Single Image Super-Resolution using Gaussian Process Regression

He He and Wan-Chi Siu
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
{07821020d, enwcsiu@polyu.edu.hk}

Abstract

In this paper we address the problem of producing a high-resolution image from a single low-resolution image without any external training set. We propose a framework for both magnification and deblurring using only the original low-resolution image and its blurred version. In our method, each pixel is predicted by its neighbors through the Gaussian process regression. We show that when using a proper covariance function, the Gaussian process regression can perform soft clustering of pixels based on their local structures. We further demonstrate that our algorithm can extract adequate information contained in a single low-resolution image to generate a high-resolution image with sharp edges, which is comparable to or even superior in quality to the performance of other edge-directed and example-based super-resolution algorithms. Experimental results also show that our approach maintains high-quality performance at large magnifications.

1. Introduction

The goal of super-resolution (SR) is to estimate a high-resolution (HR) image from one or a set of low-resolution (LR) images. This inverse problem is inherently ill-posed since many HR images can produce the same LR image. SR methods can be broadly categorized into three classes as in [19]: *interpolation-based methods*, *learning-based methods* and *reconstruction-based methods*. Interpolation-based methods (e.g., [9, 13, 21] are fast but the results are lack of fine details. Reconstruction based methods (e.g., [2, 11, 15, 18, 10]) apply various smoothness priors (e.g., [1, 5, 19]) and impose the constraint that when properly downsampled, the HR image should reproduce the original LR image. Alternatively, in learning-based methods (e.g., [14, 7, 4, 22]), detailed textures are hallucinated by searching through a training set of LR/HR image or patch pairs. However, note that the training images need to be carefully selected, otherwise unexpected details can be found [6, 12].

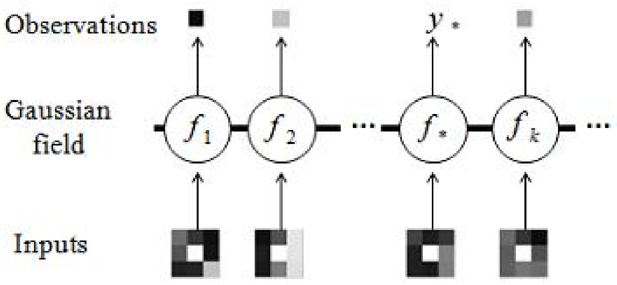


Figure 1. (Adapted from [17]) Graphical model of GP regression for image SR. Squares show the observed pixels and circles represent the unknown Gaussian field. Inputs are neighbors (predictors) of the target pixel which is at the center of each 3 × 3 patch. The thick horizontal line represents a set of fully connected nodes.

In this paper we emphasize on super-resolution using only the LR input image without any external dataset or exemplar image. Previously, Fattal [6] proposed an upsampling method based on a unique relationship between the gradient fields of LR and HR images. Subsequently, Sun *et al*. used the Gradient Profile Prior [19] learned from a large number of natural images to obtain sharp edges at large magnifications. Glasner *et al*. [8] exploited the recurrence of patches and constructed LR/HR training pairs by searching through an image pyramid.

Our contribution is two-fold. Firstly, we propose a framework to learn the reverse mapping from a LR image to the corresponding HR image pixelwise, relying on local structures defined by each pixel's neighborhood. Pixels in the HR image are first estimated by their nearest neighbors (regressors) in an initial upsampled image via Gaussian process regression (GPR). The result is then deblurred by learning from LR/HR patches obtained from its downsampled version and the LR input. This self-directed learning is ready to be extended to image deblurring and denoising as well. Secondly, we demonstrate that GPR can serve as a soft clustering method for pixels based on the similarity defined by its covariance function. Thus it could also be used in other SR algorithms to replace the nearest-neighbor based search. In our case, the covariance function enforces that a pixel is mainly estimated from training pixels embedded in similar environments, which results in significant improvement for edge recovery.

In the rest of this paper, we first state our motivation in Section 2. Then we give a brief overview of GPR in Section 3. In Section 4, we describe our algorithm and give justifications. Section 5 presents our results and the comparison to those produced by other methods. Finally, we conclude the paper with some discussion and future developments of the proposed algorithm.

2. Motivation

Our work is driven by the observation of structural redundancy in natural images. Previously, this self-similarity property has been exploited in image denoising by Non-Local Means [3], which was later generalized to a regularization method for inverse problems in [16]. Recently, it has been statistically shown in [8] that small patches (e.g., 5×5) of edges, corners and textures tend to recur both within and across image scales. In our case, the similarity between two pixels is defined as the difference between their local geometrical structures. For instance, pixels along the same edge tend to have similar neighborhoods whose intensities change fast in the direction perpendicular to that of the edge. Likewise, pixels in a smooth region have relatively invariant intensities within the neighborhood. Therefore, neighbors of a pixel indicate the local feature it is embedded in and can be used for prediction in a regression-based model.

GPR provides an elegant non-parametric Bayesian approach for inference in the function space directly. By encoding the high-level information that pixels having similar neighborhoods are strongly correlated, we essentially predict the target pixel from training examples whose local structures are similar to it (see Section 4.2). Due to the geometric duality [13] between LR and HR images, the input LR image serves as a sufficient training dataset for extracting the local structural information.

