Nonparametric Blind Super-Resolution

Tomer Michaeli and Michal Irani
Dept. of Computer Science and Applied Mathematics
Weizmann Institute of Science, Israel

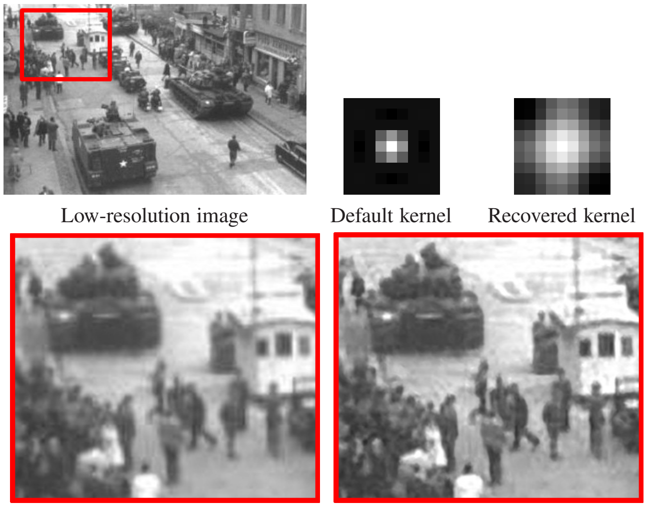
Abstract

Super resolution (SR) algorithms typically assume that the blur kernel is known (either the Point Spread Function 'PSF' of the camera, or some default low-pass filter, e.g. *a Gaussian). However, the performance of SR methods significantly deteriorates when the assumed blur kernel deviates from the true one. We propose a general framework for "blind" super resolution. In particular, we show that: (i) Unlike the common belief, the PSF of the camera is the* wrong *blur kernel to use in SR algorithms. (ii) We show how the correct SR blur kernel can be recovered directly from the low-resolution image. This is done by exploiting the inherent recurrence property of small natural image patches (either internally within the same image, or externally in a collection of other natural images). In particular, we show that recurrence of small patches across scales of the low-res image (which forms the basis for single-image SR), can also be used for estimating the optimal blur kernel. This leads to significant improvement in SR results.*

1. Introduction

Super-resolution (SR) refers to the task of recovering a high-resolution image from one or several of its low-resolution versions. Most SR methods (*e.g.* [11, 7, 2, 4, 20, 16, 15, 21]) assume that the low-resolution input image was obtained by down-sampling the unknown high-resolution image with a *known* blur kernel. This is usually assumed to be the Point Spread Function (PSF) of the camera. When the PSF is unknown, the blur kernel is assumed to be some standard low-pass filter (LPF) like a Gaussian or a bicubic kernel. However, the PSF may vary in practice with zoom, focus, and/or camera shake. Relying on the wrong blur kernel may lead to low-quality SR results, as demonstrated in Fig. 1. Moreover, we show that unlike the common belief, *even if the PSF is known, it is the wrong blur kernel to use in SR algorithms!* We further show how to obtain the optimal SR blur kernel directly from the low-resolution image.

A very limited amount of work has been dedicated to "blind SR", namely SR in which the blur kernel is not assumed known. Most methods in this category assume some parametric model for the kernel (*e.g.* still a Gaussian, but with unknown width) and infer its parameters [17, 1, 19]. Extension to multiple parametric models appears in [10]. A nonparametric kernel recovery method was presented in [12] both for deblurring and for SR. This method assumes that the kernel has a single peak, which is a restrictive assumption in the presence of motion blur. In [18, 9], methods for *jointly* estimating the high-res image and a nonparametric kernel were developed. This joint estimation strategy, however, was shown to lead to erroneous results [13].


Low-resolution image

Default kernel

Recovered kernel

SR of [8] with its default kernel

SR of [8] with our recovered kernel

Figure 1: ***Blind SR on an old low-quality image (end of World War II) downloaded from the Internet****. The blur kernel was recovered directly from the low-res image (see Sec. 3). Note small recovered details like the heads of people, the white star on the tank and the bars on the window.*

In this paper, we develop a general robust approach for blind SR. Our method is based on the universal property that natural image patches tend to recur abundantly, both across scales of the same image, as well as in an external database of other natural images [22, 8, 7]. Our contributions are fourfold. First, we address the question: What is the optimal blur kernel relating the unknown high-res image to the input low-res image? As mentioned above, we analytically show that, in contrast to the common belief, *the optimal blur kernel* k *is not the PSF*. In fact, k is typically narrower than the PSF and, counter-intuitively, it may assume negative values, a property not shared by the PSF.

Our second contribution is the observation that k can be estimated by relying on patch recurrence across scales of the input low-res image, a property which has been previously used for (non-blind) single-image SR [8, 4]. In particular, we show that *the kernel that maximizes the similarity of recurring patches across scales of the low-res image, is also the optimal SR kernel*. Based on this observation, we propose an iterative algorithm for recovering the kernel.

Many example-based SR algorithms rely on an external database of low-res and high-res pairs of patches extracted from a large collection of high-quality example images [20, 21, 7, 2]. They too assume that the blur kernel k is known a-priori (and use it to generate the low-res versions of the high-res examples). We show how our kernel estimation algorithm can be modified to work with an external database of images, recovering the optimal kernel relating the low-res image to the external high-res examples.

Our last contribution is a proof that our algorithm computes the MAP estimate of the kernel, as opposed to the joint MAP (over the kernel and high-res image) strategy of [19, 10, 18, 9]. The benefit of this approach has been studied and demonstrated in the context of blind deblurring [13].

