

译 堪称最好最全的A*算

2017年12月12日 21:50:38 denghe1

英文原文链接: <http://theory.st>英文原文参考: <http://www-cs->翻译参考链接: <http://blog.csdn>

我们尝试解决的问题是把一个游戏角色（燃料，时间，距离，装备，等）的代价（燃料，时间，距离，装备，等）的一种极端情况是，当游戏对象开始移动时，游戏对象将会沿着该路径移动而忽略其他可能的路径（一条直线取代），取而代之的是在每一步都会得到最好的效果。

1 导言

1.1 算法

1.2 Dijkstra算法与最佳优先搜索

1.3 A*算法

2 启发式算法

2.1 A*对启发式函数的使用

2.2 速度还是精确度？

2.3 衡量单位

2.4 精确的启发式函数

2.4.1 预计算的精确启发式函数

2.4.2 线性精确启发式算法

2.5 网格地图中的启发式算法

2.5.1 曼哈顿距离

2.5.2 对角线距离

2.5.3 欧几里得距离

2.5.4 平方后的欧几里得距离

2.5.5 Breaking ties

2.5.6 区域搜索

3 Implementation notes

3.1 概略

3.2 源代码

3.3 集合的表示

3.3.1 未排序数组或链表

展开

热门文章

堪称最好最全的A*算法详解（译文）

阅读数 52712

自相关函数与互相关函数

阅读数 52516

超平面是什么？——理解超平面 (SVM开篇之超平面详解)

阅读数 44323

理解卷积神经网络CNN中的特征图 feature map

阅读数 26933

目标跟踪算法——KCF 进阶

denghe1122

关注

最新评论

原创 粉丝 喜欢 评论

SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级: 访问: 65万+

qq_42554262: 是你原创的吗? 别作者连接都不加

勋章:

目标跟踪算法——KCF 进阶

qq_40877976: [reply]weixin_42126632[/reply]

face特征

最新文章

目标追踪算法KCF (kernel...

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方...

聚类算法（三）——基于密度的聚类算法

聚类算法是什么？——理解超平面 (SVM...)

qq_33885970: 我仔细想了下，是距离。。。易

蒙美空间的影响，容易把方程想成W+T

的聚类方法（补充阅读）

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其改进算法

分类标签



论文翻译

CSDN资讯

12篇

7篇

Q客服



kefu@csdn.net



400-660-0108

3篇

工作时间 8:30-22:00

机器学习/深度学习

44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月

1篇

疯狂Python精讲



写博客

3.3.2 排序数组

3.3.3 排序链表

3.3.4 排序跳表

3.3.5 索引数组

3.3.6 哈希表

3.3.7 二元堆

展开

3.3.8 伸展树

3.3.9 HOT队列

3.3.10 比较

3.3.11 混合实现

3.4 与游戏循环的交互

3.4.1 提前退出

3.4.2 中断算法

3.4.3 组运动

3.4.4 细化

4 A*算法的变种

4.1 beam search

4.2 迭代深化

4.3 动态衡量

4.4 带宽搜索

4.5 双向搜索

4.6 动态A*与终身计划A*

5 处理运动障碍物

5.1 重新计算路径

5.2 路径拼接

5.3 监视地图变化

5.4 预测障碍物的运动

6 预计算路径的空间代价

6.1 位置VS方向

6.2 路径压缩

6.2.1 位置存储

6.2.2 方向存储

6.3 计算导航点

6.4 极限路径长度

6.5 总结

2018年9月

4篇

2018年8月

3篇

2018年5月

4篇

2018年4月

4篇

2018年3月

9篇

2018年1月

14篇

2017年12月

29篇



63



9



热门文章

堪称最好最全的A*算法详解（译文）

阅读数 52712

自相关函数与互相关函数

阅读数 52516

超平面是什么？——理解超平面（SVM开篇之超平面详解）

阅读数 44323

理解卷积神经网络CNN中的特征图 feature map

阅读数 26933

目标跟踪算法——KCF 进阶

denghe1122 295

关注

最新评论

原创 粉丝 喜欢 评论

SVM++ 实现 652

929

188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级：博客 65万+ 访问qq_42554262: 是你原创的吗？原创 作者连接都不加勋章：恒 目标跟踪算法——KCF 进阶

qq_40877976: [reply]weixin_42126632[/reply]

floop特征

最新文章

目标跟踪算法KCF (kernel...)

sinat_30372583: 从1D到2D通过核函数可以转为一维求解，但是对于DCF，即不使用核函数的† ...

聚类算法（三）——基于密度的聚类算法

聚类算法是什么？——理解超平面（SVM...）

qq_33885970: 我仔细想了下，是距离。。。易

聚类空间的影响，容易把方程想成W₁+W₂=0

的聚类方法（补充阅读）

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其改进算法

分类标签



程序人生

CSDN资讯

7篇

kefu@csdn.net 3篇

400-660-0108

Q客服

VIP客服论坛

工作时间 8:30-22:00

机器学习/深度学习 44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号 1篇

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月

1篇



63

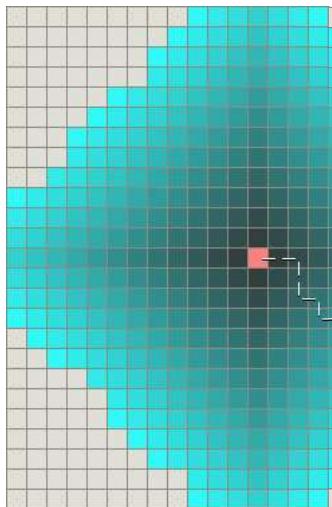


9



1 导言

Dijkstra算法从物体所在的初始点集从初始结点向外扩展，直到到达“径”是因为经常会出现许多差不多的Dijkstra算法扫描过的区域。颜色最



最佳优先搜索（BFS）算法按照选择离目标最近的结点。BFS不能停点。例如，如果目标位于出发点的正表越低的启发式值（移动到目标的

2018年9月	4篇	检查结点集中的结点，并把所有最靠近的尚未检查的结点加入待检查队列。
2018年8月	3篇	从初始点到目标点的最短路径。
2018年5月	4篇	要所有的边都有一个非负的代价值。（从初始结点，蓝色的是目标点，浅蓝色的是有障碍物的有色区域（注：原文是teal areas））
2018年4月	4篇	探测过程（exploration）：
2018年3月	9篇	（frontier）：
2018年1月	14篇	
2017年12月	29篇	

展开

热门文章

堪称最好最全的A*算法详解（译文）

阅读数 52712

自相关函数与互相关函数

阅读数 52516

超平面是什么？——理解超平面（SVM开篇之超平面详解）

阅读数 44323

理解卷积神经网络CNN中的特征图 feature map

阅读数 26933

目标跟踪算法——KCF 进阶

denghe1122

2959

关注

最新评论

原创	粉丝	喜欢	评论
SVM++ 实现	652	929	188

qq_42554262: 请问博主，matrix.h 和 math.h
这两个文件在哪呢

等级：博客 5 访问：65万+

qq_42554262: 是你原创的吗？原作者连接都不加

勋章：恒

目标跟踪算法——KCF 进阶

qq_40877976: [reply]weixin_42126632[/reply]
face特征

最新文章

目标追踪算法KCF (kernel...)

sinat_30372583: 从1D到2D通过核函数可以转为一维求解，但是对于DCF，即不使用核函数的方...

聚类算法（三）——基于密度的聚类算法

聚类算法是什么？——理解超平面（SVM...
以DBSCAN为例）

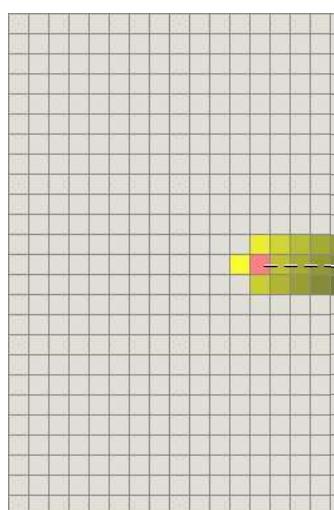
qq_33885970: 我仔细想了下，是距离。。。易

蒙二维空间的影响，容易把方程想成 $W \times T$

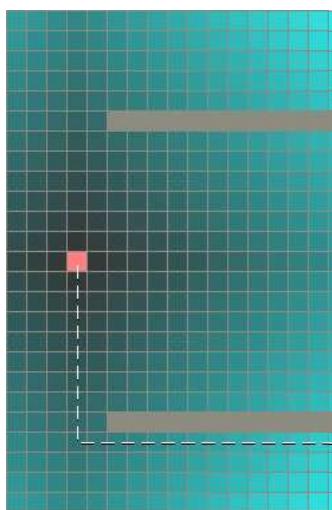
的聚类方法（补充阅读）

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其改进算法



然而，这两个例子都仅仅是最能保证找到一条最短路径：



另一方面，BFS运行得较快，但

尔为启发式的）任意结点到目标点的代价。与选择离初始结点最近的结点的算法快的多，因为它用了一个启发式函数（heuristic function）快速地在图中，越黄的结点代表越高的启发式值（移动到目标的代价高），而BFS运行得更快。

径是直线的。现在我们来考虑前边描述的凹型障碍物。Dijkstra算法运行

分类标签

论文翻译

程序人生

CSDN 资讯

Q客服 kefu@csdn.net 3篇
客服论坛 400-660-0108

工作时间 8:30-22:00
机器学习/深度学习 44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

