

二维纹理合成算法优化研究

杜常青¹ 钱文华²

(云南曲靖师范学院计算机科学与工程学院 曲靖 655000)¹

(云南大学计算机科学与工程系 昆明 650091)²

摘要 基于已有的二维纹理合成算法,提出了基于样图纹理合成的优化算法。在合成过程中,通过多种子点选择、以螺旋状路径在原图中搜索新匹配点、边界扩充、L2 距离计算优化等技术加速合成过程,提高合成质量,实验结果取得了令人满意的效果。

关键词 二维纹理,纹理合成,L2 距离,匹配搜索

中图分类号 TP391.9 **文献标识码** A

Optimization Technique of 2D Texture Synthesis

DU Chang-qing¹ QIAN Wen-hua²

(School of Computer Science and Engineering, Qujing Normal University, Qujing 655000, China)¹

(Department of Computer Science and Engineering, Yunnan University, Kunming 650091, China)²

Abstract The paper proposed a framework for exemplar-based texture synthesis. Our main idea is reducing synthesis time consuming and advancing synthesis quality by some optimization methods. During the course of process, many optimizations such as searching along a spiral path, edge expansion, L2 distance computing predigesting, accelerate synthesis speed and advance the final results. Experiments show that synthesis results are quite satisfactory and effective.

Keywords 2D texture, Texture synthesis, L2 distance, Matching search

1 引言

纹理图案的分布或者呈周期性或者呈随机性,通过输入小块纹理样本,利用纹理合成算法将小块纹理样本合成为较大空间的纹理图像,输出图像中具有自相似性。纹理合成主要采用过程纹理合成和基于样图的纹理合成两种算法。过程纹理合成通过物理过程的仿真实现纹理合成的效果,但该方法对每一种新纹理都需要设置复杂的参数反复测试;基于样图的纹理合成避免了繁琐的参数调试,受到很多研究者的关注。纹理合成技术在图像编辑、数据压缩等方面发挥作用,此外,纹理传递、纹理补色、时频领域生成视频动画等技术在真实感与非真实感效果的表现方面也有广泛的应用前景。

本文将根据纹理的局部统计特征和自相似性,采用特征匹配的方法搜寻匹配像素点,通过搜索路径改进、L2 距离优化、边界填充等方法对已有的纹理合成算法进行改进,以加速纹理合成过程,改善纹理合成质量。

2 纹理合成技术的相关工作

早期的纹理合成采用过程纹理合成的方法,通过对纹理进行分析,通常采用统计的方法进行纹理合成。Wei 采用多分辨率模型,利用每个像素与它周围的像素都具有相似性的特点,将像素点的邻域改为 L 型邻域,在搜索过程中将最佳

匹配中心像素点的值填充到结果图像中,利用矢量量化方法加速了合成的过程,算法简单、易于理解,执行效率较高,提高了合成的速度^[1]。Wei 算法是基于像素点的合成,对于随机性较强的纹理能够取得满意的合成效果,对于结构性较强的纹理合成质量较差。此外,他们的算法会模糊结果图像的细节如边缘等信息。

Efros 提出 Image Quiting 的纹理合成方法,其从样本纹理图中选取纹理块填充到合成结果图中,块与块之间采用最小匹配路径搜索待合成像素点,着重解决纹理块之间的接缝问题^[2]。算法合成最后的纹理效果图像要经过选取纹理块填充待合成结果图、调整纹理块区域、寻找最佳缝合线、平滑处理 4 个步骤来完成。它对绝大多数的自然纹理合成都能取得很好的效果。然而,最佳缝合线的寻找有时会导致效率降低,缝合线产生的边缘有时清晰可见。

Kwatra 等人提出一种基于 Graph Cut 的合成算法,算法无需设置填充块的大小等参数,以整个输入纹理样本作为填充单元,以选定位移位置填充到合成图中,算法基于 Graph Cut 来决定最终的输出区域^[3]。文献[4]针对基于结构性纹理合成 Image Quilting 技术上存在的缺陷,不仅考虑了两个匹配块之间的颜色相似性,还充分考虑了结构的差别,获得了比较理想的合成结果。基于 Wei 和 Efros 的纹理合成方法,文献[5]通过提取图像的相关纹理特征,利用图像分解技术对

到稿日期:2012-06-24 返修日期:2012-10-20 本文受国家自然科学基金(60663010,61163019),云南省自然科学基金(2010CD024)资助。

