

移动设备上基于 Wang tiles 的 双向纹理合成算法

何高奇^{1),2)} 潘志庚¹⁾

¹⁾(浙江大学 CAD&CG 国家重点实验室,杭州 310058) ²⁾(华东理工大学计算机科学与工程系,上海 200237)

摘要 为了在移动设备上实现纹理合成,首先在总结了 PC 平台上纹理合成技术特点的基础上,指出基于 Wang tiles 的纹理合成适合在资源受限的移动平台上实现;然后通过深入分析 Cohen 随机贴片算法中的不足,提出了改进的基于 Wang tiles 的双向纹理合成新算法,即包括样本纹理子图的选取、片填充和基于双向扫描的贴片算法。该算法不仅扩大了最佳切割线的搜索范围和增加了贴片的随机性,同时也减少了计算和存储需求。实验结果表明,该算法不仅能够运行在移动设备上,而且与随机贴片算法相比,还提高了非周期性视觉效果。

关键词 Wang tiles 纹理合成 随机贴片 最佳切割线 双向

中图法分类号: TP391.41 文献标识码:A 文章编号:1006-8961(2009)02-0370-07

Wang Tiles Based Bi-direction Texture Synthesis Algorithm on Mobile Devices

HE Gao-qi^{1),2)}, PAN Zhi-geng¹⁾

¹⁾(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058)

²⁾(Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237)

Abstract Characteristics of current texture synthesis techniques on PC platform are summarized. Texture synthesis based on Wang tiles is suitable to be implemented on resource-constraint mobile devices. An improved bi-directional texture synthesis algorithm based on Wang tiles is presented through analyzing the stochastic tiling algorithm suggested by Cohen. The proposed algorithm mainly consists of the following steps: selecting sample sub-images, filling the tile with texture and tiling through bi-direction scanning. Accordingly, the search region for the best cut path is enlarged, randomness of the selected tiles is increased and the requirements of computation and storage are also reduced. Experiments validate the usability of our algorithm on mobile devices. Compared with stochastic tiling, the proposed algorithm illustrates better visual effects in aperiodicity.

Keyword Wang tiles, texture synthesis, stochastic tiling, the best cutting path, bi-direction

1 引言

随着以 PDA、PocketPC 和移动电话为代表的移动设备日益普及和移动设备配置、性能的改善,以及

无线网络的成熟,移动设备图形技术作为一个崭新的方向逐渐发展起来^[1]。与 PC 机相比,由于移动设备的资源(如 CPU 速度、内存空间、电池供电能力和屏幕空间大小等)非常有限,所以移动设备图形研究在承袭 PC 图形现有的研究方法和成果的同

基金项目:国家自然科学基金重点项目(60533080);教育部博士点基金项目(J20060186)

收稿日期:2007-07-25;改回日期:2007-09-14

第一作者简介:何高奇(1974 ~),男。华东理工大学计算机科学与工程系讲师,2007 年于浙江大学 CAD&CG 国家重点实验室获博士学位。主要从事计算机图形学、虚拟现实/增强现实等研究。E-mail:hegaoqi@ecust.edu.cn

时,更需要针对特定的平台做出适当的调整和革新的设计。

纹理合成技术在不增加物体表面几何复杂度的情况下,不仅可以有效模拟和增强景物表面的细节特征,而且可增加绘制模型的真实感。PC 机中纹理合成技术已经相当成熟^[2-3],但是目前国内外对移动设备上的纹理合成技术的研究还不多。

本文主要研究移动设备上的基于多个样图的纹理合成。在对 PC 平台上纹理合成技术的发展轨迹进行总结后,本文主要关注基于 Wang tiles 的纹理合成技术。通过深入分析 Cohen 等人提出的随机贴片算法,本文提出了一种改进的双向纹理合成算法,主要从运行时间和视觉效果上进行改进。另外,算法设计时还充分考虑到移动设备计算和存储能力的限制。

2 相关工作

纹理合成有过程纹理合成和样图纹理合成这两个主要的分支。其中,过程纹理合成能够避免纹理映射带来的失真^[4],但对每一种新的纹理都需要进行参数调整和反复测试;而基于样图(TSfS)的纹理合成技术则是基于给定的小区域纹理样本来生成视觉上相似且连续的纹理。TSfS 技术既可以克服传统纹理映射方法的缺点,又避免了过程纹理合成中繁琐地调整参数,因而受到更多的关注。

