An Image Inpainting Algorithm Based on CSRBF Interpolation

Weihong Wang , and Xujia Qin

College of Software, Zhejiang University of Technology, Hangzhou, China, 310032

wwh@zjut.edu.cn, qinxj@cad.zju.edu.cn

Abstract

In this paper, we propose a novel algorithm for image inpainting based on compactly supported radial basis functions (CSRBF) interpolation. The algorithm converts 2D image inpainting problem into implict surface reconstruction problem from 3D points set. Firstly, we construct the implicit surface for approximating the points set which convert from damaged image by using radial basis functions (RBF), and then resample from the constructed surface can evaluate the pixels' value of damaged or removed portion on the image. Using compactly supported radial basis functions, the matrix of corresponding system of the linear algebraic equations is spare and bounded. So it can decrease the computational complexity of RBF algorithm. Experiments show that good results are obtained by using the proposed algorithm.
Keyword: Image Inpainting, Compactly-Supported, Radial Basis Functions.

I. Introduction and related works

Image inpainting is a technique which repair damaged image or remove extra parts from image by means of image interpolation. The technique can be used in photo editing, cultural relic restore, et al. In image inpainting, many works were based on techniques of noise removing or texture synthesis [1-3]. DeBonet uses multi-resolution pyramid technique to reconstruct texture. He import two Laplace pyramids and filter to synthesis texture. This method can synthesis complex structure texture[4]. L.Y.Wei proposed a fast multi-resolution texture synthesis way based on vector quantization [5]. Because texture synthesis methods can't meet the general needs of image inpainting, Bertalmio use partial differential equations (PDE) to restore damaged image and obtain good results [6,7]. Chan presented a total variational [8] and curvature-driven diffusion [9] way to implement image inpainting. Bertalmio and Chen's methods are considering curvature continuity on boundary. Oliverira uses Gauss convolution kernel to filter the image [10], and it can repair the damaged region of the image rapidly. But this approach only suit to damaged region is a narrow strip. Recently, Jia proposes an adaptive N-D tensor voting method [11], which can covert the color and texture information into N-D tensor, and then computing an optimum color value of each pixel in the N-D texture space.

Radial basis function (RBF) interpolation is an important method for surface reconstruction from 3D scatter points [12,13]. Using compactly supported radial basis can reduce the complexity [14,15].

In this paper we import Kojekine's surface retouch method [16] based on RBF onto damaged image restoring, and proposed a new image inpainting algorithm based on CSRBF. We convert a 2D image to 3D points cloud. Then reconstruct a implicit surface from the 3D points set. Finally, resample the reconstructed surface can restore the damaged part pixels' value. The rest of the paper is organized as follows. The next section describes RBF and CSRBF algorithm and their complexity. Section 3 discusses the image inpainting method based on CSRBF. Application examples of image impainting are presented in section 4. Section 5 concludes the paper.

II. Surface interpolating based on CSRBF

A. *Surface interpolating based on RBF*

RBF is an important method for implicit surface modeling in computer graphics recently. Surface interpolation based RBF is a main way for surface reconstruction from large-scale 3D point cloud. Give 3D points set $\{P_i\}_{i=1}^n$, RBF interpolated surface is defined by zero isosurface of the following radial basis function:

$$f(p) = q(p) + \sum_{j=0}^{n} \lambda_j \phi(|p - p_j|) \tag{1}$$

Where $q(p)$ is a polynomial of degree at most a specified number, λ_j is a real-valued weight, $\phi(\bullet)$ is a radial basis, and $|\bullet|$ is a Euclid norm.

In order to solve the weights λ_j, we can convert to solve the following linear system:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} & x_1 & y_1 & z_1 & 1 \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} & x_n & y_n & z_n & 1 \\ x_1 & x_2 & \cdots & x_n & 0 & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_n & 0 & 0 & 0 & 0 \\ z_1 & z_2 & \cdots & z_n & 0 & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

where denoting $f(p_j) = h_j$, $\phi_{ij} = \phi(|p_i - p_j|)$, $q(p) = c_0 x + c_1 y + c_2 z + c_3$.

