

АЛГОРИТМ ПОСТРОЕНИЯ ГРАФА ДЛЯ НАВИГАЦИИ ВНУТРИ ПОМЕЩЕНИЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДА STRAIGHT SKELETON

Котенко Н.А., Белов Ю.С.

*Московский государственный технический университет имени Н.Э. Баумана,
Москва, e-mail: nikita_kot2@mail.ru, yselov@bmstu.ru*

Одной из наиболее популярных функций современных интерактивных карт является поиск оптимальных маршрутов, который обычно выполняется на основе навигационного графа. Целью данной работы было рассмотрение алгоритма генерации навигационного графа для схемы помещения. В ходе исследования выполнено сравнение методов Medial Axis и Straight Skeleton для построения скелета геометрической фигуры. Был представлен и описан алгоритм, использующий метод Straight Skeleton для формирования скелета и определения полного контура помещения, с выделением ключевых точек и их последующей фильтрацией. Предложен метод формирования ребер графа с проверкой на пересечение с контуром помещения и оптимизацией соединений через удаление ребер с неоптимальными углами. Определены потенциальные улучшения алгоритма, которые можно внести для генерации более эффективного навигационного графа. Сделаны выводы, что уже на данный момент алгоритм позволяет формировать оптимальный навигационный граф для вычисления кратчайших путей, близких к реальным маршрутам, при минимальных требованиях к памяти. Данное исследование призвано продемонстрировать алгоритм, основанный на применении анализа изображений для формирования навигационного графа и его использования в реальных задачах по навигации на интерактивной карте.

Ключевые слова: навигационный граф, алгоритм, straight skeleton, medial axis, вершины, ребра, оптимальный маршрут

AN ALGORITHM FOR CONSTRUCTING A GRAPH FOR INDOOR NAVIGATION USING THE STRAIGHT SKELETON METHOD

Kotenko N.A., Belov Yu.S.

*Bauman Moscow State Technical University, Moscow,
e-mail: nikita_kot2@mail.ru, yselov@bmstu.ru*

One of the most popular features of modern interactive maps is the search for optimal routes, which is usually performed based on the navigation graph. The purpose of this work was to consider an algorithm for generating a navigation graph for a room layout. The study compared the methods of Medial Axis and Straight Skeleton for constructing the skeleton of a geometric figure. An algorithm was presented and described using the Straight Skeleton method to form a skeleton and determine the complete contour of the room, with the selection of key points and their subsequent filtering. A method is proposed for forming graph edges with a check for intersection with the contour of the room and optimizing connections by removing edges with suboptimal angles. Potential algorithm improvements that can be made to generate a more efficient navigation graph have been identified. It is concluded that already at the moment the algorithm allows you to form an optimal navigation graph for calculating the shortest paths.

Keywords: navigation graph, algorithm, straight skeleton, medial axis, vertices, edges, optimal route

Введение

Интерактивная карта – современный и крайне удобный способ представления географических данных с множеством функций, среди которых одна из самых востребованных – это навигация [1]. Но для реализации поиска оптимальных маршрутов на интерактивной карте офиса, помещения или другого замкнутого пространства необходимо иметь соответствующий навигационный граф, по которому будут строиться кратчайшие пути. Поэтому можно считать актуальной разработку алгоритма для генерации навигационного графа на основе схемы помещения.

Цель исследования – рассмотреть основные шаги разработанного алгоритма генерации навигационного графа для помещений на основе скелета его схемы, формирующегося с использованием метода Straight Skeleton.

Материалы и методы исследования

Задачу, которую решает исследуемый алгоритм, можно сформулировать следующим образом: необходимо построить оптимальный навигационный граф на основании схемы помещения для последующего его использования в определении оптимальных маршрутов на интерактивной карте.

Навигационный граф – структура, моделирующая набор возможных маршрутов внутри помещения или любого другого пространства, для которого есть необходимость анализа и поиска кратчайших путей или любых других задач, связанных с графами. Каждая вершина имеет координаты по ОХ и ОУ. Ребра соединяют вершины между собой и могут иметь предустановленный вес [2].

Критерием оптимальности графа можно считать адекватное как количество, так и расположение вершин графа в двумерном пространстве, а также набор ребер, соединяющих узлы графа.

Построенная структура помещения должна позволять строить оптимальный маршрут от одной точки до другой так, чтобы быть максимально схожим с тем, как человек добирается от одного места к другому в реальной жизни.