We show that plugging our estimated kernel into existing super-resolution algorithms results in improved reconstructions that are comparable to using the ground-truth kernel.

2. What is the correct SR kernel?

Let l denote a low-resolution image. Super-resolution can be viewed as trying to recover a high-resolution image h of the same scene, as if taken by the same camera, but with an *optical* zoom-in by a factor of α. This allows to view fine details, that are smaller than a pixel size in l.

The low-res image l is generated from a continuous-space image f and a continuous-space PSF b_{L}, as[1]

$$l[n] = \int f(x) b_{\mathrm{L}}(n-x)dx. \tag{1}$$

The *unknown* high-res image h corresponds to a finer grid,

$$h[n] = \int f(x) b_{\mathrm{H}}\left(\frac{n}{\alpha} - x\right) dx, \tag{2}$$

where b_{H} is the high-res PSF. *E.g.,* in the case of optical zoom, b_{H} is a *narrower* (scaled-down) version of b_{L}, namely

$$b_{\mathrm{H}}(x) = \alpha b_{\mathrm{L}}(\alpha x). \tag{3}$$

The relations between h, l and f are illustrated in Fig. 2. The images h and l are related by blur and subsample: $l = (h * k) \downarrow_\alpha$, *i.e.*

$$l[n] = \sum_m h[m] k[\alpha n - m]. \tag{4}$$

Here, k is a discrete blur kernel. In fact this is the SR blur kernel we are interested in. While k is often assumed to resemble the PSF, we show next that *the optimal SR kernel is not a simple discretization nor approximation of the PSF*.

Substituting (1) and (2) into (4) and requiring that the equality hold for every possible function f leads to

$$\sum_m k[m] b_{\mathrm{H}}\left(x - \frac{m}{\alpha}\right) = b_{\mathrm{L}}(x). \tag{5}$$

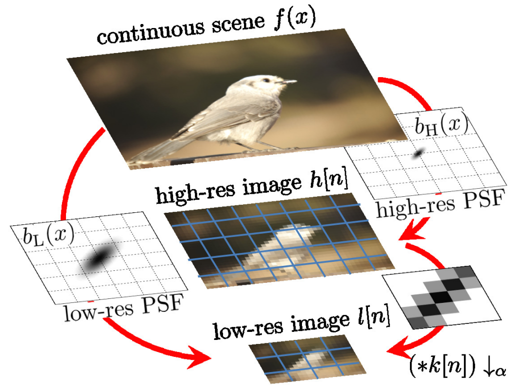


Figure 2: ***Relations between the digital low-res, digital high-res, and continuous-domain images.***

In other words, b_{L} is a linear combination of translated versions of b_{H}, and the *coefficients* of this representation constitute the SR kernel k. Generally, it is not always possible to satisfy the relation (5) exactly. The optimal discrete blur, in such cases, corresponds to the projection of b_{L} onto the space spanned by $\{b_{\mathrm{H}}(x - m/\alpha)\}$, and can be obtained as the least squares solution of Eq. (5).

Counter-intuitively, in certain settings the optimal blur kernel k relating h and l does not share much resemblance to the PSF b_{L}. In particular, k is not guaranteed to assume only positive values, as often postulated in the context of blind-deblurring [13], and is often narrower and/or of a more oscillatory nature than b_{L}. These phenomena are illustrated in Fig. 3 for a 1D rectangular PSF.

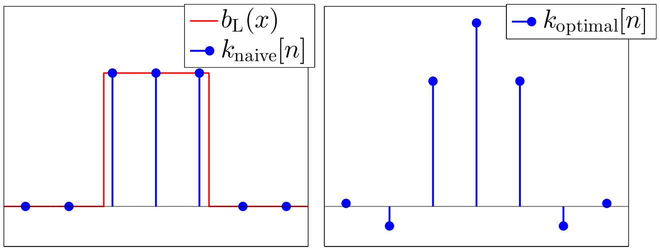
The physical interpretation of the kernel k can be intuitively understood from Fig. 2 as composed of two operations: Deconvolution with b_{H}, followed by convolution with b_{L}. Thus, intuitively (ignoring sampling issues), k should have the same effect as a continuous-domain filter k_{c} whose Fourier transform is[2]

$$K_{\mathrm{c}}(\omega) = \frac{B_{\mathrm{L}}(\omega)}{B_{\mathrm{H}}(\omega)} = \frac{B_{\mathrm{L}}(\omega)}{B_{\mathrm{L}}(\omega/\alpha)}, \tag{6}$$

where $K_{\mathrm{c}}, B_{\mathrm{L}}, B_{\mathrm{H}}$ are the Fourier transforms of $k_{\mathrm{c}}, b_{\mathrm{L}}, b_{\mathrm{H}}$. If the PSF is an ideal low-pass filter (a sinc in the image domain; a rect in the Fourier domain), then the kernel k_c indeed equals to the PSF b_{L}, because $\mathrm{rect}(\omega)/\mathrm{rect}(\omega/\alpha) = \mathrm{rect}(\omega)$. However, typically $B_{\mathrm{L}}(\omega)$ is not flat on its entire support, and it smoothly decays towards the high frequencies. Consequently, the division by $B_{\mathrm{L}}(\omega/\alpha)$ amplifies the high frequencies in $K_{\mathrm{c}}(\omega)$ with respect to $B_{\mathrm{L}}(\omega)$, implying that the optimal kernel is usually narrower than the PSF. For example, if the PSF is a Gaussian with width σ, then direct computation shows that k_c is also a Gaussian, but with width $\sigma\sqrt{1 - 1/\alpha^2}$, which is smaller than σ. When the PSF has sharp transitions, the kernel often exhibits an oscillatory

[1] To aid distinguish between continuous- and discrete-space functions, we use parentheses for the former and square brackets for the latter.