杜常青(1980-),男,硕士,讲师,主要研究方向为数字图像处理;钱文华(1980-),男,讲师,主要研究方向为图像处理、计算机非真实感等, E-mail: qwhua003@sina.com。

图像进行预处理,利用图像灰度共生矩阵提取待修复图像的纹理特征,通过确定相关纹理特征与最优样本块大小的相互关系,实现了样本块大小的自动选取,然而,合成速度和合成质量仍然有待进一步提高。

3 纹理合成算法优化

基于 Wei 以及 Efros 的思想,论文提出了相关的合成算法优化措施,旨在提高合成速度和合成质量。与 Wei 和 Efros 的算法相比,改进的算法表现在以下几个不同的方面:

(1)采用多个种子:在合成纹理图像中采用多个种子进行预填充,同时,从每个种子点开始往邻域扩充,以实现算法的并行计算。

(2)扩充边界:合成之前先对边界进行扩充,在输入纹理样本图和待合成图边界之外增加半个 L 型邻域宽度的样本纹理(见图 1),纹理合成时从原边界位置开始进行。在选取像素点扩充边缘时,将纹理样本中待选取的某一像素值与原边界某一像素值进行比较,若颜色差值小于预先设置的阈值,则将选取的像素值填充到扩充边界中,否则在纹理样本中重新选取。

由于合成过程中避免了没有纹理关系的行和列出现,在样本图和合成图任何位置都能取到整个 L 型邻域进行计算,边界处的像素点不需进行特殊处理,以便合成质量的提高。

(3)采用螺旋状搜索:采用螺旋线状搜索最佳匹配点,搜索的初始位置为上一候选点的匹配位置,按照顺时针或者逆时针方向往邻域扩充,螺旋线状搜索方法有助于提高合成的速度。

论文在搜索后续点的过程中,从前一点匹配点的当前位置开始,由其邻域向外做螺旋状搜索,若匹配误差小于设定的阈值,则搜索终止,将该像素点填充到合成图中;若遍历了样图仍然没找到满足误差要求的点,则选取误差最小的像素点写入合成图中。由于像素与它周围像素点的相关性,进行少量的搜索便可得到结果。图 1 描述了搜索过程。

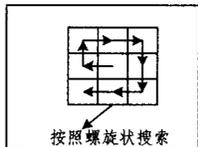


图 1 从原匹配点位置按照螺旋状搜索

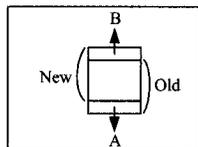


图 2 匹配窗口的上下移动

(4)采用矩形邻域代替 L 型邻域搜索:在合成结果图中,Wei 采用的 L 型搜索邻域只考虑用已合成的区域进行比较,对未合成的区域没有考虑。由于纹理图像中某一像素与它周围像素的相关性,未合成区域对当前像素点仍具有一定贡献,因此,Wei 采用的 L 型搜索邻域与后面的预置种子可能存在边界。本文采用矩形邻域搜索的方法,代替 Wei 采用的 L 型搜索邻域。

(5)L2 距离计算优化:Wei 对 L 型邻域的计算每次都重新开始,计算中存在一定冗余。由于相邻 L 型邻域的计算存在着大量相同的像素,只有少量像素加入或退出窗口,因此,减少计算的冗余信息可提高合成速度。我们采用矩形窗口按照螺旋型的顺序搜索,因此,矩形窗口只存在水平和垂直方向上的移动两种情况,图 2 说明了匹配窗口的上下移动过程。假设 N_1 、 N_2 为纹理样本和合成结果图中的不同像素点, N

为匹配窗大小,初始化时,利用式(1)在纹理样本图中搜索 L2 距离最小的匹配点,记录下匹配点的位置。

$$D(N_1, N_2) = \sum_{p, q \in N} \{ (R(p) - R(q))^2 + (G(p) - G(q))^2 + (B(p) - B(q))^2 \} \quad (1)$$

