

## 基于区域纹理合成的图像修补算法

付绍春 楼顺天

(西安电子科技大学电子工程学院 西安 710071)

**摘要:** 针对传统修补技术在图像恢复中出现的纹理模糊现象, 该文根据纹理合成技术, 提出了一种区域纹理合成的修补算法。算法根据破损点邻域特征选择模板, 去除模板内的破损点后进行匹配, 在搜索范围内寻找最佳匹配点。该文采用由边缘向中心逐步推进的修补次序, 提高破损区域内纹理结构延伸方向的正确性以及连续性, 避免出现纹理模糊现象。算法能够适用于多种性质的纹理, 在处理真实图像时使人在视觉上获得满意的效果。

**关键词:** 图像修补; 纹理合成; 模板选择; 匹配

中图分类号: TN911.73

文献标识码: A

文章编号: 1009-5896(2009)06-1319-04

## Regional Texture Synthesis Based Image Inpainting

Fu Shao-chun Lou Shun-tian

(School of Electronic Engineering, Xidian University, Xi'an 710071, China)

**Abstract:** For solving the problem of texture blurring brought by traditional inpainting method, a regional texture synthesis based inpainting algorithm is proposed. The feature in the adjacent domain of each broken pixel is analyzed for choosing the template, and the broken pixels should be excluded from the template. The optimal solution for the broken pixel is determined by matching the template in searching area. The inpainting process proceeds from the edge to the center, for improving the validity and continuity of propagating direction of the texture in the flawed region, meanwhile the blurring can be eliminated. The algorithm is available to a wide variety of real-world textures, and produces a satisfying result for human visual system.

**Key words:** Image inpainting; Texture synthesis; Template choosing; Matching

### 1 引言

作为图像恢复技术的一个分支, 数字图像修补已经成为计算机视觉领域的一个热点问题。最初的图像修补技术主要是非纹理修补, 即针对图像的卡通结构进行恢复。Bertalmio的研究将数字技术引入图像修补领域中<sup>[1]</sup>, Chan等人运用偏微分方程(PDE)进行图像修补<sup>[2]</sup>, 取得了不错的效果, 并在此基础上提出了CDD模型<sup>[3]</sup>。在此之后, Criminisi和Sun等根据沿图像等照度线推进的技术提出了新的算法, 并将其用于去除图像实体的特效技术<sup>[4,5]</sup>。之后, Bertalmio又提出了基于图像灰度级与向量域联合插值的填充方法<sup>[6]</sup>, 以及基于三阶最优偏微分方程的方法<sup>[7]</sup>。

纹理合成技术可以概括为从已知图像中获得有限的纹理样本, 整合这些样本并扩充到新生成的纹理图像中。纹理合成技术的一项重要应用就是在目标图像中生成所需的纹理, 合成的纹理可以扩充至任意大小的图像中。通常将图像的纹理分为规则纹理, 即包含重复出现的纹理结构部分, 和随机纹理, 即包含随机出现的不规则纹理部分, 而实际中所遇到的图像纹理多为这两种类型的结合。基于像素块的方法是以MRF模型为基础, 广泛运用的纹理合成技术之一。这种方法

对于图像编码缺损的恢复具有很好的效果。文献[8]提出的Image Quilting算法, 这是一种规则像素块合成算法, 成功地解决了纹理样本块之间的过渡和衔接问题。之后, Wei将非参数二维合成技术扩展到合成实体纹理领域当中, 但这种方法会造成一定程度的模糊, 同时会丢失细小的纹理结构。Kopf等人又对此算法进行了改进<sup>[9]</sup>。三维空间的纹理合成与覆盖问题也成为了近期的研究热点之一, 文献[10]中所提出的算法也取得了不错的效果。

