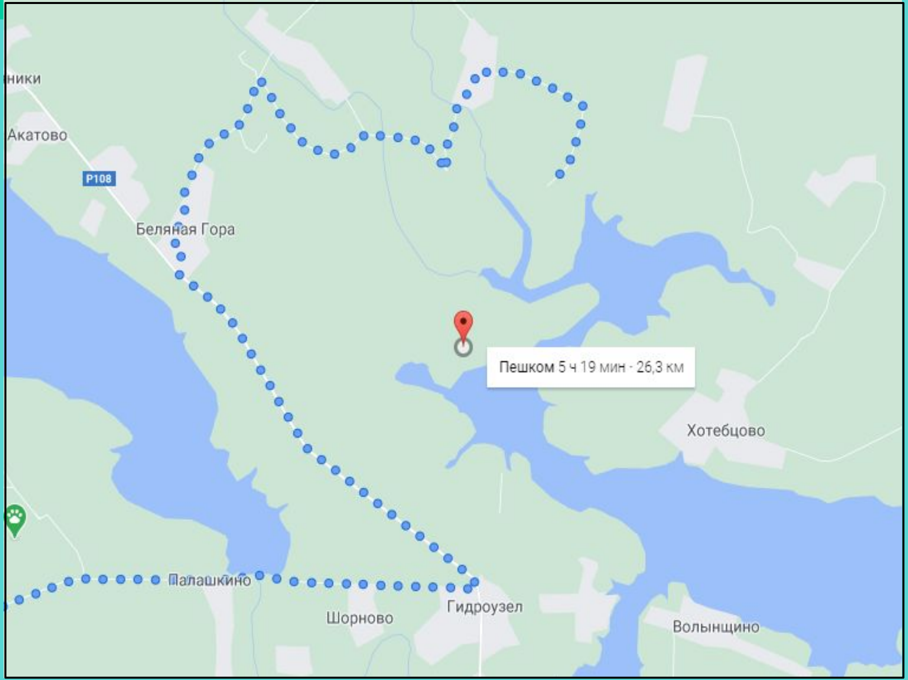
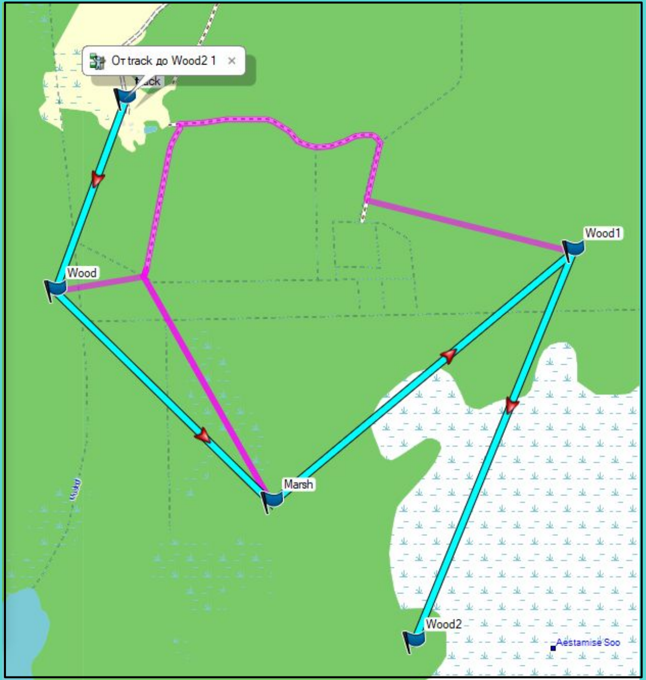
Методы и алгоритмы построения маршрутов на пересеченной местности

Козуб Денис

Как появилась идея?