[2] More rigorously, least-square problems of the type (5) are known to have a closed form solution [3], which corresponds to samples of a function, whose Fourier transform is $\frac{B_{\mathrm{L}}(\omega)B_{\mathrm{H}}^*(\omega)}{\sum_m |B_{\mathrm{H}}(\omega - 2\pi\alpha m)|^2}$.



Discretization of the PSF

Optimal blur k computed from (5)

Figure 3: ***The optimal blur kernel is not a simple discretization of the low-res PSF*** $b_L(x)$ *(computed for* $\alpha = 2$*).*

behavior, as can be seen[3] in Fig. 3.

3. Kernel estimation using internal examples

Recurrence of small image patches (*e.g.*, 5×5, 7×7) across scales of a single natural image was analyzed and quantified in [8, 22] for a very wide diversity of images. Such cross-scale patch recurrence has been shown to provide a strong cue for SR from a single image [8, 6]. These SR methods, however, assumed a known blur kernel. We next show how the recurrence of patches across scales can also be exploited to recover the correct blur kernel k relating the unknown high-res image h with the low-res image l. In fact, we show that the kernel which maximizes similarity of recurring patches across scales of the low-res image l, is also the optimal kernel k between images h and l.

The observation that small image patches recur across scales of an image, implies that small patterns recur in the continuous scene at multiple sizes. Let $f(x)$ be such a pattern, recurring elsewhere in the continuous scene as $f(x/\alpha)$, *i.e.*, α times larger (depicted by blue stars in Fig. 4(a)). These two continuous patterns are observed in the low-res image by two discrete patterns of different sizes, contained in patches q and r (see Fig. 4(b)). We next show that the low-res patches q and r are related to each other by blur and subsampling with the (unknown) *optimal blur kernel* k derived in Eq. (5). This is despite the fact that the two patches are captured with the same low-res PSF b_L (depicted by an orange ellipse in Fig. 4(a)). This observation forms the basis for our kernel recovery algorithm.

Patches q and r are generated by the low-res PSF b_L as

$$q[n] = \int f(x) b_L(n - x)dx, \tag{7}$$

$$r[n] = \int f\left(\frac{x}{\alpha}\right) b_L(n - x)dx. \tag{8}$$

With a change of integration variables, r can be rewritten as

$$r[n] = \int f(x)\alpha b_L(n - \alpha x)dx = \int f(x) b_H\left(\frac{n}{\alpha} - x\right)dx, \tag{9}$$

where we used (3) for the last step. Eqs. (7) and (9) entail that q and r are low-res and high-res manifestation of the same continuous scene f: q with the PSF b_L and with grid spacing 1, while r corresponds to a high-res version of q with the PSF b_H and grid spacing $1/\alpha$. Our claim is that q corresponds to a down-sampled version of r with the optimal SR kernel k derived in Sec. 2, namely

$$q[n] = \sum_m r[m]k[\alpha n - m]. \tag{10}$$

This claim can be verified by substituting (7) and (9) into (10) and using the property (5) of the optimal k.

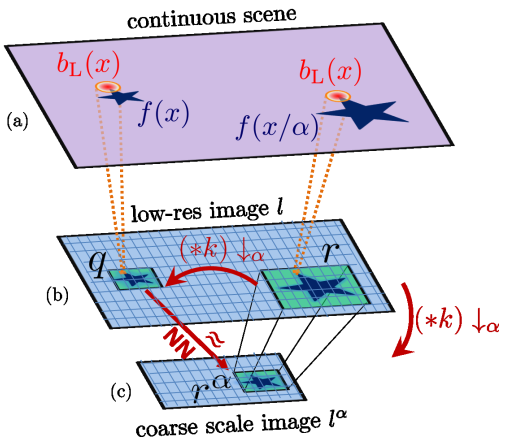


Figure 4: ***Exploiting patch-recurrence across scales for kernel recovery.*** *(a) A small pattern* $f(x)$ *recurs in the continuous scene at multiple sizes (blue stars). (b) Patches* q *and* r *are discrete observations of these patterns in the low-res image* l*. Orange ellipses depict the camera PSF, relating each pixel in* l *to its source area in the continuous scene. Eq. (10) entails that these low-res patches are related by the unknown optimal SR kernel* k*:* $q = (r * k)\downarrow_\alpha$*. (c)* r^α *is the child patch of* r *in the coarser image* l^α*. If the coarse image is generated with the kernel* k*, then* $r^\alpha = q$*. See* `www.wisdom.weizmann.ac.il/~vision/BlindSR.html` *for a more elaborate visual explanation.*

3.1. Recovering the blur kernel

Eq. (10) induces a linear constraint on the unknown coefficients of the kernel k. Multiple such patch pairs in the low-res image would thus allow recovery of k. Note that, as shown in Fig. 4, the righthand side of Eq. (10) corresponds to the coarser-scale patch r^α in the coarser version l^α of the input image l, namely $r^\alpha = q$ (see Fig. 4(c)). Therefore, detecting Nearest-Neighbors (NNs) for the patch q in the coarser image l^α provides candidate parents $r \in l$ for q. The problem with this strategy is, however, that to form the “correct” coarse image l^α we need to know k, which is what we were looking for in the first place. Nonetheless, our key observation is that Eq. (10) implies that *the correct blur kernel* k *is also the one which maximizes similarity of NNs across scales in the low-res image* l. We use this property in our algorithm to obtain the optimal kernel k.

