

Система поиска минимального пути в среде с полигональными препятствиями

К. Заева, А. Семенов

Факультет прикладной математики и кибернетики

Тверской государственный университет, Тверь, Россия

kapitanka.ki@gmail.com, semenov@tversu.ru

Аннотация

В статье описан подход нахождения кратчайшего пути между двумя точками в двумерной среде с полигональными препятствиями, использующий непрерывное представление данных. Рассматривается задача векторизации и аппроксимации с контролируемой точностью бинарного черно-белого изображения для получения полигональных областей.

Ключевые слова: поиск минимального пути, среда с препятствиями, векторизация изображения, граф видимости.

1. ВВЕДЕНИЕ

Реализована система, позволяющая по карте местности, которая подается пользователем на вход, получить визуальное представление кратчайшего пути между любыми двумя заданными пунктами (точками). Карта местности представляет собой двумерную плоскость, а все препятствия на местности, сквозь которые невозможно пройти, аналогично, заданы двумерными непересекающимися фигурами (многоугольниками). Начальный и конечный пункты пути могут быть заданы любыми точками на плоскости, которые не являются частью препятствия. Заметим, что искомым путь может проходить вдоль границы объектов, но не может их пересекать. Иллюстрацию примера работы системы можно увидеть на изображении ниже (рис. 1).

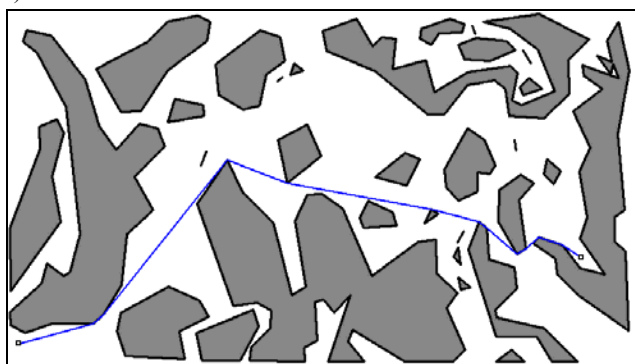


Рис. 1. Система поиска минимального пути в среде с полигональными препятствиями

Несмотря на наличие различных уже существующих способов решения задачи поиска минимального пути в двумерной среде с препятствиями, данная задача не теряет своей актуальности в связи с постоянным развитием приложений, которые требуют более быстрых и точных решений. Большинство уже существующих методов поиска пути предполагают, что пространство разбито на квадратные или шестиугольные ячейки. Для решения задачи непрерывное

пространство сводится к нескольким дискретным вариантам, к которым уже применяются различные алгоритмы поиска маршрута [3]. Однако дискретное решение имеет множество недостатков, основным из которых является возникновение ошибок накопления в ходе вычислений. Поэтому для получения более точных результатов мы используем непрерывный аналог решения задачи.

На вход системе подается бинарное черно-белое изображение – «карта» с препятствиями, где в качестве препятствий выступают простые полигоны без самопересечений. Исходное растровое изображение карты преобразовывается к непрерывному представлению с заданной точностью, после чего пользователь может выбирать начальную и конечную точки, получая кратчайший путь между ними. Минимальный путь вычисляется с помощью построения графа видимости полигональной области и нахождения маршрута на этом графе одним из классических алгоритмов [3].

2. ВЕКТОРИЗАЦИЯ ИЗОБРАЖЕНИЙ

В качестве среды с препятствиями может использоваться любое бинарное растровое изображение, которое система получает на вход. Для получения набора полигонов нам необходимо преобразовать контуры изображений из дискретного (растрового) представления в непрерывное (векторное) представление.

На вход задачи подается бинарное изображение, в котором присутствуют только два цвета: 1 – белый, 0 – черный. На выходе имеем последовательность граничных точек.

Вычисление границы растрового бинарного образа осуществляется через направленный перебор точек с целочисленными координатами вдоль границы объекта. Граничной точкой на растре называется точка, имеющая смежную точку другого цвета. Граничной парой (мостом) будем называть пару соседних точек на изображении, имеющих переход цветов. Обход границы осуществляется против часовой стрелки (рис. 2).

