一种改进的快速图像修复算法

郭 犇,肖创柏,孙翘楚,刘 璐

(北京工业大学 计算机学院,100081 北京,mars_0705@qq.com)

摘 要:目前基于结构的图像修复算法中,基于快速行进的图像修复算法能够简单快速的修复数字图像中的破损区域,但是修复效果一般,特别是边缘区域的保持效果较差.在快速行进算法中引入了梯度权函数和距离权函数,通过对邻近点的加权计算进行排序,然后按权值大小对破损区域进行逐步修复,并利用梯度排序对边缘进行保持.实验结果表明该算法修复效果要优于原快速行进算法,但在速度方面却难以保持.引入梯度和距离权函数可以使修复效果更好.

关键词:图像修复;快速修复;结构模型

中图分类号:TP391.41 **文献标志码**:A **文章编号**:0367-6234(2011)05-0115-04

An improved fast image inpainting method

GUO Ben, XIAO Chuang-bai, SUN Qiao-chu, LIU Lu

(College of Computer Science and Technology, Beijing University of Technology, 100081 Beijing, China, mars_0705@qq.com)

Abstract: In the current structure-based image inpainting algorithm, the Fast Marching Method can simply and rapidly repair the damaged region in the digital image. However, the effect is not very good, especially poor to preserve the edge information. By introducing the gradient function and the distance weight function, an improved algorithm has been proposed, which sorts the neighboring points with weight to repair the damaged region and uses the gradient-sorting estimation to preserve the edge. The results of experiments have indicated that the improved one is better. The gradient function and the distance weight function can make results more better.

Key words: image inpainting; Fast Marching Method; structure model

数字图像修复是图像处理领域的一个重要部分,是利用图像中的有效信息填充指定区域缺损数据的一种技术.该技术已经被广泛的用于各个领域,包括修复医学图像和古典文物、修补有划痕和裂痕的照片、移除图像中的遮挡物等等.

目前,国内外最具代表性的图像修复技术主要有2类:结构图像修复和纹理图像修复[1-2].本文所研究与提出的算法都是基于结构图像的修复算法.对于结构图像的修复,目前大都采用基于偏微分方程(PDE)的方法,其主要思想是将破损区域周围的信息按一定方式延伸到区域内部,达到修补图像的目的.基于上述思想的主要的方法有:Bertalmio[3-4]提出的基于3阶偏微分方程的方法;Chan[5]提出的整体变分(TV)模型;基于曲率驱动扩散(CDD)模型[6-7].这些基于PDE的求解需要大量的迭代运算,因此速度很慢.Telea[8-9]提出了一种新的算法,将破损区域看成水平集并用快速行进法(FMM)来传递图像信息,具有较快的修复速度.该算法首先利用快进算法选择修复路径,并使用方向、距离和水平集3个权值对邻域像素进行加权平均,然后沿着等照度线进行平滑估计.由于其邻域像素的权值和梯度分散,所以使用加权平均后对于边缘保持效果不理想.

本文通过对图像局部特性的分析,对于单个像素点的修复是利用修复像素与周围邻近像素的相关性,通过定义权函数来确定相关性大小的方

收稿日期:2010-04-02.

基金项目:国家高技术研究发展计划基金资助项目(2009AA012437);北京市自然科学基金资助(4072004, 4092006).

作者简介:郭 犇(1986—),男,硕士研究生;
削创柏(1962—),男,教授,博士生导师.

法进行的. 采用快进法选择修复路径符合信息扩散的思想并使得整个修复过程可一次完成,因此修复速度很快. 实验结果表明,该算法对平滑区域和边缘细节均有良好的修复能力.

