

8-13-2020

## Multi-agent Path Planning mACO Algorithm in Web3D Mountain Scene

Fengting Yan

*Tongji University, Shanghai 201804, China;*

Jinyuan Jia

*Tongji University, Shanghai 201804, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

## Multi-agent Path Planning mACO Algorithm in Web3D Mountain Scene

### Abstract

**Abstract:** There is amount of data in a mountain scene, and the path planning algorithm in it is very complex, so it is not impossible to be shown detailed in Web. Usually the common potential path planning using contours can not find an optimal path because the path is easily cut under the cliff. To solve the above problems, a *mACO (mountain ACO) path planning algorithm was addressed for the Web3D application, a planar grid ACO path planning algorithm and a A\* path planning were completed in Web3D environment*. Then a typical battle scene case was used for the three kinds of algorithms to be compared in effectiveness, efficiency, and FPS. The result can be seen that all the three kinds of algorithms are real-time, but the mACO algorithm is more accurate than the others. *Based on the optimal path and using leader-follower idea, a real-time effective multi-agent path planning is implemented.*

### Keywords

Web3D mountain scene, mACO (mountain ACO) algorithm, planar grid ACO algorithm, A\* algorithm, multi-agent path planning

### Recommended Citation

Yan Fengting, Jia Jinyuan. Multi-agent Path Planning mACO Algorithm in Web3D Mountain Scene[J]. Journal of System Simulation, 2016, 28(10): 2312-2320.

# Web3D 山地场景中多智能体 mACO 路径规划算法

闫丰亭, 贾金原

(同济大学, 上海 201804)

**摘要:** 山地场景数据量大, 路径规划算法复杂, 难以在网页上精确显示, 通常采用的基于等高线的势能路径规划, 往往得不到最优路径, 且容易被隔断在悬崖下面。为解决以上问题, 提出并实现了 Web3D 上的 mACO (mountain ACO) 路径规划算法, 并在 Web3D 上实现了基于平面网格的 pgACO (planar grid ACO) 路径规划算法, 以及一个 Web3D 上的 A\* 路径规划算法。再以典型战斗场景为案例, 针对 mACO 算法、pgACO 算法以及 A\* 算法, 就实现效果、效率、网页刷新率(FPS)做了对比实验, 结果显示, 三种算法均可达到实时性, 但 mACO 算法规划的路径更加精确。最后根据规划出来的最优路径, 采用 leader-follower 思想, 在 Web3D 上实现了实时高效的多智能体路径规划方案。

**关键词:** Web3D 山地场景; mACO 算法; 平面网格 pgACO 算法; A\* 算法; 多智能体路径规划

中图分类号: TP391

文献标识码: A

文章编号: 1004-731X (2016) 10-2312-09

## Multi-agent Path Planning mACO Algorithm in Web3D Mountain Scene

Yan Fengting, Jia Jinyuan

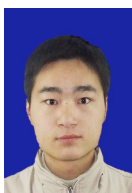
(Tongji University, Shanghai 201804, China)

**Abstract:** There is amount of data in a mountain scene, and the path planning algorithm in it is very complex, so it is not impossible to be shown detailed in Web. Usually the common potential path planning using contours can not find an optimal path because the path is easily cut under the cliff. To solve the above problems, a mACO (mountain ACO) path planning algorithm was addressed for the Web3D application, a planar grid ACO path planning algorithm and a A\* path planning were completed in Web3D environment. Then a typical battle scene case was used for the three kinds of algorithms to be compared in effectiveness, efficiency, and FPS. The result can be seen that all the three kinds of algorithms are real-time, but the mACO algorithm is more accurate than the others. Based on the optimal path and using leader-follower idea, a real-time effective multi-agent path planning is implemented.

**Keywords:** Web3D mountain scene; mACO (mountain ACO) algorithm; planar grid ACO algorithm; A\* algorithm; multi-agent path planning

## 引言

“Internet+”和“VR+”的到来, 使得传统 3D 技术和机器学习算法更加广泛地应用于 Web3D



收稿日期: 2016-05-31 修回日期: 2016-07-14;  
基金项目: 国家自然科学基金(61272270);  
作者简介: 闫丰亭(1980-), 男, 山东, 博士生, 研究方向为虚拟现实与机器学习; 贾金原(1963-), 男, 内蒙古, 博士, 教授, 研究方向为分布式虚拟现实、Web3D、游戏引擎。

上, 尤其是在大场景和大数据领域上的应用更是当前的一个研究热点。山地场景的几何面片数量众多, 地形网格模型数据、地表植被网格模型数据、树木网格模型数据量大, 使得大场景山地场景在 Web 上加载和渲染比较困难, 尤其是加入了丰富的场景语义信息, 就会占用更多的内存、GPU 和 CPU 资源。而网页的计算能力相对本机计算能力弱, 手机移动终端上网页的计算能力更低, 这就使得

