

基于约束的多样图纹理设计与合成算法^①

郑秋旭 陈昭炯 (福州大学 计算机科学与技术 福建 福州 350108)

摘要: 提出了多样图纹理设计的概念, 将块拼贴纹理合成方法和点匹配纹理合成方法相结合合成纹理图案。利用块拼贴合成方法合成背景及一些前景的部分, 避免了基于点的纹理合成所产生的条带化、模糊等现象。同时, 采用点匹配的合成方法合成一些前景及一些前景的部分, 解决了块拼贴在小区域覆盖到其他区域的问题, 并且利用图像平滑技术, 使拼接处自然过渡。由于采用了以上两个算法相结合, 在合成速度上已经大大超过了徐晓刚约束多样图纹理合成算法。

关键词: 纹理设计 多样图 纹理合成 动态合成窗口 相关性

Texture Design and Synthesis Based on Constrained Multiple Sources

ZHENG Qiu-Xu, CHEN Zhao-Jiong

(Department of Computer Science and Technology, Fuzhou University, Fuzhou 350108, China)

Abstract: This paper presents a concept of texture design based on the current constrained multiple sources texture synthesis algorithm, and applies the patch-based texture synthesis to synthesis of texture design. With the predominance of the patch-based texture synthesis to synthesize background, it avoids the phenomenon of texture element malposition, the appearance of strip and blur based on only points matching. At the same time, it adopts the Ashikhmin's texture synthesis algorithm to synthesize foreground based on points matching to solve the problem of patch-based texture synthesis. The other region is covered with big pieces, by utilizing the image fusion technology to flat the place of splicing naturally. Owing to coalesce the above two algorithms, the speed of this algorithm has preponderated over Xu's greatly.

Keywords: texture synthesis; multiple sources; texture design; dynamic synthesis windows; pertinence

1 引言

纹理通常指的是在局部区域内呈现不规则, 而在整体上表现出某种规律的特征图像, 而纹理合成是基于给定小样图的纹理, 按照纹理表面的几何形状特征, 生成在视觉上连续并且和样图相似的大面积的曲面纹理。

纹理合成技术除了在真实感绘制、计算机动画、图像压缩、图像修复和编辑等方面有着广泛的应用, 近年来, 还被应用于艺术图像风格的学习与生成等方面。

多样图纹理合成是基于给定的多个小样图纹理,

合成样图各占一定比率的新大面积曲面纹理。在目前的纹理合成研究中, 其研究仍然较少, 大部分的合成算法只适合于单样图的纹理合成。Bar-Joseph 曾提出一种采用统计学习的方法对两种纹理样本进行纹理合成^[1], 但是该算法复杂, 只能处理部分纹理, 有较大的局限性。目前国内对于约束的多样图纹理合成的研究有^[2,3], 都是利用点匹配纹理合成算法, 在合成速度上比较慢, 也比较容易产生条带化现象。总之, 利用已有样图生成纹理图片已经满足不了人们在实际生活中各个方面的需求。

① 基金项目: 国家自然科学基金(60805042); 福建省自然科学基金(A0710006)
收稿时间: 2009-05-11

综上,提出一种新的基于约束的多样图纹理合成算法,并提出纹理设计的概念,即输入多个纹理图 and 用户提供的目标图,通过我们的算法,自动生成符合用户需求的纹理图案。该算法可以合成丰富多彩的纹理图案,满足用户各种各样的需求。所谓的目标图指的是用于约束纹理图案合成的人工设计图案或者照片等。

本文将 Efrog-Freeman 块匹配纹理合成方法^[4]和 Ashikhmin 提出的基于相关性原理的点匹配纹理合成方法^[5]相结合,提出了一些改进措施,并且将其应用到多样图纹理设计。利用块拼贴纹理合成的优势合成背景及一些前景的部分,避免了基于点的纹理合成所产生的条带化等现象。同时,采用 Ashikhmin 提出的基于点匹配的纹理合成方法合成一些前景及一些前景的部分,解决了块拼贴在小区域覆盖到其他区域的问题,并且利用图像平滑技术,使拼接处自然过渡。该算法不但提高了合成速度,而且在一些纹理取得了良好的效果。

本文算法的特点有:

- ① 分割目标图。
- ② 允许指定分割区域对应输入纹理。
- ③ 采用动态纹理合成窗口。

2 传统的基于样图的纹理合成方法

2.1 Ashikhmin 点匹配纹理合成方法

Ashikhmin 纹理合成算法是对 Li-Yi Wei 算法^[6]的改进,它利用相关性原理,把纹理的搜索范围限制在当前点的邻域,很大程度上减小了搜索空间,使合成速度基本与样图大小无关,提高了合成速度。

该算法的基本思想是根据 L 邻域点在样本纹理中的对应位置偏移相应量的候选点,即当前像素邻域的各个像素点在源图像中对应的相应偏移量的像素点,例如是 3*3 的 L 邻域,那么对应的像素点分别为倒 L 的右边、右下、下边和左下像数点。利用公式(1)比较各候选点邻域与当前点的邻域的误差值,选取邻域误差最小的点作为匹配点,把它的像素值拷入合成纹理中的当前像素的位置。

$$d(N_i, N_o) = \sum_{p, q \in L} \sqrt{(R(p) - R(q))^2 + (G(p) - G(q))^2 + (B(p) - B(q))^2} \quad (1)$$

公式(1)中, R()、G()、B()分别为像素 p、q 的红、绿、蓝三颜色通道。N_i、N_o分别是输入纹理和输出纹理中某一点的 L 邻域。

该算法对自然纹理和随机纹理的合成取得了很好的效果,如草地、灌木丛等,而对结构性较强的纹理,该算法合成质量不如 WL 算法,其主要原因是相关性度量的范围过窄,把所搜范围限制在当前点的邻域,对许多边界比较明显的结构性纹理,相关性度量范围相对过小。

2.2 Efrog-Freeman 块拼贴纹理合成方法

Efrog-Freeman 在 2001 年提出了一种基于块拼贴的纹理合成算法^[4],其思想如下:

在输入样图中任取一块 B₁,放在输出图像中,然后在输入样图中查找 B₂,使 B₂放入输出后与 B₁有一定的重叠,且匹配边界误差控制在一定范围内;接着在 B₁, B₂ 的重叠区域找出一条误差最小的路径作为 B₂ 的边缘,把 B₂ 贴入合成的图中。反复重复以上过程,直到合成结束。误差最小的路径通过以下方法进行计算:设 B₁, B₂ 沿垂直边重叠,误差曲面定义为 $e = (B_1^{ov} - B_2^{ov})^2$,通过公式(2)重叠区最后一行的各点误差。其中 B₁^{ov}、B₂^{ov}分别为纹理块 B₁和 B₂各自的重叠区域。

$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i,j-1}, E_{i+1,j-1}) \quad (2)$$

取误差最小的一点,反向跟踪获得最佳缝合路径。对于水平方向重叠区域,可以采用类似的方法得到最佳缝合路径。Efrog-Freeman 算法在合成纹理时,在水平和垂直方向都会出现重叠,两条路径会在某一点相交,选取最小误差的路径作为最佳缝合线。

该算法比起以往算法,在纹理合成的时间,合成纹理的视觉效果方面都有很大的提高,避免了以往算法易引起的纹元错位、条带化、模糊等现象,但该算法存在的问题是有时纹理重复过多,边界不匹配。

3 一种新的基于约束的多样图纹理设计与合成算法

基于约束的多样图纹理合成指的是,在目标图的约束下,通过多样图合成算法合成相应纹理图案。而多样图纹理设计指的是,用户提供一个目标图,并且这个目标图能分割成许多区域分别对应于输入样图,将其与样图输入我们的算法,自动生成符合用户提供目标图时所期望得到的纹理图案。

由于 Efrog-Freeman 块拼贴纹理合成方法每次可以合成多个像素,所以保持了纹理的局部信息,对结构性纹理和随机性纹理的合成获得了很好的效果,然而基于这个方法带约束纹理合成方法(纹理传递)

却不能应用到带约束的多样图纹理合成,其原因是块拼贴的基本要求是合成窗口大小为一个基本纹元,而纹理传递通过迭代并且每次纹理块缩小,使得很多情况下合成窗口大小小于一个基本纹元,导致合成纹理图案许多纹元不完整,甚至模糊不清。因此,纹理传递只适合于输入样图为随机纹理,因为随机纹理的基本纹元较小。

