



计算机科学

COMPUTER SCIENCE

基于接缝一致性准则的结构纹理图像快速合成算法

靳利贞, 李庆忠

引用本文

靳利贞, 李庆忠. [基于接缝一致性准则的结构纹理图像快速合成算法](#)[J]. 计算机科学, 2022, 49(6): 262-268.

JIN Li-zhen, LI Qing-zhong. [Fast Structural Texture Image Synthesis Algorithm Based on Seam Consistency Criterion](#)[J]. Computer Science, 2022, 49(6): 262-268.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于融合元路径图卷积的异质网络表示学习](#)

Graph Convolution of Fusion Meta-path Based Heterogeneous Network Representation Learning
计算机科学, 2020, 47(7): 231-235. <https://doi.org/10.11896/jsjcx.190600085>

[融入结构信息的指代消解](#)

Coreference Resolution Incorporating Structural Information
计算机科学, 2020, 47(3): 231-236. <https://doi.org/10.11896/jsjcx.190100108>

[用于文本分类的 CNN_BiLSTM_Attention 混合模型](#)

CNN_BiLSTM_Attention Hybrid Model for Text Classification
计算机科学, 2020, 47(11A): 24-27. <https://doi.org/10.11896/jsjcx.200400116>

[合成纹理图像的视觉相似性眼动分析](#)

Eye-movement Analysis of Visual Similarity Perception on Synthesized Texture Images
计算机科学, 2018, 45(8): 223-228. <https://doi.org/10.11896/j.issn.1002-137X.2018.08.040>

[自相似性匹配的纹理合成](#)

Texture Synthesis Based on Self-similarity Matching
计算机科学, 2018, 45(6A): 215-219.

基于接缝一致性准则的结构纹理图像快速合成算法

靳利贞 李庆忠

中国海洋大学工程学院 山东 青岛 266100

(jinlizhen@stu.ouc.edu.cn)

摘要 基于块的结构纹理图像合成算法存在诸多问题,如结构不连续、边界扭曲、接缝不齐以及合成速度慢等,对此提出了一种新的基于双接缝一致性准则的纹理图像非重叠拼接快速合成算法,该算法有效提高了结构纹理图像的合成质量与速度。首先基于更符合人眼视觉特性的 HSI 颜色空间建立了同时考虑色调、饱和度、亮度和边缘特征的接缝线一致性判别准则;然后提出并实现了基于双接缝线一致性准则的子块搜索策略和非重叠拼接算法。实验结果表明,与传统算法相比,所提算法能够显著提高结构纹理图像的合成质量和速度。

关键词 纹理合成;接缝线一致性;非重叠拼接;HSI 颜色空间;结构信息

中图分类号 TP391

Fast Structural Texture Image Synthesis Algorithm Based on Seam Consistency Criterion

JIN Li-zhen and LI Qing-zhong

School of Engineering, Ocean University of China, Qingdao, Shandong 266100, China

Abstract Aiming at the problems of patch-based synthesis algorithm of structured texture images, such as discontinuity of structure, distortion of boundary, seam misalignment, and low synthesis speed, a new fast non-overlapping synthesis algorithm of texture images is proposed based on the consistency criterion of double-seam lines, thereby effectively improving the synthesis quality and speed of structured texture images. Firstly, the seamline consistency criterion considering hue, saturation, brightness and edge characteristics simultaneously is established in HSI color space that is more consistent with human visual characteristic. Then, a sub-block search strategy and a new non-overlapping splicing algorithm based on the consistency criterion of double-seam line are proposed and implemented. The experiment results show that the proposed algorithm can significantly improve the synthesis quality and speed of structured texture images in comparison with the traditional algorithms.

Keywords Texture synthesis, Seam line consistency, Non-overlapping splicing, HSI color space, Structure information

1 引言

基于样图的纹理图像合成指根据某种形式的小样本纹理图像生成大视场的纹理图像,同时保证生成的纹理图像在视觉上与样本图像具有相似性和连续性。该技术在数据压缩、图像编辑、图像修复和大规模场景绘制等方面具有重要的应用价值^[1-5]。

Raad 等^[6]按照图像合成过程中过程参数的有无,将纹理合成技术分为两类,即基于统计的纹理合成算法^[7-9]和基于块的纹理合成算法^[10-16]。第一类算法通过估计描述潜在随机过程的统计参数来表征给定纹理样图。这类方法尽管可以很好地再现样图的部分全局统计数据,并能较好地合成随机纹理和半结构化纹理,但对结构纹理(按照固定的纹元结构有序排列而成的纹理)的合成效果并不理想,特别是对于纹理样图较小但包含较大对象的情况。第二类算法以一定的方式重新排列输入纹理样图的局部子块,进而将其拼接成一幅在视觉

上与输入样图具有相似性和连续性的大纹理合成图像^[10]。Efros 等^[11]首次提出了一种基于块的图像缝合算法,该算法按从左往右、从上至下的扫描线顺序,通过计算重叠区域的颜色误差来搜索最佳匹配块,然后依次按照子块重叠方式逐块合成纹理图像,并在相邻块的重叠区域内寻找颜色误差最小的切割路径来缝合相邻块,以消除视觉上颜色的不连续性。在文献[11]的基础上,Zhu 等^[12]提出了动态计算缝合块大小,以自相似匹配方式计算相似度,并引入贪心算法思想来提高边界重合度的方法,该方法对于多种纹理图像都能取得较好的合成效果。Pu 等^[13]用 r 型重叠区域取代 L 型重叠区域,将寻找颜色误差最小路径的区域从一个块扩大到同一行的所有块,使算法在保证合成质量的同时加快合成速度,但此方法对于更结构化的纹理合成结果仍会出现纹理单元破裂和不完整的情况。Lara 等^[14]利用快速傅里叶变换计算颜色误差,并实现了拼接过程的部分并行化,在多核平台上明显提高了运行速度。为了提高结构化纹理图像的合成质量,Shen

到稿日期:2021-04-02 返修日期:2021-09-06

基金项目:国家重点研发计划项目(2017YFC1405202);海洋公益性行业科研专项(2016050021)

This work was supported by the National Key Research and Development Program of China(2017YFC1405202) and Ocean Public Welfare Scientific Research Project(2016050021).

