

基于复用计算的大纹理实时合成

陈 昕^{1),2)} 王文成¹⁾

¹⁾(中国科学院软件研究所计算机科学国家重点实验室 北京 100190)

²⁾(中国科学院研究生院 北京 100039)

摘 要 文中提出一种基于复用计算的纹理合成方法,逐步地利用已合成的部分纹理来生成更大的纹理块,以进行后续的纹理合成计算.由此,该方法可节省大量耗时的纹理块选择及缝合计算,提高了合成效率.实验表明,新方法可实时合成 2048×2048 像素的大纹理,而已有工作至多只能以交互的速度进行这样的合成.

关键词 纹理合成;复用计算;实时合成;大纹理;块合成方法

中图法分类号 TP391 DOI号: 10.3724/SP.J.1016.2010.00768

Reusing Partially Synthesized Textures for Real-Time Synthesis of Large Textures

CHEN Xin^{1),2)} WANG Wen-Cheng¹⁾

¹⁾(State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190)

²⁾(Graduate University of Chinese Academy of Sciences, Beijing 100039)

Abstract Example-based texture synthesis has proved to be an effective approach to generate a perceptually similar texture from a given small input exemplar image. It has found applications in many areas such as Virtual Reality and Geography Information Systems. Till now, a large number of techniques have been proposed, being able to synthesize textures of medium sizes in real time. However, the wide requirement in applications for real time synthesis of large textures is still a big challenge to existing techniques. With regard to this, this paper proposes to reuse partially synthesized textures to speed up texture synthesis, and by a lot of tests, it can real time produce large textures in 2048×2048 pixels. By its scheme, it first produces a small part of the output texture based on general constraint computation, and then iteratively generates bigger patches from the synthesized parts of the output texture to produce new parts of the output texture. In this way, the operations on selecting patches and stitching neighboring patches can be greatly reduced for acceleration. Meanwhile, the patch sizes and the width of the overlapping zones for stitching patches are computed optimally by investigating the periodic features of the example texture, so that the new method can produce high quality textures as existing advanced techniques such as texture optimization, one of the well-known techniques for producing high quality textures. The new method is simple and robust, very useful for real time synthesis of large textures.

Keywords texture synthesis; reuse; real-time synthesis; large textures; patch-based texture synthesis

1 引言

纹理合成旨在由输入的小块样本纹理生成视觉上相似的大块纹理,以有效重用光照信息,提高绘制效率. 尽管已有的方法已能够实时地生成中等尺寸的纹理,但对于一些需要使用大规模纹理的场景以及某些对速度要求更为严格的应用,已有工作的合成速度仍难以达到要求,制约了纹理合成技术的推广应用. 因此,尽可能提高纹理合成速度,始终是国际上探讨的热点问题.

目前,大多数的纹理合成方法都遵循 Markov 概率模型,即纹理中任一位置的色彩只由其局部有限邻域的情况决定. 因此,根据邻域情况就能逐步地扩展,完成目标纹理的生成. 而邻域搜索的效率就成为决定纹理合成效率的一个关键因素. 为此,人们提出了大量加速搜索技术,如用树结构组织纹理的邻域空间^[1]、用 K -coherence 来限定每次只搜索最相关的少量数据^[2]等. 这些早期的纹理合成技术,每次只是根据邻域生成一个新像素的色彩,能较好地反映纹理变化的丰富性,但对于纹理结构化信息的保持以及快速计算是不利的. 为此,2000 年以后,基于块的纹理合成方法逐渐发展起来. 块合成方法每次可根据邻域的情况生成一个包含很多像素的纹理块,这能很好地加快合成速度,并有利于保持纹理块内部的结构化信息^[3-4]. 2003 年提出的 Wang Tiles 方法^[5],事先对纹理块之间的可拼接性组合进行预处理,使得合成计算时不进行邻域搜索的计算,由此能进一步提高合成速度,但该方法可处理的纹理种类不多,且合成质量不是很高. 近年来,随着图形加速硬件的快速发展,相继提出了一些并行化的纹理合成方法,以利用 GPU 的并行计算能力. 2005 年提出的并行可控纹理合成技术^[6],将串行的约束关系转换为图像金字塔各层间的约束关系,由此可逐步求精地利用 GPU 并行地计算,以获得很高的速度,这是目前已知的合成速度最快的方法之一. 最近,文献^[7]采用了一种新的邻域约束计算方式,可在 GPU 上并行地实现块纹理合成,取得了比文献^[6]的方法更高的合成速度,但对于 2048×2048 这样的大纹理也只能以交互速度生成.

