

考虑局部和整体特征的手绘图案合成

刘东威^{1,2)}, 张俊松^{1,2)*}, 周昌乐^{1,2)}

¹⁾ (厦门大学智能科学与技术系艺术认知与计算实验室 厦门 361005)

²⁾ (厦门大学福建省仿脑智能系统重点实验室 厦门 361005)

(zhangjs@xm.u.edu.cn)

摘要: 为了以用户绘制的小块图案为样本,合成在视觉感受上与样本相近的大面积的新图案,提出一种结合局部和整体特征的手绘图案合成方法.首先提取样本中具有独立视觉意义的图案元素,然后分析图案样本的局部和整体特征,最后使用样本中的图案元素,并以这些特征作为参数来生成新的图案.实验结果表明,该方法所合成的图案在继承样本风格的同时,又不是样本图案的简单重复.与相关方法相比,文中方法较好地解决了手绘图案合成中仅考虑局部或整体特征的问题.

关键词: 人机交互;非真实感图形学;手绘图案;图案合成;遗传算法
中图法分类号: TP391

Stroke Pattern Synthesis Based on Local and Global Features

Liu Dongwei^{1,2)}, Zhang Junsong^{1,2)*}, and Zhou Changle^{1,2)}

¹⁾ (Mind, Art & Computation Group, Cognitive Science Department, Xiamen University, Xiamen 361005)

²⁾ (Fujian Key Laboratory of the Brain-like Intelligent Systems, Xiamen University, Xiamen 361005)

Abstract: In this paper, a new method for the synthesis of stroke patterns is proposed, which considers both local and global features of an input stroke pattern. The proposed method takes a small pattern sketched by the user as an input sample. First the representative elements in the pattern image are extracted from the sample and clustered into classes. Then the local and global features of the sample are analyzed. Finally the clustered elements extracted from the sample are arranged based on these features to form large patterns perceptually similar to the input small sample. The experimental results show that the patterns synthesized by the proposed method inherit the style of the input sample and do not simply repeat the small sample. Compared with the related methods, the proposed method well solves the problem by utilizing both local or global features in the synthesis of stroke patterns.

Key words: human computer interaction; NPR (non photorealistic rendering); stroke pattern; pattern synthesis; genetic algorithm

图案是一种带有装饰意味的花纹或纹样,它由有限种类的图案元素按照一定的拓扑结构排列而成,广泛应用于动漫设计、纺织品、工艺美术品和建筑物上.近年来,随着多媒体技术的不断发展,计算机绘图环境日益完善,方便艺术设计人员创作的数

位板、数位屏和手写笔等绘图设备得到广泛应用,大量图案创作和设计的工作开始在计算机上完成.

手绘图案主要由笔画构成,如图 1 所示,基于笔画的绘制技术是非真实感图形学的重要研究内容^[1],多用来模拟一些传统的绘画效果,如水墨画、

收稿日期:2010-07-09;修回日期:2010-08-24.基金项目:国家自然科学基金(60903129,60975076);福建省自然科学基金(F0910149).
刘东威(1986—),男,硕士研究生,主要研究方向为计算机图形学、认知科学;张俊松(1978—),男,博士,副教授,论文通信作者,主要研究方向为非真实感图形学、艺术认知与计算;周昌乐(1959—),男,博士,教授,博士生导师,CCF高级会员,主要研究方向为人工智能及其应用技术.

点画、钢笔画、油画、版画等。但基于笔画的绘制方法较少考虑图案中的笔画元素间的结构特征, 因此它不适用于合成手绘图案。



图 1 图案示例

纹理合成技术可以依据指定的纹理样本合成出与其相似的大范围纹理^[2], 但是这种技术也不适用于手绘图案的合成。与纹理样本不同, 图案元素往往在知觉上具有独立的意义。例如图 1 中的每朵“花”都可以被作为单独的个体而感知。无论是参数化^[3]的还是非参数化^[4,5]的纹理合成, 主要靠相邻区域的颜色匹配进行图像拼接, 进而合成出大规模的纹理。Liu 等^[6]借助 2 幅样本纹理图像从一幅纹理图像提取纹理元素分布特征, 从另一幅纹理图像中提取纹理元素, 综合二者合成一幅新的纹理图像, 该方法适合处理纹理元素在纹理图像中呈重复出现的情况。

