Accurate Blur Models vs. Image Priors in Single Image Super-Resolution

Netalee Efrat, Daniel Glasner, Alexander Apartsin, Boaz Nadler, Anat Levin
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science,ISRAEL

Abstract

Over the past decade, single image Super-Resolution (SR) research has focused on developing sophisticated image priors, leading to significant advances. Estimating and incorporating the blur model, that relates the high-res and low-res images, has received much less attention, however. In particular, the reconstruction constraint, namely that the blurred and downsampled high-res output should approximately equal the low-res input image, has been either ignored or applied with default fixed blur models. In this work, we examine the relative importance of the image prior and the reconstruction constraint. First, we show that an accurate reconstruction constraint combined with a simple gradient regularization achieves SR results almost as good as those of state-of-the-art algorithms with sophisticated image priors. Second, we study both empirically and theoretically the sensitivity of SR algorithms to the blur model assumed in the reconstruction constraint. We find that an accurate blur model is more important than a sophisticated image prior. Finally, using real camera data, we demonstrate that the default blur models of various SR algorithms may differ from the camera blur, typically leading to oversmoothed results. Our findings highlight the importance of accurately estimating camera blur in reconstructing raw low- res images acquired by an actual camera.

1. Introduction

Single image Super-Resolution (SR) is the problem of estimating a High-Resolution (HR) image x from a single input Low-Resolution (LR) image y. Mathematically, the relation between these two images is typically modeled (in vectorized form) as a linear transformation

$$y = Ax + n \, , \tag{1}$$

where n denotes imaging noise and the matrix A encodes the processes of blurring and downsampling. Recovering x from y involves two non-trivial challenges. First, since the matrix A has far fewer rows than columns, the SR problem is under-constrained, with an infinite number of solutions. Thus, to recover a visually pleasing HR image, SR algorithms typically employ some image prior. The second challenge is to enforce the *reconstruction constraint* $Ax \approx y$, which implies that up to imaging noise, the recovered HR image should be consistent with the LR input. This constraint requires knowledge of the matrix A, which, in turn, relies on an accurate estimation of the camera's optical blur.

Developing sophisticated image priors has been the focus of much single image SR research in the past decade, with many significant successes [7, 20, 23, 8, 6, 9, 21, 27, 25, 5, 19, 22, 1, 8, 13]. In contrast, the reconstruction constraint has received relatively little attention. Some algorithms do not enforce $Ax \approx y$ at all. Those that do often assume a predefined blur kernel. Examples include antialiasing with bicubic interpolation (Matlab's default *imresize* function) [8, 27], Gaussian blur [3], Gaussian blur followed by bicubic interpolation [7], simple pixel averaging [5], and sampling without any pre-smoothing [15]. A critical concern in applying such SR algorithms to real images is how well these synthetic forward models approximate real camera blur. For example, the bicubic interpolation used by many algorithms is generally not physically feasible, as it involves negative weights. Furthermore, in most SR algorithms, the blur kernel is not an input parameter: it is coupled to various internal components which are not easily adjusted. A few single-image SR works which do attempt to estimate or take the unknown kernel into account include [3, 24, 17, 10, 11, 12].

This state of affairs naturally raises the following questions, which are the focus of our paper: i) what is the effect of an incorrect blur model on SR algorithms? ii) what is the importance of the reconstruction constraint compared to that of the image prior?

We make several contributions towards answering these questions. First, we argue that the reconstruction constraint is at least as important as the image prior. In particular, we demonstrate that combining a simple prior, an L_2 penalty on image gradients, with an accurate reconstruction constraint, provides SR results almost as good as those produced by state-of-the-art SR algorithms with sophisticated priors.

Second, we empirically examine the sensitivity of several SR algorithms to the accuracy of the estimated blur kernel. We show that incorporating an accurate estimate into these algorithms improves their output, by allowing them to take full advantage of the reconstruction constraint. In contrast, when the SR algorithms utilize an inaccurate blur kernel, the resulting images are either too blurred or contain over-sharpening artifacts. These results are consistent with results from previous work on performance bounds in multi-frame SR [2, 18]. For the L_2 prior case, we also present a theoretical analysis explaining these phenomena, via a frequency analysis of the kernel mismatch.

Finally, we demonstrate the importance of accurately estimating camera blur when applying SR to raw images captured with a real camera. We show that the default kernels used by many algorithms are not sufficiently close to the camera blur, and produce over-smoothed results. Moreover, we show that incorporating a more accurate estimate of the camera blur improves the results. Our findings highlight the importance of modeling and estimating the camera blur, a topic which has not received much attention in the context of single-image SR.

2. Problem Setup

Let x be a HR image of size $n_1 \times n_2$ pixels, and let y be its corresponding LR image of size $(n_1/s) \times (n_2/s)$, where $s > 1$ is the downsampling factor. The relation between x and y is typically expressed as

$$y = k * x \downarrow^s + n \tag{2}$$

where k denotes a blur kernel (low pass filter), $\downarrow^s$ denotes subsampling by factor s, and n is imaging noise. Further details of this model can be found in [2]. In vectorized form, we can express the above relation as a linear transformation

$$y = Ax + n \tag{3}$$

where A is a matrix of size $(n_1 \cdot n_2/s^2) \times (n_1 \cdot n_2)$, y, n are $(n_1 \cdot n_2/s^2) \times 1$ vectors and x is $(n_1 \cdot n_2) \times 1$.

The SR problem, recovering x from y, poses two challenges. First, we need to know A, that is, to have an accurate estimate of the blur k. Second, the linear system in Eq. (3) is underconstrained. To recover a visually plausible x, it is thus common practice to use some natural image prior.

