

# 基于笔划绘制方法研究概述\*

## A Summarize of Stroke-Based Rendering Research

蒋凌琳, 张显全

JIANG Ling-lin, ZHANG Xian-quan

(广西师范大学计算机科学系, 广西桂林 541004)

(Department of Computer Science, Guangxi Normal University, Guilin, Guangxi, 541004, China)

**摘要:**概述非真实感图形绘制技术的基于笔划绘制方法(SBR)的原理以及 SBR 的优化算法和贪婪算法, 其中优化算法通过反复的绘制笔划然后调整它们的位置以使目标函数值最小, 贪婪算法通过不停的绘制笔划使其与目标图像相匹配。SBR 的研究才刚刚起步, 还需要更深入的研究和探讨。

**关键词:**图形绘制 非真实感图形 优化算法 贪婪算法

中图分类号: TP391 文献标识码: A 文章编号: 1002-7378(2006)02-0136-05

**Abstract:** Describe the theory of Stroke-Based Rendering (SBR) of Non-Photorealistic Rendering, and introduce the tow approaches to designing SBR algorithms, Greedy algorithms and Optimization algorithms. Greedy algorithms greedily place the strokes to match the target goals, and optimization algorithms iteratively place and then adjust stroke positions to minimize the objective function. The research of Stroke-Based Rendering is just beginning, and in many place we need to study and discuss more deeply.

**Key words:** graphics rendering, Non-Photorealistic Rendering, optimization algorithms, greedy algorithms

近年来非真实感图形绘制技术得到了广泛的应用, 许多学者提出了不同绘画风格的算法, 如油画<sup>[1,2]</sup>、水粉画<sup>[3]</sup>、钢笔画<sup>[4~7]</sup>、马赛克镶嵌画<sup>[8,9]</sup>、圆点画<sup>[10~13]</sup>、流线视觉效果以及矢量域视觉效果<sup>[14]</sup>等。通过绘制离散的元素(如绘画的笔触或圆点)来自动生成非真实感图形的方法称为基于笔划绘制方法(SBR)<sup>[15]</sup>。对于 SBR 算法设计, 可分为优化算法和贪婪算法, 优化算法反复的绘制笔划然后调整他们的位置直到满意为止, 贪婪算法不停的绘制笔划使其与目标图像相匹配。所有的 SBR 算法基本上都依照两个原则来绘制笔划生成图像, 一是在一定程度上限制笔划的数量, 使结果看上去像是一幅图画而不是照片, 二是使生成的画面看上去与输入的图

像相似。本文对基于笔划绘制方法的研究内容进行综述, 并指出 SBR 算法存在的问题。

### 1 基于笔划的绘制<sup>[15]</sup>

在进行 SBR 绘制之前, 先要确定笔划的外观。笔划模型通过参数控制, 不同的参数设置会产生不同的笔划位置和外观。以圆点这一简单的笔划模型为例, 可由两个参数来描述它: 位置由圆心  $(x, y)$  坐标来确定, 大小由半径  $r$  来确定, 笔划确定好之后, 可在一个画板上将其进行组合从而创作出一幅图画。画板参数包括定义了颜色或纹理的画布和设置好的笔划排列表, 创建图像的过程就是在画板上按顺序绘制笔划的过程。使用 SBR 能量函数来作为目标函数评价绘制方法的好坏, SBR 能量函数  $E: I \rightarrow R$ , 其中  $I$  是可能的图像结构的集合,  $R$  是实数集。  $E(I)$  把图像作为输入端, 然后输出一个指代图像质量的数值。SBR 算法的目标就是用最小的能量来生成一幅图像, 表示为:

$$E(I) = E_{match}(I) + \tau \omega_{abs} E_{abs}(I),$$

收稿日期: 2005-05-23

修回日期: 2005-09-06

作者简介: 蒋凌琳(1977-), 女, 广西桂林人, 在读硕士研究生, 主要从事计算机图形学研究。

\* 广西自然科学基金(0447035)资助。

$$E_{match}(I) = \sum_{(x,y) \in I} \|I(x,y) - S(x,y)\|^2,$$

$$E_{abs}(I) = I \text{ 中的笔划数,}$$

上式中  $w_{abs}$  是权重参数,  $E_{match}(I)$  是源图像与绘制结果的方差和。调整参数  $w_{abs}$  的值可以控制绘画风格抽象的程度,  $w_{abs}$  取较小值时可得到逼真的绘制效果,  $w_{abs}$  取较大值时就可以得到抽象的绘制风格。