<http://www.china-simulation.com>

• 2312 •

Web3D 上的山地场景的加载与渲染存在严重的困难, 造成场景显示难以达到实时性。

而基于山地场景的多智能体的路径规划, 在网页上的实现也存在诸多困难: 1) 实时性路径规划计算; 2) 找到一条最优路径; 3) 实现多智能体沿最优路径实时行走的行为可视化。

针对这些困难, 本文选用了—个典型大场景山地场景-黄洋界山地场景。基于卫星灰度图进行该山地场景的建模, 使用 Egret.ts 加载该山地模型, 借助射线碰撞检测到山地模型表面, 获得整个山地场景的表面高度数据  $y$ , 从而实现山地地面坐标和平面网格坐标一一映射对应, 将 3D 地形和 2D 平面网格形成投影对应关系。本文采用了三种算法针对山地场景进行路径规划。无论是 A\*路径规划算法还是基于平面网格的 ACO 路径规划算法<sup>[1]</sup>, 或者本文提出的 mACO 路径规划算法<sup>[2]</sup>, 都是对传统基于等高线势能的路径规划的改进, 因为本文中的路径规划解决了基于等高线势能路径规划全局不是最优, 而又容易陷入到被悬崖阻断无路可走的情况, 从而确保只要有路可走, 则一定会寻找到一条最优路径的方案。

由于本文使用了智能寻路算法, 使得局部路径规划可以实时执行, 不需要一定知道全局场景数据, 就可以找到一条最优路径。由于场景较大, 所以对大规模场景进行了分割, 分割是基于地理位置进行的, 对于发生蓝军和红军战斗的场景数据, 进行了单独的分割, 便于针对该部分的山地场景数据进行解析、读取、加载完毕后进行渲染。

由于 A\*算法成熟且应用广泛, 例如在 Unity3D 上的使用, 在魔兽争霸游戏中的使用。所以, 本文在实际实验中, 以 A\*的路径规划为标准, 检验本文提出的 mACO 算法的实现情况, 并使用 Egret.ts 技术实现了平面网格上的 ACO 算法。通过对多种算法实现的路径规划, 进行对照, 从而检验本文提出来的新算法是否符合实际需求。

## 1 背景

由于 HTML5 标准的制定, 基于网页上的 3D 显示就变得切实可行并可被广泛普及。事实上 Flash3D 早在 2011 年就已经在网页上实现了栩栩如生的 3D 效果, 用户可以与虚拟场景进行灵活的交互操作。然而, Flash3D 并不支持 HTML5 格式, 虽然其网页插件可以进行便捷更新, 但依然避免不了“免插件”更新操作, 这为用户的 Web3D 体验带来糟糕的影响。而 HTML5 标准被制定后, 众多第三方 Web3D 插件迅速涌现, 其中 Egret.ts 相对而言是非常成熟, 功能也很齐全, 图形绘制效率在具有运行时网页上也是最高, 其用户量、社区论坛很多, 技术支持力度非常大, 长期处于热更新状态的技术框架。该框架是基于 WebGL 标准技术, 经过封装打包 WebGL 原生 API 的细节, 形成的可方便使用的 Web3D 图形开发框架。

而 ACO 算法, 是意大利学者 Dorigo 在 1991 年提出的一种模拟自然界中蚁群觅食行为的模拟进化算法, 即人工蚂蚁算法, 简称蚂蚁算法(Ant Colony Optimal Algorithm-ACO), 该算法具有分布计算、信息正反馈和启发式搜索的特征<sup>[1]</sup>。智能体路径规划问题, 就是要在某些确定的或者优化目标(最短路径<sup>[3]</sup>、最低的能量消费、或最短的时间内)下, 在存在障碍物的环境中, 规划出来一条从起点到终点的最优路径<sup>[4]</sup>。

智能路径规划过程包括两个步骤: 环境创建和规划方法。在环境创建模型中, 这些路径规划可以分为全局路径规划和局部路径规划两部分<sup>[5]</sup>。全局路径规划包括: 空间分配方法, 广义锥方法, 顶点图像法, 网格滚动法。局部路径规划人们通常采用人工势场法进行路径规划。已有很多种智能路径规划的方法是基于空间模型的, 例如 A\*算法, 蚁群算法, 粒子群算法, 遗传算法以及神经网络算法<sup>[4-7]</sup>。

智能路径规划算法主要是基于搜索的几何模型, 虚拟势场, 导航函数和数学优化方法等。存在各种各样的多边形, 如果两个多边形的顶点的连接

不会穿过障碍物，我们就采用该连接线来解决候选路径的分段问题，在这里称之为可视方法<sup>[8]</sup>。Kuwata 和 How 已经将该方法应用到了无人机路径规划中。可视方法的缺陷就是通常得到的并不是优化路线<sup>[9]</sup>。人工势场是基于虚拟势场和导航函数的，这在二维空间里得到了广泛应用。其优势是算法简单，执行效率高。Kitamura 等人提出了一种 3D 路径规划方法，该方法是在人工势场模型建立的情况下，使用八叉树来表示智能体的三维配置空间。