Perhaps the simplest prior is a penalty on the image gradients. It leads to the following optimization problem, similar to total variation [1],

$$\hat{x} = \arg\min_x \|Ax - y\|^2 + \lambda \sum_i (\rho(g_{h_i}(x)) + \rho(g_{v_i}(x))) \tag{4}$$

where $g_{h_i}(x), g_{v_i}(x)$ denote horizontal and vertical derivatives at the i-th pixel, and ρ is a penalty function. For example, $\rho(z) = |z|^2$ gives a Gaussian (L_2) prior, while $\rho(z) = |z|^\alpha$ for $\alpha \leq 1$ yields a sparse prior.

Most modern SR algorithms are not expressed explicitly as the minimizer of a functional as in Eq. (4), for example when their prior is non-parametric. Moreover, some methods are patch-based and either do not enforce the global reconstruction constraint $Ax \approx y$ at all, or apply it only in a separate post-processing step, with a default blur kernel.

In this paper we study, both qualitatively and quantitatively, the importance of an accurate blur model and the corresponding reconstruction constraint in SR algorithms. We start with an empirical evaluation in Sec. 3 and provide a theoretical analysis in Sec. 4. Additional results to the ones presented in the paper, and a supplementary file, are available online at our projct page [4].

3. Empirical Evaluation of SR Algorithms

Evaluation Metric: As our goal is to evaluate various aspects of SR, it is important to distinguish between two different SR problems: i) the computer graphics problem of image *hallucination* – the addition of visually plausible HR details, even if these are not actually present; and ii) the task of image *reconstruction* – the recovery of accurate data for some application, such as seismic measurements, or car license plates captured by a security camera.

Since visual plausibility is somewhat subjective, we focus on the reconstruction task, which can be evaluated numerically against ground truth. We adopt two measures, PSNR and SSIM. While the correlation between numerical and visual measures of success is not perfect, the former method is useful to objectively compare various SR algorithms and their variants.

To numerically evaluate an SR algorithm, we calculate its Mean Squared Error (MSE) over all test image pixels,

$$\text{MSE} = \frac{1}{N} \sum_{\text{HR images } x} \sum_i |x^0(i) - x(i)|^2 , \tag{5}$$

where $x(i), x^0(i)$ are the i-th pixel in the HR image and in the algorithm output, respectively. Scaling image intensity to be in the $[0, 1]$ range, we then compute PSNR $= -10\log_{10}$ MSE, and SSIM as described in [26]. To avoid boundary artifacts, for both measures we discarded all pixels at distance up to 60 pixels from the image boundary.

Experimental Setup: We performed experiments on images synthetically downsampled with a known kernel, as well as on raw LR images acquired by a real camera.

First, we compared several SR algorithms on the same set of HR/LR image pairs, using 50 test images from the Berkeley Segmentation Dataset [16] (BSDS). Each LR image was synthesized from a corresponding HR image, thus providing a ground truth solution. The algorithms we considered are: Freeman *et al.* [7], Kim&Kwon [13], and Yang *et al.* [27] whose implementations are available online, as well as Glasner *et al.* [8], whose code was kindly provided by the authors. In addition, we also tested the simple gradient regularization approach of Eq. (4) with both a Gaussian

$\eta = 10^{-2}$	$k_A =$ camera	$k_A = b$	$k_A = b * g_1$	$k_A = b * g_2$	original
$k_T =$ camera	**26.774 / 0.787**	26.115 / 0.779	26.739 / 0.785	23.312 / 0.702	25.956 / 0.780
$k_T = b$	26.130 / 0.805	**27.481** / 0.835	25.570 / 0.790	20.664 / 0.646	27.329 / **0.837**
$k_T = b * g_1$	26.111 / 0.759	25.549 / 0.751	**26.294 / 0.764**	23.361 / 0.698	25.494 / 0.756
$k_T = b * g_2$	23.281 / 0.594	22.991 / 0.597	23.272 / 0.591	**24.079 / 0.642**	23.031 / 0.609
$\eta = 10^{-4}$	$k_A =$ camera	$k_A = b$	$k_A = b * g_1$	$k_A = b * g_2$	original
$k_T =$ camera	**27.675 / 0.871**	26.447 / 0.824	26.136 / 0.836	14.232 / 0.390	26.138 / 0.807
$k_T = b$	25.411 / 0.846	**27.902 / 0.877**	21.186 / 0.709	11.599 / 0.281	27.578 / 0.863
$k_T = b * g_1$	27.000 / 0.845	25.815 / 0.793	**27.675 / 0.864**	15.904 / 0.462	25.654 / 0.782
$k_T = b * g_2$	23.586 / 0.663	23.101 / 0.633	23.600 / 0.662	**25.529 / 0.751**	23.119 / 0.634

Table 1. **Kernel sensitivity of Glasner et al [8].** Rows indicate the kernel with which the image was downsampled. Columns correspond to the kernel the algorithm assumes. The rightmost column presents results of the unmodified algorithm with its default bi-cubic kernel. Noise level $\eta = 0.01$ on the top and $\eta = 0.0001$ on the bottom. In each cell the first number is a PSNR value and the second is SSIM.