## 2 SBR 算法

### 2.1 优化算法

SBR 使用两种优化算法, 一种叫 Voronoi 算法, 它研究 SBR 问题的特殊性质, 采用有效的循环更新步骤来解决问题; 另一种叫反复试验算法, 采用一般结构和启发式选择测试来减少能量。通常说来 Voronoi 算法效率高, 但不能解决所有的问题, 而反复试验算法可以解决所有的问题, 却需要较多计算时间。

#### 2.1.1 Voronoi 算法<sup>[8]</sup>

在基于笔划的绘制生成的图像过程中包含许多相同但不重叠的笔划, 且笔划的排列密度不自然, 使用 Voronoi 算法能有效的解决这一问题<sup>[8]</sup>。它采用的基本原理是用计算几何中的技术在图像中均匀地绘制笔划, 并且可以用图形硬件来加速。然而, 由于这一算法的能量函数是按照笔划的密度来定义的, 从基于图像的衡量标准来看(即绘制结果应与目标色调一致)它并没有直接做到最优化, 还可以继续对其进行优化。

Voronoi 算法过程采用迭代最优化程序在图像中生成一个均匀的空间点集, 并为这一程序定义一个适当的能量函数。设  $P = (x, y)$  为像素的位置,  $C_i$  为矩心点的位置,  $L_p^i \in \{0, 1\}$  为像素的二元标志, 如果  $L_p^i = 1$ , 则像素  $p$  将被分配到矩心  $i$  上, 其能量函数为:

$$E(I) = \sum_{p \in I} L_p^i \|p - C_i\| = \sum_{p \in I} L_p^i ((p_x - C_x)^2 + (p_y - C_y)^2).$$

矩心和标志都是图像  $I$  的内含成员, 从上式中可看到, 算法的目标是选择矩心集和标志使能量函数最小, 像素越接近矩心所消耗的能量越小。如果预先知道矩心集  $C_i$ , 那么计算最优标志就会容易些, 只需把每一个像素赋予与之最近的矩心即可, 最终得到的标志称为 Voronoi 图(如图 1a), 它依照平面上每一点最近的矩心  $C_i$  来划分平面。

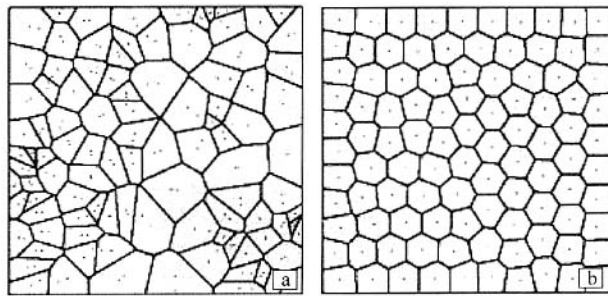


图 1 Voronoi 图

a. Voronoi 图; b. 优化后的 Voronoi 图

在图 1a 中, 随机地选择点集然后计算 Voronoi 图并不能得到一个满意的矩心排列结果, 但可以通过调整中心点来改善矩心集, 使其更好的进行分割, 即用矩心来修改标记和优化能量函数, 矩心优化公式为:

$$C_i = \sum_p L_p^i p / \sum_p L_p^i,$$

$C_i$  是其周围像素位置的平均值; 它可以通过计算  $\partial E(I) / \partial C_i = 0$  求解获得  $C_i$ 。图 1b 显示了同样的点优化后的结果。在实际应用时, 对一幅完整的图像运行这个优化程序时速度慢, 但可使用图形硬件进行加速<sup>[8,16]</sup>。

文献[12]提出一种从图像生成点画的随机分布快速算法和策略, 该算法定义了点(stipple)的模型, 模型以参数化方式提供, 并提供了多样的密度分布函数, 既为点本身提供多样的变化, 也使最后生成的点画风格多样, 具有更多的艺术效果; 同时制定的随机分布策略, 能够保证迅速的生成点画, 实现高质实时的点画效果, 并且提出三维实体模型绘制点画的算法和策略, 可以生成带有三维效果的点画。文献[13]提出的点画绘制则利用了三维几何空间的信息, 将二维信息与三维物体空间结合起来, 生成有立体感的点画图像。