本文提出一种新的合成方法,以有效复用已合成的部分纹理来加快合成计算. 其基本步骤是:先利用文献^[7]的方法生成目标纹理上的一小块纹理,然后迭代地根据已生成的纹理获取尺寸更大的纹理

块,以进行目标纹理余下部分的合成. 在此过程中,为保证纹理块生成和复用的有效进行,我们运用文献^[8]的工作对样本纹理进行周期性分析,以使所生成的纹理块尺寸能高效地反映纹理特征变化的周期性,由此提高纹理合成的质量. 同时,我们还引入扰动及带有约束的随机选块计算,以增加纹理合成的多样性,进一步改善质量. 由于大量纹理块的选择计算和缝合计算的结果可被重复使用,因此,新方法的速度很快. 实验表明,新方法合成速度很快,能实时地合成 2048×2048 像素的大纹理,并具有很高的质量.

本文在第 2 节对相关工作进行对比讨论,在第 3 节对新方法进行详细的介绍,然后,在第 4 节对实验结果进行分析,并在第 5 节进行总结.

2 相关工作

根据处理对象的不同,纹理合成方法可分为两类:点合成方法^[1-2,6,9]和块合成方法^[3-5,10]. 点合成方法每次生成一个像素点的色彩,而块合成方法则是每次生成一个包含许多像素的纹理块. 一般而言,点合成方法便于反映纹理变化的多样性,但不利于保持纹理的结构化信息,且合成速度相对较慢;而块合成方法的速度较快,能较好地保持块内的纹理特征信息,但块之间的色彩过渡可能不很平滑,会引起合成质量的下降. 本文提出的方法是一个基于块的合成方法,它与相关工作的比较,在下面分别讨论.

块纹理合成方法有很多,其中文献^[3]的方法和文献^[4]的方法是最早提出的两种基于邻域搜索的块合成方法. 其中,文献^[3]的方法通过寻找相邻块在重叠区域的最小误差接缝进行块的缝合,而文献^[4]的方法则对重叠区域进行“羽化”(feathering)的色彩融合计算. 2003 年,文献^[10]提出的基于 Graph Cut 的方法,将重叠区域的缝合当作最大流/最小割问题进行处理,进一步改善了质量. 同年提出的 Wang Tiles 方法^[5],则在预计算时求出纹理块之间可组合拼接的模式,以便在合成计算时节省邻域搜索的开销,取得了很高的合成速度,但由于其所生成的可组合拼接的纹理块集合一般不大,合成结果缺乏多样性且可适用的纹理类型不多. 以上这些方法合成时均采用固定的块尺寸,并逐块地进行匹配搜索. 而本文方法在合成过程中不断扩大纹理块的尺寸,有利于减少合成过程需要计算的纹理块数目,以更好地加速,并对大多数纹理有效.

邻域搜索是合成计算中开销最大的部分,是加速方法研究的重点内容.文献[1]提出固定邻域尺寸的大小,以便于建树结构进行管理,从而提高邻域检索的效率的思想. K -coherence 方法虽然最先在点合成方法中提出^[2],但也可在块合成方法中得到应用.该方法预先为每个纹理块计算出可拼接的 K 个最相似纹理块,合成计算时只在该集中寻找合适的纹理块,以降低搜索空间. Jump Map 方法^[11]则是为每个纹理块记录可与其拼接的下一个纹理块的可能的位置集合,以便合成计算时随机选用,而不进行邻域匹配的计算.文献[7]为每个纹理块预先建立可拼接的纹理块集合,将搜索计算简化为求交计算,能进一步提高邻域搜索的效率,且可并行实现,但需要逐块的执行邻域匹配计算.本文工作在生成目标纹理的初始区域时采用文献[7]的邻域搜索方法,但在其它区域合成时是基于复用计算的,可节省大量的纹理块搜索计算,具有更快的合成速度.

随着图形硬件的发展,利用 GPU 加速合成的方法得到越来越多的关注.2004 年,文献[12]将 Wang Tiles 方法并行化以在图形硬件上高速实现,但其所适用的纹理种类非常有限.2005 年,Lefebvre 等人提出并行可控纹理合成方法^[6],建立高斯图像堆栈对纹理进行逐步求精的计算,能很好地利用 GPU 进行并行计算,这是目前所知的最快的纹理合成方法之一.为提高其合成质量,文献[9]将邻域信息也记录并参与合成计算中,尽管引入了多种加速策略,但也只能以交互的速度合成 512×512 像素的纹理.2008 年提出的多尺度纹理合成方法^[13],以文献[6]的工作为基础,可实现多尺度纹理的合成,但其合成质量受文献[6]的方法的影响,并不是很高.这些并行化的方法大都采用了多分辨率的合成模式,每层内也需要多次迭代,限制了合成速度的进一步提高.本文的新方法对所有像素只需处理一遍,且合成过程许多步骤也可并行化实现,因此将具有更高的合成速度.