TSfS 纹理合成技术根据所采用的统计模型,可以分为基于参数统计模型^[5]和基于采样的合成方法两类。其中,前者适合于对纹理图像进行分析,但适用范围没有后者广;基于采样的方法根据采样所取局部区域的大小,又可以分为基于像素的^[6]和基于块的纹理合成。其中,基于像素的方法一般能产生高质量的绘制效果,但搜索匹配像素的代价相当高,所以一般运行速度都比较慢。

基于块的采样方法是以一组像素组成的块为单位进行合成,大致可分为基于样图的 patch 和基于多个样图两类,其主要优点是合成速度快、使用范围广。基于样图的 patch 合成方法^[7]是以块为单位,每次从样图中选取一个单位的纹理添加到合成图像中。一种更加简单的操作是,先将每次添加的纹理块组成一个有限的样图系列,再利用该样图系列按照一定的规则进行纹理合成,这就形成了基于多个样图的纹理合成方法。代表性的方法是在 2003 年

和 2006 年的 SIGGRAPH 会议上,Wang tiles 被利用来进行纹理合成^[8]和非照片真实感绘制^[9]。

Wang tiles 是一组每条边都有着颜色的正方形,在 20 世纪 60 年代初提出后逐渐应用到纹理合成领域^[10]。Cohen 等人提出了一种随机算法^[8],即以一系列 Wang tiles 来非周期性的贴片平面,其既能保持重复利用 tiles 来创建大场景的优点,又可避免明显的周期性的视觉瑕疵。Wang tiles 纹理合成算法在继承了基于块采样方法优点(合成速度快,使用范围广)的同时,一个突出的特点就是对内存的需求比传统的基于像素或 patch 的纹理合成方法要小得多,这比较适合于在资源受限的移动平台上实现,并能提供较强的真实感。

3 基于 Wang tiles 随机贴片算法

Cohen 等人提出的随机填充算法是相当直观的,其主要表现在以下几方面:

(1)首先确定样本子图像集合,即从源图像中取出若干个正方形样本子图像(记为集合 P),每个子图像对应一个片的水平边或垂直边的颜色;

(2)然后构造每个片,即先根据每个片的 4 条边的颜色,从 P 中取出对应的样本子图像,并旋转 45° 成为菱形,再通过寻找 4 条最佳切割线来组合这 4 个菱形样本,以形成最终的片;

(3)最后是纹理合成阶段,即利用上步生成的片的集合,按随机贴片算法去贴一个平面。

在该文献中,虽然提出了用于纹理合成的自动片生成技术,但对如何从源纹理图像中选取样本子图像并没有给出清晰的说明,这些子图像不同的选取方案必将产生不同的片组合。

使用该算法,合成纹理的视觉瑕疵仍然存在,问题根源主要有以下两个^[11]:(1)只使用了源图像中有限的样本子图像;(2)由文献[7]中的缝合算法引入的瑕疵。

对第 1 个问题,虽然可以通过增加 Wang tiles 集合的片数来解决,但相应的计算和存储需求都显著增加。

对第 2 个问题,主要原因是在片构造算法中,不能保证用来构成每个片的 4 个样本子图像是最优选择,以及不能在组合生成片时找到 4 条最佳切割线。如果缝合误差比较大,则会直接影响到最终合成图像的质量。对此,Cohen 等人提出用多组样本图像

进行迭代试验,通过计算切割线附近的像素的颜色误差总和,同时用使得该误差最小的那一组来合成片。这种迭代虽有助于提高合成结果的质量,但这样搜索的计算量比较大,而且如何预设误差阈值和在有限的迭代步骤内终止计算等方面也没有明确的评价指标。

4 双向纹理合成算法

通过对 Cohen 纹理合成算法的深入分析,本文提出了一种改进的双向纹理合成算法。

4.1 算法流程

图 1 表示基于 Wang tiles 的双向纹理合成算法的完整流程图。整个过程分为 7 步,图 1 中每个模块左上角的数字表明了算法的执行顺序,平行显示的模块没有严格地执行先后顺序,例如模块 1 和 2。

