

混沌粒子群优化的纹理合成算法研究

瞿 中 李 楠

(重庆邮电大学计算机学院 重庆 400065)

摘 要 粒子群算法在搜索后期由于搜索空间有限,容易陷入局部极值,过早地进入早熟状态。针对这种情况,将混沌优化搜索技术用于粒子群算法,利用混沌运动的遍历性、随机性等特点,提出了一种混沌粒子群优化的块采样纹理合成算法。实验结果表明,混沌粒子群算法比粒子群算法具有更好的全局寻优能力,克服了粒子群算法的缺点,得到了较高质量的纹理合成图像。

关键词 混沌粒子群算法,纹理合成,块采样,粒子群算法

中图分类号 TP391.41 文献标识码 A

Algorithm of Texture Synthesis Based on Chaos Particle Swarm Optimization

QU Zhong LI Nan

(College of Computer Science & Technology, Chongqing University of Posts & Telecommunications, Chongqing 400065, China)

Abstract As the searching space is limited in its later search, the particle swarm optimization algorithm factors is easy to fall into local minimum, and access to the premature state early. In response to these circumstances, a new patch-based method for texture synthesis based on chaos particle swarm optimization was proposed. Chaos optimization search technique was used in particle swarm optimization in this paper. The experimental result shows that comparing with particle swarm optimization, chaos particle swarm optimization has better optimization performance, overcomes the disadvantage of particle swarm optimization, and gains the texture synthesis image of higher quality.

Keywords Chaos particle swarm optimization, Texture synthesis, Patch-based sampling, Particle swarm optimization

1 引言

纹理合成是以人工方式生成纹理为目的,是当前计算机视觉、计算机图形学、图像处理等领域的研究热点之一。纹理合成技术的发展主要经历了三个阶段:第一阶段是纹理映射,常用于绘制复杂场景真实感图形,它把样图直接匹配到目标物体表面上,在物体表面形成真实的纹理图像。但是纹理映射常常只能在纹理空间和表面参数空间进行一对一的映射,没有特征提取和模型优化;第二阶段是过程纹理合成,它通过直接仿真某种表面纹理的物理生成过程来进行纹理合成,但是该技术计算量大,合成速度慢;第三阶段是基于样图的纹理合成,该技术^[1,2]是近几年迅速发展起来的一种新的纹理拼接技术,它利用纹理本身具有的局部相似性,给定一块小区域纹理样本,按照其表面的几何形状,拼合生成整个曲面的纹理,该技术受到越来越多研究人员的关注。

在文献[3]中,粒子群(particle swarm optimization, PSO)算法容易陷入局部最优,找不到最优的匹配块,本文将混沌粒子群优化(chaos particle swarm optimization, CPSO)算法^[4]应用于纹理合成中,提出了一种以块算法为基础、CPSO 算法为搜索策略的新型纹理合成算法。

2 块纹理合成算法

块纹理合成算法^[5]是以块为合成单位,每次在输出图像中合成一个块,并利用块与块之间的重叠边界比对来选取合适的匹配块。其一般合成过程如图 1(a)所示。

块纹理合成算法的主要思想为:首先从样图中随机地选择一块复制到输出纹理图像中,且每两个块之间有一定宽度的重叠边界部分,每次合成当前块时,都从样图中挑选一系列块,使其边界与已有的纹理的边界匹配误差小于给定的阈值,然后从中随机选择一个填入输出图像区域中。若无法找到满足误差小于给定阈值的块,则选择具有最小误差的块填入输出图像中,直至图像合成完毕。图 1 中的(b)(c)(d)分别阐明了 3 种可能的纹理块匹配情况。

