = f(\boldsymbol{y}_0 + \boldsymbol{y} - \boldsymbol{y}_0)\\ =& f(\boldsymbol{y}_0) + \nabla f^T(\boldsymbol{y}_0)(\boldsymbol{y} - \boldsymbol{y}_0) + o(\|\boldsymbol{y} - \boldsymbol{y}_0\|_2^2) \\ \approx& \boldsymbol{x}_0 + \nabla f^T(\boldsymbol{y}_0)(\boldsymbol{y} - \boldsymbol{y}_0),\end{aligned} \quad (1)$$

[2]This is a reasonable assumption, as one will not expect a dramatic change in the high-resolution patch for a minor change in the low-resolution image patch, especially in the case of small scaling factors.

which is based on the facts that $\boldsymbol{x}_0 = f(\boldsymbol{y}_0)$ and $\boldsymbol{y}$ is close to $\boldsymbol{y}_0$.[3] The equation is a first-order approximation for the mapping function f. Instead of learning f directly, we learn the derivative function ∇f, which is better constrained by the in-place example pairs and should be simpler.

For simplicity, we approximate the function ∇f as a piece-wise constant function by learning the function values on a set of anchor points $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_n\}$ sampled from the low-resolution patch space. Given a set of training example pairs $\{\boldsymbol{y}_i, \boldsymbol{x}_i\}_{i=1}^m$ and their corresponding prior in-place example pairs $\{\boldsymbol{y}_{0i}, \boldsymbol{x}_{0i}\}_{i=1}^m$[4], we can learn the function ∇f on the n anchor points by

$$\min_{\{\nabla f(\boldsymbol{c}_j)\}_{j=1}^n} \sum_{i=1}^m \|\boldsymbol{x}_i - \boldsymbol{x}_{0i} - \nabla f(\boldsymbol{c}_*^i)(\boldsymbol{y}_i - \boldsymbol{y}_{0i})\|_2^2, \quad (2)$$

where $\boldsymbol{c}_*^i$ is the nearest anchor point to $\boldsymbol{y}_{0i}$. The above optimization can be easily solved by least squares. With the function values on the anchor points learned, for any patch $\boldsymbol{y}$, we first search its in-place example pair $\{\boldsymbol{y}_0, \boldsymbol{x}_0\}$, find $\nabla f(\boldsymbol{y}_0)$ based on its nearest anchor point, and then use the first order approximation to compute the high-resolution patch $\boldsymbol{x}$.

Discussions In previous example-based super-resolution works [7, 6], the high-resolution image patch $\boldsymbol{x}$ is obtained by transferring the high-frequency component from the best prior example pair $\{\boldsymbol{y}_0, \boldsymbol{x}_0\}$ to the low-resolution image patch $\boldsymbol{y}$, i.e.,

$$\boldsymbol{x} = \boldsymbol{y} + (\boldsymbol{x}_0 - \boldsymbol{y}_0) = \boldsymbol{x}_0 + (\boldsymbol{y} - \boldsymbol{y}_0). \quad (3)$$

Instead of high frequency transfer, why not using $\boldsymbol{x}_0$ directly as the estimation of $\boldsymbol{x}$? Note that $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{x} - \boldsymbol{x}_0 = \boldsymbol{x}_0 + \triangle\boldsymbol{x}$. The above equation simply uses $\triangle\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{y}_0$ to approximate $\triangle\boldsymbol{x}$ for *error compensation*, which is reasonable because $\triangle\boldsymbol{y}$ is a blurred version of $\triangle\boldsymbol{x}$. Comparing the above equation with Eqn. 1, we can see that high-frequency component transfer is an approximate of the first-order regression model by setting the derivative function ∇f to be the identity matrix. By learning the derivative function, we actually learn a set of adaptive linear sharpening filters that sharpens $\triangle\boldsymbol{y}$ to approximate $\triangle\boldsymbol{x}$, and thus our approximation to the mapping function f is more accurate. On the other hand, using $\boldsymbol{x}_0$ directly as an estimation for $\boldsymbol{x}$ can be seen as a zero-order approximation of f from Eqn. 1, which does not account for error compensation and thus has larger approximation errors. Figure 3 shows the super-resolution comparisons between zero- and our first-order approximations. The results are obtained by recovering overlapping low-resolution image patches which

[3]Here, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ are in their vectorized form

[4]The prior in-place example pairs are found based on $\{\boldsymbol{y}_i\}_{i=1}^m$ only as discussed before.