支持向量机(SVM)<sup>[10]</sup>是一种经典的路径规划方法，该方法基于数学优化。其思想是使用间隔最大化规则解决支持向量机(SVM)的分类问题，从而得到非线性积分面，然后从分离出来的面中选出适合的聪明的移动路径<sup>[11]</sup>。

基于智能路径规划的方法主要包括：受生物遗传学启发的遗传算法，受鸟群、鱼群启发的粒子群算法，以及受人的大脑神经系统启发神经网络算法<sup>[12-13]</sup>。基因算法收敛速度快，解的质量高，会有一个好的初始解。但是，其最大的缺陷是不能满足实时的需要。粒子群算法(PSO)在已知的局部路径规划或未知的三维路径规划中是准三维路径规划<sup>[14]</sup>。神经网络算法在解决高非线性和不确定系统控制方法中潜力巨大，但是该算法需要规划大量的障碍物，并随着障碍物数量的增加，计算复杂度不断增加<sup>[15]</sup>。

## 2 山地场景数据映射到 2D 平面网格

井冈山黄洋界，是典型的山地场景，以该山地场景作为研究对象具有非常高的实际价值与意义，尤其是满足用户在该山地战场战斗中的沉浸式体验，具有重要的作用。为了确保该研究的价值与意义，通过井冈山红色旅游博物，获得了该山地场景的 3D 地形模型，这个模型是通过卫星云图获得的数据，进行三维重建而获得。

我们使用 Egret.ts 框架，通过模型导入函数将该山地场景模型导入到 Web3D 开发平台，使用射

线检测方法，获得山体表面的高度值。具体数据采集如下：

首先根据黄洋界战场东西南北近似一个 210 m×210 m 的走俏山坡地况，将地形精确度限定在 1 m×1 m。然后，采用 planar grid 的方法，映射山地地形数据到平面网格。在平面网格交点上，记录对应山体表面地理位置信息以及该位置上的危险信息。图 1 是山地场景数据映射到平面网格上的数据情况。

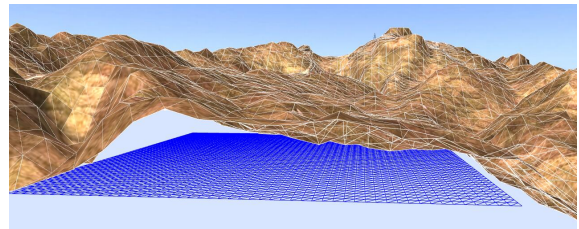


图 1 一个 210 m×210 m 的山地

山地地表信息映射到平面网格上，实现 3D 场景数据在 2D 平面上的映射，映射的内容包括地理信息和语义信息。地理信息是 $(x, y, z)$ ，语义信息包括(竹钉阵、壕沟、荆棘丛)。

具体算法步骤是：

Step 1: 定义平面网格对象，定义向上的射线方向；

Step 2: 射线位置初始化，从网格的交叉点射出，定义射线和地形相交变量： $yPos: number = 0$ ；

Step 3: 开始启动射线检测函数，将射线与地形网格相交的结果放到 result 里；

Step 4: 从 result 里面提取 yPos 值以及语义信息；

Step 5: 形成 2D 平面节点和 3D 山体地表对应点的 Mapping，并存进坐标及语义信息，算法执行结束。

该算法的伪代码描述如下：

```
Begin()
{
    初始化网格对象；
    初始化射线位置、方向；
    初始化交点高度；
```

```

    If(射线和模型相交)
    {
        获得相交结果;
        从结果中提取出相交的高度值;
        从结果中提取出相交点的语义信息;
    }
    返回相交节点 3D 坐标(x, y, z);
    返回相交节点处语义信息;
}
End

```

经过以上工作,可获得山地地表的精确坐标数据,同时可以获得山地地表与平面二维网格对应的关系,即实现了 2D 与 3D 间的 Mapping 关系。

### 3 mACO, pgACO 和 A\*算法路径规划

路径规划方法众多,其在虚拟现实以及游戏场景中应用广泛,其中 A\*算法因其计算方法简单,效果逼真,能够满足一般场景中路径规划的需要,是被采用最多的一种路径规划算法。然而 A\*算法最优路径的误差不可避免,且不易被改正。而 ACO 算法,可以利用提高蚂蚁的数量以及迭代的次数,借助信息素的释放与蒸发,逐渐找到一条最优路径,利用并行寻路方式。然而 Web3D 上 ACO 算法的路径规划,既要具有实时性,又要确保寻路结果的精确性。考虑到这两个矛盾的需求,本文提出了在一秒内规划出一条精确的最优路径算法:mACO 算法。

#### 3.1 mACO 算法的路径规划

mACO 路径规划算法是面向 Web3D 的实时路径规划,该算法在实时( $t < 1$  s)路径规划标准前提下,考虑到寻路过程中避障、避开悬崖峭壁、躲过竹钉阵、绕过壕沟和荆棘丛,使用 3D 场景中欧式距离的倒数作为一个启发值,对于障碍位置、悬崖峭壁、竹钉阵、壕沟和荆棘丛,均设置为不可行区域,并放入到禁忌表 `tabuList` 中。在可通行区域里,进行蚁群路径规划,将寻找到的路径与 A\*算法寻找到的路径、基于平面上的 ACO 算法寻到的路径