Our approach is iterative by nature. Namely, we start with some initial guess $\hat{k}$ for k, such as a delta function, and

[3] Note that in this case the PSF is a rect in the image domain and a sinc in the Fourier domain, so that $K_c(\omega) = \text{sinc}(\omega)/\,\text{sinc}(\omega/\alpha)$.

use it to down-sample l. Next, for each small patch q_i in l we find a few nearest neighbors (NNs) in l^α and regard the large patches right above them as the candidate "parents" of q_i. The so found parent-child pairs (q, r) are then used to construct a set of linear equations (using (10)), which is solved using least-squares to obtain an updated $\hat{k}$. These steps are repeated until convergence. Note that the least-squares step does not recover the initial kernel $\hat{k}$ we use to down-sample the image, but rather a kernel that is closer to the true k. This is because, as our experience shows, even with an inaccurate kernel, many of the NNs of q_i are located at the same image coordinates in l^α as the NNs corresponding to the correct k. Thus, in each iteration the number of correct NNs increases and the estimate gets closer to k. As long as the root mean squared error (RMSE) between patches in l and their NNs in l^α keeps decreasing, we keep iterating. Our experiments show that this process converges to the correct k very quickly (typically ~ 15 iterations), and is *insensitive* to the choice of initial guess $\hat{k}$ (see Figs. 6 and 7 in Sec. 6).

Not all patches in l need to recur (have similar patches) in the coarse image l^α in order to be able to recover k. For example, recovering a 7×7 discrete kernel k relating high-res h with low-res l (49 unknowns) may be done with as little as one good 7×7 patch recurring in scales l and l^α (providing 49 equations). When using multiple patches, the NNs should intuitively be weighted according to their degree of similarity to the query patches $\{q_i\}$, so that good NNs contribute more than their poor counterparts. It may also be desired to add a regularization term to the least-squares problem solved in each iteration, in order to impose smoothness constraints on the kernel. We defer the discussion on these issues to Sec. 5, where we also provide a detailed summary of the algorithm (see Alg. 1).

Figs. 1,5,9 show examples of single-image SR with the method of [8], once using their default (bicubic) kernel, and once using our kernel recovered from the low-res image.

4. Kernel estimation using external examples

Many example-based SR algorithms rely on an external database of high-res patches extracted from a large collection of high-res images [20, 21, 7, 2]. They too assume that the blur kernel k is known a-priori, and use it to down-sample the images in the database in order to obtain pairs of low-res and high-res patches. Roughly speaking, these pairs are used to learn how to "undo" the operation of down-sampling with k, so that, given an input image l, its corresponding high-res version h can be reconstructed. We first explain the physical interpretation of the optimal kernel k when using an external database of examples, and then show how to estimate this optimal k.

Let us assume, for simplicity, that all the high-res images in the external database were taken by a single camera with a single PSF. Since the external images form examples of the high-res patches in SR, this implicitly induces that the high-res PSF b_H is the PSF of the external camera. The external camera, however, is most likely not the same as the camera imaging the low-res input image l (the "internal" camera). Thus, the high-res PSF b_H and the low-res PSF b_L may not necessarily have the same profile (*i.e.* one is no longer a scaled version of the other). It is important to note that the discussion in Sec. 2 remains valid. Namely the kernel k relating the high-res and low-res images is still given by Eqs. (5) and (6) in these settings, but with a PSF b_H of a different camera. According to Eq. (6), the intuitive understanding of the optimal kernel k when using external high-res examples is (in the Fourier domain):

$$K_\mathrm{c}(\omega) = \frac{B_\mathrm{L}(\omega)}{B_\mathrm{H}(\omega)} = \frac{\mathcal{PSF}_\mathrm{Internal}(\omega)}{\mathcal{PSF}_\mathrm{External}(\omega)}. \qquad (11)$$

Thus, in SR from external examples the high-res patches correspond to the PSF b_H of the external camera, and the low-res patches generated from them by downsampling with k *should* correspond, by construction, to the low-res PSF b_L of the internal camera. This k is generally unknown (and is assumed by external SR methods to be some default kernel, like a Guassian, or a bicubic kernel).

Determining the optimal kernel k for external SR can be done in the same manner as for internal SR (Sec. 3.1), with the only exception that the "parent" patches $\{r_i\}$ are now sought in an external database rather than within the input image. As before, we start with an initial guess $\hat{k}$ to the kernel k. We down-sample the external patches $\{r_i\}$ with $\hat{k}$ to obtain their low-res versions $\{r_i^\alpha\}$. Then each low-res patch q in the low-res image l searches for its NNs among $\{r_i^\alpha\}$, to find its candidate parents $\{r_i\}$ in the database. These "parent-child" pairs (q, r) are used to recover a more accurate kernel $\hat{k}$ via a least-squares solution to a system of linear equations. This process is repeated until convergence to the final estimated kernel. Fig. 5 provides an example of external SR with the algorithm of [21], once using their default (bicubic) kernel k, and once using our kernel k estimated from their external examples.