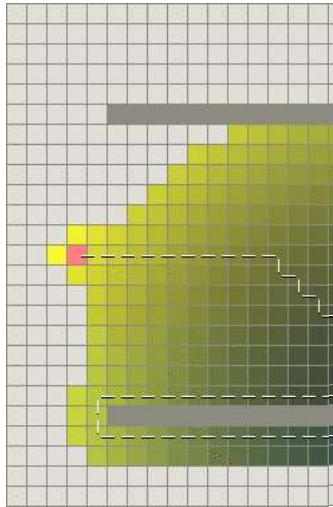
©1999-2019 北京创新乐知网络技术有限公司

展开 网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月 1篇



2018年9月	4篇
2018年8月	3篇
2018年5月	4篇
2018年4月	4篇
2018年3月	9篇
2018年1月	14篇
2017年12月	29篇

展开



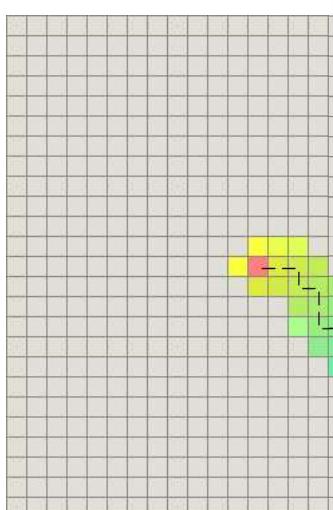
问题在于BFS是基于贪心策略的，很长，它仍然继续走下去。

结合两者优点不是更好吗？1. 同的是，类似BFS的启发式方法经常

1.3 A*算法

我将集中讨论A*算法。A*是路

和其它的图搜索算法一样，A*是通过启发式函数（heuristic）引导它自己。在简单的



在凹型障碍物的例子中，A*找

到。由于它仅仅考虑到达目标的代价，而忽略了当前已花费的代价，于是

uristic approaches) 如BFS，和常规方法如Dijkstra算法结合在一起的尽管A*基于无法保证最佳解的启发式方法，A*却能保证找到一条最短路

，并且能用于多种多样的情形之中。

一样，A*能用于搜索最短路径。和BFS一样，A*能用启发式函数（注：

目标跟踪算法——KCF 进阶
denghe1122
2959

关注

最新评论

原创 粉丝 喜欢 评论
SVM++ 实现 652 929 188

qq_422554262: 请问博主，matrix.h 和 math.h
这两个文件在哪呢

等级： 搬家 5 访问： 65万+

粉丝： 1339390 是你原创的吗？原创作者连接都不加

勋章： 恒

目标跟踪算法——KCF 进阶

qq_40877976: [reply]weixin_42126632[/reply]

fbog特征

最新文章

目标跟踪算法KCF (kerneli...

sinat_30372583: 从1D到2D通过核函数可以转为一维求解，但是对于DCF，即不使用核函数的† ...

聚类算法（三）——基于密度的聚类算法

超平面是什么？——理解超平面 (SVM... (以DBSCAN为例)

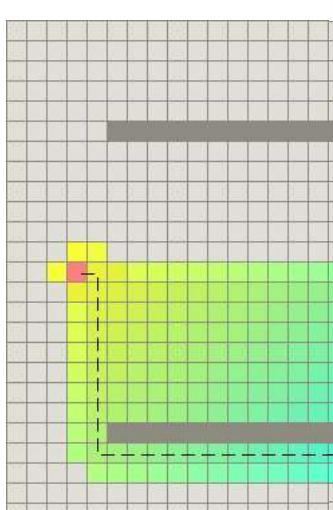
qq_33885970: 我仔细想了下，是距离。。。易

蒙采空间的影响，容易把方程想成 $Wx + b = 0$

的聚类方法（补充阅读）

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其改进算法



分类专栏
12篇
CSDN资讯 7篇

Q客服
kefu@csdn.net
客服论坛

3篇
工作时间 8:30-22:00
400-660-0108

机器学习/深度学习 44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号 1篇

©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月

1篇



成功的秘诀在于，它把Dijkstra到任意结点n的代价， $h(n)$ 表示从结点的结点。当从初始点向目标点移动时。

2 启发式算法

启发式函数 $h(n)$ 告诉A*从任意结点n到目标点的估计代价。

2.1 A*对启发式函数的依赖

启发式函数可以控制A*的行为：

- 一种极端情况，如果 $h(n)$ 是0，A*将只沿着从n到目标的最短路径。
- 如果 $h(n)$ 经常都比从n移动到目标的代价小，A*将找到一条最短路径。
- 如果 $h(n)$ 精确地等于从n移动到目标的代价，A*将在一些特殊情况下让它们精确地找到最短路径。
- 如果 $h(n)$ 有时比从n移动到目标的代价大，A*将不扩展别的任何结点，这将非常快。
- 另一种极端情况，如果 $h(n)$ 比 $g(n)$ 大，A*将永远找不到最短路径。

所以我们得到一个很有趣的情况：如果我们的目标太低，A*将只沿着从n到目标的最短路径。如果我们的目标太高，A*将永远找不到最短路径。

在游戏中，A*的这个特性非常有用。通过平衡 $g(n)$ 和 $h(n)$ ，你可以修改任意一个结点的评估距离。

注：在学术上，如果启发式函数值是固定的，那么A*和A*+是一样的。但在游戏编程领域并不区别A和A*。

2.2 速度还是精确度？

A*改变它自己行为的能力基于以下事实：A*并不真正需要得到最好的路径，仅需得到一个合理的路径。

假设你的游戏有两种地形，平原和山地 mountainous land。这是因为有可能比把3和1比较好。It is not as dissimilar as 3 and 1. Instead, it is more similar to 3 and 1. Now it will search something quicker.

速度和精确度之间的选择前不是静态的，而是动态的。你可以根据地图上的难度或者其它任何因素来进行动态的选择（cost function）用于测量（scale）。

$$g'(n) = 1 + \alpha * (g(n) - 1)$$

如果 α 是0，则改进后的代价函数将起作用，然后你得到了A*的变种。

你也可以考虑对启发式函数的设置进行动态调整。例如，如果你的地图大部分道路，那么你可以考虑让启发式函数忽略这些道路。

速度和精确度之间的选择并不是静态的。如果地图大部分是草地，而只有少数的道路，那么你可以考虑让启发式函数忽略草地，从而更快地找到路径。

2.3 衡量单位

A*计算 $f(n) = g(n) + h(n)$ 。为了使 $h(n)$ 的值既不太大也不太小，因而你将不能得到正确的结果。

2.4 精确的启发式函数

2018年9月

4篇

2018年8月

3篇

2018年5月

4篇

2018年4月

4篇

2018年3月

9篇

2018年1月

14篇

2017年12月

29篇

近目标点的结点) 的信息块。凸包 (convex hull) 表示所有结点的集合。在讨论A*的标准术语中， $g(n)$ 表示从结点n到目标点的估计代价 (estimated cost)。在上图中， $yellow(h)$ 表示远离目标的结点而 teal(g) 表示靠近目标的结点。它检查 $f(n)$ 最小的结点 n ，其 $f(n) = g(n) + h(n)$ 。

一个好的启发式函数是重要的。

展开

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面 \(SVM开篇之超平面详解\)](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122 2959

算法，这保证能找到最短路径。

找到一条最短路径。 $h(n)$ 越小，A*扩展的结点越多，运行就得越慢。

而不扩展别的任何结点，这将非常快。尽管这不可能在所有情况下都实现（注：原文为At exactly the right point），但A*运行得更快。

路径，但它运行得更快。

A*算法。

得什么。理想情况下（注：原文为At exactly the right point），我们想如果我们的目标太高，那我们就放弃了最短路径，但A*运行得更快。

到一条好的路径 ("good" path) 而不是一条完美的路径 ("perfect" pa

算法（原文为simply A）。然而，我继续称之为A*，因为在实现上是一样的。

最新评论

原创 粉丝 喜欢 评论

SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级： 拥抱 5 访问： 65万+

亲热 (关于博文) 的作用——图像预处理

qq_42554262: 是你原创的吗？原创作者连接都不加

勋章： 恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

最新特征

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方

法，我仔细想了下，是距离。。。易

聚类算法 (四)、基于高斯混合分布的聚类方法 (补充阅读)

聚类算法 (二)、聚类算法的系统性比较

聚类算法 (一) —— k-means算法以及其改进算法

分类标签



论文翻译

程序员人生



CSDN资讯

7篇

Q客服

VIP

客服论坛

工作时间

8:30-22:00

机器学习/深度学习

44篇

关于我们

招聘

广告服务

网站地图

百度提供站内搜索

京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务

经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心

家长监护

版权申诉

2019年1月

有用。在速度和精确度之间取得折衷将会让你的游戏运行得更快。在很多游戏中发生着什么，或者运行游戏的机器有多快。

则也是3。A* is going to search three times as far along flat land as it does along mountainous land. If we set the evaluation distance between nodes to 1.5, A* will spend as much time trying to find a way around it. Alternatively, we can tell A* to ignore mountains just tell A* that the movement cost on mountainous terrain is higher than on flat land. Either approach gives up ideal paths to get to the goal faster.

索的时间片数、地图上物体 (units) 的数量、物体的重要性、组 (groups) 等。建立一个启发式函数用于假定通过一个网格空间的最小代价是1，然后建立

完全忽略，A*工作变成简单地判断一个网格是否通过。如果alpha是1，那么任意值。

代价。例如，如果你的地图大部分地形是代价为2的草地，其它一些地方是代价为1的山地。

重要的，你可以基于此进行动态选择。例如，假设我们可能在某点停止重试。为什么对后续路径的精确度感到厌烦？或者，对于在地图上的一个安全区域，我们可以忽略它。 (译者注：译者认为这里指的是，在安全区域，可以考虑不寻找精确路径，即路径的精确度是重要的。VIP会员可以多花点时间用于寻找精确路径。)

不同的衡量单位。如果 $g(n)$ 用米来衡量而 $h(n)$ 用米来衡量，那么A*将会认为

如果你的启发式函数精确地等于在每一结点它都计算 $f(n) = g(n) + h(n)$ path) 上的所有结点的 f 值均大于正肯定不会偏离最短路径。

2.4.1 预计算的精确启发式

构造精确启发函数的一种方法是

- Fit a coarse grid on top of the map.
- Precompute the shortest path between any pair of cells.