近几年, 图形学界开始关注矢量手绘图案合成技术^[7,9]。如 2006 年由 Barla 等^[7]首次以一个小尺寸的手绘图案作为样本, 将样本中具有接近或重叠关系的笔画看作图案元素, 通过分析得到元素之间的局部相邻关系; 然后随机生成一些种子点, 用 Lloyd 方法调整种子点使其分布均匀; 再从中心的种子点开始, 依照样本中元素的局部拓扑关系逐一将每个种子点替换为样本中的一个图案元素; 最后生成大规模图案。该方法从样本中提取出图案的风格特征, 试图合成与样本图案风格上类似的图案。但是, 由于其合成过程只考虑元素相邻关系这种局部特征, 因此合成的图案与样本图案只具有局部的相似性, 而没有整体的相似性, 比如合成图案中各元素的比例不能保证与样本相近。Ijiri 等^[8]也提出了类似的方法, 但合成的图案与样本图案同样只具有局部相似性, 忽略了整体相似性。

近来, Hurtut 等^[9]也研究了基于样本的手绘图案合成。首先由用户绘制样本, 并标注出哪些笔画属于同一个图案元素; 然后依据图案元素的形状特征将其聚类, 并统计各类元素出现的密度; 最后按照样本中的元素密度随机地放置各类元素, 得到合成的图案。由该方法合成的图案与样本具有部分整体相似性, 但是由于不考虑各类图案元素的空间位置关

系, 此方法只适用于不存在明确的内部结构特征的图案合成。

人类的视觉机制研究表明, 人脑对视觉信息的处理过程中同时考虑场景的局部特征和整体特征^[10]。虽然不同流派对于这两方面特征在认知过程中的先后顺序还存在分歧, 但是各种理论普遍认为场景的局部和全局特征都对人脑的认知过程产生重要的影响。本文提出一种兼顾局部特征和整体特征的手绘图案合成算法, 以用户绘制的小规模图案样本为输入, 通过对样本的分析合成出在视觉感受上与样本相近的大规模图案。该算法以图案中具有独立知觉意义的笔画集合(称为图案元素)为基本单位。合成过程兼顾图案元素的形状、元素间的拓扑关系这些局部特征, 以及元素分布的密度、均匀程度、各类图案元素的比例这些整体特征。

1 本文算法原理概述

本文算法首先通过绘图板等工具绘制一个图案样本, 并设定合成图案的尺寸, 然后合成出视觉感受与样本相似的图案。该算法包括样本绘制、样本分析和图案合成 3 个模块。

在样本绘制模块中, 本文以绘图工具产生的笔画轨迹信息作为图案样本。笔画轨迹记录了使用者绘制的笔画上的一组采样点。每个采样点包含坐标和笔尖压感信息, 要求使用者在绘制图案样本的过程中指明哪些笔画属于一个元素, 按照使用者绘制的顺序对元素进行编号, 记作 eID 。

样本分析模块中, 本文首先依据图案元素形态上的特征, 采用基于特征空间密度的聚类方法将其聚类, 形成元素库; 然后分析各元素之间的局部相邻关系, 并对样本中元素的密度、均匀程度以及各类元素的比例进行统计。这些信息将用于图案合成模块中。

图案合成模块首先按照样本中元素的密度和均匀程度在规定尺寸范围内放置种子点, 使种子点的分布与样本中元素的分布相似, 并通过分析种子点的局部相邻关系得到种子图; 然后使用遗传算法^[11]求一个映射, 将每个种子点替换为某个类型的元素, 使替换后得到的图案中元素的拓扑关系与样本相似, 同时使各类图案元素出现的比例也与样本相近; 最后按此映射将各种子点替换为元素库中的元素, 即得到新图案。为使图案更生动, 使用者还可以对图案元素的角度等信息进行微调。