5. Interpretation as MAP estimation

We next show that both our approaches to kernel estimation (internal and external) can be viewed as a principled Maximum a Posteriori (MAP) estimation. Some existing blind SR approaches attempt to *simultaneously* estimate the high-res image h and the kernel k [19, 10, 18, 9]. This is done by assuming a prior on each of them and maximizing the posterior probability $p(h, k|l)$. In the context of blind deblurring, this $\mathrm{MAP}_{h,k}$ approach has been shown to lead to inaccurate estimates of k [13]. This is in contrast to the MAP_k method [5, 13, 14] which seeks to estimate k alone, by maximizing its marginal posterior, $p(k|l)$. As we show next, the algorithm presented in Secs. 3 and 4 computes the MAP_k estimate. However, unlike [14], our MAP_k is based on a *nonparametric* prior.

```
Input: query low-res patches $\{q_1, \dots, q_M\}$, example
       high-res patches $\{r_1, \dots, r_N\}$
Output: The MAP$_k$ estimate $\hat{k}$
initialize $\hat{k}$ with a delta function;
for $t = 1, \dots, T$ do
    1. down-sample example patches:
       $r_j^\alpha = \boldsymbol{R}_j \hat{k}$
    2. find NNs and compute weights:
       $w_{ij} = \frac{\exp\{-\frac{1}{2}\|q_i - r_j^\alpha\|^2/\sigma^2\}}{\sum_j \exp\{-\frac{1}{2}\|q_i - r_j^\alpha\|^2/\sigma^2\}}$
    3. update $\hat{k}$:
       $\hat{k} = \left(\frac{1}{\sigma^2}\sum_{i,j} w_{ij}\boldsymbol{R}_j^T\boldsymbol{R}_j + \boldsymbol{C}^T\boldsymbol{C}\right)^{-1} \sum_{i,j} w_{ij}\boldsymbol{R}_j^T q_i$
end
```

Algorithm 1: MAP kernel estimation

In Sec. 4 we used a collection of patches $\{r_i\}$ from an external database as candidates for constituting "parents" to small patches from the input image l. In Sec. 3, it were large patches from l itself, which played the role of potential "parents". Common to both approaches, therefore, is the use of a set of patches $\{r_i\}$ which constitute a good nonparametric representation of the probability distribution of high-res patches acquired with the PSF b_{H}. In the sequel, we therefore assume we have access to a database $\{r_1, \dots, r_N\}$ of patches, independently drawn from the distribution $p(h)$.

Assume that the low-res image l corresponds to a noisy down-sampled version of the high-res image h. Then, every patch q_i in l can be expressed in terms of the corresponding high-res patch h_i in h as

$$q_i = \boldsymbol{K} h_i + n_i. \tag{12}$$

Here, h_i and q_i are column vectors representing the high-res and low-res patches, $\boldsymbol{K}$ is a matrix that corresponds to convolution with k and sub-sampling by α, and n_i is noise. Assuming that $\{h_i\}$, $\{n_i\}$ and k are statistically independent[4], the MAP$_k$ estimate is given by

$$\begin{aligned} \hat{k} &= \arg\max_k p(k) \prod_{i=1}^{M} p(q_i|k) \\ &= \arg\max_k p(k) \prod_{i=1}^{M} \int_{h_i} p(q_i|h_i, k) p(h_i) dh_i, \end{aligned} \tag{13}$$

where M is the number of patches extracted from l. The above expression can be written in terms of the distribution of the noise. For concreteness, if n_i is normally distributed with zero mean and covariance $\sigma^2 \boldsymbol{I}$, then Eq. (13) becomes

$$\arg\max_k p(k) \prod_{i=1}^{M} \int_h \exp\left\{-\frac{\|q_i - \boldsymbol{K}h\|^2}{2\sigma^2}\right\} p(h) dh, \tag{14}$$

where we used the fact that $\{h_i\}$ are identically distributed. In blind deblurring, *parametric* priors $p(h)$, which promote sparsity of image gradients, are often used [13, 14]. In many of these parametric families, the integral in Eq. (14) cannot be calculated in closed form, thus calling for approximate solutions [14]. In contrast, we approximate $p(h)$ in a non-parametric way using the training patches $\{r_j\}$.

Our key observation is that the integral in Eq. (14) can be written in terms of an expectation over h, so that the objective in (14) may be expressed as

$$\max_k p(k) \prod_{i=1}^{M} \mathbb{E}_h \left[\exp\left\{-\frac{\|q_i - \boldsymbol{K}h\|^2}{2\sigma^2}\right\}\right]. \tag{15}$$

We approximate this expectation by an *empirical mean* (sample mean) over all high-res training patches $\{r_i\}$

$$\max_k p(k) \prod_{i=1}^{M} \frac{1}{N} \sum_{j=1}^{N} \exp\left\{-\frac{\|q_i - \boldsymbol{K}r_j\|^2}{2\sigma^2}\right\}. \tag{16}$$

Under our assumption on $\{r_j\}$, this approximation is guaranteed to tend to the MAP$_k$ solution as N tends to infinity. Therefore, for a large enough collection of training patches, Eq. (16) is practically equivalent to the MAP$_k$ criterion.

To emphasize the dependence of Eq. (16) on k we note that the term $\boldsymbol{K}r_j$ can be equivalently written as $\boldsymbol{R}_j k$, where k is the column-vector representation of the kernel and $\boldsymbol{R}_j$ is a matrix corresponding to convolution with r_j and sub-sampling by α. Furthermore, a reasonable prior for k is $p(k) = c \cdot \exp\{-\|\boldsymbol{C}k\|^2/2\}$, where $\boldsymbol{C}$ can be chosen, *e.g.* to penalize for non-smooth kernels. With these substitutions (and taking the log), Eq. (16) is equivalent to

$$\min_k \frac{1}{2}\|\boldsymbol{C}k\|^2 - \sum_{i=1}^{M} \log\left(\sum_{j=1}^{N} \exp\left\{-\frac{\|q_i - \boldsymbol{R}_j k\|^2}{2\sigma^2}\right\}\right). \tag{17}$$

The intuition behind this optimization problem is simple. A kernel k achieving good score is such that if it is used to down-sample the training patches in the database, then each of our query patches $\{q_i\}$ should find as many good nearest neighbors (NNs) as possible.