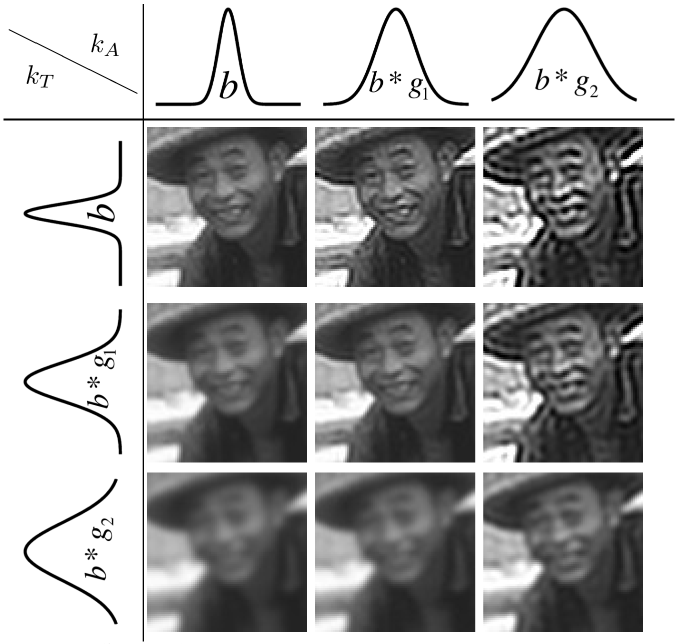

Figure 1. **Sensitivity to kernel mismatch.** On the diagonal, the algorithm uses $k_A = k_T$. On the upper right off-diagonal images, the assumed kernel is smoother than the true one, leading to over sharpening artifacts. For the lower left, the assumed kernel is sharper than the correct one, leading to over-smoothed results.

and a sparse prior. The sparse prior was optimized using the iterative re-weighted least squares algorithm [14], with parameters chosen using a separate training set.

We denote by k_T the true kernel used to synthesize a test LR image and by k_A the kernel assumed by a SR algorithm. To test the role of different kernels we prepared 4 sets of test images, each set blurred synthetically with a different kernel k_T: 1) antialiased bicubic interpolation $k_T = b$, (Matlab's *imresize*); 2-3) Gaussian smoothing followed by bicubic interpolation, denoted as $k_T = b * g_1, k_T = b * g_2$, where g_ℓ is a Gaussian with std of ℓ pixels; 4) a real camera blur kernel, estimated by capturing a known calibration target with a Canon 5D Mark II camera, shown in Fig. 2. We added to the test images Gaussian zero mean i.i.d. noise with variance η^2. We tested both realistic and nearly noise-free images with $\eta = 0.01$ and $\eta = 0.0001$.

Our experiments study what happens when the two kernels k_T, k_A mismatch. Furthermore, if an algorithm is given an exact estimate of the kernel, can it be modified to take this into account and what are the benefits of doing so?

The algorithms of Yang, Kim, and Glasner use as default the bicubic kernel, $k_A = b$, whereas Freeman uses $k_A = b * g_1$. Unfortunately, these algorithms do not accept a kernel as one of their input parameters and adjusting them to use a different kernel is not straightforward. For example, Glasner *et al*. upsample the image in gradual steps, and defining a kernel for each intermediate step is non-trivial.

As a simple modification, we introduced two kernel-dependent changes to the algorithms. First, (if applicable) the LR/HR training patch pairs were prepared with a desired blur. Second, following [27], we introduced a reconstruction constraint with the desired kernel in post-processing. To this end, we found the image x^1 which minimizes

$$x^1 = \arg\min_x \lambda \|Ax - y\|^2 + \|x - x^0\|^2 \qquad (6)$$

where x^0 is the original algorithm output[1]. The weighting factor λ was selected to minimize the empirical error on a separate set of training images. Eq. (6) is not needed for the simple regularization algorithms which already optimize the reconstruction constraint in Eq. (4).

3.1. Evaluation Results

Kernel sensitivity: Table 1 reports numerical results, with the method of Glasner *et al*., on a set of 4×5 SR experiments. SR was applied to the 4 test sets (prepared with different kernels k_T), each time adjusting the algorithm to use a different kernel k_A in reconstruction. The fifth column shows the original authors' results, unmodified, using the default bicubic kernel.

[1] The algorithms of Yang *et al*. and of Glasner *et al*. already include a back-projection process which minimizes a constraint similar to Eq. (6). We replaced their back-projection procedures with Eq. (6), where x^0 is their output before the final back-projection stage. Note that Eq. (6) can modify an algorithm's results, even when applied with its default kernel.

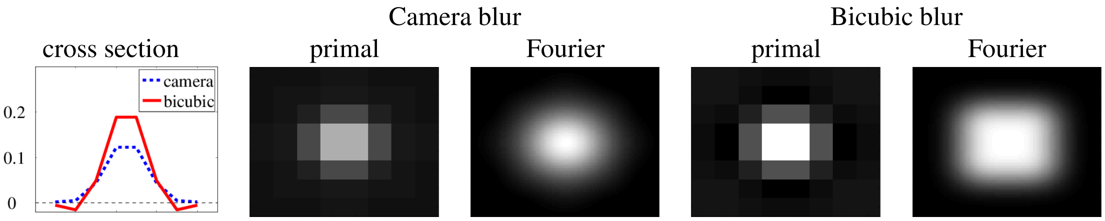

Figure 2. **Estimated camera blur vs. bicubic kernel**. Despite the seemingly similar shape, the kernels' frequency content is different. The camera blur attenuates high frequencies more than the bicubic one. This highlights the importance of accurate kernel estimation.

