Single Image Super-Resolution using Multi-Task Gaussian Process Regression

JianHong Li[1], Dong Wang[2,], Xiaonan Luo[1]*
1. National Engineering Research Center of Digital Life, School of Information Science & Technology, Sun Yat-sen University, Guangzhou 510006, China
2. College of Information, South China Agricultural University, Guangzhou 510642, China
**. Corresponding author: wdngng@163.com*

Abstract

In this paper, we present a simple and novel approach for solving single image super-resolution (SISR) without external data set. Based on a variant of gaussian process regression (GPR), SISR is formulated as a multi-task regression problem in which each learning task refers to estimation of the regression function for each image patch. Unlike conventional methods, which need to specify the form of the regression function or determine many parameters in the function using inefficient method, the form of regression function in our proposed is implicit defined by the kernel function and all its model parameters can be learned from training set automatically. Experimental results demonstrate that the propose method can preserve fine details and produce natural looking results with sharp edges. Compared with the state of the art algorithms, the results of our proposed is equivalent or superior to them.

1. Introduction

Single image super-resolution (SR) seeks to generate a high-resolution (HR) image from the corresponding low-resolution (LR) input image. The problem is seriously ill posed, since many HR images can generate the same LR image, and thus it is necessary to rely on some competent image priors for estimating the HR image.

Until now, existing SR methods can be classified into three categories: interpolation based methods [1, 2, 3, 4], reconstruction based methods [5, 6, 7] and learning based methods [8, 9, 10, 13, 14, 15, 16, 17]. Interpolation based methods usually apply a basic function or an interpolation kernel to estimate the unknown pixel in the HR grids. These methods are simple and low complexity, but perform poorly near edges in that they are prone to produce blurring artifacts. Reconstruction based methods emphasized the final HR images are obtained via imposing constraints based on a set of prior knowledge in the solution space of the inverse problem [6, 7]. However, the main problem is that the reconstruction edges are usually too piercing to look natural. Moreover, these methods often introduce unpleasant artifacts, such as ringing, in the HR image, especially along salient edges. The basic assumption of the learning based methods is that the high-frequency details lost in an HR image can be learnt from a set of low and high-resolution image pairs. In contrast to other methods, this category of methods generates an HR image from a single LR image, with the help of low and high-resolution image pair as training data. Extensive research results have demonstrated their powerful SR capabilities. Yang et al. [10, 11, 12] proposed using sparse linear combinations to recover the missing high-frequency, allowing a much more compact dictionary model based on sparse coding. However, a huge set of training patches are necessary, resulting in excessively heavy computation cost. Meanwhile, these methods are sensitive to the image training data. Example based SR methods can produce obvious artifacts and unwanted noise into the HR result.

Some recent studies show that natural images generally possess a great amount of self-similarities, i.e., local image structures tend to recur within and across different image scales [13, 15, 16, 17], and image super-resolution can be regularized taken into account these self-similar examples instead of some external database [8, 9, 10]. Particularly, Glasner et al. [15] use self-examples within and across multiple image scales to regularize the ill-posed classical super-resolution problem.

This paper makes use of the original LR input image and its low frequency component version to construct the training data set, without the help of any external image, find out the regression function which is solved by an iterative way to get the optimal values of the global and specific parameters. Then extract feature vector from the up-sampled version of the input image and compute its corresponding HR pixel. At last, scanning the HR image and then calculating the HR image pixelwise.

In summary, the main contribution of this paper is three-fold.

1. Firstly, our work computes out the HR image with simple style and unified framework without any external data set. This promises the generalization of this method.
2. Secondly, our work takes advantage of the global parameters which mine the relative information among patches, and the specific parameter which

978-1-4799-4284-8/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDH.2014.23

keeps each patch be different from others to depict the regression function. This considerable improve the efficiency.

3. At last, the global parameters and the specific parameter are learned from the training set automatically by an iterative way. Except the input LR image, there is no other parameters need to be set manually. So this method is easy to use.

In the rest of this paper, we first review the Gaussian process regression in Section 2. Then we formulate single image super-resolution as a multi-task gaussian process regression problem in Section 3. Section 4 presents our Reconstructed HR results and others produced by state of the art algorithms. Finally, we conclude the paper with some discussion and future developments of the proposed algorithm.