由于带约束的多样图纹理合成是在不同区域合成不同的纹理,因此在进行匹配纹理块的查找时,边界重叠区域是否取自同一样本纹理是能否获取匹配纹理块的关键。如果某待合成纹理块的边界重叠区域取自不同的样本纹理,利用同样约束规则获得的匹配块,与合成区域在很大程度上并不匹配,不能获得无缝拼接。而对于包含有自然纹理的带约束的多样图合成,块拼贴更是显得无能为力。

Ashikhmin 提出的基于相关性原理的点匹配纹理合成方法,在自然纹理和随机纹理的合成中都取得了很好的效果。徐晓刚提出的一种基于此方法的多样本纹理合成算法^[2],虽然在提高了合成速度,但合成纹理图案仍然存在条带化现象。由于 Ashikhmin 提出的基于相关性原理的点匹配纹理合成方法相关性度量范围相对过小,并且由相关性计算得到的像素位置不在输入样图的范围之内则随机选取一点作为当前的合成像素,因此可能导致纹理合成图模糊和产生条带化现象。

综上所述,本文提出一种新的基于约束的多样图纹理设计与合成算法,解决了以上问题。

3.1 算法思想及分析

本文将 Efros-Freeman 块匹配纹理合成方法和 Ashikhmin 提出的基于相关性原理的点匹配纹理合成方法相结合,利用块拼贴纹理合成的优势合成背景及一些前景的部分,避免了基于点的纹理合成所产生的条带化等现象。同时,采用 Ashikhmin 提出的基于点匹配的纹理合成方法合成一些前景及一些前景的部分,解决了块拼贴在小区域覆盖到其他区域的问题,并且利用图像平滑技术,使拼接处自然过渡。

这种方法扬长避短,提高了合成速度的同时避免合成纹理图案纹元错位、条带化、模糊等现象,在一些纹理图案的合成中都取得了不错的效果。

3.1.1 目标图分割及区域对应输入纹理的指定

由于带约束的多样图纹理合成是在不同区域合成不同的纹理,会出现了目标图区域分割问题和不同区

域的输入纹理的选择问题。

当目标图为用户手工画时,先将目标图灰度化,然后利用灰度图像域值分割算法将其分割成背景和前景,并利用该算法将前景继续细分直到符合用户需求,这样就将目标图分割成各个对应输入纹理的合成区域。然而,当目标图是一张相机拍摄得到的照片时,以上算法就不那么行之有效了。因此,在以上算法得到灰度图像域值 T_0, T_1, \dots, T_n 的辅助之下,与用户进行交互,让用户可以根据当前域值进行调整,直到用户满意对应输入纹理合成区域的划分。

以往方法在不同区域的输入纹理的选择,通常是设定一个目标函数,当目标图与某输入纹理相近时,则以较大的概率取该纹理作为当前区域的输入纹理^[2]。而作为用户,希望的是能合成符合自己需求的纹理图案,则以往的算法有可能和用户想法南辕北辙。因此,对于对应区域输入纹理的指定问题,可以采用以往方法,也可采用和用户交互,让用户在指定区域指定相应的输入纹理,达到提高最终纹理合成图案质量的目的。未来,我们可以根据用户喜好,保存一些相关信息,供下次合成纹理图案使用。

3.1.2 不同纹理样本合成窗口的选择

由于我们采用块拼贴纹理合成算法合成背景和一些前景的部分,这样就出现了不同纹理样本合成窗口大小的选择问题。在一般情况下,利用块拼贴纹理合成算法合成纹理,要求纹理块大小为一个基本纹元大小。

假设多个输入样图纹理的基本纹元大小分别为 $b_1 * b_1, b_2 * b_2, \dots, b_n * b_n$, 一般情况下我们取合成窗口 $B = \max(b_1 * b_1, b_2 * b_2, \dots, b_n * b_n)$ 代替所有合成窗口,但这样的选择往往又有另一方面的缺陷,如果 B 比许多多样图的基本纹元大太多,使得某些合成纹理图案重复太多,质量下降;又若选择合成窗口太小,小于许多多样图的基本纹元大小,则出现许多纹元不完整,合成图案质量大打折扣,用户不满意。