上述基于像素块的合成算法在产生和扩充结构相同的纹理时是非常有效的, 然而在处理真实图像修补问题时会遇到特殊的问题。由于真实图像中的破损部分通常会包含连续性的纹理, 以及多种不同类型的纹理结构。在破损区域中会含有不同特征集合的边界, 这就使得上述纹理合成方法在应用于图像修补问题时遇到了困难。因此, 直接将纹理合成技术用于图像恢复将不能得到理想的效果。针对上述问题, 本文以像素为基础提出了一种有效的修补算法。这种算法不但能够突出破损域内纹理的延伸与变化, 同时在破损边缘处, 恢复出的纹理具有良好的连续性, 避免出现人眼容易识别的边缘效应。

### 2 基于模板匹配的修补算法

由Wei<sup>[11]</sup>以及Efros<sup>[12]</sup>等提出的纹理合成算法得到的启

发, 本文提出了一种基于像素纹理合成的图像修补算法。算法的目的是要在图像中的缺失部分得到连续、清晰、无缝的纹理。基于像素的修补是以像素为单位合成与修复纹理, 其基本过程是将图像破损区域内的一个像素点作为修补对象, 根据邻域相似性从破损点周围的完好像素中选择一个最佳匹配点, 选中的像素将会替换当前位置上的破损点, 不断重复该过程, 直到图像的破损区域被填满为止。本文算法可以概括为以下几个过程:

- (1) 通过图像掩码确定破损区域;
- (2) 根据破损点的邻域特征确定模板的大小和形状;
- (3) 在搜索区域内将每个像素块与模板进行匹配, 寻找最佳匹配点, 用此像素点替换待修补像素, 修补域之外的像素点保持不变。

## 2.1 模板匹配理论

本文运用马尔可夫域(MRF)模型对纹理修补过程进行建模, 这种模型能够广泛地适用于多种类型的纹理图像。MRF模型是对纹理图像较为客观的一种认识, 也是很多合成算法所采用的建模工具。MRF模型关注纹理的局部特性, 用一块像素区域或是一个像素的邻域部分唯一确定该图像块或像素的亮度值, 因此可以保证图形纹理细节的连续过渡。这里假设一个像素点的亮度值的概率分布只与其空间邻域的亮度值有关, 而与图像其他部分无关<sup>[12]</sup>。因此, 在修补破损像素点时只考虑其邻域内点的像素值。基于上述局域性和稳定性的假设, 本文算法将每个待修补的像素点的邻域作为模板, 并在此模板周围扩展出一个搜索区域以寻找最佳匹配点。

## 2.2 确定修补区域

令  $I(i, j)$  表示破损图像中位于  $(i, j)$  的像素值,  $I'(i, j)$  表示修补后的像素值,  $p$  表示位于  $(i, j)$  的一个像素点。由于已经知道破损的位置, 因此很容易通过一个二值标记量  $f(i, j)$  来表示一个像素是否破损。如果  $f(i, j) = 1$ , 说明像素  $p$  破损;  $f(i, j) = 0$ , 说明像素点  $p$  是完好的。因此, 我们将算法的作用区域设定在  $f(i, j) = 1$  的区域, 而在  $f(i, j) = 0$  的区域像素点保持不变。在数字图像处理中, 一幅图像是通过编码形式表示的, 传输过程中很容易出现信息丢失的现象, 这时可以通过遍历整幅图像来确定破损区域。而对于如照片, 绘画这一类的图像, 图像破损的原因不是像素的丢失, 而是裂痕, 画面损坏等因素, 这就需要人为确定所要修补的区域, 运用特定的算法来得到掩码图像<sup>[13]</sup>。

## 2.3 模板选择

根据 Efros 提出的基于非参数化的 MRF 模型算法, 我们知道一个像素的概率分布只与其邻域像素值有关, 因此将待修补像素  $p$  的邻域  $\Omega = f(p)$  定义为以  $p$  为中心, 大小为  $N \times N$  的正方形窗体, 令  $T = (N - 1) / 2$ ,  $\Omega$  可表示为

$$\Omega = \{p(i + m, j + n) \mid -T \leq m \leq T, -T \leq n \leq T\} \quad (1)$$