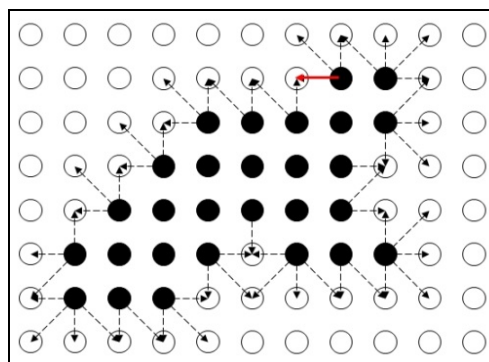


Рис. 2. Начало алгоритма обнаружения границ объекта

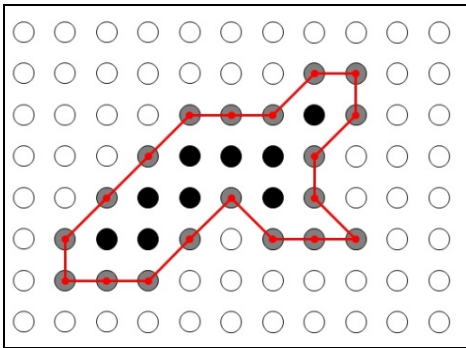


Рис. 3. Результат обнаружения границ объекта

Основные шаги выделения границ объектов [4]:

1. Проследивание начинается с обнаружения начальной граничной пары.

2. На каждом шаге один конец моста переносится в новую (пробную) точку. Основное правило перемещения моста:

- если пробная точка оказалась черной, то в нее переносится левый конец моста;
- если пробная точка оказалась белой, то в нее переносится правый конец моста.

3. Определение координат новой пробной точки:

Пусть левый (черный) конец моста имеет координаты $L = (Lx, Ly)$, правый (белый) конец моста – координаты $R = (Rx, Ry)$, пробная точка – координаты $T = (Tx, Ty)$. В зависимости от положения моста координаты точки T вычисляются следующим образом:

- вертикальное или горизонтальное положение моста

$$T = (Rx - (Ry - Ly), Ry + (Rx - Lx));$$

- диагональное положение моста

$$T = 1/2(Lx + Rx - Ry + Lx, Ly + Ry + Rx - Lx);$$

4. В граничную последовательность будет добавляться новая точка всякий раз, когда происходит перемещение левого (черного) конца моста

5. Условие завершения обхода границы – мост занял свое изначальное положение, с которого начиналось проследивание.

Алгоритм оконтуривания применяется последовательно ко всем объектам – препятствиям входного изображения. По завершению алгоритма выделения границ одного объекта все точки, участвующие в процессе, помечаются как уже посещенные, после чего мы переходим к «поиску» следующего объекта, если такой существует.



Рис. 4. Исходное входное изображение

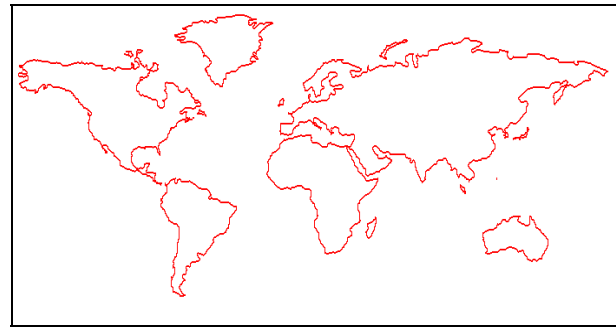


Рис. 5. Получение контуров объектов решением задачи векторизации

3. АППРОКСИМАЦИЯ ГРАНИЦ ОБЪЕКТОВ

Результатом выполнения алгоритма векторизации бинарного изображения является набор граничных точек. Однако если на вход мы подадим изображение не маленького формата, то этот набор будет иметь чрезвычайно большое количество объектов, так как рассматривается каждый пиксель изображения. Применять последующие действия к такому списку точек крайне нерационально, поэтому для дальнейших действий необходимо решить задачу аппроксимации набора граничных точек объекта.

