Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 435924, 13 pages
doi:10.1155/2012/435924

Research Article

Energy-Driven Image Interpolation Using Gaussian Process Regression

Lingling Zi[1,2] and Junping Du[1]

[1] *Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China*

Correspondence should be addressed to Junping Du, junpingdu@126.com

Received 1 March 2012; Accepted 27 April 2012

Academic Editor: Baocang Ding

Image interpolation, as a method of obtaining a high-resolution image from the corresponding low-resolution image, is a classical problem in image processing. In this paper, we propose a novel energy-driven interpolation algorithm employing Gaussian process regression. In our algorithm, each interpolated pixel is predicted by a combination of two information sources: first is a statistical model adopted to mine underlying information, and second is an energy computation technique used to acquire information on pixel properties. We further demonstrate that our algorithm can not only achieve image interpolation, but also reduce noise in the original image. Our experiments show that the proposed algorithm can achieve encouraging performance in terms of image visualization and quantitative measures.

1. Introduction

Image interpolation is a very important aspect of image processing and involves the use of a known pixel set to produce an unknown pixel set, resulting in an image of higher resolution [1, 2]. This technique is widely used in remote sensing, aerospace, infrared imaging, low-light level night imagery, and other fields [3–5]. However, maintaining image quality during image interpolation is still a difficult issue [6]. To address this, many image interpolation methods have been proposed. For example, traditional bilinear interpolation computes the unknown pixel value using the location information between the adjacent pixels. This technique does not consider the contents of the image, so edge blurring will occur in the interpolated image [7, 8]. In order to capture image details more clearly, an artifact-free image upscaling method called ICBI [9] has recently been proposed, which uses

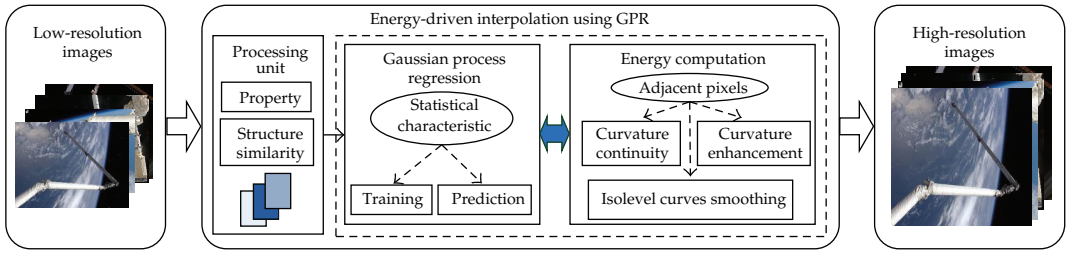


Figure 1: Overview of our approach for image interpolation.

iterative curvature-based interpolation to obtain a high image quality, but does not take into account underlying local information between image patches. Local image information can be mined according to its structural redundancy characteristic, as proposed by Glasner et al. [10]. This characteristic can lay the foundations for the training and predicting of a statistical model [11, 12]. A statistical model known as Gaussian process regression (GPR) was first applied in the reconstruction of high-resolution images in 2011 and has been shown to be capable of generating an image with sharp edges by extracting the necessary information from a low-resolution image [13]. However, it should be noted that this method only uses the local structural information for each pixel's neighborhood, so it can still generate unexpected details. To develop the above techniques, we propose here a novel energy-driven interpolation algorithm employing Gaussian process regression (EGPR) (Figure 1). This algorithm not only emphasizes the influence of adjacent pixel properties on interpolated values, but also brings into full play the role of the statistical model.

Our contribution is twofold. Firstly, we propose a framework for both magnification and deblurring in order to fulfill the interpolation task for low-resolution images with low noise. Secondly, we demonstrate an energy-driven approach based on the properties of adjacent pixels within this framework. In addition, we define the processing unit and its properties for better implementation of the EGPR algorithm.

The rest of the paper is structured as follows. Section 2 discusses GPR. Section 3 illustrates the proposed EGPR algorithm. Section 4 presents experimental work carried out to demonstrate the effectiveness of our algorithm. Section 5 concludes the paper.

2. Gaussian Process Regression

In recent years, GPR has become a hot issue in the field of machine learning and has attracted great academic interest [14–16]. It has many advantages, including its rigorous underlying statistical learning theory, easy regression process implementation, few parameters, and improved model interpretability [17–19]. As a result of these benefits, it has been used in many areas [20–23]; however, to the best of our knowledge, it has not yet been fully utilized in image interpolation. Rasmussen and Williams [24] defined the Gaussian process and noted in particular that a Gaussian process is completely specified by its mean and covariance functions ((2.1) and (2.2), resp.):

$$\mu(x) = E[Y(x)], \tag{2.1}$$

$$\mathrm{COV}(x, x') = E\left[\left(Y(x) - \mu(x)\right)\left(Y(x') - \mu(x')\right)\right], \tag{2.2}$$

where x and x' are any random variables. In particular, they could represent n-dimensional input or output vectors. The Gaussian process can be written as follows:

$$g(x) \sim GP(\mu(x), \mathrm{COV}(x, x')). \tag{2.3}$$

There are a variety of covariance functions, of which one of the most commonly used is the squared exponential (SE) covariance function

$$\mathrm{COV}(g(x_p), g(x_q)) = \exp\left(-\frac{1}{2}\left|x_p - x_q\right|^2\right). \tag{2.4}$$