Equation (2) may be rewritten in matrix form as:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} h \\ 0 \end{pmatrix} \tag{3}$$

where $A_{ij} = \phi(|p_i - p_j|)$, $P = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix}$, $i, j = 1, 2, \cdots, n$.

In linear system (2), the coefficient matrix is obviously real symmetric, and with proper selection of basis functions it can be made positive-definite. Thus, a solution always exists to the linear system[15].

The point's value which on the RBF interpolated surface is always zero, that is $f(p_i) = h_i \equiv 0$. In order to avoid the trivial solution that $f(p_i)$ is zero everywhere, off-surface points are appended to the input data and are given non-zero values. Those off-surface points are generated by projecting along surface normal.

B. *Algorithm analysis of RBF interpolating*

Reconstructing surface using RBF from points cloud, the complexity of the algorithm may be considered in the following three parts:

(a) Constructing the linear system;

(b) Solving the system; and

(c) Evaluating the interpolating function (1).

For constructing coefficient matrix of the linear system, the distance of each two position should be calculated. Although the symmetry of the matrix cuts the computational cost in half, the computational complexity is still $O(n^2)$. To solve the linear system (2), Turk and O'Brien use LU factorization (an $O(n\ )$ algorithm), and they correctly point out that it is possible to solve this system in $O(n^2)$ by iterative means [17].

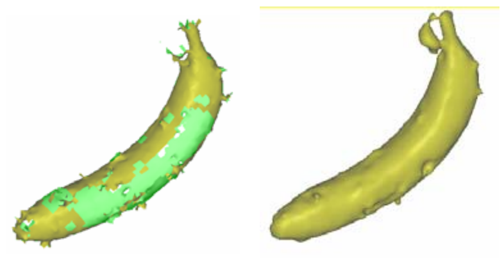
C. *CSRBF interpolating*

In 1995, Wendland constructed compact, locally-supported radial basis functions, which guarantee the coefficient matrix of linear system (2) positive-definiteness [14]. The basis has the unify form:

$$\phi(r) = \begin{cases} (1-r)^q\, p(r) & 0 \le r < 1 \\ 0 & otherwise \end{cases} \tag{4}$$

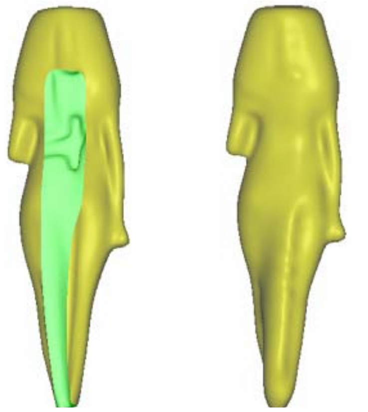
Wendland has derived various degrees of desired continuity (C^k) and dimensionality (d) of the interpolated function. Where the form $d = 3$, C^0 is $\phi(r) = (1-r)_+^2$, and $d = 3$, C^2 is $\phi(r) = (1-r)_+^4(4r+1)$. The equation (4) has radius r equal or small to 1. Scaling of the basis functions (i.e. $\phi(r/\alpha)$) allows any desired radius of support α.

Because compactly supported of RBF, there is $\phi(|P_i - P_j) = 0$ when distance between P_i and P_j is



a b

Fig.1 CSRBF interpolation for banana model. a) original surface, b) CSRBF repaired surface



a b

Fig.2 CSRBF interpolation for nun model. a) original surface, b) CSRBF repaired surface

larger than the supported radius r. Thus property of compactly supported RBF can make the linear system's coefficient matrix spare and bounded. It can reduce the computational complexity of RBF interpolating. Reference [15] has given out the complexity of CSRBF: (1) computational complexity of constructing the linear system is $O(n \log n)$, $O(n)$ storage is required to represent the system; (2)

$O(n^{1.5})$ computation is required to solve the system of equations; (3) $O(\log n)$ computation is required to evaluate the interpolating function.