3. Gaussian Process Regression

Gaussian Process (GP) defines a distribution over function f, where f is a mapping from the input space $\mathcal{X}$ to $\mathfrak{R}$, such that for any finite subset of $\mathcal{X}$, its marginal distribution $P(f(\mathbf{x}_1), f(\mathbf{x}_2), ...f(\mathbf{x}_n))$ is a multivariate normal distribution, where $\mathbf{x}$ an input vector. In this section we give a brief review of GPR for implementation purpose; further details can be found in [17].

Gaussian Process is parameterized by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ such that

$$\mathbf{f}|\mathbf{X} \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{X}, \mathbf{X})) \qquad (1)$$

or equivalently,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j)). \qquad (2)$$

where rows of the design matrix X are input vectors, $\mathbf{f}$ is a vector of function values and $K(X, X)$ denotes the n by n matrix of covariances such that $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

In GPR we impose a GP prior on the target function value. Let y denote one observation with Gaussian noise ϵ and we have the GPR model

$$y = f(\mathbf{x}) + \varepsilon, \ \ \varepsilon \sim \mathcal{N}(0, \sigma_n^2). \qquad (3)$$

The joint distribution of the training outputs $\mathbf{y}$ and the test outputs $\mathbf{f}_*$ with a zero mean function is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X})) + \sigma_n^2 I & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right), \qquad (4)$$

where $\mathbf{X}$ and $\mathbf{X}_*$ are design matrices for training data and test data respectively. Conditioning $\mathbf{f}_*$ on the observation $\mathbf{y}$, we can derive the predictive distribution

$$\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, V(\mathbf{f}_*)), \text{ where} \qquad (5)$$

$$\bar{\mathbf{f}}_* = K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}, \qquad (6)$$

$$V(\mathbf{f}_*) = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} K(\mathbf{X}, \mathbf{X}_*). \qquad (7)$$

The notation $K(\mathbf{X}_*, \mathbf{X})$, $K(\mathbf{X}, \mathbf{X}_*)$, $K(\mathbf{X}_*, \mathbf{X}_*)$ for matrices of covariances are similar to the previous notation $K(\mathbf{X}, \mathbf{X})$.

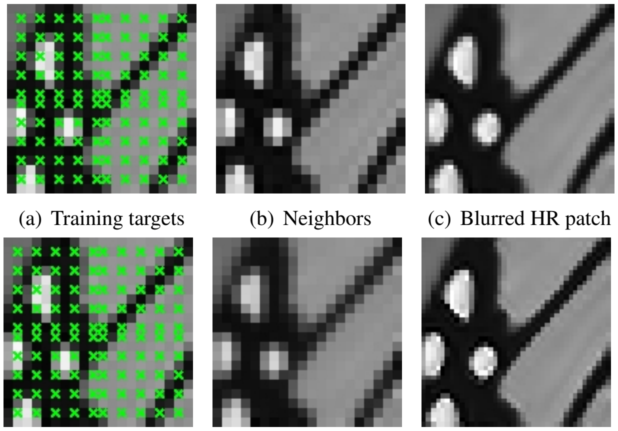
Equation 6 shows that the mean prediction is a linear combination of the noisy observation $\mathbf{y}$. Generally, GPR encodes the assumption that close inputs are likely to give similar outputs, hence examples similar to the test point are assigned higher weights. By specifying a proper covariance function, we can define the similarity based on domain knowledge.

4. GPR for Super-resolution

In our regression-based framework, patches from the HR image are predicted pixelwise by corresponding patches from the LR image (Section 4.1). GPR provides a way for soft-clustering of the pixels based on the local structures they are embedded in (Section 4.2). Given the intuitive interpretation of hyperparameters of the covariance function in our case, we can optimize their values through MAP estimation (Section 4.3).

4.1. Single Image SR

Figure 1 shows a chain graph representation of GPR for image SR in our setting, where each 3×3 patch from the input image forms a predictor-target training pair. Thus in


(a) Training targets (b) Neighbors (c) Blurred HR patch

(d) Training targets (e) Neighbors (f) Deblurred HR patch

Figure 2. Single image SR framework. (a) and (d) are input LR patches, in which the crossed points are target pixels for training produced by simple grid sampling. (b) is the same patch as (a), where neighbors are taken as the intensity values of the crossed points' surrounding pixels. (e) is obtained by blurring and downsampling (c), the output of stage one. Neighbors are taken in the same way as in stage one from (e). (f) shows the deblurred result of the second stage.

Equation 3, the observed y is the intensity of the pixel at the center of a 3×3 patch and $\mathbf{x}$ is an eight-dimensional vector of its surrounding neighbors. In order to adapt to different regions of the image, it is partitioned into fixed-sized and overlapped patches (e.g., 30×30), and the algorithm is run on each of them separately. The patch-based results are then combined to give the whole HR image.