通信作者:李庆忠(liqingzhong@ouc.edu.cn)

等^[15]用 Hausdorff 距离度量结构的相似度,以引导子块的搜索。此外,Tang 等^[16]利用对旋转、噪声和尺度都有很好鲁棒性的 Chamfer 距离来度量纹理结构的相似度;同时,为解决 Chamfer 距离计算量巨大而费时的问题,采用 GPU 并行计算来提高合成效率,从而达到在提高合成质量的同时也保证了合成效率的目的。近年来,Rafi 等^[17]提出了一种新的基于灰度共生矩阵和限制互相关的块合成算法,并设计了一种简单有效的基于最小差值的过渡区融合方法。该方法避免了在重叠区的两个相邻块之间追溯最小切割路径的必要性,但只能高质量地合成随机纹理和半结构化纹理的灰度图像。

总之,以上基于块的纹理合成方法虽然对于一般的纹理图像可以得到较好的合成质量,但对结构化较强的纹理图像的合成仍然存在诸多问题,如结构不连续、边界扭曲、接缝走样以及合成速度过慢等。为解决以上问题,本文提出了一种基于双接缝一致性准则的结构纹理图像快速合成算法,该算法在符合人眼视觉特性的 HSI 颜色空间建立了同时考虑接缝曲线的色调、饱和度、亮度和边缘特征一致性的判别准则;同时,提出了基于双接缝线匹配准则的子块搜索策略和非重叠拼接算法。所提算法在显著改善结构纹理图像合成质量的同时,还具有较好的实时性。

2 基于双接缝一致性的纹理合成算法

为了在提高结构性较强纹理图像的合成质量的同时保证纹理合成的快速性,本文提出了一种基于双接缝一致性的纹理图像快速合成算法,如图 1 所示。

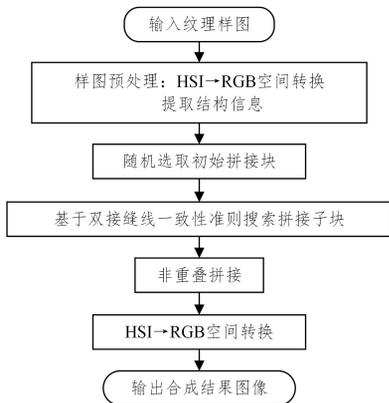


图 1 基于双接缝一致性的纹理合成算法框架

Fig. 1 Framework for texture synthesis algorithms based on the consistency criterion of double-seam line

本文算法的整体流程如下:

- (1) 输入 RGB 颜色空间的纹理样图;
- (2) 对输入纹理样图进行预处理,包括 HSI 颜色空间转换和边缘结构信息提取;
- (3) 从纹理样图中随机选取一匹配块,将其置于 HSI 合成结果图像的左上角作为合成的初始块;
- (4) 根据双接缝一致性匹配准则,寻找与待合成块匹配最佳的候选块并进行非重叠拼接,直至所有子块拼接完成;
- (5) 将最终合成的结果图像转换至 RGB 颜色空间,并输出合成结果图像。

由以上算法步骤可知,整个算法主要由样图预处理模块、基于双接缝线一致性准则搜索拼接子块模块以及非重叠拼接模块组成。

2.1 样图预处理

为了更加客观地评价相邻拼接子块的颜色与结构一致性,本文对输入的纹理样图进行了颜色空间转换和边缘结构信息提取两步预处理。将纹理样图从 RGB 颜色空间转换至便于视觉感知的 HSI 颜色空间,并以亮度图像(即 I 分量)的边缘检测结果作为样图的边缘结构信息。

2.1.1 颜色空间转换

HSI 颜色空间用 3 个属性描述色彩,即色调、饱和度以及亮度。该空间从人的视觉系统出发,具有两大重要特征:亮度属性不随其他两个属性的改变而改变;色调与饱和度属性符合人眼对景物颜色的感知。RGB 颜色空间到 HSI 颜色空间的转换公式如下:

$$\begin{cases} H = \begin{cases} \theta, & B \leq G \\ 360^\circ - \theta, & B > G \end{cases} \\ \theta = \arccos \left\{ \frac{\frac{1}{2}(R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\} \\ S = 1 - \frac{3}{(R+G+B)} [\min(R+G+B)] \\ I = \frac{1}{3}(R+G+B) \end{cases} \quad (1)$$

2.1.2 边缘结构信息提取

在确保边缘检测准确性和快速性的前提下,为了获得更详尽且连续的边缘,在 Sobel 算子的基础之上加入 45° 和 135° 方向的梯度模板,并将其分别与样图进行卷积运算,得到 4 个方向的梯度分量 G_0, G_{45}, G_{90} 和 G_{135} ,以及梯度幅值 G 。

$$\begin{cases} G_0 = (P_{31} + 2P_{32} + P_{33}) - (P_{11} + 2P_{12} + P_{13}) \\ G_{45} = (P_{23} + 2P_{33} + P_{32}) - (P_{12} + 2P_{11} + P_{21}) \\ G_{90} = (P_{13} + 2P_{23} + P_{33}) - (P_{11} + 2P_{21} + P_{31}) \\ G_{135} = (P_{12} + 2P_{13} + P_{23}) - (P_{21} + 2P_{31} + P_{32}) \end{cases} \quad (2)$$

$$G = \sqrt{G_0^2 + G_{45}^2 + G_{90}^2 + G_{135}^2} \quad (3)$$

其中, P_{ij} ($i, j = 1, 2, 3$) 为目标像素的 8 邻域。

对于得到的边缘图像,采用大津算法(Otsu)^[18]来确定梯度图像 G 的阈值 T ,将梯度图像 G 分为背景和强边缘前景两部分。记前景像素点数占整幅图像的比例为 r_1 ,前景平均梯度为 u_1 ,背景像素点数占整幅图像的比例为 r_2 ,背景平均梯度为 u_2 ,则图像总体均值为:

$$u = r_1 u_1 + r_2 u_2 \quad (4)$$

阈值 t 的最大类间方差的表达式为:

$$f(t) = r_1 (u_1 - u)^2 + r_2 (u_2 - u)^2 \quad (5)$$

当 $f(t)$ 取最大值时,对应的 t 即为最佳阈值 T ,利用该阈值对梯度图像进行二值化,便能获得样图的边缘结构信息。

2.2 非重叠拼接过程

在基于块的图像缝合算法^[10-15]中,相邻拼接块之间存在 3 种类型的重叠区域,在重叠区域中寻找一致性最佳的缝合路径来缝合相邻子块,具体如图 2 所示。可见,基于块的图像缝合算法在合成速度方面主要存在两个问题:1)重叠区域计算量较大且存在计算冗余;2)在重叠区域内寻找误差最小的切割路径缝合相邻块比较费时。导致上述问题的主要原因是这些基于块的合成方法均利用重叠区域去搜索待拼接子块,重叠区域的所有像素都要参与计算,因此计算量大且耗时。特别是如图 2(c)所示的 L 型重叠的深灰色区域,涉及到左侧的垂直重叠平方差和 (Sum-of-Squared Differences, SSD)

误差、顶部的水平重叠 SSD 误差以及二次重叠部分的 SSD 误差这 3 种不同的误差计算,导致了计算冗余。虽然在文献 [13]中,L 型重叠被优化为如图 2(d)所示的 r 型重叠区域,在一定程度上解决了计算冗余的问题,提高了运算速度,但是参与计算的数据量依然不少。同时,由于使用了重叠区域,为使块与块之间的连接更为平滑,需要花费一定时间来寻找某条最佳切割路径以缝合相邻块。

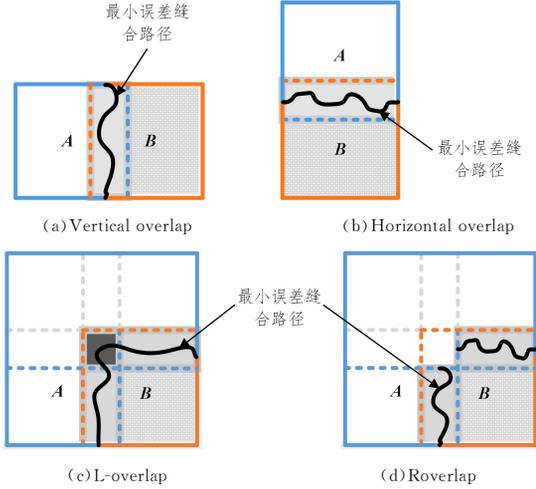


图 2 重叠区域与缝合路径示意

Fig. 2 Diagram of overlap regions and stitching paths

为解决以上问题,本文根据人类在拼图过程中主要考查相邻拼图块接缝处的一致性原则,提出了一种基于双接缝一致性准则的非重叠拼接纹理图像快速合成算法。该算法的拼接过程如图 3 所示。下文以水平方向相邻子块拼接过程为例,来说明本文算法的拼接过程。

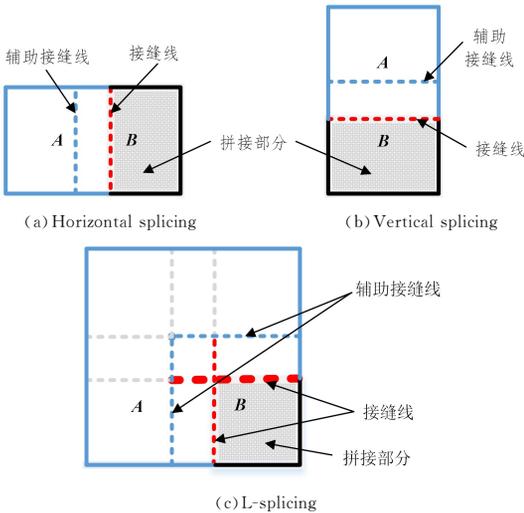


图 3 基于双接缝线一致性准则的拼接过程示意(电子版为彩图)

Fig. 3 Diagram of the splicing process based on the double-seam line consistency criterion

由图 3(a)可见,已合成块 A(白色区)与待合成子块 B(阴影区)的真正接缝线是红色标注的垂直虚线,理论上只要 A 与 B 的红色接缝线完全一致,就可以把 A 与 B 子块在红色接缝线处直接拼合在一起。但是,只满足单一接缝线一致性条件,有时并不能保证相邻子块拼接合成后结构图案的合理性。如图 4 所示,对于左边已合成的子块(红色部分),在样图中寻找与之接缝一致的子块,虽然接缝线①、接缝线②都满足一致性

条件,但只有按接缝线①拼接后合成的结构图案才是合理的,按接缝线②拼接会出现不合理的合成结构图案。

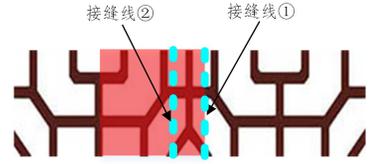


图 4 按单一接缝线一致准则拼接出现的问题(电子版为彩图)

Fig. 4 Problem of stitching process based on single seam line consistency criterion

为避免此类问题产生,在图 3(a)中增加一条蓝色标注的辅助接缝线,即红色与蓝色两条垂直接缝线必须同时满足一致性条件,B 子块才是 A 子块的最佳拼接块,这时 B 块直接在红色接缝处与 A 块贴合即可,不必像传统的基于块的图像缝合算法那样要求取最佳缝合路径。可见,本文算法的拼接属于非重叠拼接,只是增加了一条辅助接缝线,以辅助子块的匹配判别。垂直方向拼接以及 L 型拼接过程与水平方向拼接类似,都是根据双接缝一致性准则搜索待拼接子块,在满足条件的待拼接子块中随机选取一个子块,直接按照真正的接缝进行无重叠拼接即可。显然,上述方法既能保证高精度拼接,又大大缩短了拼接计算时间。

