

# 基于自回避随机游动方法的曲流河河道的建模研究

田金琴<sup>1</sup>, 李志勇<sup>2</sup>

(1. 北方民族大学计算机科学与工程学院, 银川 750021; 2. 郑州航空工业管理学院数理系, 郑州 450005)

**摘要:**研究了游动者在给定约束条件下的一种基于网格的自回避随机游动算法的实现, 并利用该算法实现了对曲流河河道形态的建模。影响游动者迁移的约束条件有游动主方向、格点的吸引因素、排斥因素。在网格区域中完成一次随机游动之后, 记录游动者经过的格点, 通过样条插值对连接格点的路径进行曲线拟合, 将拟合后的曲线加宽并光滑处理, 实现对自然界中曲流河河道的模拟, 该系统利用 OpenGL 图形库在 C 开发环境进行了实现。利用该算法, 还可实现海岸线等自然界具有蜿蜒曲折的形态物体的在条件限定情况下的建模。

**关键词:**自回避游动; 特征项; 迁移概率

**中图分类号:** TP202.4

**文献标识码:** A

## 引言

自然界中许多物理、化学、生物过程乃至一些社会现象都与随机过程有关。许多比较简单的随机现象可用通常环境下随机粒子的运动来描述, 例如, 花粉在水中的运动(布朗运动)可用随机游动(RW)模型描述<sup>[1]</sup>; 对于通常环境, 粒子游动的空间是各向同性的, 随机粒子在不同方向游动的迁移概率皆相同。但是在现实世界中, 许多随机现象非常复杂, 不能用简单随机游动模型描述, 它们所处的环境是各向异性的, 很多约束条件影响着粒子的运动, 而粒子的运动又使环境发生改变。当前研究的随机游动模型达十几种之多<sup>[2-5]</sup>, 本文研究了一种受主方向、格点的吸引、排斥等因素制约的自回避随机游动算法实现, 并利用该算法实现了对曲流河河道形态的建模。

## 1 随机游动模型建立

设一个平面连通区域  $D$ ,  $G$  为这个连通区域上的一个矩形网格图, 令  $n_x, n_y$  分别表示  $x$  方向和  $y$  方向的网格数, 则  $G = (V, E)$  是一个网格数为  $n_x \times n_y$  的离散化网格图,  $|V| = (n_x + 1) \times (n_y + 1)$ ,  $|E| = (n_x + 1) \times n_y + n_x \times (n_y + 1)$ 。  $dx, dy$  分别表示  $x$  方向和  $y$  方向的网

格跨度, 即网格的长和宽, 并把  $dx, dy$  分别称为  $x$  和  $y$  方向的步长。将随机游动者  $M$  看成一个质点,  $M$  的起始点在网格内随机产生, 它沿着网格结点  $v_i$  进行迁移, 每次只能以一个步长迁移, 分别用  $R_i, U_i, L_i, D_i$  来表示向右, 向上, 向左, 向下四个方向游动一步后的节点, 如图 1 所示。

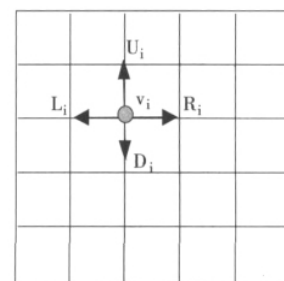


图 1 游走者随机行走示意图

由于有限制条件的约束, 这四个方向的迁移概率是不相等的。迁移概率由迁移到下一个节点的方向和主方向的夹角, 自回避因素, 吸引或排斥因素三个特征项确定, 分别用  $F, G, H$  来表示这三个特征项, 用  $\alpha, \beta, \gamma$  表示这三个特征项的权重。节点  $v_i$  的四个一步行走迁移方向可表示为  $R_i(F, G, H), U_i(F, G, H), L_i(F, G, H), D_i(F, G, H)$ 。若权重分别为  $\alpha_i, \beta_i, \gamma_i$ , 则这四个迁

收稿日期: 2011-08-24

作者简介: 田金琴(1978-)女, 宁夏银川人, 讲师, 硕士, 主要从事嵌入式软件开发及计算机图形图像处理方面的研究 (E-mail) tiqding@126.com

