DIGITAL IMAGE INPAINTING USING MONTE CARLO METHOD

Jianping Gu, Silong Peng and Xuelin Wang

National ASIC Design Engineering Center
Institute of Automation, Chinese Academy of Sciences
Beijing 100080-2728, China
jianping.gu@mail.ia.ac.cn

ABSTRACT

Image Inpainting refers to the ill-posed problem of filling in the missing data in digital images by interpolating from the vicinity. It is shown in this paper how inpainting can be performed by means of random simulation of boundary integral, which we call the Monte Carlo method. Our method is computationally less taxing than the classical diffusion methods, and yields a solution that may have strong edges.

1. INTRODUCTION

Image inpainting is a practice of filling in the missing data in digital images by interpolation from the vicinity. It is an ill-posed problem, in which image prior model play a crucial role. The continuity and smoothness of the natural images are the bases for us to build such a prior model, as demonstrated by the Esedoglu and Shen's Mumford-Shah-Euler image model[3] and Chan, Kang and Shen's elastica image model[6].

In this paper we present an inpainting algorithm based on an improved Mumford-Shah image model:

$$E[u,\Gamma] = \frac{\gamma}{2}\int_{\Omega\setminus\Gamma} |\nabla u|^2 dx + \int_\Gamma (\alpha|X'(s)|^2 + \beta|X''(s)|^2)ds, \qquad (1.1)$$

where$X(s) = [x(s), y(s)]$, $s \in [0,1]$. The first term is analogous to an elastic energy and minimizing $E(u,\Gamma)$ force the image function to be harmonic on the smooth region $\Omega \setminus \Gamma$. We use the random simulation of boundary integral to interpolate the smooth region of the inpainting domain. We call it a Monte Carlo method. The second term is the elastic energy of the active contour models[12]. We use it to connect the broken edges reaching the boundary of the inpainting domain.

Our algorithm is a fast image inpainting method. The previous diffusion methods need several minutes to fill in a generally small area. Such a speed is unacceptable. Our approach needs about a second to fill in a general non-texture inpainting domain, and yields a more natural solution. Our algorithm can also preserve sharp edges.

2. MONTE CARLO METHOD

We denote Ω the entire image domain, and D the inpainting domain where the data is missing, which we suppose to have piecewise smooth boundary ∂D. In this section, we suppose that the inpainting domain D is a smooth domain and introduce our Monte Carlo method. In this situation we only need to interpolate it from the vicinity by minimizing

$$E[u] = \int_D |\nabla u|^2 dx, \qquad (2.1)$$

The Euler-Lagrange equation of it is $\Delta u = 0$. Considering the vicinity as the boundary condition, the inpainting problem of a smooth region constitute a dirichlet problem. Define the Green's function for the inpainting domain D [13]:

$$\mathcal{G}(z,z_0) = \frac{1}{2\pi}\log\frac{1}{r} + g(z,z_0), \quad r = |z - z_0| \qquad (2.2)$$

where g is a harmonic function such that $\mathcal{G}|_\Gamma = 0$. Thus $g(z) = (\log r)/2\pi,\ z \in \partial D$.

An interpolation method for numerically computing the function $g(z)$ is as follows: In the polar coordinate system let z_0 be taken as the pole, and let the polar axis be directed parallel to the x-axis, thus $z - z_0 = re^{i\theta}$. Let $P(z)$be the analytic function with $g(z)$ as its real part, and have the series representation

$$P(z) = \sum_{k=0}^{\infty} c_k(z - z_0)^k,\ \ c_k = a_k + ib_k.$$

Then we take $g(z) \approx R\{\sum_{k=0}^{n} c_k(z-z_0)^k\}$, i.e.,

$$g(r,\theta) \approx a_0 + \sum_{k=1}^{n} r^k(a_k\cos k\theta - b_k \sin k\theta).$$

Supported by funds from National Science Foundation of China under Grant No. s. 60272042, 101710007.

Since this polynomial has $(2n+1)$ coefficients, we take $(2n+1)$ arbitrary points $z_1, \cdots, z_{2n+1}$ on ∂D and choose the coefficients a_k, b_k, such that at the points $z_j, j = 1, \cdots, 2n+1$, $g(z_j) = (log r_j)/2\pi$, i.e., the coefficients a_k, b_k are determined from the system of equations:

$$a_0 + \sum_{k=1}^{n} r_1^k (a_k \cos k\theta_1 - b_k \sin k\theta_1) = \frac{1}{2\pi} \log r_1$$

$$a_0 + \sum_{k=1}^{n} r_2^k (a_k \cos k\theta_2 - b_k \sin k\theta_2) = \frac{1}{2\pi} \log r_2$$

$$\cdots \qquad \cdots \qquad \cdots$$

$$a_0 + \sum_{k=1}^{n} r_{2n+1}^k (a_k \cos k\theta_{2n+1} - b_k \sin k\theta_{2n+1})$$

$$= \frac{1}{2\pi} \log r_{2n+1}$$

where $z_j - z_0 = r_j e^{i\theta_j}, j = 1, \cdots, 2n+1$.