2.3 双接缝线一致性准则

为了衡量相邻拼接子块接缝处的一致性,分别从 HSI 相似性和边缘结构相似性两方面来确定一致性准则。

(1) 接缝线 HSI 的相似度 D_{HSI} 为:

$$D_{HSI} = \alpha \cdot \rho_H + \beta \cdot \rho_S + \gamma \cdot \rho_I \quad (6)$$

其中, $\alpha, \beta, \gamma \in (0, 1)$, 且 $\alpha + \beta + \gamma = 1$; ρ_H, ρ_S 和 ρ_I 分别为相邻接缝曲线 l_1 和 l_2 在 H, S 及 I 这 3 个通道的相关系数,该值的绝对值越大,相关性就越强,即接缝曲线越相似。

每个通道的相关系数 ρ 为:

$$\rho(l_1, l_2) = \frac{\sum_{i=1}^n (l_1(i) - \bar{l}_1)(l_2(i) - \bar{l}_2)}{\sqrt{\sum_{i=1}^n (l_1(i) - \bar{l}_1)^2 \sum_{i=1}^n (l_2(i) - \bar{l}_2)^2}} \quad (7)$$

其中, $l_1(i), l_2(i)$ 分别表示该通道已合成区域的接缝曲线 l_1 和待匹配块的接缝曲线 l_2 的第 i 个灰度值; n 是该通道已合成区域的接缝曲线 l_1 和待匹配块的接缝曲线 l_2 的像素数目; \bar{l}_1 和 \bar{l}_2 分别是该通道已合成区域的接缝曲线 l_1 和待匹配块的接缝曲线 l_2 的灰度均值。通常情况下,相关系数的取值范围在 0.8~1.0 时判断为相关性较强。

(2) 接缝线边缘结构的相似度。对纹理样图进行预处理,提取其边缘图像,假设边缘图像相邻接缝处的两条边缘曲线分别为 E_1 和 E_2 , 则边缘曲线相似度 D_E 为:

$$D_E(E_1, E_2) = \frac{1}{n} \sum_{i=1}^n (E_1(i) - E_2(i))^2 \quad (8)$$

其中, $E_1(i)$ 和 $E_2(i)$ 分别表示该通道已合成区域的接缝边缘曲线 E_1 和待匹配块的接缝边缘曲线 E_2 的第 i 个灰度值; n 是该通道已合成区域的接缝边缘曲线 E_1 和待匹配块的接缝边缘曲线 E_2 的像素数目。由于纹理样图的边缘图像是二值特征图,因此相邻接缝处的边缘曲线 E_1 和 E_2 的各个边缘点的取值为 0 或 1。

(3) 双接缝线一致性匹配准则。在本文的纹理合成过程中,新的匹配准则是由一条接缝曲线和一条辅助接缝曲线

共同作用,希望能够搜索到更符合人眼视觉特性的小于某阈值的待拼接子块,因此总的匹配准则函数为:

$$F = F_1 + F_2 \quad (9)$$

其中, F_1, F_2 分别表示已合成子块与待匹配块的真实接缝曲线和辅助接缝曲线的相似度,该相似度同时考虑色调、饱和度、亮度和边缘特征点等信息。每个相似度 F_i 可表示为:

$$F_i = \frac{D_{HSI}(l_1, l_2)}{1 + \mu \cdot D_E(E_1, E_2)} \quad (10)$$

其中, $i=1, 2$; D_{HSI} 和 D_E 的定义如式(6)、式(8)所示;常数 μ 反映边缘结构相似度点所占的权重,其值越小则边缘结构特征点所占比重就越大。

3 实验结果与分析

为了验证本文合成算法的有效性和快速性,本文对大量的纹理样图进行了合成实验,并将所提算法与近期有代表性的3种算法,即文献[11]、文献[13]和文献[14]中的算法^[14]进行了对比实验。其中,文献[14]中的算法通过 IPOL 网站在线测试得到纹理合成的实验结果;本文算法、文献[11]和文献[13]中的算法的纹理合成实验则是在 Windows 10 操作系统中进行的,基于 1.70 GHz Intel Core(TM) i5-4210U CPU, 4 GB 内存,采用 Matlab R2018a 编程实现。

3.1 主观评价

本文在 DTD(Describable Textures Dataset) 图像数据集上进行性能测试对比。在众多实验结果中,选取了 16 幅典型的结构纹理合成结果进行主观视觉效果的对比,结果如图 5、图 6 所示。本文算法、文献[11]和文献[13]中的算法均设定拼接块个数为 9×9 , 其中,样图 1—样图 13 的拼接块是边长为 50 像素的正方形块,而样图 14—样图 16 的拼接块是边长为 100 像素的正方形块。辅助接缝线与真实接缝线的距离一般与传统算法重叠区域的宽度相当,将样图 1—图样 13 的辅助接缝曲线与接缝曲线间距设定为 8 像素,将样图 14—样图 16 的辅助接缝曲线与接缝曲线间距设定为 17 像素。

由图 5、图 6 可以看出:文献[14]中的算法对样图 3、样图 8 和样图 16 的合成效果较好,但文献[11]、文献[13]和文献[14]中的算法在样图 1 和样图 6 的合成结果图像中都出现了结构模糊的现象;样图 2—样图 5,样图 7—样图 13 和样图 15 在文献[11]和文献[13]中的算法的合成结果图像中明显存在结构不连续和轮廓断裂的缺点,同样的缺点也存在于文献[14]中的算法对样图 4、样图 5、样图 9—样图 14 的合成结果图像中;文献[11]和文献[13]中的算法对样图 14 和样图 16 的合成结果图像及文献[14]中的算法对样图 2 和样图 7 的合成结果图像中都产生了样图里没有的纹理图案;文献[11]、文献[13]、文献[14]中的算法在样图 6 和样图 15 的合成结果图像中出现了纹理图案排列不规则的问题。而本文算法合成的纹理图像都较好地保持了图像中轮廓线条的完整和连续。其主要原因在于,对于结构性纹理图像的合成,接缝处边缘线条的对准最为重要,但传统的 3 种方法采用最佳缝合路径拼接时,只能保证整个接缝线上颜色差值最小,不能保证重要的边缘线条是否对准,因此合成结果会出现纹理结构不连贯的问题。从主观视觉效果上看,无论是对局部结构的保持,还是对结构纹理整体性的保持,本文算法都能够取得较为理想的合成效果。