(译者：此处不好翻译，暂时保留)

然后添加一个启发函数 h' 用于评估

$$h(n) = h'(n, w_1) + \text{distance}(w_1, w_2)$$

或者如果你希望一个更好但是更昂贵的启发式函数，评估

2.4.2 线性精确启发式算法

在特殊情况下，你可以不通过预计算

如果你正使用简单的启发式函数来衡量单位的问题，或者你所选择的后验信息

2.5 网格地图中的启发式

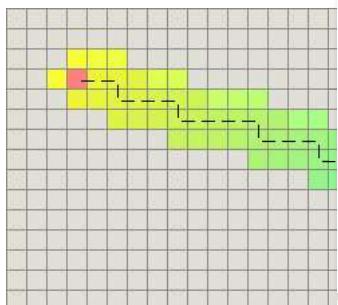
在网格地图中，有一些众所周知的启发式函数：

2.5.1 曼哈顿距离

标准的启发式函数是曼哈顿距离 ($H(n) = D * (abs(n.x - goal.x) + abs(n.y - goal.y))$)。该是曼哈顿距离的D倍：

$$H(n) = D * (abs(n.x - goal.x) + abs(n.y - goal.y))$$

你应该使用符合你的代价函数的衡量



(Note: the above image has a tie)

(译者注：曼哈顿距离——两点在同一条街道上，从一点到达另一点的距离应该是两点在不同街道上的距离之和。当坐标轴变动时，点间的距离就会不同——)

2.5.2 对角线距离

如果在你的地图中你允许对角运动那代替，所以启发函数应该是 $4*D$ 。这个

$$h(n) = D * \max(\text{abs}(n.x - goal.x), \text{abs}(n.y - goal.y))$$

2018年9月

4篇

部分的图中所示，你会看到

*扩展的结点将非常少。A*算法内部发

2018年8月

3篇

原文为match) 时， $f(n)$ 的值

63 该路径时将不会改变。不在正确路径

2018年5月

4篇

应该是指最短路径)。如果

较低 f 值的结点，A*将不考虑 f 值较高的

2018年4月

4篇

度。在许多游戏的地图中这

行。然后，有几种方法可以近似模拟这

2018年3月

9篇

path between any pair of c

grid locations.

2018年1月

14篇

is a generalization of the

grid approach.

2017年12月

29篇

展开

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面 \(SVM开](#)

篇之超平面详解)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature](#)

map

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122

关注

最新评论

原创 粉丝 喜欢 评论

SVM++ 实现

652

929

188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级： 播客 5 访问： 65万+

朱洪斌(文字)的作用——图像预处理

朱洪斌: 是你原创的吗? 创作者连接都不加

勋章： 恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

刚刚特征

最新文章

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方

聚类算法 (二) —— 基于密度的聚类算法

聚类算法 (三) —— 理解超平面 (SV...

以 DBSCAN 为例)

qq_33885970: 我仔细想了下，是距离。。。易

蒙思维空间的影响，容易把方程想成 $W \times T$ 的

聚类方法 (补充阅读)

聚类算法 (二)、聚类算法的系统性比较

聚类算法 (一) —— k-means 算法以及其

改进算法

分类标签



12篇

7篇

Q客服

VIP

客服论坛

kefu@csdn.net

3篇

400-660-0108

工作时间 8:30-22:00

机器学习/深度学习

44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

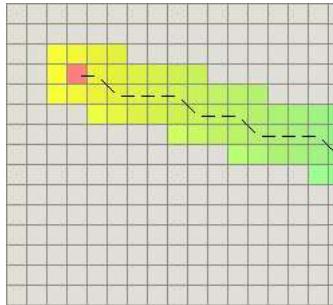
2019年1月

并找到从一个位置移动到邻近位置的最小代价D。因此，我的游戏中的后

，即 $D(I, J) = |X_I - X_J| + |Y_I - Y_J|$ 。对于一个具有正南正北、正东正西方向

向上的旅行，因此曼哈顿距离又称为出租车距离，曼哈顿距离不是距离

4 north) 的曼哈顿距离将 D 。然而，你可以简单地移动 (4 nortl都是 D ：



2018年9月	4篇
2018年8月	3篇
2018年5月	4篇
2018年4月	4篇
2018年3月	9篇
2018年1月	14篇
2017年12月	29篇

展开

如果对角线运动的代价不是D，但类比于曼哈顿距离，对角线运动的代价是D2。所以，如果对角线运动的代价不是D，但类比于曼哈顿距离，对角线运动的代价是D2。

$h_{\text{diagonal}}(n) = \min(\text{abs}(n.x - g.x), \text{abs}(n.y - g.y))$

热门文章

$h_{\text{straight}}(n) = (\text{abs}(n.x - g.x) + \text{abs}(n.y - g.y))$

堪称最好最全的A*算法详解（译文）

阅读数 52712

$h(n) = D2 * h_{\text{diagonal}}(n) + D * (l - h_{\text{diagonal}}(n))$

自相关函数与互相关函数

阅读数 52516

这里，我们计算 $h_{\text{diagonal}}(n)$ ：沿里是曼哈顿距离的步数减去2倍的斜

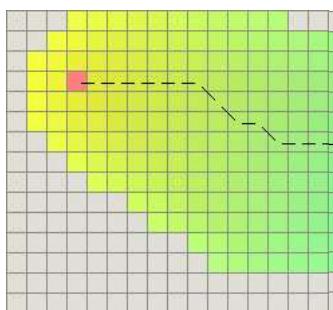


2.5.3 欧几里得距离

如果你的单位可以沿着任意角度移动

$h(n) = D * \sqrt{((n.x - g.x)^2 + (n.y - g.y)^2)}$

然而，如果是这样的话，直接使用A*将运行得更久一些：



最新评论

原创 粉丝 喜欢 评论
SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h
这两个文件在哪呢

等级：博主 访问：65万+

朱泓（文字园）的作用——图像预处理
qq_42554262: 是你原创的吗？原作者连接都不加

勋章：恒

目标跟踪算法——KCF 进阶
qq_40877976: [reply]weixin_42126632[/reply]
fbog特征
最新文章

目标跟踪算法KCF (kernel...)

sinat_30372583: 从1D到2D通过核函数可以转为一维求解，但是对于DCF，即不使用核函数的...

聚类算法（三）——基于密度的聚类算法

超平面是什么？——理解超平面 (SVM...
(以DBSCAN为例))

qq_33885970: 我仔细想了下，是距离。。。易

蒙美算法(四)、基于高斯混合分布

的聚类方法 (补充阅读)

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其改进算法

2.5.4 平方后的欧几里得距离

我曾经看到一些A*的网页，其中提到

$h(n) = D * ((n.x - g.x)^2 + (n.y - g.y)^2)$

不要这样做！这明显地导致衡量单位的距离，这样做会靠近 $g(n)$ 的极端情

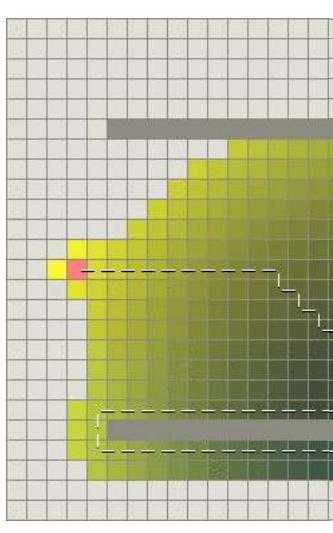
曼哈顿距离；然后合并这两项，让所有的斜线步都乘以D2，剩下的所有直

用直线距离：

itch启发函数 h 。因为欧几里得距离比曼哈顿距离和对角线距离都短，你

距离中昂贵的平方根运算：

离的平方将比 g 的代价大很多，并且你会因为启发式函数评估值过高而停



分类标签
kefu@csdn.net 3篇
400-660-0108

工作时间 8:30-22:00
机器学习/深度学习 44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号 1篇

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月 1篇

2.5.5 Breaking ties Breaking ties

导致低性能的一个原因来自于启发式，我们只需要搜索其中的一条：

2018年9月	4篇
2018年8月	3篇
2018年5月	4篇
2018年4月	4篇
2018年3月	9篇
2018年1月	14篇
2017年12月	29篇

展开

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面（SVM开篇之超平面详解）](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122

关注

最新评论

原创 粉丝 喜欢 评论

SVM++ 实现 652 929 188

qq_42554262: 请问博主, matrix.h 和 math.h

这两个文件在哪呢

等级: 搭客 5 访问: 65万+

粉丝: 15390: 是你原创的吗? 创作者连接都不加

勋章: 恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

hog特征

最新文章

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解, 但是对于DCF, 即不使用核函数的f ...

[聚类算法（三）——基于密度的聚类算法](#)

[超平面是什么？——理解超平面（SVM...（以DBSCAN为例）](#)

qq_33885970: 我仔细想了下, 是距离。。。易

蒙采空间的影响, 容易把方程想成Wx+b=0

的聚类方法 (补充阅读)

[聚类算法（二）、聚类算法的系统性比较](#)

[聚类算法（一）——k-means算法以及其改进算法](#)

为了解决这个问题，我们可以为启发式添加附加值，而且它必须让f值体现区别。因为A*

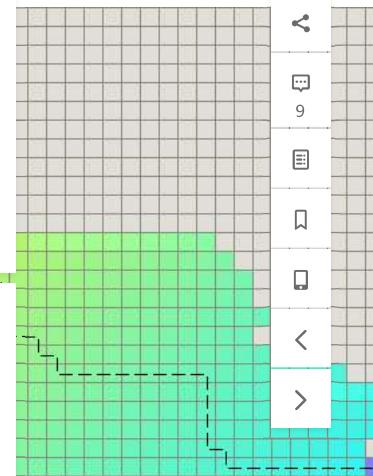
一种添加附加值的方式是稍微改变移动的时候f将逐渐增加。很不幸，downwards scale h upwards slight

heuristic * = (1.0 + p)

选择因子p使得p < 移动一步 (step) 结果是，A*比以前搜索的结点更少

为什么）。当某些路径具有相同的f值的时候，它们都会被搜索 (explored)。

63

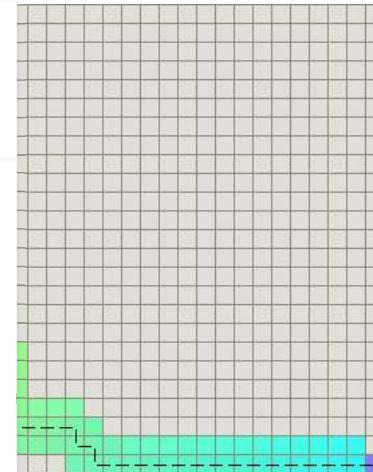


in f values.

small tie breaker)。附加值对于结点必须是确定性的（也就是说，不能是“equivalent”的f值会被检测。

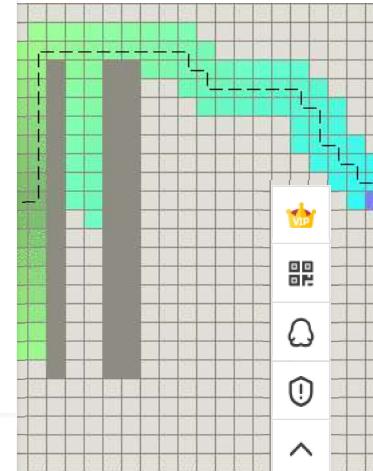
如果我们减少衡量单位（译者注：原文为scale it downwards），那么当我们在一个方向上增加附加值时，而不是靠近目标的结点。我们可以增加衡量单位（译者注：原文为scale it upwards），这样A*会更快地找到靠近目标的结点。

如果你不希望你的路径超过1000步（step），你可以使p = 1 / 1000。添加



scaling added to heuristic.

