

# 一种基于图像平均灰度值的快速图像修复算法

彭坤杨 董兰芳

(中国科技大学计算机科学与技术系, 合肥 230027)

**摘要** 基于纹理合成的图像修复技术用于修复大面积破损区域, 目前此类算法都存在时间复杂度高的缺点。针对纹理算法的匹配技术进行改进, 提出了一种基于图像平均灰度值的快速图像匹配算法。该算法在匹配之前预先计算纹理块的平均灰度值以及分割后的纹理块的局部平均灰度值, 以取代计算复杂的匹配项 SSD (sum of squared differences); 匹配过程只需对平均灰度值进行快速比较, 结合阈值控制筛选掉大部分候选纹理块。实验结果表明, 该算法在不损害图像修复质量的同时, 将纹理修复的效率提高到实时水平。同时在纹理合成和纹理修复中具有普遍的适用性。

**关键词** 图像修复 纹理合成 平均灰度值

中图法分类号: TP391.41 文献标志码: A 文章编号: 1006-8961(2009)01-0050-06

## A Fast Image Inpainting Algorithm Based on Average Gray Value

PENG Kun-yang DONG Lan-fang

(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)

**Abstract** Image inpainting techniques based on texture synthesis can be used to remove large objects from digital images. However, these algorithms have high time complexity. To improve the matching technology in inpainting, this paper presents an efficient algorithm that is based on average gray value. Average gray values of texture blocks are calculated to avoid complex calculation of SSD (sum of squared differences). Then in the matching process, most blocks are rapidly eliminated by comparing their average gray values with threshold control. The result shows that the algorithm obtains both real-time performance and high quality result. The proposed algorithm has general applicability in texture synthesis and image inpainting.

**Keywords** image inpainting texture synthesis average gray value

## 0 引言

图像修复技术<sup>[1]</sup>利用图像中的背景区域信息, 对图像中破损的区域进行填充和修正, 以期达到视觉上的修复效果。针对大面积破损区域的修复, 目前主要应用的是纹理合成算法, 即根据图像的纹理性质, 将具有相似纹理特征的图像复制到破损区域。

基于纹理合成<sup>[3-7]</sup>的图像修复算法需要解决两个重要问题: 一是纹理合成顺序, 纹理修复算法虽然

可以较好地保存图像的纹理特征, 但是单纯的纹理合成会导致修复区域的边缘特征模糊, 影响修复的视觉质量, 需要合理安排纹理块的填充顺序; 一是搜索最优匹配块, 通常采用的匹配项为纹理合成中常用的 SSD (sum of squared differences) 匹配。目前纹理修复代表性的成果为 Criminisi 等人提出的算法<sup>[2]</sup>, 该算法通过一个自定义的优先级函数, 结合受损区域周围图像块像素灰度值变化的整体趋势, 较好地解决了纹理合成修复顺序问题, 从而使纹理修复的视觉质量达到了满意的效果。

收稿日期: 2008-08-19 改回日期: 2008-10-28

第一作者简介: 彭坤杨 (1986—), 男, 中国科学技术大学计算机系硕士研究生。主要研究方向为图像处理。E-mail pengkuny@mail.ustc.edu.cn

Crim inisi的优先级算法简单有效,在此优先级算法上衍生了大量的改进算法<sup>[8]</sup>,使得纹理修复算法具有更广的适用性和更好的视觉质量。由于这些算法在搜索最优匹配块时采用的多是 SSD 匹配,导致合成过程非常耗时,不能达到实时的效果。本文通过对 Crim inisi算法的分析改进和实验比较,提出一种基于图像特征值的简单快速的实时纹理修复算法。

## 1 Crim inisi算法

### 1.1 填充顺序

Crim inisi算法提取修复边界  $\delta\Omega$ ,依次计算以边缘上各点为中心的块的优先级,取优先级数值最大的作为待匹配块。而优先级的计算考虑了修补区周围的图像边缘特征。

$I$ 表示原图,如图 1所示,  $\Phi$ 为背景区域( $\Phi=I-\Omega$ ),  $p$ 为修复边界  $\delta\Omega$ 上的点,  $\Psi_p$ 区是以点  $p$ 为中心的块。