We predict the HR image using a two-stage coarse-to-fine approach, which is consistent with the image formation process. As shown in [2, 18], the imaging process can be modeled as a degradation process of the continuous spatial domain depending on the camera's Point Spread Function (PSF). After discretization, this process can be further expressed as

$$\begin{aligned} L &= (f * H)\downarrow^d, \text{ or equivalently,} && (8)\\ \tilde{H} &= f * H \text{ and } L = \tilde{H}\downarrow^d && (9)\end{aligned}$$

where L and H denote the LR and HR image respectively, $\tilde{H}$ denotes the blurred HR image, f is the blur kernel and $\downarrow^d$ denotes the downsampling operator with a scaling factor d. As a reasonable inversion of the degradation process, in the first stage we intend to upsample the LR image without loss of details and obtain $\tilde{H}$. In the second stage we use L and its blurred version $\tilde{L}$ (downsampled from $\tilde{H}$) to simulate the blurring process, thus refine the estimates to recover H. The above ideas can be implemented by Algorithm 1.

Figure 2 gives a real example of the resolving process. Figure 2(a), 2(b) and 2(c) show the first stage, where both the training targets sampled in Figure 2(a) and their neighbors in Figure 2(b) come from the LR input patch. Neighbors for predicting pixels in the HR image are obtained by upsampling Figure 2(a) using bicubic interpolation. Figure 2(d), 2(e) and 2(f) show the second stage, where the training targets 2(d) are the same as those in the first stage, while the neighbors come from the blurred patch (Figure 2(e)). Figure 2(f), the deblurred result shows sharper edges than Figure 2(c) from stage one.

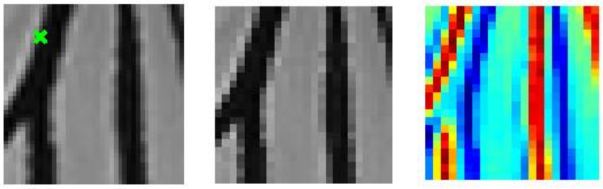
```
Algorithm: SRGPR($L$)
  $\tilde{H} \leftarrow$ Upsample($L$)
  $\tilde{L} \leftarrow \tilde{H}\downarrow^d$ (Blur and downsample)
  $H \leftarrow$ Deblur($\tilde{H}, L, \tilde{L}$)
  return $H$
Function: Upsample ($L$)
  Bicubic interpolation: $H_b \leftarrow L\uparrow^d$
  Partition $L$ into $n$ overlapped patches $P_1, ..., P_n$
  for $p_L = P_1, ..., P_n$ do
      Sample pixels in $p_L$ to obtain the target vector $\mathbf{y}$
      Put the eight neighbors of each element of $\mathbf{y}$ in
      $X_{NL}$ as a row vector for training
      Train a GPR model $M$ using $\{\mathbf{y}, X_{NL}\}$
      Put the eight neighbors of each pixel of $H_b$ in
      $X_{NH_b}$ as a row vector for prediction
      $p_{\tilde{H}} \leftarrow M(X_{NH_b})$
  end
  return $\tilde{H}$ constructed from $p_{\tilde{H}}$
Function: Deblur ($\tilde{H}, L, \tilde{L}$)
  Partition $\tilde{L}$ into $n$ overlapped patches $P_1, ..., P_n$
  corresponding to those in $L$
  for $p_{\tilde{L}} = P_1, ..., P_n$ do
      Obtain the same target vector $\mathbf{y}$ in $p_L$
      Put the eight neighbors in $p_{\tilde{L}}$ of each element of
      $\mathbf{y}$ in $X_{N\tilde{L}}$ as a row vector for training
      Train a GPR model $M$ using $\{\mathbf{y}, X_{N\tilde{L}}\}$
      Put the eight neighbors of each pixel of $\tilde{H}$ in
      $X_{N\tilde{H}}$ as a row vector for prediction
      $p_H \leftarrow M(X_{N\tilde{H}})$
  end
  return $H$ constructed from $p_H$
```

Algorithm 1: Super-resolution using GPR.

4.2. Covariance Functions

The covariance function plays the central role in GPR as it encodes our assumptions about the underlying process by defining the similarity between functions. We model the image as a locally stationary Gaussian Process and choose the squared exponential covariance function:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j)}{\ell^2}\right), \quad (10)$$



(a) Test point (b) Training patch (c) Covariance matrix

Figure 3. Covariance Function. (b) is a patch from the LR input image and (a) is its corresponding patch from the upsampled image (bicubic interpolation). The crossed point in (a) is a test point $\mathbf{x}_*$ to be estimated from the traning set $\mathbf{X}$ in (b). (c) shows its covariance matrix, where each pixel's intensity value is proportional to the covariance between $\mathbf{x}_*$ and the training target at the same location in (b), evaluated by the covariance function, Equation 10

.