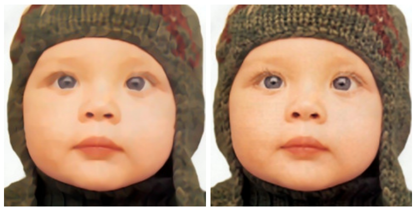

Figure 3. Example-based super-resolution results ($2\times$). Left: zero-order approximation; right: our first-order regression.

are later averaged at the same pixel locations. Because the zero-order approximation has large approximation errors, the overlapping pixel predictions do not agree with each other. As a result, averaging them will severely blur the image. Our first-order approximation is much more accurate and thus preserves the image details.

Motivated by locally linear embedding [12], Chang et al. [2] generalized the framework of [7] by proposing a locally linear model to directly learn the mapping function f. However, the algorithm still requires a large number of training examples in order to approximate f well, resulting in expensive computations for practical applications. In addition, as shown in [19], Chang's algorithm is very sensitive to noise.

3.4. Aggregating In-place Example Regressions

Due to the discrete resampling process in downsampling and upsampling for small scaling factors (non-dyadic), one typically cannot find the exact in-place example; instead, one will find multiple approximate in-place examples for $\boldsymbol{y}$ in the neighborhood of (x_r, y_r), which contains at most 9 patches. To reduce the regression variance, we can perform regression on each of them and combine the results by a weighted average. Given the in-place prior example pairs $\{\boldsymbol{y}_{0i}, \boldsymbol{x}_{0i}\}_{i=1}^{9}$ for $\boldsymbol{y}$, we have

$$\boldsymbol{x} = \sum_{i=1}^{9} w_i \left\{ \boldsymbol{x}_{0i} + \nabla f^T(\boldsymbol{y}_{0i})(\boldsymbol{y} - \boldsymbol{y}_{0i}) \right\}, \quad (4)$$

where $w_i = 1/z \cdot \exp\left(-\|\boldsymbol{y} - \boldsymbol{y}_{0i}\|_2^2 / 2\sigma^2\right)$ with z the normalization factor. The above aggregated regression on multiple in-place examples is of important practical values. In real image super-resolution scenarios, the test image might be contaminated by noise, e.g., photos shot by mobile cameras, or by compression artifacts, e.g., images from the internet. It is extremely important for the super-resolution algorithm to be robust to such image degradations. By aggregating the multiple regression results, our algorithm can handle different image degradations well in practical applications. It is worthy to note that our formulation only uses regression results on extremely localized in-place examples in a lower spatial scale, which is different from that of the non-local means algorithm [1] that operates on raw image patches in a much larger spatial window at the same spatial scale.

Table 1. Prediction RMSEs for different approaches on testing patches and images for one upscaling step ($1.5\times$).

Approaches	bicubic	zero order	transfer	regression
testing patches	8.59	14.67	9.34	**8.21**
kid	3.47	4.93	3.45	**2.88**
barbara	8.31	10.40	8.14	**6.87**
worker	8.44	9.36	7.20	**6.29**
lena	4.31	5.01	3.66	**3.21**
koala	3.79	5.12	3.59	**2.96**
girl	4.32	5.45	3.97	**3.39**
wall	10.08	11.11	9.09	**8.15**