Поэтому граф обязательно должен быть связным, либо в случае наличия набора островных точек необходимо потом вручную их удалять или соединять с графом другого помещения (этажа).

Минимизация количества вершин и ребер графа связана с необходимостью хранить его отдельно в базе данных и памяти приложения. Поэтому важно получить такой набор вершин и ребер, который позволит эффективно строить кратчайшие пути и при этом не будет занимать много памяти.

Общая структура разработанного алгоритма представлена на следующей схеме (рис. 1). Далее последует подробный разбор каждого шага алгоритма из представленной схемы и применяемых методов.

Методы Medial Axis u Straight Skeleton

В основе алгоритма лежат методы для построения скелета помещения, из которого далее будут выделяться ключевые точки и проводиться ребра графа. Среди наиболее популярных методов можно выделить два основных: Medial Axis (медиальная ось) и Straight Skeleton (прямолинейный скелет).

Medial Axis можно рассматривать как отдельное определение – набор точек, имеющих более одной ближайшей точки на контуре геометрического объекта. То есть для каждой точки медиальной оси можно построить окружность определенного радиуса, и она будет касаться внешнего контура в двух и более точках [4].

Таким образом, получается топологический скелет, который определяет форму и сохраняет основные детали геометрической фигуры.

Straight Skeleton же сильно схож с Medial Axis, но имеет отличие в использовании преимущественно прямых отрезков в своей основе, в то время как медиальная ось может включать в себя параболические кривые (рис. 2) [5].

Именно данное различие повлияло на выбор прямолинейного скелета в качестве основы для рассматриваемого алгоритма, так как такой формат позволит проще определять ключевые точки. Рассматриваемый алгоритм можно адаптировать и под использование Medial Axis в качестве скелета помещения, но придется переопределить набор ключевых точек.

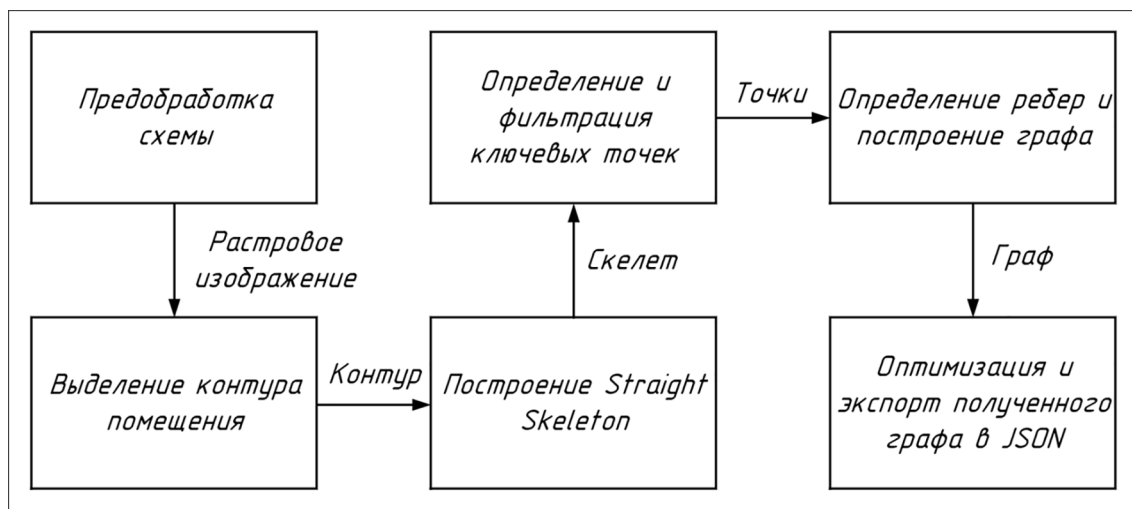


Рис. 1. Схема алгоритма
Источник: составлено авторами на основе [3]