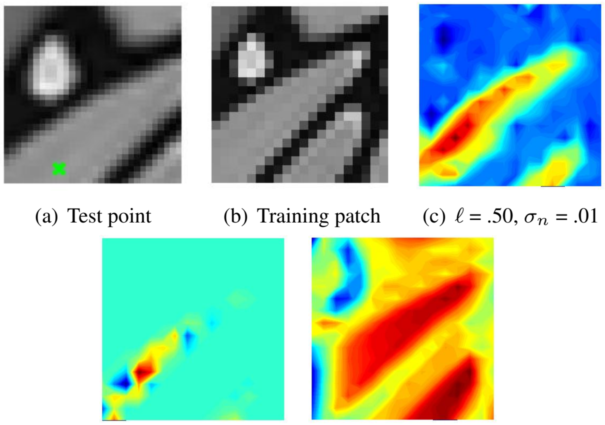
where σ_f^2 represents the signal variance and ℓ defines the *characteristic length scale*. Therefore, in our case similarity is based on the Euclidean distance between two vectors of the eight neighboring pixels' intensities, since the difference of intensity values around the target pixel indicates its location (e.g. edges or smooth regions) and the direction of the edge if it is an edge-pixel.

Figure 3(c) shows the covariance matrix K for one edge pixel $\mathbf{x}_*$ (the crossed point) in Figure 3(a) predicted by training points $\mathbf{X}$ in the LR training patch, where $\mathbf{K}_{ij} = k(\mathbf{x}_*, \mathbf{X}_{ij})$ and i, j correspond to pixel indices in Figure 3(b). Observe that high responses (red regions) from the training patch are largely concentrated on edges, which justifies our choice of the covariance function. In addition, it is noted that high-responsive regions do not only include points on the same edge as the test point, but also other similar edges within the patch. Thus, the process can capture both local and global similarity within a patch. In general, pixels embedded in a similar structure to that of the target pixel in terms of the neighborhood tend to have higher weights during prediction.

4.3. Hyperparameter Adaptation

So far, we have taken the parameter values in the covariance function to be given. Here we explore the effect of varying these hyperparameters and illustrate how to set them through optimization.

In Bayesian statistics, hyperparameters refer to parameters in the prior distribution, which in our case are the signal variance, σ_f^2, the *characteristic length scale*, ℓ in Equation 10 and the noise variance, σ_n^2 in Equation 3. While signal variance and noise variance are relatively self-explicit, the *characteristic length scale* can be regarded as parameters controlling the number of upcrossings of a level u in a one-dimensional GP. Assuming a zero mean function, it is shown in [17] that the *characteristic length scale* is inversely proportional to the mean number of level-zero upcrossings, given the squared exponential covariance function.



(a) Test point (b) Training patch (c) ℓ = .50, σ_n = .01

(d) ℓ = .05, σ_n = .001 (e) ℓ = 1.65, σ_n = .14

Figure 4. Hyperparameter adaptation. (a) is the upsampled patch (bicubic interpolation) used for prediction. (b) is the original low-resolution patch used for training. (c) to (e) show the contour plots of the weight matrix given the hyperparameters listed below each image. The hyperparameters used in (c) is obtained by MAP estimation while the other two sets of hyperparameters are chosen by hand for comparison. The signal variance σ_f^2 is fixed as 0.16^2 for all three settings.

Figure 4 gives an interpretation of the hyperparameters in terms of the weighted linear combination of training targets, $\mathbf{y}$, in Equation 6. Figure 4(a) and Figure 4(b) are the upsampled patch (bicubic interpolation) and the LR input patch respectively. To predict the crossed point as shown in Figure 4(a), we use the mean of the predictive distribution (Equation 6). Let the n by 1 weight vector $\mathbf{w}$ be $K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}$. The linear predictor can then be written as $\mathbf{w}^T\mathbf{y}$. We reshape the vector $\mathbf{w}$ into a square matrix W and plot its contour. Each of W's entry is the weight of a corresponding training point in Figure 4(b). The signal variance is set by optimization of the marginal likelihood (further explained in subsequent paragraphs), which is the same in all three settings. In Figure 4(c), the hyperparameters are obtained by the MAP estimation. Figure 4(d) shows the effect of a shorter length scale with low noise and Figure 4(e) shows the result of a longer length scale with high noise. We can see that in Figure 4(d) the contour map has several peaks, indicating a small number of highly correlated training points. In contrast, the longer length scale in Figure 4(e) over-smoothes the signal and results in large high-responsive regions. Generally, this phenomenon stands for two possible views of a given signal: the former model assumes a quickly-varying field and fits to high-frequency details that could be noise; the latter model assumes a slowly-varying field thus some observed genuine data could be ignored. A non-optimal solution either results in high-frequency artifacts in the image or a blurry image.

In Bayesian model selection, we usually use hierarchical models and impose a hyper-prior $p(\boldsymbol{\theta}|\mathcal{H})$ over the hyperparameters $\boldsymbol{\theta}$. Then the posterior can be expressed as follows:

$$p(\boldsymbol{\theta}|\mathbf{X},\mathbf{y},\mathcal{H}) = \frac{p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta},\mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}{\int p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta},\mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})d\boldsymbol{\theta}}. \tag{11}$$

Depending on the model, the evaluation of the normalizing constant in Equation 11 may be intractable, thus calls for variational methods to approximate or Markov chain Monte Carlo (MCMC) methods to sample from the posterior. Alternatively, in the context of image SR we can take advantage of the above intuitive interpretation of the hyperparameters and solve the problem through MAP estimation.