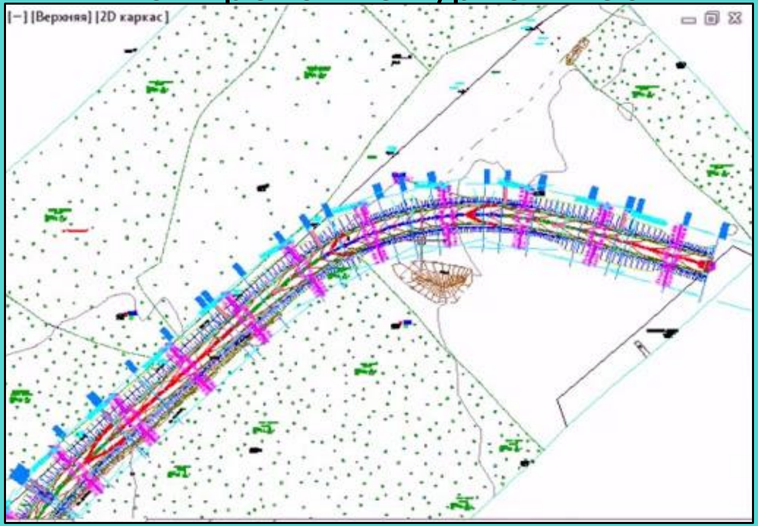
Google Maps



Garmin BaseCamp

Область применения

- ❑ Расширение возможностей существующих систем
- ❑ Прокладывание дорог, газо- и нефтепроводов, линий электропередач
- ❑ Планирование спасательных операций
- ❑ Планирование туристических маршрутов

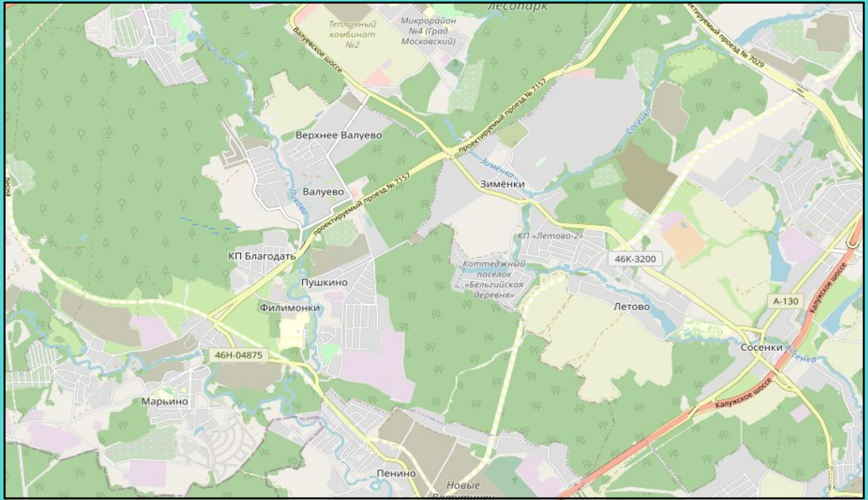


Основные требования:

1) Скорость построения маршрута
2) Минимизация хранимых данных
3) Регулируемая точность маршрута
4) Динамическое изменение характеристик

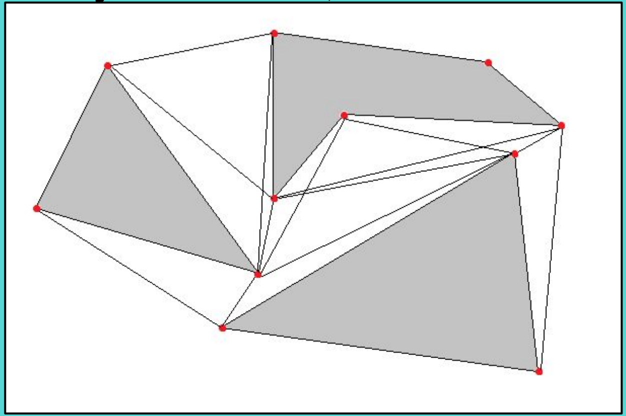
Дискретизация местности

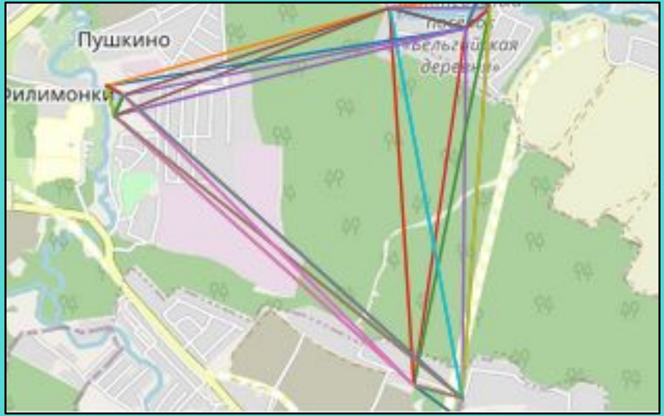
1) Построить граф
2) Определить веса ребер
3) Применить алгоритм поиска



