НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Факультет компьютерных наук
Образовательная программа бакалавриата «Прикладная математика и информатика»

АВТОМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ ТРАЕКТОРИИ

Выполнил студент группы БПМИ-173

Глушков Игорь Владимирович

Научный руководитель:

Руководитель проекта, доцент фкн НИУ ВШЭ, базовой кафедры «Интеллектуальные технологии системного анализа и управления» ФИЦ «Информатика и управление» РАН

доцент, канд. физ.-мат. наук

КРАТКОЕ ОПИСАНИЕ ПРОЕКТА

Программа предназначена для оптимального поиска пути от одной точки до другой в двумерном пространстве, используя в качестве приближения клетчатое поле, в итоге построив наиболее оптимальную ломанную. Задача сводится к поиску пути в графе между парой вершин. Областью применения может быть построение оптимальной траектории для роботов.

ОСНОВНЫЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ, ТЕРМИНЫ

Open-вершины – вершины, минимальное расстояние до которых на данной стадии алгоритма ещё не найдено в процессе алгоритма.
Close-вершины – вершины, минимальное расстояние до которых найдено в процессе алгоритма.
Open – список, хранивший open-вершины.
Close – список, хранивший close-вершины.
В процессе работы алгоритма рассчитывается функция f пути от стартовой вершины до конечной.
f = g + h.
g – наименьшее расстояние, найденное в процессе до от стартовой до конкретной вершины.
h – эвристическое приближение до конечной вершины.

АКТУАЛЬНОСТЬ РАБОТЫ

Программа может быть предназначена для пользователей, требующим возможность оптимального поиска пути в двумерном пространстве. Например при нахождении оптимального пути для беспилотных роботов, поиска пути на карте для курьера и других задачах на поиск пути в двумерном пространстве.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы

Программа находит оптимальное расстояние от одной точки, до другой, используя различные графовые эвристики.

Задачи работы

1. Определение наличия пути и самого пути в случае, если он существует.
2. Вывод данных, описывающий найденный путь в xml-файл
3. Вывод данных, описывающих эффективность пути(кол-во OPEN-вершин и общее число шагов)

ВЫБОР АЛГОРИТМОВ ДЛЯ РЕАЛИЗАЦИИ

Для реализации поиска пути используется алгоритмы: A*, дейкстра, а также различные эвристики, предназначенные для соответственного типа карты. Также реализованы разные типы движений. Среди эвристик используются: манхэттенская, диагональная, Евклидова и эвристика Чебышева.

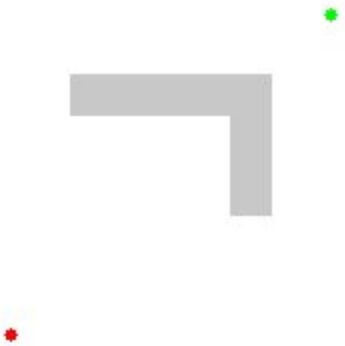
ОПИСАНИЕ A*

Алгоритм работает следующим образом. Для начала достаётся одна из вершин с минимальным расстоянием из open, с наиболее благоприятным для нас потенциалом, изменяя минимальное расстояние до всех вершин в open и соответственно потенциал, добавляет данную вершину в close.

Все вершины из OPEN отбираются по f, при этом задействован BREAKINGTIES, определяющий, по какому критерию стоит отбирать вершину, в случае равенства f: при меньшем значении g или при меньшем значении h. Вершины в OPEN добавляются в качестве соответственной пары в приоритетную очередь.

Вершины в CLOSE добавляются в хеш-таблицу в качестве значения соответственного указателя, при этом хранится в значении относительном размеру таблицы(если weight – ширина, a height – длина, x, y – координаты, то weight * x + y – соответствующее значение для нужной вершины).

ДЕМОНСТРАЦИЯ РАБОТЫ A*



ОПИСАНИЕ ТИПОВ ДВИЖЕНИЯ

Allowdiagonal – разрешено перемещение в случае, когда обе угловые клетки при перемещении свободны.
cutcorners – разрешено лишь в случае если занято не более одной клетки из угловых.
allowsqueeze – разрешено перемещение по диагонали независимо от боковых клеток во время перемещения.

ОПИСАНИЕ ТИПОВ ЭВРИСТИК

Пусть dx абсолютное расстояние по абсциссе, dy по ординате, тогда эвристики имеют вид

Манхэттенская – dx + dy
Евклидова – √(dx * dx + dy * dy)
Диагональная – (dx + dy) + (√2 - 2) * min(dx, dy)
Чебышева – (dx + dy) - min(dx, dy)

ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ

Для написания кода используется qt, код написан на c++. Использован алгоритм A* с различными эвристиками, аналогично приведённым ниже ссылками. Для реализации поиска пути использована стандартная приоритетная очередь, находящаяся в STL.

Все вершины хранятся в структуре Node, которая содержит в себе координаты вершины, указатель на предка, а также f, g, h показатели, которые отвечают за нужный потенциал. Т.е g = минимальное найденное расстояние до конкретной вершины, h – некий потенциал(величина эвристики), отвечающий за оценку расстояния до конечной вершины. f = g + h, т.е f – ф-ция, отвечающая за оценку расстояния, проходящую через данную вершину. Эти показатели нужны для A*. В дейкстре мы считаем h(v) = 0 для любой v.

Алгоритм представляет из себя поиск пути от одной вершины до другой, храня список open-вершин и close-вершин.

Для open и close используется STL. Используется приоритетная очередь из STL(priotiy_queue), которая хранит список из open-вершин. Для close используется хеш-таблица пар ключей(unordered_map).

ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ

Требуется найти следующее: существует ли путь(отвечает булевская переменная pathfound), длину пути(pathlength отвечает за длину, но принимает нулевое значение в случае отсутствии пути), список вершин, заданный в Node, по которым проходит путь в том виде, как его нашёл алгоритм(указатель на список в lppath), список вершин, заданный в Node, сокращённый по вектору направления, т.е в случае, когда соседние 3 координаты образуют прямую, то 2-я удаляется, т.к направление в данном случае не меняется(необходим для Theta*, указатель в hppath), общий список задействованных вершин(т.е суммарное кол-во вершин, находящихся в open и close, хранящихся в переменной nodescreated), общее число шагов алгоритма(хранится в numberofsteps). Все значения хранятся в структуре SearchResult.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Программа определяет наличие пути между вершинами, находит расстояние, а также сам путь с учётом всех эвристик.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

https://www.cs.helsinki.fi/u/bmmalone/heuristic-search-fall-2013/Korf1996.pdf
http://www.cs.cmu.edu/~maxim/files/ad_aij08_preprint.pdf
https://www.cs.cmu.edu/~maxim/files/tutorials/robschooltutorial_oct10.pdf
http://theory.stanford.edu/~amitp/GameProgramming/
https://ru.wikipedia.org/wiki/A*
http://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_A*

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Спасибо за внимание!

Глушков Игорь Владимирович,
ivglushkov@edu.hse.ru

Москва - 2019