除加速技术外,还有许多工作着眼于提高纹理合成的质量.其中基于全局优化的方法^[14]是目前合成质量最好的方法之一.该方法采用全局的相似性计算,降低全局误差,取得了很好的合成质量,但合成速度很慢.文献[15]的方法通过引入 K -coherence 以及 PCA 等技术使得文献[14]的方法可在 GPU 上高速实现,但其执行仍需要多层迭代,对较大尺寸的纹理只能以交互速度合成.新方法采用了

复用的思想,能够有效地利用已有的合成结果,减少选块以及缝合的时间开销;同时,新方法充分考虑了纹理的周期性特征,能很好地保持纹理的全局性特征,以生成高质量的纹理.

此外,一些纹理分析方法也可很好地提高纹理合成的效率.其中,文献[8]提出了一种优化块合成参数的方法,选择能够较好地反映样本纹理中的特征分布情况,有效地提高了纹理合成的效率.本文也采用了类似文献[8]的参数优化方法,以改善合成.

3 基于复用的块纹理合成方法

纹理是具有一定重复模式的图像,往往含有某种周期性特征.因此,合成过程中很多信息也会呈一定周期性地出现,可加以重复利用,以提高合成效率.为此,本文提出一种计算方式,以有效地复用纹理合成过程中已合成的部分纹理结果,促进合成效率的进一步提高.

3.1 方法概述

新方法的工作步骤是:先在目标纹理上按照邻域匹配搜索的方式合成一个小的初始区域,然后,对其余部分则重复利用已有的合成结果,从已合成部分中选择更大的纹理块进行填充,其工作流程如图 1 所示.图中左上的蓝色区域为初始合成的纹理区域,而红、黄、蓝 3 色的虚线框则表示复用计算时,从已合成纹理中选取的新的大纹理块.这些新选取的纹理块,将被填入到未合成区域中,形成一个扩大的连续的已合成的纹理区域,以完成 1 次复用计算.这样的复用计算可迭代地从已合成部分生成更大的纹理块,去合成待合成的区域.合成初始区域时,我们采用了类似文献[7]中的方法.根据优化的纹理块参数选取纹理块,并按照相位以及样本的相似性对纹理块进行组织.这里所说的纹理相位是仿照周期函数的描述,对样本纹理进行一次按块大小的均匀划分所得的纹理块,就是同一相位的纹理块;不同位置的划分,就产生不同相位的纹理块.此外,相似的纹理块也被认为是同一相位的.合成时,随机选择一个纹理块,并随机抽取与其相位相近的纹理块在该区域均匀地间隔地布块,然后再对空白区域进行填充.在对空白区域填充时,各个空白区域可独立地根据其周围已分布纹理情况进行匹配性计算来找到合适的纹理块进行填充.显然,间隔布块和空白区域的填充这两步计算都是可并行进行的.

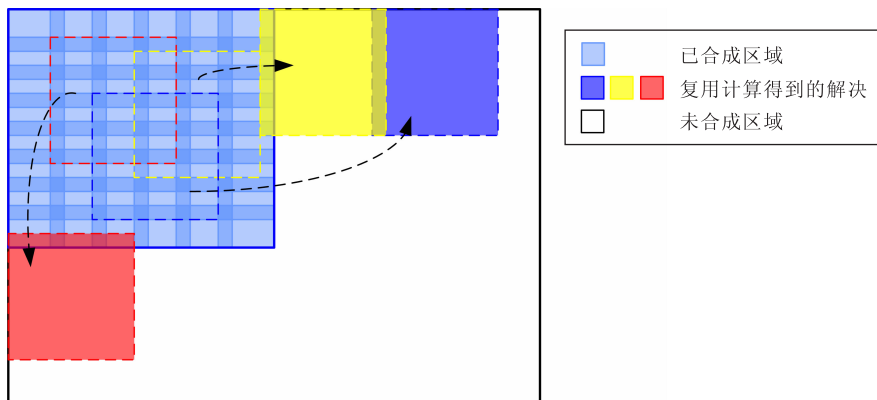


图 1 基于复用的一次合成计算

关于初始纹理的生成以及基于复用的合成将分别在 3.3 和 3.4 节中详细介绍. 在此之前, 我们在 3.2 节先介绍纹理块参数的优化选择.