3.5. Selective Patch Processing

Natural images typically contain large smooth regions with sparse discontinuities. Although simple interpolation methods result in artifacts along the discontinuities, they perform well on smooth regions. This observation suggests that we only need to process the textured regions with our super-resolution model, while leaving the large smooth regions to simple and fast interpolation techniques. To differentiate smooth and textured regions, we do SVD on the gradient matrix of a local image patch, and calculate the singular values $s_1 \geq s_2 \geq 0$, which represent the energies in the dominant local gradient and edge orientation. Specifically, we use the following two image content metrics defined in [20] to find the textured regions: $R = \frac{s_1 - s_2}{s_1 + s_2}, Q = s_1 \frac{s_1 - s_2}{s_1 + s_2}$, where R and Q are large for textured regions and small for smooth regions. Therefore, we can selectively process image patches with center R and Q values larger than some predefined thresholds, which leads to a speedup of $2 \sim 3$ times without compromise in the image quality.

4. Experimental Results

In this section, we evaluate our algorithm on both synthetic test examples used in the super-resolution literature and real-world test examples. In both cases, our algorithm can produce compelling results with a practically fast speed.

Parameters We choose patch size $a = 5$ and iterative scaling factor $s = 1.5$ in our experiments, in order to satisfy the in-place matching constraints in the proposition. The low-frequency band Y of the target high-resolution image is approximated by bicubic interpolation from X_0. The low-frequency band Y_0 of the input image X_0 is obtained by a low-pass Gaussian filtering with a standard deviation of 0.55. The image patches of Y are processed with overlapping pixels, which are simply averaged to get the final result. For clean images, we use the nearest neighbor in-place example for regression, and for noisy images, we average all 9 in-place example regressions for robust estimation, where σ is the only tuning parameter to compute w_i in Eqn. 4 depending on the noise level.

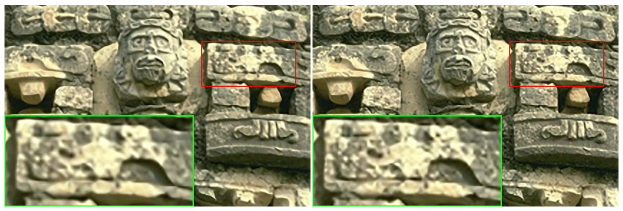

Figure 4. Super-resolution ($3\times$) comparison between frequency transfer and regression.

Training To train the regression model, we start from a collection of high-resolution natural images $\{X_i\}_{i=1}^r$ from the Berkeley Segmentation Dataset [11]. The corresponding lower spatial scale images $\{X_{0i}\}_{i=1}^r$ are generated by blurring and downsampling the high-resolution images by a factor of s. The two low-frequency band image sets $\{Y_i\}_{i=1}^r$ and $\{Y_{0i}\}_{i=1}^r$ are generated as described above. We then randomly sample image patch pairs from $\{Y_i\}_{i=1}^r$ and $\{X_i\}_{i=1}^r$ to obtain the low- and high-resolution patch pairs $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^m$, and meanwhile get the corresponding in-place matching image pairs $\{\boldsymbol{x}_0^i, \boldsymbol{y}_0^i\}_{i=1}^m$ from $\{X_{0i}\}_{i=1}^m$ and $\{Y_{0i}\}_{i=1}^r$. The anchor points $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_n\}$ are obtained by simple unsupervised clustering with the random projection tree [8] from $\{\boldsymbol{y}_{0i}\}_{i=1}^m$. By using the random projection tree for clustering, we are able to efficiently find the nearest anchor point for a given image patch. In our experiments, we find that the learned regression model can be very compact; using $2^7 = 128$ anchor points already suffice for our purpose.

4.1. Regression Evaluations

To measure the regression accuracy from the recovery perspective, we first conduct quantitative comparisons for different methods in terms of RMSE. Table 1 reports the results on both testing patches and synthetic images taken from [11] for one upscaling step ($1.5\times$). As shown, zero-order regression performs the worst due to its large approximation errors. Frequency transfer performs much better than zero-order regression due to the error compensation for $\Delta \boldsymbol{x}$ by $\Delta \boldsymbol{y}$. Our first-order regression model performs the best.[5] In Figure 4, we show super-resolution results ($3\times$) on the "Mayan architecture" image with frequency transfer and our regression model, respectively. By comparison, our regression estimation is more accurate and thus can preserve more sharp details compared with frequency transfer.