作比较,在时间耗费、内存占用量、CPU 占用率和 FPS 四个维度上做了对比分析。

#### 3.1.1 mACO 算法

模型定义:规划空间范围为 210 m×210 m 的山地,其中平面上的 x 轴, y 轴方向每个节点的间距为 1 m,纵向方向上的 z 轴节点间的距离由射线检测得到。路径起点在规划空间的序号为[1 10 4],终点在规划空间的序号是[210 4 5]。设置算法的基本参数为:种群规模是 20,算法迭代是 150 次。

算法流程图:首先获得 Web3D 山体场景中地形的数据,考虑到每个节点的位置信息,语义信息,开始 mACO 路径规划,从起点开始,依次走向下一个位置,搜索终点位置,判断是否到达终点,如果到达终点,那么所走过的所有节点均需要进行一次信息素更新;如果没有到达终点,则继续搜索下一个位置,直到本次迭代完成,在更新完毕信息素之后,判断是否完成了全部的迭代次数,如果还没有完成全部的迭代次数,则需进入到下一代 mACO 算法,直到完成所有的迭代,获得一条最优路径。整个算法的流程如图 2 所示。

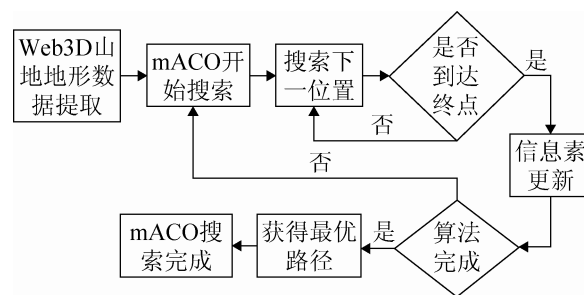


图 2 mACO 算法流程图

信息素的更新:首先将整个搜索空间离散为一系列的三维离散点,这些离散点为蚁群算法需要搜索的节点。而信息素就存储在模型的离散点中,每个离散点有一个信息素的值,值的大小代表对蚂蚁的吸引程度,各点信息素在每只蚂蚁经过后进行更新,更新的方式有局部更新和全局更新。局部更新是指当蚂蚁经过该点时,该点的信息素就减少,局部更新的目的是为了增加蚂蚁搜索未经过点的概

率, 达到全局搜索的目的<sup>[1]</sup>。局部信息素更新随着蚂蚁的搜索进行, 信息素更新公式为:

$$\tau_{ijk} = (1 - \zeta)\tau_{ijk} \quad (1)$$

其中:  $\tau_{ijk}$  为点  $(i, j, k)$  上所带的信息素值;  $\zeta$  为信息素的衰减系数。

全局路径规划是指, 当蚂蚁完成一条路径的搜索时, 以该路径的长度作为评价值, 从路径集合中选出最短路径, 增加最短路径各节点的信息素值, 信息素更新公式如下:

$$\tau_{ijk} = (1 - \rho)\tau_{ijk} + \rho\Delta\tau_{ijk} \quad (2)$$

其中:  $\rho$  为信息素系数;  $K$  是系数。

改进的启发函数: 根据启发函数来计算可视区域内各点的选择概率, 启发函数为:

$$H_{(i,j,k)} = D_{(i,j,k)}w_1 \times S_{(i,j,k)}w_2 \times Q_{(i,j,k)}w_3 \times Z_{(i,j,k)}w_4 \quad (3)$$

其中:  $D_{(i,j,k)}$  为两点间路径长度, 促使蚂蚁选择距离较近的点;  $S_{(i,j,k)}$  为安全因素, 当选择点不可到达时, 该值为零, 促使蚂蚁选择安全点, 当遇到悬崖、竹钉阵、壕沟、荆棘丛或火力点的时候, 作为不可行区来处理;  $Q_{(i,j,k)}$  为下一点到目标点的路径长度, 促使蚂蚁选择距离目标最近的点;  $Z_{(i,j,k)}$  为红军火力点区, 如果下一点没有红军的火力点, 则  $w_4$  为零。  $w_1$ ,  $w_2$ ,  $w_3$  为系数, 代表上述各因素的重要程度。

$D_{(i,j,k)}$  的计算公式如下:

$$D_{(i,j,k)} = \text{sqrt}[(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2] \quad (4)$$

其中:  $a$  为当前点,  $b$  为下一点。

$S_{(i,j,k)}$  的计算公式如下:

$$S_{(i,j,k)} = (\text{Num} - \text{Unum}) / \text{Num} \quad (5)$$

其中:  $\text{Num}$  表示在面中可视点的数量;  $\text{Unum}$  表示可视点中不可达区域的点的数量。

$Q_{(i,j,k)}$  的计算公式如下:

$$Q_{(i,j,k)} = \text{sqrt}[(x_b - x_d)^2 + (y_b - y_d)^2 + (z_b - z_d)^2] \quad (6)$$