#### 2.1.2 反复试验算法<sup>[17]</sup>

Voronoi 方法的不足之处在于它不能处理颜色信息及笔划重叠问题, 目前, 能解决这些问题的唯一方法是反复试验算法, 理论上反复试验算法可以解决所有的 SBR 问题。反复试验算法方法简单, 通过直接对图像的结构进行变形, 如果变形能够减少能量, 则将变形并入图像结构当中, 否则将变形抛弃, 然后重复算法。这一算法最关键的部分是提出一个好的变形机制, 纯随机的变形机制只能事倍功半, 而采用手工调节的方法可以很快得到预期的结果。由于反复试验算法和贪婪算法都需要人工参与设计, 因而它们所消耗的时间是接近的。不同在于, 反复试验算法可通过检查确定变形方案是否能提高图像的

质量,并且这一过程可反复的改进先前绘制的笔划。图2为使用该算法得到的实例,它采用的方法都是随机扰乱一定数量的笔划,直到由矩形组成的图像与源图的差别减少到指定要求为止。



图2 用反复试验算法生成的图片

反复试验法常用于美术绘制,这一算法搜索与源图相匹配的简练画法,尽量采用最少的笔划进行绘制,并将笔划覆盖源图。笔划由笔刷半径及控制点表确定,其能量函数为:

$$E(I) = E_{app}(I) + E_{nstr}(I) + E_{cov}(I),$$

$$E_{app}(I) = \sum_{(p) \in I} \omega_{app}(p) \| I(p) - S(p) \|,$$

$$E_{nstr}(I) = \omega_{nstr}(I \text{ 中的笔划数}),$$

$$E_{cov}(I) = \omega_{cov}(I \text{ 中的非空像素点}),$$

在上面的式子中能量是三个函数线性组合而成,第一项  $E_{app}$ ,从像素上衡量源图  $S$  与绘制结果的不同;第二项  $E_{nstr}$  与笔划数相关,用于控制减少笔划的数目;第三项覆盖项  $E_{cov}$ ,确定画布被笔划完全填满,如果要求达到,将  $\omega_{cov}$  设置为一个大的数值;权重值  $\omega$  是由用户设定的,度量参数  $\omega_{app}(p)$  由度量图像来定义。能量函数的前两项用来平衡两个相互排斥的条件:既要用尽可能少的绘制笔划,又要尽量与源图的外观相匹配。用户可通过调整  $\omega_{app}$  和  $\omega_{nstr}$  的相关比例来指定这两方面相应的重要程度,从而生成不同的绘制效果。默认情况下,  $\omega_{app}(p)$  的值由一个二元的边缘图像来初始化。如果度量值在画布上不断变化,那么在图像的不同部分,可能会得到能量函数不同而效果相似的结果(如图3所示),度量图  $\omega_{app}(p)$  可让用户指定在图像的不同区域所需要显示细节的程度,度量图像可自动生成,也可以用手工来绘制,这能让用户在更高层次上控制生成的图像,而无需做一些低层选择。

## 2.2 贪婪算法

在贪婪算法中笔划按照单一途径添加到图像结构中,并且笔划生成后就不再修改。贪婪算法采用谨慎而有启发性的方法设计笔划绘制步骤,能很快生成高质量的结果,但是缺乏灵活性。在贪婪算法中虽然有时能量函数可能存在,但由于设计一个适当的能

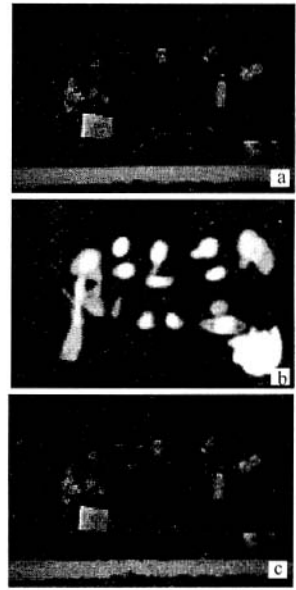


图3 反复试验法绘制的图像

a. 源图;b. 权值图;c. 由权值生成的图像

量函数会比较困难,而且一个有效的算法无需能量函数也会工作得很好,因而贪婪算法很少由能量函数来定义。

### 2.2.1 单点笔划

单点笔划是一个简单的半自动绘图算法,首先由用户提供一幅源图像,然后从空白画布开始进行绘制。用户使用鼠标或绘图笔在绘图区点击或拖动,在每次鼠标点击的位置上绘制一个笔划,系统自动根据源图像在对应点的色彩提取出笔刷的颜色,并由图像的梯度曲线确定笔划方向,这样用户决定笔划的位置而算法决定笔划的外观,使用者可通过调整设置或改变画板接触面的压力来改变其它的一些参数(如笔划的大小),系统为用户提供了一个有趣并且简单的绘画工具,并无需使用者有任何绘画技巧。为了模仿艺术家进行绘画创作,可采用多层美术绘制算法<sup>[1]</sup>,它能生成大小不同的笔划。在绘制过程中从粗略图开始,然后在此基础上使用更小的笔刷来添加细节,最终完成绘画的创作,这种算法能够不断地优化能量函数,减少图画与源图间的差别和使用的笔划数。