[5]It is worthy to note that the algorithms discussed here are based on multiple iterative upscaling steps for large scaling factors. So, minor structure distortions and estimation errors may propagate and accumulate over several iterations. As a result, it is hard to compare the image quality after several steps in terms of RMSE. Therefore, we only make comparisons for one upscaling step to keep the experiment clean.

4.2. Visual Quality Evaluations

We compare our algorithm with the recent state-of-the-art algorithms [9, 18, 6] in terms of visual quality on both commonly adopted super-resolution test examples and real-world images. Specifically, for methods based on external examples, we choose the representative work [18] for comparison; for methods based on self-examples, we choose the recent two representative works [9, 6] for comparison. For visual quality comparison, conventional algorithms have been focusing on image sharpness, ignoring the image naturalness (affected by visual artifacts) to a large extent. For user experience in real applications, both image sharpness and naturalness, and algorithm speed are all important. Therefore, in the following comparisons, we pay attentions to all three aspects. Note that the results and comparisons are best viewed in zoomed PDF. We also provide the original images in the supplementary material for better comparison.

In general, the algorithm of [18] can produce natural-looking results with a pre-trained universal dictionary, even though they are a little blurry. However, it also creates small artifacts across the image due to the fact that some unique patches to the image cannot be well represented by the universal dictionary, and the algorithm is also much slower than ours. The algorithms of [9] and [6] are based on self-examples, which tend to create artificially sharp edges and artifacts due to the insufficient number of matched self-examples. In contrast, by leveraging learning from external examples and learning from self-examples, our algorithm can produce sharp details without noticeable visual artifacts.

Figure 5 shows the super-resolution results of different approaches on "kid" and "chip" by $4\times$ and on "cameraman" by $3\times$. As shown, the results of Glasner et al. [9] are overly sharp, resulting in noticeable visual artifacts, e.g., ghost artifacts along the cheek in "kid", jaggy artifacts on the long edge in "chip", and artifacts in the camera area in "cameraman". The results of Yang et al. [18] are generally a little burry and they contain many small artifacts across the image upon a closer look. Freedman's algorithm is good at enhancing edges but is prone to create rigid artificial edges and small structure distortions, e.g., character "A" in "chip" and grass region in "cameraman", due to the insufficient number of self-examples. It also fails to preserve some local details, e.g., pupil in "kid", as it is using frequency transfer to generate high-resolution patches. In comparison, our algorithm is able to recover local texture details as well as sharp edges without sacrificing the naturalness.

Most previous super-resolution works focus on synthetic test examples with simple edge structures, but the large body of natural images typically contain diverse textures and rich fine structures. Figure 6 shows another set of super-resolution results ($3\times$) on images with cluttered texture re-