为了保证  $p$  作为邻域的中心, 参数  $N$  应选择一个大于 1 的奇数。

这个正方形窗体将作为匹配过程中的基准模板。窗体的大小是一个自由参数, 这个参数将会影响修补区域内纹理的随机性。在纹理合成技术中, 窗口的大小与形状将会直接影响到合成的结果。这个窗体要包含一块完整的纹理结构。而对于修补问题, 实践经验表明, 窗体尺寸的增大并不一定会给纹理修补带来显著的提高。

值得注意的一个问题是, 图像中的破损区域往往不是一个孤立像素, 而是包含连续破损点的一块区域, 如果模板内含有破损点, 将会影响匹配的准确性。也就是说, 上文提出的正方形窗体只是针对领域内无其他破损点的孤立像素而言。算法将在修补区域内进行逐点扫描, 根据待修补点邻域内的破损情况, 不断调整基准模板, 这对于保证纹理的连续性和方向性是非常重要的。因此, 我们选择模板时还要附加一个约束条件, 即去除破损像素正方形邻域内的其他破损点。因此, 将修改后的模板表示为

$$\Omega' = \{p(i + m, j + n) \mid f(i + m, j + n) = 0, -T \leq m \leq T, -T \leq n \leq T\} \quad (2)$$

利用上述二值函数  $f(i, j)$  可以确定某一破损点  $p$  周围  $N \times N$  的邻域内有效像素点的个数  $M$ , 将其表示为

$$M = \sum_{m=-T}^T \sum_{n=-T}^T 1 - f(i + m, j + n) \quad (3)$$

当  $p$  的领域内有效像素个数过少时, 匹配的准确性将会受到影响, 因此要对  $M$  设置一个门限值来控制最小有效像素点的个数。这里以大小为  $N \times N$  的正方形窗体为例, 可将门限值设为  $\lambda$ 。要得到有效的模板, 算法通过下式选择窗体的大小

$$N = \begin{cases} N, & M \geq \lambda \\ N + 2, & M < \lambda \end{cases} \quad (4)$$

因此当模板内有效像素点不符合要求时, 需要扩大模板尺寸, 使其能够包含更多的有效像素点。

## 2.4 模板匹配

以待修补像素  $p$  为中心(即选定的模板中心), 在其周围扩展一个  $S \times S$  的正方形区域  $\Phi$  作为搜索域。纹理合成算法对像素  $p$  的概率分布  $P\{I(p) \mid I(q), q \in \Phi(p)\}$  进行了一种近似, 即在搜索区域内寻找一个像素点, 其邻域与模板“最相似”, 这个点就是  $p$  的最佳匹配点。以  $\Phi$  中的每个像素点为中心, 取与  $p$  点相同的邻域范围作为参与匹配的像素块, 以均方误差(MSE)作为候选像素块与模板匹配程度的判定标准。在  $\Phi$  中找到点  $p'$ , 其邻域与  $p$  点邻域之间的 MSE 值最小, 那么  $p'$  就是  $p$  的一个最佳匹配点, 并将  $p'$  的像素值填充到破损点  $p$  中, 如果存在多个这样的点, 将选择与  $p$  点距离最近的点替换像素  $p$ 。

这里会遇到上述类似的问题, 就是参与匹配的候选块中同样包含破损点。如果破损点参与了匹配过程, 这依然会影响匹配结果, 从而无法得到正确的恢复结果。本文的解决方法是匹配进行之前, 将  $\Phi$  中的每个像素块与模板预先进行比较。事实上如果搜索区域  $\Phi$  包含一块连续破损区域, 要参与

匹配的像素块中必定会包含破损点,因此在匹配时要去除像素块中的破损点,破损点将不参加匹配。

现在以上述  $N \times N$  正方形模板来说明匹配过程。以模板左上方的像素为起点,可以将模板表示成  $\{x(i+m, j+n) | 0 \leq m \leq N-1, 0 \leq n \leq N-1\}$  的形式。进行匹配时首先利用下式计算出可供匹配的完好像素点的个数,

$$\text{Nbr} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} (1-f(u+m, v+n))(1-f(i+m, j+n)) \quad (5)$$