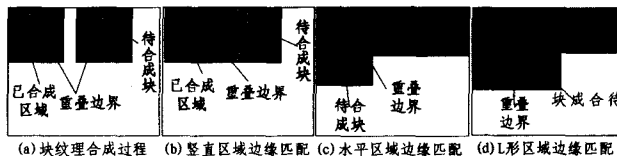


图 1 块纹理合成过程及匹配的 3 种情况

到稿日期:2009-11-03 返修日期:2010-02-01 本文受重庆市教委项目(0834218),重庆邮电大学博士启动基金(A2009-11),重庆邮电大学项目(2009ZDKC2)资助。

瞿 中(1972-),男,博士,主要研究方向为数字图像处理、普适计算等,E-mail:quzhong@hotmail.com;李 楠(1983-),男,硕士生,主要研究方向为数字图像处理。

3 混沌粒子群优化算法

3.1 粒子群优化算法

粒子群优化算法^[6,7]是1995年由Kennedy等人提出的一种进化技术,它以组织社会行为代替进化算法的自然选择机制,并通过种群间个体协作来实现对问题最优解的搜索。

在粒子群模型中,优化问题的每个解对应搜索空间的一个粒子,每个粒子有一个速度决定它的方向和位置。粒子群优化算法首先初始化粒子群,计算出每个粒子的适应值,然后通过迭代搜索最优解。在每次迭代中,对于每个粒子,粒子通过追踪两个极值来更新自己:一个是粒子自己找到的最优解,称为个体极值;另一个是整个粒子群目前找到的最优解,称为全局极值。当找到个体极值和全局极值后,粒子按式(1)和式(2)来更新自己的速度和位置:

$$v(t) = \omega v(t-1) + \mu_1 \eta_1 (p(t-1) - x(t-1)) + \mu_2 \eta_2 (g(t-1) - x(t-1)) \quad (1)$$

$$x(t) = x(t-1) + v(t-1) \quad (2)$$

式中, t 为当前的进化代数; ω 为惯性权重; μ_1, μ_2 为加速因子; η_1, η_2 为 $[0, 1]$ 之间的随机数; $p(t-1)$ 为粒子 $t-1$ 时的个体极值; $g(t-1)$ 为整个种群 $t-1$ 时的全局极值解。

3.2 粒子群早熟判别标准

本文判别粒子群是否早熟的主要依据是平均粒距和种群的适应度方差。如果平均粒距小于阈值 T ,且种群的适应度方差小于阈值 S ,则可认定粒子群发生了早熟。平均粒距定义为

$$dis = \frac{1}{M \cdot L} \cdot \sum_{i=1}^M \sqrt{\sum_{d=1}^D (p_{id} - p_d)^2} \quad (3)$$

式中, L 为搜索空间对角最大长度, D 为解空间维数, M 为种群规模大小, p_{id} 为第 i 个粒子的第 d 维坐标值, p_d 为所有粒子第 d 维坐标值均值。

种群适应度方差^[8,9]则是用来反映种群中个体的聚集程度,其定义为

$$\sigma^2 = \sum_{i=1}^M \left(\frac{f_i - f_{avg}}{f} \right)^2 \quad (4)$$

式中, f_i 为第 i 个个体的适应度, f_{avg} 为当前种群的平均适应度, f 为归一化定标因子,主要是用来限制 σ^2 的大小,其定义为

$$f = \begin{cases} \max_{1 \leq i \leq M} |f_i - f_{avg}|, & \max |f_i - f_{avg}| > 1 \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

根据式(3)可得知, dis 越小,表明种群越集中,反之, dis 越大,表明种群就越分散。而由式(4)可得知, σ^2 越小,表明种群中个体的聚集程度越大,反之, σ^2 越大,则表明其聚集程度越小。种群的个体适应度会随着迭代次数的增加越来越接近,从而, σ^2 会越来越小。如果 σ^2 小于给定的阈值 S 时,就可认为粒子群发生了早熟。

3.3 混沌优化搜索

混沌是一种具有随机性的运动状态,可以由确定性方程所得到,它具有遍历性、随机性等特点。本文所采取的 Logistic 映射即是一个典型的混沌系统,其定义为

$$z_{n+1} = \mu z_n (1 - z_n) \quad (6)$$

混沌优化搜索的算法过程:

(1) 随机产生一个在 $[0, 1]$ 之间的初始混沌变量。

(2) 利用式(6)产生混沌序列,并将其变换到变量的取值区间。

(3) 计算每个混沌序列的适应度值,并比较所得到的适应度值。

(4) 选取适应度最高的粒子作为全局最优粒子。

3.4 混沌粒子群优化算法

CPSO算法的主要思想是:首先运行粒子群优化算法,如果当粒子陷入局部最优,进入早熟状态时,则对其进行混沌优化搜索,使粒子跳出局部最优,扩大搜索空间,搜索全局最优解,并加快收敛。

算法的具体流程如图2所示。

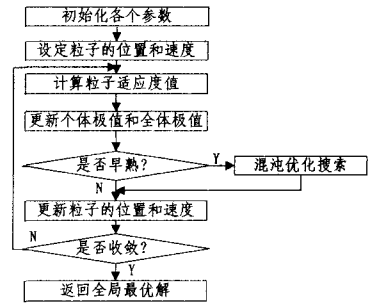


图2 CPSO算法流程图

4 CPSO的纹理合成算法

在纹理合成过程中,如何在输入样图中选择与输出图像中的已有块边缘最相似的匹配块,这就需要设定一定的匹配准则。本文采用RGB模型,用纹理块3种颜色的平方差的和来度量两个块的匹配特性,即目标函数,并设定一个比较阈值 δ 。其目标函数定义为

$$d(s_1, s_2) = \sum_{i=1}^A \{ (R_1(p) - R_2(p))^2 + (G_1(p) - G_2(p))^2 + (B_1(p) - B_2(p))^2 \} \quad (7)$$

式中, s_1 和 s_2 是两块同样形状和大小的纹理块; $R_1, R_2, G_1, G_2, B_1, B_2$ 分别是两块纹理在对应点 P 的红、绿、蓝3种颜色值; A 为边缘区域中像素点的个数。

该匹配准则逐一比较两个边缘区域中像素点的RGB值,计算出误差总量,误差总量越小说明两个边界区域越相似,当比较区域的相似性误差小于指定的阈值 δ ,或者达到最大的迭代次数时即停止迭代,把当前的最优解粒子的位置记录下来,并把它决定的纹理块复制到输出图像中。

CPSO的纹理合成算法过程:

(1) 首先设定纹理块大小(如 30×30),然后随机地在输入样本纹理中选择该大小的一个纹理块。

(2) 把所选择的纹理块复制到输出纹理图像中。

(3) 在当前已合成的输出纹理图像中,选取一定宽度的重叠纹理边界。

(4) 使用CPSO算法在输入样本纹理图像中选择与该重叠纹理边界相匹配的纹理块。

(5) 如果能够在输入纹理中找到满足小于 δ 的纹理块,则把所找到的纹理块复制到输出纹理图像中,否则待循环到最大次数,把当前的全局最优值所决定的纹理块复制到输出纹理图像中。

(6) 重复执行(3)~(5),直到输出的纹理图像全部合成完成。

(7) 用羽化算法^[5]融合输出纹理图像中的重叠纹理边界的连接处。

5 实验结果及分析

本文的算法以 Matlab7.0 为平台,在 Intel Core2 CPU 主频 1.66GHz,内存 2G 的微机机上实现。

5.1 CPSO 和 PSO 的函数优化比较

采用 4 个典型的函数对 PSO 算法和 CPSO 算法进行优化比较实验。实验中参数的设置如下:粒子的最大迭代次数为 600, $c_1=c_2=1.47$,混沌搜索的最大次数为 200,函数维数取 30,平均粒距的阈值设为 0.01,种群适应方差的阈值设为 0.001。实验所选取的函数为:

(1) Ackley 函数

$$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e, -30 \leq x_i \leq 30$$

(2) Griwank 函数

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, -300 \leq x_i \leq 300$$

(3) Quadric 函数

$$f(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2, -100 \leq x_i \leq 100$$

(4) Rosenbrock 函数

$$f(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), -50 \leq x_i \leq 50$$

分别用 PSO, CPSO 算法对这 4 个函数进行优化,各做 30 次独立实验,结果如表 1 所列。

表 1 函数优化结果比较

函数	适应值	Ackley	Griwank	Rosenbrock	Quadric
PSO	最大	6.9109	0.0817	8.3984	3.2469
	最小	0.0020	3.7512	0.0017	9.6511
	平均	2.4108	0.0182	2.9812	1.9102
CPSO	最大	1.2435	0.0542	0.0936	1.6920
	最小	0.0060	2.8437	0.0099	2.1289
	平均	0.4896	0.0109	0.0517	1.5601
变量范围		[-30, 30]	[-300, 300]	[-50, 50]	[-100, 100]

从表 1 可以看出,同样的参数设置,在 30 次独立试验中, CPSO 每次实验所得到的最大适应值和最小适应值的变化幅度要小于 PSO,因此 CPSO 每次得到的最优值比 PSO 更趋向于稳定。同时, CPSO 得到的平均适应值比 PSO 得到的平均适应值好。

5.2 搜索成功率比较

两种搜索算法中粒子数量均为 20,各算法均独立运行 50 次对目标函数式(7)进行优化,且从同一个固定的起始块开始匹配搜索。如果每次运行的结果在全局最优值的 3% 范围内,则称该次运行行为“成功运行”,搜索成功率定义为

$$\text{搜索成功率} = \frac{\text{成功运行次数}}{\text{独立运行总次数}} \times 100\%$$

两种算法的搜索成功率结果如表 2 所列。

表 2 搜索成功率比较

算法	平均迭代次数	成功运行次数	失败次数	搜索成功率
PSO	87	44	6	88
CPSO	49	50	0	100

由表 2 可见, CPSO 算法在规定的迭代次数内总可以以较高的概率找到全局最优解,且对于成功的运行而言, CPSO

算法所用的迭代次数最小,较 PSO 更稳定。此外, CPSO 搜索到全局最优解的成功率为 100%,而 PSO 有 6 次搜索失败,成功率仅为 88%,说明, PSO 容易陷入早熟,而 CPSO 却可以避免陷入早熟,跳出局部最优,搜索到全局最优解。因此,用 CPSO 搜索到的纹理匹配块要优于用 PSO 搜索到的纹理匹配块,从而在纹理图像的合成质量上, CPSO 合成的结果质量较高。

5.3 粒子数对合成质量和合成时间的影响

为了分析粒子数对本文算法在合成质量和合成时间上的影响,本文采用两组不同大小的输入样本纹理做了实验,得到了 256×256 的输出图像。

图 3 显示了不同粒子数合成结果。其中(a),(b)为两组输入样图,(a1)-(a3),(b1)-(b3)分别为两组样图在粒子数为 10、20、30 时应用本文算法合成的结果。从图 3 中可以看出,粒子数的选取会一定程度上影响到能否找到最合适的匹配块和合成后的纹理的质量。在本实验中,当粒子数为 20 时, CPSO 取得了很好的搜索效果,合成出来的图像质量较高。