2. The Review of Gaussian Process Regression

Gaussian process regression(GPR) is a powerful tool for accurate function approximation in high dimensional space [18], which is usually formulated as follows: suppose we are given a training set $(x_i, y_i)_{i=1}^N$ with the *i*-th point $x_i \in R^d$ and its corresponding output $y_i \in R$. We want to learn a function f transforming the input vector x_i into the target value y_i given a model $y_i = f(x_i) + \varepsilon_i$, where $\varepsilon_i \sim N(0, \sigma^2)$ [20]. As a result, the observed targets can also be described by a Gaussian distribution $y \sim N(0, K(X,X) + \sigma^2 I)$, where X denotes the data set containing all input points x_i and K denotes the covariance matrix defined on X. The joint distribution of the observed target values $\{y_i\}_{i=1}^N$ and the predicted value $f(x_*)$ for a test point x_* is given by

$$p\begin{pmatrix} y \\ f(x_*) \end{pmatrix} \sim N\left(0, \begin{bmatrix} K + \sigma^2 I & k_* \\ k_*^T & k_\theta(x_*, x_*) \end{bmatrix}\right),$$

where $k_* = (k_\theta(x_*, x_1), ..., k_\theta(x_*, x_N))^T$. According to the joint distribution, the predicted mean value $f(x_*)$ and variance $V(x_*)$ can be computed as

$$f(x_*) = k_*^T (K + \sigma^2 I)^{-1} y,$$

$$V(x_*) = k_\theta(x_*, x_*) - k_*^T (K + \sigma^2 I)^{-1} k_*.$$

The optimal parameter values of GPR for a specified data set can be automatically estimated by maximizing the log marginal likelihood using optimization methods such as Quasi-Newton methods [19].

3. Our Proposed

SISR with GPR [17] was proposed by He etc.. The method first divide the LR image into patches, and for each patch, it constructed the training set, and then train the regression function. Finally, their method computes the HR patch pixelwise. From the above illustration, we can see that all patches take the same method to compute out its corresponding HR patch, and these patches are independent with each other in their method. However, in fact, natural images have many similar patches in its content. In Fig.1, there are various similar patches since the butterfly was symmetrical. Patches with same color frames have symmetry structure and similar texture. When processing the image by taking advantage of the multi-task learning approaches, these patches can help each other to enhance the result.

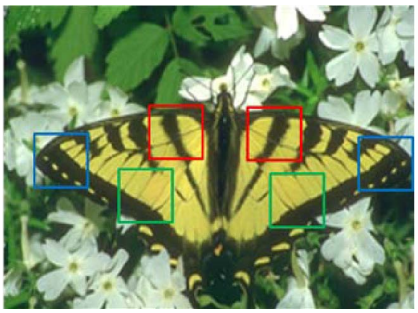


Fig.1 patches with symmetry structure and similar texture. In this image, patches with same color frame have similar texture and structure which can be used to help each other improve the processing result.

3.1. Image Super-Resolution as A Multi-Task Regression Problem

In this paper, we focus on up-sampling a LR image by making use of multi-task learning approach. Fig.2 illustrates the framework of this approach. The input image is denoted as $L_0 \in R^{K_1 \times K_2}$, from which we obtain its low frequency component $L_1 \in R^{K_1 \times K_2}$ by a low pass Gaussian filtering. We up-sample L_0 using bicubic interpolation by a factor of s to get $H_1 \in R^{sK_1 \times sK_2}$. We use H_1 to approximate the low frequency component of the unknown HR image H_0. From L_0, L_1 and H_1, we aim to estimate the HR image H_0. First, we divide L_0 into same size patches (e.g. 32×32) without overlapping. For each image patch p_i (i =1...m), there should be exit a regression function which maps the LR patch p_i to its corresponding HR patch. Finding the regression function can be considered to be a task, and there are *m* tasks in total. Then sample pixels y_i^j in p_i as the targets, where *i* and *j* represents *i*-th patch *j*-th sampled pixel. For each pixel y_i^j,