After the Green's function for the region D is obtained, the dirichlet problem for the region D can be solved in explicit form by using Green's function,

$$u(z_0) = \int_{\partial D} u \frac{\partial \mathcal{G}}{\partial n} ds, \tag{2.3}$$

where n denote the inward normal to the boundary ∂D. We will use the random simulation of the boundary integral in (2.3)to solve the inpainting problem. It can be proved that $\int_{\partial D} \partial \mathcal{G}/\partial n ds = 1$, so $\partial \mathcal{G}/\partial n$ is a distribution on the boundary ∂D. We randomly take N points $z_1, \cdots, z_N$ on ∂D according to this distribution, and let

$$u(z_0) = \frac{1}{N} \sum_{k=1}^{N} u(z_k). \tag{2.4}$$

It is an approximative solution of the dirichlet problem. An alternative method is: we randomly take N points $z_1, \cdots, z_N$ on the boundary ∂D according the uniform distribution, and let

$$u(z_0) = \frac{L}{N} \sum_{k=1}^{N} u(z_k) \frac{\partial \mathcal{G}}{\partial n}(z_k). \tag{2.5}$$

Where $L = \int_{\partial D} ds$ is the length of ∂D. We use this operator to fill in the missing data in digital images.

A more fast method is to take N points $z_1, \cdots, z_N$ on the boundary ∂D according the uniform distribution, and let

$$u(z_0) = \frac{1}{C_1} \sum_{k=1}^{N} u(z_k) \frac{1}{r_k^p}, \tag{2.6}$$

where $C_1 = \sum_{k=1}^{N} 1/r_k^p, r_k = |z_k - z_0|, p \geq 1$. It is a random weighted average operator.

3. PRACTICAL INPAINTING SCHEME

For a general inpainting domains, we detect edges using 'Canny' operator along the boundary of the inpainting domain. If there are not edges reaching it, the inpainting domain will be considered as a smooth region, it will be reconstructed using the Monte Carlo method proposed above. Figure 1 shows an example in this case. It is produced in 0.32 second. The inpainting result is so natural that we can't discern the inpainting domain in the reconstructed image.

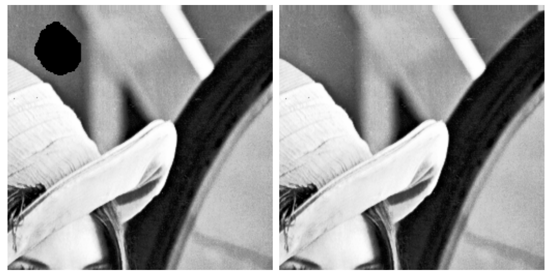

Fig. 1. Left: The image to be filled in; Right: The reconstructed image.

If there are edges reaching the boundary of the inpainting domain, we will first extend these edges into the inpainting domain, they naturally partitioning the domain into smooth regions. Then we complement these smooth region using random simulation of boundary integral. In fact, Nitzberg $et\ al$[14] have firstly proposed the idea of extending the edges and piecewise reconstructing the image. Our work is devoted to design a simulation method to speed up the inpainting process.

We detect edges along the boundary $\partial\Omega$ in the counter-clock direction. The point P_i where there is a edge run to the boundary $\partial\Omega$ is denote as a T-junction. And for each T-junction we compute the intensity of the two side of the broken edge, and denote them as I_i^1 and I_i^2, where the superscript denote respectively the two side in the counter-block order, while the subscript i denote the ist T-junctions in the counter-block direction. We also compute the tangent direction $\overrightarrow{n_i}$of the broken edge. So we can denote a T-junction by $(p_i, \overrightarrow{n_i}, I_i^1, I_i^2)$.

In order to decide that two T-junctions can be connected, we compute

$$\sigma_{i,j} = \sqrt{(I_i^1 - I_j^2)^2 + (I_i^2 - I_j^1)^2}.$$

The two T-junctions $(p_i, \overrightarrow{n_i}, I_i^1, I_i^2)and(p_j, \overrightarrow{n_j}, I_j^1, I_j^2)$which minimize $\sigma_{i,j}$ are considered should be connected. If there is an edge crossing another in the inpainting domain, this method will fail.

For the inpainting problem, the ultimate difficulty lies in producing a natural configuration of edges, which depending on the prior model. We use the internal energy of the snake model[13] to connect the edges, it is to minimize the energy

$$E((p_i, \overrightarrow{n_i}), (p_j, \overrightarrow{n}_j)) = \int_\Gamma (\alpha |X'(s)|^2 + \beta |X''(s)|^2) ds,$$

where Γ denote edge connecting p_i and p_j, and s denote the arc-length. The descent flow of the energy is:

$$X_t(s,t) = \alpha X''(s,t) - \beta X''''(s,t).$$

We first connect two T-junctions using a polygon, and use it as the initial solution of the iterative method. In this paper we let $\alpha = 0.3, \beta = 0.7$.