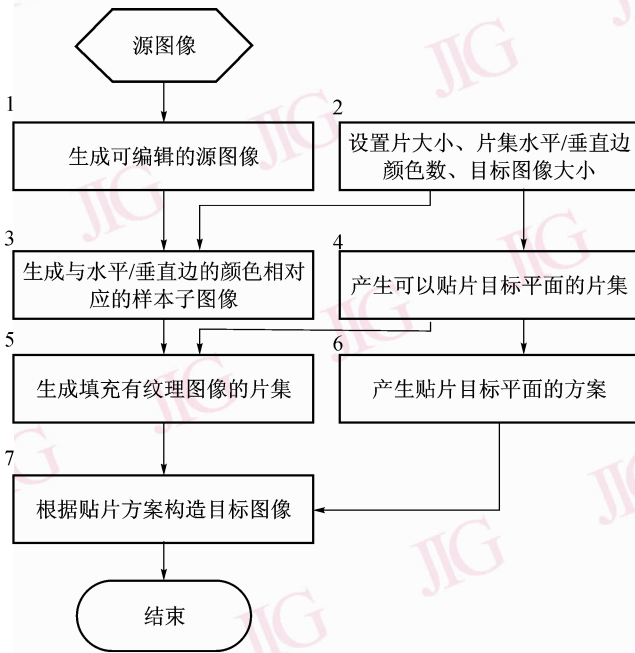


图 1 双向纹理合成算法

Fig. 1 Bi-direction texture synthesis algorithm

双向纹理合成算法步骤如下:

(1) 根据输入的源图像文件,生成易于处理的源图像 I_{source} , 并获取源图像文件的长 $W_{source-image}$ 、高 $H_{source-image}$ 、数据位等信息,以利于后续图像处理。

(2) 设置片的大小 S_{tile} 、片集的水平方向的颜色数 N_{color}^H 和垂直方向颜色数 N_{color}^V 、目标图像的长 $W_{target-image}$ 和高 $H_{target-image}$ 。这个可以通过交互界面由

用户指定。

(3) 根据片集的水平方向和垂直方向的颜色数,从源图像 I_{source} 中生成相应的样本子图像集合 $G_{sample-img}^H$ 和 $G_{sample-img}^V$ 。

(4) 根据片集的水平方向和垂直方向的颜色数来产生 Wang tiles 集合 $G_{Wangtiles}$ 。 $G_{Wangtiles}$ 中包含有 $2 \times N_{color}^H \times N_{color}^V$ 个片,这些片可以贴片整个目标平面。

(5) 对集合 $G_{Wangtiles}$ 中的每个片,根据其水平方向(W和E)和垂直方向(N和S)的颜色,分别从集合 $G_{sample-img}^H$ 和 $G_{sample-img}^V$ 中取出对应的样本子图像,并组合生成填充有纹理的片(称为纹理片,实际表达为一幅 $S_{tile} \times S_{tile}$ 大小的图像)。本步骤的输出结果是一个纹理片集合 $G_{texture-imgs}$ 。这里的算法类似于 Cohen 等人提出的自动片生成算法,不仅减少了计算量,还扩大了寻找最佳切割线的区域范围。

(6) 根据集合 $G_{Wangtiles}$ 中的片,生成一个贴片目标平面的具体方案 tiling(贴片),其在目标平面中可表示为由 (i, j) 表示的区域,可使用集合 $G_{Wangtiles}$ 中的片 k 来填充。本步骤类似于 Cohen 等人提出的随机填充算法,但从计算效率和随机性上进行了改进。

(7) 对于目标平面中由 (i, j) 表示的区域,可先根据 tiling 取对应的片 k ,然后从 $G_{texture-imgs}$ 中取得与片 k 对应的图像,最后将像素拷贝到目标图像的该区域。

4.2 样本子图像选取

根据交互指定的水平边颜色数目和垂直边颜色数目(步骤 2),纹理合成过程需要从源图像中选取 $N_{color}^H + N_{color}^V$ (记为 N_{sample}) 个样本子图像。样本子图像的长和宽都是 S_{tile} ,本文取它的中心位置 $p_c = (S_{tile}/2, S_{tile}/2)$ 代表该子图像, p_c 的位置限定在点 p_{TL} (下角 TL 代表 top-left) $= (S_{tile}/2, S_{tile}/2)$ 和点 p_{LR} (下角 LR 代表 low-right) $= (W_{source-image} - S_{tile}/2, H_{source-image} - S_{tile}/2)$ 所限定的矩形范围内(记为 CheckBox)。现在就是如何选取样本子图像,并将其转化为在 CheckBox 中确定 N_{sample} 个点 p_i ,同时使得这些点 p_i 具有良好的分布特性的问题。