In Gaussian processes, the marginal likelihood $p(y \mid X)$ at a point is very useful and is the integral of the likelihood multiplied by the prior probability

$$p(y \mid X) = \int p(y \mid g, X) p(g \mid X) dg. \tag{2.5}$$

We can rewrite (2.5) as follows:

$$\log p(g \mid X) = -\frac{1}{2} g^T \mathrm{COV}^{-1} g - \frac{1}{2} \log|\mathrm{COV}| - \frac{n}{2} \log 2\pi. \tag{2.6}$$

We can make use of Gaussian identities to obtain (2.7), in order to compute the log marginal likelihood. The conjugate gradients method has been applied to solve this equation. Using this approach, we can obtain the hyperparameters of the covariance function. Further details of GPR can be found in [24] as follows:

$$\log p(g \mid X) = -\frac{1}{2} y^T \left(\mathrm{COV}(X, X) + \sigma_n^2 I\right)^{-1} y - \frac{1}{2} \log\left|\mathrm{COV}(X, X) + \sigma_n^2 I\right| - \frac{n}{2} \log 2\pi. \tag{2.7}$$

3. The Proposed Algorithm

In this paper, we combine the energy-driven approach with GPR to accomplish the task of image interpolation. The proposed algorithm models low-resolution image data as a function of a probability distribution that satisfies a local static Gaussian process. This algorithm framework is shown in Figure 2 and is broadly divided into the training process and prediction process. Firstly, the GPR model can be established using the low-resolution image data. Next, this model is used to predict the unknown pixel values of a high-resolution image by adopting an energy computation approach. Through the above two steps, we produce a high-quality enlarged image. The process is further clarified by the following:

$$L' = d * L, \qquad H' = L'\uparrow^s, \qquad H = f * H', \tag{3.1}$$

where L and H denote the input low-resolution image with a little noise and the output high-resolution image, respectively, L' denotes the noise-free low-resolution image, H' denotes the initial high-resolution image, $s$denotes the upsampling factor, and d and f denote the clear transfer function and energy transfer function, respectively.

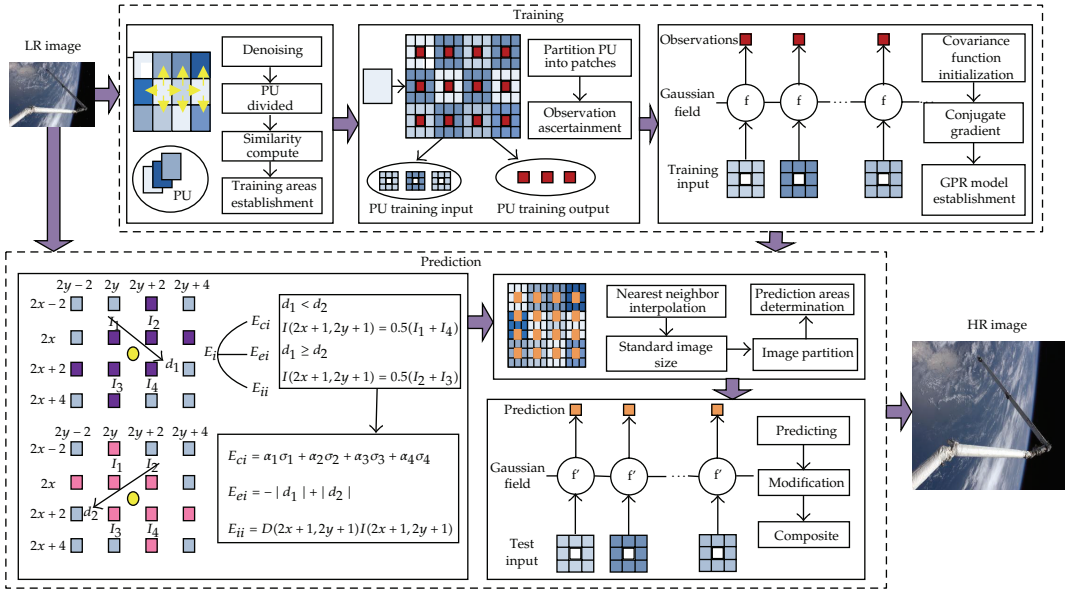


Figure 2: Architecture of the proposed algorithm.