3.2 纹理块参数的优化

纹理块的大小尺寸和相邻块之间的重叠区域宽度等参数对块合成方法的合成效率有很大影响. 对此, 文献[8]进行了深入的探讨, 并提出了一种参数优化的算法, 可以获得能很好反映纹理特征周期性变化的块尺寸大小和有效约束搜索的重叠区域的宽度. 本文采用了类似文献[8]的参数优化方法, 并略作调整. 该方法主要是探测不同尺寸和矩形形状的纹理块划分对样本纹理周期性全局特征的反映程度, 以此来决定纹理块的划分. 具体地, 要进行两种度量: 纹理块的信息包容性度量和纹理块的周期性度量, 并以这两种度量的结果都比较好的纹理块大小作为优化的纹理块尺寸.

一般地, 我们可以找到多个符合条件的块尺寸. 从合成速度方面考虑, 文献[8]倾向于选择较大的块尺寸. 但是, 过大的块尺寸, 会减少纹理块的数目, 降低合成纹理的多样性, 并降低邻接块之间缝合的效率. 因此, 实现新方法时, 我们通常选择样本纹理尺寸大小在 $1/4$ 至 $1/2$ 之间的那些符合条件的块尺寸, 作为纹理块划分的尺寸.

关于重叠区域宽度的尺寸, 文献[8]的方法是考察每种宽度下的重叠区域对纹理块进行区分的效率, 并选用高效的宽度作为重叠区域的宽度. 对于一种宽度而言, 如果该宽度下的重叠区域相似, 则它们对应的纹理块也相似, 反之, 则不是, 那么这样的宽度就是高效的. 文献[8]对纹理块的重叠区域分为左边界、上边界和左上边界, 分别计算. 但为简化缝合计算的复杂性, 本文则进行统一的一种宽度的计算, 经实验表明, 对合成质量影响不大.

3.3 初始纹理区域的合成

新方法首先合成一个较小的纹理区域, 而其余部分则根据已合成结果进行复用来合成. 这里, 我们采用类似文献[7]的邻域约束方式, 以进行初始区域的纹理合成. 首先, 在该区域上分布一些纹理块, 使得水平方向以及垂直方向的纹理块之间均间隔一个块大小的空白区域; 再根据周围已填入的块的情况, 选取合适的块填入空白区域, 以完成合成. 在初始布块时, 我们选取相位相近的纹理块进行布块; 在空白区域填充时, 为样本中的每个纹理块预计算出可与其邻接匹配的样本块的集合(称为相容性集合), 而在合成时, 就根据周围已有的纹理块, 从它们各自的相容性集合的交集中随机选取一块即可. 对于交集为空的情况, 则选取在各集合中出现次数最多的纹理块. 实验证明, 这样的操作对合成结果的质量影响不大. 此外, 考虑到纹理的随机性, 我们引入了一个类似文献[6]方法中的随机性算子 Γ , 先如前所述选出一个备选块, 再在这个块的基础上在其距离 Γ 范围内随机选择一个块作为实际填入的块. 而 Γ 的取值可根据样本的不同而进行调整. 这样保证了合成结果既能够反映样本的周期性特征又能体现出其随机性. 文献[6]的方法及本文的实验结果均证实了其有效性. 选定了纹理块后, 需要将它们写入并缝合至目标纹理中. 这里, 我们采用了“羽化(feathering)”的缝合方法^[4], 即对重叠位置的像素根据其到边界的距离进行色彩的线性插值计算. 该方法计算快捷, 尽管对于一些结构化特征较强烈的纹理缝合结果不够理想, 但从结果来看, 基本上可以保证合成结果的质量.

在以上的合成过程中, 布块操作和空白区域填充操作中各个局部之间的的计算相关性很小, 可分别并行计算. 为此, 文献[7]的方法将其在 GPU 中

实现,以有效利用 GPU 的并行计算能力.但是在块的写入缝合过程中,具有大量的连续存储空间的读取,而 GPU 的逐像素处理方式使其无法利用这一连续性.为此,本文方法在实现时,充分考虑这一连续存储关系,并在 CPU 中借助多线程等并行手段,对这一合成过程进行加速.在不考虑复用的情况下,亦可取得较文献[7]的方法更高的合成速度.