Minimizing (17) can be done in an iterative manner, whereby in each step we down-sample the training patches using the current estimate $\hat{k}$ of k, find nearest neighbors to the query patches $\{q_i\}$, and update $\hat{k}$ by solving a weighted least-squares problem. More specifically, setting the gradient of (17) to zero, leads to a version of the iteratively re-weighted least squares (IRLS) procedure, as outlined in Alg. 1. Each equation induced by an NN r^α of a low-res patch q is weighted by the degree of similarity between r^α and q. Note that in practice, only those patches in the database whose distance to q is small (not much larger than σ) are assigned non-negligible weights, so that the number of required NNs per low-res patch is typically small. This interpretation provides a more concrete methodology for implementing the general approaches of Secs. 3 and 4.

[4] This is a reasonable assumption if the query patches $\{q_i\}$ are extracted from distant locations in the low-res image l.

6. Experimental results

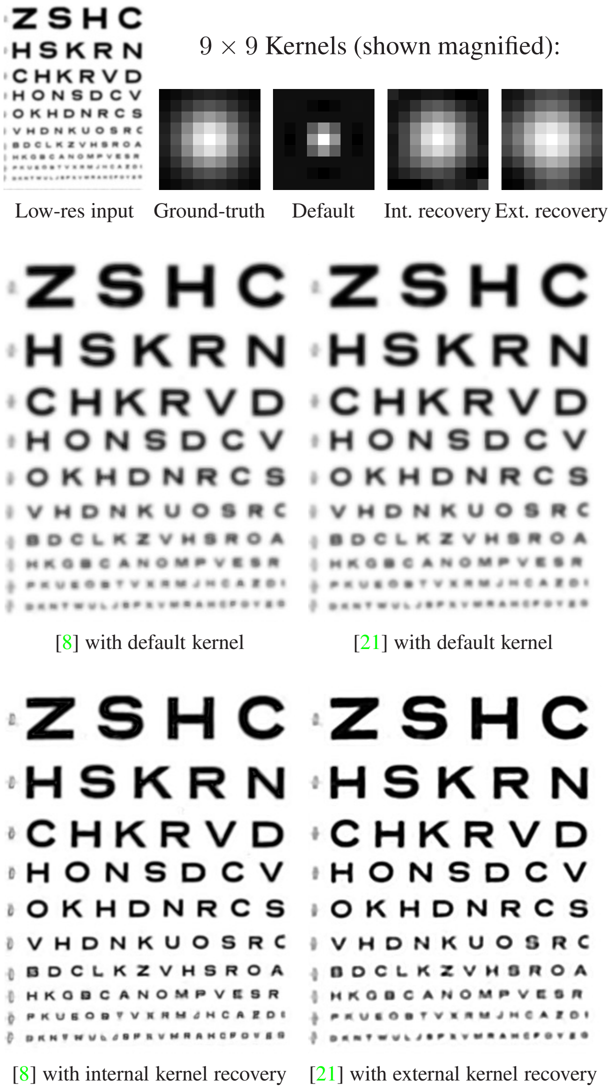
We validated the benefit of using our kernel estimation in SR algorithms both empirically (on low-res images generated with ground-truth data), as well as visually on real images. We use the method of [8] as a representative of SR methods that rely on internal patch recurrence, and the algorithm of [21] as representative of SR methods that train on an external database of examples.

In our experiments the low-res patches q and r^{α} were 5×5 patches. We experimented with SR×2 and SR×3. We typically solve for 9×9, 11×11 or 13×13 kernels. The noise level was assumed to be $\sigma = 10$. For the external kernel recovery we used a database of 30 natural images downloaded from the Internet (most likely captured by different cameras). The regularization in the least-squares stage of each iteration (see Alg. 1) was performed with a matrix $\boldsymbol{C}$ corresponding to derivatives in the x and y directions.

To quantify the effect of our estimated kernel $\hat{k}$ on SR algorithms, we use two measures. The *Error Ratio to Ground Truth **(ERGT)***, which was suggested in the context of blind deblurring [13], measures the ratio between the SR reconstruction error with our $\hat{k}$ and the SR reconstruction error with the ground-truth k, namely ERGT $= \|h - \hat{h}_{\hat{k}}\|^2 / \|h - \hat{h}_k\|^2$. Values close to 1 indicate that the estimated kernel $\hat{k}$ is nearly as good as the ground-truth k. The *Error Ratio to Default **(ERD)*** measure quantifies the benefit of using the estimated kernel $\hat{k}$ over the default (bicubic) kernel k_{d}, and is defined as ERD $= \|h - \hat{h}_{\hat{k}}\|^2 / \|h - \hat{h}_{k_{\mathrm{d}}}\|^2$. Values much smaller than 1 indicate a large improvement.

Fig. 5 shows that plugging our recovered kernel $\hat{k}$ into the SR methods of [8] and [21] leads to substantial improvement in the resulting high-res image over using their assumed default (bicubic) kernel. Note that the recovered kernels are much wider than the default one. The differences in the resulting high-res images are observed both visually, and numerically when compared to the ground-truth image. Indeed, [8] with internal kernel recovery achieves ERD $= 0.45$ and ERGT $= 1.02$ and [21] with external kernel recovery achieves ERD $= 0.43$ and ERGT $= 1.01$.

Fig. 6 shows the convergence of the internal kernel estimation algorithm applied to the low-res input image of Fig. 5. As can be seen in Fig. 6(a), the RMSE between patches in the image and their NNs in the coarser scale rapidly decreases. This demonstrates that our algorithm indeed maximizes the similarity of recurring patches across scales, as expected. Moreover, the ERGT of the SR reconstruction (Fig. 6(b)) decreases with each iteration towards 1. This implies that the estimated kernel converges to SR performance of the ground-truth kernel.