3.1. Training

The following definitions are used in the EGPR algorithm.

Definition 3.1. A given image L is divided into many regions of equal size, and each region is defined as a processing unit (PU). Each PU is also divided into 3×3 overlapping image patches (the total number is M). The center of each patch is defined as an output vector Y_{TR} of PU, where $Y_{TR} = (y_1, y_2, \ldots y_M)^T$, while the nearest eight values are defined as an input vector X_{TR} of PU, where

$$X_{TR} = \left\{ \begin{array}{c} x_{11}, x_{12} \ldots, x_{18} \\ x_{21}, x_{22} \ldots, x_{28} \\ \ldots. \\ x_{M1}, x_{M2} \ldots, x_{M8} \end{array} \right\}. \tag{3.2}$$

Definition 3.2. Given a total of N pixels in each PU, the pixels are sorted and denoted as $I_1, I_2 \ldots I_N$. $I_{\max\ ave}$, $I_{\min\ ave}$, and I_{ave} are defined using the following formulae:

$$I_{\max\ ave} = \sum_{i=1}^{\text{Top}} \frac{I_i}{\text{Top}}, \qquad I_{\min\ ave} = \sum_{i=1}^{\text{Below}} \frac{I_i}{\text{Below}}, \qquad I_{ave} = \sum_{i=1}^{N} \frac{I_i}{N}, \tag{3.3}$$

where top represents the number of the largest pixels used and below the number of the smallest pixels used. Then $I_{\max\ ave}$, $I_{\min\ ave}$, and I_{ave} are called basic properties of PU.

To facilitate the operation of the PU, it is necessary to introduce some properties in advance.

Property 1. Given a number N in each PU, if $I_{\max\ \mathrm{ave}} = 0$, then pixel value $I_i = 0$, where $i \leq N$.

Property 2. Given x_{ij}, if $x_{ij} = a$, then its corresponding output vector value is $y_{ij} = a$, where $y_{ij} \in Y_{TR}, i \in M, j = 1$.

Denoising is the first step in the EGPR algorithm, and we use the following formula (3.4) to obtain noise-free images:

$$I_i = \begin{cases} I_{\mathrm{neighbor}}, & (I_{\max\ \mathrm{ave}} - I_{\min\ \mathrm{ave}}) \langle \theta \,\&\, I_i \rangle (I_{\mathrm{ave}} + B), \\ I_i, & \text{otherwise}, \end{cases} \tag{3.4}$$

where θ and B represent empirical values, and I_{neighbor} represents the adjacent pixel value.

Before applying GPR, we can obtain the particular relationship between the input and output vectors of PU according to Properties 1 and 2. Pixels with this relationship need not be included in the following GPR training, so the predicted values can be directly obtained, saving time and speeding up the EGPR algorithm.

Training plays an important role in the EGPR algorithm, and we adopt a different approach from that used in [13]. Our algorithm contains two processes: training domain establishment and GPR model foundation. In the first stage, we search possible training domains along the four directions of each specific PU. Next, we compute the structural similarity between directions to determine the definite training domain. Inspired by the concept of image SSIM, we define the PU structural similarity as follows.

Definition 3.3 (*PU structural similarity*). Given two processing units P and Q, their structural similarity is defined as

$$S(P,Q) = \frac{(2m_P m_Q + C_1)(2\varphi_{PQ} + C_2)}{\left(m_P^2 + m_Q^2 + C_1\right)\left(\varphi_P^2 + \varphi_Q^2 + C_2\right)}, \tag{3.5}$$

where C_1 and C_2 are constants, and the other components are calculated as follows:

$$m_P = \frac{1}{N}\sum_{i=1}^{N} p_i, \quad p_i \in P, \qquad m_Q = \frac{1}{N}\sum_{i=1}^{N} q_i, \quad q_i \in Q, \qquad \varphi_P = \left(\frac{1}{N-1}\sum_{i=1}^{N}(p_i - m_p)^2\right)^{1/2},$$
$$\varphi_Q = \left(\frac{1}{N-1}\sum_{i=1}^{N}(q_i - m_q)^2\right)^{1/2}, \qquad \varphi_{PQ} = \frac{1}{N-1}\sum_{i=1}^{N}(p_i - m_p)(q_i - m_q). \tag{3.6}$$

When the search step count reaches the predefined number, or if the PU structure similarity falls below a certain value, the first stage is complete. In the second stage, we apply a Gaussian process prior probability and establish the GPR model with Gaussian noise γ (see (3.7) below) using the image data from training domains. In (3.7), "GP" denotes a Gaussian process

$$y = g(X) + \gamma, \quad \gamma \sim GP\left(0, \sigma_n^2\right). \tag{3.7}$$

(a) Iteration = 50 (b) Iteration = 100

Figure 3: Images obtained after adaptation with different numbers of iterations. In (a), many black points are observed, each indicating a zero prediction for the pixels. In (b), the black points have been eliminated.