基于以上问题,我们提出采用动态大小合成窗口方法,即将事先将各个样本合成窗口大小保存下来,合成时就采用相应的合成窗口。未来,我们将研究样图纹理样本合成窗口的自动提取,避免由于观察引起的不准确。

3.1.3 带约束的纹理背景图合成

当利用块拼贴算法合成带约束的纹理背景图时,不仅需要考虑到当前合成块和已经合成区域的重叠错误,而且还需要把当前合成块和目标图对应区域的颜色值之间的错误也考虑进来,从而找到一个错误在容

忍范围内的纹理块作为当前合成块。

由于 Efron-Freeman 块拼贴纹理合成方法中纹理传递只适合于随机纹理,在本文的研究中并不适用,因此对其改造,只进行一轮迭代来合成背景图。

利用一个二维数组进行标注,若为背景的像素对应位置标记为 0,前景区域分别标记为 1,2,...,n。利用从左到右,从上到下的扫描线顺序进行块拼贴纹理合成背景,当合成当前块对应区域包含 0 则合成,直到包含 0 的区域被全部覆盖,并将已合成区域对应的像素位置标记为 0 的全部标志成-1,表示已合成。

3.1.4 纹理设计前景图的合成

若前景存在大于等于对应输入纹理合成窗口大小的区域,则采用块拼贴纹理合成直到不存在这样的区域,剩余的不规则区域采用点匹配算法进行合成;若前景不存在这样的区域直接采用点匹配算法进行合成。

分别对前景区域,利用从左到右,从上到下的扫描线顺序进行块拼贴进行纹理合成。并将已合成区域对应的像素位置全部标志成-1,表示已合成。

分别对一些前景和前景通过块拼贴后未合成的区域,利用 Ashikhmin 点匹配方法,按“之”字线扫描线顺序(可避免随机选取像素点后可能产生的条带化现象),依次合成各个像素点,将已合成点对应的像素位置标志成-1,直到合成结束。

3.1.5 不同纹理样图合成边界处理技术

相邻的不同纹理区域合成后,有明显的边界现象,显得不和谐。如果只是简单的采用边界相邻两点像素的平均值来代替当前边界点,边界仍然比较明显。因此,本文采用图像平滑技术,对边界区域进行平滑,达到边界区域自然过渡的效果。

平滑技术一般采用平滑滤波器进行模糊处理,即用滤波模板确定的邻域内像素的平均值去代替图像中的一些像素点的值,这种处理可以减少图像的“尖锐”变化——明显边界现象。

通过灰度图域值分割后,得到了分割区域,搜索出各个区域边界,利用平滑技术对其进行平滑。若对整幅纹理图案进行平滑,会导致纹理图案亮度下降甚至模糊,与样图纹理不相似。边界区域,我们设定为平行于边界的两侧各为对应区域输入样图合成窗口大小的 1/12。

我们在 Matlab 环境中进行各种滤波器的实验,结果发现其中 average 滤波器对结构性强的纹理取得了较好的效果,而 disk 则对自然纹理和随机纹理取得

了较好的效果。从实验中我们得出一个结论,滤波器的选择与输入样图的纹理结构有关,而滤波器的大小则与输入样图的窗口大小有关。

3.2 算法实现步骤

本文合成方法步骤描述具体如下:

① 设计目标图 T,如图 1 中(a)(b)(c)(d)中各图中从左到右的第三张图,可以是手工画,也可以是照片。

② 将彩色图像目标图 T 灰度化,然后利用灰度图像域值分割算法,或者对分割得到的域值进行调整,将目标图分为前景和背景,继续通过灰度图像域值算法对前景进行分割,直至符合用户的要求。

③ 利用 Efron-Freeman 块拼贴纹理传递方法的第一轮迭代将背景图拼贴出来。

④ 分别对前景区域,利用从左到右,从上到下的扫描线顺序进行块拼贴进行纹理合成,分别对一些前景和一些前景的部分,按“之”字线扫描线顺序,利用 Ashikhmin 点匹配方法依次合成各个像素点,直到合成结束。