以后，A*搜索的区域非常少：



heuristic, works nicely with obstacles.

当存在障碍物时，当然仍要在它们周围搜索



分类标签
论文翻译
程序人生

Q客服
答疑论坛

kefu@csdn.net
400-660-0108

工作时间 8:30-22:00
机器学习/深度学习

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

展开
网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月

1篇

Steven van Dijk建议，一个更直截

一个不同的添加附加值的方法是，他

$dx_1 = current.x - goal.x$

$dy_1 = current.y - goal.y$

$dx_2 = start.x - goal.x$

$dy_2 = start.y - goal.y$

$cross = abs(dx_1*dy_2 - dx_2*dy_1)$

$heuristic += cross*0.001$

这段代码计算初始-目标向量 (start vectors don't line up, the cross product区域，而且它找到的路径看起来非常

2018年9月

4篇

2018年8月

3篇

2018年5月

4篇

2018年4月

4篇

2018年3月

9篇

2018年1月

14篇

2017年12月

29篇

展开

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面 \(SVM开篇之超平面详解\)](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122 2955

关注

TA的个人主页 >

最新评论

原创 粉丝 喜欢 评论
SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h
这两个文件在哪呢

等级：博主 访问：65万+

qq_42554262: 是你原创的吗？原创作者连接都不加

勋章：恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

聊聊特征

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方...

[聚类算法（三）——基于密度的聚类算法](#)

[聚类算法是什么？——理解超平面 \(SV... \(以DBSCAN为例\)](#)

qq_33885970: 我仔细想了下，是距离。。。易

蒙采空间的影响，容易把方程想成 $W \times T$ 。

的聚类方法（补充阅读）

[聚类算法（二）、聚类算法的系统性比较](#)

[聚类算法（一）——k-means算法以及其改进算法](#)

分类标签



12篇



CSDN资讯

7篇

C 客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:30-22:00

机器学习/深度学习

44篇

[关于我们 招聘 广告服务 网站地图](#)

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

然而，因为这种附加值倾向于从初

rison function)。当f值相

比函数检查h，然后添加附加值。

为了交互地研究这种附加值方法的改
(http://www.vision.ee.ethz.ch/~mrt/Classic_A_star.html)，你会看到附加

然而另一种添加附加值的方法是，从

你也许也想看看能够更灵活地（译者
得到的路径是否能达到最佳仍在研
实现，所以从它们开始吧，如果你是

凸

63

比较

附加值

。

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

然

如果你想搜索邻近目标的任意一个结点，
h1, h2, h3是邻近结点的启发函数。
并建立一条路径了。

3 Implementation

3.1 概略

如果不考虑具体实现代码，A*从初始结点。CLOSED集保存已考查过的内部 (interior)。每个结点同时

在主循环中重复地从OPEN集中入CLOSED集。然后检查它的邻居n定会被检查的，所以我们现在不考

(*这里我忽略了一个小细节。你确

OPEN = priority queue containir

CLOSED = empty set

while lowest rank in OPEN is not

current = remove lowest rank it

add current to CLOSED

for neighbors of current:

cost = g(current) + movement

if neighbor in OPEN and cost

remove neighbor from OPEN

if neighbor in CLOSED and cost

remove neighbor from CLOS

if neighbor not in OPEN and r

set g(neighbor) to cost

add neighbor to OPEN

set priority queue rank to g(

set neighbor's parent to curr

reconstruct reverse path from go

by following parent pointers

(**) This should never happen if

3.2 源代码

我自己的（旧的）C++A*代码是可
(<http://theory.stanford.edu/~am>)
一样，它在Steve Woodcock'的游

在网上，你能找到C, C++, Visual
(<http://www.codeproject.com/cs>)
(<http://www.geocities.com/jhe>)

3.3 集合的表示

2018年9月

该建立一个启发函数 $h'(x)$, 凸 $h'(x)$ 为 $h_1(x), h_2(x), h_3(x)$ 。... 的最区域的中心。一旦你从OPEN

2018年8月

中取得任意一个邻近目标的结点，你就

2018年5月

的最区域的中心。一旦你从OPEN

2018年4月

中取得任意一个邻近目标的结点，你就

2018年3月

的最区域的中心。一旦你从OPEN

2018年1月

中取得任意一个邻近目标的结点，你就

2017年12月

的最区域的中心。一旦你从OPEN

展开

的最区域的中心。一旦你从OPEN

热门文章

堪称最好最全的A*算法详解（译文）

阅读数 52712

自相关函数与互相关函数

阅读数 52516

超平面是什么？——理解超平面 (SVM开篇之超平面详解)

阅读数 44323

理解卷积神经网络CNN中的特征图 feature map

阅读数 26933

目标跟踪算法——KCF 进阶

denghe1122 2959

关注

TA的个人主页 >

最新评论

原创 粉丝 喜欢 评论
SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h
这两个文件在哪呢

等级： 拥客 65 访问： 65万+

qq_40877976: 是你原创的吗？原作者连接都不加

勋章： 恒

目标跟踪算法——KCF 进阶

qq_40877976: [reply]weixin_42126632[/reply]

hog特征

最新文章

目标跟踪算法KCF (kerneli...

sinat_30372583: 从1D到2D通过核函数可以转为一维求解，但是对于DCF，即不使用核函数的† ...

聚类算法（三）——基于密度的聚类算法

聚类算法是什么？——理解超平面 (SV...

qq_33885970: 我仔细想了下，是距离。。。易

蒙美空间的影响，容易把方程想成 $W \times Y$

的聚类方法（补充阅读）

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其改进算法

分类标签

程序人生

7篇

Q客服

VIP

客服论坛

kefu@csdn.net

3篇

400-660-0108

工作时间 8:30-22:00

机器学习/深度学习

44篇

关于我们 招聘 广告服务 网站地图

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

1篇

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月

1篇

1篇

你首先想到的用于实现OPEN集和C结构，为了选择一种，我们应该考虑在OPEN集上我们主要有三种操作：除最佳是优先队列（<http://members.csdn.net/membertop>）选择哪种数据结构不仅取决于操作，操作对每个被访问的结点都执行一次，必须考虑fringe(F)的最大值。

另外，还有第四种操作，虽然执行中的结点要好（这很少见），那么OPEN集中目前只有四种操作：插入操作（Insertion）、删除操作（Delete）、查找操作（Search）和移动操作（Move）。

3.3.1 未排序数组或链表

最简单的数据结构是未排序数组或链表（Finding the best element）操作中，查找结点花费O(F)，改变值/位置花费O(F)。

3.3.2 排序数组

为了加快删除最佳操作，可以对数组进行排序，这样插入操作花费O(F)以移动所有的元素。查找最佳元素（array），删除最佳元素操作花费将

3.3.3 排序链表

在排序数组中，插入操作很慢。如果插入操作花费O(1)时间，但是查找正确位置需要花费O(F)，改变值/位置花费O(1)。

3.3.4 排序跳表

在未排序链表中查找元素是很慢的。key）的话，集合关系检查操作会很慢，删除一个元素也是O(1)。调整操作：

如果我们用地图位置作为跳表的排序键，查找操作是O(1)。因为集合关系检查更快，所以

3.3.5 索引数组

如果结点的集合有限并且数目是适当的，和它们不同，对所有的n，索引数组是网格的，让索引函数密集起来。

假设i(n)是O(1)的，集合关系检查将最佳操作是O(numnodes)，因为我们

3.3.6 哈希表

索引数组使用了很多内存用于保存哈希表。哈希表的大小等于N的两倍，以便发生O(numnodes)，因为我们需要搜索

3.3.7 二元堆

一个二元堆（不要和内存堆混淆）是两个二元堆的高效实现，我在我自己的

在二元堆中，集体关系检查花费O(F)时间找到节点，并且很神奇，只用O(F)

2018年9月	4篇	-样，你可能想到“数组”。凸	可能想到“链表”。我们可以使用很多
2018年8月	3篇	63	
2018年5月	4篇	司邻居结点时需要检查它是否	合里面；访问邻居结点时需要插入新结
2018年4月	4篇	典型操作。	9
2018年3月	9篇	结点是否在集合中这一操作：	被访问的结点的每个邻居结点都执行一
2018年1月	14篇	不被访问的是搜索空间边缘	9e) 的结点。当评估数据结构上面的这
2017年12月	29篇		

另外，还有第四种操作，虽然执行中的结点要好（这很少见），那么OPEN集中目前只有四种操作：插入操作（Insertion）、删除操作（Delete）、查找操作（Search）和移动操作（Move）。

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面（SVM开篇之超平面详解）](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122

295

关注

TA的个人主页 >

最新评论

原创 粉丝 喜欢 评论
SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级：博客 访问：65万+

qq_39753940: 是你原创的吗？原创作者连接都不加

勋章：恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

最新特征

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方

法，速度慢，而且精度低。

聚类算法（三）——基于密度的聚类算法

（以DBSCAN为例）

qq_33885970: 我仔细想了下，是距离。。。易

蒙采样空间的影响，容易把高程想成WXY中T

的聚类方法（补充阅读）

聚类算法（二）、聚类算法的系统性比较

聚类算法（一）——k-means算法以及其

改进算法

正被检查的结点已经在OPEN集中，（这经常发生），并且如果它的f值比已删除结点（f值不是最佳的结点）大，那么插入操作是O(1)。这两个步骤必须被最优化为一个步骤将移动结点。