算法分为三步,第一步首先输入一幅源图像和一个画刷尺寸表,画刷尺寸由半径  $r_1, r_2, \dots, r_n$  表示。第二步生成一系列的图层,每一图层对应一个笔划半径,半径按照由大到小的顺序生成,一般采用  $r_i = r_1 * 2^{i-1}$  来计算笔划半径。第三步通过对源图像进行模糊,生成每一层的参考图像,参考图像与当前图层的笔刷大小相对应,即可用当前笔刷来捕捉图像中与其尺寸相同的画面细节。在第三步中可以使

用一个图层子程序来使画刷  $r_i$  在参考图像的基础上绘制相应的图层,程序的方法是先定位到画布上与参考图像不同的区域,用新的笔刷覆盖它们,当画布中的颜色与参考图像的颜色在阈值( $T$ )的范围内匹配时,则停止修改,增大阈值参数可生成较粗糙的图像,减少阈值则会生成与源图较一致的图像。

该算法将注意力放到图片上包含细节信息最多(高频信息)的区域,并在这些区域中绘制了许多细小的笔划,包含细节信息较少的区域则用大一些的笔触来绘制。这样,笔划可以与源图细节层次相适应。

文献[2]的有立体感的图像绘制将多层笔划绘制算法应用到了立体图像的绘制上,最终生成有立体感的包括不同细节层次的图像,生成的效果图如图4所示。



图4 有立体感的多层笔划绘制效果

于金辉等[3]提出一个三维计算机水粉笔划模型,该模型包括笔刷单元选取、颜料模拟、颜料扩散以及整体控制四部分。该模型结构简单并可以直接应用到传统三维曲面上生成手工绘制水粉画的效果,如图5所示,虽然模型结构十分简单,却能很好地再现水粉效果,并且不需要生成参考图像,也不需要进行繁杂的流体模拟,从而大大地减少了计算量。



图5 三维计算机水粉笔划模型生成的图像

### 2.2.2 长曲线笔划

一些绘画或制图使用长曲线笔划来代替短的笔划,最典型采用长曲线笔划来进行绘制的方法是钢笔画法,许多类型的钢笔插图都采用贪婪算法来优化笔划的效果。最简单的算法是直接沿着指定方向绘制笔划来获得需要的笔划密度(这样可得到目

标色调)。Jobard等[6]提出了一种有效的贪婪算法,用笔划间的距离 $d$ 来定义目标密度,算法包括两步:

(1) 确定图像中的种子点与其它所有曲线的距离大于等于 $d$ ;

(2) 在矢量域中从一个种子点开始绘制一条曲线,当这条曲线与其它曲线的距离小于 $d$ 时停止曲线的绘制。

算法反复运行,直至图像被一定密度的笔划所覆盖。这一程序确保能生成沿着矢量域方向的曲线,并保持曲线间的距离至少为 $d$ ,它采用启发式方法来指导对种子点的选择。

文献[18]采用长曲线笔划来生成简单的线条画,他们提出 $G$ 笔划这一概念,通过特性参数和风格参数来控制线条的样式生成最后的结果。

## 3 结束语

基于笔划的绘制方法的研究才刚起步,到目前为止,大部分SBR算法都只是在处理相对简单的艺术风格,如简单的印象派绘制等,作为一个新兴的方向,在许多方面还需要更深入的研究和探讨。目前在SBR算法领域仍存在许多的问题,如在交互式应用中大多数算法都太慢而用处不大;在大部分的应用中,用户需要更好的工具来控制生成的风格等。今后需开发一些更复杂更有效的艺术风格,如运用SBR算法生成的绘画应用到动画上面去等。

### 参考文献:

- [1] HERTZMANN A. Fast Paint Texture[C]//Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, 2002. ACM Press, 2002: 91-96, 161.
- [2] EFSTATHIOS STAVRAKIS, MARGRIT GELAUTZ. Stereoscopic Painting with Varying Levels of Detail [C]//Proceedings of SPIE-Stereoscopic Displays and Virtual Reality Systems XII, San Jose, California, USA, January 16 - 20, 2005. SPIE/IS&T, 2005: 450-459.
- [3] 于金辉, 徐晓刚, 彭群生. 一个三维计算机水粉笔刷模型[J]. 计算机辅助设计与图形学学报, 2000, 12(9): 664-667.
- [4] MICHAEL P. SALISBURY, MICHAEL T. WONG, JOHN F. HUGHES, DAVID H. SALESIN. Orientable textures for image-based Pen-and-Ink illustration[C]//Proceedings of SIGGRAPH 97 Conference Proceedings, Los Angeles, California, USA, 1997. T. Whitted, Ed., 1997: 401-406.