图 2 所示为本文算法的整体流程。

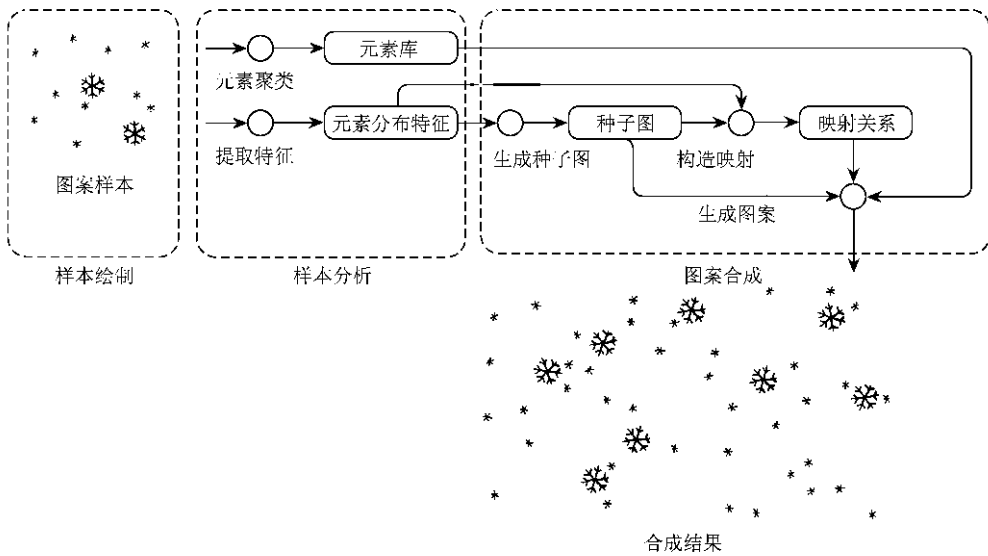


图 2 本文算法流程

2 样本分析

2.1 图案元素聚类

如图 1 所示, 图案中包含着大量的重复图案元素, 而这种重复也正是图案合成考虑的一个重要特征. 有些图案元素在形状上虽然存在小的差异, 但是在知觉上我们将它们归为相同的图案元素, 如图 1 中的图案元素大体上可以分为大雪花和小雪花 2 类. 因此, 本文以图案元素的形状特征作为依据, 对其进行聚类. 对于图案元素整体, 选取 a 和 r 2 个特征; 对于图案元素中每个笔画 m , 选取 l_m , d_m 和 c_m 3 个特征. 其中 a 为图案元素的包围盒面积, r 为包围盒的长宽比, l_m 为笔画的长度, d_m 为笔画方向, c_m 为笔画弯曲度. 本文采用基于密度聚类的思想, 首先计算各图案元素的特征参数, 然后将参数空间中密度达到一定水平的连续区域聚成一类. 按照产生的顺序对图案元素的类别进行编号, 记作 clD . 为了度量参数空间中点的密度, 本文对元素进行两两比较, 如果它们的差异小于一个阈值, 则相当于参数空间中此处密度较高, 这 2 个图案元素应属于同一类, 本文称这 2 个元素“相关”. 通过“相关”关系相连接的元素相当于处在参数空间中同一个高密度连通区域, 聚为一类. 例如 A 和 B 相关, B 和 C 相关, 则 A , B , C 属于同一类. 这种方法可以实现图案元素任意形状下的聚类, 并且对元素的顺序不敏感. 具体计算方法如下:

1) 提取元素特征. 定义元素的包围盒为该元素的一个面积最小的外接矩形, 令较长的边为包围盒

的长, 与其垂直的边为宽, 如图 3 b 所示. 对图案元素的每个笔画 m 分别计算特征, 先旋转图案元素, 使其包围盒的较长边与 y 轴重合, 且元素第一笔的起点在偏左的位置. 以笔画的骨架曲线的长度作为笔画的长度, 以笔画的起点指向终点的方向作为笔画的方向; 如果笔画的起点和终点距离很近, 经验值是小于笔画长度的 $1/5$, 则认为该笔画没有明显的方向性, 令其方向为 0. 笔画的弯曲度为笔画的骨架上的点到起点和终点连线的最大值, 如图 3 c 所示. 笔画特征的计算公式为

$$a = lth \cdot wth,$$

$$r = lth/wth;$$

$$l_m = \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2},$$

$$d_m = \arctan \left(\frac{y_n - y_0}{x_n - x_0} \right),$$

$$c_m = \max_{i \in (0, n)} \left| \frac{(x_n - x_0) \cdot (x_i - x_0) + (y_n - y_0) \cdot (y_i - y_0)}{\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2}} \right|.$$

其中, lth 为包围盒的长, wth 为包围盒的宽, n 为笔画轨迹上采样点的个数, x_i 和 y_i 为笔画中第 i 个采样点的坐标.