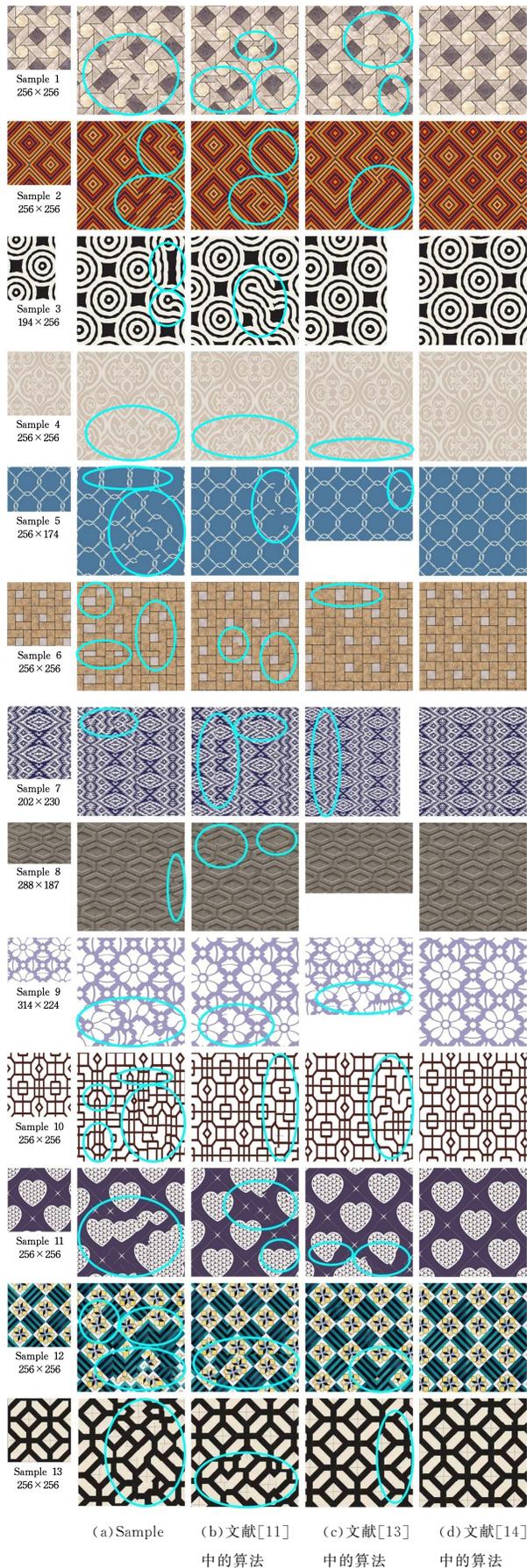


图 5 边长为 50 像素的拼接块合成结果对比

Fig. 5 Comparison of texture synthesis results using block with side length of 50 pixels

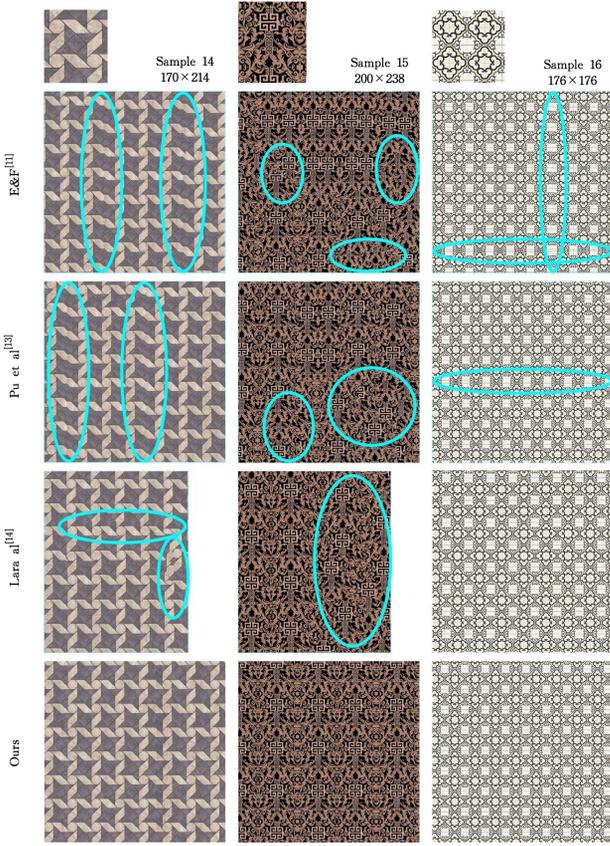


图6 拼接块边长为100像素的合成结果对比

Fig. 6 Comparison of texture synthesis results using block with side length of 100 pixels

3.2 客观评价

本文采用灰度直方图、灰度自相关函数、局部方向直方图和方向自相关函数等指标^[19],来客观衡量尺寸较大的合成纹理图像和较小的纹理样图的颜色及结构的一致性。

(1) 灰度自相关函数

为了更精确地评估输入和输出纹理灰度之间的相似性,引入二阶统计量自相关。

自相关函数用于考查某窗口内某一像素和与其偏离一定距离的像素之间的相似度^[20]。设灰度图像为 $f(i, j)$, 其自相关函数的定义为:

$$c(x, y) = \frac{\sum_{i=1}^M \sum_{j=1}^N f(i, j) f(i+x, j+y)}{\sum_{i=1}^M \sum_{j=1}^N f^2(i, j)} \quad (11)$$