In our model, the marginal likelihood is

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X})p(\mathbf{f}|\mathbf{X})d\mathbf{f} \tag{12}$$

Given the likelihood $\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f},\sigma_n^2 I)$ (Equation 3) and the GP prior over the latent function $\mathbf{f}$ (Equation 1), this integral can be analytically solved, giving rise to the *log marginal likelihood*

$$\log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}log|\mathbf{K}_y| - \frac{n}{2}\log 2\pi, \tag{13}$$

where $K_y = K(\mathbf{X},\mathbf{X}) + \sigma_n^2 I$, representing the covariance matrix of the noisy observation $\mathbf{y}$. The optimal solution θ^* can be obtained by gradient descent with the following partial derivative

$$\frac{\partial\mathcal{L}}{\partial\theta_i} = \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_i}\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}tr(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_i}), \tag{14}$$

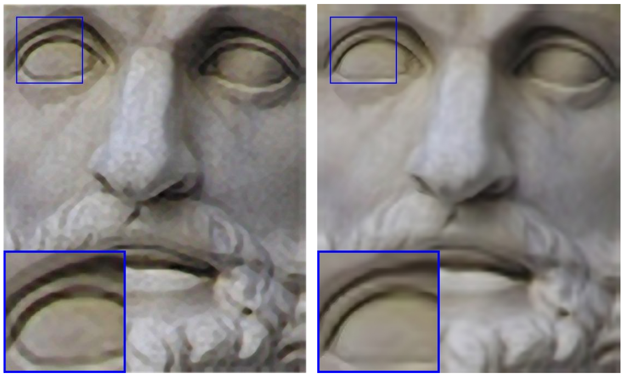
where $\mathcal{L}$ refers to the above *log marginal likelihood*. In our experiments, satisfactory results can be obtained by initializing ℓ at 0.223, σ_n at 0.05 and σ_f^2 to be the variance of the training targets.

We conclude this section by giving the following advantages of our patch-based GP regression for image SR. Firstly, recurring patterns (3×3 neighborhood) in the LR patch can be captured for efficient training. Secondly, different image regions (e.g., 30×30 patches) can have a customized set of hyperparameters to adapt to its global characteristics. Lastly, the $\mathcal{O}(N^3)$ time complexity is reduced to $\mathcal{O}(Mn^3)$, where M is the number of patches which grows linearly with the image size N and n is the number of training points in one patch which is a constant.

5. Experiments

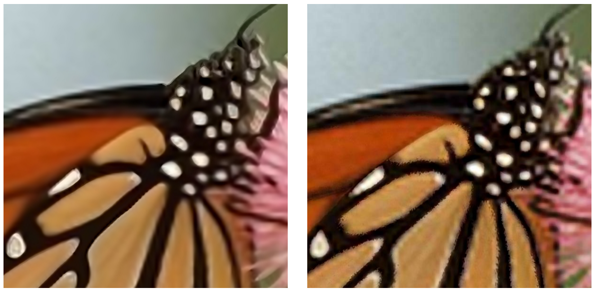
Implementation: We test our method on a variety of images with a magnification factor from 2 to 10. In most of our experiments we began by using the original image as LR input and magnify it with a scale factor of 2. For further magnification, we used the previous output image as the input LR image and solved its HR image. In the second stage of our method, we made use of the original LR image for deblurring, which also serves as a constraint that when downsampled, the HR image should reproduce the LR image. We set the patch size to 20×20 for the original LR image and increase it according to the magnification factor in later magnifications. For each patch, approximately $2/3$ area is overlapped with neighboring patches. After running our algorithm on each patch, we combined them by linear smoothing in the overlapping areas. When processing color images, we worked in the *YIQ* color space and only applied our SR method to the illuminance (*Y*) channel since humans are more sensitive to illuminance changes. The other two chrominance channels (*I* and *Q*) are simply upsampled using bicubic interpolation. The three channels were then combined to form our final HR image.

Results: In Figure 5, we compare our method with the bicubic interpolation and the Gradient Profile Prior reconstruction [19]. The bicubic interpolation introduces ringing and jaggy artifacts, for example along edges on the wing



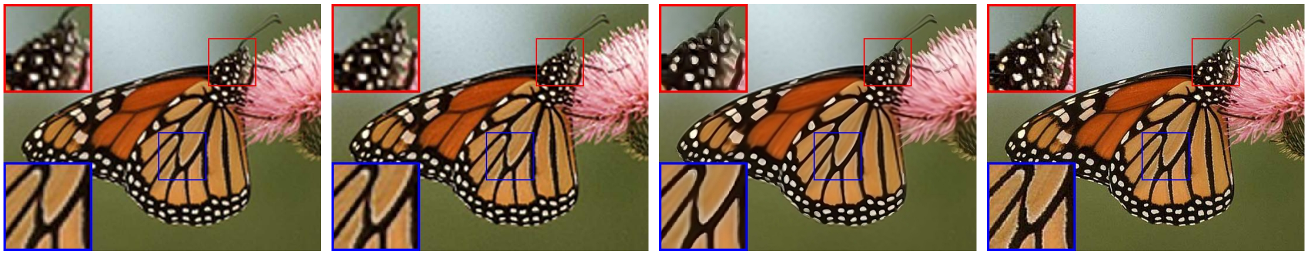
(a) GPP (b) Our result

Figure 6. Super-resolution (8×) comparison of Gradient Profile Prior (GPP) reconstruction [19] and our methods.