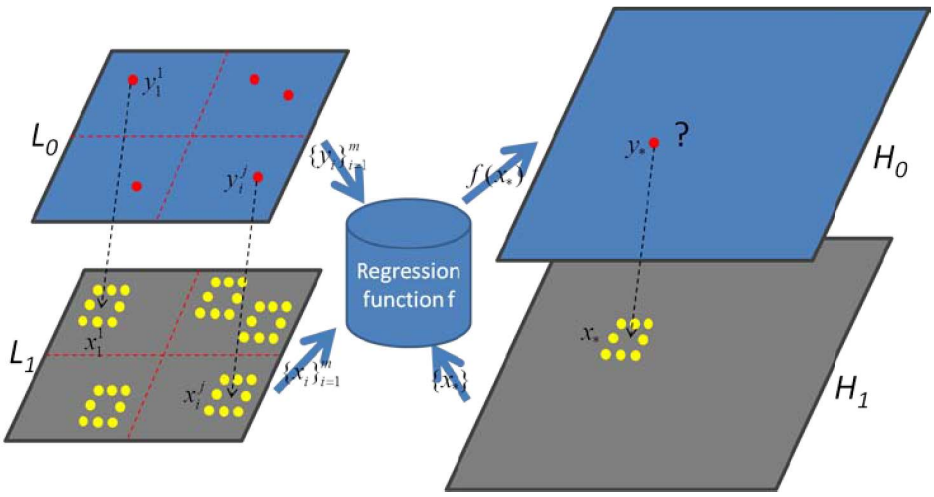


Fig.2 the framework of single image super resolution using multi-task gaussian process regression.

we extract its feature vector x_i^j in L_1, such as the intensity values of its corresponding eight neighborhoods of y_i^j in L_1. $X_i = (x_i^1, ..., x_i^{n_i})$ denotes the data matrix for the *i*-th task, n_i is the number of sampled pixels in p_i, the number of total points is $N = \sum_{i=1}^{m} n_i$. $X = (X_1, ..., X_m)$ denotes the total data matrix for all tasks. According to the m tasks, we wish to find a unified mapping function f which is depicted by some parameters. For each unknown pixel y_* in H_0, we can extract its feature vector x_* in its corresponding low frequency version H_1 as the input of function f to estimate the value of y_*. Finally, scanning H_0 and repeating the above operation, we can obtain the reconstructed HR image.

3.2. The Regression Function

In this method, a latent variable f_i^j is defined for each data point x_i^j and the prior distribution of $f = [f_1^1, ..., f_1^{n_1}, ..., f_m^1, ..., f_m^{n_m}]^T$ can be given as

$$P(f \mid X) \sim N(0, K), \tag{1}$$

where K denotes the covariance matrix defined on X using a kernel function $k_\theta(\cdot,\cdot)$ parameterized by θ. In a real world situation, the targets $\{y_i^j\}_{i=1}^N$ may not satisfy the Gaussian process assumption. Similar to [21, 22], we introduce a latent variable z_i^j which defined as $z_i^j = g(y_i^j)$. The prior distribution for $z = (z_1^1, ..., z_1^{n_1}, ..., z_m^1, ..., z_m^{n_m})^T$ is defined as

$$p(z \mid f) \sim N(f, D), \tag{2}$$

where D denotes a $N \times N$ diagonal matrix. When a data point belongs to the *i*-th task, the corresponding diagonal element of D is set to be σ_i^2, where σ_i defines the noise level of the *i*-th task. This is different from GPR. In GPR, The noise level is identical for all data points, but in our proposed, the noise level is only the same for data points belonging to the same task. Since we assume that different regression functions for different patches share some similarities, we enforce a prior on each σ_i to enforce all σ_i to be close to each other. Because $\sigma_i \in R^+$, thus, we enforce a log-normal prior on them

$$p(\sigma_i) \sim LN(\mu, \rho^2), \tag{3}$$

where $LN(\mu, \rho^2)$ denotes the log-normal distribution with the probability density function as

$$p_t(\mu, \rho^2) = \frac{1}{p(2\pi)^{1/2} t} \exp\left(\frac{-(\ln t - \mu)^2}{2\rho^2} \right).$$

This is equivalent to enforce a normal prior on $\ln \sigma_i$.