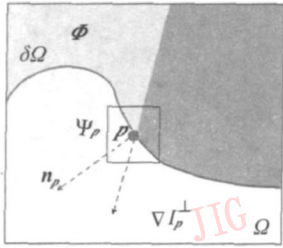


图 1 优先级的计算

Fig 1 Computing patch priorities

点  $p$  的优先级函数  $P(p)$  定义如下:

$$P(p) = C(p) \cdot D(p) \quad (1)$$

式中,  $C(p)$  和  $D(p)$  定义如下:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I-\Omega)} C(p)}{|\Psi_p|} \quad (2)$$

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \quad (3)$$

式中,  $\alpha$  是归一化系数,  $n_p$  是点  $p$  与边缘  $\delta\Omega$  正交的单位向量,  $\nabla I_p^\perp$  是边缘梯度向量。  $C(p)$  反映了  $\Psi_p$  中属于图像部分的像素比重,  $D(p)$  反映了修复区域周围的结构特征。通过这两个因子的互相制约,最终形成一个良好的填充顺序,从而使得修复效果要优于一般的纹理修复算法。

### 1.2 搜索最优匹配块

基于纹理合成技术,传统的块匹配方式为计算

块之间相似度的最近似匹配,即通过遍历样图计算以每个像素为中心的块与待匹配块之间的相似度,取相似度最高的作为匹配块,目前还没有很好的相似度量函数,大部分算法采用的是颜色平方差的和 SSD。

对于待修复块  $\Psi_p$ ,对背景区域中的每一个候选块  $\Psi_q$  进行匹配计算:

$$d(\Psi_p, \Psi_q) = \sqrt{\sum_{i=1}^m \sum_{j=1}^m (p_{ij}^R - q_{ij}^R)^2 + (p_{ij}^G - q_{ij}^G)^2 + (p_{ij}^B - q_{ij}^B)^2} \quad (4)$$

式中,  $p_{ij}^R, q_{ij}^R, p_{ij}^G, q_{ij}^G, p_{ij}^B, q_{ij}^B$  分别为纹理块  $\Psi_p$  和  $\Psi_q$  内对应像素点的红、绿、蓝的颜色值。

最后,得到最优匹配块:

$$\Psi_{q^*} = \arg \min_{\Psi_q \in \Phi} d(\Psi_p, \Psi_q) \quad (5)$$

SSD 虽然可以有效地度量两个纹理块之间的相似度,但是计算量大,无法预处理,每次搜索最优匹配块都需要遍历样图对每一个块进行比较,导致纹理修复算法普遍存在时间复杂度高的缺点。

## 2 本文算法

本文在 SSD 匹配之外,增加两个匹配项——平均灰度值 AGV (average gray value) 和局部平均灰度 LAGV (local average gray value)。在进行精确的 SSD 匹配之前先用平均灰度值和局部平均灰度值筛选淘汰大部分灰度差异较大的匹配块,然后将精确的 SSD 匹配局限在较少的候选块之间,从而提升算法效率。

选取平均灰度值作为匹配项的优点:

1) 可以预计算,而且计算方便。由于避免了 SSD 计算,单次搜索只需要遍历样图,将耗时下降到接近线性水平。

2) 可控性好。通过控制平均灰度值的匹配阈值,就可以有效地控制纹理修复效果,这样就可以在不明显损害修复效果的同时,提升速度。

3) 更新简单。只需要在刚修复过的块附近更新完好区域的平均灰度值,其他像素块的预计算的平均灰度值仍然保持原值,无需重新计算。

如图 2 所示,考虑到待修复块只包含了部分的已知像素(图中白色背景部分),为使匹配更加精确,引入二级匹配项——局部平均灰度值 (LAGV)。

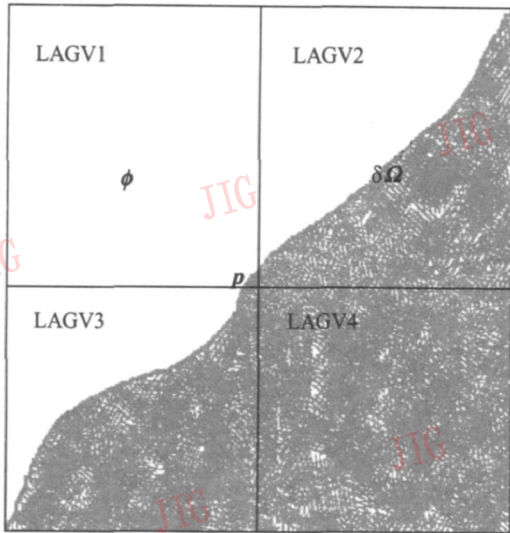


图 2 计算局部平均灰度值

Fig 2 Computing local average gray values

设平均灰度值阈值为  $T_{agv}$ , 局部平均灰度值阈值为  $T_{lagv}$ , 则改进算法如下:

**算法步骤:**

- 1) 提取用户选定的待修复区域  $\Omega$  的边界  $\delta\Omega$ ;
  - (1) 预计算  $AGV q, LAGV q, \forall q \in \Phi$ .
  - (2) 将所有候选块  $q$  按照  $AGV q$  值从小到大排列成有序链表  $L$ .
- 2) 重复下述步骤直到结束: ( $t$  为迭代变量)
  - (1) if  $\delta\Omega^t = \emptyset$ , exit
  - (2) 根据式 (1) 计算优先级  $P(p), \forall p \in \delta\Omega^t$ ;
  - (3) 找到优先级最大的待修复块  $\Psi p$ ;
  - (4) 在背景区域中搜索最优匹配块  $\Psi \hat{q}$  遍历表  $L$ :
    - ① 使用平均灰度值阈值  $T_{agv}$  筛选候选块;
    - ② 使用局部平均灰度值阈  $T_{lagv}$  筛选候选块;

- ③ 根据式 (4) 进行精确 SSD 匹配;
- ④ 根据式 (5) 更新最优匹配块  $\Psi \hat{q}$ .
- ⑤ 将  $\Psi \hat{q}$  中对应的像素复制到  $\Psi p$  中,  $\forall p \in \Psi p \cap \Omega$ .

3) 更新  $C(p), \forall p \in \Psi p \cap \Omega$ ;

更新  $AGV p, LAGV p, \forall q \in \Psi p$ ;

更新有序表  $L$ .

与本算法相结合的优化手段:

1) 对预计算的可匹配块按平均灰度值排序, 搜索最优匹配块过程将在一个有序线性表上进行, 通过二分查找定位合适的起始匹配位置, 这样 SSD 匹配将完全在符合灰度值阈值范围内的候选块中进行, 从而将 SSD 匹配次数减少到最低水平。

2) 让用户划定背景区域, 避免不必要的全图搜索, 减少总搜索次数。纹理合成具有一个重要特点: 背景区域只要包含足够的纹理信息即可合成目标纹理。因此, 只要选取足够信息的背景区域用于修复即可, 这样不但可以减少总搜索次数, 使得机器自动修复过程加速进行, 还可以避免无用背景信息的干扰, 提高修复质量。对于大尺寸图像, 这一点尤为重要。

### 3 实验结果

本文选取了多幅具有代表性的图像进行修复, 通过实验对比本文算法与 Criminisi 算法的合成速度, 本文在参数设置 (纹理块大小  $T_{agv}, T_{lagv}$ ) 上基于以下前提: 保证合成效果与 Criminisi 算法效果持平, 即在不损害合成效果的同时尽可能提高算法效率。为保证可比性, 背景 (样本) 区域均设定为全图。实验环境为 PC 机, AMD Athlon 64 × 21.7GHz, 1.25G RAM。

表 1 实验结果

Tab 1 The results

实验图片	原图大小	本文算法参数 纹理块大小 $T_{agv} T_{lagv}$	算法耗时 /s		提高效率 %
			原算法	本文	
图 3	350 × 262	9 × 9 / 0.9 / 1.5	166.843	1.330	99.2
图 4	206 × 308	9 × 9 / 5 / 6.5	69.731	3.509	95.0
图 5	334 × 251	9 × 9 / 26 / 30	328.894	10.964	96.7
图 6	223 × 186	7 × 7 / 1 / 5	45.474	1.164	97.4
图 7	212 × 285	9 × 9 / 1 / 3	18.536	0.460	97.5