Figure 1 and Figure 2 are experiments of surface restore for incomplete surface using CSRBF. Figure 1a and 2a are incomplete surfaces, and Figure 1b and 2b are result surfaces restored by CSRBF interpolation.

III. Image inpainting based on CSRBF interpolating

A. *Surface interpolating based on RBF*

For a gray image $M \times N$, the value of a pixel $f(x,y)$ looks as a z value in 3D cardinal space. That is $z = f(x,y)$. Thus the image data can convert to 3D points cloud data. Those points are absent convert from the damaged parts of the image. Those absent points we can be evaluated by CSRBF interpolation. We first reconstruct the surface from the 3D points cloud data by CSRBF. Then obtain the absent points by resampling the reconstructed surface. The absent points' z value is those

a b c d

Fig. 3 Image inpainting based on CSRBF interpolation. a) damaged image, b) surface convert from the gray image a), c) surface restored by CSRBF interpolation, and d) the repaired image.

damaged pixels original gray value. In the algorithm discusses in this paper, the support radius of CSRBF is equal to the width of the damaged region of the image. Figure 3a is a damaged gray image. Figure 3b is incomplete surface convert from the gray image. Figure 3c is a surface restored by CSRBF interpolation. Figure 3d is an image inpainting result.

The process for gray image inpainting is described as following: firstly, label the damage parts (or the objects to remove) in the image, and create a mask image (see Figure 5c). The pixels that labeled is need to interpolating. Then restore the labeled pixels using CSRBF interpolation. The mask can create automatic by proper algorithm, also it can create by manual interactively.

B. *Normal evaluating*

When constructing RBF implicit surfaces from points cloud, in order to avoid the trivial solution of RBF linear system, some points are offset a small distance in direction of surface normal. So we should compute the point's normal. In this paper, we use point's 8 neighbors to evaluate its normal. In Figure 4 P_0 ~ P_7 is P 's 8 neighbors. At first, we compute each vector v_i.

$$v_i = P_i - P \quad i = 0,1,\cdots,7$$

and the normal vector n_i :

$$n_i = v_i \times v_j \quad i = 0,1,\cdots,7 \qquad j = (i+1)\%8$$

The point's normal is evaluated as following equation:

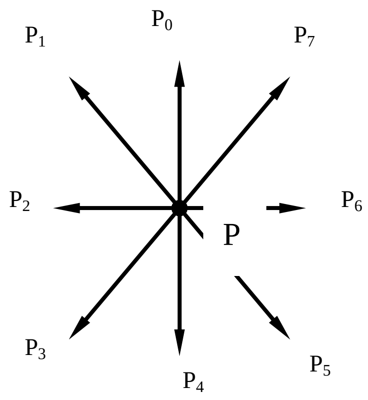


Fig. 4 Point's 8 neighbors for normal evaluate.

$$n = \sum_{i=0}^{N-1} n_i / N$$

Where $N = 8$ if P is not on the boundary, otherwise, $N < 8$.