3.4 基于复用计算的纹理合成

新方法在合成过程中,逐步利用已合成的计算结果来生成更大的纹理块,以完成整个纹理的合成.图 1 是一次复用计算的简单示意图.图中左上的蓝色区域是已合成的纹理区域,它是由若干小纹理块缝合拼接而成的.为了完成余下白色区域的合成,我们计算新的纹理块尺寸,并从已合成的蓝色区域中选取纹理块以填充.图中蓝色区域上的红、黄、蓝三色的虚线框所包含的区域分别代表 3 个新尺寸的纹理块,选取之后,分别填入未合成区域的对应位置.这里,新的纹理块尺寸除去重叠区域宽度 e 的因素,应完整地包含整数个初始纹理块,即若初始合成区域合成阶段采用的块尺寸为 (w, h) ,则新的块尺寸 (w', h') 应满足 $w' = a \times (w - e) + e$, $h' = b \times (w - e) + e$, a 和 b 均为正整数,且 a, b 的取值随着已合成区域尺寸的增长而增大,增长方式可根据区域增长的大小进行调整,本文多采用成倍增长的方式.这种采用大纹理块的方式,既可以减少合成中需要处理的纹理块的数目,又可以有效地利用已有的选块和缝合结果,对提高纹理合成效率非常有利.实现中,我们将新块的尺寸设为已合成区域长宽的各一半,当已合成区域的长宽各扩大一倍后,我们将该区域作为新的已合成区域,开始下一次的复用计算,即重新计算纹理块尺寸,并选块进行下一步的合成.

需要说明的是,复用阶段选块时,我们只需按照相位的标准计算一个随机的块位置即可,而无需邻域搜索计算.这是得益于以下两方面的工作:(1) 预计算时我们参照文献[8]的方法对块参数进行了优化计算,使得优化后的块尺寸能够很好地反映样本整体的周期性分布情况;(2) 初始区域合成时,我们按照相位相近的原则分布纹理块,使得已合成区域的结果也符合这一分布特征.因此,复用阶段,只要满足这一相位约束,即可保证所选的纹理块符合样本的周期性特征,且可以和相邻位置的块较好地拼接.具体地,复用计算时,块的坐标为 $(s \times (w - e), t \times (h - e))$, s 和 t 均为正整数,在不越界的前提下

可随机选取,这样也有利于破坏纹理的重复性,增强合成结果多样性.

综上,新方法只在合成初始纹理区域时,需要采用集合求交计算进行块的搜索;而在复用阶段,只需几次简单的四则运算以及求随机值即可完成选块操作.由于块尺寸相同的各块间的关联性很小,因此新方法可利用多线程等手段进行并行计算以进一步加速.总之,新方法能大幅减少需要处理的纹理块数目,并无需多层迭代就能进行并行计算,因此能很好地加速合成计算.

4 实验结果及分析

4.1 合成速度的分析

我们用 Visual C++ 2005 实现了本文的新方法,并在一台 Dell Optiplex 755 的 PC 上进行了实验.这台微机配有一个 Intel® Core™ 2 Duo E6550 2.33GHz 的 CPU, 2GB 内存.通过测试多种不同样本纹理合成的情况,表 1 列出了新方法合成不同尺寸目标纹理的平均时间开销以及同其它一些先进方法的合成时间的对比.其中,非复用方法是指采用本文初始区域合成阶段的合成策略,而不使用复用计算的合成方式;文献[7]的合成时间来自其文章中的数据;文献[6]是一个基于 GPU 的算法,考虑到近年来硬件性能的快速发展,为了不失公平性,这里采用的是 SIGGRAPH'2008 上发表的论文^[13]中测试得到的文献[6]的合成速度,在配置有一个 GeForce 8800 GTX 的 GPU 的机器上的处理效率是 15.3M 像素/秒,根据该数据换算得到合成各种尺寸纹理所需的时间.从这些数据可知,新方法很好地加速了合成计算,能对 2048×2048 像素的大纹理进行实时合成.

表 1 新方法的合成计算效率 (时间单位:ms)

纹理尺寸 (像素)	新方法 合成时间	非复用的 合成时间	文献[7]的 合成时间	文献[6]的 合成时间(换算)
256×256	0.68	0.67	10.23	4.08
512×512	1.39	2.65	12.41	16.34
1024×1024	3.81	10.82	21.78	65.36
2048×2048	13.47	45.79	58.93	261.44

为考察复用计算对合成效率的影响,图 2 列出了新方法及未采用复用计算的合成效率,以平均合成每兆(M)像素所需的合成时间进行度量.从图 2 中可以看出,除了在合成 256×256 的纹理时,新方法比非复用方法略慢外,其它情况下,新方法的合成