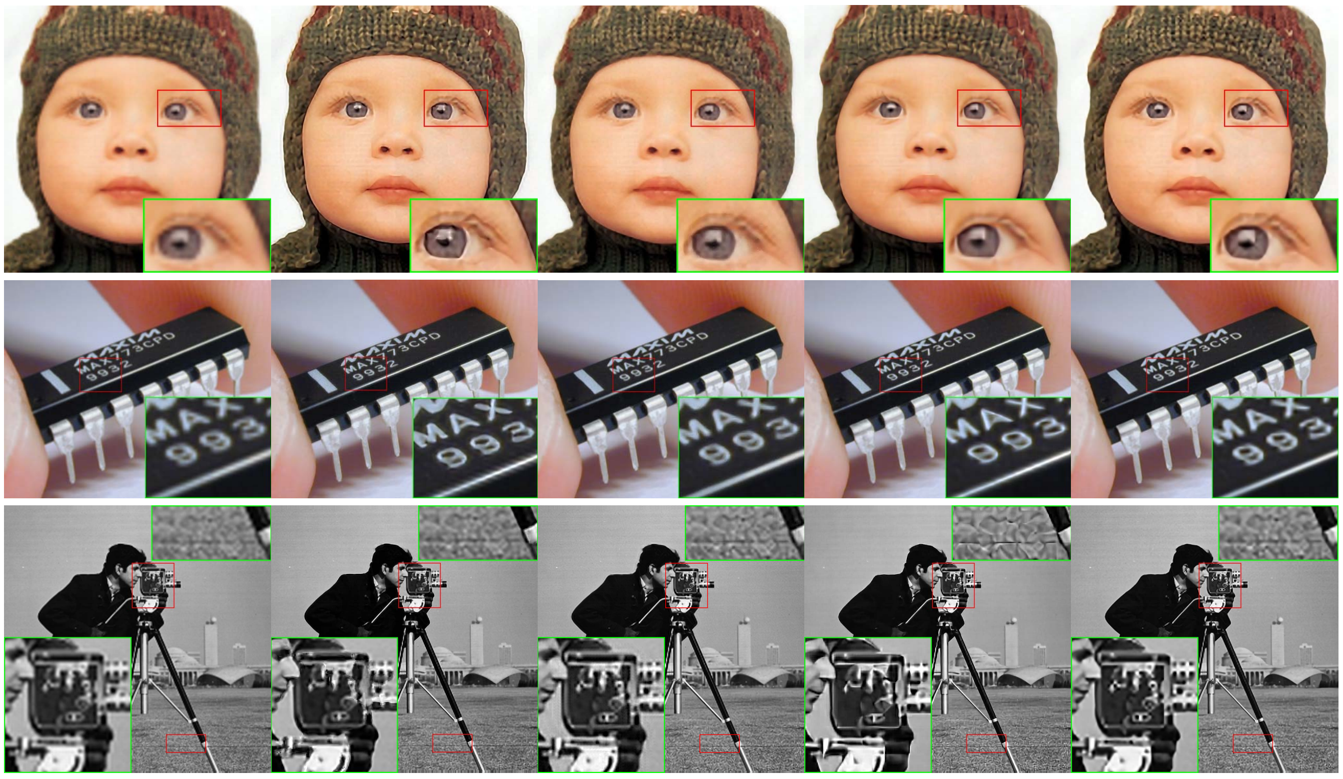

Figure 5. Super-resolution results on “kid” (4×), “chip” (4×) and “Cameraman” (3×). Our algorithm can generate natural-looking results without noticeable visual artifacts. Results are better viewed in zoomed PDF.

gions. The algorithm in [18] can produce natural looking results, but the textures are a little blurry with occassional small artifacts. Freedman and Fattal’s again fails to recover the fine details and produces many rigid false edges in the textured regions. Besides images with complex and diverse structures, we also encounter frequently noisy images in real applications, e.g., images captured by low cost sensors are typically contaminated by some amount of sensor noise and internet images with JPEG compression artifacts. It is extremely important that the super-resolution algorithm is robust to such degradations. By averaging the regression results on multiple in-place self-examples, our algorithm can naturally handle noisy inputs. Figure 7 shows one more set of super-resolution results (3×) on real-world images that are corrupted with either sensor noise or compression artifacts. As shown, the algorithms in [9] and [6] cannot distinguish noise from the signal and thus enhance both, resulting in magnified noise artifacts, while our algorithm almost completely eliminates the noise and at the same time preserves sharp image structures. The de-noising effectiveness of our algorithm also validates our proposed in-place self-similarity.

Computational Efficiency With fast in-place matching and selective patch processing, our algorithm is much faster than Glasner’s algorithm [9], is at least one order of magnitude faster than Yang’s algorithm [18], and is comparable with Freedman’s algorithm [6]. For example, on a modern machine with Intel CPU 2.8 GHz and 8 GB memory, it takes 3.2 seconds to upscale the “cameraman” image of 256×256 pixels to 1024×1024 pixels. The algorithm can easily be parallelized with GPU for real-time processing.