1 快速行进算法原理

1.1 数学模型

如图1所示,记Ω为图像中的待修复区域,$\partial\Omega$为该区域的边界. 假定p为$\partial\Omega$上的一点. 围绕p取一个小邻域$S\varepsilon(p)$,p点的修复应该由其邻域像素$S\varepsilon(p)$来决定. 也就是说p点与局部信息$S\varepsilon(p)$有关. 利用这一基本原理,可将$S\varepsilon(p)$中的点进行加权平均来估算p点的灰度值$I(p)$.

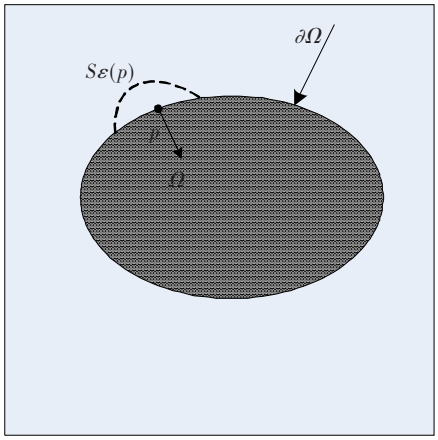

图1 修复原理

1.2 快速行进算法

快速行进法基本思想是在图像外围构造一个活动边界,边界内部的到达时间未知,当前修复边界利用逆向差分格式向内修复,凡是修复过的点,就重置其到达时间,然后构造新的活动边界. 如此循环,就可以得到整个平面上每个点的到达时间.

简单的说,快速行进算法就是对待修复区域内部的点,解 Eikonal 方程为

$$|\nabla T| = 1. \tag{1}$$

式中:T为待修复区域Ω中的点到边缘$\partial\Omega$的距离.

在应用快速行进法之前,由图 1 可见图像的像素可以分为 3 种类型. 并对 T 进行初始化.

1) Boundary. 正被处理的像素,即边界 $\partial\Omega$ 上的点,其 T 值将要被更新.

2) Known. $\partial\Omega$ 外部的像素,属于已知图像区域的像素,其 T 值和灰度值 I 已知.

3) Inside: $\partial\Omega$ 内部的像素,属于未知图像区域的像素,其 T 值和灰度值 I 未知.

令 $D_{i,j}^{\pm x}$,$D_{i,j}^{\pm y}$分别为 x 方向和 y 方向的差分. 利用逆向差分法,方程的稳定解为

$$[\max(D_{i,j}^{-x}T,0)^2 + \min(D_{i,j}^{+x}T,0)^2 + \max(D_{i,j}^{-y}T,0)^2 + \min(D_{i,j}^{+y}T,0)^2]^{1/2} = 1. \tag{2}$$

其中,$D_{i,j}^{-x}T = T_{i,j} - T_{i-1,j}$,$D_{i,j}^{+x}T = T_{i+1,j} - T_{i,j}$;

$D_{i,j}^{-y}T = T_{i,j} - T_{i,j-1}$,$D_{i,j}^{-y}T = T_{i,j+1} - T_{i,j}$.

通过式(1) ~ 式(2) 求出 Ω 内部所有点到 $\partial\Omega$ 的距离 T,然后按照 T 由小到大的顺序选择路径进行修复工作. 该算法总是最先修复离已知图像最近的像素点,直到待修复区域全部被处理.

1.3 权重函数的计算

在实验中发现,由于 FMM 算法其邻域像素权值和梯度分散,所以边缘保持效果不理想. 针对这种情况,提出了改进方案. 本文引入距离权函数和梯度权函数相结合的方法来保持像素点的等照度线方向. 修复过程中可以求出单个像素点然后再不断迭代求出所有像素点的过程.