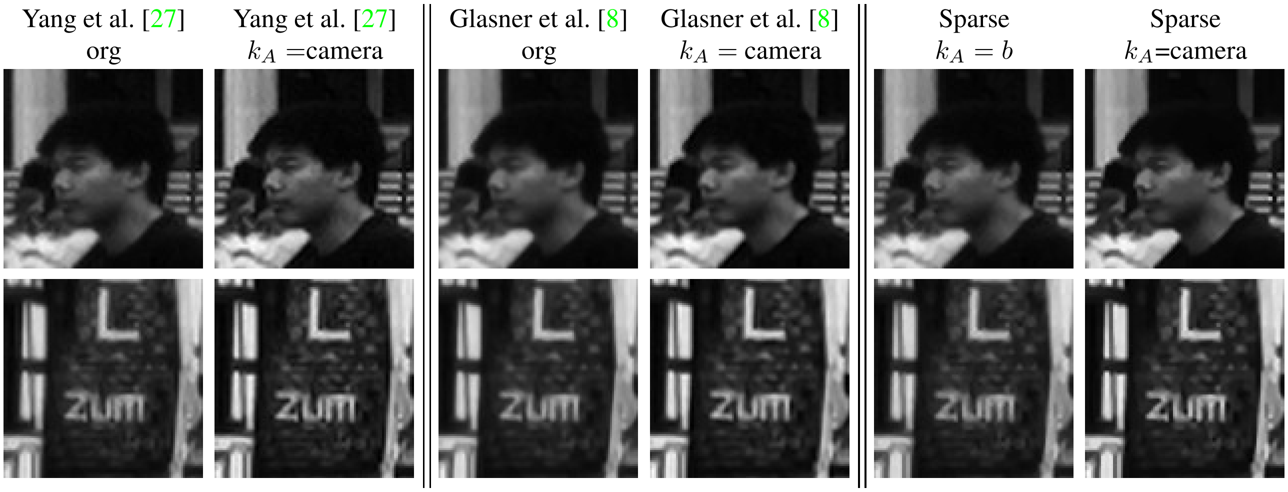

Figure 3. **Results on real images:** The default implementation of SR algorithms (assuming a bicubic kernel) produces over-smooth results. Sharpness is improved by adjusting the algorithm to incorporate the camera kernel. Images embedded at 96PPI.

This table leads us to several observations. First, incorporating the reconstruction constraint with the true kernel improves accuracy. One can see this by comparing the fifth column (the original algorithm with its default kernel), with the diagonal entries, for which the reconstruction constraint uses $k_A = k_T$. Similar trends hold for all other algorithms, as shown in [4]. Moreover, SR using an incorrect kernel drastically increases reconstruction error. The diagonal entries in Table 1, in terms of both PSNR and SSIM, are larger than the off-diagonal ones. In particular, when the assumed kernel is smoother than the true kernel, the recovered image is blurred. On the other hand, when the assumed kernel is sharper than the true kernel, high frequency ringing artifacts appear, as illustrated in Fig. 1. Though the first type of error may be less visually disturbing than the second, if we are aiming to increase resolution, over-smoothing constitutes an equally undesirable effect. We provide a theoretical analysis of these observations in Sec. 4.

Real images: Given the sensitivity of SR algorithms to the assumed blur kernel, it is interesting to assess their performance on raw LR images acquired by an actual camera. To this end, we captured images with a Canon 5D Mark II camera, and estimated its blur using a known calibration target (calibration details can be found in [4]). Fig. 2 compares our estimated camera blur with the bicubic kernel. Despite their seemingly similar shapes in the primal domain, their frequency content is quite different. As seen in the Fourier domain, the camera kernel attenuates high frequencies more than the bicubic one.

Our theoretical analysis (Sec. 4) predicts that using the sharper bicubic kernel will result in over-smoothed SR images. Fig. 3 indeed demonstrates this on actual camera images (for additional images, see [4]). The default implementation of various algorithms that assume a bicubic kernel indeed yields over-smoothed results, while adjusting the algorithms to incorporate the camera kernel sharpens them. This highlights the sensitivity of SR algorithms to correctly modeling the kernel. Such modeling is a promising venue for future improvements, if SR is to be applied to real camera data.

Image prior vs. reconstruction constraint: Next, we consider the relative importance of the assumed image prior vs. the reconstruction constraint, with the correct kernel. To this end, Table 2 compares all algorithms[2] and their modified versions, which incorporate the reconstruction constraint, on two test sets - blurred with $k_T = b$ and with $k_T = b * g_1$. As rows 2 and 4 show, *using simple gradient*

[2] For a fair comparison, where applicable, all algorithms were trained on the same separate training dataset.