5. Conclusions

In this paper, we propose a robust first-order regression model for image super-resolution based on justified in-place self-similarity. Our model leverages the two most successful super-resolution methodologies of learning from an external training database and learning from self-examples. Taking advantage of the in-place examples, we can learn a fast and robust regression function for the otherwise ill-posed inverse mapping from low- to high-resolution patches. On the other hand, by learning from an external training database, the regression model can overcome the problem of insufficient number of self-examples for matching. Compared with previous example-based approaches, our new approach is more accurate and can produce natural looking results with sharp details. In many practical applications where images are contaminated by noise or compression artifacts, our robust formulation is of particular importance.

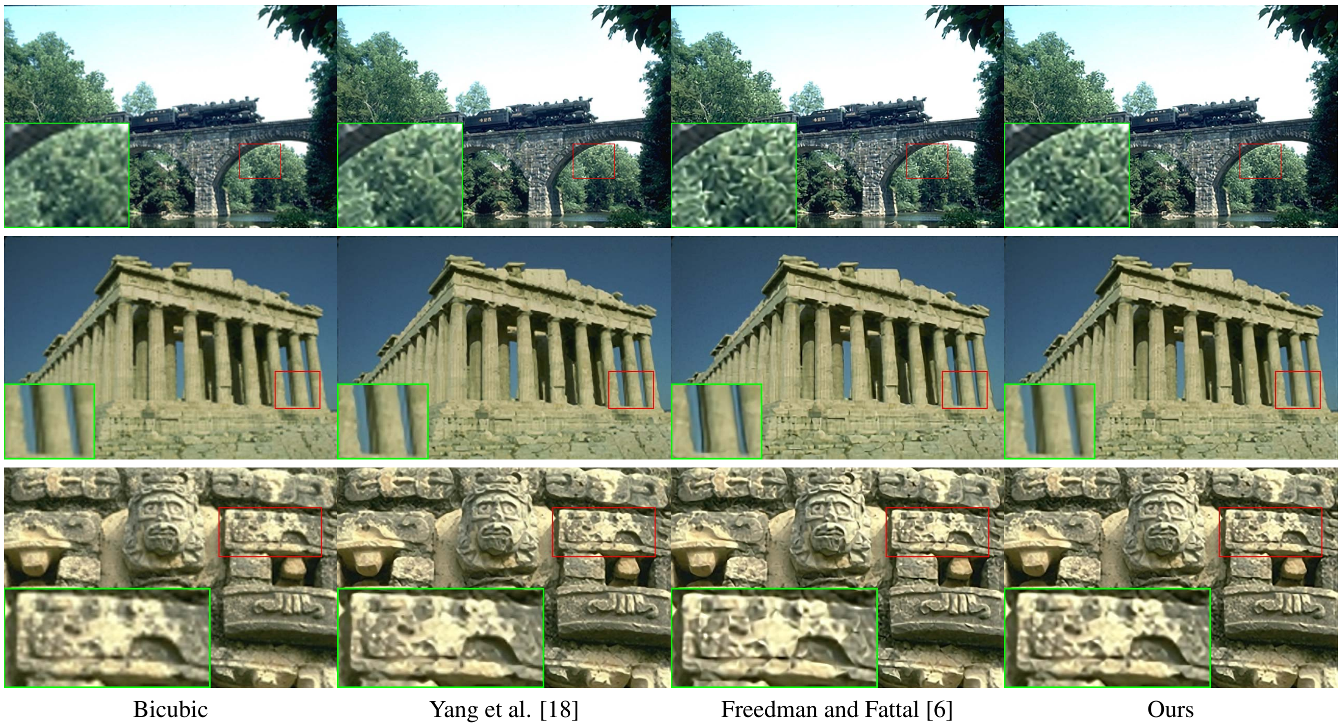

Figure 6. Super-resolution results by $3\times$. The results of [18] contain small artifacts along edges (best perceived in zoomed PDF). The results of [6] contain many artificial edges and distorted structures across the image. Our results are more natural looking with preserved details. Note that the algorithm of [18] is at least one order of magnitude slower than our algorithm.