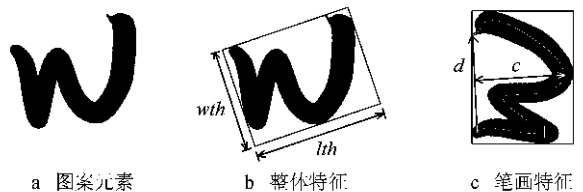


图 3 图案元素聚类特征

2) 比较图案元素. 对于 2 个图案元素 e_1 和 e_2 , 如果其笔画数目不同, 则认为它们不“相关”; 否则,

设 2 个图案元素的差异为 Dis , $Dis = C_1 \cdot Da + C_2 \cdot Dr + C_3 \cdot Ds$. 若 $Dis < 100$ (经验值), 认为 e_1 和 e_2 “相关”. 其中面积差异 $Da = \frac{a_1 - a_2}{\min(a^1, a^2)}$, 长宽比差异 $Dr = \frac{r_1 - r_2}{\min(r_1, r_2)}$, 2 个图案元素笔画上的差异 $Ds = \sum_{i=1}^n (C_4 \cdot \frac{c_{1i} - c_{2i}}{\min(c_{1i}, c_{2i})} + C_5 \cdot \frac{l_{1i} - l_{2i}}{\min(l_{1i}, l_{2i})} + C_6 \cdot (d_{1i} - d_{2i})) \cdot p$; n 为图案元素中笔画的数目. 参数 $C_1 \sim C_6$ 为各分量的权重, 其经验值分别取为 $C_1 = 60, C_2 = 10, C_3 = 40, C_4 = 0.2, C_5 = 2, C_6 = 3$.

3) 建立类库. 将 0 号图案元素归入第 0 类. 按照密度聚类的思想, 将每个待归类的元素 e_i 与已归类的各元素进行比较, 若与元素 e_j “相关”, 即将 e_i 归入 e_j 所在的类中; 若 e_i 与任何已归类的元素都不相关, 则新建一个类型, 将 e_i 归入其中, 如图 4 所示.

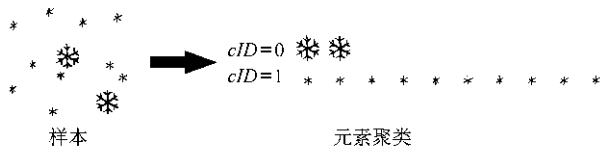


图 4 图案元素聚类

2.2 提取元素分布特征

图案元素的分布特征包含 2 个方面: 图案元素间相邻关系这种局部特征; 图案元素密度、均匀程度、比例等整体特征.

首先确定样本图案中图案元素间相邻关系这一特征. 以一个元素上所有点的坐标平均值作为这个元素的中心点, 将样本中所有元素的中心点进行

Delaunay 三角剖分^[12], 得到图 Ge . 对于样本中的一个元素 e_i , 其邻居为图 Ge 中所有与其直接相邻的图案元素, 如图 5 c 所示. 从 x 轴的负向开始, 按顺时针顺序记录其所有的邻居的编号 eID , 得到一个序列. 为了归纳出各类图案元素间的相邻关系, 按聚类信息将这个序列中的 eID 替换为对应的 cID , 得到 e_i 的邻居序列 $String_i$. 例如图 5 c 正中间的图案元素的邻居序列为 10101. 假设 e_i 属于 j 类, 则将 $String_i$ 作为 j 类图案元素的邻居的一种情况. 以一个二维数组 Te 记录样本中所有元素邻居序列, 表 1 列出了图 5 中样本的 Te 数组.

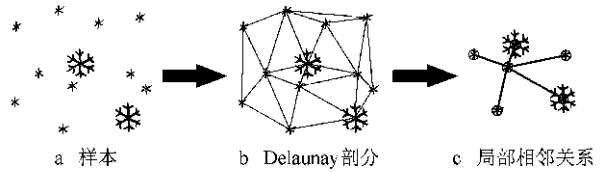


图 5 提取图案元素间的相邻关系

整体特征的计算如下: 图案元素的密度 $Density = Ne/Ae$, 其中 Ne 为样本中图案元素的个数, Ae 为样本的面积, 以像素为单位. 图案元素比例取样本中各类图案元素的个数和总图案元素个数的比值. 对于一个样本图案, 如果其图案元素排列很均匀, 则各元素间的距离相差不大; 否则, 各图案元素间的距离有大有小. 在样本的面积和元素个数确定的情况下, 图案元素排列越凌乱, 则各图案元素间距的最小值就越小. 因此, 本文取样本中 2 个图案元素的中心点之间距离的最小值 D_{min} 作为样本中图案元素排列均匀程度的度量.

表 1 图 5 样本的 Te 数组

元素类型	邻居序列									
0	11111	1111								
1	111	11011	1110	1111	110111	10101	011101	110	111	1110

3 图案合成

3.1 生成种子

本文在指定平面范围内用一组点来模拟样本中图案元素的分布规律, 这些点在后续过程中将被替换为样本中出现过的图案元素, 称这种点为种子点. 传统的四方连续图案的结构是完全重复性的, 这种图案在空间上具有明显的周期性, 视觉效果较为死板. 本文的方式既保留了样本中元素分布的整体特

征, 又具有一定的随机性, 合成的图案在结构上不重复, 效果生动自然.