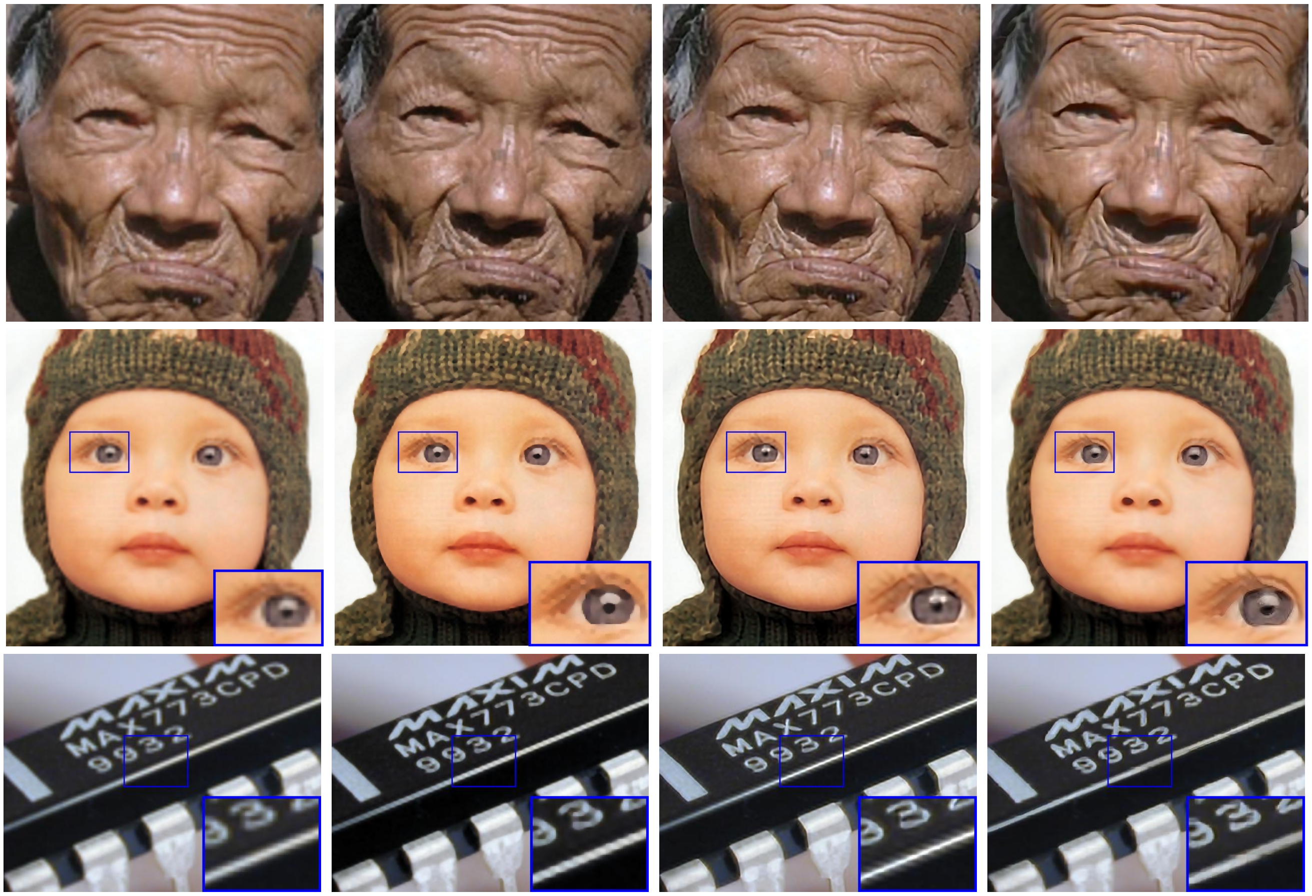
(a) 10× our result (b) 10× detail synthesis

Figure 7. Super-resolution result of direct magnification (3×) and 10× comparison with the result from [20].



(a) Bicubic (MSSIM=0.84) (b) GPP (MSSIM=0.84) (c) Our result (MSSIM=0.86) (d) Ground truth

Figure 5. Super-resolution (3×) comparison of bicubic interpolation, Gradient Profile Prior (GPP) reconstruction [19] and our methods. The MSSIM scores with respect to the ground truth are listed below each result.



(a) Bicubic (b) Imposed edge statistics (c) Patch redundancy (d) Our result

Figure 8. Super-resolution (4×) comparison of bicubic interpolation, reconstruction with imposed edge statistics [6], single image super-resolution using redundant patches across scales [8] and our method.

