

РЕАЛИЗАЦИЯ РАСЧЕТА ПОЛЯ РАССТОЯНИЙ ДЛЯ МОДЕЛИРОВАНИЯ ЛЮДСКОГО ПОТОКА

Попел Е.В.,

Научный руководитель канд. физ.-мат. наук Кирик Е.С.

Сибирский федеральный университет

При реализации математической модели движения человека во время эвакуации основным параметром, от которого зависит направление его движения, является расстояние до выхода из помещения. Для того, чтобы каждый раз не считать это расстояние из всех точек траектории движения человека, предварительно рассчитывается так называемое поле расстояний для всего здания. В процессе этого расчета здание покрывается мелкой ортогональной сеткой (шаг сетки обычно составляет 1-10 см), в узлах которой вычисляется минимальное расстояние до каждого выхода из здания. Затем эта информация записывается в отдельный файл, который в дальнейшем используется при каждом сеансе моделирования эвакуации в данном здании. Таким образом, поле расстояний является статическим и не зависит от положения людей.

Чтобы добиться максимальной точности вычислений надо, очевидно, делать шаг сетки наименьшим. Однако в таком случае на порядок возрастает и время расчета поля, и объем выходного файла. Поэтому остро стоит задача создания такого алгоритма расчета расстояний, который будет выдавать корректные результаты как можно быстрее, и в то же время, результаты эти должны в наименьшей степени отличаться от истинных значений.

1. Постановка задачи

Все пространство внутри здания (или этажа здания) за вычетом пространства, занимаемого препятствиями, будем называть расчетной областью.

Расчетная область ограничена заданным набором векторов (направленных отрезков) – прямых участков стен. Векторы заданы координатами их начала и конца в некоторой декартовой системе координат. Подразумевается, что вектора образуют замкнутую ломаную без самопересечений. Причем вектора ориентированы так, что внутренность здания лежит по правую сторону от вектора. По тому же принципу векторами заданы выходы из здания (или с этажа). Каждый из этих отрезков-выходов принадлежит одному из отрезков, задающих стены. Внутри полученного многоугольника таким же образом (множеством векторов, образующих замкнутые кривые без самопересечений) заданы элементы препятствий (мебель и т.п.).

Требуется получить оценку расстояния в каждой точке поля до каждого из выходов. Расстояние в данном случае – длина минимальной траектории от точки поля до выхода, причем ни один из участков траектории не должен пересекать стены и препятствия. Оценка расстояния в каждой точке поля должна быть оптимальна, т.е. одновременно должна соблюдаться минимальность погрешности оценки расстояния (минимальная разность между полученным значением расстояния и истинным, евклидовым значением длины траектории) и минимальность времени работы алгоритма расчета значения поля в каждой точке.

2. Решение

Для покрытия сеткой всей расчетной области на основании начальных данных находятся $x_{\min}, y_{\min}, x_{\max}, y_{\max}$ – минимальные и максимальные координаты концов отрезков, задающих положение стен. Эти координаты и будут задавать координаты крайних узлов сетки, которая будет иметь форму прямоугольника. Хранение в

оперативной памяти информации о расстоянии для каждого узла логически будет осуществляться в виде прямоугольной матрицы, число элементов в которой равно количеству узлов сетки. Шаг сетки обозначим d . Тогда через порядковые номера строки и столбца для ячейки можно выразить ординату и абсциссу узла, соответствующего выбранной ячейке матрицы:

$$\begin{cases} x_{ij} = x_{\min} + j \cdot d \\ y_{ij} = y_{\max} - i \cdot d \end{cases}$$

Разумеется, для каждого выхода будет свое поле расстояний и, соответственно, своя матрица узловых значений сетки. Опишем алгоритм расчета расстояний для одного произвольного фиксированного выхода. При этом все другие выходы будем считать отсутствующими. Поля расстояний для всех остальных выходов будут рассчитаны аналогично. Поле расстояний, учитывающее все выходы из области, вычисляется как минимальное по всем полям, рассчитанным для одиночных выходов, для каждого рассматриваемого узла сетки.

