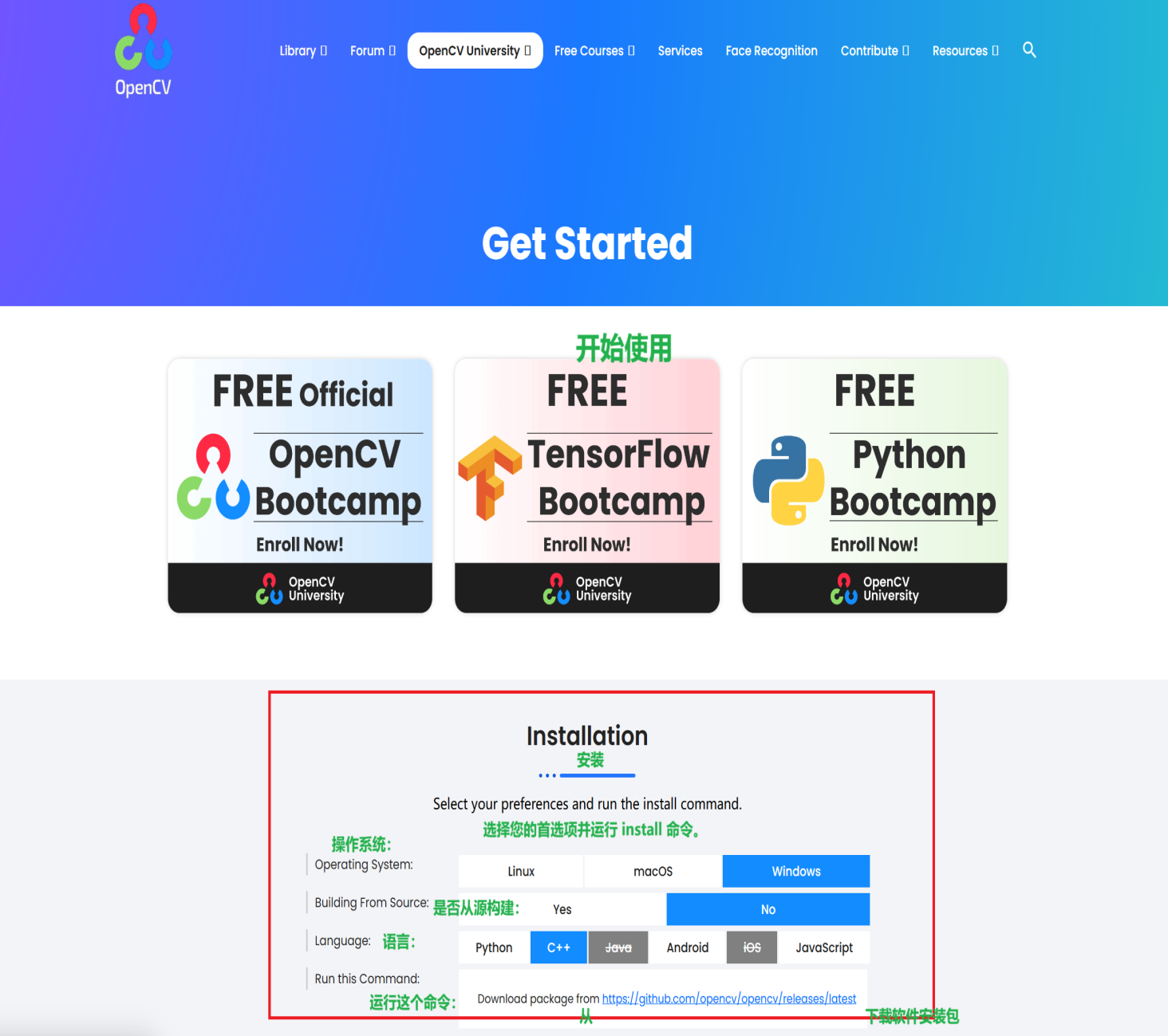
# 关于寻路与避障问题的解决方案与应用

常见的寻路孙发有A\*（A-Star）、IDA\*（是在A\*算法的迭代加深）、Breadth-First-Search（简称BFS，广度优先搜索）、Best-First-Search（简称BFS，最佳优先搜索）

）、Dijkstra（迪杰斯特拉）、Jump Point Search（简称JPS，跳点搜索）、Orthogonal Jump Point Search（正交跳点搜索，与JPS相似）等算法，考虑到实际应用中，地形相对比较复杂，本文档使用Jump Point Search算法（以下简称“JPS算法”或“跳点搜索算法”）减少一些运算以优化性能，且会使用3D、2D坐标系描述。由于JPS算法是基于A\*算法上改进的，本文档会适当加入一些A\*算法的原理等。

参考文献中的参考代码有C++、Python编程语言版本的示例代码，其中，Python部分使用了heapq、math、numpy、matplotlib、matplotlib、random等模块，C++部分使用了OpenCV，若对其不熟悉可以删除部分代码改为控制台输入输出或文件输入输出，也可以查看以下CSDN文章安装OpenCV：[http://blog.csdn.net/u011775793/article/details/134830399](https://blog.csdn.net/u011775793/article/details/134830399) ，其方法如下：

进入OpenCV官网（https://opencv.org/），并根据自身需要（如操作系统等）选择配置，并进入下载网址下载，例：



下载后，将其解压至目录“D:\opencv\opencv4.8.0”即可：sources:为源代码目录；build基于vs 2019构建的版本；include包含相关头文件x64使用vs 2019 构建的x64的Debug和Release版本；binwindows下相关exe工具，以及包含所有模块的动态库（dll），带'd'为Debug版本；lib包含所有模块的库文件，带'd'为Debug版本。

JPS算法的原理：

    强迫邻居的定义：当节点x的八个邻居中存在障碍，且节点x的父节点p，经过节点x到达节点n的距离代价总是小于不经过节点x到达节点n的任意路径的距离代价，则称节点n是节点x的强迫邻居。

跳点的定义：满足以下任意要求即为跳点：(i)节点x是起点或者终点；(ii)节点x至少有一个强迫邻居；(iii)节点x的父节点p在斜线方向，并且节点x的直线方向（水平或垂直）上存在满足条件一或二的节点。

以上所有内容均适用于很多场景，如游戏开发、导航等。且N维（如二维空间、三维空间）经过一定的修改后都适用。也可以使用伪随机让一个固定的种子只会生成一个固定的地形等功能。寻路与避障还需要搭配地形生成算法辅助生成地形，可以使用人为生成、侵蚀算法、柏林噪声、利用高度图等方法和算法，但这超出了本文的范围。

# 参考文献

寻路算法可视化网站：

<https://qiao.github.io/PathFinding.js/visual/>

JPS算法论文：

<http://grastien.net/ban/articles/hg-aaai11.pdf>

CSDN文章：

JPS算法：[http://blog.csdn.net/weixin\_51995147/article/details/141351237](https://blog.csdn.net/weixin_51995147/article/details/141351237)

JPS算法：[http://blog.csdn.net/LIQIANGEASTSUN/article/details/118766080](https://blog.csdn.net/LIQIANGEASTSUN/article/details/118766080)

C++语言的OpenCV的安装方法（含Linux、Windows操作系统）：<https://blog.csdn.net/u011775793/article/details/134830399>

参考代码：

（源于CSDN文章：[http://blog.csdn.net/weixin\_51995147/article/details/141351237](https://blog.csdn.net/weixin_51995147/article/details/141351237)

）：https://github.com/Benxiaogu/PathPlanning/tree/main/JPS