点采样是计算机图形学中的一个基本而重要的研究内容,现已出现了很多新技术,如 jittered、泊松圆盘和 N-rook 等技术,本文采用一种基于 Hammersley 点集^[12]的 2 维平面采样技术。

若记正整数基为 t (大于 1),则任何一个非负整数 k 可以表示为

$$k = a_0 + a_1 t + a_2 t^2 + \dots + a_r t^r \quad (1)$$

其中, $a_i \in [0, t-1]$, $i=0, \dots, r$ 。

定义 k 的函数 Φ_i 为

$$\Phi_i(k) = \frac{a_0}{t} + \frac{a_1}{t^2} + \dots + \frac{a_r}{t^{r+1}} \quad (2)$$

则 2 维平面基为 2 的 Hammersley 点集定义为

$$\left(\frac{k}{n}, \Phi_2(k) \right) \quad k=0, 1, 2, \dots, n-1 \quad (3)$$

其中, n 是需要的点的总数。

2 维 Hammersley 点集在单位正方形内具有良好的均匀分布特性、随机采样的外观和低差异性, 图 2 显示了在 100×100 大小的区域内的 100 个 Hammersley 点的分布。

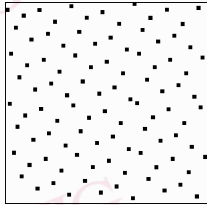


图 2 Hammersley 点集; 100×100 大小的区域内的 100 个点
Fig. 2 Hammersley points set, 100 points in 100×100 region

样本子图像的选取步骤如下:

(1) 在单位正方形内生成包含 N_{sample} 个元素的 Hammersley 点集 H ;

(2) 从单位正方形到 CheckBox, 将 H 中的 Hammersley 点 p 映射到源图像中的对应位置 q ;

(3) 以 q 为中心获取的样本子图像, 其用在 I_{source} 中的正方形区域表示。

4.3 改进的纹理填充算法

在双向纹理合成算法步骤 5 中, 对片集 $G_{\text{Wangtiles}}$ 中的每一个片 t , 根据其 4 条边的颜色从样本子图像集中取得 4 个子图像, 组合生成填充有纹理的图像。Cohen 等人提出的自动片填充算法如图 3 所示, 首先, 4 个样本子图像分别旋转 45° 成为菱形图像(标记为 W 、 E 、 S 和 N), 其相邻的两个片之间有搭边区域; 然后通过寻找搭边区域的最佳切割线, 并沿着各个子图像的对角线进行裁剪, 即可获得最终的填充了纹理的片。

这个算法有以下两点明显的不足:

(1) 旋转后子图对像素的访问不直接。如 $O = (i, j)$ 表示旋转后子图像 E 和 S 搭边区域内的某个

像素点, 点 O 在旋转后的子图像 E 和 S 中分别对应不同的坐标 $p = (i_1, j_1)$ 和 $q = (i_2, j_2)$ 。为获取像素 p 、 q 的颜色值需要分别执行一个旋转变换, 可通过旋转前的正方形子图像对应位置的像素取得。

(2) 最佳切割线的搜索范围有限。如图 3 所示, 由于最佳切割线仅在搭边区域内寻找, 搭边区域一般只是长为 S_{tile} 、宽为 $S_{\text{tile}}/4$ 的矩形范围, 而且由于片的大小 S_{tile} 本身就很有限, 所以在有限范围内搜索的最佳切割线的精度就很有限了。

图 3 4 个样本子图像组合生成一个片

Fig. 3 One tile generate by four sample sub-image

本文介绍一种改进的 Wang tiles 纹理填充算法。该新算法对要填充纹理的片 t , 首先沿着各边的中点将它等分为 4 个部分(如图 4(a)所示); 然后对每个部分, 选取带有颜色的边所对应的两个样本子图像的相邻区域 R_1 和 R_2 , 通过寻找 R_1 和 R_2 的最佳切割线来填充该部分。

(1) 确定要填充的片(图 4(a))和用于填充的样本子图像(图 4(b)~图 4(e))。要填充的片的 4 个部分分别标记为 NW 、 NE 、 SE 、 SW , 其中 N 、 E 、 S 、 W 代表片的 4 条边, 分别对应于图 4(b)~图 4(e)所示的样本子图。