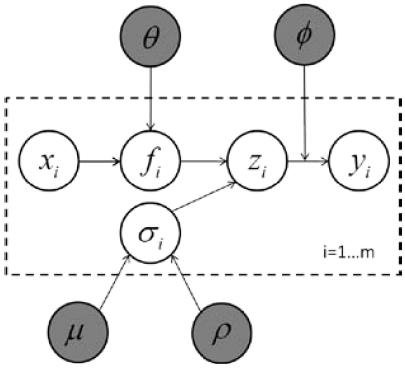
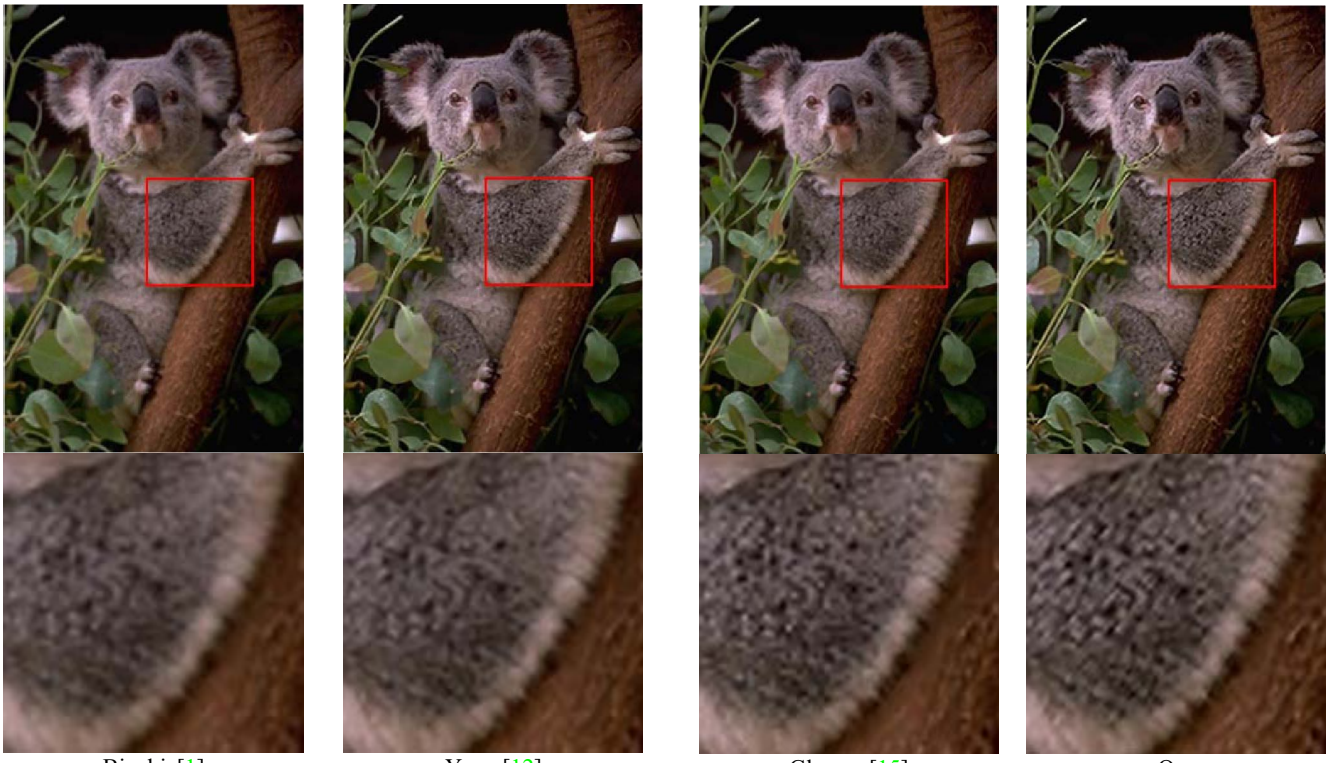


Fig.3 the graphical model of our proposed. X_i denotes the data matrix for the *i*-th task, $y_i = [y_i^1, \cdots, y_i^{n_i}]^T$ is the targets of X_i. $f_i = [f_i^1, ..., f_i^{n_i}]^T$ and $z_i = [z_i^1, ..., z_i^{n_i}]^T$ are latent variables corresponding to X_i and y_i. θ, φ, μ and ρ are global parameters shared by all tasks, and σ_i is specified parameter which model the noise level of *i*-th task.