这里对参加匹配的像素个数设置一个最小门限制,当像素块中位于模板相同位置上的完好像素点数小于这一门限值时,这个像素块将会被放弃,以防止得到的 MSE 值不能准确体现近似程度。同样以大小为  $N \times N$  的正方形窗体为例,利用式(5)计算出选定像素块中可用像素个数 Nbr。在此设定当  $\text{Nbr} < \zeta$  时,说明此像素块中包含的可用信息太少,因此匹配过程也将不再执行。算法继续对下一像素块进行检测。

对于符合以上要求的像素块,算法利用像素块中完好像素点,根据均方差(MSE)准则与模板进行匹配。根据下式可以得到搜索域  $\Phi$  中每个像素相应的 MSE 值

$$\text{MSE} = \frac{1}{\text{Nbr}} \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} (1-f(u+m, v+n))(1-f(i+m, j+n)) \cdot [I(u+m, v+n) - I(i+m, j+n)]^2 \quad (6)$$

遍历搜索域  $\Phi$ , 寻找到使得 MSE 达到最小的点,作为待修补像素  $p$  的候选点。如果有多个这样的候选点,可以随机选取,而更好的方法是选择与  $p$  点距离最短的点<sup>[11]</sup>。

### 2.5 边缘处理

在文献[11]中,为消除纹理填充区域的不连续性,算法采用了以图像金字塔理论为基础的对称邻域扩展方法。同时,算法以螺旋式像素搜索代替光栅式的搜索,也有效地减少了在边缘处产生的纹理延伸方向的偏差。本文提出的修补模型沿边缘对破损像素逐个进行修补,突出了边缘纹理结构的变化,从而避免破损区域出现模糊边缘。同时,本文采用由外向内逐次推进的修补次序,与螺旋式搜索类似,能够有效地保持纹理的延伸方向,使得边缘处的纹理结构误差尽量减小。

### 2.6 修补次序

为了保持破损区域内纹理的连续性,本文采用了一种从破损区域边缘开始,由外向内逐次推进的修补方法。其实现方法是:修补之前,遍历整个破损区域,利用式(3)计算破损域内每个像素周围有效邻域点的个数。算法将从具有最多有效邻域像素的破损点开始修补,由此实现从破损边缘向中心的推进过程。实践经验说明修补次序对纹理结构延伸方向的正确性与连续性有显著的影响。本文通过实验发现,采用传统的光栅式修补次序时,在纹理结构复杂,特别是多种纹理混合的情况下,纹理的延伸方向会出现偏差,这将影响整个纹理结构的走向。同时,由于纹理结构的这种方向性偏差会导致在修补边缘处出现比较明显的假边缘,因此破坏了纹理结构的连续性。本文采用的由外向内的修补次序能够很好地保持边界处纹理结构的方向性,避免了虚假边缘的出现,在

实践中取得了很好的效果。

## 3 实验结果

在 MATLAB 环境下对本文提出的算法进行真实图像的恢复实验。在仿真过程中,模板的最大尺寸取为  $5 \times 5$  的正方形区域,搜索域的范围设定为以待修补点为中心,大小为  $9 \times 9$  的正方形区域。设定模板内有效像素点个数的门限值  $\lambda = 8$ ,可用匹配像素点个数的门限值为  $\zeta = 5$ 。实验中,以  $512 \times 512$  的 Lena 和 Baboon 图像中纹理丰富的部分来验证算法的修复性能。这里假设原始图像是已知的,为检验算法性能,本文采用标准峰值信噪比(PSNR)来评价算法的修复性能:

$$\text{PSNR} = 10 \lg \left( \frac{255^2}{\|u - u_0\|_2^2} \right) \quad (7)$$

式中  $u_0$  表示完好的原始图像,  $\|u - u_0\|_2$  表示标准  $L^2$  范数。

图1与图2给出了本算法与非纹理修补模型(CDD模型)方法的修补结果。从实验结果中可以看出,CDD模型得到的结果在纹理丰富的破损区域呈现模糊状,而且人眼很容易识别出修补的痕迹。本文算法的结果能够在破损域内恢复出与原图像相似的纹理结构,同时在修补的边缘处纹理连续性较好,而且人眼不易识别出修补的痕迹。算法对图像纹理部分的修补效果更为显著,而这正是基于卡通结构(即非纹理)恢复方法<sup>[1,2]</sup>所欠缺的。本文算法与CDD模型算法得到的PSNR值如表1所示,可以看出与CDD算法相比,本文算法得到的修补图像具有更高的PSNR值,即修补结果更加接近原始图像。

## 4 结束语

本文首先介绍了图像修补领域中的非纹理修补算法,针

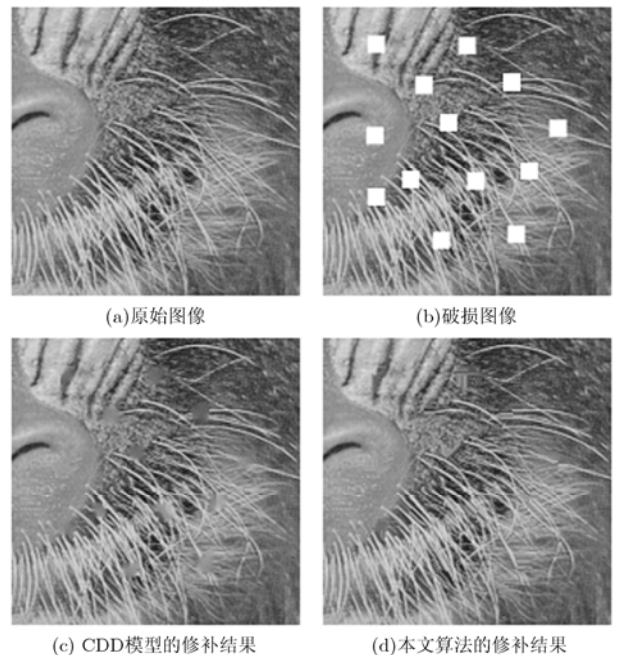


图1 本文算法与CDD模型算法对Baboon图像的修补结果

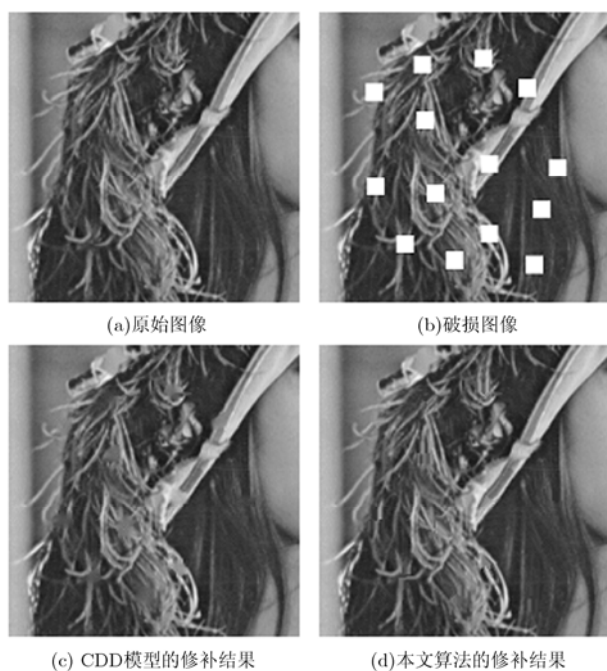


图2 本文算法与 CDD 模型算法的对 Lena 图像的修补结果

表1 破损图像与修补后图像的 PSNR 比较

原图	破损图(dB)	CDD 模型算法(dB)	文本算法(dB)
Baboon	18.7934	31.5951	32.1313
Lena	16.9721	33.7619	34.9754