(2) 填充 NW 部分 NW 。片的 NW 部分 NW 的边 N 和边 W 带有颜色, 选取它们所对应的样本子图像(图 4(b)和图 4(e)), 图 4(b)左下角的正方形区域(记为 R_1)和图 4(e)右上角正方形区域(记为 R_2)位置相邻(如图 5(a)所示)。根据 Dijkstra 算法求 R_1 和 R_2 的最佳切割线, 如图 5(a)中的白色曲线所示。根据求得的切割线, 将 R_1 中标有 N 的区域和 R_2 中标有 W 的区域的像素拷贝到目标片的左上



图 4 待填充的片和用于填充的样本子图像
Fig. 4 Tile and four sample sub-image for filling

角(NW 部分)。

(3) 分别填充片的 NE、SW、SE 部分 **NE**、**SW**、**SE**(如图 5(b)~图 5(d)所示)。

(4) 组合生成的填充有纹理的片(如图 6 所示)。标记有 **N**、**W**、**S**、**E** 的部分表示该区域的像素是从该片中的与边 **N**、**W**、**S**、**E** 对应的样本子图像中



图 5 片纹理填充算法
Fig. 5 Algorithm of filling texture in tile



图 6 组合生成的填充有纹理的片
Fig. 6 The generated tile filled with texture

取得的像素值。

本文提出的片纹理填充算法改进了 Cohen 等人提出的自动片生成算法的以下两点不足:

(1) 直接访问像素。由于正方形区域 R_1 和 R_2 大小相等,其中的任一个像素点相对于各自区域的左上角的偏移量相同,所以只要根据当前位置进行加法运算(偏移量),就可以在源图像 I_{source} 中获得像素值,这样计算量就显著减少。

(2) 增大最佳切割线的搜索范围。切割线是在区域 R_1 和 R_2 这样 $S_{tile} \times S_{tile}$ 大小的正方形区域中寻找,其比 Cohen 算法中的搭边区域的范围要大将近 4 倍。这种较大范围内的搜索有利于提高片纹理填充的质量。

4.4 双向贴片算法

Cohen 等人提出了一个随机贴片算法,从左上角开始,按照扫描线顺序向右向下进行填充,且每次添加片的过程是一个独立的随机过程。本文在此机



图 7 双向贴片算法
Fig. 7 Bi-direction tilling algorithm

制的基础上,提出了一个双向贴片算法,进一步改善了合成结果的非周期性。

双向贴片算法的基本原理是:分别从目标平面的左上角和右下角开始,沿着向右向下和向左向上的方向执行随机贴片算法(如图7(a)所示)。这两种填充机制分别标记为 F_{LT} (下角 LT 代表 left-top,下同)和 F_{RL} (下角 RL 代表 right-low,下同)。

执行过程中需要处理以下3方面的问题:

(1)从右下角执行 F_{RL} 随机贴片算法时新片的选择机制。每行开始时,最右边的片的边 S 与下面的片的边 N 的颜色必须相同;确定每行中其他位置的片时,新增加的片的边 E 和边 S 分别与右边片的边 W 和下面片的边 N 的颜色匹配。

(2)如果目标平面的高度是片大小的偶数倍(记为 n),则用 F_{LT} 选择新的片来填充第 $n/2$ 行,第 $n/2$ 行右边一半的片要参考该片上、下、左边的片的颜色来确定。当用 F_{RL} 选择新的片填充第 $n/2 + 1$ 行,则第 $n/2 + 1$ 行左边一半的片要参考该片上、下、右边的片的颜色来确定。如图7(a)中,在选择(3,4)位置的片时,需要参考已填充的上面片 a 的边 S 、左边片 b 的边 E 和下面片 h 的边 N 的颜色,才能确定此位置应该填充片 b 。

(3)如果目标平面的高度是片大小的奇数倍(记为 m),在第 $m/2 + 1$ 行,可用 F_{LT} 选择新的片填充左边一半,而用 F_{RL} 选择新的片来填充右边一半。将要填充的片需要根据已填充的相邻片的颜色来确定,其中边界位置的片要参考上下2个片的两种颜色,非边界位置的片要参考上、下、左(F_{LT} 时)或右(F_{RL} 时)边的3个片的3种颜色。

图7(b)为双向贴片算法的贴片结果。由于Cohen算法只用 F_{LT} 机制填充,使得目标平面中心位置右下方的片的选择需要参考左上部分已填充的片的颜色信息,并一直往右往下延伸,所以右下方的片对左上方的片是有依赖关系的。但在双向贴片算法中,目标平面左上和右下部分分别采用 F_{LT} 和 F_{RL} 机制,这是两个相互独立的随机过程。这样既保证了每个片的选择是随机的,又保证了两个部分的片没有依赖关系,这样就增加了拼贴效果的非周期性。