其中, $i=1, 2, \dots, M; j=1, 2, \dots, N$ 。

(2) 局部方向直方图^[19]

结构张量 \mathbf{S} 概括了某像素点 P 在特定邻域内梯度的主要方向, 以及这些方向的相干程度, 通常定义为图像的一阶偏导数的局部协方差矩阵, 表达式为:

$$\mathbf{S}(P) = \mathbf{S} = \begin{bmatrix} \mathbf{W} * \mathbf{I}_x^2 & \mathbf{W} * \mathbf{I}_x \mathbf{I}_y \\ \mathbf{W} * \mathbf{I}_x \mathbf{I}_y & \mathbf{W} * \mathbf{I}_y^2 \end{bmatrix} \quad (12)$$

其中, \mathbf{W} 为高斯核; \mathbf{I}_x 和 \mathbf{I}_y 分别为图像在 x 轴和 y 轴方向的梯度。

在像素点 P 处对结构张量 $\mathbf{S}(P)$ 进行特征值分解的结果是两个表示局部图像边缘强度的非负特征值 $\lambda_1(P)$ 和 $\lambda_2(P)$,

以及两个正交和平行于局部图像边缘的特征向量 $\mathbf{e}_1(P) = [\mathbf{e}_{1x}(P), \mathbf{e}_{1y}(P)]$ 和 $\mathbf{e}_2(P) = [\mathbf{e}_{2x}(P), \mathbf{e}_{2y}(P)]$ 。图像中像素点 P 的局部方向变化 $O(P)$ 可以通过特征值中较大的值所对应的特征向量计算取得, 取值范围为 $[-90^\circ, 90^\circ]$, 定义如下:

$$O(P) = \tan^{-1} \left(\frac{\mathbf{e}_{1y}(P)}{\mathbf{e}_{1x}(P)} \right) \quad (13)$$

局部方向直方图类似于灰度直方图的定义, 反映了局部方向变化时某个角度出现的频率。

(3) 方向自相关函数^[19]

方向自相关函数与灰度自相关函数的定义类似, 由于方向自相关不是直接的, 因此引入复数表达式。

设像素点 P 的复局部方向变化 $\mathbf{V}(P)$ 为:

$$\mathbf{V}(P) = C(P) e^{j2O(P)} \quad (14)$$

其中, $C(P)$ 是相干因子, 是局部各向异性的一种度量, 能表征梯度方向的分散性, 定义为:

$$C(P) = \frac{\lambda_1(P) - \lambda_2(P)}{\lambda_1(P) + \lambda_2(P)} \quad (15)$$

其中, $\lambda_1(P)$ 和 $\lambda_2(P)$ 为结构张量 $\mathbf{S}(P)$ 在像素点 P 处的特征值。

定义局部方向自相关函数 $\mathbf{A}_v(\omega)$ 为:

$$\mathbf{A}_v(\omega) = \|\langle \mathbf{V}(P) \mathbf{V}^*(P + \omega) \rangle\| \quad (16)$$

其中, ω 表示偏移的像素数; $\mathbf{V}^*(P)$ 是在像素点 P 处的复局部方向 $\mathbf{V}(P)$ 的共轭复数; $\langle \cdot \rangle$ 表示在不超出图像尺寸的情况下取平均; $\|c\|$ 表示复数 c 的模。

为比较纹理样图和合成结果图的灰度直方图及局部方向直方图, 选用 Kullback-Leibler(K-L) 散度^[21] 来衡量两个直方图的差异程度, 其表达式为:

$$D_{KL}(P \| Q) = \sum_{i=1}^N P(x_i) \log \frac{P(x_i)}{Q(x_i)} \quad (17)$$

其中, P, Q 分别为输入的纹理样图、输出的合成结果图像的灰度直方图或局部方向直方图; x_i 表示图像某种灰度级或局部方向的某一角度; N 是灰度级或角度总个数。

由式(17)可知, K-L 散度越小, 输入的纹理样图、输出的合成结果图像的直方图分布越接近。前面 16 组纹理样图的灰度直方图与局部方向直方图的 K-L 散度对比结果如表 1 所列。表 1 中, 灰度直方图 K-L 散度值反映了合成纹理图像与样图纹理图像在颜色上的差异程度, 而局部方向直方图 K-L 散度值反映了合成纹理图像与样图纹理图像在结构纹理上的差异程度。在颜色一致性方面, 由表 1 可知, 本文算法合成的结果全部优于文献[11]中的算法; 在 16 幅纹理样图中, 上文有 15 幅图像在颜色一致性上优于文献[13]和文献[14]中的算法, 只有样图 8 的结果略低于后者, 其主要原因是样图 8 的颜色比较单一, 接近灰度图像。在结构纹理一致性方面, 本文算法的结果全部优于文献[11]、文献[13]和文献[14]中的算法, 这完全符合合成图像视觉评价的结果。此外, 本文还利用灰度自相关函数和方向自相关函数的欧氏距离, 来衡量合成纹理图像与样图纹理图像在颜色和结构纹理上的一致性。自相关函数的欧氏距离越小, 颜色或结构纹理的一致性就越高, 具体对比结果如表 2 所列。