When aiming to achieve high-quality images, the conjugate gradients method is chosen to obtain the model hyperparameters, including mean, variance, and log marginal likelihood. Notice that different iteration numbers in the conjugate gradients method may lead to different prediction accuracies. Figure 3 shows the interpolation images obtained after 50 iterations and 100 iterations, where it can be seen that the latter is better than the former.

3.2. Prediction

Inspired by the ICBI algorithm, we firstly compute the initial pixel value $H'(2x+1, 2y+1)$ according to the following formulae:

$$\begin{aligned} \tau_1 &= I(2x+4, 2y) + I(2x+2, 2y-2) + I(2x, 2y+4) + I(2x-2, 2y+2), \\ \tau_2 &= I(2x+4, 2y+2) + I(2x+2, 2y+4) + I(2x, 2y-2) + I(2x-2, 2y), \end{aligned} \tag{3.8}$$

$$\begin{aligned} d_1(2x+1, 2y+1) &= \tau_1 + I(2x+2, 2y+2) + I(2x, 2y) - 3\big(I(2x, 2y+2) + I(2x+2, 2y)\big), \\ d_2(2x+1, 2y+1) &= \tau_2 + I(2x+2, 2y) + I(2x, 2y+2) - 3\big(I(2x, 2y) + I(2x+2, 2y+2)\big), \end{aligned} \tag{3.9}$$

$$H'(2x+1, 2y+1) = \begin{cases} \dfrac{1}{2}\big(I(2x, 2y) + I(2x+2, 2y+2)\big), & d_1(2x+1, 2y+1) < d_2(2x+1, 2y+1), \\ \dfrac{1}{2}\big(I(2x+2, 2y) + I(2x, 2y+2)\big), & d_1(2x+1, 2y+1) \geq d_2(2x+1, 2y+1). \end{cases} \tag{3.10}$$

However, the pixel value obtained is only a roughly estimated value and needs further refinement. Following [9], we establish (3.11) to calculate the energy of each interpolated pixel, and the initial estimate can be modified accordingly,

$$E(2x+1,2y+1) = cE_c(2x+1,2y+1) + eE_e(2x+1,2y+1) + iE_i(2x+1,2y+1), \quad (3.11)$$

where c, e, and i are chosen to adjust the energy contributions from the three parts.

E_c represents the curvature continuity energy and can be computed with the following formulae:

$$\begin{aligned}
\sigma_1 &= \left|d_1(2x,2y) - d_1(2x+1,2y+1)\right| + \left|d_2(2x,2y) - d_2(2x+1,2y+1)\right|,\\
\sigma_2 &= \left|d_1(2x,2y) - d_1(2x+1,2y-1)\right| + \left|d_2(2x,2y) - d_2(2x+1,2y-1)\right|,\\
\sigma_3 &= \left|d_1(2x,2y) - d_1(2x-1,2y+1)\right| + \left|d_2(2x,2y) - d_2(2x-1,2y+1)\right|,\\
\sigma_4 &= \left|d_1(2x,2y) - d_1(2x-1,2y-1)\right| + \left|d_2(2x,2y) - d_2(2x-1,2y-1)\right|,\\
E_c(2x+1,2y+1) &= \alpha_1\sigma_1 + \alpha_2\sigma_2 + \alpha_3\sigma_3 + \alpha_4\sigma_4,
\end{aligned} \quad (3.12)$$

where $\alpha_i (i = 1 \ldots 4)$ are weight values (see (3.13)), and d_1 and d_2 have the same meanings as above. θ is set as the threshold

$$\alpha_i = \begin{cases} 1 & \text{if } \sigma_i < \theta, \\ 0, & \text{otherwise.} \end{cases} \quad (3.13)$$

The second energy term E_b represents the curvature enhancement energy and can be computed by

$$E_e(2x+1,2y+1) = \left|d_2(2x+1,2y+1)\right| - \left|d_1(2x+1,2y+1)\right|. \quad (3.14)$$

The third energy term, E_i, represents the isolevel curves smoothing energy and can be computed with

$$E_i(2x+1,2y+1) = D(2x+1,2y+1)I(2x+1,2y+1), \quad (3.15)$$

where $D(2x+1,2y+1)$ can be computed as follows:

$$\begin{aligned} D(2x+1,2y+1) = {} & \frac{2d_3(2x+1,2y+1)d_4(2x+1,2y+1)d_5(2x+1,2y+1)}{d_3(2x+1,2y+1)^2 + d_4(2x+1,2y+1)^2} \\ & + \frac{-d_3(2x+1,2y+1)^2 d_2 - {d_2}^2 d_3(2x+1,2y+1)}{d_3(2x+1,2y+1)^2 + d_4(2x+1,2y+1)^2}, \\ d_3(2x+1,2y+1) = {} & \frac{1}{2}(I(2x,2y) - I(2x+2,2y+2)), \\ d_4(2x+1,2y+1) = {} & \frac{1}{2}(I(2x,2y+2) - I(2x+2,2y)), \\ d_5(2x+1,2y+1) = {} & \frac{1}{2}(I(2x+1,2y-1) + I(2x+1,2y+3) \\ & - I(2x-1,2y+1) - I(2x+3,2y+1)). \end{aligned} \tag{3.16}$$