放置种子点之前首先计算需要生成的种子点的个数, $Ns = Density \cdot As$, 其中 As 为合成图案的尺寸, 由使用者预先指定. 本文采用迭代的方式逐个放置种子点, 为了使种子点的分布接近样本中元素的分布, 放置一个种子点时在指定尺寸范围内随机取一个坐标点, 并查看以这点为圆心、半径为 D_{min} 的圆形范围内是否已经有其他种子点. 若没有, 则在当前位置放置种子点, 否则, 重新取一个坐标点, 重复

上一步,即可令种子点排列的均匀程度与样本相当.按照放置的先后顺序,从 0 开始对每个种子点进行编号,记为 sID .

将种子点置换成元素的过程中需要考虑各点间的相邻关系,与分析样本的方法类似,对种子点的集合进行德朗宁分割,得到种子图 G_s ,如图 6 a 所示.对每个种子点,从 x 轴负向开始顺时针记录其邻居,得到种子点集合的相邻关系,记为 T_s .数组 T_s 的结构与第 3.2 节中 T_e 的结构相同.

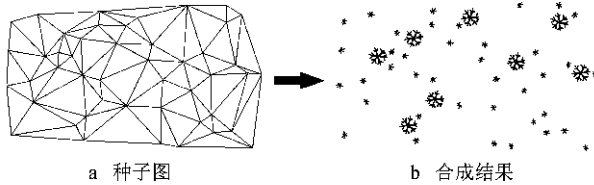


图 6 生成图案

3.2 构造映射关系

在种子点的基础上合成图案就是寻找一个映射,将每个种子点映射到一个元素类型,即将每个 sID 置换为一个 cID ,使置换后图案的局部相邻关系以及各类图案元素的比例与样本相近.这是一个优化问题,本文使用遗传算法来求解,将待求解的问题编码以一个数字序列表示,序列上的每个数字表示一个基因,对这个序列的一个赋值称为一个个体,若干个个体组成一个种群.生成一个初始种群后,通过计算适值、选择、交叉、变异使种群不断进化,最后取一个最优秀的个体,对其解码后就是问题的解.使用遗传算法主要包括设计编码方式、生成初始种群方式、适值函数和遗传操作 4 个部分.

1) 编码方式.采用十进制编码方式以一个整数序列作为问题的编码.序列的长度为种子点的个数,每个基因的取值限定在 cID 的取值范围之内.序列中第 i 个位置上的数值为 j 就表示将 $sID = i$ 的种子置换成 $cID = j$ 的类型的某个元素.这样,每个个体就代表将每个种子点置换为元素类型的一个映射.

2) 生成初始种群方式.设定种群大小为 20,随机生成 20 个个体作为初始种群.由于每个个体都是一个种子到元素类型的映射,这 20 个个体分别对应于 20 个图案.由于这些个体是随机生成的,这些图案可能与样本的相似度不高,但是随着遗传算法的迭代进行,整个种群将不断进化,最终得到的个体将产生与样本相似度很高的图案.

3) 设计适值函数.为了衡量一个个体与样本在局部和整体上的相似程度,本文对个体和样本中元

素的局部相邻关系和类型比例这 2 个特征进行对比,求得适值的 2 个分量,并令适值取这 2 个分量的加权和.设适值为 $Fitness$,相邻关系分量为 F_n ,元素比例分量为 F_r .

受到文本处理方法的启示,在计算相邻关系分量时,采用 Levenshtein 距离^[13]度量 2 个邻居序列的差异程度.在一个个体中,每一个种子都对应一个元素类型.若一个种子 s_i 对应 j 类型的元素,则将其邻居序列与 T_e 数组中 j 类型的所有邻居序列进行对比,并计算编辑距离,取这些编辑距离中的最小值记为 $\min Levenshtein(s_i, T_e)$.Levenshtein 距离是文本处理的一种算法,用于衡量 2 个字符串之间的差异程度.对一个邻居序列 $String_1$ 进行下列 3 种操作,若最少进行 n 次操作才能使 $String_1$ 与另一邻居序列 $String_2$ 相等,则这 2 个邻居序列的 Levenshtein 距离为 n .3 种操作如下:在任意位置插入一个邻居,删除一个邻居,改变一个邻居. Levenshtein 距离越大,表示 2 个邻居序列的差异越大,该值不会大于较长序列的长度.一个元素或种子的邻居数一般不会大于 10,为了使较大的适值表示较小的差异,用 10 减去 Levenshtein 距离作为这个种子上的适值分量;将所有种子上的适值分量累加,并标准化,即得到这个个体适值的相邻关系分量 F_n .

