New Interpolation Algorithm for Image Inpainting

Lin Chang *, Yu Chongxiu

Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Ministry of Education, P.O. Box 72 (BUPT), Beijing 100876, China

Abstract

In this paper, the relationship between image pixels’ coordinators and their colors is considered as a mapping function. A novel algorithm based on radial basis function network is applied to construct a local best approximation of this function, and then the damaged pixels are restored via interpolation. In the algorithm, the mapping function is constructed pixel by pixel to minimize the interpolation error and decrease its complexity. To proceed over larger damaged areas, we compute interpolations of different overlapping coefficients to combine global frequency and local spatial information together. This contributes to restore more detail information. Experimental results show that the algorithm can restore the missing information correctly and can be effectively used in reconstruction of complex signal functions.

Keywords: Image Inpainting; Interpolation; Radial Basis Function

1. Introduction

The object of inpainting is to reconstitute the missing or damaged portions of image, in order to make it more legible and to restore its unity [1]. The significance of inpainting in image processing can be clearly seen from its wide application in: restoration of photographs, films and paintings; text and objects removal in images; special effects in movies; disocclusion in computer vision [2].

Inpainting was initially introduced into image processing by Bertalmio et al [1]. The authors used partial differential equations (PDE) method to restore images. Their algorithm fills in the areas to be inpainted by smoothly propagating information from the surrounding areas along the isophote direction. It does a good job in small damaged regions, but for larger areas, the results usually look blurry. Lately,
Chan and Shen proposed two inpainting models: the Total Variation (TV) [2] and the Curvature-Driven Diffusion (CDD) models [3]. They converted the problem into finding out the extrema of energy functions. However, their models still only aim at handling local non-texture inpainting [2,3]. The difficulty of real inpainting problems is due to the rapid variations of isophote and the roughness of image functions.

Besides, exemplar-based and region-filling methods [4,5] search the optimal matching patches within the valid image regions, and fill them to the damaged areas according to the order of repair priority. Experiments show that these methods can effectively fill the larger damaged image areas, but the search process is very slow and prone to false match.

Inpainting is essentially an interpolation problem, but the complexity of image functions make it highly non-trivial. In this paper, we define a mapping function between image pixels’ coordinator and their colors, and apply an algorithm based on radial basis function to construct a local best approximation of this mapping. To increase accuracy of the solution and decrease complexity of the algorithm, the approximation function is constructed pixel by pixel. For larger damaged areas, we compute interpolations of different overlapping coefficients and superimpose them at a rational ratio. This contributes to restore more detail information. Examples show that our algorithm work well for most real images, even those with rich texture structures.

2. The Inpainting Algorithm

For arbitrary image, let \( X \) represents the pixel’s coordinates (\( X \in \mathbb{R}^2 \)), and \( Y \) represents the pixel’s color (for true-color images, \( Y \) contains R, G and B three components, it is a 3-by-1 vector). Then \( X \) and \( Y \) form a mapping relationship: \( Y = F(X) \). Hence, the image inpainting problem can be expressed as: Based upon the valid image information around the damaged portions, apply an appropriate method to find out the best approximation of implicit mapping \( F : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \). Then for each damaged pixel \( i \), its color can be obtained by computing the function value \( F(x_i) \), \( x_i \) is the coordinates of pixel \( i \).

Finding out the mapping \( F \) is a typical functional approximation problem. Radial basis functions (RBF) network has universal approximation property [6,7], so it is employed for functional approximation here. Commonly, Radial basis function is embedded into a three-layer feed-forward network. In our algorithm model, the input layer consists of two nodes (equal to the dimensionality of input vector \( X \)). Hidden layer has \( N \) neurons, their activation function is given by: \( \phi_j(X) = \varphi(\|X - C_j\|), j = 1, 2, \ldots, N \). Usually, \( \phi_j(X) \) is a Gaussian function, i.e. \( \phi_j(X) = \exp(-\|X - C_j\|^2/2\sigma^2_j) \). \( N \) equals to the number of reference points used to construct the approximation function. \( C_j \) is the radial basis function center of the \( j \)-th unit. \( \sigma_j \) is the expansion constant, it is a measure of the width of the \( j \)-th Gaussian function with center \( C_j \). Output layer contains three neurons. It implements a weighted sum of hidden-unit outputs: \( y_i = \sum_{j=1}^{N} w_{ij} \phi_j(X) \), for \( i = 1, 2, 3 \). Here \( W = \{ w_{ij} \}_{i,j} \in \mathbb{R}^{3\times N} \) is the weight vector.

For functional approximation, the selection of parameters \( N, C_j, \sigma_j \) is critical. If \( N \) is too small, the achieved approximation function will be not accurate. If \( N \) is too large, the training process will be very
slow, and it is difficult to converge on the stable solution. At this time, the network will be easily misled by noise, and its generalization ability is reduced. \( C_j \) is usually computed by K-means clustering algorithm. In simple case, it can be set to the coordinates of corresponding reference point directly. \( \sigma_j \) is determined by the formula: \( \sigma_j = k d_j \), here \( d_j \) is the distance between \( j \)-th function center and its nearest neighbor. \( k \) is called “overlapping factor”, it affects the network's generalization ability directly. When \( N, C_j, \sigma_j \) are settled, we will get the linear equations of weight vector \( W \). Suppose the training set is \( \{X_i, Y_i\}_{i=1}^M \), \( M \) is number of samples, \( M > N \), then we have:

\[
\begin{bmatrix}
\Phi_{11} & \Phi_{12} & \cdots & \Phi_{1N_1} \\
\Phi_{21} & \Phi_{22} & \cdots & \Phi_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
\Phi_{N1} & \Phi_{N2} & \cdots & \Phi_{MN_N}
\end{bmatrix}
\begin{bmatrix}
w_{11} \\
w_{12} \\
w_{13} \\
w_{21} \\
w_{22} \\
w_{23} \\
\vdots \\
w_{N1} \\
w_{N2} \\
w_{N3}
\end{bmatrix}
= 
\begin{bmatrix}
Y_{1T} \\
Y_{2T} \\
\vdots \\
Y_{MT}
\end{bmatrix}
\tag{1}
\]