⑤ 对不同纹理样图合成边界区域进行图像平滑。

4 实验结果与分析

基于上述算法,我们进行了实验,为简单起见,我们的输入样图数为 2。实验中,在 RGB 空间中进行误差计算,L-邻域采用 3*3。在 Pentium IV 2.93GHz, 1G 的 PC 机上,一般情况下,十秒内可以完成。对于合成图片的大小、邻域大小的不同和不同纹理,所需要时间也不一样。本文算法合成的效果图如图 1:

图 1(a)中,通过目标图灰度化后,进行域值分割,将绿色的作为背景,而将白色和红色的作为前景,因此草地作为背景利用块拼贴合成,而红色的区域比较窄小,故夹杂着许多花瓣。

图 1(b)中,合成图不管背景和前景都十分清晰,背景是结构性强的纹理,利用快拼贴避免纹元错位,模糊等问题,而前景半结构半随机输入合成的,由于 Ashikhmin 对随机纹理效果很好,所以合成图的效果比较理想。

图 1(c)中,由于通过域值算法得到的域值不是很理想,3 朵白花几乎被覆盖了,但是合成效果还是令人满意。



(a)603*453, 192*192, 498*498, 498*498



(b)128*128, 128*128, 530*353, 530*353



(c)201*201, 258*258, 516*516, 516*516



(d)603*453, 192*192, 720*524, 720*524

图中从左到右依次为背景输入纹理、前景输入纹理、目标图、合成图。

图1 约束双样图合成

图1(d)中,由于目标图是黑白纹理设计图,不存在域值分割的问题,并且有些块较大,这样利用Ashikhmin算法合成前景,就出现取点取到边界之外的问题,此时随机取点,造成了某些部分的前景模糊,

但是总体效果还是不错的。

在前景的合成,对于前景区域大小形状都没有任何限定,因此纹理目标设计图可以随心所欲,合成的纹理结果图可以丰富多彩。

上述实验结果表明本文算法是有效的,并且对于许多纹理的合成取得了不错的效果。

5 结论

① 合成纹理图案效果较好。

本文算法将 Efors-Freeman 块拼贴纹理合成方法和 Ashikhmin 点匹配纹理合成方法想结合,具有块拼贴算法的优势,减少了背景图的纹元错位、条带化和模糊等现象,而 Ashikhmin 对于自然纹理和随机纹理的合成效果不错,最终使得合成图的效果比较好。

② 纹理目标设计图完全不受限制。

利用本文的合成算法,任何复杂的设计图案、或者照片,都能用来进行带约束的多样图纹理合成。

③ 合成速度上有优势。

背景图由于采用了块拼贴纹理传递,已经达到实时,合成算法的瓶颈就在 Ashikhmin 算法本身,而 Ashikhmin 算法采用了相关性原理,因此和输入纹理大小几乎无关,因此在合成速度上,还是令人相当满意。

④ 算法仍存在缺陷。

一些前景及一些前景的部分合成算法是 Ashikhmin 算法,当前景是合成结构性纹理的时,效果不是很好。还有当背景占的比例不是很大的时候,

或者前景各个样图对应的合成区域很分散或者较小的时候,合成速度受到了 Ashikhmin 算法影响。最终边界的平滑处理,若边界两边的样图纹理差距较大,平滑效果并不是很好,这些是需要进一步改进的地方。

参考文献

- 1 Bar Joseph Z. Statistical learning of multidimensional textures. Department of Computer Science, Hebrew University, 1999.
- 2 徐晓刚,鲍虎军,马利庄.基于相关性原理的多样图纹理合成方法.自然科学进展,2002,12(6):665-668.
- 3 徐晓刚.纹理合成技术研究.浙江大学博士后研究工作报告,2001.
- 4 Efors A, Freeman WT. Image quilting for texture synthesis and transfer. Proc. SIGGRAPH,2001,(8):341-346.
- 5 Ashikhmin M. Synthesizing natural textures. ACM Symposium on Interactive 3D Graphics,2001,(3):217-226.
- 6 Wei LY, Levoy M. Fast texture synthesis using tree-structured vector quantization. Proc. of the SIGGRAPH 2000. New Orleans.