其中:  $b$  表示下一点,  $d$  表示目标点。

$Z_{(i,j,k)}$  的计算公式如下:

$$Z_{(i,j,k)} = \begin{cases} M(m_1, m_2, \dots, m_r), m_r \text{ 是危险数值} \\ 1, \text{ if } w_4 = 0 \end{cases} \quad (7)$$

在蓝军攻击中, 最短的攻击路线不一定是最优的攻击路线, 因为还要考虑攻击路线上是否存在障碍区或者火力点区。这些障碍区和火力点区是需要考虑进来, 才能确保规划出来的路径更符合实际情况。

### 3.1.2 基于 mACO 算法的路径规划

#### 1) 蚁群搜索的步骤

蚂蚁在平面  $\Pi_1$  上的当前点  $p_i$  选择平面上的下一点  $p_{i+1}$  的步骤如下:

Step 1: 根据环境可视化数据确定下一个平面里可行点集。当前的点是  $i$ ;

Step 2: 基于公式 6, 依次计算点  $p_i$  到下一平面内可行点集合的启发值  $H(i+1, u, v)$ 。

Step 3: 计算下一平面内任一点的选择概率  $P_{(i+1, u, v)}$ :

$$P_{(i+1, u, v)} = \begin{cases} \frac{\tau_{\alpha+1, \mu, \nu} H_{\alpha+1, \mu, \nu}}{\sum \tau_{\alpha+1, \mu, \nu} H_{\alpha+1, \mu, \nu}}, & \text{可行点} \\ 0, & \text{不可行点} \end{cases} \quad (8)$$

其中,  $\tau_{\alpha+1, \mu, \nu}$  为平面  $\Pi_{11}$  上点的信息素值。

Step4: 根据各点选择概率, 我们使用轮盘赌法选择下一平面内的点。

基于蚁群优化的三维路径规划算法:

Begain()

```
{
    蚂蚁个数初始化;
    迭代次数初始化;
    信息素的初始化;
    If( 完成了一次迭代&运行没有结束)
    {
        计算最优适应度值;
        得到最优适应度值;
        从走过的路径里, 选择最优的路径;
        信息素更新;
    }
}
```

### 3.1.3 路径规划结果及行走场景

从整个山地场景截取部分陡峭山地地形数据, 使用 Egret.ts 绘图函数, 构建 21 m×21 m 场景, 进

行 mACO 算法的路径规划, 实验结果如图 3 所示。

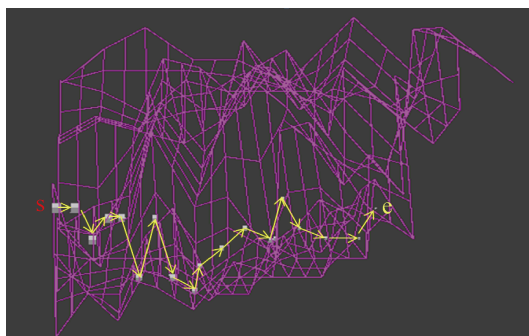


图 3 Web3D 山体路径规划

从上图可以看到规划出的路径是沿着箭头方向, 从起点  $s$  到终点  $e$ 。为了进一步的测验该算法的实时性, 采用了规模为  $21\text{ m} \times 21\text{ m}$  的场景进行的路径规划, 耗时  $0.178\text{ s}$ , 网页刷新频率是  $55\text{ FPS}$ 。将该算法应用到完整的井冈山场景中路径规划, 得到的结果如图 4 所示。

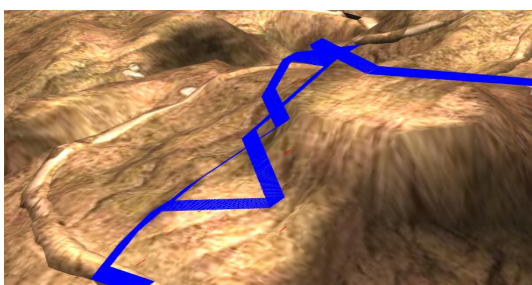


图 4 mACO 规划的路径

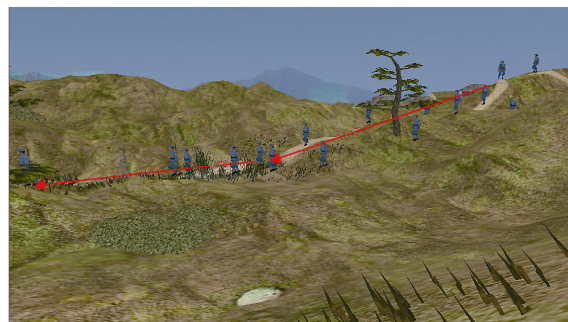
图 5(a)是规划出最优路径后, 蓝军虚拟化身基于 leader-follower 规则<sup>[16]</sup>, 沿路径在黄洋界上行进的场景。

图 5(b)是置换了场景贴图后利用 mACO 进行的路径规划, 虚拟化身沿该路径在基于 leader-follower 规则下行走的场景。

Leader-follower 思想是通过一人进行 mACO 算法的路径规划, 而其它人群(Followers)只需有序跟随 Leader 就可以向着目的地行进, 这样就避免了每个虚拟化身的寻路计算, 大大降低了 mACO 的计算时间复杂度。