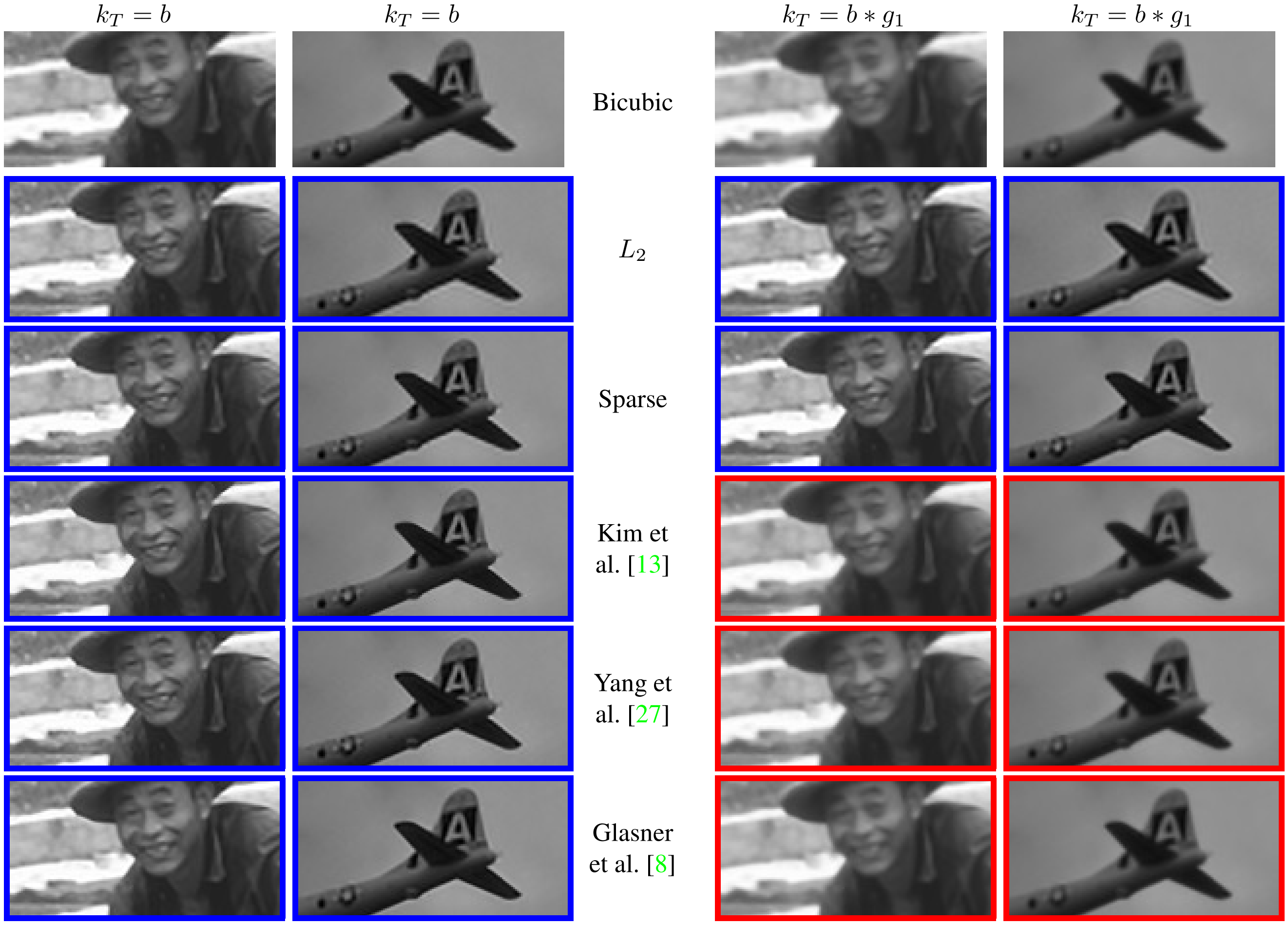

Figure 4. **Image prior vs. reconstruction constraint.** The left columns show SR results of various algorithms on LR images downsampled with $k_T = b$. Columns on the right show results on images downsampled with $k_T = b * g_1$. All LR images were corrupted by noise at level $\eta = 0.0001$. One can see the of a more sophisticated prior by comparing rows 2-3 to rows 4-6 in the left columns. Comparing the same rows in the right columns shows the effect of using the correct blur kernel instead of the default one. $k_A = k_T$ images are marked in blue, $k_A \neq k_T$ in red. The effect of using the exact blur kernel is more dominant than that of the prior. Images embedded at 96PPI.

regularization produces results which are roughly comparable to sophisticated SR algorithms, both in terms of PSNR and SSIM values. The first 2 columns of Fig. 4 show representative visual results, when all algorithms use the true kernel. While visual comparison is somewhat subjective and one may argue in favor of one algorithm or the other, overall the results of all algorithms (except the baseline bicubic interpolation) are not significantly different.

Second, the influence of different image priors is much smaller than the effect of kernel mismatch. This can be seen from Table 2. Contrast the changes along a row (0.1dB-0.5dB), which correspond to different image priors, to the difference of almost 2dB between the third and fourth rows, which capture the effect of using a correct kernel instead of the default bicubic one. Visually, this is seen by contrasting the changes along the rows of Fig. 4. In the two left columns all algorithms use the true kernel and hence produce comparable results. In contrast, in the two right columns, only the sparse and L_2 algorithms use the correct kernel, others use their default one. This kernel mismatch leads to inferior results. This emphasizes that an accurate reconstruction constraint can be more important than a sophisticated prior.

The fact that simple regularization give results comparable to those of sophisticated priors seems surprising at first sight. One possible explanation relates back to the reconstruction constraint. Example-based algorithms process the image locally using small patches. While each LR/HR patch pair satisfies the reconstruction constraint, this property is not retained when the patches are fused into a global solution. Since this is not a trivial task most algorithms only impose the global reconstruction constraint in post-processing. In contrast, simple gradient regularization methods explicitly optimize a functional which jointly accounts for the global reconstruction constraint and the prior.

	L_2	Sparse	Freeman [7]	Kim [13]	Yang [27]	Glasner [8]
$k_T = b$, original			25.16 / 0.77	27.83 / 0.87	27.45 / 0.87	27.57 / 0.86
$k_T = b$, RC with $k_A = b$	27.47 / 0.87	27.68 / 0.87	26.78 / 0.84	27.90 / 0.87	27.39 / 0.86	27.90 / 0.88
$k_T = b * g_1$, original			24.81 / 0.73	25.82 / 0.79	25.73 / 0.79	25.65 / 0.78
$k_T = b * g_1$, RC with $k_A = b * g_1$	27.50 / 0.86	27.77 / 0.87	27.03 / 0.84	27.68 / 0.86	27.56 / 0.86	27.67 / 0.86

Table 2. **Kernel accuracy is more important than choice of image prior.** The original implementation of each algorithm with its default kernel is compared with a modified one which accepts a kernel as a parameter, when applicable learns a dictionary with it, and also enforces the reconstruction constraint (RC). In the first two rows the image was downsampled with $k_T = b$, in the bottom rows with $k_T = b * g_1$. In each cell the first number is a PSNR value and the second is SSIM. Most algorithms use $k_A = b$ as default, thus in the 3rd row algorithms process images with $k_A \neq k_T$ and in the 4th row adjust to the correct kernel $k_A = k_T$. The difference between the 3rd and 4th rows is much larger than the gap between different algorithms on the same row.

