第37卷 第14期 计 算 机 工 程 2011年7月
Vol.37 No.14 Computer Engineering July 2011

·开发研究与设计技术· 文章编号：1000—3428(2011)14—0262—03 文献标识码：A 中图分类号：TP391.41

曲面渐变纹理合成中重采样问题研究

聂俊岚，赵静萍，王艳芬

(燕山大学信息科学与工程学院，秦皇岛 066004)

摘 要：在渐变纹理合成过程中，纹理尺度和方向的变化需要不断对原样图进行重采样。为避免不断重采样对纹理合成速度的影响，在基于块合成方法的基础上，以线性函数作为纹理尺度的过渡函数，在二维空间生成渐变纹理，将二维渐变纹理作为新的样本，利用矢量场控制曲面纹理合成，采用灰度图像辅助合成，以避免纹元破裂。实验结果表明，该方法对自然界中大多数纹理进行合成时能取得较好的效果，能保持纹理的视觉连续性，合成效率较高。

关键词：纹理合成；曲面纹理合成；渐变纹理；线性函数；灰度图像

Research of Resample Problem in Progressively-variant Texture Synthesis on Surfaces

NIE Jun-lan, ZHAO Jing-ping, WANG Yan-fen

(Institute of Information Science and Engineering, Yanshan University, Qinghuangdao 066004, China)

【Abstract】In the process of progressively-variant texture synthesis, because of the change of texture scale and direction, it needs constant resampling. In order to avoid resampling and accelerate the speed of texture synthesis, this paper adopts linear function as the transition function of texture scale, first of all, progressively-variant texture is synthesized in two-dimensional space, then uses this new texture sample and the control of vector to synthesize a progressively-variant texture on a 3D mesh. In order to prevent texture elements break apart, it adopts gray-scale image to assist synthesizing. Experimental results show that the method can produce smooth texture in most nature texture synthesis and obtain satisfactory results in synthesis efficiency.

【Key words】texture synthesis; texture synthesis on surfaces; progressively-variant texture; linear function; gray-scale image

DOI: 10.3969/j.issn.1000-3428.2011.14.089

1 概述

在计算机图形学中，经常用纹理表现几何模型的表面细节。基于样图的纹理合成(Texture Synthesis from Samples, TSFS)是近年来计算机图形学领域的研究热点之一。

TSFS技术可分为二维纹理合成、曲面纹理合成和视频纹理合成。在曲面纹理合成中常用的方法为：基于像素点的纹理合成和基于块的纹理合成。为减少纹理生成过程中误差累积的问题，文献[1]提出使用最大期望值求解全局纹理合成能量函数得到最优化纹理合成结果。传统的纹理合成方法研究的大多为纹元大小和走向一致的纹理，但在现实中大多数纹理都呈变化性的规律。针对这一特征，文献[2-3]分别提出了尺度和走向不同的纹理合成方法。前者采用基于向量场的变形和基于特征的变形2种方法实现纹元的变化；后者采用雅克比行列变换，对纹理进行局部旋转和变形。两者均在基于像素点合成方法的基础上进行纹理合成，对于每一个待合成像素，当纹理尺度和方向发生变化时，需要进行重采样。文献[4]在基于块合成算法的基础上提出一种用户可控的渐变纹理合成方法，与基于像素点的合成方法相比，效率有所提高，但在纹理尺度发生变化时仍然需要重采样，即对每一个待合成的三角形，均需按照其矢量尺度对样本纹理进行缩放。由于三维模型像素点或三角面片数量非常多，不断地重采样会影响纹理的合成效率，因此本文提出一种更高效的渐变纹理合成方法。

2 基于块的合成方法

基于块的合成方法具有高效性和结构稳定性的优点，本文采用块合成算法进行纹理合成。以三角面片为单位，在三维模型表面生成带有方向和尺度的矢量场，其中，矢量的方向为纹理的走向；矢量的尺度表示纹理的尺度，通过矢量场方向和尺度的变化控制纹理的走向和尺度。为了避免曲面渐变纹理合成过程中不断重采样的问题，首先根据矢量尺度对矢量场进行聚类分析，然后在线性函数指导下在二维目标区域生成渐变纹理，并以新合成的渐变纹理作为曲面纹理合成的新样本，进行曲面渐变纹理合成。要完成本文算法，需要解决2个问题：(1)如何在传统块拼接算法的基础上实现二维渐变纹理的合成；(2)如何将二维渐变纹理的尺度变化恰当地反映到曲面上，防止纹元的破裂。

3 渐变纹理合成

3.1 预处理

为了使合成的纹理更加光滑连续，先采用Delaunay剖分算法，将网格剖分成大小较均匀的三角面片，然后在三维模型表面生成矢量场控制纹理的走向和尺度。方法为：由用户指定 N 个三角面片的切矢量(称为种子三角形)，余下三角形上的切向量通过“种子三角形”的矢量加权插值法得到。