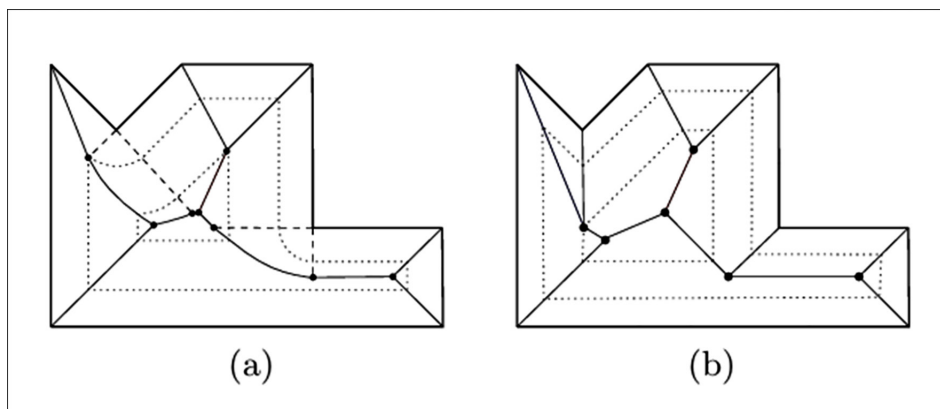


Рис. 2. Сравнение скелетов многоугольника (a – Medial Axis, b – Straight Skeleton)
Источник: составлено авторами на основе источников [4; 5]

Предобработка схемы

Перед выполнением основных шагов алгоритма необходимо провести преобразование схемы помещения. Нужно убрать лишние детали, которые могут помешать определить контур и скелет помещения, и преобразовать схему (из SVG или другого формата) в растровое изображение.

Так как в итоге выполнения алгоритм экспортирует граф с координатами по ОХ и ОУ, то важно, чтобы загружаемое изображение было оригинального разрешения для корректного наложения полученного графа при загрузке в интерактивную карту.

В случае с форматом SVG это делается достаточно просто, если правильно настроена разметка, которая позволит вручную или программно удалить лишние части схемы. Но при преобразовании такой схемы в растровое изображение могут возникнуть проблемы с масштабированием, особенно при использовании онлайн-сервисов. Здесь необходимо определить коэффициент масштабирования разрешения схемы, временно уменьшить ее разрешение с помощью SVG-атрибутов кратно полученному коэффициенту и выполнить преобразование в растр.

В итоге изображение обрабатывается алгоритмом: выполняется загрузка, бинаризация и устранение шумов, которое убирает небольшие помехи и не препятствует дальнейшему формированию корректного контура помещения.

Выделение контура помещения

Далее следует шаг по вычислению контура помещения. Данный алгоритм реализован в математических пакетах языка программирования Python, поэтому для по-

строения контура использовалась функция библиотеки OpenCV *findContours*.

Важно также при определении контура использовать в функции для параметра *mode* значение *RETR_TREE* для получения иерархии контуров и далее объединить эту иерархию в единый полный контур помещения. Это необходимо для последующей оптимизации ребер графа. На выходе получается бинарная матрица, которая показывает, есть ли точка контура для конкретного пикселя изображения.

Построение скелета помещения

Далее необходимо построить скелет помещения. Метод Straight Skeleton реализован в библиотеке *skimage.morphology* и представлен функцией *skeletonize*. Также в библиотеке есть функция *medial_axis* для построения медиальной оси изображения, так что алгоритм можно доработать и выборочно использовать как прямолинейный скелет, так и медиальную ось в качестве основы для генерации графа. Но в случае Medial Axis будут отличаться дальнейшие шаги по выделению ключевых точек из-за ранее упомянутых различий двух подходов к построению скелета. На выходе данного блока алгоритма также получается бинарная матрица, которая показывает, присутствует ли точка скелета в конкретном пикселе изображения.

Выделение и фильтрация ключевых точек

Далее предстоит выделить ключевые точки скелета, которые станут вершинами графа. Для всех точек прямолинейного скелета (в виде бинарной матрицы) можно применить 8-связный шаблон: пиксель считается связанным с другим, если является соседом как по горизонтали, так и по диагонали матрицы [3].

Выделяемые ключевые точки можно поделить на 3 типа:

1. Листья – в качестве соседа имеют только одну точку. Обычно представляют собой переходы в кабинеты, лестницы, лифты и т.д.

2. Узлы – в качестве соседей имеют 3 и более точек, то есть являются разветвляющимися точками.

3. Точки изгиба – с них начинается изгиб отрезков прямолинейного скелета, то есть для этих точек применим формат шаблона, при котором они имеют одного горизонтального соседа и при этом одного соседа по диагонали, что отражает положение точки на углу (не 90° и 180°) соединения двух отрезков скелета.

Так как количество точек после их поиска на скелете может получиться огромным (особенно точек изгиба из-за неровности отрезков), то необходимо отфильтровать их относительно друг друга по расстоянию между собой. Было определено, что оптимально фильтровать каждый полученный набор точек по отдельности, листья относительно узлов и точки изгиба относительно набора узлов и листьев.