Suppose that the low-resolution image L_{ij} is of size $m \times n$ and that it is changed to the corresponding interpolated image $H_{ij'}$ of size $((m \times 2^{\text{scale}}) - (2^{\text{scale}} - 1)) \times ((n \times 2^{\text{scale}}) - (2^{\text{scale}} - 1))$, where "scale" denotes the magnification factor. Then we use the nearest interpolation algorithm for the missing pixels in order to obtain the image $H_{ij'}$ of size $(m \times 2^{\text{scale}}) \times (n \times 2^{\text{scale}})$.

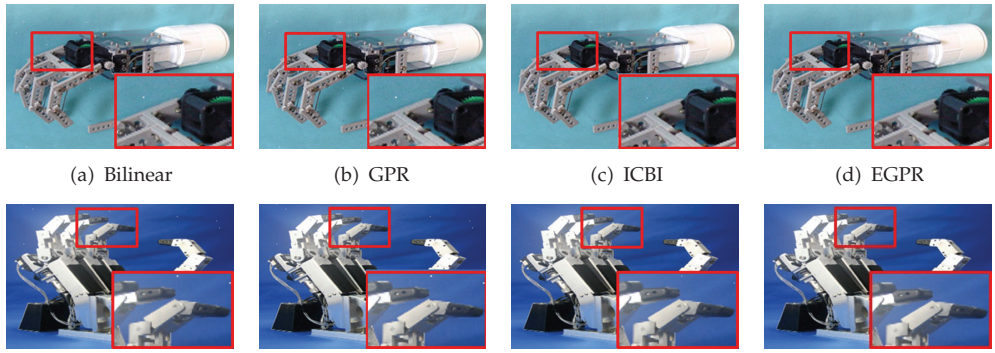
Similarly, we partition $H_{ij'}$ into overlapping processing units. The eight adjacent pixels of each pixel are treated as GPR model test input data based on the model M which was obtained from the training process. Note that if PU has Property 1 or Property 2, then we can directly obtain the corresponding pixel values. Otherwise, we capture the prediction distribution of unknown pixels in the initial high-resolution image. The joint distribution of the training domain output y and the test output f' is given by the following equation:

$$\begin{bmatrix} y \\ g' \end{bmatrix} \sim GP\left(0, \begin{bmatrix} \text{COV}(X,X) + \sigma_n^2 I, \text{COV}(X,X') \\ \text{COV}(X',X), \text{COV}(X',X') \end{bmatrix}\right), \tag{3.17}$$

where X denotes the GPR training data matrix, X' is the test matrix, and $\text{COV}(X,X')$ is the $n \times n$ matrix of covariances. Therefore, we can derive the predictive distribution based on the obtained model M:

$$\begin{aligned} g' \mid X, y, X' &\sim GP\left(\overline{g'}, V\left(\overline{g'}\right)\right), \\ \overline{g'} &= \text{COV}(X',X)\left[\text{COV}(X,X) + \sigma_n^2 I\right]^{-1} y, \\ V\left(\overline{g'}\right) &= \text{COV}(X',X') - \text{COV}(X',X)\left[\text{COV}(X,X) + \sigma_n^2 I\right]^{-1} \text{COV}(X,X'). \end{aligned} \tag{3.18}$$

During the prediction of high-resolution image pixels, two rules should be obeyed. Firstly, the PU divided by the initial high-resolution image should correspond to that divided by the low-resolution image. Secondly, the gradient algorithm should satisfy the common positive definite matrix. If not, it will lead to a zero prediction, and the prediction value



(a) Bilinear (b) GPR (c) ICBI (d) EGPR

(e) Bilinear (f) GPR (g) ICBI (h) EGPR

Figure 4: Comparison of images obtained using four methods, with scale = 1. Parts (a)–(d) show image 1. Parts (e)–(h) show image 2.