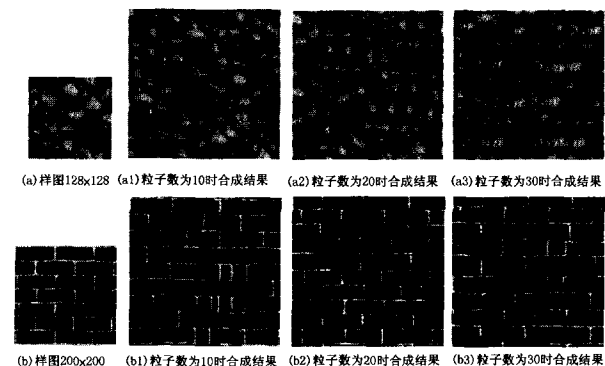


图 3 不同粒子数的合成结果

图 4 给出了粒子数对合成时间上的影响。从图 4 可以发现,用本文算法和 PSO 算法合成同样大小纹理所需的时间,都随着粒子数的增加而增长。若是选取的粒子数过多,在很大程度上会降低合成的速度,大大延长合成的时间;在同样粒子数时,本文算法合成的时间比 PSO 合成的时间略微长一些,这是由于本文算法加入了混沌优化搜索,但对于盲目无序的随机穷举搜索, CPSO 耗时已大大减少。

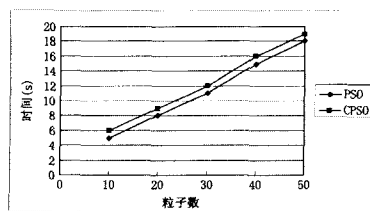


图 4 不同粒子数的合成时间比较

因此,在进行纹理合成时,要根据合成质量和合成速度两方面因素综合考虑,在较少的时间花费上,实现较高质量的合成结果,从图 3 和图 4 综合可知,当随机粒子数为 20 时,本文算法在合成质量和合成速度上都取得了较理想的结果。

5.4 合成结果的比较

在粒子数为 20,最大迭代次数为 300 的条件下,分别采用 PSO 和 CPSO 算法合成的结果如图 5 所示。

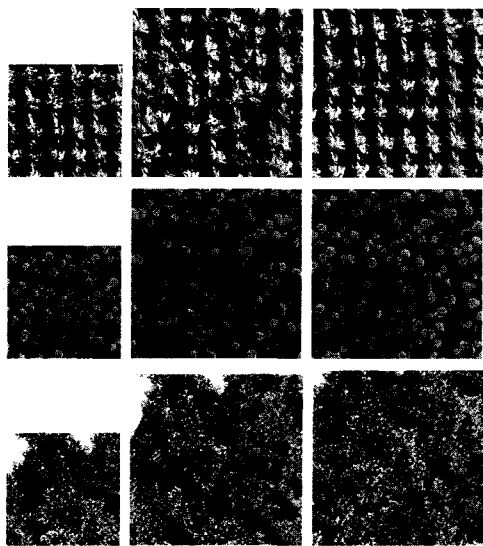


图5 分别使用 PSO 和 CPSO 合成的 256×256 图像结果比较

图5 分别使用 PSO 和 CPSO 合成的 256×256 图像结果比较

图5(a)为输入样图,图5(b)是运用 PSO 算法的合成图像,由于 PSO 算法搜索空间有限,容易陷入局部最优,导致找不到全局最优的纹理匹配块,从图中就可以看出合成的纹理图像中存在着许多的重复和过度不连续的区域,而图5(c)中使用的是 CPSO 算法,CPSO 算法能够使其在搜索空间上最大化,防止早熟,快速有效地搜索到全局最优匹配块,有效地避免了使用 PSO 算法合成时的缺陷,从图中可以很明显地看出用本文算法合成的纹理图像的质量优于使用 PSO 算法合成的纹理图像的质量。