- [5] GEORGES WINKENBACH, DAVID H. SALESIN. Computer-generated pen-and-ink illustration [C]// Proceedings of the 21st annual conference on Computer graphics and interactive techniques, Los Angeles, California, USA, 1997. ACM Press, 1997: 91-100.
- [6] JOBARD B, LEFER W. Creating Evenly-Spaced Streamlines of Arbitrary Density [C]// Proceedings of the 8th Eurographics Workshop on Visualization in Scientific Computing, Boulogne-sur-Mer, France, 1997. Eurographics Workshop Press, 1997: 43-56.
- [7] HERTZMANN AARON, ZORIN DENIS. Illustration Smooth Surfaces [C]// Proceedings of the 27th annual conference on Computer graphics and interactive techniques, New Orleans, Louisiana, USA, 1999. ACM Press, 1999: 517-526.
- [8] KENNETH E. HOFF III. Fast Computation of generalized voronoi diagrams using graphics hardware [C]// Proceedings of the 26th annual conference on Computer graphics and interactive techniques, Los Angeles, California, USA, 1999. ACM Press, 1999: 277-286.
- [9] ALEJO HAUSNER. Simulating Decorative Mosaic [C]// Proceedings of SIGGRAPH 2001, Los Angeles, California, USA, 2001. ACM Press, 2001: 573-578.
- [10] OLIVER DEUSSEN, STEFAN HILLER, CORNELIUS VAN OVERVELD, et al. Floating points: A method for computing stipple drawings [J]. Computer Graphics Forum, 2000, 19(s): 40-51.
- [11] ADRIAN SECORD. Weighted voronoi stippling [C]// Proceedings of the NPAR international Symposium, Annecy, France, 2002. ACM Press, 2002: 37-43.
- [12] 周亮. 点画实现及其多样化——艺术风格图像渲染 [D]. 大连: 大连理工大学, 2004.
- [13] XIAORU YUAN, MINH X NGUYEN, NAN ZHANG, et al. Stippling and Silhouettes Rendering in Geometry-Image Space [C]// Proceedings of Eurographics Symposium on Rendering, Konstanz, Germany June 29— July 1, 2005. EUROGRAPHICS, 2005: 193-200.
- [14] GREG TURK, DAVID BANKS. Image-Guided streamline placement [C]// Proceedings of SIGGRAPH 96, New Orleans, Louisiana, 1996. ACM Press, 1996: 453-460.
- [15] AARON. HERTZMANN. A Survey of Stroke-Based Rendering [J]. Computer Graphics and Applications, 2003, 23(4): 70-81.
- [16] NEIDER JACKIE, DAVIS TOM, WOO MASON. Open GL programming guide: the official guide to learning open GL [M]. Paris: Addison-Wesley Developers Press, 1997.
- [17] A HERTZMANN. Paint by relaxation [C]// Proceedings of computer graphics international 2001, Hong Kong, China, 2001. IEEE Computer Society Press, 2001: 47-54.
- [18] TOBIAS ISENBERG, ANGELA BRENNECKE. G-Strokes: A Concept for Simplifying Line Stylization [C]// Technical Report: Department of Computer Science, University of Calgary, No 2005-780-11, Canada, April, 2005. Calgary: University of Calgary, 2005.

(责任编辑: 韦廷宗)

(上接第 131 页)

足各级政府对项目管理的需要, 用户反映良好, 正准备向区内各地、市、县政府相关职能部门及国内其它省区推广。

## 5 结束语

本系统充分利用现有网络基础, 采用成熟先进的开发工具和技术手段, 解决了相关政府职能部门传统项目手工管理的各种弊端, 实现了项目资料管理的规范化、科学化, 极大地提高了政府工作效率。同时, 也为下一步政府全面实现办公自动化奠定了

良好的基础。

参考文献:

- [1] GEETANJALI ARORA. C# 专业项目实例开发 [M]. 北京: 中国水利水电出版社, 2003.
- [2] SCOTT SEELY. SOAP: XML 跨平台 Web Service 开发技术 [M]. 北京: 机械工业出版社, 2002.

(责任编辑: 邓大玉)