表1 4种算法的K-L散度对比结果

Table 1 Comparison results of four algorithms in K-L divergence

Texture Sample	K-L Divergence of Gray Level Histogram				K-L Divergence of Local Orientation Histogram			
	文献[11]中的算法	文献[13]中的算法	文献[14]中的算法	Ours	文献[11]中的算法	文献[13]中的算法	文献[14]中的算法	Ours
1	0.0113	0.0102	0.0093	0.0077	0.0092	0.0054	0.0058	0.0053
2	0.0094	0.0083	0.0086	0.0082	0.0133	0.0291	0.0283	0.0131
3	0.0035	0.0030	0.0028	0.0028	0.0290	0.0192	0.0112	0.0112
4	0.0432	0.0400	0.0421	0.0315	0.0128	0.0085	0.0084	0.0082
5	0.0149	0.0130	0.0135	0.0128	0.0154	0.0146	0.0139	0.0133
6	0.0408	0.0386	0.0385	0.0380	0.0095	0.0060	0.0058	0.0052
7	0.0103	0.0081	0.0078	0.0076	0.0093	0.0052	0.0069	0.0049
8	0.0140	0.0118	0.0123	0.0123	0.0091	0.0068	0.0064	0.0064
9	0.0103	0.0086	0.0073	0.0055	0.0097	0.0059	0.0063	0.0056
10	0.0048	0.0048	0.0051	0.0032	0.0211	0.0154	0.0182	0.0145
11	0.0206	0.0200	0.0183	0.0152	0.0101	0.0081	0.0099	0.0078
12	0.0130	0.0120	0.0106	0.0097	0.0189	0.0118	0.0082	0.0059
13	0.0217	0.0206	0.0192	0.0179	0.0929	0.0860	0.0890	0.0305
14	0.0231	0.0222	0.0215	0.0180	0.0178	0.0150	0.0163	0.0136
15	0.0129	0.0105	0.0112	0.0101	0.0103	0.0086	0.0084	0.0080
16	0.0169	0.0130	0.0127	0.0127	0.0099	0.0076	0.0072	0.0072

表2 4种算法的欧氏距离对比结果

Table 2 Comparison results of four algorithms in Euclidean distance

Texture Sample	Euclidean Distance of Gray Level Autocorrelation Coefficients				Euclidean Distance of Orientation Autocorrelation Coefficients			
	文献[11]中的算法	文献[13]中的算法	文献[14]中的算法	Ours	文献[11]中的算法	文献[13]中的算法	文献[14]中的算法	Ours
1	0.6020	0.4762	0.0507	0.5201	0.2427	0.1868	0.1873	0.1626
2	1.3551	1.2768	1.2363	1.1162	0.9246	0.5010	0.4893	0.3448
3	1.8815	1.7393	1.6488	1.6488	1.4957	1.1867	1.1486	1.1486
4	1.1681	1.1658	1.1646	1.1632	0.9467	0.9836	0.9317	0.8581
5	2.0274	2.0154	1.9462	1.8772	0.4010	0.2304	0.2284	0.2256
6	1.3296	1.2339	1.2293	1.2482	0.5186	0.2824	0.2535	0.2449
7	1.4473	1.3805	1.4216	1.3754	0.5228	0.3915	0.2104	0.1011
8	1.5470	1.4515	1.4929	1.4929	0.4288	0.4289	0.2589	0.2589
9	1.2597	1.0730	1.0783	1.0025	0.4355	0.3472	0.3721	0.3380
10	0.9031	0.8918	0.8952	0.8761	0.8375	0.8894	0.8214	0.5634
11	0.5768	0.4988	0.4853	0.4425	0.5693	0.4646	0.4185	0.3208
12	1.0382	1.0221	0.8779	0.8583	0.6854	0.6091	0.3866	0.2852
13	1.4108	1.3750	1.2574	0.8717	0.2120	0.2017	0.0993	0.0789
14	2.7372	2.9540	2.8367	2.7144	0.4523	0.4552	0.4632	0.4133
15	2.0445	1.9480	1.7845	1.9726	0.2833	0.1970	0.1852	0.1677
16	3.6055	3.4991	3.5298	3.5298	0.2934	0.2103	0.2227	0.2227

由表2可知,在颜色一致性方面,利用灰度自相关函数的欧氏距离进行度量,本文算法的结果全部优于文献[11]中的算法;但与文献[13]和文献[14]中的算法相比结果各有优劣,主要原因是样图的颜色比较单一或一致,体现不出本文采用HSI度量准则的优越性。但在结构纹理一致性方面,利用方向自相关函数的欧氏距离进行度量,本文算法的所有结果都优于其他3种算法的结果。

总之,表1、表2的对比结果总体上证实了3.1节中的主观评价结果,即本文算法合成的纹理图像在结构纹理一致性上优于其他3种对比算法,显著改善了结构纹理图像的合成质量,从而证明了其在合成结构纹理时的优越性。

3.3 合成时间对比

由于本文算法、文献[11]和文献[13]中的算法合成的纹理结果图像大小都相同,但它们与文献[14]中的算法合成的纹理结果图像大小不一致,因此选择比较本文算法、文献[11]和文献[13]中的算法的合成时间。对图5、图6中的样图进行实验,合成时间取对一个纹理样图进行50次合成实验时间的平均值,以避免单次合成时间中由其他因素引起的误差。3种算法的合成时间对比结果如表3所列。

表3 3种算法合成时间的对比

Table 3 Synthesis time comparison of three algorithms

Texture Sample	Sample Size	Synthesis Size	文献[11]中的算法/s	文献[13]中的算法/s	Ours /s
1	256×256	386×386	5.30	4.26	3.61
2	256×256	386×386	5.28	4.50	3.53
3	194×256	386×386	4.18	3.14	2.62
4	256×256	386×386	5.34	4.07	3.68
5	256×174	386×386	4.02	3.33	2.83
6	256×256	386×386	5.18	4.19	3.66
7	202×230	386×386	4.12	3.25	2.48
8	288×187	386×386	5.11	4.19	3.30
9	314×224	386×386	6.25	4.89	3.99
10	256×256	386×386	5.31	4.47	3.45
11	256×256	386×386	5.21	4.22	3.60
12	256×256	386×386	5.07	4.13	3.68
13	256×256	386×386	5.36	4.44	3.66
14	170×214	764×764	9.80	7.54	2.59
15	200×238	764×764	11.31	8.79	3.49
16	176×176	764×764	9.19	7.11	2.43

由表3可知,在不同的样图尺寸和合成图像尺寸下,本文算法的合成时间都短于文献[11]和文献[13]中的算法。其中,对于样图1—样图13的合成结果,本文算法比文献[11]