Here, \( \Phi_{ij} = \varphi(\|X_i - C_j\|); j = 1, 2, \cdots, N; i = 1, 2, \cdots, M \)

In this paper, we use the recursive least-squares (RLS) algorithm to solve equations (1). After training, the network will model the demand input-output mapping accurately, and thus can be employed for interpolation. But for practical application, we still have to consider two problems: first is how to reduce the computational complexity, second is how to proceed over larger areas. Here we propose the following methods to settle these issues:

1. In practical situation, constructing a mapping covering the whole inpainted areas is impossible. Since the spatial complexity of RLS is \( O(M^2) \), this will encounter dimensionality disaster. Meanwhile the solution of equations (1) will be unstable. In fact, the mapping function should be constructed locally pixel by pixel. Here “locally” means that we only select reference points around the inpainted pixel, and it also means that the mapping function is only used to compute the interpolation of current pixel. This can effectively reduce the complexity of the algorithm and improve the solution accuracy.

2. To proceed over larger damaged areas, we use a multiple-interpolation method: Based on the fact that image signals can be decomposed into different frequency components, we select reference points more sparsely in a larger context and use a larger overlapping factor \( k \) to approximate the “low-frequency components”, which represents the background information. Meanwhile we select reference points more intensively in a smaller context and use a smaller overlapping factor to approximate the “high-frequency components”, which contains most of detail information. Finally we superimpose the different components at a rational ratio. Experiments show that this can improve the restoration results effectively.

3. Each damaged area is processed from outside inward. Let \( \Omega \) be an area to be inpainted and let \( \partial \Omega \) be its boundary. For each point in \( \partial \Omega \), we construct its implicit mapping function and compute its interpolation one by one. After the whole \( \partial \Omega \) is processed, we mark \( \partial \Omega \) as repaired, then revise \( \Omega \), and repeat the process until the entire area is restored.

3. Experimental Results

We apply our algorithm to a variety of true-color images to assess its performance. We arbitrarily add texts or scratches on the images, and then use the algorithm to remove them. To evaluate the quality of inpainting, we use the mean-square error (MSE) of the restored images as a measure of the reconstruction.
The results are summarized in Table 1. Notice that for images without intensity discontinuity (such as tarja and tulip) the MSE are very small. For images which texture and structure are more complicated (such as fruit and ocean), the algorithm still achieves the desired results even the scratches width is up to 10 pixels (The total MSE are 706.83 and 497.53, respectively. In this paper, the MSE of R, G and B components are added together).

Table 1. MSE of the restored images

<table>
<thead>
<tr>
<th>Image</th>
<th>Scratch width (pixels)</th>
<th>Number of restored pixels</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>fruit.bmp (24bits, 600×406 BMP)</td>
<td>5</td>
<td>8829</td>
<td>452.01</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>17845</td>
<td>706.83</td>
</tr>
<tr>
<td>ocean.bmp (24bits, 800×600 BMP)</td>
<td>5</td>
<td>14761</td>
<td>281.30</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>24396</td>
<td>497.43</td>
</tr>
<tr>
<td>tarja.bmp (24bits, 682×1024 BMP)</td>
<td>5</td>
<td>21415</td>
<td>94.33</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>32250</td>
<td>324.46</td>
</tr>
<tr>
<td>tulip.bmp (24bits, 480×640 BMP)</td>
<td>5</td>
<td>9353</td>
<td>55.17</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>19286</td>
<td>223.94</td>
</tr>
</tbody>
</table>

In literature [9], inpainting algorithms such as Bertalmio, Oliveira, Inverse Distance Weight and Anisotropic Interpolation were implemented. Their inpainting results were compared and shown in the literature [9] Table 2. From the table we can see that their average MSE for each RGB channels are 912.03, 576.27, 662.09 and 458.46 respectively (Their scratch widths are 7-12 pixels). Statistically, our results compare favourably to those obtained by above methods.

Figure 1 and 2 demonstrate the effectiveness of our algorithm. From the examples we can see that, in the regions where structure changes linearly, the colors can be smoothly reconstructed. While in texture-rich regions, it also can restore some texture details.

![Fig. 1. ocean.bmp (24bits, 800×600 BMP). (a) Image covered by scratches (scratch width=10 pixels); (b) Restored result (total MSE=497.43).](image-url)
4. Conclusions

The task of image inpainting is to restore the missing information from what is available. It is essentially an interpolation problem, but the complexity of image functions make it very difficult to solve. In this paper, we proposed a novel algorithm using radial basis function to construct the local best approximation of image function and restore the damaged pixels via interpolation. RBF network is capable of providing arbitrarily good approximation to prescribed functionals of a finite number of real variables [6]. On this basis, we constructed the approximation function pixel by pixel. This can increase accuracy of the solution and decrease complexity of the algorithm. For larger damaged regions, we use a multiple-interpolation method to restore the detail information as far as possible. Experiments show that our algorithm is very stable; it can restore the missing information correctly and can be used to reconstruct complex signal functions effectively.

References