（1）很慢，扫描整个结构花费O(F)。插入操作很快，添加到末尾花费O(1)。删除最佳元素（Removing the best element）花费O(F)，而链表则是

g F)，因为我们可以使用折半查找。插入操作会很慢，为了给新元素腾出位置花费是O(1)。如果我们保证最佳排序至数组的尾部（best sorts to the end），改变值/位置花费O(F)。

查操作很慢，需要花费O(F)用于扫描链表。插入操作是很快的，插入新元素间，因为最佳元素已经在表的尾部。删除最佳元素也是O(1)。调整操作中

i(Skip_list) 代替链表的话，可以加速这个操作。在跳表中，如果有排序键，和链表一样，插入操作也是O(1)。如果排序键是f，查找最佳元素很快

完成集合关系检查后，插入操作是O(1)。查找最佳元素是O(F)，删除一个

（1）。查找最佳元素是O(1)，删除一个结点是O(1)。这并不比排序链表好。

数i(n)把结点n映射到一个数组的索引。未排序与排序数组的长度等于OPEN函数是密集的（没有不被使用的索引），max(i(n))将是地图中结点的数

[i(n)]是否包含任何数据。Insertion is O(1), as we just set Array[i(n)].val

用哈希表。哈希表使用了一个哈希函数h(n)把地图上每个结点映射到一个集体关系检查操作花费O(1)；插入操作花费O(1)；删除最佳元素操作花费O(1)。

树通过指针指向子结点所

二元堆使用索引来查找子结点。C++ S

费O(log F)而删除最佳操作花

是O(log F)。调整操作很微妙（tricky）



交互式的（尤其是实时的）游戏对路径好。

一般来说，计算靠近初始结点的路径，哪怕是沿着一条不理想的路径，

可以对物体编程让它们根据自己的行动（这是实际上使用的方法，并非一条路径。让物体在开始时依照本身发生在同一时刻。

2018年9月

2018年8月

2018年5月

2018年4月

2018年3月

2018年1月

2017年12月

4篇

3篇

4篇

4篇

9篇

14篇

29篇

解决方案比得到最佳方案可能更简单。然而在所有其他因素都相同的情况下，。开始原理（The principle of immediate start）：让游戏中的物体尽可能早地开始行动，应该更多地关注A*的延迟（latency）而不是吞吐量（throughput）。好的路径）来行动。除非它们没有足够的信息告诉它们怎么行动，否则它们就根据可用到）。和立即计算所有路径不同，让游戏在每一个，两个，或者三个结点上同时开始（同时），然后才为它们寻找路径。这种方法让让路径搜索的代价趋于平缓，因

展开

3.4.1 提前退出

可以从A*算法的主循环中提前退出，以找到当前最佳结点的路径。这个结点是到目前为止已经找到的最近的结点。

可以提前退出的情况包括检查了一条路径一个比全路径（full path）小白兔的路径。

3.4.2 中断算法

如果需要进行路径搜索的物体较少，

3.4.3 组运动

路径请求并不是均匀分布的。即时请求。

在这种情况下，为某个物体寻找到达部分，对每一个物体，前10步和后10步从P[0]到目的地的路径。

为每个物体寻找的路径是较短的（ ≤ 20 步）。沿着相同的路径移动时，将对游戏产生负面影响。

另一种方法是让每个物体都意识到谁领导寻路。然后用flocking算法让它

然而还有一种方法是利用A*算法的限制。不要消除OPEN和CLOSED集；用A化）

3.4.4 细化

如果地图中没有障碍物，而有不同类型的障碍物，那么A*会考虑通过3个草地以避免1个找到一条路径（这接近于精确启发式）。移动代价去计算更好的路径。

4 A*算法的变种

4.1 beam search beam search

在A*的主循环中，OPEN集保存所有通向一条好的路径的结点将被抛弃。

4.2 迭代深化

迭代深化是一种在许多AI算法中使用的方法。通过查看前面更多步来提高树的深度。ID-A*中，深度是f值的一个cutoff。问的结点都将增加。如果你发现路径太长，<http://www.apl.jhu.edu/~hall/AI/>

我本人认为在游戏地图中没有太大的坐标而已。我认为不保存这些结点以

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面（SVM开篇之超平面详解）](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122 2995

关注

最新评论

原创 粉丝 喜欢 评论

SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h 这两个文件在哪呢

等级： 访问量：65万+ 来源：博客园

博主mp.weixin.qq.com: 是你原创的吗？原创作者连接都不加

勋章：

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

聊聊特征

[目标追踪算法KCF \(kernel...](#)

sinat_30372583: 从1D到2D通过核函数可以转为一维求解，但是对于DCF，即不使用核函数的...

[聚类算法（三）——基于密度的聚类算法（以DBSCAN为例）](#)

qq_33885970: 我仔细想了下，是距离。。。易受二维空间的影响，容易把方程想成WXY中Y的聚类方法（补充阅读）

[聚类算法（二）、聚类算法的系统性比较](#)

[聚类算法（一）——k-means算法以及其改进算法](#)

分类专栏



CSDN资讯 7篇

 客服

kefu@csdn.net 3篇

 客服论坛

400-660-0108 3篇

 工作时间

8:30-22:00 3篇

 机器学习/深度学习

44篇

 关于我们

招聘 广告服务 网站地图 3篇

 百度提供站内搜索

京ICP备19004658号 3篇

 ©1999-2019 北京创新乐知网络技术有限公司

1篇

展开

网络110报警服务 经营性网站备案信息 1篇

北京互联网违法和不良信息举报中心 1篇

中国互联网举报中心 家长监护 版权申诉 1篇

2019年1月

多个物体并命令它们朝着同样的目标移动。这给路径搜索系统以沉重的负担，寻找一条从物体的中心到目的地中心的路径P。对所有物体使用该方法从它的开始点到P[10]的路径，紧接着是共享的路径P[10..len(P)-10]，最

大多数路径只寻找一次并且为所有物体所共享。然而，当玩家们看到物体稍微沿着不同的路径运动。一种方法是选择邻近结点以改变路径。

“领导”物体，或者是通过选择一个能够最好地意识到当前情况的物体）动的多个物体共享，只要物体共享相同的启发式函数和代价函数。当主循环（下一个物体的开始位置）。（这可以被看成是中断算法和提前退出）

来计算一条初始路径。例如，如果草地的代价是1，山地代价是2，山脉的代价看成1.2来计算初始路径，A*将会用更少的时间去设法避免山脉，而可以开始移动，游戏循环就可以继续了。当多余的CPU时间是可用的时候。

的一个变种，这种算法限定了OPEN集的尺寸。如果OPEN集变得过大，这限制了可供选择的数据结构。

能得到更精确的解。该名称： 游戏树搜索，需要查看前面几步（比如，就可以认为你已经得到足够的信息）。当你想要进一步精确化时，它不会如，它不会被加入OPEN集，而是第一次迭代只处理很少的结点。此后每步都可以停止了。更多的细节请参阅这些关于ID-A*的资料：

十算时间而减少内存需求。然在地图路径搜索中，“结点”是很小的一

在动态衡量中，你假设在开始搜索时

$$f(p) = g(p) + w(p) * h(p)$$

启发函数中带有一个权值（weight）

4.4 带宽搜索

带宽搜索（Bandwidth Search）有
限的代价将不会比最佳路径的代价走

另一个特性是，你可以丢弃OPEN集
的结点。这是一个奇怪的特性。对

好奇地（Curiously），你可以对这
个OPEN集中去掉哪些结点。

4.5 双向搜索

与从开始点向目标点搜索不同的是，
路径。

这听起来是个好主意，但我不会给你
小的搜索树。然而我的试验表明，在
运行得如此快的原因——无论你的
会更有用。

面对面的方法（The front-to-front
前向搜索结点—— $g(\text{start},x) + h(x,$

Retargeting方法不允许前向和后向
间结点，从前向最佳中间结点向后向

4.6 动态A*与终身计划

有一些A*的变种允许当初始路径计
能纠正那些错误而不用过多的时间。
路径。然而，D*和LPA*都需要很多
是否需要就地图的改变对路径作调整
术而设计的，这种情况下只有一个

- Overview of D* (<http://www.cs.cmu.edu/~mparr/pubs/>)
- D* Paper 1 (<http://www.cs.cmu.edu/~mparr/pubs/dstar.pdf>)
- D* Paper 2 (http://www.cs.cmu.edu/~mparr/pubs/lpa_star.pdf)
- Lifelong planning overview
- Lifelong planning paper (PDF)
- Lifelong planning A* applet

5 处理运动障碍物

一个路径搜索算法沿着固定障碍物
障碍物也许到达那儿。处理该问题的
分中讨论。这一部分将对路径搜索方

5.1 重新计算路径

当时间渐渐过去，我们希望游戏世界
决定什么时候需要重新计算路径：

- 每N步：这保证用于计算路径的
任何可以使用额外的CPU时间的
当物体拐弯或者跨越一个导航点
当物体附近的世界改变了的时候

2018年9月

4篇

搜索接近结束时，最重要的：凸到目标点。

2018年8月

3篇

63

2018年5月

4篇

2018年4月

4篇

2018年3月

9篇

2018年1月

14篇

2017年12月

29篇

个权值；这降低了启发函数的准确性，同时增加了路径真实代价的相对重量。假设h是过高估计的值，但不等于e。如果这就是你遇到的情况，那么，最终效果就越好。

展开

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面（SVM开篇之超平面详解）](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122 2959

关注

最新评论

原创 粉丝 喜欢 评论
143++ 实验 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级：博客 5 访问：65万+

qq_40877976: 是你原创的吗？原创者连接都不加

勋章：恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

最新特征

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方

聚类算法（三）——基于密度的聚类算法

（以DBSCAN为例）

qq_33885970: 我仔细想了下，是距离。。。易

蒙美空间的影响，容易把方程想成 $W \times Y$ 中 Y

的聚类方法（补充阅读）

[聚类算法（二）、聚类算法的系统性比较](#)