C. *Image inpainting for color images*

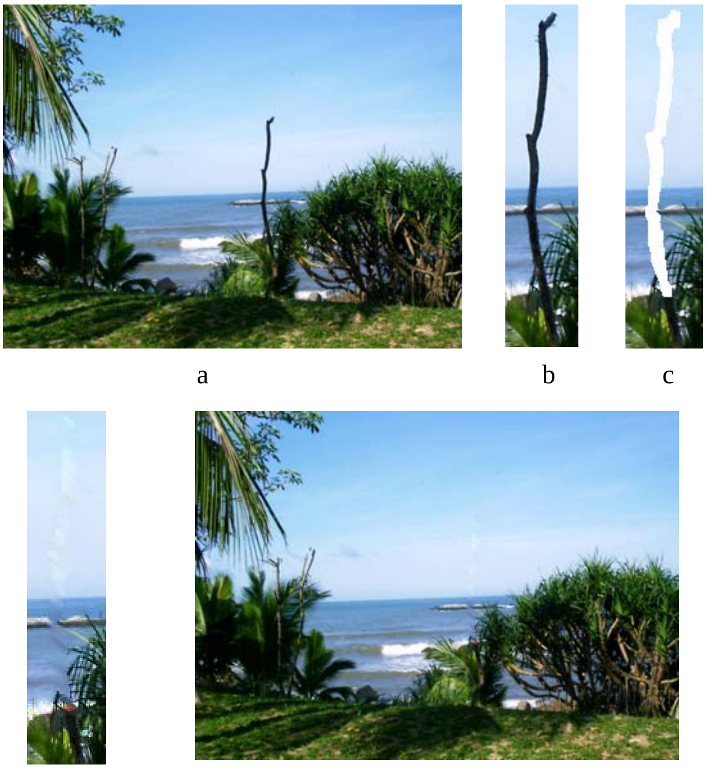
If the image is color image, we separate the color channel to three gray images (R,G, and B), and then inpainting each channel gray image using CSRBF interpolation method which discussed above. Finally, compose them (the three channel image) into the color image.

D. *Defining the region for inpainting*

The principle for image inpainting based on RBF is to build an implicit surface using those points (pixels) near the boundary of the damaged parts. If the damaged part only takes up a small portion of the image, it is not necessary to build the RBF surface using all points (pixels) except the damaged parts. So we can use a strip region around the damaged parts to build the RBF surface (see Figure 6). In simply, we also use a rectangle region as a RBF interpolation region (see Figure 5)

IV. Experiments and results

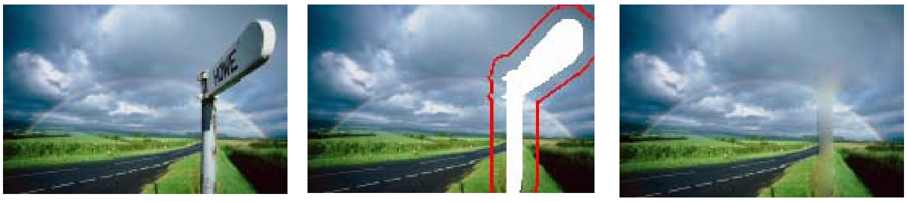
The method in this paper is implement on the platform of VC++6.0. The inpainting region of images is created by manual. Figure 5 is to remove a stump from the photo. Since the stump is occupy a small part of the photo, we use a rectangle region include the stump as a CSRBF interpolating region, see Figure 5b. Figure 6 is to remove road sign from the image. Figure 6b is a mask image; the region is created by manual interactively. Figure 6c is the result removed the road sign from the image.



a b c

d e

Fig.5 Remove a stump from a photo. a) original photo, b) stump part of the photo, c) mask image, d) inpainting local part, and e) the final result



a b c

Fig. 6 Remove the road sign from a photo. a) original image, b) mask image, and c) final result.

a b c

Fig. 7 Remove iron wire netting from a photo. a) original photo, b) mask image, and c) the final result.

a b

Fig. 8 Remove characters from an image. a) an image with characters, and b) the image removed the characters.

Figure 7 is to remove the iron wire netting from a photo, and Figure 8 is to remove characters from an image.

V. Conclusion

A practical algorithm for image inpainting is proposed. The gray image-inpainting problem is converted into a surface reconstructing based on 3D points cloud by using CSRBF interpolation. For color image inpainting, we first separate the color channel (R,G and B) into three gray images. And then repair each channel image. Finally, compose them (the three channel image) into the color image. Through analyzing the RBF surface interpolating algorithm, and consider that the linear system matrix of CSRBF is spare and bounded. We use CSRBF can decrease the computational complexity of RBF interpolating. Experiments show that good results are obtained by using the proposed algorithm.