当匹配窗口向上移动时,如图 2 所示,old 为上一次窗口所在区域,L2 距离为 $D_{old}(N_1, N_2)$,new 为窗口向上移动时所在区域,假设这一区域的 L2 距离为 $D_{new}(N_1, N_2)$,蓝色为窗口移动时的重叠区域,可减少对这部分重叠区域的计算来提高合成速度。假设图 2 中黄色区域 A 的 L2 距离为 D_A ,红色区域 B 的 L2 距离为 D_B ,匹配窗口向上移动后的 L2 距离可表示为:

$$D_{new}(N_1, N_2) = D_{old}(N_1, N_2) - D_A + D_B \quad (2)$$

式(2)在计算 L2 距离时避免了窗口移动时重叠区域的重复计算,有助于提高算法效率。论文通过选取多种子进行填充、扩充边界,采用矩形匹配窗口、L2 距离计算优化,改进了 Wei 和 Image Quiting 的纹理合成算法,得到了基于纹理样图的具体实现过程。

4 改进纹理合成算法的实现

综上所述,改进的纹理合成算法的实现过程如图 3 所示,具体实现步骤如下。

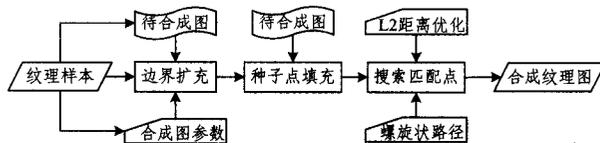


图 3 纹理合成流程图

Step1 输入纹理样本图,指定最终合成结果图的大小,设定匹配窗大小,误差门限值。采用矩形作为匹配窗,至少要覆盖纹理基元的大小。

Step2 扩充原纹理样本图和最终合成结果图的边界。将颜色差值小于指定阈值的像素点填充到纹理样本图和合成结果图的边界中。

Step3 从原纹理样本图中随机选取种子按扫描线顺序填充到结果图中。种子可以选取单一像素,也可以选取具有某种形状诸如点、线、矩形等的像素集合。一般采用较为简单处理的 $N \times N$ 的矩形作为种子,有时根据需要也可采用线状种子,种子数量需要根据种子的形状、大小、合成最终纹理图的大小、合成纹理质量等进行综合考虑。

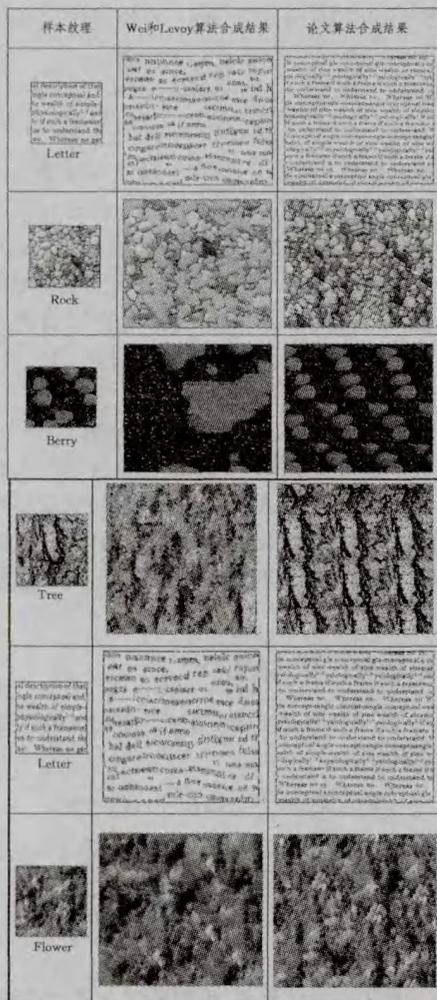
Step4 按扫描线顺序搜索合成图中未被填充的像素点,从样本图中搜寻 L2 距离小于给定阈值(可取均方差与某一常数的乘积)的像素点填充到结果图中,并记录下匹配点的位置。

Step5 从上一匹配点位置按照螺旋状的顺序搜寻合适的像素点填充到待合成像素点。

Step6 重复 Step4 和 Step5,直到合成最终的纹理结果图。

5 实验结果

本文根据以上算法思想进行了相关实验,纹理样本为 64×64 ,基于 Matlab7.0 实验平台,最终合成结果图大小为 200×200 。将论文合成结果与 Wei 和 Levoy 算法合成结果进行比较,图 4 显示了比较的结果。