Bicubic[1] Yang [12] Glasner [15] Our

Figure 4. Super-resolution results(2×) of 'Koala' comparison of bicubic interpolation, Yang's single image super resolution with sparse prior and compact dictionary [12], Glasner's single image super resolution using redundant patches across scales [15] and our multi-task Gaussian process regression method.

In summary, Eq. (1), Eq. (2) and Eq. (3) are sufficient to define the entire model of our proposed. Model parameters including θ, φ, μ and ρ which we called global parameters are shared by all tasks corresponding to the feature vectors, and σ_i which we called specific parameter model the noise level of *i*-th task. The graphical model for our proposed is depicted in Figure 3.

According to Eq. (1) and Eq. (2), we can get

$$p(z \mid X) = \int p(z \mid f)p(f, X) = N(0, K + D) . \quad (4)$$

Since $z_i^j = g(\mathrm{y}_i^j)$ and $\{\sigma_i\}_{i=1}^m$ share a log-normal prior, the regression function for the training data can be wrote out as

$$p(\mathrm{y} \mid \mathrm{X}) = N(g(y) \mid 0, K + D) \cdot \prod_{i=1}^{m} \prod_{j=1}^{n_i} g'(y_i^j) \cdot \prod_{i=1}^{m} LN(\sigma_i \mid \mu, \rho^2) , \quad (5)$$

where $g(y) = (g(y_1^1), ..., g(y_1^{n_i}), ..., g(y_m^1), ..., g(y_m^{n_m}))^T$.