Acknowledgements

This research was supported by the National Nature Science Foundation of China under Grant No. 60473024, Zhejiang provincial Nature Science Foundation of China under Grant No. Y104341 and Z105391.

References

[1] M. Ashikhmin, "Synthesizing Natural Textures", in Proceedings of 2001 ACM Symposium on Interactive 3D Graphics, Research Triangle Park, North Carolina , 2001,pp.217-226.

[2] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting", in Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR). 2003,pp. 721-728.

[3] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis", Computer Graphics, vol. 29, 1995, pp. 229-233.

[4] J. S. DeBonet, "Multiresolution sampling procedure for analysis and synthesis of texture images", in Proceedings of SIGGRAPH 97, 1997, pp. 361-368.

[5] L.Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization", in Proceedings of SIGGRAPH'2000, New Orleans, 2000, pp. 479-488.

[6] M. Bertalmío, G. Sapiro, V. Caselles and C. Ballester, "Image Inpainting", in Proceedings of SIGGRAPH'2000, New Orleans, 2000, pp. 417-424.

[7] M. Bertalmío, A. Bertozzi, G. Sapiro, "Navier-Stokes, Fluid-Dynamics and Image and Video Inpainting", in Proceedings of IEEE CVPR' 2001, Hawaii, 2001, pp.

[8] T .F. Chan and J. Shen, "Mathematical Models for local non-texture inpainting", SIAM, J.Appl.Math, ,vol. 63, no. 2, 2001, pp.1019-1043.

[9] T. F. Chan and J. Shen, "Non-texture inpainting by curvature driven diffusion (CDD)", J.Visual Comm. Image Rep, vol. 12, no.4, 2001, pp. 436-449.

[10] M. M. Oliveira, B. Bowen, R. McKenna, and Y. S. Chang, "Fast Digital Image Inpainting", in Proceedings of the Visualization, Imaging, and Image Processing IASTED Conference, Marbella, 2001, pp. 261-266.

[11] J. Jia, C. K. Tang, "Image Repairing: Robust Image Synthesis by Adaptive ND Tensor Voting", in Proceedings of CVPR'2003, 2003, pp. 643-650.

[12] J. C. Carr, R. K. Beatson, J. B. Cherrie, et al. "Reconstruction and representation of 3D objects with radial basis functions". in Proceedings of SIGGRAPH' 2001, 2001, pp. 417-424.

[13] J. C. Carr, W. R. Fright and R. K. Beatson, "Surface Interpolation with Radial Basis Functions for Medical Imaging", in Proceedings of IEEE Transactions on Medical Imaging, vol. 16, no.1, 1997,pp. 96-107.

[14] H. Wendland, "Piecewise polynomial, positive defined and compactly supported radial functions of minimal degree", AICM, vol. 4, 1995, pp. 389-396.

[15] B. S. Morse, T. S. Yoo, P. Rheingans, et al. "Interpolating Implicit Surfaces From Scattered Surface Data Using Compactly Supported Radial Basis Functions", in Proceedings of Shape Modeling Conference, Genova, 2001, pp. 89-98.

[16] N. Kojekine, V. Savchenko, “Using CSRBFs for Surface Retouching”, in Proceedings of The 2nd IASTED International Conference Visualization, Imaging and Image Processing VIIP2002, Malaga, 2002, pp. 9-12.
[17] G. Turk, J.F.O’Brien, “Variational implicit surface”, Technical Report GIT-GVU-99-15, Georgia Institute of Technology, 1999.

Weihong Wang, **born in 1969, received his Master degree from Zhejiang University in 1999. He is now an associate professor in College of Software, Zhejiang University of Technology. His Current research interests include GIS and image processing.**

Xujia Qin, **born in 1968, received his Ph.D. degree from Dalian University of Technology in 2001. He was a postdoctoral research fellow in State Key lab. Of CAD&CG, Zhejiang University (2001-03). He is now an associate professor in College of Software, Zhejiang University of Technology. His Current research interests include computer graphics and solid modeling.**