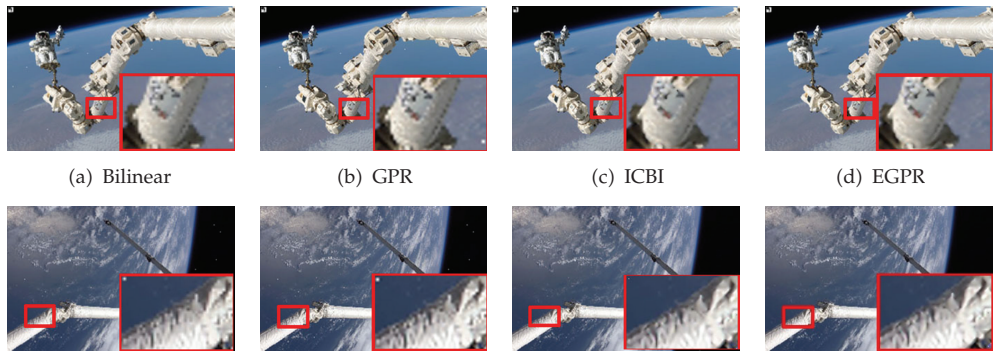
will need modifying. The modification method can be utilized to maintain the original interpolated pixel value. Finally, we combine all the processing units together in a smooth manner to obtain the high-resolution images without noise.

4. Experimental Results and Discussion

In this section, we compare the experimental results obtained using the proposed algorithm with those obtained using the bilinear algorithm, GPR algorithm [13], and ICBI algorithm [9]. Each algorithm was run in MATLAB. In order to evaluate algorithm performance, we first downsampled original high-quality images to acquire low-resolution images. Then we enlarged these low-resolution images by utilizing the different interpolation algorithms and compared the enlarged images with the original high-quality images. In all experiments, we set the PU size to 30×30, but this may be increased according to the magnification factor. At the same time, we used zero mean and square exponential functions as the respective mean and covariance functions in the EGPR. The covariance function required two hyperparameters: a characteristic length scale, the default value of which was 0.21, and the standard deviation of the signal, the default value of which was 0.08. In addition, to achieve color image interpolation, we trained and predicted the GPR model separately for each of the R, G, and B channels.

Figure 4 shows the interpolation results from the four algorithms when "scale" was set as 1. Figures 4(a)–4(d) are comparisons of image 1, and Figures 4(e)–4(h) are comparisons of image 2. In the enlarged red-bordered region, it can be seen that the bilinear method introduces jaggy effects, the GPR method reduces these jaggy effects, and the ICBI method achieves a clear edge but is still a little blurry. By employing the energy computation based on properties of adjacent pixels, our new method generates a clearer image without noise.

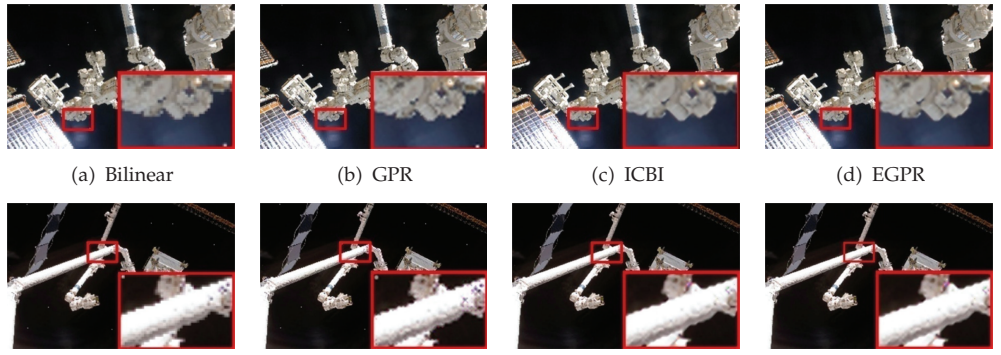
Similarly, Figures 5 and 6 demonstrate the interpolation results with scales of 2 and 3, respectively. From these figures, it can be seen that our method achieved the clearest and smoothest enlarged image of the four methods tested, for example, along edges on the root hand in Figure 6(h). Moreover, the advantages of our proposed algorithm become more enhanced at greater enlargement factors.



(a) Bilinear (b) GPR (c) ICBI (d) EGPR

(e) Bilinear (f) GPR (g) ICBI (h) EGPR

Figure 5: Comparison of images obtained using four methods, with scale = 2. Parts (a)–(d) show image 3. Parts (e)–(h) show image 4.



(a) Bilinear (b) GPR (c) ICBI (d) EGPR

(e) Bilinear (f) GPR (g) ICBI (h) EGPR

Figure 6: Comparison of images obtained using four methods, with scale = 3. Parts (a)–(d) show image 5. Parts (e)–(h) show image 6.

Table 1: Comparison of PSNR for the four interpolation methods when applied to test images.

Image	Scale	Bilinear	GPR	ICBI	EGPR
Image 1	1	32.9940	33.2792	33.3456	33.3986
Image 2	1	30.6314	30.7861	31.3684	31.4594
Image 3	2	29.5738	29.4194	29.7173	29.7213
Image 4	2	27.7717	27.4767	27.8485	27.8625
Image 5	3	23.4038	24.4366	24.7153	24.7171
Image 6	3	24.3122	25.1477	25.6880	25.6909

Table 2: Comparison of RMS for the four interpolation methods when applied to test images.