(a) 场景 1



(b) 场景 2

图 5 基于 mACO 的 leader-follower 人群行为

### 3.2 pgACO 算法的路径规划

pgACO 算法是平面网格上的 ACO 算法的缩写, 该算法是基于平面网格上 ACO 路径规划, 不是一个真正的 3D ACO 路径规划算法, 寻路准确度较 mACO 低, pgACO 算法是基于图 1 的水平网格, 根据用户鼠标点击的终点, 抽取出  $x_{end}$  坐标和  $y_{end}$  坐标, 同时抽取出虚拟化身所在位置的  $x_{start}$  坐标和  $y_{start}$  坐标, 开始计算从虚拟化身到终点的最优路径规划。

pgACO 算法: 在 Web 上实现的基于平面网格的 ACO 路径规划, 该算法不考虑  $y$  值的影响, 其距离计算公式简化为  $D(i, j)$ :

$$D(i, j) = \text{sqr}t[(x_a - x_b)^2 + (y_a - y_b)^2] \quad (9)$$

对于不可达区, 也需要在二维平面上给出表示, 并放入到禁忌表中, 在路径规划时, 不选择禁止区域行走。

pgACO 算法的路径规划: 图 6 是基于 pgACO 算法的路径规划结果。



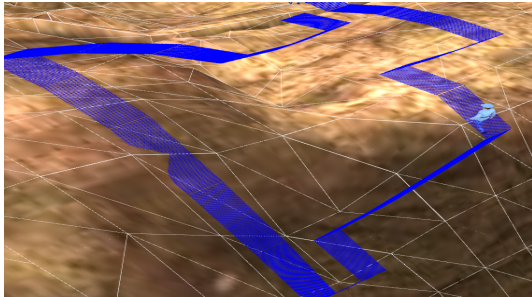


图 6 基于 pgACO 算法的路径规划

### 3.3 基于 A\*算法的路径规划

A\*路径规划是经典的一种路径规划算法，但 A\*算法规划出的路径和最优路径间的误差比较大，且又无法在寻路过程中进行自行修正。

A\*寻路算法：从出发点向着 8 个方向寻找下一步要走的位置，实验中采用的是平面网格上的 A\*寻路算法<sup>[12]</sup>。

利用启发函数：

$$f = d_{si} + h_e \quad (10)$$

寻找到一条最优路径。

A\*算法路径规划：图 7 是该 A\*算法寻到的路径。

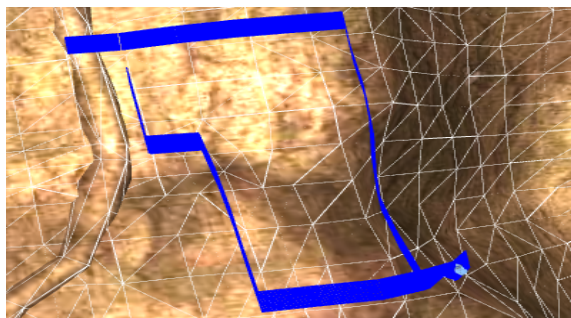


图 7 A\*算法寻到的路径

## 4 实验结果与分析

本文对 mACO、pgACO 和 A\*算法做了对比试验。就规模上分别采用了 10 m×10 m，21 m×21 m 和 40 m×40 m 的场景。规定起点为(1,1)，终点分别为(9,9)、(20,20)和(39, 39)。从四个指标考察算法特点：1) 寻路耗费时间；2) 占用内存资源数量；3) 路径长度；4) 页面刷新频率。具体数据如下 3 个表格所示：

表 1 10×10 规模场景

算法	耗时/ms	消耗内存/kB	路径长度/m	FPS
mACO	101	432	14	51
pgACO	75	304	16	51
A*	12	43	17	58

表 2 21×21 规模场景

算法	耗时/ms	消耗内存/kB	路径长度/m	FPS
mACO	201	532	36	45
pgACO	138	412	41	46
A*	22	92	47	55

表 3 40×40 规模场景

算法	耗时/ms	消耗内存/kB	路径长度/m	FPS
mACO	801	1 532	67	41
pgACO	335	844	73	43
A*	87	176	77	47

通过实验数据可以看到，虽然 mACO 算法耗时要比 mACO 和 A\*算法长，但其所耗用的时间依然小于 1 s，所以 mACO 是可以满足用户路径规划实时性需求的。同时可以看到，mACO 算法要比另两种算法规划出来的路径更加精确。