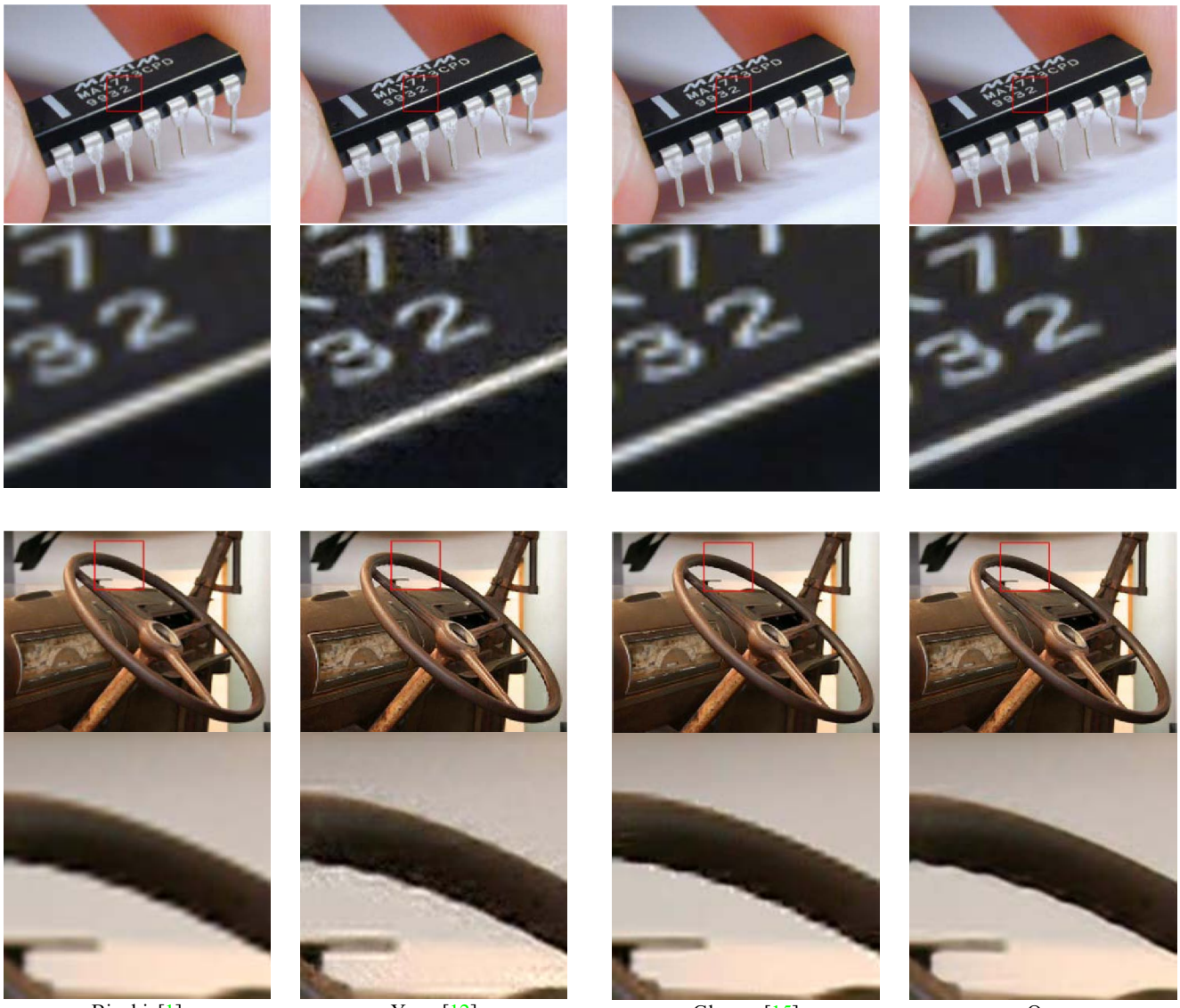
3.3. Model Parameters Learning

In our proposed, we maximize the regression function to obtain the optimal values of these parameters. For computing stability, we solve an equivalent problem by minimizing the negative log regression functions which is given as follows

$$\arg\min_{\{\sigma_i\},\theta,\varphi,\mu,\rho} \frac{1}{2}\left[g(y)^T (K + D)^{-1} g(y) + \ln\left|K + D\right|\right] - \sum_{i=1}^{m}\sum_{j=1}^{n_i} \ln g'\left(y_i^j\right) + \sum_{i=1}^{m}\left[\ln \sigma_i + \ln \rho + \frac{(\ln \sigma_i - u)^2}{2\rho^2} \right] .$$

Here we use the letter 'R' to represent this function, and minimize the function with gradient descent algorithm. The gradients of equation R with respect to all model parameters are as follows

$$\frac{\partial R}{\partial \sigma_i} = \sigma_i tr\left(\left[\left(K + D\right)^{-1} - \left(K + D\right)^{-1} g(y) g(y)^T \left(K + D\right)^{-1}\right] I_N^i\right) + \frac{\rho^2 + \ln \sigma_i - \mu}{\sigma_i \rho^2} ,$$



Bicubic[1] Yang [12] Glasner [15] Our

Figure 5. Super-resolution results(3×) of 'chip' and 'wheel' comparison of bicubic interpolation, yang's single image super resolution with sparse prior and compact dictionary [12], Glasner's single image super resolution using redundant patches across scales [17] and our multi-task Gaussian process regression method.

$$\frac{\partial R}{\partial \theta_q} = \frac{1}{2} tr\left(\left[(K+D)^{-1} - (K+D)^{-1} g(y)^T (K+D)^{-1} \right] \frac{\partial K}{\partial \theta_q} \right),$$

$$\frac{\partial R}{\partial \varphi_q} = g(y)^T (K+D)^{-1} \frac{\partial g(y)}{\partial \varphi_q} - \sum_{i=1}^{m} \sum_{j=1}^{n_i} \frac{\partial \ln g'(y_j^i)}{\partial \varphi_u},$$

$$\frac{\partial R}{\partial \mu} = \frac{m\mu - \sum_{i=1}^{m} \ln \sigma_i}{\rho^2}, \quad \frac{\partial R}{\partial \rho} = \frac{m}{\rho} - \frac{\sum_{i=1}^{m} (\ln \sigma_i - \mu)}{\rho^3}.$$

where I_N^i denotes an N×N binary diagonal matrix where a diagonal element is equal to 1 when the data point with the corresponding index belongs to the i-th task, φ_q is the q-th element of φ. In our experiments, the function g has the form $g(x) = a \ln(bx + c) + d$ where $a, b, c \in R^+$ and $d \in R$. So $\varphi = [a, b, c, d]^T$.