用 α 表示要修复的像素,α_1、α_2、α_3、$\alpha_4$4 个点为其周围的已知像素,Δx_1、Δy_1 分别为 α 与 α_1 间的垂直和水平距离. 于是:

$$p(\alpha) = \frac{\sum_{i=1}^{4} W(\alpha_i)S(\alpha_i)p(\alpha_i)}{\sum_{i=1}^{4} W(\alpha_i)S(\alpha_i)}, \quad i = 1,2,3,4. \tag{3}$$

式中:$p(\cdot)$ 为像素的灰度值;$S(\alpha_i)$ 为距离权函数,其表达式为

$$S(\alpha_i) = (1 - \Delta x_i)(1 - \Delta y_i),\ i = 1,2,3,4. \tag{4}$$

式中: $W(\alpha_i)$ 为梯度权函数,其表达式为

$$W(\alpha_i) = (-\mu G(\alpha_i) + 1)^n. \tag{5}$$

式中:n 为一个正常数;μ 为正常数($\mu < 1$),$G(\alpha_i)$ 为

$$G(\alpha_i) = \frac{|f'_x(\alpha_i)| + |f'_y(\alpha_i)|}{2\sqrt{(f'_x(\alpha_i))^2 + (f'_y(\alpha_i))^2}}. \tag{6}$$

式中:f_x',f_y' 分别为图像在 α_i 处垂直和水平方向上的导数.

一般来讲,待修复点像素的灰度值和附近像素点的距离有关系,距离越近,影响越大. 因此引入距离权函数的计算是十分必要的. 但考虑到图像的细节信息,为了使图像的破损边缘修复更平滑,取得更好的修复效果. 在这里引入梯度权函数的扩散率函数作为另一个权值来保持边缘修复效果.

$$W(\alpha_i) = \frac{1}{1 + E(\alpha_i)/\lambda^2}, i = 1,2,3,4. \tag{7}$$

式中:λ 为待定参数,λ 值越小,越趋向于对边缘的增强;λ 值得增大可加强对图像的平滑作用. $E(\alpha_i)$ 为图像在 α_i 处的梯度模的平方,即:

$$E(\alpha_i) = f_x^2(\alpha_i) + f_y^2(\alpha_i). \quad (8)$$

由梯度权函数式(7)可知，W 与 E 成反比关系，E 越小，W 值越接近 1，而随着 E 的增大，W 逐渐趋向于 0，这一特点正符合对图像修复的要求.

将梯度权函数式(7)与距离权函数式(4)结合即可计算出待修复像素的灰度值为

$$p(\alpha) = \frac{\sum_{i=1}^{4}(W(\alpha_i) + S(\alpha_i))p(\alpha_i)}{\sum_{i=1}^{4}(W(\alpha_i) + S(\alpha_i))}. \quad (9)$$

1.4　图像修复步骤

图像修复基本步骤为.

① 初始化.

② 设定待修复区域边界 $\partial\Omega$ 上和边界外的 T 值为 0，并标记 $\partial\Omega$ 上的点为 Known.

③ 遍历 $\partial\Omega$ 上所有 Known 点的四邻点，将原来不属于 Known 的点标记为 Known，并将它们的 T 值记为 1，然后按 T 从小到大进行遍历.

④ 其他剩余的点标记为 Inside，其 T 值为 ∞，实际中一般取 $T = 10^6$.

⑤ 返回②，直到所有的像素点标记为 Known 为止.

2　实验结果与分析

本文算法以 Matlab 6.5 为平台，在 PC 机(Core2，2.13 GHz，2 GB 内存)上实现. 实验中对标记出的划痕进行修复，分别采用文献[9]中的原算法与修改过的算法进行对比.

在客观修复效果方面采用均方误差(MSE)来衡量这 2 种方法的修复效果. 公式为

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - I_i)^2. \quad (10)$$

式中：x_i 为破损图像或修复图像的像素点；I_i 为原图像像素点；N 为破损像素数.

在实验过程中分别对 2 组不同的图像(lena 图像，couple 图像)进行修复. 在修复效果方面，从对比图(图 2、3 所示)的局部放大图可以看出文献[9]中的原算法的修复结果产生了较明显的模糊，而本文所采用的修改后的算法在修复效果上更自然，边缘细节也更为平滑.