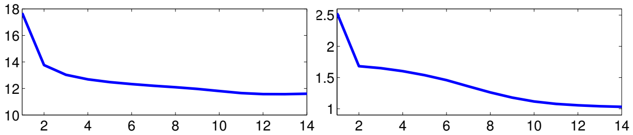

9 × 9 Kernels (shown magnified):

Low-res input Ground-truth Default Int. recovery Ext. recovery

[8] with default kernel [21] with default kernel

[8] with internal kernel recovery [21] with external kernel recovery

Figure 5: ***SR**×2 **with default vs. recovered kernel.*** *Please magnify to compare the letters at the bottom few rows.*



(a) NN error vs. iterations (b) Relative SR reconstruction error (ERGT) vs. iterations

Figure 6: ***Convergence of the algorithm.***

Fig. 7 demonstrates that our iterative approach is fairly insensitive to the initialization of the kernel. In this experiment, the image on the left was obtained by down-sampling a large image by a factor of 3 with a Gaussian kernel. We ran 30 iterations of both the internal and the external schemes, each time with a different initial kernel (green). These include a delta function, as well as randomly sampled kernels. As can be seen, both schemes consistently converged to kernels (red) which are close to the ground-truth kernel, regardless of the initial guess.

To empirically validate our algorithm, we performed a large statistical analysis on hundreds of images with randomly varying ground-truth kernels. Fig. 8 shows the er-

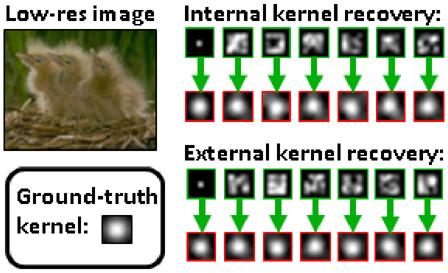


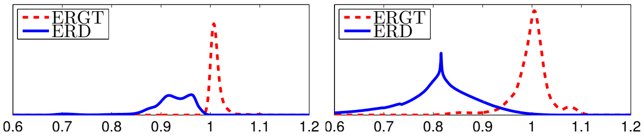
Figure 7: ***Insensitivity of the algorithm to the initial guess of the kernel.*** *Green marks the initial guesses. Red marks final recovered kernels.*

rors incurred by using our recovered kernels within the SR methods of [8, 21] (using their implementations). In this experiment, we down-sampled 100 test images from the Berkeley Segmentation Dataset by a factor of 3, each downsampled with several 7×7 kernels. The kernels were *drawn randomly* from the prior on k assumed in Sec. 5 (with the matrix $\boldsymbol{C}$ corresponding to derivatives in the x and y directions), *i.e.* randomly sampled smooth kernels. The plots depict the distribution of the errors. As can be seen, SR with our estimated kernel performs similarly to SR with the ground-truth kernel (ERGT $\approx$ 1) and better than with the default kernel (ERD $<$ 1). The improvement over the default kernel is more significant in the method of [21]. Surprisingly, sometimes the estimated kernels produce better results in [21] than the ground-truth kernel (ERGT $<$ 1). From our experience, this can be attributed to the fact that the algorithm of [21] (as opposed to [8]) does not enforce that the recovered high-res image h be consistent with the low-res image l (namely, that $l = (k * \hat{h}) \downarrow_{\alpha}$). When we introduced this consistency constraint (using back-projection), their recovery both with the estimated kernel and with the ground-truth kernel typically improved substantially, but more so with the ground-truth kernel (in which case the ERGT increases to approx. 1).

Figs. 1 and 9 show results of using our internal recovery scheme on real images. Fig. 1 – an old low-quality image downloaded from the internet; Fig. 9 – images taken by an iPhone camera. In all cases, SR with our estimated kernel is visually superior to SR with the default kernel. Note the bicycles, the tree and the patterns on the walls in Fig. 9. Note the heads of people, the white star on the tank and the bars on the window in Fig.1. The recovered kernels suggest that the original low-res images suffered from slight motion blur and defocus. For more examples and full-sized images, see `www.wisdom.weizmann.ac.il/~vision/BlindSR.html`.

7. Summary

We showed that contrary to the common belief, the PSF of the camera is the wrong blur kernel k to use in SR algorithms. We then showed that the correct k can be recovered directly from the low-res image, by exploiting the recurrence of small image patches (either at coarser scales of the same image, or in an external database of images). This is shown to be a principled MAP estimation of k, and leads to a significant improvement in SR results.



[8] with internal kernel recovery

[21] with external kernel recovery

Figure 8: ***Error distribution for SR with [8, 21] using internal and external kernel recovery*** *(statistics collected on hundreds of images – see text).*

Acknowledgements *Thanks to S. Kovalsky, M. Zontak and M. Galun for their helpful comments. Funded in part by the Israel Science Foundation, Israel Ministry of Science, Citigroup Foundation and a Viterbi Fellowship (Technion).*

References

[1] I. Begin and F. R. Ferrie. Blind super-resolution using a learning-based approach. In *ICPR*, 2004.

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

[3] Y. C. Eldar and T. Michaeli. Beyond bandlimited sampling. *IEEE Signal Process. Mag.*, 26(3):48–68, 2009.

[4] R. Fattal. Image upsampling via imposed edge statistics. *ACM TOG*, 26(3):95, 2007.

[5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM TOG*, 25(3):787–794, 2006.