Since the number of model parameters is not small, we take an iterative way to optimize R. For detailed, we first update $\{\sigma_i\}_{i=1}^m$ using gradient descent algorithm with the other model parameters fixed and then use gradient descent algorithm to update the other model parameters θ, φ, μ and ρ with $\{\sigma_i\}_{i=1}^m$ fixed. These two steps are alternative until convergence.

3.4. Prediction

Since H_1 is the low frequency component of H_0, and the relationship between them is similar to the relationship of L_0 and L_1. For each unknown pixel y_* in H_0, we first extract it's feature vector x_* in H_1. We define z_* as $z_* = g(y_*)$. Since we do not know which task x_* belongs to, we introduce a new noise level σ_* for x_* From Eq. (4), we can get

$$p\begin{pmatrix} z \\ z_* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K+D & k_* \\ k_*^T & k_*(x_*, x_*)+\sigma_*^2 \end{pmatrix}\right),$$

where $k_* = (k_\theta(x_*, x_1^1), ..., k_\theta(x_*, x_m^{n_m}))^T$. Then, the predictive distribution $p(z_* \mid x_*, X, y)$ can be computed as a Gaussian distribution with the following mean m_* and variance ρ_*^2

$$m_* = (k_*)^T (K+D)^{-1} z,$$
$$\rho_*^2 = k_\theta(x_*, x_*) + \sigma_*^2 - k_*^T (K+D)^{-1} k_*.$$

Here, we take m_* as the predicted result for z_*. From expression of m_*, we can see that m_* is independent on σ_i, so for any test data point, it is unnecessary to classify the test data point into some task. This property is very beneficial because some test data points may not even belong to any of the tasks in the training set in many practical applications. According to the relationship between y and z that $z = g(y)$ and $z_* = g(y_*)$, we can get the prediction for the output of x_* as

$$y_* = g^{-1}\left((k_*)^T (K+D)^{-1} g(y)\right). \qquad (6)$$

Scanning the whole HR image and repeating the above operation, we can compute out the HR image pixelwise.

4. Experiments

Experiments have been conducted to evaluate our method in comparison with several state-of-the-art algorithms, which include Bicubic [1], Yang [12], Glasner [15] and GPR [17]. We start by using the original image as LR input and upsample it with a scale factor of 2. For further up-sampling, we used the previous output as the input and solved its HR image. Note that, for a color image, we first transform the image from RGB to YIQ. Then, the Y channel (intensity) is up-sampled by our algorithm because human vision is more sensitive to brightness. I and Q channels are interpolated by the bicubic method. Finally, the three channels are combined to form the final HR result. Both qualitative and quantitative methods are used in order to evaluate our method. For quantitative evaluation, we use PSNR and SSIM to measure super-resolution results. The better super-resolution result should provide larger PSNR and SSIM values.

Figure 4 displays the results of image 'Koala' with 2 scale factor. We can notice that: the bicubic result of 'Koala' produces poor edge and many blurring artifacts. The result of Glasner method also produces many visual artifacts. The results of Yang and our proposed are better than the other two, and from the red close-up, we can see that our result have more pleased texture and sharper edge than Yang's.

Figure 5 illustrates the results of images 'chip' and 'wheel' with 3 scale factor. Our results have sharper edges and richer texture than the other three, especially in the closed-up of image 'wheel'. From the results of 'wheel', we can see that the result of bicubic and Glasner produce blurring edges, and Yang' result has more artifacts along the edge.

Table.1 provides the numerical comparison of our method with bicubic [1], Yang [12], Glasner [15] and GPR [17]. The test images are popular examples image processing field. For each method, there are two rows. The first row is PSNR, the second row is SSIM.