对于大多数纹理(如动物的皮毛)，纹理尺度在整体上呈现变化，但在一定的局部范围是不变的，通常可将纹理尺度分为大、中、小3类(有些复杂的模型可进一步细分)。首先

基金项目：国家部委基金资助项目

作者简介：聂俊岚(1962－)，女，教授、博士，主研方向：虚拟现实，计算机仿真；赵静萍，硕士研究生；王艳芬，助理实验师

收稿日期：2010-12-12 **E-mail：**niejll3@163.com

对矢量场进行 k_means 聚类分析，将矢量场分为 3 类，分别对应于纹理尺度的大、中、小。根据聚类分析后的结果，用双线性插值算法对样本纹理进行适当的一次放大和缩小，得到 R_a、R_b、 R_c 3 幅图像，如图 1 所示。



(a)原样图 R_a　　(b)放大后的样图 R_b　　(c)缩小后的样图 R_c

图 1　大中小 3 类纹理尺度输入样图

3.2　二维渐变纹理合成

在二维目标区域，以 R_a、R_b、R_c 为样本纹理，合成一个从 R_b 向 R_c 过渡的 2D 渐变纹理 R_s。本文采用线性函数作为纹理尺度的过渡函数指导渐变纹理合成。假定原样图 R_a 对应于单位区域[0,1]×[0,1]，现在[0,k]×[0,k]的区域合成目标纹理为 R_s。步骤如下：

(1)确定纹理尺度的过渡函数。设 S_a、S_b、S_c 为 R_a、R_b、R_c 对应的纹理尺度，为了使目标纹理 R_s 在 $y\approx k$ 时接近于 R_b，在 $y\approx 0$ 时接近于 R_c，给定纹理尺度的线性过渡函数为：

$$F(y)=\frac{S_b-S_c}{k}y+S_c \tag{1}$$

当 $y\approx k$ 时，纹理的尺度为 S_b，当 $y\approx 0$ 时，纹理尺度为 S_c，用户可通过调整比例因子 k 的大小控制目标纹理的尺寸。

(2)确定待合成纹理块的候选样图。假设 B_k 为待合成块，R_a、R_b、R_c 为 B_k 的候选样图。候选样图的选取由 B_k 的纹理尺度决定。所以，先根据给定的纹理尺度的过渡函数计算出待合成块 B_k 的纹理尺度。方法为：设 y_1、y_2 为 B_k 在目标区域中上下边界对应的纵坐标值，分别将 y_1、y_2 带入式(1)中，求得 $F(y_1)$和 $F(y_2)$。取 $F(y_1)$和 $F(y_2)$的均值 $F(B_k)=(F(y_1)+F(y_2))/2$ 为待合成块的纹理尺度。其中，$|F(B_k)-S_a|$、$|F(B_k)-S_b|$、$|F(B_k)-S_c|$ 的最小值所对应的纹理尺度 $S_n(n=a,b,c)$与 B_k 的纹理尺度最接近。与 S_n 对应的 R_n 便为待合成块 B_k 的候选样图。

(3)寻找最佳匹配块和边界融合。在确定 B_k 的候选样图后，根据 B_k 的边界邻域特征在它的候选样图中寻找最佳匹配块。基于块的合成方法与 2 个参数密切相关：待合成块的尺度和边界邻域宽度。本文采用文献[5]提出的最优参数估计方法确定 2 个参数的值。最后用 Graphcut 算法[6]对边界重叠区进行处理。合成效果如图 2 所示，在竖直方向上，纹理尺度自上而下逐渐减小。



图 2　二维渐变纹理合成效果

3.3　曲面渐变纹理合成

以 2D 渐变纹理为新样本进行曲面渐变纹理合成。如前所述，用户可以通过调整比例因子 k 的大小控制二维渐变纹理的尺寸，尺寸越大，曲面纹理的合成效果越好，但同时消耗的时间也越多。在实际应用中用户可根据自己的需求确定 k 的大小。为保证将 2D 渐变纹理的尺度变化恰当地反映到曲面上，保持纹理的连续过渡，本文做了以下处理：

(1)满足“纹理延伸”条件的三角形合成

通常相邻的三角面片在方向和尺度上变化很小，对于约束度 C=1 的三角形面片，采用“纹理延伸”的方法[7]，可以保证 2 个三角面片之间的纹理平滑过渡，同时减少纹理的合成时间。要实现三角块的“纹理延伸”操作，需要保证延伸后的纹理坐标仍然落在样本纹理中。在该步骤中是以较大尺寸的 2D 渐变纹理为样本纹理，使较多 C=1 的三角形满足这一条件。