(a) 原图像



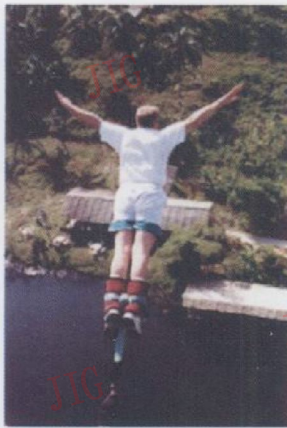
(b) 原算法修复结果



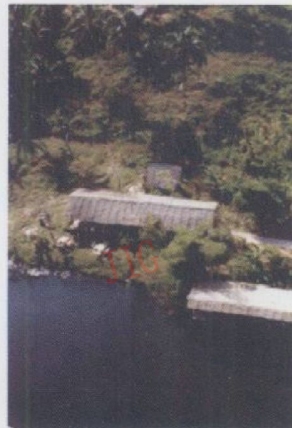
(c) 本文算法修复结果

图 3 实验结果 1

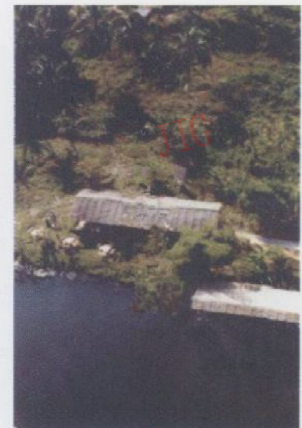
Fig. 3 Result 1



(a) 原图像



(b) 原算法修复结果



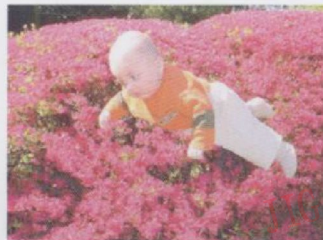
(c) 本文算法修复结果

图 4 实验结果 2

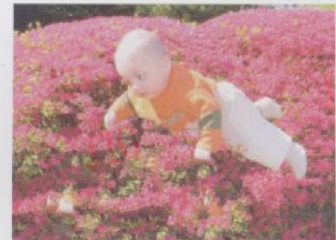
Fig. 4 Result 2



(a) 原图像



(b) 原算法修复结果



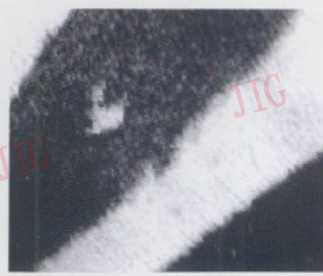
(c) 本文算法修复结果

图 5 实验结果 3

Fig. 5 Result 3



(a) 原图像



(b) 原算法修复结果



(c) 本文算法修复结果

图 6 实验结果 4

Fig. 6 Result 4



图 7 实验结果 5

Fig 7 Result 5

### 3.1 图像修补质量评价标准的选取

图像的客观评价标准是用恢复图像偏离原始图像的误差来衡量恢复图像的质量。最常用的有均方误差 (MSE) 和峰值信噪比 (PSNR)。但用它们所求得的结果常与人们主观视觉效果不一致。

客观评价不适合衡量修补后的图像质量,一方面均方误差和峰值信噪比是从总体上反映原始图像和修复图像的差别,并不能侧重反映局部的待修复区域;另一方面由于图像修复的特殊性,修复过程事先不知道原背景图像的全部信息,而且修复目标不是使得修复区域还原为原始图像,而只需保证修复后的图像部分与背景满足一定的视觉平滑性即可。从图 7 可知,原图像 (a) 和修复结果图像 (c) 之间无法应用目前的客观图像评价方法,国内外的相关文献也多不用客观评价方法来评价图像的修补质量,故本文中采用简单的主观评价方法。