速度是快的. 因为, 在合成 256×256 大小的纹理时, 新方法复用的策略没有得到应用, 两种方法的合成方式也就近似相同. 当合成纹理的尺寸逐渐增大时, 非复用方法的平均合成时间几乎不变, 而新方法的平均合成时间逐渐降低, 当合成尺寸大于 1280×1280 时, 下降幅度逐渐变小, 趋于平稳. 对此, 我们的分析如下: 新方法的合成时间由初始纹理区域的合成时间 T_0 、复用阶段的选块时间 T_{patch} 、缝合重叠区域的时间 $T_{fettinging}$ 以及复制像素的时间 T_{copy} 这 4 部分组成, 对于大尺寸纹理的合成来说, T_0 相当于一个常量; 复用计算的选块操作计算量也是固定的, 当每层复用选块个数相同时, T_{patch} 也可认为是一个常量; 而当复用计算的块尺寸逐渐增大时, 重叠区域相对于其它区域所占的比例逐渐减少, 因此, 纹理合成的时间会随着尺寸的增大越来越接近于直接进行像

素复制的时间, 这已经是一个纹理合成过程最基本的操作了. 因此, 新方法基于复用可极大地降低合成计算的开销, 非常有利于合成大尺寸纹理.

4.2 新方法的合成质量

由于新方法采用了优化的纹理块尺寸, 有效地反映了样本纹理特征的周期性, 所划分的纹理块对纹理特征的周期性变化有很好的反映, 因此, 它能高效地处理纹理合成过程中的特征变化, 生成高质量的纹理. 图 3 中列出了新方法、文献[7]的方法、全局优化合成方法^[14]和并行可控纹理合成方法^[6]生成的一些对比纹理, 其中, 对比的纹理是从这些文献及其相关网页中下载的. 从图中可见, 新方法生成的纹理质量要优于并行可控纹理合成方法, 与全局优化方法相当, 甚至某些结果还要优于全局优化方法, 如图 3(b)中关于纹理结构化信息的保持. 图 4 中还列出了新方法生成的其它许多纹理, 由于页面有限, 无法列出较大尺寸的纹理, 这里给出的都是 512×512 像素的, 这些结果都有很高的质量且合成速度很快. 作为参考, 图 2 中还给出了相应的合成参数. 从这些实验结果可知, 新方法具有很高的工作效率, 能生成高质量的纹理. 但由于新方法采用“羽化”的方法进行邻接纹理块的缝合, 所以, 它还难以高质量地处理结构化信息比较强烈的纹理, 比如图 3(c)中左边的纹理. 总的来说, 新方法可如大多数先进方法一样, 进行高质量的纹理合成.

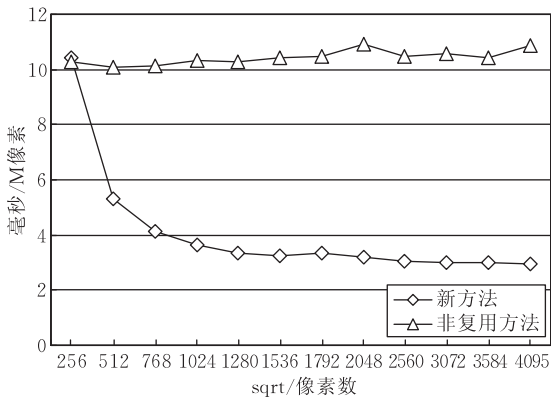
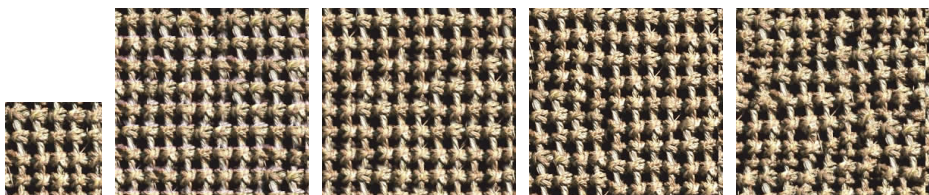


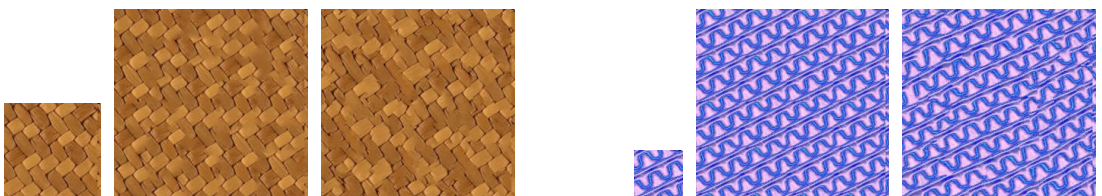
图 2 复用计算的合成效率



(a) 从左至右: 样本纹理、新方法的结果、文献[7]中方法的结果、全局优化方法的结果^[14]、并行可控方法的结果^[6]