4. Theoretical Analysis

To gain further insight into the detrimental effect of kernel mismatch, let us examine SR with a simple L_2 gradient regularization (Gaussian prior) as presented in Eq. (4). For notational simplicity we consider SR of 1D signals with a fixed upsampling factor $s = 2$.

We first rewrite the problem in the frequency domain. Neglecting image boundaries, convolution $k * x$ translates to multiplication, and sampling translates to aliasing of the HR signal. Denoting Fourier transforms by capital letters, we can express Eq. (2) as

$$Y_\omega = K_{T,\omega} X_\omega + K_{T,\omega'} X_{\omega'} + N_\omega \ , \tag{7}$$

where the discrete HR signal has a frequency range of $[0, \Omega]$, the LR one $[0, \Omega/2]$, and $\omega' = \omega + \Omega/2$ denotes the aliased replica of the frequency ω.

Recall that if K_T is an ideal low pass filter ($K_{T,\omega'} = 0$ for all $\omega' > \Omega/2$), the second term in Eq. (7) disappears. Otherwise, at all frequencies where $|K_{T,\omega'}| > 0$, the LR observation Y_ω includes aliasing.

SR algorithms that assume a kernel K_A in fact assume that Y and X are related via

$$Y_\omega = K_{A,\omega} X_\omega + K_{A,\omega'} X_{\omega'} + N_\omega \ . \tag{8}$$

A standard approach to reconstructing the signal X from the measurements in Eq. (7) is via MAP estimation. The following lemma characterizes the resulting estimator and the relation between the estimated signal $\hat{X}$ and the true signal X in case of kernel mismatch ($K_A \neq K_T$).

Lemma 1 *Let Y be given by Eq. (7), with i.i.d. zero mean Gaussian noise of variance η^2. The MAP SR estimate $\hat{X}$, assuming a Gaussian prior on X and a kernel K_A as in, Eq. (8), can be written as*

$$\begin{pmatrix} \hat{X}_\omega \\ \hat{X}_{\omega'} \end{pmatrix} = H_{A,\omega} \begin{pmatrix} \frac{K_{T,\omega}}{K_{A,\omega}} X_\omega \\ \frac{K_{T,\omega'}}{K_{A,\omega'}} X_{\omega'} \\ N_\omega \end{pmatrix} , \tag{9}$$

where

$$\begin{aligned} H_{A,\omega} \quad = \quad & \frac{1}{\sigma_\omega^2 |K_{A,\omega}|^2 + \sigma_{\omega'}^2 |K_{A,\omega'}|^2 + \eta^2} \cdot \\ & \left[\begin{array}{ccc} \sigma_\omega^2 |K_{A,\omega}|^2 & \sigma_\omega^2 K_{A,\omega}^* K_{A,\omega'} & \sigma_\omega^2 K_{A,\omega}^* \\ \sigma_{\omega'}^2 K_{A,\omega'}^* K_{A,\omega} & \sigma_{\omega'}^2 |K_{A,\omega'}|^2 & \sigma_{\omega'}^2 K_{A,\omega'}^* \end{array} \right] \end{aligned} \tag{10}$$

and σ_w^2 is inversely related to $G_{h,}$ the Fourier transform of the derivative operator g_h, via $\sigma_w^2 = 1/|G_{h,\omega}|^2$.

Proof: Up to constants, the log-likelihood with K_A is

$$-\log P(Y|X) = \frac{1}{2\eta^2} \sum_{w=0}^{\Omega/2} |K_{A,\omega} X_\omega + K_{A,\omega'} X_{\omega'} - Y_\omega|^2 \ . \tag{11}$$

Next, the assumed gradient prior in Eq. (4) can be written as a function of the convolution of the signal with derivative filters. For a 1-D signal with only horizontal gradients,

$$-\log P(x) = \|g_h * x\|^2 + const. \tag{12}$$

From Parseval's theorem, $\|g_h * x\|^2 = \sum_\omega |G_{h,\omega} \cdot X_\omega|^2$. In the frequency domain, the prior thus becomes diagonal:

$$-\log P(X) = \sum_{\omega=0}^{\Omega} \frac{|X_\omega|^2}{2\sigma_\omega^2} + const, \quad \sigma_\omega^{-2} = |G_{h,\omega}|^2 \ . \tag{13}$$

In other words, an L_2 prior on image gradients is a diagonal Gaussian prior in the Fourier domain, whose variance at each frequency ω is the power of the derivative filter. Hence, σ_ω^2 decays to zero as ω increases, in agreement with well-studied properties of natural image statistics.