Table 1. PSNR and SSIM for 3 scale factor

	Bicubic [1]	Yang [12]	Glasner [15]	GPR [17]	Our
Lena	30.8492	31.3239	30.3197	30.4197	**31.3537**
	0.8653	0.8659	0.8672	0.8695	**0.8953**
Baboon	21.8617	21.9884	20.6943	22.0003	**22.1031**
	0.5429	0.5936	0.4782	0.5731	**0.5779**
Peppers	29.5054	**29.8484**	29.4152	29.6830	29.7135
	0.8469	**0.8498**	0.8353	0.8541	0.8473
Airplane	28.4908	28.3233	27.8095	27.7303	**28.3573**
	0.8726	0.8810	0.8196	0.8831	**0.8913**
Koala	29.1492	29.5286	28.6700	29.1361	**29.1559**
	0.8009	0.8162	0.7064	0.8131	**0.8236**
Baby	32.6288	32.3168	32.5818	**32.6504**	32.5896
	0.8931	0.8882	0.8393	**0.8996**	0.8964

5. Conclusion

In this paper, we present a simple and novel algorithm to solve SISR problem. Supported by multi-task gaussian process regression, our method mines the relative information among patches with global parameters and keep each patch be different from others by specified parameter to computer the reconstructed HR image pixelwise. Experimental results demonstrate that the proposed method can preserve fine details and produce natural looking results with sharp edges. In the future, we plan to adapt the method for the problem of video super-resolution.

Acknowledgement

This research is supported by the National Science Foundation of China (61320106008, 61202294), and the Science and Technology Planning Project of Guangzhou (2013J4300059).

References

[1] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," IEEE Trans. Signal Process., vol. 26, no. 6, pp. 508–517, Dec. 1978.

[2] X. Li and M. T. Orchard, "New edge-directed interpolation," IEEE Trans. Image Process., vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[3] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," IEEE Trans. Image Process., vol. 15, no. 8, pp. 2226–2238, Aug. 2006.

[4] M. Li and T. Nguyen, "Markov random field model-based edge-directed image interpolation," IEEE Trans. Image Process., vol. 17, no. 7, pp. 1121–1128, Jul. 2008.

[5] Fattal R. Image upsampling via imposed edge statistics[C]. ACM Transactions on Graphics (TOG). ACM, 2007, 26(3): 95.

[6] Shan Q, Li Z, Jia J, et al. Fast image/video upsampling[C]. ACM Transactions on Graphics (TOG). ACM, 2008, 27(5): 153.

[7] Jian Sun, Zongben Xu, and Heung Shum. Image super-resolution using gradient profile prior. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, June 2008.

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

[9] Chang H, Yeung D Y, Xiong Y. Super-resolution through neighbor embedding[C]. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. IEEE, 2004, 1: I-I.

[10] Yang J, Wright J, Huang T, et al. Image super-resolution as sparse representation of raw image patches[C]. Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008: 1-8.

[11] Yang J, Wright J, Huang T S, et al. Image super-resolution via sparse representation[J]. Image Processing, IEEE Transactions on, 2010, 19(11): 2861-2873.

[12] Yang J, Wang Z, Lin Z, et al. Coupled dictionary training for image super-resolution[J]. Image Processing, IEEE Transactions on, 2012, 21(8): 3467-3478.

[13] Yang J, Lin Z, Cohen S. Fast image super-resolution based on in-place example regression[C]. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013: 1059-1066.

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

[15] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In IEEE International Conference on Computer Vision (ICCV), pages 349–356, 2009.

[16] Freedman G, Fattal R. Image and video upscaling from local self-examples[J]. ACM Transactions on Graphics (TOG), 2011, 30(2): 12.

[17] He H, Siu W C. Single image super-resolution using Gaussian process regression[C]. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011: 449-456.

[18] Nguyen-Tuong D, Seeger M, Peters J. Model learning with local gaussian process regression[J]. Advanced Robotics, 2009, 23(15): 2015-2034.

[19] Dennis, Jr J E, Moré J J. Quasi-Newton methods, motivation and theory[J]. SIAM review, 1977, 19(1): 46-89.

[20] C. E. Rasmussen and C. K. Williams, Gaussian Process for Machine Learning. Massachusetts Institute of Technology: MIT-Press, 2006.

[21] Zhang Y, Yeung D Y. Multi-task warped gaussian process for personalized age estimation[C]. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 2622-2629.

[22] E. Snelson, C. E. Rasmussen, and Z. Ghahramani. Warped Gaussian processes. In NIPS 16, 2004.