Граф видимости

Граф видимости для множества многоугольников на плоскости - граф, в вершинах которого находятся вершины многоугольников, а ребра соединяют вершины, являющиеся видимыми друг для друга (никакое ребро графа видимости не пересекает ни один из данных многоугольников).





Определение веса ребра

- ➔ Различные типы поверхностей
- ➔ Погодные условия
- ➔ Время года
- ➔ Время суток

1) Собраны данные о туристических походах по пересеченной местности и сельских дорогах

2) Данные собраны в Европе, Азии и Америке. Общая протяженность - несколько десятков тысяч километров.

3) В будущем с помощью этих данных будут определен вес ребра для каждой из поверхностей

Иерархический подход

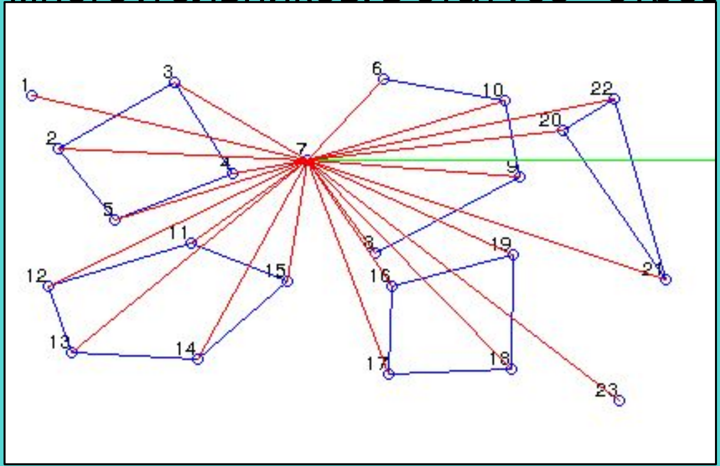
Время построения маршрута не должно зависеть от размеров области и количества объектов на ней. Иерархический подход:

- Построение маршрута до дороги -> маршрут по дороге -> маршрут до точки назначения
- Рассмотрение только элементов, размеры которых сопоставимы с размерами области, на которой строится маршрут

Вывод: не получится заранее построить граф видимости. Встает задача эффективной обработки массовых запросов на построение всех ребер видимости для данной точки q и множества многоугольников $P = \{P_i\ i=1...h\}$. Это задача нахождения ребер графа видимости, инцидентных данной вершине без построения графа видимости для алгоритма поиска пути.

Построение графа видимости

Пусть $n = \sum_{i=1}^{h} n_i$ где n_i - кол-во вершин в многоугольнике P_i. Тогда n - общее количество вершин. Алгоритм вращающейся заметающей прямой: O(n log n) В очереди событий находятся вершины многоугольников, в статусе - отрезки, которые пересекает заметающая прямая. Затраты памяти линейны. Проблемы: большая временная сложность, итоговый граф не оптимизирован. Ребра оптимизиро- ванного графа - опорные прямые к многоугольникам.