Итак, для начала создадим матрицу и заполним ее начальными величинами. Если какой-то узел сетки попал на линию выхода, то, очевидно, расстояние для этого узла до выхода равно нулю. Это значение и заносим в матрицу. Узлов, попавших на линию выхода, может быть несколько (в таком случае для каждого из них в матрицу запишем нули).

Возможна ситуация, что ни один узел сетки не попадет на линию выхода. В этом случае находятся, так называемые, опорные узлы сетки по следующему алгоритму:

- по геометрическому расположению выхода найти узлы, ближайšie к линии выхода. Узлы учитываем только те, что находятся справа от вектора, задающего линию выхода, т.е. эти узлы лежат внутри помещения (расчетной области);
- сгустить сетку в районе линии выхода, то есть в отдельные переменные записать координаты пересечения ребер сетки и линии выхода - узлы дополнительной сетки;
- узлам основной сетки, ближайшим к линии выхода, присвоить евклидово расстояние до линии выхода: каждому узлу найти среди узлов дополнительной сетки узлы с равными координатами по одной из осей (ординате или абсциссе).

В итоге, если линия выхода не параллельна координатным осям, наберется всего 3 узла (1 из основной сетки, 2 - из дополнительной), которые являются вершинами прямоугольного треугольника с известными сторонами. Искомое значение узла основной сетки будет равно длине высоты такого треугольника, опущенной на гипотенузу.

Теперь у нас имеются один или несколько узлов сетки, называемых «опорными», до которых мы будем искать расстояние. Все остальные ячейки матрицы заполняем максимальным значением (для типа double).

Алгоритм поиска расстояния между узлами сетки аналогичен алгоритму поиска минимального пути на графе. В таком случае в качестве вершин графа рассматриваются узлы сетки, которые соединяются между собой по определенному шаблону формирования смежных узлов (рис 1).

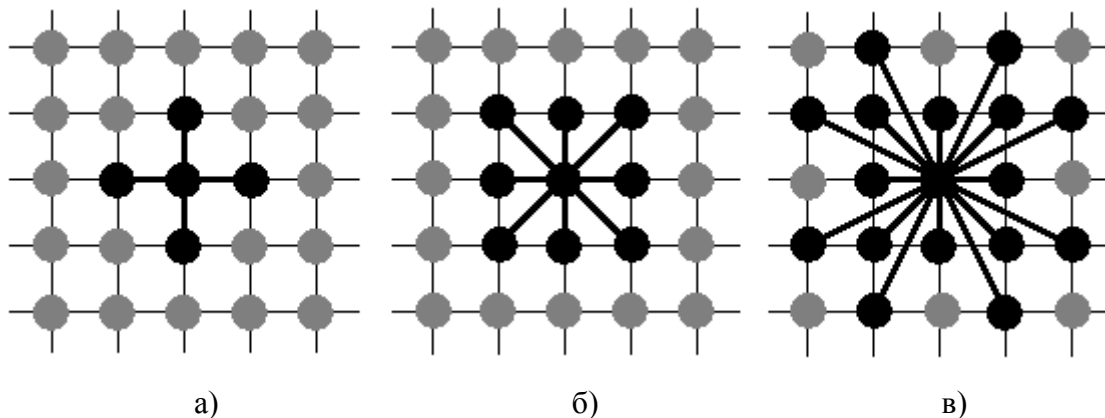


Рис. 1. Шаблоны смежных узлов: а) 4-узловой б) 8-узловой в) 16-узловой

Очевидно, чем большее число смежных узлов входит в шаблон, тем точнее вычисляется расстояние между узлами. Причем в сравнении с точным евклидовым расстоянием получаемое таким образом расстояние всегда будет больше.