Так как приемлемое расстояние между точками может зависеть от разрешения изображения, то предусмотрено назначение минимальных допустимых расстояний в качестве констант. На выходе получается три набора ключевых точек, для которых определены их координаты по ОХ и ОУ.

Построение графа

Следующим шагом алгоритма будет непосредственно формирование графа на основе ранее выделенных ключевых точек. Но сначала необходимо понять, как проводить ребра между вершинами.

Принцип формирования ребер между ключевыми точками следующий: для каждой вершины проводятся аппроксимированные отрезки к другим ключевым точкам, и если данное ребро пересекает или проходит слишком близко к полному контуру помещения, то это ребро не рассматривается. Все остальные ребра сохраняются и входят в сформированный граф [3]. Такой подход позволит соединять между собой только те точки, на пути которых нет преград.

Оптимизация полученного графа

Но полученный набор ребер все равно избыточен из-за множества перекрестных соединений или соединений под слишком острыми или тупыми углами. Самым оптимальным набором ребер можно считать такой набор, в котором соединения идут под $\sim 45^\circ$, $\sim 90^\circ$ и $\sim 180^\circ$ [6].

Поэтому необходимо отфильтровать полученный набор ребер с учетом неоптимальных углов. Сделать это можно перебором всех ребер, вычислением углов между ними с помощью формулы косинуса угла между векторами и удалением тех соединений, которые проходят под некорректным углом. Для этого также была выделена константа, определяющая «чувствительность» таких углов.

Дальше остается только экспортировать сформированный граф в нужный формат (например, JSON) и использовать его для построения кратчайших путей на интерактивной карте [1].

Результаты исследования и их обсуждение

В качестве примера работы алгоритма выбран третий этаж учебного корпуса № 3 кампуса Калужского филиала МГТУ им. Н.Э. Баумана. Схема этажа представлена в виде SVG-документа. Выполнено преобразование плана коридора в растровое изображение с сохранением оригинального разрешения. В результате выполнения алгоритма получается следующий граф, наложенный на схему коридора (рис. 3).

Результат программного поиска кратчайшего маршрута между аудиториями на интерактивной карте представлен на следующем изображении (рис. 4).

Хотя сейчас алгоритм генерирует вполне адекватный и рабочий граф, все равно есть моменты, которые можно усовершенствовать.

1. Объединение близких точек в кластеры. Вместо фильтрации можно объединять рядом стоящие вершины и аккумулировать все соединения в этот кластер. Данное улучшение сделает граф более подходящим для поиска реальных маршрутов [7].

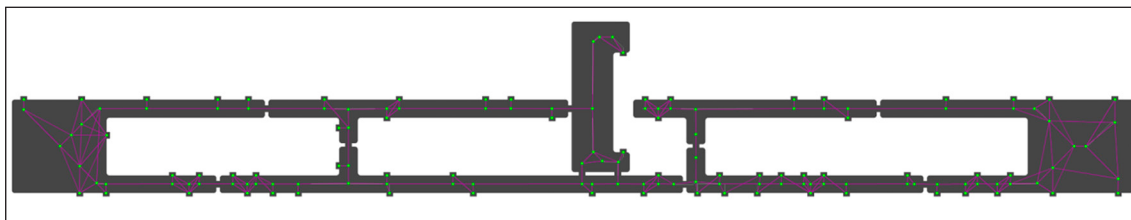


Рис. 3. Результат применения алгоритма на схеме коридора третьего этажа УАКЗ кампуса КФ МГТУ
Источник: составлено авторами по результатам работы алгоритма