Image	Scale	Bilinear	GPR	ICBI	EGPR
Image 1	1	16.4437	15.8419	15.7032	15.6004
Image 2	1	21.1046	20.3410	19.1890	18.9614
Image 3	2	24.9225	24.8861	24.4329	24.4191
Image 4	2	31.0516	32.1427	30.6882	30.6118
Image 5	3	50.8161	44.8762	43.3017	43.2633
Image 6	3	45.9833	41.5720	39.0571	39.0412

Table 3: Comparison of MSSIM for the four interpolation methods when applied to test images.

Image	Scale	Bilinear	GPR	ICBI	EGPR
Image 1	1	0.936	0.937	0.938	0.940
Image 2	1	0.946	0.947	0.953	0.955
Image 3	2	0.905	0.906	0.909	0.910
Image 4	2	0.812	0.808	0.815	0.816
Image 5	3	0.818	0.837	0.850	0.851
Image 6	3	0.857	0.865	0.878	0.879

To further validate our algorithm, we also provide objective measurements. Peak signal-to-noise ratio (PSNR) and root mean square (RMS) error are traditional quantitative measures of accuracy, and by comparing their values for the above images, we can conclude that the proposed EGPR algorithm yields interpolated pixel values that are much closer to their original high-quality values than those obtained with the bilinear algorithm, GPR algorithm, and ICBI algorithm. Tables 1 and 2 summarize the PSNR and RMS values for each algorithm at different magnification factors and for each image. It can be observed that the PSNR values for images obtained using the EGPR algorithm are the highest, and those using the bilinear algorithm are the lowest. Further, RMS values for images obtained using the EGPR algorithm are the lowest, and those using the bilinear algorithm are the highest. Overall, it can be clearly demonstrated that our new method outperforms the other three algorithms.

MSSIM [25] is an image quality assessment index which assesses the image visibility quality from an image formation point of view under the assumption of the correlation between human visual perception and image structural information. We compared the MSSIM obtained using the EGPR algorithm at different scale values with the corresponding values obtained using the bilinear, GPR, and ICBI algorithms, as shown in Table 3. It is noted that our new algorithm achieves a greater MSSIM than the other three algorithms, and the results show that the images obtained using our algorithm are closer to the original high-resolution images in terms of image structure similarity.

In addition, Figure 7 clearly demonstrates the quantitative assessment results for each image at different magnification levels. In this figure, the blue dots represent the quality scores of the images obtained using the comparison algorithms, and the red dots represent those obtained using our algorithm. Our interpolation algorithm is notably superior to the other algorithms, according to all three objective measurements. The proposed algorithm therefore yielded encouraging performance in terms of image visualization and quantitative quality assessment, making it a competitive image interpolation algorithm.

5. Conclusions

In this paper, we have presented a novel EGPR method for image interpolation. The main feature of this new algorithm is its ability to obtain relatively high prediction accuracy of the unknown pixels by fully utilizing underlying image patch information. The implementation process involves two steps: training and prediction. The former creates a GPR model using only single-image data as the training set, and the latter combines energy computation with the acquired model to produce a high-resolution image. Experiments have shown that our algorithm can yield encouraging performance not only in terms of image visualization

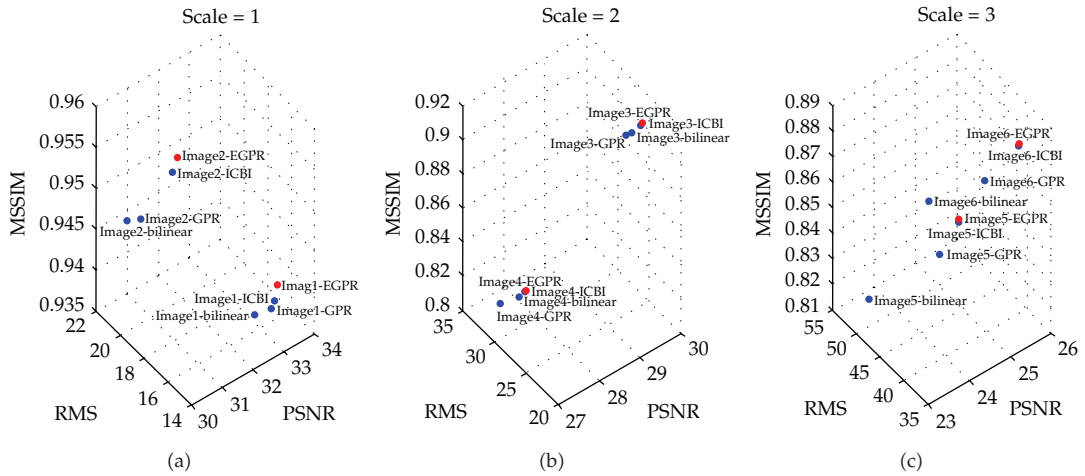


Figure 7: Quantitative quality assessment results for the four interpolation methods.