Combining Eqs. (11) and (13), The MAP estimate is

$$\begin{aligned} \hat{X} &= \arg\max \log P(X|Y) \\ &= \arg\max(\log P(Y|X) + \log P(X)) \\ &= \arg\min \sum_{w=0}^{\Omega/2} \frac{|K_\omega X_\omega + K_{\omega'} X_{\omega'} - Y_\omega|^2}{\eta^2} \\ &\qquad + \frac{|X_\omega|^2}{\sigma_\omega^2} + \frac{|X_{\omega'}|^2}{\sigma_{\omega'}^2} \ . \end{aligned} \tag{14}$$

A short calculation shows that the solution $\hat{X}$ is given by

$$\begin{pmatrix} \hat{X}_\omega \\ \hat{X}_{\omega'} \end{pmatrix} = F_\omega Y_\omega , \tag{15}$$

where F_ω is a generalized Wiener filter

$$F_\omega = \left(\begin{pmatrix} K^*_{A,\omega} \\ K^*_{A,\omega'} \end{pmatrix} (K_{A,\omega}, K_{A,\omega'}) + \begin{bmatrix} \frac{\eta^2}{\sigma^2_\omega} & 0 \\ 0 & \frac{\eta^2}{\sigma^2_{\omega'}} \end{bmatrix} \right)^{-1} \begin{pmatrix} K^*_{A,\omega} \\ K^*_{A,\omega'} \end{pmatrix}$$

$$= \frac{1}{\sigma^2_\omega |K_{A,\omega}|^2 + \sigma^2_{\omega'} |K_{A,\omega'}|^2 + \eta^2} \begin{pmatrix} \sigma^2_\omega K^*_{A,\omega} \\ \sigma^2_{\omega'} K^*_{A,\omega'} \end{pmatrix} .$$

When $K_T{=}K_A$, combining the above with Eq. (7) yields Eq. (9). The case $K_T{\neq}K_A$ follows by writing Eq. (7) as:

$$Y_\omega = K_{A,\omega} \left(\frac{K_{T,\omega}}{K_{A,\omega}} X_\omega \right) + K_{A,\omega'} \left(\frac{K_{T,\omega'}}{K_{A,\omega'}} X_{\omega'} \right) + N_\omega . \quad (16)$$

■

We now study the implications of this lemma. First, note that when $K_T = K_A$, the recovered $\hat{X}_\omega, \hat{X}_{\omega'}$ are shrunk versions of the input Y_w. For instance, consider an ideal low pass filter with $K_{A,\omega'} = 0$, with noise-free data, $\eta = 0$. We find that at the high frequency ω', which did not contribute to Y_ω (since $K_{A,\omega'} = 0$), $X_{\omega'} = 0$. At the low frequency, meanwhile, $\hat{X}_\omega$ is exactly the true value, $\hat{X}_\omega = Y_\omega / K_{A,\omega} = X_\omega$. In the presence of noise with variance η^2, at frequencies where $\sigma^2_\omega \ll \eta^2$, the elements of F_ω are close to zero. This makes sense because at these frequencies, the measurement Y_ω is dominated by noise, and the actual signal X_ω cannot be accurately recovered. We should instead estimate it as some quantity significantly shrunk towards zero. If the filter includes aliasing, i.e. $|K_{A,\omega'}| > 0$, the recovered signal includes aliasing as well. That is, some of the high frequency component $X_{\omega'}$ contributes to the low frequency reconstruction at $\hat{X}_\omega$, and visa versa.

Kernel Mismatch: Next, we study the implication of the lemma when the signal was blurred with K_T while the MAP estimator assumed a different kernel $K_A \neq K_T$. According to Eq. (9), H_A now acts on the signal $K_{T,\omega}/K_{A,\omega} \cdot X_\omega$ instead of on the true X_ω. At high frequencies, where signal variance is lower than noise level, $H_{A,\omega} \approx 0$, and kernel mismatch has little effect on the output.

In contrast, at other frequencies, an incorrect kernel may have strong detrimental effects. As in deconvolution, if the true K_T blurs more than the assumed K_A, then typically $|K_{T,\omega}| < |K_{A,\omega}|$, and thus X_ω is more attenuated. If K_T is sharper than K_A, $|K_{T,\omega}| > |K_{A,\omega}|$, and X_ω is amplified. In the first case $K_{T,\omega}/K_{A,\omega}$ acts as a blurring filter and in the second as a sharpening filter (see Fig. 1).

As an illustration of this effect, Fig. 5 simulates two Gaussian filters. If

$$K_{T,\omega} = \exp\left(-\frac{1}{2}\beta_T |\omega|^2\right), \quad K_{A,\omega} = \exp\left(-\frac{1}{2}\beta_A |\omega|^2\right) \quad (17)$$

then

$$\frac{K_{T,\omega}}{K_{A,\omega}} = \exp\left((\beta_T - \beta_A)|\omega|^2\right) . \quad (18)$$

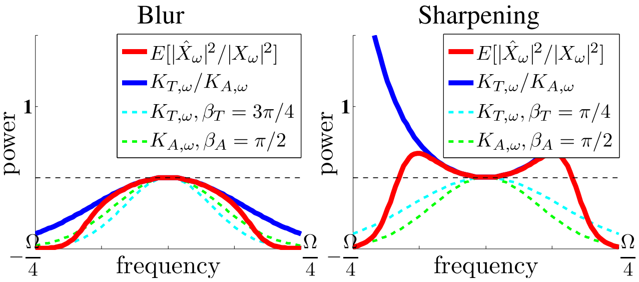

Figure 5. **Effect of an inaccurate kernel.** $\beta_T > \beta_A$ yields a Gaussian blur filter (left); $\beta_T < \beta_A$ gives a sharpening filter (right).

For $\beta_T > \beta_A$ we obtain a Gaussian blur filter, and for $\beta_T < \beta_A$ a sharpening filter (the exponent is positive). Assuming for simplicity $K_{A,\omega'} = 0$, the red curve of Fig. 5 demonstrates the expected power scaling of each filter, namely $E[|\hat{X}_\omega|^2/|X_\omega|^2]$. For $\beta_T > \beta_A$ the amplitude is decreased (red curve below 1) while for the $\beta_T < \beta_A$ case the amplitude is magnified at the middle band of frequencies. High frequencies whose expected power is below the noise variance are not amplified by the reconstruction filter.