中的算法平均快 1.50 倍,比文献[13]中的算法平均快 1.21 倍;对于样图 14—样图 16,本文算法比文献[11]中的算法平均快 3.60 倍,比文献[13]中的算法平均快 2.79 倍。这说明选择的拼接块尺寸越大,本文算法参与运算的数据量越小于文献[11]和文献[13]中的算法的数据量,越能体现本文算法的快速性,同时也验证了本文算法的双接缝一致性准则和非重叠拼接方式的有效性。

结束语 本文提出了一种基于双接缝一致性准则的非重叠拼接纹理图像快速合成算法,以接缝曲线的色调、饱和度、亮度和边缘特征点等信息共同形成接缝线一致性判别准则,从而快速/准确地实现拼接块的搜索和非重叠拼接。本文算法解决了基于块的纹理合成算法在合成结构性较强的纹理图像时容易出现结构不连续、边界扭曲和接缝走样等问题,得到了与纹理样图具有相同视觉效果的大视场合成纹理图像,满足人们对视觉真实感的要求,同时提高了合成算法的实时性。未来将进一步考虑扩大该算法的适用范围,使其应用于多种类型纹理图案的合成;同时加入 GPU 并行加速,进一步提高合成速度。

参 考 文 献

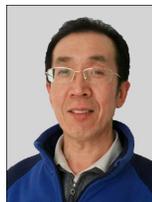
- [1] LEE J, KIM D, KIM Y, et al. A Training Method for Image Compression Networks to Improve Perceptual Quality of Reconstructions[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop(CVPRW). IEEE, 2020:585-589.
- [2] YAO S, YAN J, WUM, et al. Texture synthesis based thyroid nodule detection from medical ultrasound images: interpreting and suppressing the adversarial effect of in-place manual annotation[J]. *Frontiers in Bioengineering and Biotechnology*, 2020, 8(3):599-610.
- [3] GUTIERREZ J, RABIN J, GALERNE B, et al. Ondemand solid texture synthesis using deep 3D networks[J]. *Computer Graphics Forum*, 2019, 39(4):511-530.
- [4] JIA R, WANG F, YUAN H W, et al. Research on Fuzzy Adaptive Fusion Algorithm for Ultraviolet Polarization Image of Latent Fingerprint[J]. *Computer Engineering*, 2020, 46(11):267-272, 278.
- [5] CHEN Q, LI G, XIE L, et al. Structure guided image completion using texture synthesis and region segmentation[J]. *Optik*, 2019, 185(7):896-909.
- [6] RAAD L, DESOLNEUX A, MOREL J M. A Conditional Multi-scale Locally Gaussian Texture Synthesis Algorithm[J]. *Journal of Mathematical Imaging & Vision*, 2016, 56(2):260-279.
- [7] PORTILLA J, SIMONCELLI E P. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients[J]. *International Journal of Computer Vision*, 2000, 40(7):49-70.
- [8] HEEGER D J, BERGEN J R. Pyramid-based texture analysis/synthesis[C]// Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. ACM, 1995:229-238.
- [9] GALERNE J M. Random Phase Textures: Theory and Synthesis[J]. *IEEE Transactions on Image Processing A Publication of*

the IEEE Signal Processing Society, 2011, 20(1):257-267.

- [10] BARNES C, ZHANG F L. A survey of the state-of-the-art in patch-based synthesis[J]. *Computational Visual Media*, 2017, 4(1):3-20.
- [11] EFROS A A, FREEMAN W T. Image quilting for texture synthesis and transfer[C]// The 28th International Conference on Computer Graphics and Interactive Techniques Conference(SIGGRAPH'01). ACM, 2001:341-346.
- [12] ZHU R C, QIAN W H, PU Y Y, et al. Texture Synthesis Based On Self-similarity Matching[J]. *Computer Science*, 2018, 45(S1):228-232.
- [13] PU Y Y, XU D, QIAN W H, et al. An improved texture synthesis algorithm[J]. *Transactions on Edutainment XI, Lecture Notes in Computer Science*, 2015, 8971(1):103-113.
- [14] RAAD L, GALERNE B. Efros and Freeman Image Quilting Algorithm for Texture Synthesis[J]. *Image Processing on Line*, 2017, 7(6):1-22.
- [15] SHEN J W, WANG S Z. Structural matching in texture synthesis based on image quilting[J]. *Journal of Shanghai University (Natural science)*, 2010, 16(1):10-14.
- [16] TANG Y, LIN Q F, XIAO T Z, et al. GPU-based texture synthesis with preserved structures[J]. *Computer Science*, 2016, 43(4):299-302, 312.
- [17] RAFI M, MUKHOPADHYAY S. Image Quilting for Texture Synthesis of Grayscale Images Using Gray-Level Co-occurrence Matrix and Restricted Cross-Correlation[J]. *Progress in Advanced Computing and Intelligent Engineering, Advances in Intelligent Systems and Computing*, 2019, 713(8):37-47.
- [18] CHEN B, ZHANG X, WANG R T, et al. Detect concrete cracks based on OTSU algorithm with differential image[J]. *The Journal of Engineering*, 2019, 23:9088-9091.
- [19] AKL A, YAACOUB C, DONIAS M, et al. A survey of exemplar-based texture synthesis methods[J]. *Computer Vision and Image Understanding*, 2018, 172(5):12-24.
- [20] HU Z J, LIU G H, SU Y. Application of Local Autocorrelation Function in Content-based Image Retrieval[J]. *Computer Science*, 2018, 45(S2):259-262.
- [21] KULLBACK S, LEIBLER R A. On Information and Sufficiency[J]. *Annals of Mathematical Statistics*, 1951, 22(1):79-86.



JIN Li-zhen, born in 1998, postgraduate. Her main research interests include image processing and so on.



LI Qing-zhong, born in 1963, Ph.D, professor, postgraduate supervisor. His main research interests include image processing, signal processing and pattern recognition.