(b) 从左至右: 样本纹理、新方法的结果、全局优化方法的结果^[14]



(c) 从左至右: 样本纹理、新方法的结果、并行可控方法的结果^[6]

图 3 新方法、文献[7]中的方法、全局优化方法^[14]和并行可控纹理合成方法^[6]所生成纹理的对比

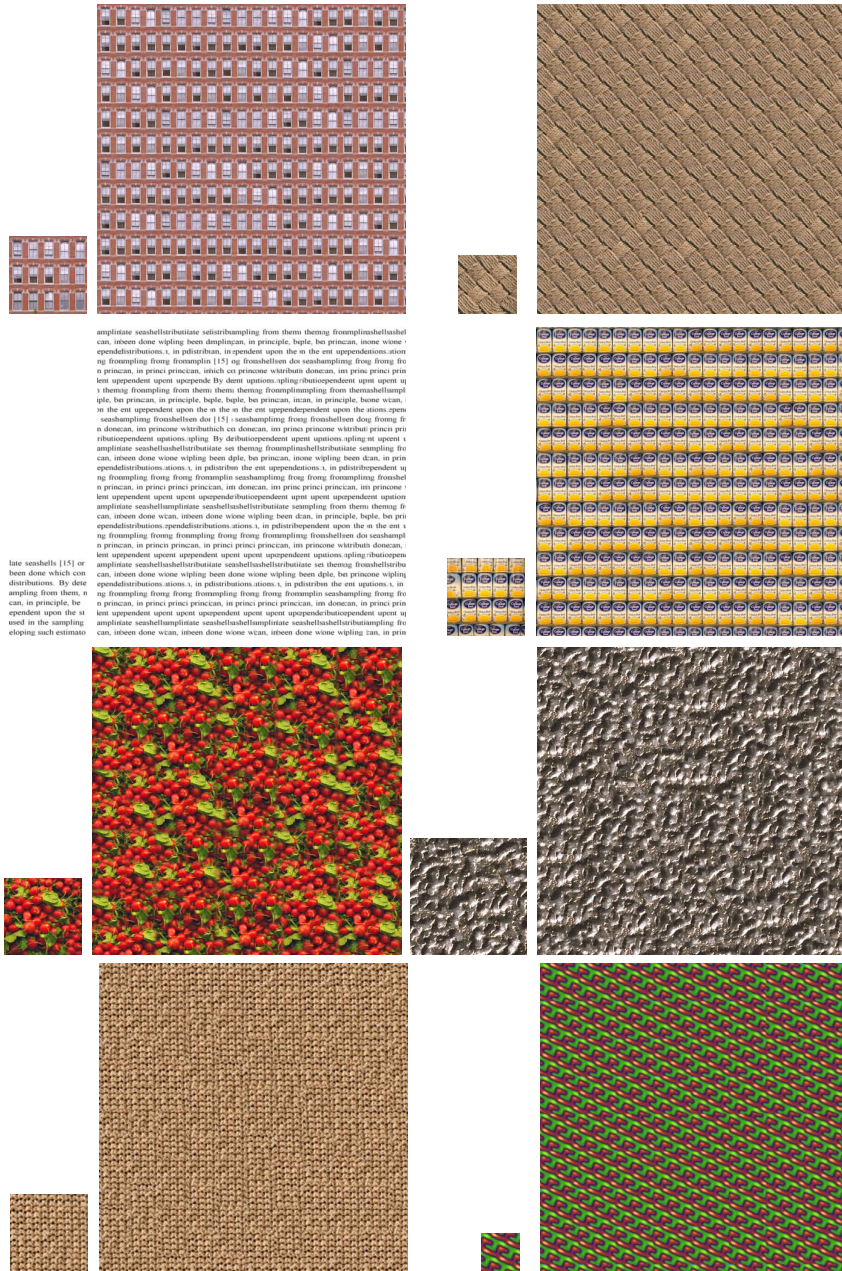


图 4 新方法所生成的多种纹理(512×512)(这里,样本纹理位于左侧,合成结果在右侧)

5 结语

本文提出了一种新的纹理合成方法,只需基于邻域搜索合成一个较小的初始目标纹理,即可基于复用的方式利用已合成结果逐步生成更大的纹理块,以高速完成目标纹理的合成.这种复用的合成方式,大大减少了邻域搜索以及缝合计算的计算量,非常适合于大尺寸纹理的合成.实验表明,与已有工作相比,新方法极大地提高了合成速度,生成一个2048×2048像素的纹理,只需十几毫秒,而已有工作最多只能以交互的速度合成这样大的纹理.