将个体中各类元素的比例与样本中的比例进行差运算,求绝对值后累加并标准化,得到个体适值的元素比例分量 F_r .具体运算公式如下:

$$Fitness = F_n + C \cdot F_r,$$

$$F_n = \sum_{i=0}^n (10 - \min Levenshtein(s_i, T_e)),$$

$$F_r = \sum_{k=0}^m \left| \frac{C_{s_k}}{N_s} - \frac{C_{e_k}}{N_e} \right|.$$

其中, n 为种子数量, m 为元素类型数量, C_{s_k} 和 C_{e_k} 分别表示该个体和样本中第 k 类元素的数目.

4) 遗传操作.遗传操作包括选择、交叉、变异.

a. 选择.选择操作使适值高的个体有更多的机会被复制,进而使种群整体水平不断提高.采用转轮法,从 0 开始连续地为每个个体分配一个区间,最后一个个体的区间上限为 max .引入精英主义的思想,繁殖之初下一代种群为空,先将上一代最优个体加入下一代;然后每次取一个 0 到 max 的整数,其落在哪个个体的区间内,就将这个个体加入下一代种群中,直到下一代种群的大小与当前代相等.为了加快收敛速度,给适值小于平均值的个体分配一个

长度仅为 1 的区间, 而适值大于均值的个体所分配的区间长度为其适值与均值的平方差。

b. 变异. 变异操作是进化的源头, 变异使一个个体发生轻微的改变, 比如在一个个体中, 种子点 s_i 原本映射到 j 类型元素, 而改变后它映射到 k 类型的元素. 这种改变可能是进化也可能是退化, 即有可能使一个个体对应的图案与样本的相似程度更高, 也有可能反之. 退化的个体会被选择操作淘汰掉, 进化的个体则会被保留下来. 仍然采用精英主义的思想, 使种群中最优的一个个体不发生任何变化, 对于其他个体, 首先在该个体序列上随机选择一个位置, 然后给这个位置赋以一个随机的 cID 值. 这样, 可以保证每代中最优个体的适值不小于上一代。

c. 交叉. 交叉的作用是将不同个体的优秀基因集于一体. 本文算法中, 一个基因代表一个种子到元素类型的映射, 而评价个体优劣的标准是该个体映射出的图案中各元素的相邻关系以及各类元素的比例, 所以单个基因不具有优秀与否的性质. 因此本文不采用交叉算子, 而是采用一种类似无性繁殖的方式。

本文算法流程如下:

随机生成 20 个个体, 作为初始种群;

循环 1000 次:

计算每个个体的适值;

选择:

从 0 开始连续的为每个个体分配一个区间, 最后一个区间的上限记为 max , 即生成转轮;

生成一个空的新种群;

将上一代的最优个体复制到新种群;

循环 19 次:

取一个 $0 \sim max$ 之间的随机数;

这个数落在哪个个体的区间内, 就将此个体复制

到新种群;

用新种群取代旧的种群;

变异:

查找最优个体, 这个个体不发生改变;

对于其他个体:

随机选取个体序列上的一个位置;

随机取一个 cID 值, 赋给这个位置上的数;

取当前种群中适值最高的个体作为结果输出;

算法结束;

4.3 生成图案

按照遗传算法所得的结果逐一替换种子点, 即得到合成结果, 如图 6 所示. 若结果中一个种子置换为第 i 类元素, 则从第 i 类等概率随机选取一个元素进行置换. 为使合成的图案不显单调, 可以对图案中的元素添加微调, 微调操作对图案中的每一个元素进行一个小幅度的旋转, 各元素的旋转角度都取随机值. 旋转时, 指定一个角度范围, 然后在此范围内取一个随机数 θ , 将元素上的每个点做简单的投影, 即

$$\begin{cases} x_1 = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta \\ y_1 = y_0 \cdot \cos \theta - x_0 \cdot \sin \theta \end{cases}$$

其中 x_0, y_0 为一个点映射前的坐标, x_1, y_1 为映射后的坐标。

4 实验结果分析与比较

我们在 Windows XP 系统下用 VC++ 6.0 编程环境实现了本文算法, 硬件环境为 P4 3.0 GHz CPU, 1 GB 内存, 手绘图案样本输入采用 Wacom (bamboo one CTE-631) 数位板. 图 7 所示为用本文算法合成的 4 组结果, 每组图片中左边的小图是手绘图案样本, 右边的大图是合成图案结果。