(2)不满足“纹理延伸”条件的三角形合成

对于约束度 C>1 和 C=1 纹理延伸后越界的三角形，采用梯形模板匹配的曲面纹理合成方法[8]。为达到渐变效果，本文在寻找最佳匹配之前，先确定当前待合成三角形 T_c 在 2D 渐变纹理中的候选空间 $G(T_c,\varepsilon)$，候选空间中的三角形 t_c 需满足条件：

$$|Fo(t_c)-F_s(T_c)|<\varepsilon \tag{2}$$

其中，Fo 为纹理尺度的过渡函数；F_s 为待合成三角形 T_c 的矢量尺度；ε 为尺度的匹配误差。接下来在候选空间中搜索待合成三角形的最佳匹配块。

为了防止渐变纹理合成过程中纹元的破裂，本文在寻找最佳匹配时采用灰度图像进行辅助合成。步骤为：

(1)提取 2D 渐变纹理的灰度图像 T_H 和待合成三角形梯形检测模板所对应的灰度图像 $H(T_c)$；

(2)因为灰度图像能更好地体现纹理的结构特征，在寻找最佳匹配时，先在 TH 所对应的候选空间中寻找与 $H(T_c)$相似的三角形 $H(t_c)$；

(3)在 2D 渐变纹理的彩色图像中，对具有相似 $H(t_c)$的三角形进一步筛选，使其相应的外部梯形边界区域与检测模板的误差最小。将得到的纹理块作为待合成三角形的目标纹理。实现算法如下：

```
For each triangle Tc on surface     // Tc 为待合成三角形
Extract test template N(Tc)
Bool   NG        //判断当前三角形是否满足纹理延伸条件
if(NG=true)：do texture extention
else：
Extract Grayscale test template H(Tc )
Build candidate pool G(Tc, ε)
Smallest_match=INFTY
For each triangle tc in G(Tc, ε)
Extract Grayscale test template H(tc) and test template N(tc )
New_match=dist((N(Tc), N(tc)) +( H(Tc), H(tc)))
// N 为彩色空间，H 为灰度空间
If(New_match<(Smallest_match))
Smallest_match=New_match;
Ts(Tc)=To(tc) // Ts 为目标区域，To 为样本区域
```

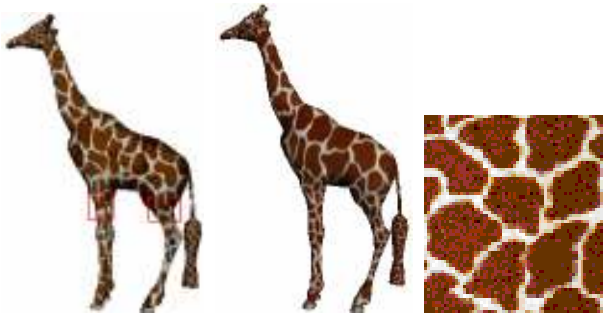
4　实验结果与分析

本文在 P4 2.4 GHz、内存 2 GB 的 PC 上使用 VC++、OpenGL 实现了该算法。在进行曲面纹理合成时，需先合成二维渐变纹理作为曲面渐变纹理合成的样本。文献[9]提出一种为每一个待合成点建立一个面向多分辨率样本图像的正向

映射，并采用以灰度图控制要合成的纹理尺度实现二维纹理尺度变化的方法。

与文献[9]相比，本文算法采用基于块的方法进行合成，用线性函数控制纹理尺度，与基于点的方法相比，基于块的方法可以有效地提高纹理的合成效率，更好地体现纹理的结构特征。

文献[4]算法是采用块合成方法实现纹理尺度变化的典型，其主要思想是：对于每一个待合成三角形，按照其矢量尺度对样本纹理进行缩放，在缩放后的样图中寻找最佳匹配实现纹理尺度变化的效果。在三维模型中，对于体形变化较大的区域，相邻 2 个三角面片的矢量尺度相差较大。对于结构性较强的样本纹理，采用文献[4]的方法，按照相邻三角形之间的矢量尺度比例对样本纹理进行缩放后，很难在样本空间中找到待合成三角形的最佳匹配块，从而引起过渡区域的不连续现象，如图 3(a)所示。本文采用已合成的二维渐变纹理作为输入样图，先根据待合成三角形的矢量尺度，在二维渐变纹理中寻找与待合成三角形纹理尺度所对应的过渡区域，然后在过渡区域中寻找最佳匹配，实现了纹理从大尺度到小尺度的平滑过渡。并且在寻找最佳匹配时，用灰度图像进行辅助，更好地体现了纹理的结构特征(图 3(b))，从而改善了不连续现象。