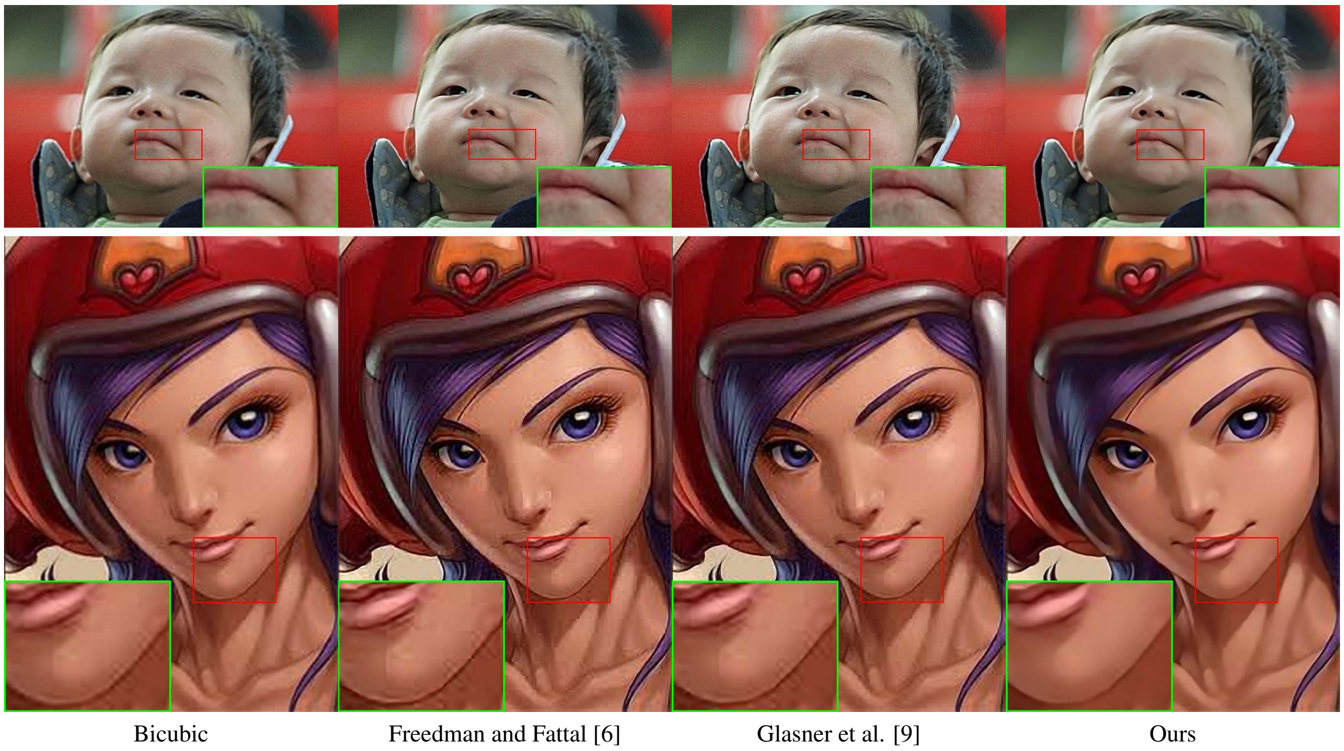

Figure 7. Super-resolution by $3\times$ on real-world noisy images. The first image was captured by a mobile camera in a daily situation, which is contaminated by a certain amount of sensor noise. The second image was collected from the internet, which is corrupted by compression artifacts. Our algorithm can handle these noise well.

References

[1] A. Buades, B. Coll, and J. M. Morel. A non local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[2] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 275–282, 2004.

[3] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[4] M. Ebrahimi and E. Vrscay. Solving the inverse problem of image zooming using self-examples. In *Internaltional Conference on Image Analysis and Recognition*, pages 117–130, 2007.