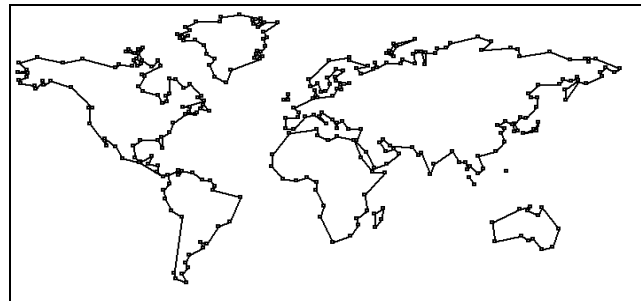


Рис. 6. Аппроксимация границ меньшим количеством точек (с точностью 3 пикселя)

На вход алгоритма мы получаем пронумерованную последовательность точек. Для решения задачи будем использовать простой метод кусочно-линейной аппроксимации, названный итеративным подбором концевых точек [5]. Идея такой кусочно-линейной аппроксимации состоит в следующем. Пусть нам дана последовательность точек $P_i, i = 0, \dots, N - 1$, являющихся граничными точками объекта, и определена точность аппроксимации ϵ .

- Решаем задачу кусочно-линейной аппроксимации отрезком прямой множества точек $P_i, i = 0, \dots, N - 1$.

- Определяем точку из множества P_i , максимально удаленную от полученного отрезка. В качестве расстояния между отрезком прямой и точкой P_i следует брать длину перпендикуляра, опущенного из точки P_i на отрезок.

- Если максимальное расстояние больше заданной наперед точности аппроксимации ϵ , то исходное множество разбивается на два подмножества, точкой максимально удаленной от построенного отрезка прямой, и задача аппроксимации рекурсивно решается отдельно для каждого из полученных подмножеств.

Ниже приведен пример подобной аппроксимации, где показаны четыре начальных итерационных шага для множества, состоящего из 19 точек (рис. 7).

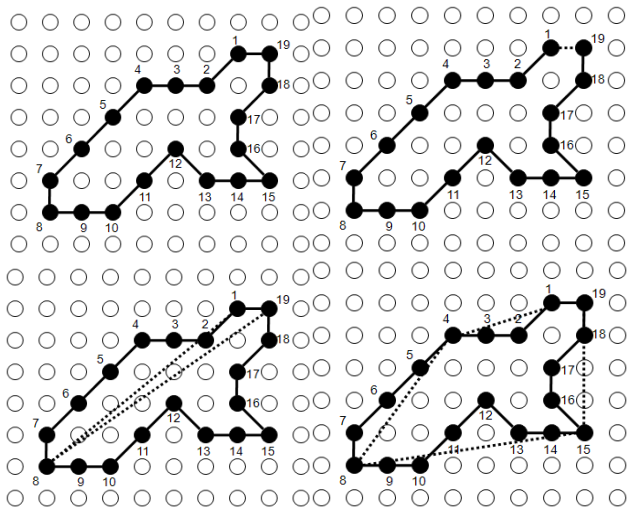


Рис. 7. Итерационный алгоритм аппроксимации

На выходе алгоритма мы получаем набор отрезков, которыми аппроксимируется заданная фигура. Начало и конец этих отрезков и составляют искомое нами множество граничных точек, по набору которых в дальнейшем будут строиться полигональные препятствия. Основное преимущество этого алгоритма состоит в его простоте.

4. ПОСТРОЕНИЕ ГРАФА ВИДИМОСТИ

Будем называть две точки взаимовидимыми, если отрезок, их соединяющий, не содержит в себе внутренних точек препятствий-многоугольников [1]. Пример взаимовидимых и не являющихся таковыми вершин многоугольника можно увидеть на иллюстрации ниже (рис. 8).

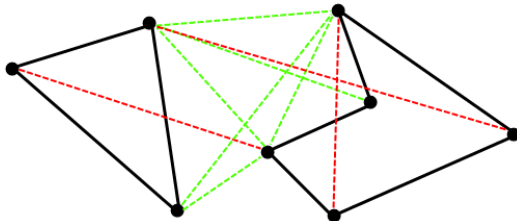


Рис. 8. Взаимовидимые (соединены зеленым цветом) и не взаимовидимые (соединены красным цветом) вершины многоугольника

Заметим, что соседние вершины всегда являются взаимовидимыми (на рис. 8 отмечены только видимые не соседние вершины).