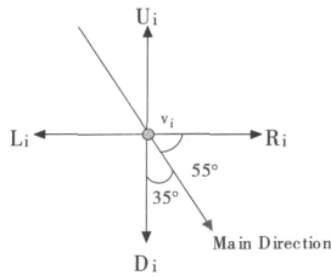


图 2 受主方向影响的随机行走示意

移方向又可分别表示为  $R_i(F_i, \alpha_i; G_i, \beta_i; H_i, \gamma_i)$ ,  $U_i(F_i, \alpha_i; G_i, \beta_i; H_i, \gamma_i)$ ,  $L_i(F_i, \alpha_i; G_i, \beta_i; H_i, \gamma_i)$ ,  $D_i(F_i, \alpha_i; G_i, \beta_i; H_i, \gamma_i)$ , 简记为  $R_i(\alpha_i; \beta_i; \gamma_i)$ ,  $U_i(\alpha_i; \beta_i; \gamma_i)$ ,  $L_i(\alpha_i; \beta_i; \gamma_i)$ ,  $D_i(\alpha_i; \beta_i; \gamma_i)$ 。例如假设节点  $v_i$  向右迁移一步受上述特征因素影响的权重分别为 0.28, 0, 1, 则可表示为  $R_i(0.28, 0, 1)$ 。

下边分别讨论受主方向影响, 自回避, 吸引和排斥四个特征条件约束下的随机游动模型。

### 1.1 主方向因素

不受条件约束的随机游动容易走出团状图样, 为了模拟河流状线形的效果, 给行走者设定一个行走主方向的限制。实质上就是给行走者限定了一个大致的方向, 让其在总目标是沿主方向行走的约束条件下进行随机行走, 如上图 2 所示。特征项  $F$  的权重  $\alpha$  由  $R, U, L, D$  和主方向的内夹角的大小关系确定, 呈现递减的线性关系, 如表 1 所示:

表 1 受到主方向影响时的权重情况

方向	$\vec{V}_0 \vec{V}_i$ 与主方向的夹角 $(i)$	$i / ((i) - 1)$	$1 - (i / ((i) - 1))$	权重 $\alpha = (1 - (i / ((i) - 1))) / 3$
R	55	0.152778	0.847222	0.282407
U	145	0.402778	0.597222	0.199074
L	125	0.347222	0.652778	0.217593
D	35	0.097222	0.902778	0.300926
合计	360	1	3	1

### 1.2 自回避因素

进一步加强条件限制, 在受主方向因素影响的条件下, 为了避免游动者游动时返回到原来游动过的位置, 规定游动者不能迁移到已经过的节点。具体做法是当游动者进行自回避随机游动时, 在网格结点上设置布尔计数器, 没有经过的节点记为 0, 否则记为 1, 游动者进行下一步迁移时首先判断下一个节点上计数器的值, 若为 0, 则可以迁移过去, 否则游动者就回避该节点, 重新随机选取另外的节点进行同样的判断, 依次进行迁移游动。因此特征项  $G$  的权重  $\beta$  只有两个值 0 或 1。

$$\beta(v_i) = \begin{cases} 0, & v_i \text{ 为自回避节点} \\ 1, & \text{否则} \end{cases}$$

### 1.3 吸引和排斥因素

在实际应用的过程中, 网格中的某些节点的属性值是已知的, 期望随机游动的路线中必须通过某些点或者避开某些点, 这就是吸引或排斥因素。设  $V_A$  为包含吸引因素的网格节点的集合,  $V_E$  为包含排斥因素的网格节点的集合,  $V_N$  为非排斥、非吸引节点的集合, 则  $V_A \cup V_E \cup V_N = V$ , 且  $V_A, V_E, V_N$  两两不相交。

$$\gamma(v_i) = \begin{cases} 999, & v_i \in V_A \\ 1, & v_i \in V_N \\ 0, & v_i \in V_E \end{cases}, i = 1, 2, \dots, |V|$$

### 1.4 综合因素

上述 3 个特征因素由已知条件分配权重之后, 网格上的所有节点和当前点的关系可以用如下形式的表 2 来表示。

表 2 各特征项权重情况

特征项权重 迁移方向	$\alpha$	$\beta$	$\gamma$
R	0.282407	1	0
U	0.199074	0	1
L	0.217593	1	999
D	0.300926	1	1

表 2 中  $\alpha$  的值按照上述的规则确定,  $\beta$  和  $\gamma$  根据邻接节点的属性来取值。则向 R 方向转移的概率为  $p_R =$

$$\frac{\alpha_R \beta_R \gamma_R}{\sum_{R, U, L, D} \alpha \beta \gamma}$$

$$p_U = \frac{\alpha_U \beta_U \gamma_U}{\sum_{R, U, L, D} \alpha \beta \gamma}, p_L = \frac{\alpha_L \beta_L \gamma_L}{\sum_{R, U, L, D} \alpha \beta \gamma}, p_D = \frac{\alpha_D \beta_D \gamma_D}{\sum_{R, U, L, D} \alpha \beta \gamma}$$

下边举一种情况来说明, 如图 3 所示, 设行走者 M 当前点  $V_i(x_i, y_j)$ , 其相邻的四个邻接点  $V_{i1}(x_i + 1, y_j)$ ,  $V_{i2}(x_i, y_j - 1)$ ,  $V_{i3}(x_i - 1, y_j)$ ,  $V_{i4}(x_i, y_j + 1)$  称之为可迁移节点。主方向  $\vec{V}_i \vec{M}$  与  $\vec{v}_i v_{i1}, \vec{v}_i v_{i2}, \vec{v}_i v_{i3}, \vec{v}_i v_{i4}$  四个方向的夹角分别为 55、145、125、35。