5 实验结果

双向纹理合成算法针对移动设备,从存储空间、处理器和电源消耗等方面做了针对性的设计。

为减少存储空间的需求,除了保存从文件中加载的源纹理图像外,算法流程中的模块2,3,4,6(图1)并没有存储实际的像素信息。在步骤(3)中,集合 $G_{sample-img}^H$ 和 $G_{sample-img}^V$ 中每个样本子图像实际上仅由一个四元素结构来表示,该结构描述了样本子图像在源图像中的起始位置和范围。

为了充分利用处理器频率,点集的获取和最佳切割线的计算所采用的算法的计算量较少。如Hammersley点集的计算复杂度为 $n \log_2 n$,且计算过程只涉及到移位、加法和乘法运算,而不需要耗时耗电的除法运算。

由于访问外部数据非常消耗电池,所以双向纹理合成算法中图像数据是一次性加载进内存,这样后续的数据访问都在内存中进行,不仅减少了外部数据访问的机会,相应地也减少了电池的消耗。

双向纹理合成算法在EVC的开发环境中实现,并得到了仿真器和HP iPAQ上的结果(如图8所示)。

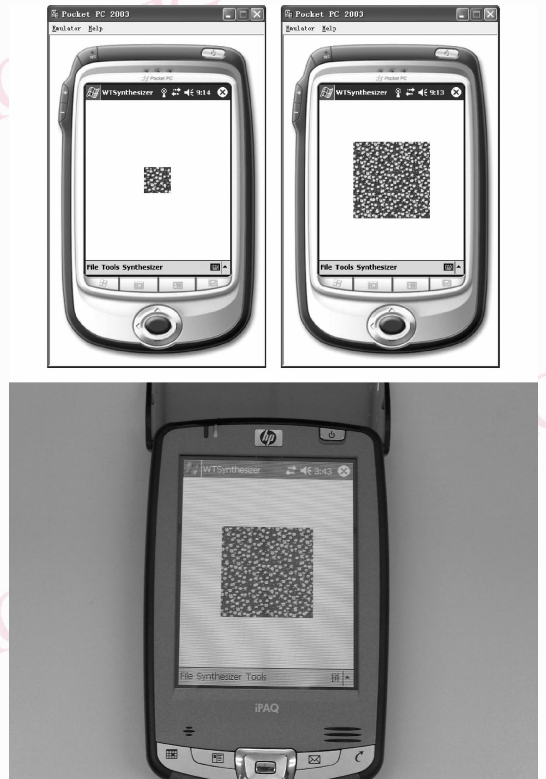


图8 双向纹理合成算法结果

Fig. 8 Result of Bi-direction texture Synthesizer

纹理合成图像的周期性现在并没有一个统一的衡量方法。虽然其可以通过统计纹理基元过渡变化

的快慢来表示,但对更大范围内的周期性特征就很难完全统计,所以目前更多地要依赖视觉判断。

图 9 列出了一组实验结果,通过对不同的输入纹理(左),比较了随机贴片算法(中)和本文算法(右)的视觉效果。从视觉效果上看,双向贴片算法在保持纹理的结构特征和非周期性方面比随机贴片算法要好。如果以纹理片在水平和垂直方向重复出现的几率 f 来衡量纹理基元过渡变化的快慢,则本文算法中的 f 比随机贴片算法要低,所以非周期性特征更为明显。