由于新方法采用“羽化”的方式进行邻接纹理块的缝合,难以高质量地处理结构化特征较强的纹理.这也是纹理合成技术中长期探讨的一个难点问题.为促进纹理合成技术的推广应用,我们将进一步探索合成高质量纹理的方法.

参 考 文 献

[1] Wei L, Levoy M. Fast texture synthesis using tree-structured vector quantization//Proceedings of the ACM SIGGRAPH 2000, ACM Press/ACM SIGGRAPH, New Orleans, 2000: 479-488

- [2] Tong X, Zhang J, Liu L, Wang X, Guo B, Shum H Y. Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Transactions on Graphics*, 2002, 21(3): 665-672
- [3] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer//Proceedings of the ACM SIGGRAPH 2001, ACM SIGGRAPH, New York. Fiume E ed. *Computer Graphics Proceedings, Annual Conference Series*, ACM, 2001: 341-347
- [4] Liang L, Liu C, Xu Y, Guo B, Shum H. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 2001, 20(3): 127-150
- [5] Cohen M F, Shade J, Hiller S, Deussen O. Wang Tiles for image and texture generation. *ACM Transactions on Graphics*, 2003, 22(3): 287-294
- [6] Lefebvre S, Hoppe H. Parallel controllable texture synthesis. *ACM Transactions on Graphics*, 2005, 24(3): 777-786
- [7] Chen X, Wang W. Real-time synthesis of large textures. *Journal of Software*, 2009, 20(Supplement): 193-201 (in Chinese)
(陈昕, 王文成. 大尺寸纹理的实时合成. *软件学报*, 2009, 20(增刊): 193-201)
- [8] Wang Y, Wang W, Wu E. Optimizing Implementation of patch-based texture synthesis. *Journal of Computer-Aided Design & Computer Graphics*, 2006, 18(10): 1502-1507 (in Chinese)
(王一平, 王文成, 吴恩华. 块纹理合成的优化计算. *计算机辅助设计与图形学学报*, 2006, 18(10): 1502-1507)
- [9] Lefebvre S, Hoppe H. Appearance-space texture synthesis//Proceedings of the SIGGRAPH. New York, NY, USA: ACM, 2006: 541-548
- [10] Kwatra V, Schodl A, Essa I, Turk G, Bobick A. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 2003, 22(3): 277-286
- [11] Zelinka S, Garland M. Jump Map-based interactive texture synthesis. *ACM Transactions on Graphics*, 2004, 23(4): 930-962
- [12] Wei L. Tile-based texture mapping on graphics hardware//Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware. Grenoble, France, 2004: 55-63
- [13] Han C, Risser E, Ramamoorthi R, Grinspun E. Multiscale texture synthesis. *ACM Transactions on Graphics*, 2008, 27(3), Article No. 51
- [14] Kwatra V, Essa I, Bobick A, Kwatra N. Texture optimization for example-based synthesis. *ACM Transactions on Graphics*, 2005, 24(3): 795-802.
- [15] Huang H, Tong X, Wang W. Accelerated parallel texture optimization. *Journals of Computer Science and Technology*, 2007, 22(5): 761-769



CHEN Xin, born in 1982, Ph. D. candidate. Her research interests include texture synthesis and realistic rendering.

WANG Wen-Cheng, born in 1967, Ph. D., professor, Ph. D. supervisor. His research interests include virtual reality, visualization and large-scale scenes rendering.

Background

Texturing is a core process for computer graphics applications. To acquire textures, there are a variety of methods such as manual drawing and photography. However, example-based texture synthesis remains the most powerful method for its ability to generate a perceptually similar texture from a given exemplar image. The output texture can be arbitrarily large and does not contain any unnatural artifacts. Furthermore, the storage requirement has been strongly reduced by the on-the-fly generation of texture content.

In the last decade, many techniques have been proposed to promote the efficiency of texture synthesis. Textures in medium sizes have already been able to be generated efficiently. However, large textures in high quality are needed in

many applications for tasks such as terrain rendering and exhibition with display walls. Thus, synthesizing large textures in real time becomes a hot topic in computer graphics.

A novel method is proposed in this paper to synthesize large textures. It reuses partially synthesized textures to speed up texture synthesis. According to the experiments, the new method can run efficiently to be able to produce large textures of high quality in real time, e. g. generating a texture in 2048×2048 at over 74fps on a common personnel computer, which is still a big challenge to existing methods.

This project is sponsored by the National Natural Science Foundation of China (Nos. 60773026, 60873182, 60833007).