[聚类算法（一）——k-means算法以及其改进算法](#)

分类标签



访问：65万+

粉丝 652

喜欢 929

评论 188

qq_42554262: 请问博主，matrix.h 和 math.h

这两个文件在哪呢

等级：博客 5 访问：65万+

qq_40877976: 是你原创的吗？原创者连接都不加

勋章：恒

[目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply]

最新特征

[目标跟踪算法KCF \(kernel...\)](#)

sinat_30372583: 从1D到2D通过核函数可以转为

一维求解，但是对于DCF，即不使用核函数的方

聚类算法（三）——基于密度的聚类算法

（以DBSCAN为例）

qq_33885970: 我仔细想了下，是距离。。。易

蒙美空间的影响，容易把方程想成 $W \times Y$ 中 Y

的聚类方法（补充阅读）

[聚类算法（二）、聚类算法的系统性比较](#)

[聚类算法（一）——k-means算法以及其改进算法](#)

2019年1月

1篇

搜索接近结束时，最重要的：凸到目标点。

63

个权值；这降低了启发函数的准确性，同时增加了路径真实代价的相对重量。假设h是过高估计的值，但不等于e。如果这就是你遇到的情况，那么，最终效果就越好。

1

的时候（对于某些d），你会发现那些f值比OPEN集中的最好结点的f值还要在这个范围之外的结点将被丢弃掉，因为这个结点肯定不会是最优的。使用一个启发函数可以得到解决。使用一个启发函数可以保证你得到的路径不会太差，另一个

>

开始点向目标点，另一个从目标点向开始点。当它们相遇时，你将得到一个环。

生成了一棵在地图上散开的树。一棵大树比两棵小树差得多，所以最好在当前位置附近的区域，但是又不像Dijkstra算法那样散开。事实上，这路径是疯狂的。它只尝试搜索地图上小范围的区域。如果你的地图很复杂

种算法选择一对具有最好的 $g(\text{start},x) + h(x,y) + g(y,\text{goal})$ 的结点，而不是 $g(\text{goal},x) + h(x,y)$ 。

运行前向搜索一段时间，然后再朝这个结点运行后向搜索。然后选择一直到两个中间结点碰到一块。

有全局所有信息的时候。如果你没有所有的信息，A*可能会出错；D*的缺点是地图发生改变时，路径将变得无效；LPA*可以重新使用之前A*的计算结果（OPEN和CLOSED集，路径树，g值），当地图发生改变时，D*或者LPA*通常不希望保存所有这些信息，所以D*和LPA*在这里并不适用。它们使用内存。如果你的游戏只有一个或者少数几个物体，你可以研究一下D*和LPA*。

<https://iki/pub/Csci3903s03/KellysPaper/seminar.pdf>

样？当一个物体到达一个特写的位置，原来的障碍物也许不再在那儿了，（movement algorithms）替代，这就不能look far ahead；这种方法：

也许是最佳的了。对旧

用新的信息进行更新是有价值的。以下

1

体数量多时，或者运行游戏

1

比较慢时，每个物体对CPU的使用可得

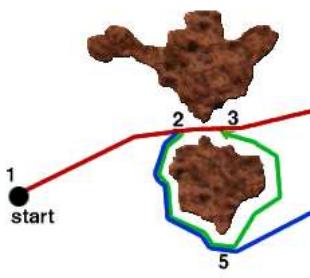
重计算路径的主要缺点是许多路径相交。如果M=550，对M步长的路径，大约需要重新计算550²=302500次。

5.2 路径拼接

当一条路径需要被重新计算时，意味着必须在附近寻找好的路径，同时假设远处的障碍物没有改变。

1. 令 $p[1..p[N]]$ 为路径 (N步) 的前部分
2. 为 $p[1]$ 到 $p[M]$ 计算一条新的路径
3. 把这条新路径拼接 (Splice)

展开



因为 $p[1]$ 和 $p[M]$ 比分开的 M 步小 (J)，所以。上面的图显示了这种情况。最近代，结果是物体沿着路径 1-2-5-3-4 移动。

通常可以通过查看新路径的长度检测这种短的路径，算法返回错误代码；这

对于其它情况，对于 N 步的路径，路径拼接的代价和拼接的步数 M 无关，拼接的路径可能太短以致不能很好地适应你的地图最合适。

路径拼接确实比重计算路径要快，但不是所有的时机，所以可以对该方法进行调整。

Note: Bryan Stout 有两个算法，I 2007 (<https://www.cmpevents.com/>)

Implementation Note:

反向保存路径，从而删除路径的开始部分的元素总是下一个要使用的。

5.3 监视地图变化

与间隔一段时间重计算全部或部分地图的所有区域，也可以只是包含部分区域。以路径将被重新计算以适应障碍物的移动。

这种技术有许多变种。例如，可以在物体检查区域，而不是让区域通知物体。

监视地图变化允许当障碍物不改变时。

5.4 预测障碍物的运动

如果障碍物的运动可以预测，就能知道到达一点的时间需求 (通过计算是否可以通过)。这个改进不是完美的，但可以接受的。

2018年9月	4篇	每10步重新计算一次，路径将被重新计算550 ² =302500次。
2018年8月	3篇	凸的路径，重计算不是个好主意。使用路径信息比丢弃它更好。
2018年5月	4篇	
2018年4月	4篇	
2018年3月	9篇	对地图中当前邻近的区域，比对远处的区域了解得更多。因此，我与重新计算整个路径不同。
2018年1月	14篇	可以重新计算路径的前 M 步：
2017年12月	29篇	

热门文章

[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么？——理解超平面 \(SVM开篇之超平面详解\)](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

denghe1122 2995

关注

最新评论

原创 粉丝 喜欢 评论
SVM++ 实现 652 929 188

qq_42554262: 请问博主，matrix.h 和 math.h 这两个文件在哪呢

等级：博客 5 访问量 65万+ 来自于mp.weixin.qq.com: 是你原创的吗？原创作者连接都不加

勋章：恒 [目标跟踪算法——KCF 进阶](#)

qq_40877976: [reply]weixin_42126632[/reply] thank特征

[目标跟踪算法KCF \(kernel...](#)

sinat_30372583: 从1D到2D通过核函数可以转化为一维求解，但是对于DCF，即不使用核函数的...

[聚类算法（三）——基于密度的聚类算法（以DBSCAN为例）](#)

qq_33885970: 我仔细想了下，是距离。。。易受多维空间的影响，容易把方程想成WXY中Y的聚类方法（补充阅读）

[聚类算法（二）、聚类算法的系统性比较](#)

[聚类算法（一）——k-means算法以及其改进算法](#)

分类标签



热门文章



Q客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:30-22:00

[机器学习/深度学习](#)

44篇

[关于我们 招聘 广告服务 网站地图](#)

百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

展开

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

2019年1月

每10步重新计算一次，路径将被重新计算550 ² =302500次。	凸的路径，重计算不是个好主意。使用路径信息比丢弃它更好。
	对地图中当前邻近的区域，比对远处的区域了解得更多。因此，我与重新计算整个路径不同。
	可以重新计算路径的前 M 步：

n M steps apart)，看起来新路径不会很长。不幸的是，新的路径也许如果我们到达2并且发现从2到达3的路径被封锁了，路径拼接技术会把蓝色的路径1-2-5-4是一条更好的路径。

是不好的。一个简单的解决方法是，为搜索算法设置一个最大路径长度。从而得到路径1-2-5-4。

新路径的频率。对于对世界的改变作反应的能力而言，这个代价是相当大的。质量的折衷。如果M太大，物体的移动将不能快速对地图的改变作出反应。1-2-5-3-4 将被找到。尝试不同的M值和不同的拼接标准(如每3/4 M步)。

常可以发现这种情况并用路径重计算来取代。也可以调整一些变量，如M

到灵感，并在实践中运行得很好。他出席了GDC

（Session ID=4608）；一旦他把资料放在网上，我将链接过去。

，因为这两个操作都将在数组的末尾进行。本质上你可以把这个数组看成

重计算。地图可以分成区域，每个物体都可以对某些区域感兴趣（可以是离开一个区域，该区域将被标识为已改变，所有对该区域感兴趣的物体都

。多个改变可以成组地触发一个通知，因此避免了额外的重计算。另一

或并不经常改变时，考虑这些。

如A*的算法有一个代价函数，代价函数可以考虑穿越地图上一点的代价有多难。A*并用预测的障碍物位置检查在某个时刻障碍物自动离开的可能性，同时区分到达相同目的地的不同的路径，

6 预计算路径的空间

有时，路径计算的限制因素不是时间
(OPEN和CLOSED集) 通常比保存结
的数据结构的不同，最小化临时空间

6.1 位置VS方向

一条路径可以用位置或者方向来表示，容易被查询；只有沿着整个路径移动的空间就够了。如果物体只能沿着四条边走，那么空间节省效果明显。Hannu Karttunen 举了一个例子：对于一个立方体来说意义不大。比如，如果你的物体只能沿着四条边走，那么在立方体上走有三个方向（直走，左旋60度，右旋60度）。

6.2 路径压缩

一旦找到一条路径，可以对它进行反向的。在做决定之前，考察你的决策并且在同一时刻只有50个在移动，即

6.2.1 位置存储

在障碍物比地形对路径搜索影响更大时，
运动过程将包含检查下一结点和沿着前

6.2.2 方向存储

保存方向时，有一种情况是同一个方向。一种方法是保存方向以及朝着该方向位置压缩的直线来说，方向压缩是可行的。Hannu Kankaanpaa指出，在一个方向上。

另一种保存路径的方法是变长编码。向。在一个四方向地图中，你只能在

变长编码比run length encoding更走2步) 用run length encoding表而对于变长编码, 你用1比特表示每1步, 则每次拐弯都节省了一个比特encoding表示是[(NORTH, 200)].

6.3 计算导航点

一个导航点 (waypoint) 是路径上 collapse (译者：不好翻译，保留原动词) 将在两个导航点之间运行

6.4 极限路径长度

当地图中的条件或者秩序会发生改变
步，然后当路径已经沿用时重新计算

6.5 总结

在游戏中，路径潜在地花费了许多存储空间，减少了空间需求。Waypoints rely on being well-known paths calculated in advance to save storage space, which can limit path length.