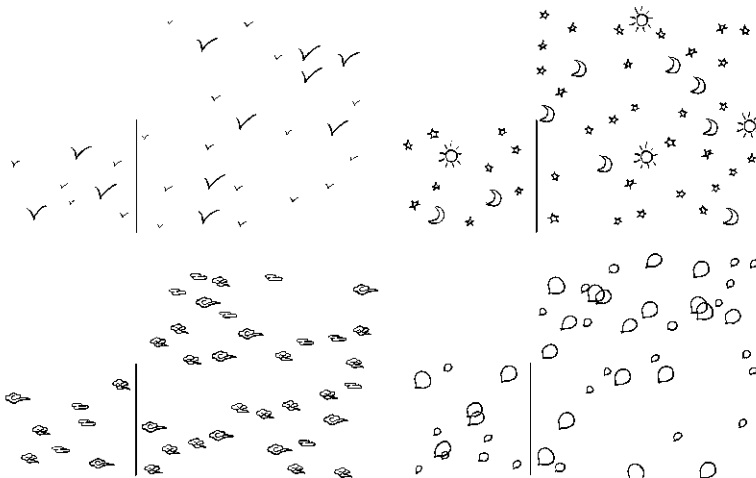


图 7 合成结果

图 8 所示为合成图案添加自动微调后的效果,可以看出,本文算法合成的图案既保留了样本的局

部和整体视觉特征,又不是图案样本的简单机械重复.

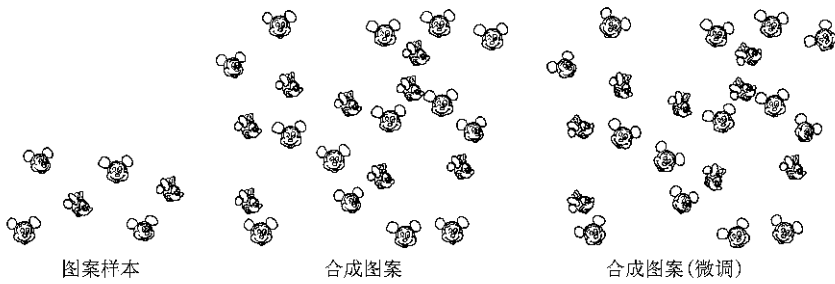


图 8 微调前后效果对比

在放置种子点的过程中, Barla 等^[7]的算法不考虑样本中元素分布的整体特征,而是假设样本中的元素是准均匀分布的,在生成种子时采用 Lloyd 算法使种子点的分布趋于均匀;本文算法则对样本中元素分布的密度和均匀程度进行了度量,并以此为依据放置种子,使种子的分布与样本中元素的分布更相似.

把握仅局限在元素的形态上,忽略了图案元素之间的相互关系,该方法假设图案样本没有明确的结构特征,而这个假设只在一部分情况下成立.本文算法则能兼顾图案的局部和整体特征,使合成的图案与样本在视觉效果上具有更好的相似性.图 9 所示为对这 3 种算法的对比结果,不难看出,Barla 等的算法生成的图案过于整齐,与样本存在较大差异;Hurtut 等的结果在图案元素局部分布方面与样本不符.图 9b,9c 中,圆圈标出部分与样本存在明显的差异.本文算法结果在视觉效果则与样本更为相似.

在生成图案的过程中, Barla 等的算法仅考虑图案元素邻居关系匹配这种局部关系,不考虑图案的整体特征;而 Hurtut 等^[9]的方法对图案局部特征的

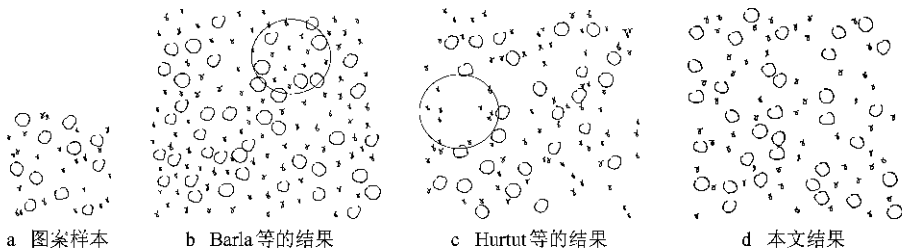


图 9 3 种算法结果比较

但是本文算法也有不足之处,主要是遗传算法比较耗时;另外,对于结构非常规则的图案(如砖墙图案等),其生成的结果还不理想.不过这点也是 Barla 和 Hurtut 等算法存在的共同问题,有待于在以后的工作中加以解决.