(a)文献[4]合成效果 (b)本文算法合成效果 (c)样本纹理

图 3 本文算法与文献[4]合成效果对比

本文算法与文献[4]算法合成时间的对比结果如表 1 所示，其中，二维渐变纹理合成时，k 取值为 2，平均合成时间为 1.8 s。

从表 1 可以看出，对于网格数越多的模型，本文算法与文献[4]算法相比可以节约更多的时间。原因是本文算法避免了为实现纹理尺度的改变而对样本纹理进行不断缩放后再重采样的繁琐过程。

此外，文献[4]算法采用梯形模板匹配方法进行纹理合成，本文算法对约束度为 1 的三角形采用“纹理延伸”的方法，减少了寻找最佳匹配的时间，从而进一步减少了纹理的合成时间。

表 1 本文算法与文献[4]算法合成时间对比

模型	网格数	本文算法合成时间/s	文献[4]合成时间/s	节约时间/s
茶壶	1 024	16.4	28.4	8.0
兔子	1 500	22.8	32.2	10.6
奶牛	3 000	52.6	98.3	45.7
长颈鹿	4 800	63.4	126.5	63.1

5 结束语

本文在基于块合成方法的基础上，提出一种以线性函数为指导的曲面渐变纹理合成方法。该方法计算简单，避免了大量的重采样工作，提高了合成效率且便于用户控制。实验结果表明，本文算法对自然界中存在的大多数纹理进行合成时都取得了较好的效果。针对现实世界中各种各样的纹理和复杂多变的三维模型，如何实现更加丰富和多变的纹理效果，进一步提高纹理的合成效率，在虚拟仿真学科领域具有重要的意义，是下一步工作的重点。

参考文献

[1] Kwatra V, Essa I, Bobick A, et al. Texture Optimization for Example-based Synthesis[J]. ACM Transactions on Graphics, 2005, 24(3): 795-802.

[2] Zhang Jingdan, Zhou Kun, Velho L, et al. Synthesis of Progressively-variant Textures on Arbitrary Surfaces[J]. ACM Transactions on Graphics, 2003, 22(3): 295-302.

[3] Lefebvre S, Hoppe H. Apperance-space Texture Synthesis[J]. ACM Transactions on Graphics, 2006, 25(3): 541-548.

[4] 汤 颖, 孙汉秋, 张宏鑫, 等. 用户控制的纹理合成[J]. 计算机辅助设计与图形学学报, 2004, 16(10): 1412-1418.

[5] Narkdej J, Kanongchaiyos P. An Efficient Parameters Estimation Method for Automatic Patch-based Texture Synthesis[C]// Proceedings of the 6th International Conference on Computer Graphics. [S. l.]: IEEE Press, 2009: 91-99.

[6] Kwatra V, Schodl A, Essa I, et al. Graphcut Textures: Image and Video Synthesis Using Graph Cuts[J]. ACM Transactions on Graphics, 2003, 22(3): 277-286.

[7] 薛 峰, 张佑生, 江巨浪, 等. 基于纹理延伸和三角块拼接的快速曲面纹理合成[J]. 计算机辅助设计与图形学学报, 2007, 19(2): 221-226.

[8] Wu Fuli, Mei Chunhui, Shi Jiaoying. Method of Texture Synthesis on Arbitrary Surfaces[J]. Journal of Computer Science & Technology, 2005, 19(5): 643-649.

[9] 李大锦. 可控的连续多尺度纹理合成[J]. 计算机工程, 2009, 35(24): 211-212, 215.

编辑 张正兴

(上接第 261 页)

[7] Yu Yongguang, Zhang Suochun. Hopf Bifurcation in the Lü System[J]. Chaos Solitons & Fractals, 2003, 17(5): 901-906.

[8] Tigan G, Constantinescu D. Heteroclinic Orbits in the T and the Lü Systems[J]. Chaos Solitons & Fractals, 2009, 42(1): 20-23.

[9] Chen Dilan, Sun Jitao, Wu Qidi. Impulsive Control and Its Application to Lü's Chaotic System[J]. Chaos Solitons & Fractals, 2004, 21(5): 1135-1142.

[10] Li Damei, Lu Junan, Wu Xiaoqun, et al. Estimating the Bounds for the Lorenz Family of Chaotic Systems[J]. Chaos, Solitons & Fractals, 2005, 23(2): 529-534.

[11] 刘明华, 冯久超. 一个新的超混沌系统[J]. 物理学报, 2009, 58(7): 4457-4460.

[12] 周 晟, 唐驾时. 一个新超混沌系统的自适应反同步[J]. 噪声与振动控制, 2009, 29(1): 14-15.

编辑 顾姣健