(region in the blue-bordered rectangle). The GPP reconstruction alleviates the artifacts yet the overall result is still blurry, as shown in the red-bordered rectangle. Our result generates sharp edges with rare artifacts and is most close to the ground truth, in terms of both subjective visual evaluation and the Mean Structural Similarity (MSSIM) index [23], which assesses the image quality from the perspective of image formation. Figure 6 further shows a comparison between GPP [19] and our methods under 8× magnification. In the enlarged rectangle region, we can see that the result of GPP is less realistic while our result shows clear carved curve of the eye.

To further inspect the effectiveness of our method, we show a comparison of our large 10× SR result to that of the single image detail synthesis with edge prior method from [20] in Figure 7(a) and Figure 7(b). With the example texture, the detail synthesis approach can produce realistic details which are usually lost under large magnification. Our results, however, demonstrate comparable quality in terms of the sharpness of edges, for example in regions of the antennae and the head.

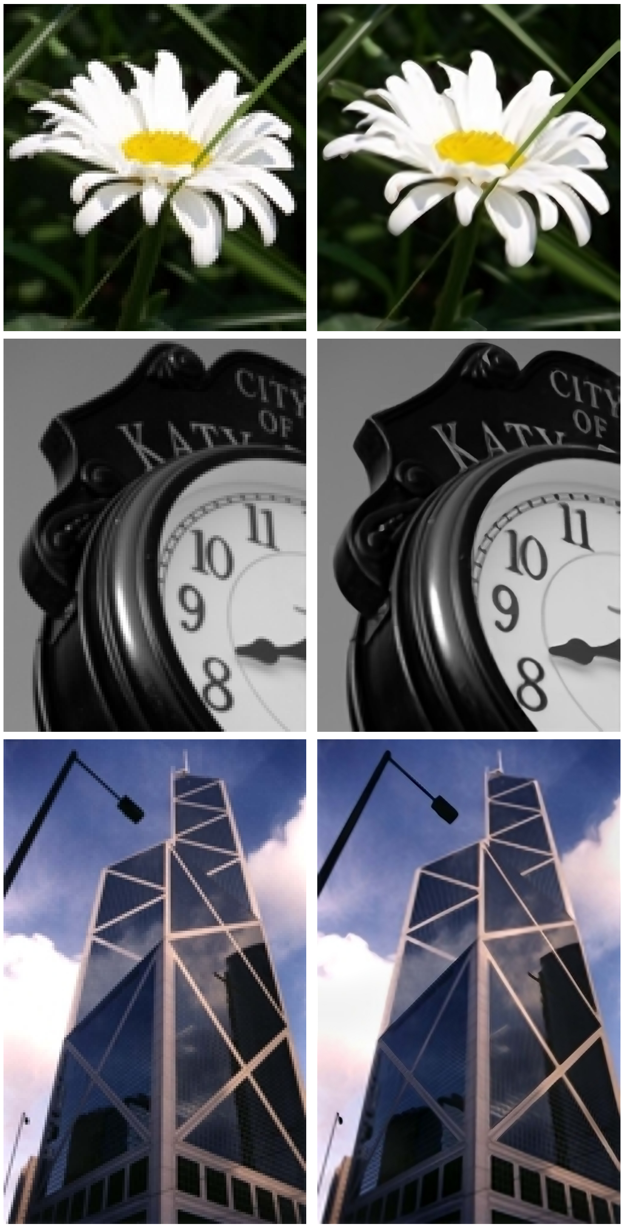
Figure 8 compares our method with bicubic interpolation, the imposed edge statistics approach [6] and the single image SR approach using patch redundancy across scales [8]. The image of the old man's face has various edges of different scales. The imposed edge statistics method produces some unrealistic edges as in the case of Figure 6(a). For instance, the forehead wrinkles show lineation effect. In the image of the chip connector, the long edge of the side is approximately constructed by only one pixel in the input LR image. Both the imposed edge statistics approach and the single image SR approach have staircase or ringing artifacts along the major edge. By capturing the pixels' local structures and searching similar predictors globally in a patch, our method produces sharp edges with the least artifacts.

Figure 9 shows more results using our method. Images on the left are LR inputs shown by nearest-neighbor interpolation and images on the right are HR images with a magnification factor of 8. The results show that our method can produce sharp edges reliably at high magnification factors. In Table 6 we give objective measurements (RMS and MSSIM [23]) to evaluate our method in comparison to bicubic interpolation and New Edge Directed Interpolation [13] (NEDI). We notice that the performance of NEDI deteriorates fast when the magnification factor is larger than 2. The measurement shows that our method outperforms bicubic interpolation and NEDI with lower RMS error and higher MSSIM index.

Noisy input: In Figure 10, we demonstrate that our SR method is resistant to minor noise in the input LR image. For noisy input, we set σ_n according to the noise level in the LR image and optimize σ_f and ℓ by the MAP estimation, such that the model can correctly interpret noisy signals. We can see that in Figure 10(b), most of the noise are smoothed out.