设  $V_{i2}$  为当前节点  $V_i(x_i, y_j)$  的前任节点, 则  $V_{i2}$  是自回避节点, 即权重  $\beta_U = 0$ , 设  $V_{i1} \in V_E$  为排斥点,  $V_{i3} \in V_A$  为吸引点,  $V_{i4} \in V_N$  为非吸引非排斥点, 则  $\gamma_R = 0, \gamma_L = 999, \gamma_D = 1$ 。表 2 所示即为所讨论情况的各种特征项的权重。

将各值代入概率公式可求出  $p_R = 0, p_U = 0, p_L = 0.9986, p_D = 0.0014$ 。

即当前节点  $V_i(x_i, y_j)$  向左迁移一步的概率为 0.9986, 向下迁移一步的概率为 0.0014。

## 2 计算机模拟实现

完成一次随机游动之后, 记录下随机游动的节点顺

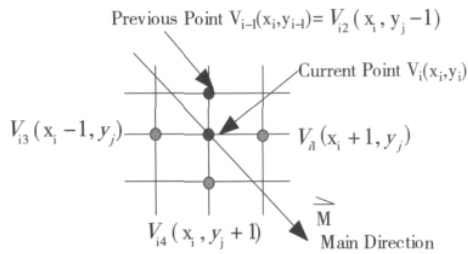


图3 当前点受综合因素影响示意图

序并记录下游动路径便构成了随机游动路线。计算机模拟就是要描绘出这样一条游动的轨迹线,并对其曲线拟合,最后将拟合后的曲线加宽并光滑处理,实现对自然界中曲流河河型的模拟。

### 2.1 算法描述

依据上面的论述,算法描述如下:

Step1: 将连通区域网格化,依据样本点数据确定网格节点处的属性。

Step2: 在网格中随机选出一个格点,将该点定义为起始点,并标记起始点为当前节点,记录游动的路径。

Step3: 计算邻接网格节点的迁移概率,依迁移概率进行随机游动,迁移到下一个网格节点,并将迁移后的节点赋为当前节点,记录游动路径。

Step4: 若当前节点满足结束条件,则完成一次游动。否则转向 Step3。

说明 游动的结束条件为当前点为目的点,或者不是目的点,但是根据自回避算法限定已经无路可走。由于计算机模拟的随机性是一种伪随机,随机种子相同时,产生的随机序列也相同,在进行模拟时随机种子可以指定或随机产生。

### 2.2 代码实现

```
Grid gridForRandomWalk; //定义游动网格
Grid gridForDisplay; //屏幕显示网格
Float mainDirect; //游动主方向参数
initGrid ( out gridForRandomWalk , out gridForDisplay ); //初始化网格区域
.....
Data [ ]samples = initSamples ( ) ; //定义随机种子序列
int seed = Config. isAutoSeed? Config. autoSeed: Config. manuSeed;
this. view. setRandomSeed ( this. view. randomWalk. seed ); //指定随机种子是系统自动生成或者由操作者给定
```

```
GridConvertor. Convert ( this. view. randomWalk. startIndexX , this. view. randomWalk. startIndexY , gridForRandomWalk , gridForDisplay , ref this. view. startIndexX , ref this. view. startIndexY ); // 屏幕显示网格与实际网格转换
.....
this. view. ReadPointFromFile ( " VirtualPoint.txt" ); //记录权重格点坐标的文件
this. view. randomWalk. GenerateFootmarks ( Config. randomWalkType ); //生成随机游动路径,记录坐标信息
this. view. Bending ( ref this. view. randomWalk. footmarksList ); 曲线光滑
.....
GridConvertor. Convert ( this. view. randomWalk. footmarksList , gridForDisplay , ref this. view. gridNodesForRandomFootmarks ); //显示网格与实际网格转换
this. view. writeFluvialGridFile ( " FluvialGridWell.txt" ); //路径坐标信息写入文件
Widden ( gridForDisplay ); //加宽处理
this. view. Reshape ( ) ; //OpenGL 视图显示
this. view. Invalidate ( ) ; //视图刷新
.....
```

```
domWalk , gridForDisplay , ref this. view. startIndexX , ref this. view. startIndexY ); // 屏幕显示网格与实际网格转换
.....
```

```
this. view. ReadPointFromFile ( " VirtualPoint.txt" ); //记录权重格点坐标的文件
this. view. randomWalk. GenerateFootmarks ( Config. randomWalkType ); //生成随机游动路径,记录坐标信息
this. view. Bending ( ref this. view. randomWalk. footmarksList ); 曲线光滑
.....
GridConvertor. Convert ( this. view. randomWalk. footmarksList , gridForDisplay , ref this. view. gridNodesForRandomFootmarks ); //显示网格与实际网格转换
this. view. writeFluvialGridFile ( " FluvialGridWell.txt" ); //路径坐标信息写入文件
Widden ( gridForDisplay ); //加宽处理
this. view. Reshape ( ) ; //OpenGL 视图显示
this. view. Invalidate ( ) ; //视图刷新
.....
```