5 结 论

本文提出一种基于样本的手绘图案合成算法,由使用者实时绘制一个小规模图案样本,将样本中具有独立知觉意义的图案元素视为合成图案的基本单元,利用图案元素在图案中重复出现的特点,首先按照视觉特征对图案元素进行聚类;然后根据样本手绘图案中图案元素分布的密度、均匀程度、各类图案元素的比例、局部相邻关系这些特征,合成出视觉效果与样本图案相似的大范围图案.实验结果表明

根据本文算法合成的图案与样本图案在局部和整体特征上都具有较好的相似性,优于仅考虑局部特征或整体特征的其他方法.

基于样本的手绘图案合成还是一个很新的研究领域,目前主要考虑图案元素呈离散分布状态下的图案合成,即图案元素可以明确地划分开,并且分布带有一定随机性的情况.因此,具有一定结构并呈连续分布的手绘图案合成是一个有意义的研究方向.除了图案元素的形状和拓扑特征外,图案元素的色彩也是一个重要的视觉特征,我们会在以后的工作中加以考虑.

参考文献(References):

[1] Hertzmann A. A survey of stroke-based rendering [J]. IEEE Computer Graphics and Applications, 2003, 23(4): 70-81

- [2] Zhu Wenhao, Wei Baogang. The technology of sampled-based texture synthesis [J]. *Journal of Image and Graphics*, 2008, 13(11): 2063–2069 (in Chinese)
(朱文浩, 魏宝刚. 基于样本的纹理合成技术综述[J]. *中国图象图形学报*, 2008, 13(11): 2063–2069)
- [3] 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
- [4] Efros A A, Leung T K. Texture synthesis by non-parametric sampling [C] // *Proceedings of the IEEE International Conference on Computer Vision*. Washington D C: IEEE Computer Society Press, 1999: 1033–1038
- [5] Wei L Y, Levoy M. Fast texture synthesis using tree-structured vector quantization [C] // *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press, 2000: 479–488
- [6] Liu Y M, Wang J P, Xue S, *et al.* Texture splicing [J]. *Computer Graphics Forum*, 2009, 28(7): 1907–1915
- [7] Barla P, Breslav S, Thollot J, *et al.* Stroke pattern analysis and synthesis [J]. *Computer Graphics Forum*, 2006, 25(3): 663–671
- [8] Ijiri T, Mèch R, Igarashi T, *et al.* An example-based procedural system for element arrangement [J]. *Computer Graphics Forum*, 2008, 27(2): 429–436
- [9] Hurtut T, Landes P E, Thollot J, *et al.* Appearance-guided synthesis of element arrangements by example [C] // *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*. New York: ACM Press, 2009: 51–60
- [10] Treisman A M, Gelade G. A feature-integration theory of attention [J]. *Cognitive Psychology*, 1980, 12(1): 97–136
- [11] Goldberg D E, Holland J H. Genetic algorithms and machine learning [J]. *Machine Learning*, 1988, 3(2/3): 95–99
- [12] Lee D T, Schachter B J. Two algorithms for constructing a Delaunay triangulation [J]. *International Journal of Computer and Information Sciences*, 1980, 9(3): 219–242
- [13] Konstantinidis S. Computing the edit distance of a regular language [J]. *Information and Computation*, 2007, 205(9): 1307–1316

(上接第 143 页)

- [7] Vlachos A, Peters J, Boyd C, *et al.* Curved PN triangles [C] // *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*. New York: ACM Press, 2001: 159–166
- [8] Boubekeur T, Reuter P, Schlick C. Scalar tagged PN triangles [OL]. [2010-07-10]. <http://iparda.labri.fr/publications/2005/BRS05/STPN.pdf>
- [9] Brabec S, Seidel H P. Shadow volumes on programmable graphics hardware [J]. *Computer Graphics Forum*, 2003, 22(3): 433–440
- [10] Benichou F, Elber G. Output sensitive extraction of silhouettes from polygonal geometry [C] // *Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*. Washington D C: IEEE Computer Society Press, 1999: 60–69
- [11] Dyken C, Reimers M. Real-time linear silhouette enhancement [OL]. [2010-07-10]. <http://heim.ifi.uio.no/~erikd/pdf/silhouette.pdf>
- [12] Kirsanov D, Sander P V, Gortler S J. Simple silhouettes for complex surfaces [C] // *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. Aire-la-Ville: Eurographics Association Press, 2003: 102–106
- [13] Pop M, Duncan C, Barequet G, *et al.* Efficient perspective-accurate silhouette computation and applications [C] // *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*. New York: ACM Press, 2001: 60–68
- [14] Wang B, Wang W, Wu J, *et al.* Silhouette smoothing by boundary curve interpolation [OL]. [2010-07-10]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.4663&rep=rep1&type=pdf>