日期	文章数	操作	摘要
2018年9月	4篇	↑	
2018年8月	3篇	63	路径搜索器需要空间以运行：保存路径。算法运行所需的临时空间（
2018年5月	4篇	↖	时间计算一条路径，可以把空间数量最小化。另外，为OPEN和CLOSE
2018年4月	4篇	↗	优化用于计算路径的空间代价函数。
2018年3月	9篇	...	
2018年1月	14篇	...	
2017年12月	29篇	国	
展开		↑	易于查询路径中的任意位置而不用沿着路径移动。当保存方向时，每走一步是32位。而方向是很少的，
		↓	，位置可以被保存为两个16位，每走一步只需要三位。这些对于保存路
热门文章		□	物体能沿着6个或者8个方向而不是绝对方向（朝北走）。有些相对方向性很小。在只有六种方向的情况下是有效的（比如，沿着陡坡走之字形的路径时）。
		□	就是保存相对方向（右旋60度，你只有五个有意义的方向。在某些地

不进行讨论。使用特定的压缩算法可以缩小路径的存储，无论它是基于位外还要考虑实现和调试，代码量，and whether it really matters.如果仅仅是只有不到50k，总之，没有必要担心压缩的效果。

果是这种情况，那么路径只需要包含直线部分的终止点（有时叫waypoint）

空间

一个方向并不是移动很多次时，这种优化的效果反而不好。同样的，对于动的方向关联。通过相对方向，你可以把“继续前进”当作可能的方向挂左和向右135度（假设你的地图允许这个），然后你可以仅用两个比特保

最一般的步骤：向前走。使用一个“1”表示拐弯，后边再跟几个比特表示“11”表示右转。

的直线路段则不然。序列(向北直走6步,左转,直走3步,右转,直走5步(WEST,2])。如果每个方向用2比特,每个距离用8比特,保存这条路径:0001000011000001000]——一共24比特。如果初始方向和每一步用变长编码保存更长的路径时需要更多的空间。序列(向北直走200步)序列则是[NORTH 00...],一共需要202比特。

在进行路径搜索之后，一个后处理 (post-processing) 的步骤可能会把矢量路径上那些方向发生改变的地方，或者在一个重要的 (major) 位置如：



在从某些点开始，后边的路径已经没有用了。每个物体都可以保存路径并
且每个物体使用数据的总量的管理。

本需要寻路时。路径压缩，
on so that we have to stop
ed places on the map. (译
[省空间，信息可以被丢弃，
和beacons通过把多个步骤保存为一个
图例 y the endpoints, while beacons rely
[处不好翻译，暂时保留原文) 如果路
才重新计算它。

<p>【算法入门】A*寻路算法具体过程 http://www.360doc.com/content/16/</p> <p>想对作者说点什么</p> <p>0000000000(不用数了十个0): 大</p> <p>baidu_39591849: 谢谢大佬,写的</p> <p>不要放辣: 请问BFS到底属不属于</p> <p>A*算法 (C++实现) 简易地图如图所示简易地图,其中绿色方块</p> <p>航迹规划算法——总结 确定性方法 传统的确定性方法包括,</p> <p>A星详解 A*寻路算法原文地址: http://www.gar</p> <p>A*算法图解 记得好象刚知道游戏开发这一行的时候老</p> <p>A*算法 A*算法注: 讲解部分来自博客园Nickname</p> <p>[转]Amit's Astar Page中译文 如此好贴,不能不转! 原文地址: http</p> <p>A*算法的具体思想 (我见过的写的 A*算法的具体思想 (我见过的写的最好的</p> <p>A星算法详解(个人认为最详细,最通 A*寻路算法原文地址: http://www.gar</p> <p>A*算法详解 一: 简化搜索区域将地图转化为栅格型网</p> <p>堪称最好最全的A*算法详解(译文) 堪称最好最全的A*算法详解(译文)</p> <p>堪称最好的A*算法 - CSDN博客</p> <p>A*算法计算步骤总结 A*算法计算步骤总结1, 从点A开始, 并</p> <p>xgf415 70篇文章 排名:千里之外 关注</p> <p>A*算法详解 - m0_38054145的博客</p> <p>A*算法计算步骤总结 - u010894131的博客</p> <p>A*算法介绍 不得不叹服, 强大算法背后, 都是简单得</p>	<p>2018年9月 4篇 2018年8月 3篇 2018年5月 4篇 2018年4月 4篇 2018年3月 9篇 2018年1月 14篇 2017年12月 29篇 (#7楼) 展开 楼) 查看回复(1)</p> <p>热门文章 堪称最好最全的A*算法详解 (译文) 阅读数 52712 自相关函数与互相关函数 阅读数 52516 超平面是什么? ——理解超平面 (SVM开篇之超平面详解) 阅读数 44323 理解卷积神经网络CNN中的特征图 feature map 阅读数 26933 目标跟踪算法——KCF 进阶 denghe1122 粉丝 295 原创 143 + + 实现 652 喜欢 929 评论 188 qq_42554262: 请问博主, matrix.h 和 math.h 这两个文件在哪呢 等级: 搬客 5 访问: 65万+ qq_42554260: 是你原创的吗? 原作者连接都不加 勋章: 目标跟踪算法——KCF 进阶 qq_40877976: [reply]weixin_42126632[/reply] fhoog特征 最新特征 目标跟踪算法KCF (kernel... sinat_30372583: 从1D到2D通过核函数可以转为 一维求解, 但是对于DCF, 即不使用核函数的 聚类算法 (三) —— 基于密度的聚类算法 (以DBSCAN为例) qq_33885970: 我仔细想了下, 是距离。。。易 蒙骗空间的影响, 容易把方程想成Wx+b=0 的聚类方法 (补充阅读) 聚类算法 (二)、聚类算法的系统性比较 聚类算法 (一) —— k-means算法以及其 改进算法</p> <p>分类专栏 论文翻译 程序人生 CSDN资讯 Q客服 CPR 客服论坛 工作时间 8:30-22:00 机器学习/深度学习 关于我们 招聘 广告服务 网站地图 百度提供站内搜索 京ICP备19004658号 ©1999-2019 北京创新乐知网络技术有限公司 展开 网络110报警服务 经营性网站备案信息 北京互联网违法和不良信息举报中心 中国互联网举报中心 家长监护 版权申诉 2019年1月 1篇</p>	<p>63 来自: summer 阅读数 6483 分享 收藏 打印 举报</p> <p>9 查看9条热评</p> <p>阅读数 5697 方块(用B表示)是目的地,为... 来自: A_L_A_N 查看全文</p> <p>阅读数 5424 可视图方法、人工势场法、... 博文 来自: 漫长IT路 查看全文</p> <p>阅读数 430 叙述虽然掌握了 A* 算法的... 博文 来自: yuxikuo_1的专栏 查看全文</p> <p>阅读数 4571 有去看这方面的资料,前几... 博文 来自: 峻峰飞阳的博客 bl... 查看全文</p> <p>阅读数 432 js.com/21207-iHome原文... 博文 来自: Fighting~~~ 的博客 查看全文</p> <p>阅读数 6912 netic/AmitAStar.mht本文... 博文 来自: 赖勇浩的编程私伙局 查看全文</p> <p>12-28 的一份) 下载</p> <p>阅读数 8万+ 叙述虽然掌握了 A* 算法的... 博文 来自: Colin、 查看全文</p> <p>阅读数 1963 来寻找最短路径的方块 (... 博文 来自: m0_38054145的博... 查看全文</p> <p>阅读数 1389 列表就像一张购物清单。 ... 博文 来自: u010894131的博客 查看全文</p> <p>不懂音乐的欣赏者 85篇文章 排名:千里之外 关注</p> <p>little_two_two 933篇文章 排名:776 关注</p> <p>VIP QQ 云 盾 ^</p> <p>阅读数 311 要让用A*算法。什么都没... 来自: Sudouble的专栏 查看全文</p>
---	--	--