Вершинами графа видимости являются вершины препятствий-многоугольников. Рёбрами соединены только взаимовидимые вершины графа. Кратчайший путь на полученном графе видимости и будет являться искомым минимальным маршрутом между двумя точками [2].

Таким образом, для построения графа видимости полигональной области необходимо найти все пары взаимовидимых вершин из множества, состоящего из вершин всех полигональных препятствий на плоскости. То есть для каждой пары вершин мы должны проверить, пересекает ли соединяющих их отрезок какую-либо сторону из всех существующих многоугольников. Однако такой подход

выполняется большим количеством итераций и не является эффективным. Для уменьшения времени работы, затрачиваемого на построения графа видимости, можно внести некоторые дополнительные проверки. Как уже было упомянуто выше, соседние вершины многоугольника всегда взаимовидимы, и ребро графа будет совпадать со стороной фигуры, соединяющей эти вершины. Два ребра, исходящие из одной вершины, образуют некий сектор (рис. 9), и все вершины, лежащие внутри него, не могут являться взаимовидимыми с вершиной сектора. Таким образом, проверке подвергаются только вершины, лежащие вне сектора, образуемого исследуемой вершиной, что позволяет сократить количество проверок.

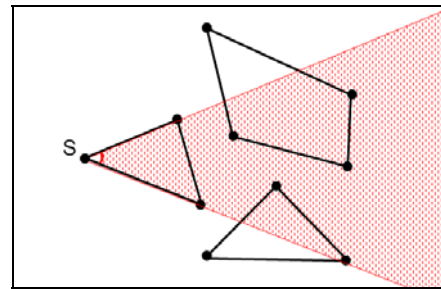


Рис. 9. Сектор обзора для вершины S

Заметим, что построение кратчайшего пути по графу видимости будет вестись лишь по выпуклым вершинам многоугольника (вершинам, внутренний угол которых меньше 180 градусов). То есть проверки можно сократить до проверок отрезков, соединяющих точки из множества всех выпуклых вершин, что существенно уменьшит количество общих проверок на взаимовидимость.

Проиллюстрируем пример построения графа видимости для областей, состоящих из нескольких полигональных объектов (рис. 10, 11).

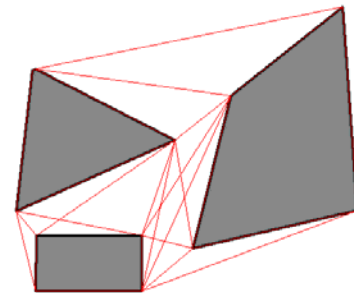


Рис. 10. Граф видимости для нескольких простых полигональных фигур

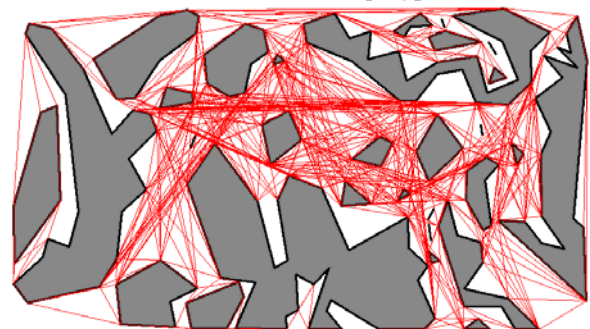


Рис. 11. Граф видимости для различных полигональных областей

5. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Для решения задачи поиска минимального пути между двумя точками граф видимости расширяется – добавляются начальная и конечная точки поиска пути к списку вершин, после чего отрезками соединяются видимые с ними вершины, аналогично описанному выше алгоритму. По полученному графу ищется кратчайший путь одним из стандартных алгоритмов поиска пути на графе (рис. 12, 13).