结束语 本文提出的混沌粒子群优化的纹理合成算法是

一种有效的纹理合成算法,该算法能够有效地克服 PSO 容易早熟、陷入局部最优的缺点,从而快速有效地搜索到全局最优的匹配块,实现了较高质量的纹理合成结果。

参考文献

- [1] Kwatra V, Essa I A, Bobick A F, et al. Texture optimization for example-based synthesis [J]. ACM Transaction on Graphics, 2005, 24(3): 795-802
- [2] 曹璐, 杨刚, 王忠芝, 等. 源自样图的树皮纹理合成技术研究[J]. 计算机工程与应用, 2009, 45(30): 194-197
- [3] Zhang Yan, Meng Yu L, Wen-hui, et al. A fast algorithm for image analogy using particle swarm optimization [C] // Proc. of the 3rd International Conference on Machine Learning and Cybernetics. Shanghai, 2004: 26-29
- [4] 刘军民, 高岳林. 混沌粒子群优化算法[J]. 计算机应用, 2008, 28(2): 322-325
- [5] Liang L, Liu C, Xu Y, et al. Real-time texture synthesis using patch-based sampling. Microsoft Research [R]. 2001. MSRTR. 2001: 40
- [6] Kennedy J, Eberhart R C. Particle swarm optimization [C] // IEEE Service Center. Proc IEEE Int'l Conf. on Neural Networks. Vol. IV, 1995
- [7] 张焱, 高兴宝. 一种改进的粒子群算法[J]. 计算机工程与应用, 2009, 45(26): 58-59, 93
- [8] 杨延西, 刘丁, 辛菁. 基于混沌粒子群优化的图像相关匹配算法研究[J]. 电子与信息学报, 2008, 30(3): 529-533
- [9] 黄贤英, 张丽芳. 基于粒子群优化的模糊聚类算法[J]. 重庆工学院学报: 自然科学版, 2008, 22(11): 120-123

(上接第 241 页)

点,选择相应的笔触绘制在场景中客体表面的分点处。由于在 3.1 节客体表面分点的分类中已经在观察坐标系中对各分点按照 z 分量由小到大进行了排序,这样分点处进行笔触绘制时可按照排好序的顺序进行笔触绘制,避免了额外的笔触消隐;同时在每一分点处绘制笔触时,除了笔触自身的漫反色信息外,还加上了前面确定的双光源下各分点的明暗信息和镜面反色信息,从而保证了绘制作品场景中整体色调的协调性和合理性。

结束语 本文对双光源环境下非真实感绘制的色调信息获取方法进行了研究,提出了一种基于“小球”的双光源色调信息提取方法,该方法假设场景中客体表面均匀覆盖了一层大小相同的小球,这些小球起到了“提取器”的作用,通过计算这些小球在双光源下被遮挡的明暗信息和近似的镜面反射信息等来确定小球所在分点位置的色调信息,在绘制过程中利用这些色调信息在相应位置上绘制“笔触”,从而保证在双光源环境下各种非真实感绘制作品的整体色调的合理性。此外,所提出的色调信息提取方法具有“提取器”结构简单、处理统一和方便局部信息调整等特点,可方便应用于各种非真实

感绘制模式中。

参考文献

- [1] 苏延辉, 韦欢, 费广正, 等. 非真实感绘制技术研究[J]. 中国传媒大学学报: 自然科学版, 2006, 13(2): 15-21
- [2] 辛玲. 模拟绘画的非真实感绘制方法研究[D]. 大连: 辽宁师范大学, 2008
- [3] 钱小燕, 肖亮, 吴慧中. 基于多分辨率的非真实感绘制[J]. 南京理工大学学报, 2006, 30(3): 348-351
- [4] Pang Y J. Combining computer graphics with Chinese traditional painting [J]. Computer & Graphics, 1987, 11(1): 63-68
- [5] Pang Y J, Zhong H X. Drawing Chinese traditional painting by computer [C] // Proc. IFIP WG5. 10 Working Conference on Modeling in Computer Graphics. Tokyo, Japan, 1991: 321-328
- [6] 王相海. “体元”纹理真实感图像生成研究[D]. 长春: 吉林大学, 1995
- [7] 王相海, 庞云阶. 模拟绘画的三维几何纹理生成研究[J]. 计算机学报, 2002, 25(9): 982-986
- [8] 辛玲, 王相海. 非真实感油画绘制的色调获取方法研究[J]. 计算机科学, 2008, 35(12): 216-219