As discussed by Chan, Kang and Shen in[6], there are three kinds of elementary edge shapes in a local inpainting domain: smooth curve, corner and T-shaped junction. It is easy to continue the smooth edges, but it is difficult to tell apart smooth edges from corners. The existence of a corner or the exact configuration of T-shape inside the inpainting domain will need extra information or require additional models and algorithms. This is still an open problem. Supposing we do know the right class that the inpainting domain belongs to, we only need to connect those corresponding pair of T-junctions and corner points. The connected edges of our paradigm are illustrated in the up picture of Figure 2.

When the edges behind the inpainting domain are all determined, the inpainting domain is divided into smooth pieces. These smooth pieces are independent to each other, so we fill in these smooth pieces one by one using Monte Carlo method. For each smooth piece, we use its borderline to determine the harmonic function g and the Green's function. But in the interpolation process we only consider the tangent section of the boundary of inpainting domain, i.e., we randomly take Npoints on the tangent section of the boundary of inpainting domain, and compute $u(z_0)$ using (2.5). As illustrated in the up picture of figure 3.2, the white line represent the connected edge, it go through the inpainting domain and divide it into two smooth pieces, it also divide the boundary into two section. The left smooth piece only depend on the left section of the boundary, and is independent to the right section. So we compute the Green's function using the left section of the boundary and the connected edge, and interpolate this smooth piece only from the left section of the boundary. It is analogous for the right smooth piece. The result of interpolation is illustrated in the middle picture of Figure 2.

As illustrated in the middle picture of Figure 2, edges formed by our interpolation method are nearly ideal step edges, we will transform them into natural step edges by a convolution with the gaussian function with width σ, which is obtained from the corresponding T-junctions. The down picture of Figure 2 shows the result of edge convolution.

Fig. 2. Up: The inpainting domain is divided into two smooth pieces by the connected edge, and we will interpolate them one by one from their tangent section of the boundary of the inpainting domain; Middle: The result of interpolation. Down: The result of edge convolution.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

We have implemented the algorithm in C++ using a 800MHz Pentium III PC with 256MB of memory running Windows98. Results shown in Figure 3 were produced in about a second using the operator (2.5). Inaccurate edge detecting and connecting caused the artifact on the bright edge of the nose.

Fig. 3. LENA: (Up)Picture with two missing block; (Down)Restored image obtained with our algorithm.

We have presented a digital image inpainting algorithm based on Monte Carlo method. The results produced by this method are more natural than those previously inpainting methods based on diffusion, and is greatly faster.

The presented algorithm is intended for filling in generally large areas, which is difficult for some previously inpainting method. Our algorithm is certainly also fit for locally small areas, but is not fit for long-narrow areas, for example the random trace of a pencil. We will solve this problem in our future work.

5. REFERENCES

[1] M. Bertalmo, G. Sapiro, V. Caselles, and C. Ballester, “Image Inpainting,” computer graphics, SIGGRAPH, pp. 417-424, 2000.

[2] C. Ballester, M. Bertalmło, V. Caselles, G. Sapiro and J. Verdera, “Filling-In by Joint Interpolation of Vector Fields and Gray Levels,” IEEE Trans. on Image Processing, 10 (8), pp. 1200-1211, 2001.

[3] S. Masnou, “Disocclusion: A variational approach using level lines,” IEEE Trans. on Image Processing, 11 (2), pp. 68-76, 2002.

[4] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” $Physica$ D, 60, pp. 259-268. 1992.

[5] T. Chan and J. Shen, “Mathematical Models for Local Non-Texture Inpainting,” SIAM J. Appl. Math, 62(3), pp. 1019-1043, 2001

[6] T. Chan, S. H. Kang and J. Shen, “TEuler’s Elastica and Curvature Based Inpainting,” SIAM J. Appl. Math. 63(2), pp. 564-592, 2002.

[7] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” Comm. Pure Apllied. Math, XLII, pp. 577-685,1989.

[8] P. Perona and J. Malik, “scale-space and edge detection using anisotropic diffusion,” IEEE Trans, on Pattern Anal. math. intell, 12, pp. 629-639, 1990.

[9] S. Esedeglu and J. Shen, “Digital inpainting based on the Mumford-Shah-Euler image model,” European J. Appl. math,13, pp. 353-370, 2002.

[10] M. Oliveira, B. Bowen, R. Mckenna and Y. Chang, “Fast Digital image inpainting,” IVIIP 2001, Marbella, Spain, September 3-5, pp. 261-266, 2001.

[11] G. Kanizsa, “Organization in Vision: Essays in Gestalt Perception,” New Yok: Praeger, 1979.

[12] M. Kass, A. Witkin, and D. Terzopoulos, “nakes: Active contour models,” Int. J. Comput. Vis, Vol. 1, pp. 321-331, 1987.

[13] Prem K. Kythe, “Computational conformal mapping,” Birkhuser, Boston, c1998.

[14] M. Nitzberg, D. Mumford and T. Shiota, “Filtering, segmentation and depth, in Lecture Notes in Computer Science,” Germany: Springer-Verlag, Berlin, vol. 662, 1993.