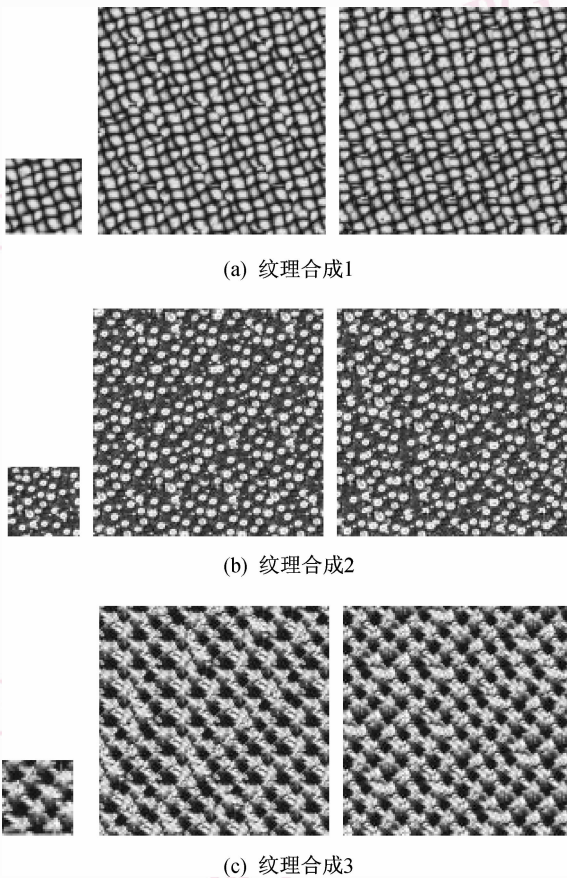


图 9 合成结果比较(从左至右:纹理,随机贴片;双向贴片)

Fig. 9 Comparison of synthesis results(From left to right: source texture, stochastic tiling and bi-direction algorithm)

6 结 论

目前针对移动平台上的纹理合成研究还不多。本文通过总结 PC 机上纹理合成技术研究的发展轨迹指出,作为样图纹理合成技术的一种,基于 Wang tiles 的纹理合成技术比较适合在移动设备上实现。Cohen 等人提出的随机贴片纹理合成算法既能保持重复利

用 tiles 来创建大场景的优点,又可避免明显的周期性的视觉瑕疵。但该算法存在较小的最佳切割线的搜索范围等不足,本文提出了改进的双向纹理合成算法,明确指出如何选择样本子图像,并通过片填充算法扩大最佳切割线的搜索范围,以及双向扫描技术增加片的非周期性。实验结果表明,该算法在视觉效果上具有一定改进,并可以在移动设备上运行。

参考文献 (References)

- 1 Akenine-Moller T, Strom J. Graphics for the masses: A hardware rasterization architecture for mobile phones [J]. ACM Transactions on Graphics, 2003, 22(3): 801-808.
- 2 Xu Xiao-gang, Bao Hu-jun, Ma Li-zhuang. Study on texture synthesis [J]. Journal of Computer Research and Development, 2002, 39(11): 1405-1411. [徐晓刚, 鲍虎军, 马利庄. 纹理合成技术研究[J]. 计算机研究与发展, 2002, 39(11): 1405-1411.]
- 3 Guo Yan-wen. Studies on Texture Mapping, Synthesis and Replacement [D]. Zhejiang: Department of Mathematics, Zhejiang University, 2006. [郭延文. 纹理映射、合成与替换算法研究[D]. 浙江: 浙江大学理学院数学系, 2006.]
- 4 Witkin A, Kass M. Reaction-diffusion textures [A]. In: Proceedings of ACM SIGGRAPH (Annual Conference Series) [C], Las Vegas, Nevada, USA, 1991:299-308.
- 5 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(1):49-71.
- 6 Ashikhmin M. Synthesizing natural textures [A]. In: Proceedings of the ACM Symposium on Interactive 3D Graphics [C], Research Triangle Park, North Carolina, USA, 2001: 217-226.
- 7 Efros A A, Freeman W T. Image quilting for texture synthesis and transfer [A]. In: Proceedings of ACM SIGGRAPH [C], Los Angeles, CA, USA, 2001:341-346.
- 8 Cohen M F, Shade J, Hiller S, et al. Wang tiles for image and texture generation [J]. ACM Transactions on Graphics, 2003, 22(3): 287-294.
- 9 Kopf J, Cohen-Or D, Deussen O, et al. Recursive Wang tiles for real-time blue noise [J]. ACM Transactions on Graphics, 2006, 25(3): 509-518.
- 10 Stam J. Aperiodic texture mapping [R]. Technical Report ERCIM-01/97-R046, European Research Consortium for Informatics and Mathematics (ERCIM), France, January, 1997.
- 11 Dong W M, Sun S X, Paul J C. Optimal sample patches selection for tile-based texture synthesis [A]. In: Proceedings of IEEE Computer Society Ninth International Conference on Computer Aided Design and Computer Graphics [C], Washington, DC, USA, 2005: 503-508.
- 12 Wong T T, Luk W S, Heng P A. Sampling with Hammersley and Halton Points [J]. Journal of Graphics Tools, 1997, 2(2): 9-24.