(a) lena 破损及局部放大图　　(b) 原算法修复及局部放大图　　(c) 本文算法修复及局部放大图

图 2　Lena 图的修复结果及比较

(a) Couple 破损及局部放大图　　(b) 原算法修复及局部放大图　　(c) 本文算法修复及局部放大图

图 3　Couple 图的修复结果及比较

为了使结果更具有说服力,本文采用常用的均方误差(*MSE*) 来对修复质量进行客观评测. 实验结果如表 1、2 所示. 其中:*MSE*1 为原图像与破损图像的均方误差,2 组表中 *MSE*1 值相同表明在修复过程中采用的是同一破损图,*MSE* 表示修复结果与原图像的均方误差,从 2 组表中的数据可以看出本文修改过的算法其 *MSE* 值要比原算法小很多. 这也从客观上证明本文采用的算法修复结果要优于原快速行进算法.

表 1 Lena 图的均方误差分析

算法	缺损像素数	*MSE*	*MSE*1
原算法	1 353	313.659 3	5 424.9
本文算法	1 353	193.726 5	5 424.9

表 2 Couple 图的均方误差分析

算法	缺损像素数	*MSE*	*MSE*1
原算法	1 724	158.218 1	21 005
本文算法	1 724	91.425 6	21 005

3 结 论

1)基于快速行进的图像修复算法在修复速度方面优势比较明显,但其修复效果尤其是破损边缘效果不理想. 针对这一问题本文在修复过程中引入距离权函数与梯度权函数相结合的计算方法来进行改进.

2)从主观方面及客观实验数据也可以看出修复结果得到了比较好的提升. 同时该算法继承与原快速行进算法,因此在速度上也得到了很好的保持.

参考文献:

[1] CRIMINISI A, PEREZ P, TOYAMA K. Region filling and object removal by exemplar-based image inpainting [J]. IEEE Trans Image Processing, 2004, 13(9): 1200 – 1212.

[2] DRORI I, COHEN-OR D, YESHURUN H. Fragment-based image completion [C]//proceedings of SIGGRAPH 2003 Acm Transactions on Graphics. New York, NY: ACM, 2003: 303 – 312.

[3] BERTALMIO M, SAPIRO G, CASELLES V, *et al.* Image inpainting [C]//proceedings of SIGGRAPH 2000 Acm Transactions on Graphics. New York, NY: ACM, 2000: 417 – 424.

[4] BERTALMIO M, BERTOZZI A L, SAPIRO G. Navier-Stokes, fluid dynamics, and image and video inpainting [C]//Proceedings of the 2001 IEEE Computer Society Conference on CVPR. Washington, NJ: IEEE, 2001: 355 – 362.

[5] CHAN T F, SHEN J H. Non-texture inpainting by curvature driven diffusions[J]. Journal of Visual Communication and Image Representation, 2001, 12(4): 436 – 449.

[6] CHAN T F, SHEN J H. Mathematical models for local non-texture inpainting [J]. SIAM Journal on Applied Mathematics, 2001, 62(3): 1019 – 1043.

[7] CHAN T F, KANG S H, SHEN J H. Euler's elastic and curvature based inpainting [J]. SIAM Journal on Applied Mathematics, 2002, 63(2): 564 – 592.

[8] SETHIAN J A. A fast marching level set method for monotonically advancing fronts [J]. National Academy of Sciences, 1996, 93(4): 1591 – 1595.

[9] TELEA A. An image technique based on the fast matching method [J]. Journals of Graphics Tools, 2004, 9(1):23 – 34.

[10] OLIVEIRA M M, BOWEN B, MCKENNA R, *et al.* Fast digital image inpainting [C]//Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP2001). [S. l.]: ACTA Press, 2001: 261 – 266.

[11] QU Lei, WEI Sui, LIANG Dong, *et al.* A fast image inpainting algorithm by adaptive mask [J]. Journal of Image and Graphics, 2008, 13(1): 24 – 28.

(编辑 张 红)