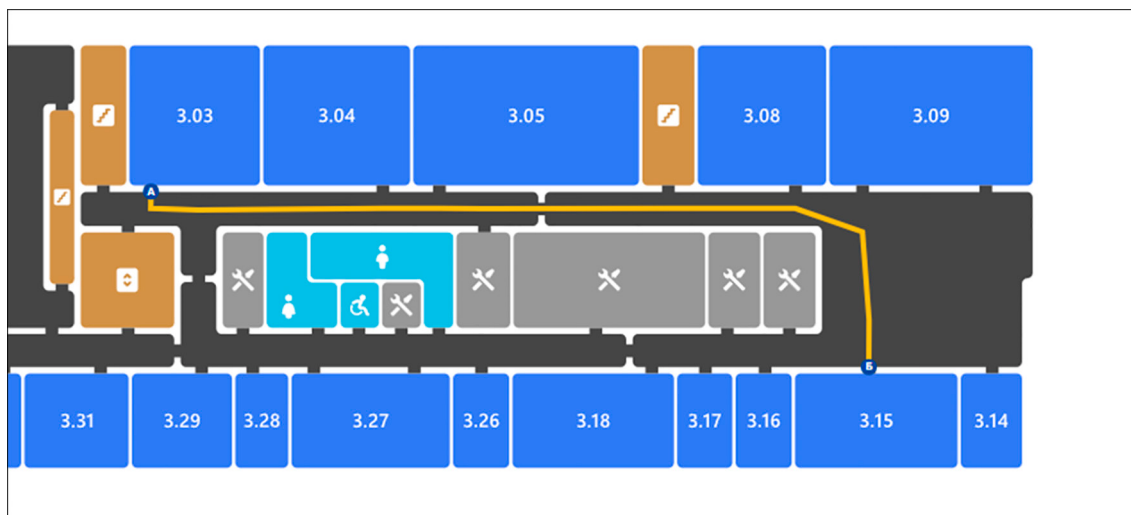


Рис. 4. Пример программного построения оптимального маршрута на основе сформированного графа
 Источник: составлено авторами по результатам работы алгоритма

2. Расширение области генерации графа до аудиторий. Сейчас граф строится только на однородном коридоре и не учитывает кабинеты. Это не очень удобно, так как приходится преобразовывать схему, убирая лишние помещения и детали, и исключаются случаи, когда нужно расширить область графа до внутренних комнат.

3. Доработка фильтрации ребер. Сейчас алгоритм удаляет множество лишних ребер за счет выборки по углу, но часто отсекаются корректные ребра или остаются некорректные при небольших изменениях порогового угла.

4. Реализация собственного алгоритма скелетизации помещения. Это позволит на этапе построения прямолинейного скелета отсечь ненужные точки и оставить только ключевые, что повысит производительность алгоритма [5].

Заключение

В данной статье были рассмотрены основные шаги алгоритма построения графа для навигации внутри помещения и обоснован выбор метода Straight Skeleton вместо Medial Axis. Разработанный алгоритм позволяет быстро вычислять навигационный граф по схеме помещения, но имеет ряд недоработок, которые можно исправить в будущем.

Список литературы

1. Котенко Н.А., Белов Ю.С. Анализ архитектуры веб-приложения интерактивной карты университета // Научные технологии в приборостроении и машиностроении и развитие инновационной деятельности в вузе: материалы региональной научно-технической конференции, (Калуга 23–25 апреля 2024 г.). Т. 2. Калуга: Издательство МГТУ им. Н.Э. Баумана, 2024. С. 22–27.
2. Flikweert P., Peters R., Díaz Vilariño Lucia, Voûte Robert, Staats B. Automatic Extraction of a Navigation Graph Intended for Indoor Navigation From an Indoor Point Cloud // ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. 2019. Vol. IV-2/W5. P. 271–278. DOI: 10.5194/isprs-annals-IV-2-W5-271-2019.
3. Clementini Eliseo, Pagliaro Andrea. The Construction of a Network for Indoor Navigation. 6th International Conference on Geographical Information Systems Theory // Applications and Management. 2020. P. 254–261. DOI: 10.5220/0009488902540261.
4. van Toll Wouter, Cook IV, Kreveld Marc, Geraerts Roland. The Explicit Corridor Map: A Medial Axis-Based Navigation Mesh for Multi-Layered Environments. 2017. P. 1–33. DOI: 10.48550/arXiv.1701.05141.
5. Cheng S.-W., Mencil Liam, Vigneron A. A faster algorithm for computing straight skeletons // ACM Transactions on Algorithms. 2016. Vol. 12. No. 3. P. 1–21. DOI: 10.1145/2898961.
6. Fu Meiqing, Qi Bing, Issa Raja. Generating straight skeleton-based navigation networks with Industry Foundation Classes for indoor way-finding // Automation in Construction. 2020. Vol. 112. No. 4. P. 1–13. DOI: 10.1016/j.autcon.2019.103057.
7. Mortari Filippo, Clementini Eliseo, Zlatanova Sisi, Liu Liu. An indoor navigation model and its network extraction // Applied Geomatics. 2019. Vol. 11. No. 2. P. 413–327. DOI: 10.1007/s12518-019-00273-8.