针对以上四个指标，本文对 mACO、pgACO 和 A\*算法的特点做了进一步对比分析：

在不同规模场景中，三种算法路径规划耗用的时间变化情况如图 8 所示。

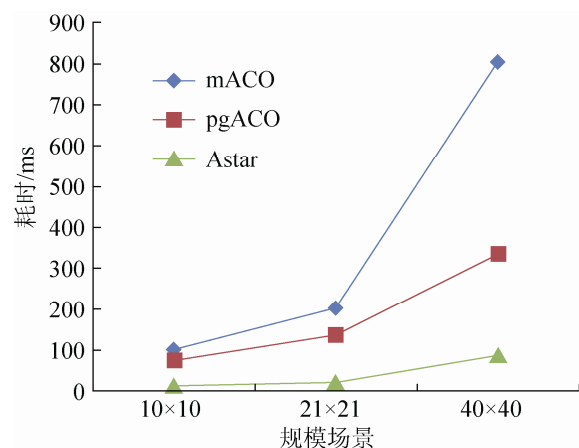


图 8 三种算法耗时比较

在不同规模场景中，三种算法路径规划占用的内存变化情况如图 9 所示。

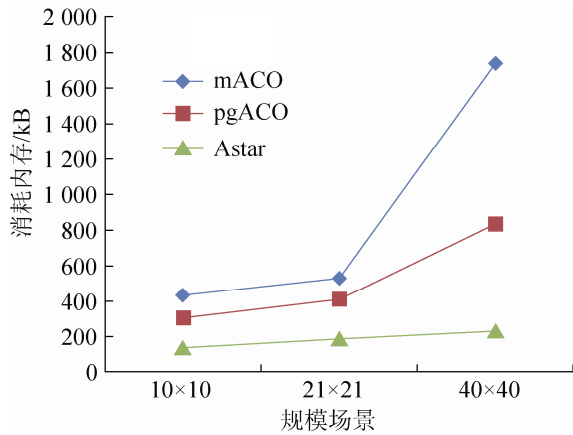


图 9 三种算法占用内存情况

在不同规模场景中, 三种算法规划出来的路径长度情况如图 10 所示。

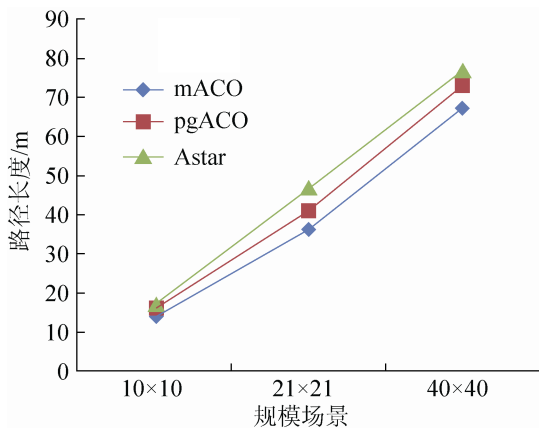


图 10 规划出来的路径长度比较

在不同规模场景中, 三种算法在路径规划过程中帧频情况如图 11 所示。

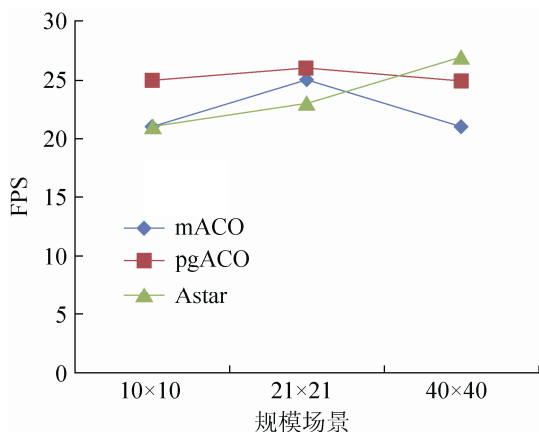


图 11 三种算法不同规模场景下的 FPS

从图 8~11 可以看到, 在一个 40 m×40 m 规模战斗场景中, 可以满足路径规划实时性的需求, 并且帧频一直大于 24 FPS, 屏幕刷新率良好, 对于内存空间的占用量一直低于 2 M, 最优路径预测的精确度, 以 mACO 为最优。

## 5 结论

针对山地场景在 Web 端实时路径规划问题, 使用 Egret.ts 实现了三种路径规划算法, 就算法的原理以及实现结果做了比较。实验证明, 针对井冈山典型山地场景的实时路径规划, 各具特点。A\* 算法计算速度最快, 但是精确度最差, 对于偏重于高精度度路径预测的需要, 显然不提倡使用 A\* 算法来预测蓝军的进军路线。pgACO 算法的精确度虽然较 A\* 算法的精确度高出一些, 但依然不如本文提出的 mACO 算法更加精确。三种算法均达到 Web3D 上的实时路径规划标准, 所占用网页计算资源不高, 刷新频率大于 24FPS, 在这种情况下, 采用 mACO 算法预测山地战斗场景的精确路径规划, 更为合适。

另外, 本文进一步通过编程实现了一个简单的 leader-follower 多人路径规划方案, 从而降低了寻路算法的时间复杂度, 进一步弥补了寻路过程中需要占用过多资源的缺陷。

显然, 进一步的工作还需要深入的展开, 算法效率需要进一步提高。下一步需要重点考虑如何更好的重用蚁群算法中的信息素, 为多源多目标的多智能体进行实时、精确的路径规划。

## 参考文献:

- [1] Deepika P Vinchurkar, Alpa Reshamwala. Robot Path Planning Using An Ant Colony Optimization Approach: A Survey [J]. International Journal of Advanced Research in Artificial Intelligence (S2165-4069), 2013, 2(3): 65-71.
- [2] Colonia Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies [EB/OL]. (2010-09-14) [2016-09-12]. 2010-09. ftp://iridia.ulb.ac.be/pub/mdorigo/conference/IC.0ECAL92.pdf.