but also in terms of PSNR, RMS, and MSSIM quality measures. However, better image interpolation comes at the expense of greater algorithm complexity. Methods of improving the algorithm efficiency need further investigation. In future, we can improve this algorithm to address the problem of the interpolation of image sequences. Images in the same sequence are also subject to the recurrence phenomenon, whereby images contain spatial-temporal correlation [26]. We believe that this problem can be addressed using the improved EGPR algorithm by finding an appropriate energy-driven computation and training mode.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) 2012CB821200 (2012CB821206), the National Natural Science Foundation of China (no. 91024001, no. 61070142), and the Beijing Natural Science Foundation (no. 4111002).

References

[1] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 739–758, 2000.

[2] K. S. Ni and T. Q. Nguyen, "An adaptable *k*-nearest neighbors algorithm for MMSE image interpolation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1976–1987, 2009.

[3] Z. Hou and Q. Zhang, "Interpolation algorithm for recovering the missing phase values in 3D measurement," *Optik*, vol. 121, no. 14, pp. 1324–1329, 2010.

[4] J. W. Han, J. H. Kim, S. H. Cheon, J. O. Kim, and S. J. Ko, "A novel image interpolation method using the bilateral filter," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 175–181, 2010.

[5] T. Nagata, "Smooth local interpolation of surfaces using normal vectors," *Journal of Applied Mathematics*, vol. 2010, Article ID 952420, 24 pages, 2010.

[6] Y. W. Tai, S. Liu, M. S. Brown, and S. Lin, "Super resolution using edge prior and single image detail synthesis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2400–2407, June 2010.

[7] W. S. Tam, C. W. Kok, and W. C. Siu, "Modified edge-directed interpolation for images," *Journal of Electronic Imaging*, vol. 19, no. 1, Article ID 013011, pp. 1–20, 2010.

[8] J. Sun, J. Sun, Z. Xu, and H. Y. Shum, "Image super-resolution using gradient profile prior," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.
[9] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, 2011.
[10] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 349–356, October 2009.
[11] K. I. Kim and Y. Kwon, "Example-based learning for single-image super-resolution," in *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pp. 456–465, 2008.
[12] R. Fattal, "Image upsampling via imposed edge statistics," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 1276496, pp. 95-1–95-8, 2007.
[13] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 449–456, 2011.
[14] H. Asheri, H. R. Rabiee, N. Pourdamghani, and M. H. Rohban, "A gaussian process regression framework for spatial error concealment with adaptive kernels," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 4541–4544, August 2010.
[15] G. Salimi-Khorshidi, T. E. Nichols, S. M. Smith, and M. W. Woolrich, "Using gaussian-process regression for meta-analytic neuroimaging inference based on sparse observations," *IEEE Transactions on Medical Imaging*, vol. 30, no. 7, pp. 1401–1416, 2011.
[16] J. Pasi, V. Jarno, and V. Aki, "Robust Gaussian process regression with a student-t likelihood," *Journal of Machine Learning Research*, vol. 12, pp. 3227–3257, 2011.
[17] L. You, V. Brusic, M. Gallagher, and M. Bodén, "Using Gaussian process with test rejection to detect T-cell epitopes in pathogen genomes," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 4, pp. 741–751, 2010.
[18] S. Sun and X. Xu, "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 466–475, 2011.
[19] R. Zimmermann, "Asymptotic behavior of the likelihood function of covariance matrices of spatial Gaussian processes," *Journal of Applied Mathematics*, vol. 2010, Article ID 494070, 17 pages, 2010.
[20] A. Ranganathan, M.-H. Yang, and J. Ho, "Online sparse Gaussian process regression and its applications," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 391–404, 2011.
[21] O. Rudovic and M. Pantic, "Shape-constrained Gaussian process regression for facial-point-based head-pose normalization," in *Proceedings of the International Conference on Computer Vision (ICCV '11)*, pp. 1495–1502, 2011.
[22] A. M. Feng, L. M. Fang, and M. Lin, "Gaussian process regression and its application in near-infrared spectroscopy analysis," *Spectroscopy and Spectral Analysis*, vol. 31, no. 6, pp. 1514–1517, 2011.
[23] M. M. Atia, A. Noureldin, and M. Korenberg, "Gaussian process regression approach for bridging GPS outages in integrated navigation systems," *Electronics Letters*, vol. 47, no. 1, pp. 52–53, 2011.
[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning, MIT Press, Cambridge, Mass, USA, 2006.
[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[26] J.-W. Roh and B.-K. Yi, "Efficient indexing of interval time sequences," *Information Processing Letters*, vol. 109, no. 1, pp. 1–12, 2008.