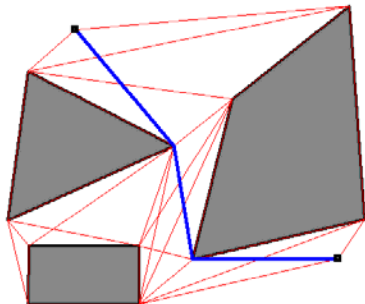


Рис. 12. Минимальный путь между двумя точками, найденный по графу видимости

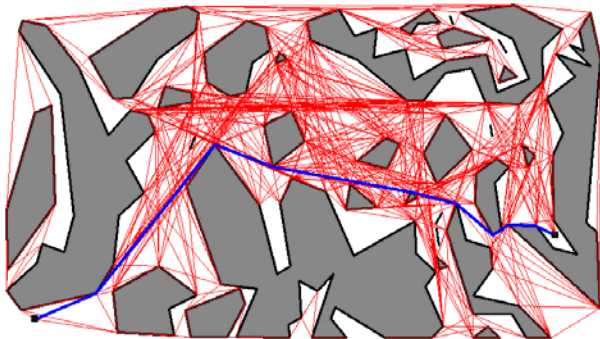


Рис. 13. Минимальный путь между двумя точками, найденный по графу видимости

В качестве проверки и апробации разрабатываемой системы были проведены вычислительные эксперименты для нахождения кратчайшего пути в произвольном лабиринте. В систему было загружено изображение лабиринта, после чего оно путем векторизации было приведено к набору полигональных областей, по которым был построен граф видимости. Найденный минимальный путь между заданными точками по графу (рис. 14, 15) будет являться кратчайшим маршрутом в лабиринте. Система решает данную задачу с высокой точностью и за короткий интервал времени.

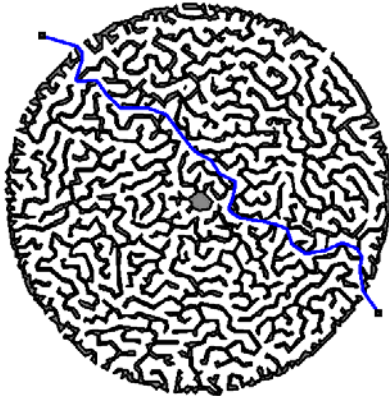


Рис. 14. Нахождение кратчайшего пути между двумя точками в лабиринте

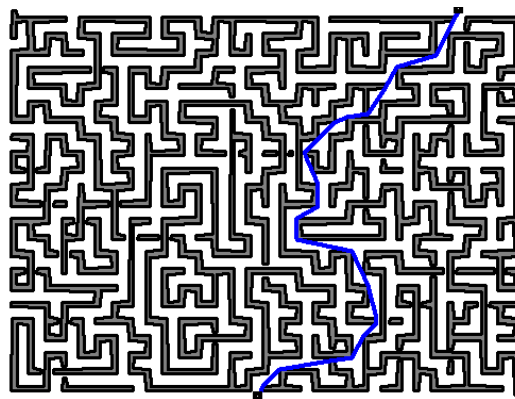


Рис. 15. Нахождение кратчайшего пути между двумя точками в лабиринте

6. ЗАКЛЮЧЕНИЕ

Описанная в статье система поиска минимального пути в среде с полигональными препятствиями представляет собой эффективный и простой инструмент, который легко может быть использован в различных приложениях.

7. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта РФФИ № 12-07-00622-а.

8. ССЫЛКИ

- [1] Asano T., Guibas L.J., Hershberger J., Imai H. Visibility of disjoint polygons. *Algorithmica*, 1: 49 - 63, 1986.
- [2] Berg M., Cheong O., Kreveld M., Overmars M. *Computational Geometry: Algorithms and Applications*. Springer-Verlag Berlin Heidelberg, Third Edition: 323 - 333, 2008.
- [3] Stout V. (оригинальная статья), 1997, Maxim Kamensky (перевод), 2000). Алгоритмы поиска пути. <http://pmg.org.ru/ai/stout.htm>
- [4] Местецкий Л.М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. – М., Физматлит, 2009.
- [5] Семенов А.Б. Методы моделирования «живого» персонального почерка // Труды международной конференции «ГРАФИКОН-2009». – М.: МГУ, 2009. – С. 390 – 391.

Об авторах

Заева Камила – студентка магистратуры кафедры информационных технологий Тверского государственного университета.

Email: kapitanka.ki@gmail.com.

Семенов Андрей – кандидат физико-математических наук, доцент кафедры информационных технологий Тверского государственного университета.

E-mail: semenov@tversu.ru.