堪称最好的A*算法	2018年9月	4篇		阅读数 27
原文地址: http://theory.stanford.edu/	2018年8月	3篇	/www-cs-students.stanfo...	来自: weixin_34128501...
A*算法详解 - O.K的博客 - CSDN	2018年5月	4篇		
	2018年4月	4篇		9
A*算法详解 - Rocky的博客 - CSDN	2018年3月	9篇		
	2018年1月	14篇		
求教, 关于A*算法的优化, 达人们!	2017年12月	29篇		论坛
以下代码只能在MFC里面运行, 因为用到	展开			但是在地图达到400*600的时候, 会... (展开)
Pareto 最优集和最优前沿	热门文章			阅读数 4670
	堪称最好最全的A*算法详解（译文）		来自: youhuakongzhi...	
A*算法详解 - ABC_ORANGE的博客	自相关函数与互相关函数			阅读数 52712
	阅读数 52516		来自: stevensonson的博...	
推荐一个学习深度学习的好网站 - c...	超平面是什么? ——理解超平面 (SVM开篇之超平面详解)			阅读数 220
	阅读数 44323		深搜和广搜。深搜的好处... 博文 来自: stevensonson的博...	
A*算法浅谈	理解卷积神经网络CNN中的特征图 feature map			阅读数 26933
A*是一种搜索, 对, 一种启(xuan)发(xu...	阅读数 295		来自: denghe1122 TA的个人主页 >	
A*寻路算法讲解+源码DEMO演示	目标跟踪算法——KCF 进阶			阅读数 4597
本文源码下载地址: <a for="" href="http://download.csdn.net/...</td><td>denghe1122</td><td>关注</td><td>回头温习云风的书, 看到A*... 博文 来自: 超然于物外 烟火于...</td><td></td></tr> <tr> <td>A*算法题解研究 - u013318356的博客</td><td>最新评论</td><td></td><td>	阅读数 6108			
	原创 粉丝 喜欢 评论		的人认为它容易, 但是对... 博文 来自: NL的博客	
	SVM++ 实现 652 929 188		qq_42554262: 请问博主, matrix.h 和 math.h	
	qq_42554262: 这两个文件在哪呢		这两个文件在哪呢	
A*算法详解 (讲的一级棒)	禁锢窗(文字窗)的作用——图像预处理			阅读数 49
转自: https://blog.csdn.net/hitwhylz/	qq_42554260: 是你原创的吗? 原作者连接都不加		在前面的博客当中, 其实我... 博文 来自: java开发指南博客 ...	
一步一步写算法 (之 A*算法)	目标跟踪算法——KCF 进阶			阅读数 8333
【声明: 版权所有, 欢迎转载, 请勿用于	qq_40877976: [reply]weixin_42126632[/reply]		很多人认为它容易, 但是对... 博文 来自: NL的博客	
	商业用途】		qq_40877976: [reply]weixin_42126632[/reply]	
A*算法是什么?	最新文章			阅读数 738
常看到大家说A*算法, 它是什么东东, 有	目标跟踪算法KCF (kernel...)		有很多可以改善的地方, 趁... 博文 来自: qq_33200959的博客	
A*算法-路径规划	sinat_30372583: 从1D到2D通过核函数可以转为		教育A*寻路A*算法基本原理... 博文 来自: 英赛德游戏研究中心	
照着A*算法的原理自己用代码实现了下,	一维求解, 但是对于DCF, 即不使用核函数的† ...		论坛	
浅谈游戏中的A*寻路算法	聚类算法 (三) —— 基于密度的聚类算法 (以DBSCAN为例)			阅读数 13万+
作者: 锐亚教育 (www.insideria.cn) 本	qq_33885970: 我仔细想了下, 是距离。。。易		理论上GMM可以拟合出任... 博文 来自: 小平子的专栏	
A*算法, 在最短路径的基础上, 我们	聚类算法 (四)、基于高斯混合分布的聚类方法 (补充阅读)			阅读数 392
using System; using System.Collections.Generics; using System.Linq; using System.Text; namespace ASt	聚类算法 (一) —— k-means算法以及其改进算法		方记录下来, 就这么突然地... 博文 来自: 我才睡醒的博客	
第四章 Dijkstra和A*寻路算法	分类标签			阅读数 3835
寻路寻路希望ai中的角色能够计算一条从		12篇	能够尽可能的合理和短。在... 博文 来自: oneDay的专栏	
高斯混合模型 (GMM) 及其EM算法		7篇		阅读数 13万+
一个例子高斯混合模型 (GaussianMixe	C Q客服 工作时间 8:30-22:00	3篇	理论上GMM可以拟合出任... 博文 来自: 小平子的专栏	
A星算法详解——我的第一篇博客	论文翻译 程序人生	44篇		阅读数 5671
A*寻路算法这是我在CSDN上的第一篇博	机器学习/深度学习	400-660-0108	关于工作时间 8:30-22:00 博文 来自: HERO_CJN的博客	
A*算法原理	百度提供站内搜索 京ICP备19004658号	1篇	关于机器学习/深度学习 博文 来自: HERO_CJN的博客	
A*简介A算法是启发式算法重要的一种,	©1999-2019 北京创新乐知网络技术有限公司		百度提供站内搜索 京ICP备19004658号 博文 来自: HERO_CJN的博客	
A*, 那个传说中的算法	网络110报警服务 经营性网站备案信息		©1999-2019 北京创新乐知网络技术有限公司 博文 来自: HERO_CJN的博客	
老子带你揭开传说中的面纱	北京互联网违法和不良信息举报中心		网络110报警服务 经营性网站备案信息 博文 来自: HERO_CJN的博客	
	中国互联网举报中心 家长监护 版权申诉		北京互联网违法和不良信息举报中心 博文 来自: HERO_CJN的博客	
	2019年1月	1篇	中国互联网举报中心 博文 来自: HERO_CJN的博客	

手把手教你用matlab做无人驾驶（二）	2018年9月	4篇		阅读数 5237
整个程序下载地址如下: https://downloa...	2018年8月	3篇		来自: caokaifa的博客
A*算法总结	2018年5月	4篇		阅读数 187
源代码传送门文章目录A*算法导言利用A*	2018年4月	4篇		来自: illiyt的博客
a*自动寻路算法详解	2018年3月	9篇		阅读数 1万+
这篇博文是在其他博客基础上加工的，主	2018年1月	14篇		来自: jialeheyeshu的博客
A*算法解决八数码问题	2017年12月	29篇		阅读数 3万+
1 问题描述1.1什么是八数码问题八数码	展开			来自: 我的新博客
热门文章				阅读数 1599
编辑距离及编辑距离算法	堪称最好最全的A*算法详解（译文）	棋子，留下一个空位。与空...		来自: 田野中的锋芒
编辑距离概念描述：编辑距离，又称Lev	阅读数 52712	弓一个所需的最少编辑操作...		
推荐一个学习深度学习的好网站	自相关函数与互相关函数	博文		阅读数 5474
给大家推荐一个学习深度学习的好网站：	阅读数 52516	受用，值得一看。还有一个...		来自: dengheCSDN的博客
异常点/离群点检测算法——LOF	超平面是什么？——理解超平面 (SVM开	博文		阅读数 9万+
局部异常因子算法-LocalOutlierFactor()	篇之超平面详解)	星和模型训练之前对数据进...		来自: wangyibo0201的...
导航路径规划之五 A*算法	阅读数 44323	量利用目前已知的诸如迭代...		阅读数 1万+
A*算法是启发式搜索，是一种尽可能基	理解卷积神经网络CNN中的特征图 feature	博文		来自: autonavi2012的博客
嵌入式驱动那年的笔试面试-有干货	map	博文		阅读数 1万+
那年的笔试面试题，面试经验总结和干货	阅读数 26933	博文		来自: Mingrenjiwei的...
一、A*搜索算法	 denghe1122	TA的个人主页 >		阅读数 25
经典算法研究系列：一、A*搜索算法	原创 粉丝 喜欢 评论	qq_42554262: 请问博主, matrix.h 和 math.h		来自: weixin_33862188...
Python实现A*算法解决N数码问题	SVM++ 实现 652 929 188	qq_42554262: 这两个文件在哪呢		
Python实现了A*算法，解决N数码的问题	qq_42554262: [reply]weixin_42126632[/reply]	更多请参阅: 十三个经典算...		阅读数 2978
A*算法在最短路问题的应用及其使	qq_40877976: [reply]weixin_42126632[/reply]	博文		来自: Erick Lv的笔记
A*算法在最短路问题的应用及其使用举	hog特征	博文		阅读数 90
史上最好的LDA(线性判别分析)教程	 sinat_30372583	从1D到2D通过核函数可以转为		来自: StarHai的博客
一、前言最近由于研究需要，要用到线性	一维求解，但是对于DCF，即不使用核函数的方...	博文		阅读数 7万+
A*算法的C#实现	 qq_33885970	聚类算法(三) —— 基于密度的聚类算法		来自: jnulz的专栏
目录1, 概述2, A*算法的基本原理3, A	聚类算法是什么？——理解超平面 (SV...	博文		阅读数 1139
A*寻路算法详解	qq_33885970: 我仔细想了下，是距离。。。易	，概述 本文的主要内容是...		来自: 霜叶飞雪的博客
原文出处: A*PathfindingforBeginners	聚类方法(四)、基于高斯混合分布	博文		阅读数 2416
A*寻路算法初探	的聚类方法 (补充阅读)	!008/10/13/3071867.aspx...		来自: limiko space
作者: PatrickLester译者: Panic2005年	分类标签			阅读数 7315
手把手教你整合最优雅SSM框架：！	12篇	是从未认真读过相关的文章...		来自: 思月行云
小疯手把手带你整合SpringMVC+Sprin	 CSDN资讯	博文		阅读数 23万+
A*算法 (完整好用)	3篇	传统的SSH框架，把它们最...		来自: 小疯的代码健身房
配合我的一篇博客 A* 的分析实现 VS20	400-660-0108	8:30-22:00		08-04
A*算法之旅之初识A*算法（一）	44篇	机器学习/深度学习		下载
A*算法之初识A*算法（一）声明在前：这	关于	关于		阅读数 539
A*寻路初探	招聘 广告服务 网站地图	招聘 广告服务 网站地图		来自: CheerseUp的博客
原文出处: A*PathfindingforBeginners	百度提供站内搜索 京ICP备19004658号	广告		阅读数 1546
	©1999-2019 北京创新乐知网络技术有限公司	经营性网站备案信息		来自: showlong 专栏 「C...
	网络110报警服务	北京互联网违法和不良信息举报中心		
	中国互联网举报中心	家长监护		
	版权申诉	版权申诉		
	2019年1月	1篇		

基于A*算法的贪吃蛇小游戏

基于A*算法的贪吃蛇小游戏,可用户与aix

2018年9月

4篇



63

03-16

2018年8月

3篇

下载

2018年5月

4篇



2018年4月

4篇

推荐了, 返回首页

2018年3月

9篇



9

2018年1月

14篇



2017年12月

29篇



展开

**热门文章**[堪称最好最全的A*算法详解（译文）](#)

阅读数 52712

[自相关函数与互相关函数](#)

阅读数 52516

[超平面是什么? ——理解超平面 \(SVM开篇之超平面详解\)](#)

阅读数 44323

[理解卷积神经网络CNN中的特征图 feature map](#)

阅读数 26933

[目标跟踪算法——KCF 进阶](#)

阅读数 22983

**最新评论**[SVM C++ 实现](#)qq_42554262: 请问博主, matrix.h 和 math.h
这两个文件在哪呢[余弦窗\(汉宁窗\)的作用——图像预处理](#)

shitianmai6440: 是你原创的吗? 原作者连接都不加

[目标跟踪算法——KCF 进阶](#)qq_40877976: [reply]weixin_42126632[/reply]
fhog特征[目标追踪算法KCF \(kerneli...](#)sinat_30372583: 从1D到2D通过核函数可以转为
一维求解, 但是对于DCF, 即不使用核函数的† ...[超平面是什么? ——理解超平面 \(SV...](#)qq_33885970: 我仔细想了下, 是距离。。。易
受二维空间的影响, 容易把方程想成 $wx+b=y$ 

—



程序人生



CSDN资讯

QQ客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:30-22:00

[关于我们](#) [招聘](#) [广告服务](#) [网站地图](#)

✿ 百度提供站内搜索 京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