左图为纹理样本,中间为 Wei 结果,右图为本文合成结果

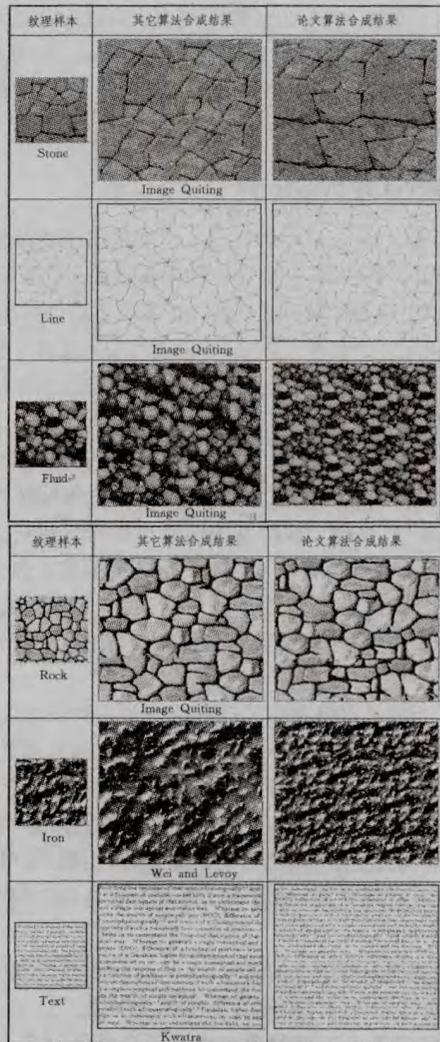
图4 纹理合成结果图

从图4的结果对比可以得到,由于 Wei 和 Levoy 采用基于像素点的纹理合成方法,对于结构性较强的纹理样本,合成的结果会导致纹元的破裂和不完整。同时,合成结果会产生模糊、不连续等现象。论文采用基于纹理块的合成方法加速了合成过程,以纹元为单位进行纹理合成,减少了合成结果图的模糊和不连续现象,质量有所提高。

图5为论文合成结果与其它算法合成结果的比较。本文取得了与 Wei、Image Quiting 和 Kwatra 相似的效果。从表1可以看出,由于采用多种子填充、边界扩充、矩形匹配窗、螺旋式搜索以及 L2 距离的优化措施,加速了纹理合成的过程。

表1 算法的合成时间比较

纹理样本	样本大小	匹配窗	合成图大小	论文合成时间	其它算法合成时间
Stone	64 * 64	20 * 20	200 * 200	12.36s	19s (Image Quiting)
Line	64 * 64	20 * 20	200 * 200	19.07s	24s (Image Quiting)
Fluid	64 * 64	20 * 20	200 * 200	17.03s	22s (Image Quiting)
Rock	64 * 64	20 * 20	200 * 200	22.15s	26s (Image Quiting)
Iron	64 * 64	20 * 20	200 * 200	8.86s	22s (Wei&Levoy)
Text	64 * 64	9 * 9	128 * 128	16.12s	30s (Kwatra)



左图纹理样本,中间为其它算法结果,右图为本文合成结果

图5 纹理合成结果图

结束语 纹理合成是纹理补色、纹理传输、视频动画等艺术效果图像产生的基础,在真实感与非真实感效果的表现方面也有广泛的应用前景。论文通过采用多个种子点、边界扩充、螺旋状搜索匹配像素点、矩形邻域代替 L 型邻域搜索、L2 距离优化的方法提高了合成的速度,获得了较好的纹理合成结果,该算法简单,容易实现。今后的工作中,将在三维纹理合成、视频纹理合成技术方面做更深入的研究,探索硬件加速达到纹理合成的实时合成技术也是将来重点研究的方向。

参考文献

- [1] Wei L Y, Levoy M. Fast Texture Synthesis Using Tree-structured Vector Quantization[C]// Proceedings of the Computer Graphics, Annual Conference Series, 2000; 479-488
- [2] Efros A A, Freeman W T. Image Quilting for Texture Synthesis and Transfer[C]// Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 2001; 341-346
- [3] Kwatra V, Schodl A, Essa I. Graphcut Textures: Image and Video Synthesis Using Graph Cuts[J]. Proceedings of the ACM Transactions on Graphics, 2003, 22(3): 277-286
- [4] 蔡志林. 基于样图的纹理合成实现[D]. 无锡: 江南大学, 2011
- [5] 张伟伟, 何凯, 等. 自适应选取样本块大小的纹理合成方法[J]. 计算机工程与应用, 2012, 17(4): 170-173