6. Discussions and Conclusions

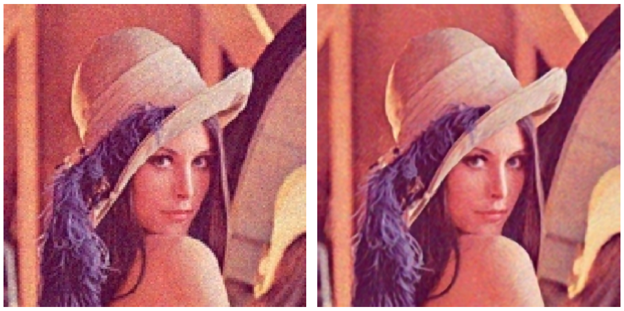
In this paper we present a novel method for single image super-resolution. Instead of imposing sophisticated edge priors as in [6, 19], we use a simple but effective covariance function to capture the local structure of each pixel through the Gaussian process regression. In addition, by using a non-parametric Bayesian model, the prediction process is able to adapt to different edges in the local image patch instead of setting a general parameter learnt from a large number of images. The experimental results show that our method can produce sharp edges with minimal artifacts. We propose a two-stage framework, where the first stage aims to recover an HR image based on the structural information extracted from the LR image and the second stage intends to refine the obtained HR image by learning from a training set constructed by the LR image itself. In comparison to the general training set as used in [7, 24], our results capture most of the details presented in the LR image and the visual quality is comparable to some example-based methods (see Figure 7(a)).



(a) Input (b) Our results

Figure 9. More Super-resolution (8×) results.

Though presented in the context of super-resolution, our framework of self-directed learning can be extended to



(a) Noisy LR input (b) Our result

Figure 10. Super-resolution (3×) result of noisy low-resolution image.

Image	RMS			MSSIM		
	BC	NEDI	Our	BC	NEDI	Our
Lena	10.25	11.12	9.65	0.80	0.79	0.82
Daisy	18.21	17.99	17.02	0.83	0.83	0.85
Tower	10.69	11.69	9.68	0.76	0.75	0.79

Table 1. RMS errors and MSSIM scores of different super-resolution (4×) methods. BC stands for bicubic interpolation. NEDI stands for new edge-directed interpolation [13].

other image reconstruction scenarios such as denoising and deblurring by modifying the covariance function and the training pairs. Finally, the performance may be further improved through using more complex covariance function to include extracted features and to express explicitly the local structure of a pixel.

Acknowledgement

This work is supported by the Center for Signal Processing (G-U863), the Hong Kong Polytechnic University and the Research Grant Council of the Hong Kong SAR Government (PolyU5278/08E: B-Q14F).

References

[1] H. A. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Trans. on IP*, 14(10):1647–1659, 2005. 449

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. on PAMI*, 24:1167–1183, 2002. 449, 451

[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 450

[4] H. Chang, D. Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, volume 1, pages 275–282, 2004. 449

[5] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. In *CVPR*, 2007. 449

[6] R. Fattal. Upsampling via imposed edges statistics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):95:1–95:8, 2007. 449, 454, 455

[7] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Comput. Graph. Appl.*, 22(2):56–65, March 2002. 449, 455

[8] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 449, 450, 454, 455

[9] H. S. Hou and H. C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoustics, Speech & Signal Proc.*, ASSP-26:508–517. 449

[10] K. W. Hung and W. C. Siu. New motion compensation model via frequency classification for fast video super-resolution. In *ICIP*, 2009. 449

[11] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993. 449

[12] K. I. Kim and Y. Kwon. Example-based learning for single-image super-resolution and jpeg artifact removal. Technical Report 173, Max Planck Institute, August 2008. 449

[13] X. Li and M. T. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10:1521–1527, 2001. 449, 450, 455, 456

[14] C. Liu, H. Shum, and W. Freeman. Face hallucination: Theory and practice. 75(1):115–134, 2007. 449

[15] B. S. Morse and D. Schwartzwald. Image magnification using level-set reconstruction. In *CVPR*, volume 1, pages 333 –340, 2001. 449

[16] G. Peyré, S. Bougleux, and L. D. Cohen. Non-local regularization of inverse problems. In *ECCV*, 2008. 450

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 1 edition, 2006. 449, 450, 452

[18] Q. Shan, Z. Li, J. Jia, and C. K. Tang. Fast image/video upsampling. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2008. 449, 451

[19] J. Sun, Z. Xu, and H. Y. Shum. Image super-resolution using gradient profile prior. In *CVPR*, 2008. 449, 453, 454, 455

[20] Y. W. Tai, S. Liu, M. S. Brown, and S. Lin. Super resolution using edge prior and single image detail synthesis. In *CVPR*, 2010. 453, 454

[21] W. S. Tam, C. W. Kok, and W. C. Siu. A modified edge directed interpolation for images. *Journal of Electronic Imaging*, 19(1), 013011:1–20, 2010. 449

[22] Q. Wang, X. Tang, and H. Y. Shum. Patch based blind image super-resolution. In *CVPR*. 449

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error measurement to structural similarity. *IEEE Trans. on IP*, 13(1):600–612, 2004. 454, 455

[24] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 2010. 455