### 2.3 模拟效果

在.NET框架下的C#环境中,基于CsGL图形库实现了算法的编程及可视化<sup>[6]</sup>。效果图如图4和图5所示。红色点表示吸引点,蓝色点表示排斥点,红色折线根据上述算法是随机产生的一条游动者路径。图5显示了系统对游动者路径进行进行了曲线拟合并加宽处理后的模拟效果。实际效果和曲流河形态非常相似。

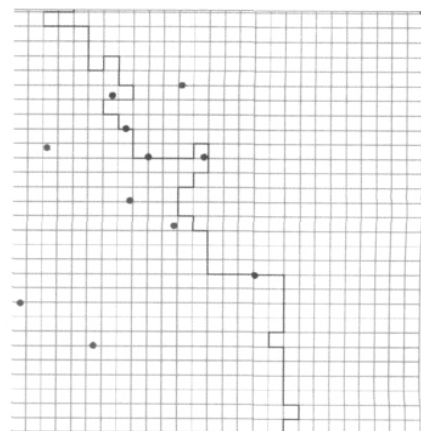


图4 综合因素影响下的一次随机行走的实现

## 3 结论

通过对普通的基于网格自回避随机游动方法进行改进,增加了主方向,节点的吸引、排斥等约束条件来影

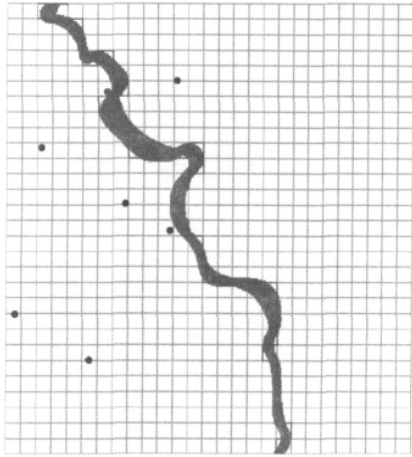


图 5 对行走路径进行采样, 曲线拟合并加宽处理后的效果图

响游动者的响迁移概率, 通过记录游动路径, 并对游动路径进行采样, 曲线拟合, 加宽后等图形化处理方法后实现对曲流河河型、海岸线等自然界具有蜿蜒曲折的形态物体的建模。结果表明, 该方法不仅能够很好的体现物体原有的形态, 又能随机的模拟出大量的, 满足人们

已掌握的对物体认识数据的形态结果。

#### 参考文献:

- [1] 蒋应琅. 随机过程原理与生命科学模型 [M]. 上海: 上海翻译出版公司, 1987.
- [2] Hughes B D. Random Walks and Random Environments [M]. Oxford: Clarendon Press, 1955.
- [3] Huang S Y, Zou X W, Zhang W B, et al. Random walks on a  $(2+1)$ -dimensional deformable medium [J]. Phys. Rev. Lett 2002 69(9): 2900-2904.
- [4] Vanderzande C, Daerden F. Dissipative Abelian sandpiles and random walks [J]. Phys. Rev 2001 63: 2301-2304.
- [5] Wang F, Landau D P. Efficient, Multiple-range random walk algorithm to calculate the density of states [J]. Phys. Rev. Lett 2001 86: 2050-2053.
- [6] 鲜乾坤, 万宁, 贺全兵. 基于 Kriging 插值的 3D 地形可视化 [J]. 四川理工学院学报: 自然科学版, 2009, 22(6): 47-49.

## Research of Snaking Stream Modeling Based on Self-avoiding Random Walk Algorithm

TIAN Jin-qin<sup>1</sup>, LI Zhi-yong<sup>2</sup>

- (1. Computer Science and Engineering College of North University for Nationalities, Yinchuan 750021, China;
2. Dept. of Math and Physics, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450005, China)

**Abstract:** A restricted self-avoiding random walk algorithm based on grid nodes is studied, and is applied to achieving snaking stream-type modeling. The constraint conditions of affecting the walker include the main direction, the node absorption and the repulsion. When one random walk completes, the passed grid nodes must be recorded in file. Using spline interpolation of the grid point to path curve fitting, and by widening and smoothing, the nature of snaking stream river-type can be simulated. This system is designed and developed by C program language and the OpenGL graphics library. The winding shape objects such as the coastline can be modeling under limited conditions by this method.

**Key words:** self-avoiding walking; characteristic item; migration probability