### 3.2 实验结果总结

#### 1) 合成质量

为保证合成质量相比原算法不存在明显下降,实验中对匹配项阈值和纹理块大小等相关参数进行了适当的设置。实验结果表明,在多种典型的纹理图像中,本文算法合成效果都与原算法持平。

实际上,由本文改进算法的思想可知,只要将灰度阈值设置得足够大(对灰度图像阈值最大为 255),本文算法和原算法一致。

#### 2) 合成效率

参与实验的上述图像,其时间效率改进分别高达 99.2%, 95.0%, 96.7%, 97.4%, 97.5%, 原来算法需要数十秒乃至数分完成的修复过程,应用本文改进算法只需要数秒即可完成。对多幅图像修复的实验数据表明,本文算法大幅改进了原算法的时间效率,并且达到了实时效果。

进一步分析可知,本文算法对于大尺寸图像的

修复效果非常明显,如果结合用户选定背景区域,本文算法将不依赖于图像尺寸,应用更加广泛。

从图 5 的修复可以看到,即使阈值设置得非常大 ( $T_{\text{agv}} = 26$ ,  $T_{\text{lagv}} = 30$ ), 仍然可以获得大幅度的速度提升。

## 4 结论

本文在纹理算法的匹配技术上改进,引入可预计算的匹配项,大大减少了不必要的 SSD 匹配计算,从而改善了纹理算法的执行效率,在不明显损害图像修复质量的同时,大幅提升了纹理合成和纹理修复的效率,取得了良好的效果。实验结果表明本文改进算法已经达到实时效果。

本文算法具有广泛的适用性,当前存在的各种纹理合成、基于纹理合成的图像修复等算法都可以应用本文思想获得效率的改进。同时,本文算法思想实现简单,可望进一步改进,以期得到更广泛的应用。

本文进一步工作如下:

1) Criminis 算法不够稳定,对初始边界很敏感,边界的微小变化可导致合成结果的很大差异。需要用户多次修复参数,积累经验,才能获得较好的修复效果。将来的工作应当能结合所修复图像的纹理特征如纹理块的灰度直方图等,自动调整最优参数。

以灰度直方图特征为例,将图像的灰度直方图均衡化后,然后动态调整阈值,使得每次修复一个纹理块时,搜索到最优匹配块时的匹配次数总是接近于总匹配次数,这样就可以有效地兼顾修复质量和算法效率。

2) 在目前仍没有比 SSD 匹配更精确更易计算的相似度匹配函数情形下,可以考虑引入更多的图像特征值作为匹配项,从而更多地筛选掉无效匹配,进一步提高算法效率。

## 参考文献 (References)

- [1] Bertalmio M, Sapiro G. Image inpainting [C] // Proceedings of SIGGRAPH. New York, NY, USA: ACM Press, 2000: 417-424.
- [2] Criminisi A, Pérez P, Toyama K. Region filling and Object removal by exemplar-based image inpainting [J]. IEEE Transactions on Image Processing, 2004, 13(9): 1200-1212.
- [3] Efros A, Freeman A. Image quilting for texture synthesis and transfer [C] // Proceedings of SIGGRAPH. New York, NY, USA: ACM Press, 2001: 341.
- [4] Cheng Wen-huang, Hsieh Chun-wei, Lin Sheng-kai et al. Robust algorithm for exemplar-based image inpainting [C] // Proceedings of International Conference on Computer Graphics, Imaging and Vision [C], Beijing, China [online], 2005: 64-69.
- [5] Efros A, Leung T. Texture synthesis by non-parametric sampling [C] // Proceedings of the International Conference on Computer Vision, Kerkyra, Greece, IEEE Press, 1999: 2: 1033-1038.
- [6] Wei Liyi, Marc Levoy. Fast texture synthesis using tree-structured vector quantization [C] // Proceedings of Siggraph. New York, NY, USA: ACM Press, 2000: 479-488.
- [7] Jeremy L, David M. Improved image quilting [C] // Proceedings of Graphics Interface. New York, NY, USA: ACM Press, 2007: 257-264.
- [8] Liang Lin, Liu Ce. RealTime Texture Synthesis by Patch-Based Sampling [J]. ACM Transactions on Graphics, 2001, 20(3): 127-150.