- [3] Anna Gorbenu, Vladimir Popov. Multi-agent Path Planning [J]. Applied Mathematical Sciences, (S1312-885X), 2012, 6(135): 6733-6737.
- [4] Dorigo M, Maniezzo V, Coloni A. The Ant System: Optimization by a Colony of Cooperating Agents [EB/OL]. (2012-09-20) [2016-09-12]. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=484436](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=484436).
- [5] Taher Niknam, Babak Amiri. An Efficient Hybrid Approach Based on PSO.ACO and K-means for Cluster Analysis [J]. J. ASC. (S1568-4946), 2010, 10(1): 183-197.
- [6] Crina Groan, Ajith Abraham, Monica Chis. Swarm Intelligence in Data Mining [J]. Studies in Computational Intelligence (SCI) (S1860-949X), 2009, 40(1): 1-20.
- [7] Zhang Weiguo, Lu Tianyu. The Research of Genetic Ant Colony Algorithm and Its Application [C]// The Second SREE Conference on Engineering Modelling and Simulation (CEMS 2012), Hong Kong, China. Procedia Engineering, doi:10.1016/j.proeng.2012.04.210 ,Volume 37, 101-106
- [8] Yoshiaki Kuwata, Jonathan P. How. Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP [J]. IEEE Transactions on Control Systems Technology. (S1063-6536), 2011, 19(2): 423-431.
- [9] Dorigo M, Gambardella L M. Ant colony system: A cooperation learning approach to the travelling salesman problem [J]. IEEE Transactions on Environment and Systems Management (S1089-778X), 1997, 1(1): 53-66.
- [10] Jiang Hui-Yan, Zong Mao, Liu Xiang-Ying. Research of software prediction model based on ACO-SVM [J]. China Journal of Computers. (S0254-4164), 2013, 34(6): 1148-1154.
- [11] Jose Bento, Nate Derbinky, Charles Math, et al. Proximal Operators for Multi-Agent Path Planning [J]. The Advancement of Artificial Intelligence. (S0738-4602), 2015, (7): 35-47.
- [12] Christopher Wilt, Adi Botea. Spatially Distributed Multiagent Path Planning [C]// Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling. Association for the Advancement of Artificial Intelligence, 2014: 332-340.
- [13] Ko-Hsin Cindy Wang, Ad Botea. MAAP: a Scalable Multi-Agent Path Planning Algorithm with Tractability and Completeness Guarantees [J]. Journal of Artificial Intelligence Research. (S0004-3702), 2014, 42(1): 55-90.
- [14] Istvan Nagy. From Exploring to Optimal Path Planning: Considering Error of Navigation in Multi-Agent Mobile Robot Domain [J]. Acta Poytechnica Hungrica (S1785-8860), 2014, 11(6): 39-55.
- [15] Meng Guo, Jana Tumova, Dimos V. Dimarogonas. Cooperative Decentralized Multi-agent Control under Local LTL Tasks and Connectivity Constraints [J]. (S0191-2216), 2015, 27: 33-45.
- [16] Ali Noormohammadi Asi, Mohammad Bagher Mehal, Atena Sajedin. Control of leader-follower formation and path planning of mobile robots using Asexual Reproduction Optimization (ARO) [J]. Applied Soft Computing Archive (S1568-4946), 2014, 14(4): 563-576.



(上接第 2311 页)

- [13] G Gerig, O Kubler, R Kikinis, et al. Nonlinear anisotropic filtering of MRI data [J]. IEEE Transactions on Medical Imaging (S0278-0062), 1992, 11(2): 221-232.
- [14] J Weickert, B Romeny, M A Viergever. Efficient and reliable schemes for nonlinear diffusion filtering [J]. IEEE Transactions on Image Processing (S1057-7149), 1998, 7(3): 398-410.
- [15] W B Tao, F Chang, L M Liu, et al. Interactively multiphase image segmentation based on variational formulation and graph cuts [J]. Pattern Recognition (S0031-3203), 2010, 43 (10): 3208-3218.
- [16] D K Panjwani, G Healey. Unsupervised segmentation of textured colour images using Markov random field models [C]// Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1993. USA: IEEE, 1993: 776-777.
- [17] C Lenglet, M Rousson, R Deriche. DTI segmentation by statistical surface evolution [J]. IEEE Transactions on Medical Imaging (S0278-0062), 2006, 25(6): 685-700.
- [18] R Unnikrishnan, C Pantofaru, M Hebert. Toward objective evaluation of image segmentation algorithms [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence (S0162-8828), 2007, 29(6): 929-944.