The off-diagonal panels in Fig. 1 illustrate these phenomena. An assumed wider kernel results in ringing, and a narrow kernel in over-blurring. As mentioned earlier, for practical SR, both effects are undesirable.

Kernel Uncertainty: In practice, SR algorithms may be applied to real images whose blur kernel is not precisely known. One approach, taken by several SR algorithms, is to ignore the true blur kernel, and utilize some default, such as bicubic. As noted previously, when the assumed default kernel is narrower than the true kernel, as occurs with our actual camera (see Fig. 2), this results in somewhat smoother SR images, but without ringing artifacts. Hence, the resulting images are visually acceptable. A second approach taken by various SR algorithms to cope with imprecise knowledge of the true kernel, is to give more weight to the image prior and reduce the weight of the reconstruction constraint (e.g. smaller λ in Eq. (6)). This makes such algorithms less sensitive to kernel mismatch, but also reduces the quality of their results.

We now derive a principled estimation strategy to take into account kernel uncertainty. As expected, this comes at the price of reduced accuracy.

Let us denote by $q(K)$ the density of possible kernels K. We denote by μ^K, Σ^K the 2×1 and 2×2 matrices of mean and covariance of $(K_\omega, K_{\omega'})$ according to the distribution $q(K)$. We see below that these first and second order moments serve as sufficient statistics.

Extending Eq. (14) we seek $\hat{X}$ minimizing

$$\int q(K) \sum_w \frac{|K_\omega X_\omega + K_{\omega'} X_{\omega'} - Y_\omega|^2}{\eta^2} + \frac{|X_\omega|^2}{\sigma^2_\omega} + \frac{|X_{\omega'}|^2}{\sigma^2_{\omega'}} dK . \quad (19)$$

A short calculation shows that the minimum is obtained by

$$\begin{pmatrix} \hat{X}_\omega \\ \hat{X}_{\omega'} \end{pmatrix} = \left(\mu^K (\mu^K)^T + \Sigma^K + \begin{bmatrix} \frac{\eta^2}{\sigma^2_\omega} & 0 \\ 0 & \frac{\eta^2}{\sigma^2_{\omega'}} \end{bmatrix} \right)^{-1} (\mu^K)^T Y_\omega \,. \tag{20}$$

Comparing the above formula to Eq. (16), if $q(K)$ had no uncertainty, then $\mu^K = (K_\omega, K_{\omega'})$ and Σ^K is the zero matrix, and Eq. (20) reduces to the Wiener filter of Eq. (16). The effect of kernel uncertainty is similar to that of noise: the kernel covariance Σ^K adds to the noise covariance part in Eq. (20), and the resulting estimator has a larger shrinkage factor. Hence the accuracy of the estimate is reduced.

In conclusion, one can derive SR algorithms which are more robust to uncertainty in the kernel, but this comes at the price of estimation quality. If an accurate model of k can be found (a smaller-norm certainty covariance Σ^K), the reconstruction accuracy is improved.

5. Discussion

In this paper, we examined the effect of two components of SR: the natural image prior and the reconstruction constraint. We showed that an accurate blur model and its corresponding reconstruction constraint are crucial to the success of SR algorithms. The influence of an accurate estimate of the blur kernel is significantly larger than that of a sophisticated prior.

These observations suggest that to advance SR algorithms, in particular for applications to real images, future research should place more emphasis on the recovery of real camera blur. While existing blind motion deblurring methods can be adapted to this task, we note that SR kernel recovery is a simpler task, since the blur is a property of the sensor and is fixed for all images captured by the same camera under similar imaging conditions. As described in [4], this camera blur can be calibrated using two images of the same calibration target.

Acknowledgements: We thank Intel, ERC and BSF for funding this research.

References

[1] H. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Trans. Img. Proc.*, 2005.

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *PAMI*, 2002.

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

[4] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution - supplementary results. 2013. www.wisdom.weizmann.ac.il/~levina/papers/supres.

[5] R. Fattal. Image upsampling via imposed edge statistics. *SIGGRAPH*, 2007.

[6] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Trans. Graph.*, 2011.

[7] W. Freeman and C. Liu. Markov random fields for super-resolution and texture synthesis. *Advances in Markov Random Fields for Vision and Image Processing*, 2011.

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

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

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

[11] Y. He, K. Yap, L. Chen, and L. Chau. A soft map framework for blind super-resolution image reconstruction. *Image Vision Comput.*, 2009.

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

[13] K. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *PAMI*, 2010.

[14] A. Levin, R. Fergus, F. Durand, and W. Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 2007.

[15] S. Mallat and G. Yu. Super-resolution with sparse mixing estimators. *IEEE Trans. Img. Proc.*, 2010.

[16] 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.

[17] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Trans. Img. Proc.*, 2009.

[18] D. Robinson and P. Milanfar. Statistical performance analysis of super-resolution. *Image Processing, IEEE Transactions on*, 15(6):1413–1428, 2006.

[19] J. Sun, Z. Xu, and H. Shum. Image super-resolution using gradient profile prior. In *CVPR*, 2008.

[20] J. Sun, N. Zheng, H. Tao, and H. Shum. Image hallucination with primal sketch priors. In *CVPR*, 2003.

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

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

[23] M. Tappen, B. Russell, and W. Freeman. Efficient graphical models for processing images. In *CVPR*, 2004.

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

[25] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications in image super-resolution and photo-sketch synthesis. In *CVPR*. IEEE, 2012.

[26] Z. Wang, A. Bovik, H. Sheikh., and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Img. Proc.*, 2004.

[27] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Img. Proc.*, 2010.