对这种算法在恢复图像纹理结构方面的不足,以纹理合成技术为基础,提出了一种基于像素纹理合成的图像修补算法。实验说明,本文在纹理修复方面比非纹理 CDD 模型算法具有明显的优势。同时本文算法得到的恢复结果更符合人类视觉系统的特征,能够突出纹理变化的连续性,使得人的视觉几乎察觉不到修补痕迹。算法的另一个优势就是实现简单,处理时间短。算法所选择的搜索区域的大小是一个自由参数。通过实验发现,选择的正方形窗体过大并不能提高恢复效果,同时会出现纹理重叠或是假纹理的现象。这里需要进一步的理论研究,以得到最优搜索窗的大小。本文算法可以与非纹理修补算法结合使用,对图像分别进行卡通结构与纹理结构的修复。文献[14]已经在这方面做了探讨。另外,本算法还可以应用于图像中物体的去除,文献[15]对此做了详细的说明。

### 参考文献

- [1] Bertalmio M, Sapiro G, and Caselles V, *et al.* Image Inpainting. Proceedings of the 27th annual conference on Computer graphics and interactive techniques, New Orleans, 2000: 417-424.
- [2] Chan T and Shen J. Mathematic method for local non-texture inpainting[J]. *SIAM J. Appl. Math.*, 2001, 62(3): 1019-1043.
- [3] Chan T and Shen J. Non-texture inpainting by curvature-driven diffusions (CDD) [J]. *Visual Communication and Image Representation*, 2001, 12(4): 436-449.
- [4] Criminisi A, Perez P, and Toyama K. Region filling and object removal by exemplar-based image inpainting[J]. *IEEE Trans. on Image Processing*, 2004, 13(9): 1200-1212.
- [5] Sun J, Yuan L, and Jia J, *et al.* Image completion with structure propagation[J]. *ACM Trans. on Graphics*, 2005, 24(3): 861-868.
- [6] Ballester C, Bertalmio M, and Caselle V, *et al.* Filling-In by Joint Interpolation of Vector Fields and Gray Levels[J]. *IEEE Trans. on Image Processing*, 2001, 10(8): 1200-1211.
- [7] Bertalmio M. Strong-continuation contrast-invariant inpainting with a third-order optimal PDE[J]. *IEEE Trans. on Image Processing*, 2006, 15(7): 1934-1938.
- [8] Efros A and Freeman W. Image quilting for texture synthesis and transfer. Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Los Angeles, 2001: 341-346.
- [9] Kopf J, Fu C W, and Cohen-Or D, *et al.* Solid texture synthesis from 2D exemplars[J]. *ACM Trans. on Graphics*, 2007, 26(3): 2.
- [10] Zhou K, Huang X, and Wang X, *et al.* Mesh quilting for geometric texture synthesis[J]. *ACM Trans. on Graphics*, 2006, 25(3): 690-697.
- [11] Wei L Y and Levoy M. Fast texture synthesis using tree-structured vector quantization. Proceedings of the 27th annual conference on Computer graphics and interactive techniques, New Orleans, 2000: 479-488.
- [12] Efros A and Leung T K. Texture synthesis by non-parametric sampling. Proceedings of the 7th IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999: 1033-1038.
- [13] Wen L, Zhang D P, and Liu Z Y, *et al.* A parallel algorithm for image information restoration. Proceedings of the 4th International Conference on High Performance Computing in the Asia-Pacific Region, Beijing, 2000: 790-793.
- [14] Bertalmio M, Vese L, and Sapiro G. Simultaneous structure and texture image inpainting [J]. *IEEE Trans. on Image Processing*, 2003, 12(8): 882-889.
- [15] Tauber Z, Li Z N, and Drew M. Review and preview: Disocclusion by inpainting for image-based rendering[J]. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2007, 37(4): 527-540.

付绍春: 男, 1983年生, 硕士生, 研究方向为图像修补。

楼顺天: 男, 1962年生, 教授, 博士生导师, 研究方向为盲信号处理、图像恢复、智能控制理论及应用等。