[6] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM TOG*, 30(2):12, 2011.

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

[8] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009.

[9] S. Harmeling, S. Sra, M. Hirsch, and B. Scholkopf. Multi-frame blind deconvolution, super-resolution, and saturation correction via incremental EM. In *ICIP*, 2010.

[10] Y. He, K. H. Yap, L. Chen, and L. P. Chau. A soft MAP framework for blind super-resolution image reconstruction. *Image and Vision Computing*, 27(4):364 – 373, 2009.

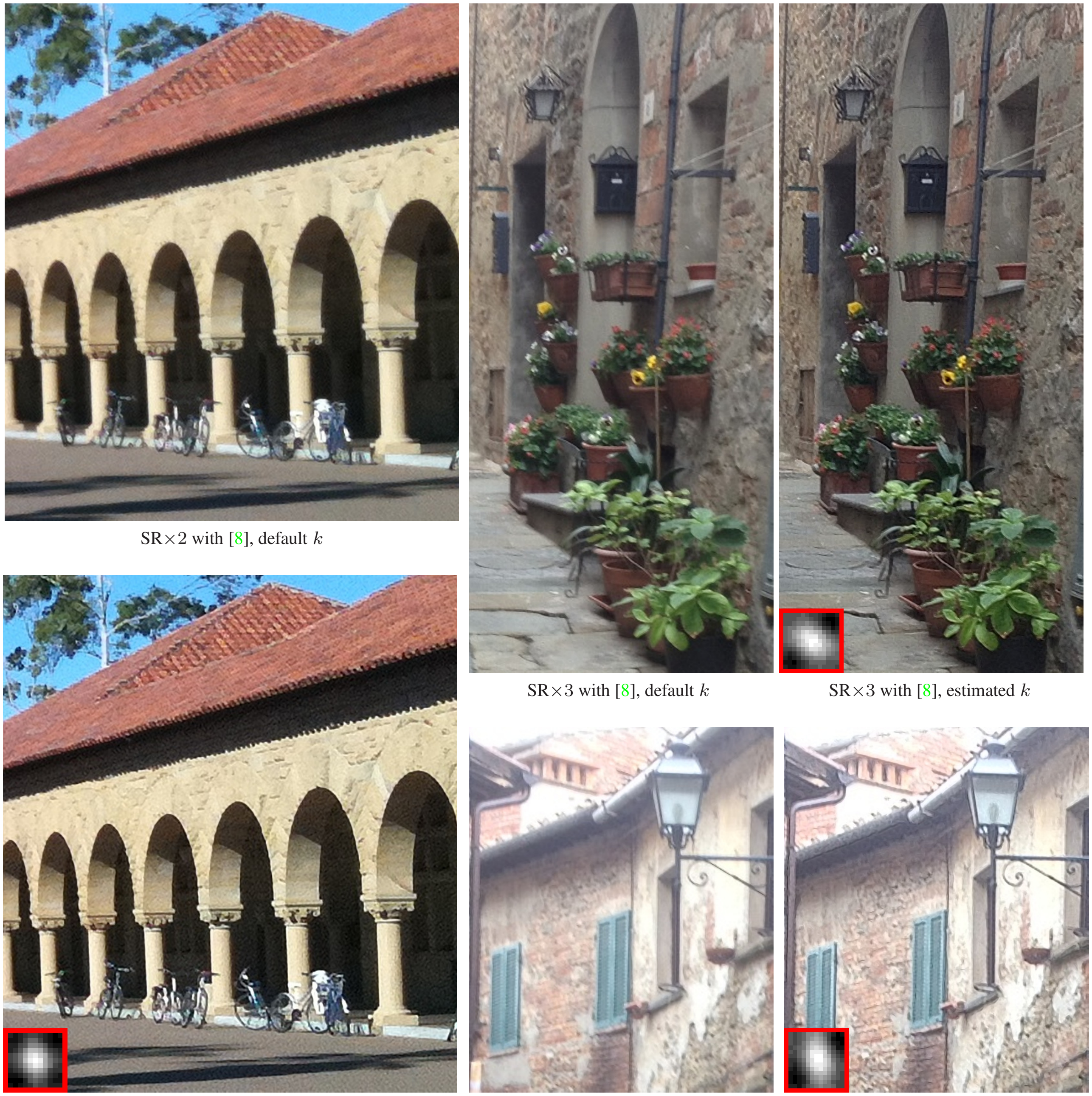
[11] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP*, 53(3):231–239, 1991.

[12] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, 2008.

[13] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009.

[14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, 2011.

[15] S. Mallat and G. Yu. Super-resolution with sparse mixing estimators. *Trans. Signal Proc.*, 19(11):2889–2900, 2010.



SR×2 with [8], default k

SR×3 with [8], default k

SR×3 with [8], estimated k

SR×2 with [8], estimated k

SR×3 with [8], default k

SR×3 with [8], estimated k

Figure 9: ***SR on images taken by an iPhone camera.*** *Note the the bicycles, the tree and the patterns on the walls. Recovered kernels (marked in red at the bottom-left of the images) are* 11×11 *(left image) and* 13×13 *(right images).*

[16] P. Milanfar, editor. *Super-resolution imaging*. CRC Press, 2010.

[17] J. Qiao, J. Liu, and C. Zhao. A novel SVM-based blind super-resolution algorithm. In *International Joint Conference on Neural Networks*, 2006.

[18] F. Sroubek, G. Cristobal, and J. Flusser. Simultaneous super-resolution and blind deconvolution. In *Journal of Physics: Conference Series*, volume 124, 2008.

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

[20] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008.

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

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