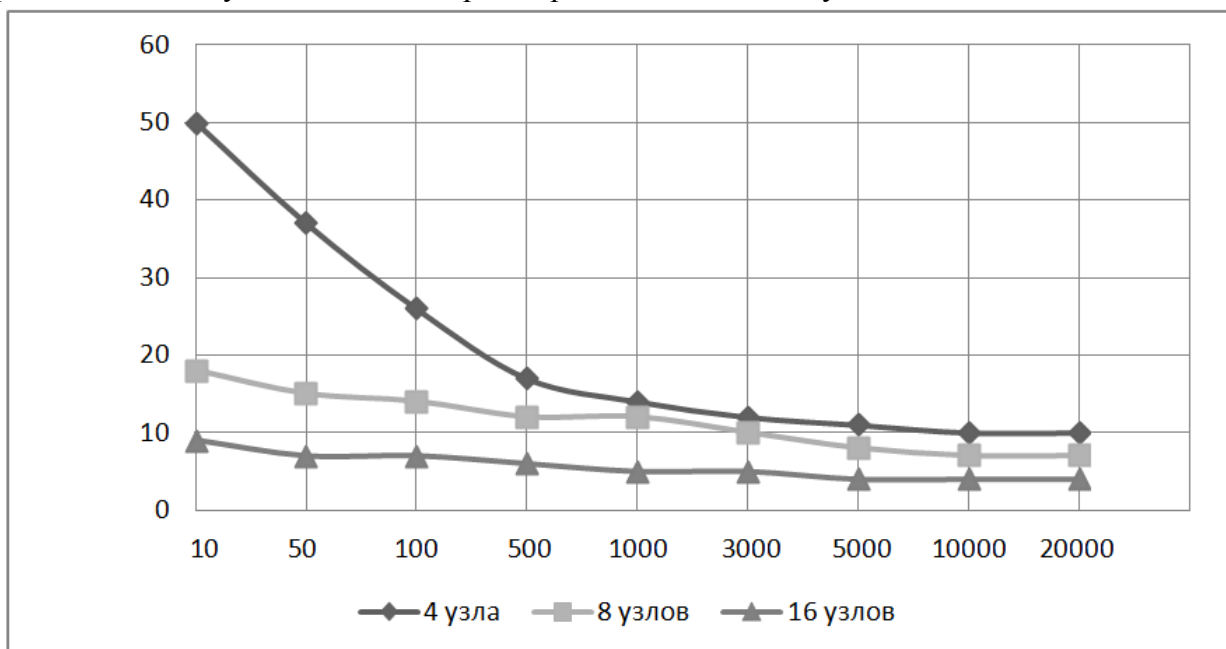


Рис. 2. Зависимость погрешности [%] от числа узлов в графе

На больших расстояниях погрешность уменьшается, что продемонстрировано на рисунке 2. Как видно из графика, 16-узловой шаблон позволяет достичь точности в 4% в то время как шаблон из 4 узлов ухудшает точность до 10%.

Помимо других вопросов для реализации алгоритма поиска минимального пути в графе необходимо на каждом этапе (рассмотрении очередного узла) исключать из рассмотрения те узлы, с которыми смежен данный узел согласно шаблону, но соединить их напрямую нельзя из-за препятствия. Для этого при рассмотрении очередной пары смежных узлов проводится проверка на пересечение ребром (между смежными узлами) какого-либо отрезка, являющегося стеной или границей препятствия. Таким образом, требуется проводить проверку на пересечение двух отрезков.

Вместо пересечения отрезков можно выполнять проверку на пересечение двух прямых, задаваемых этими отрезками. В результате, если прямые не параллельны, получим какую-то точку, которую надо проверить на принадлежность обоим отрезкам.

Для этого достаточно проверить, что эта точка принадлежит обоим отрезкам в проекции на ось X и ось Y.

Если же прямые оказались параллельными, то, если они не совпадают, то отрезки точно не пересекаются. Если же прямые совпали, то отрезки лежат на одной прямой, и для проверки их пересечения достаточно проверить, что пересекаются их проекции на оси X и Y.

Основными алгоритмами поиска минимального пути в графе являются: алгоритм Беллмана-Форда, Флойда-Воршалла, Дейкстры. Однако алгоритм Беллмана-Форда эффективен на разреженных графах, в то время как у нас у каждой вершины имеется порядка десяти смежных к ней вершин. Алгоритмы Дейкстры и Флойда-Воршалла выполняются за время одинакового порядка, но на практике после реализации каждого из алгоритмов было выявлено, что метод Дейкстры во всех случаях опережает действие алгоритма Флойда-Воршалла.

Таким образом, для наиболее эффективного расчета поля расстояний используется алгоритм Дейкстры с 16 узловым шаблоном связи.