[5] R. Fattal. Upsampling via imposed edge statistics. *ACM Transactions on Graphics*, 26(3), 2007.

[6] G. Freedman and R. Fattal. Imag and video upscaling from local self-examples. *ACM Transactions on Graphics*, 28(3):1–10, 2011.

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

[8] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma. Learning the structure of manifolds using random projections. In *NIPS*, 2007.

[9] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *IEEE International Conference on Computer Vision*, 2009.

[10] H. He and W. C. Siu. Single image super-resolution using gaussian process regression. In *CVPR*, 2011.

[11] 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 *IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[12] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2201–2372, 2000.

[13] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[14] J. Sun, N.-N. Zheng, H. Tao, and H.-Y. Shum. Image hallucinatoin with primal sketch priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 729–736, 2003.

[15] Y. W. Tai, S. Liu, M. S. Brown, and S. Lin. Super resolution using edge prior and single image detail synthesis. In *CVPR*, 2010.

[16] M. F. Tappen, B. C. Russel, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.

[17] Q. Wang, X. Tang, and H. Shum. Patch based blind image super-resolution. In *ICCV*, 2005.

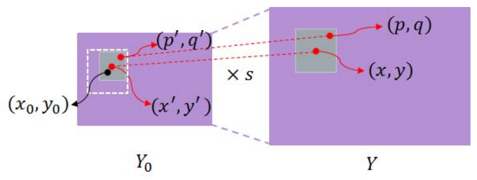

Figure 8. Illustration of the in-place matching proof. In order to get sharp results for singular structure (p, q), we want to match (p, q) with (p', q') in searching the example pair.

[18] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. S. Huang. Coupled dictionary learning for image super-resolution. *IEEE Transactions on Image Processing*, 21:3467–3478, 2012.

[19] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11), 2010.

[20] X. Zhu and P. Milanfar. Automatic parameter selection for denoising algoirthms using a non-reference measure of image content. *IEEE Transactions on Image Processing*, 19:3116–3132, 2010.

[21] M. Zontak and M. Irani. Internal statistics of a single natural image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

A. Proof of in-place matching

Proof. For simplicity, our discussion is based on continuous image signals. Given a patch $\boldsymbol{y}$ $(a \times a)$ from the low-frequency band image Y centered at (x, y), shown in Figure 8, we assume it has a singular structure centered at (p, q). This point has a shift from the patch center by $p = x + t_x$ and $q = y + t_y$ $(|t_x|, |t_y| < a/2)$. (x', y') and (p', q') are the corresponding coordinates of (x, y) and (p, q), respectively, on image plane Y_0, i.e., $x' = x/s, y' = y/s$ and $p' = p/s, q' = q/s$. For super-resolution, we desire to preserve the high-frequency information of the singular structures. Therefore, we want to match the singular point (p, q) to (p', q') in order to infer the high-frequency information. As a result, the center of the desired $\boldsymbol{y}_0$ will be at $x_0 = p' - t_x, y_0 = q' - t_y$ (note that $\boldsymbol{y}_0$ has the same size as $\boldsymbol{y}$). Therefore, for $s > 1$, we have

$$\begin{aligned} |x_0 - x'| =& |p' - t_x - x/s| = |p/s - t_x - x/s| \\ =& |x/s + t_x/s - t_x - x/s| \\ =& |(1 - s)t_x/s| < (s - 1)a/2s. \end{aligned} \tag{5}$$

If the patch size $a \leq 2$, we have $(s - 1)a/2s \leq 1$ for any s. In practice, we have $a > 2$, then $(s - 1)a/2s \leq 1$ holds for $s \leq a/(a - 2)$. For example, if the image patch size $a = 5$, we have $|(1 - s)t_x/s| < 1$, as long as the scaling factor $s \leq 5/3$. Similarly results can be shown for $|y_0 - y'|$, which means that the center of the desired match $\boldsymbol{y}_0$ will be constrained to be less than one pixel city block distance from (x', y'), and thus the matching is called *in-place*. □