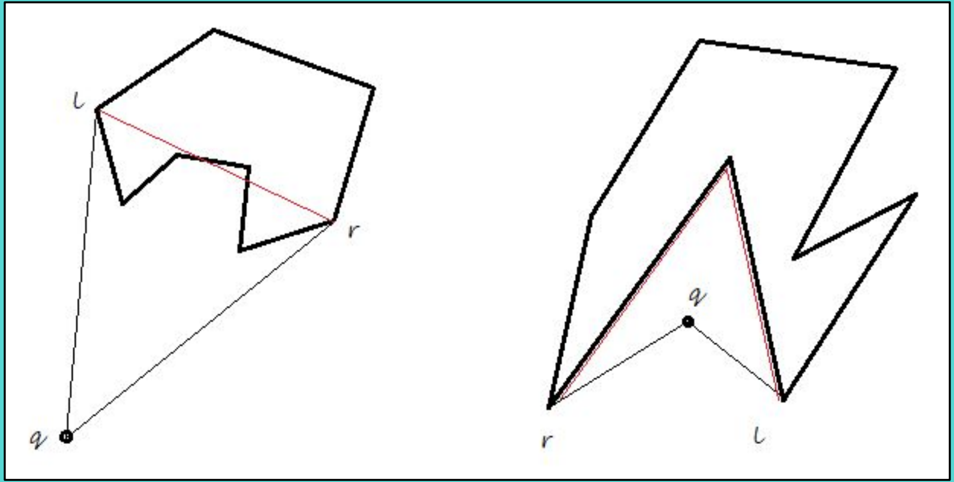


M. de Berg, O. Cheong, M. Kreveld, M. Overmars «Computational Geometry» // Springer, pp. 326-331. 2008.

Алгоритм решения

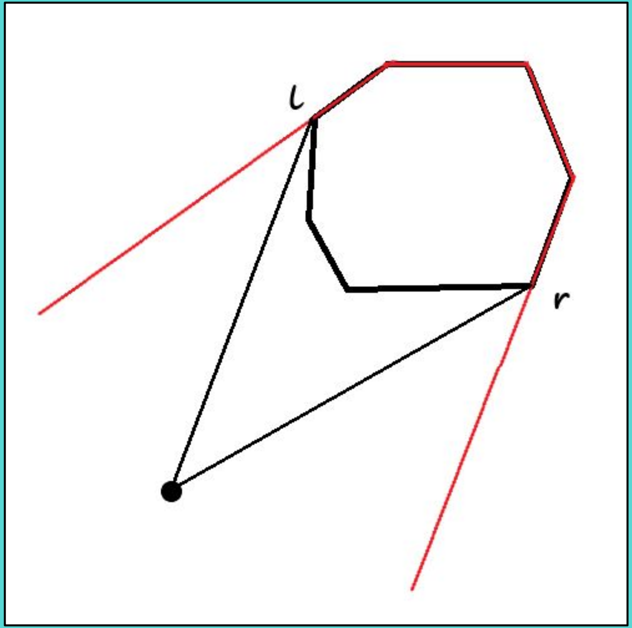
Рассмотрим точку запроса q и множество $\{P_i\}$. Имеем 2 подзадачи:

1) Найти 2 опорные точки l, r из q к каждому P_i
2) Построить граф видимости для полученных h отрезков



Опорные прямые к выпуклому многоугольнику за O(log n_i)

В выпуклом многоугольнике P_i полуплоскости, пересечением которых он образован, монотонны относительно того, содержат ли они внешнюю точку q. Это делит полуплоскости на 2 подмножества. Для организации двоичного поиска вершин, разделяющих подмножества (опорные вершины) необходимо найти внутреннюю точку каждого из них. Прямая, проходящая через q и любую внутреннюю точку P_i, пересекает ребра P_i именно в таких точках.



Алгоритм решения

Пусть $CH(P_i)$ - выпуклая оболочка P_i Есть 2 случая:

1) $q \notin CH(P_i)$ - тогда опорные прямые к P_i совпадают с опорными прямыми к $CH(P_i)$, которые могут быть найдены за $O(\log n_i)$
2) $q \in CH(P_i)$ - тогда опорные прямые могут быть найдены за $O(n^2)$

Предобработка: построение выпуклых оболочек для всех P_i. Алгоритм Чена[1]: $O(n_i \log h_i)$, где h_i - количество точек P_i, образующих замыкание $CH(P_i)$. Суммарная временная сложность: $O(\sum^{h}_{i=1} n_i \log h_i)$

Проверка принадлежности точки выпуклому многоугольнику $CH(P_i)$: $O(\log n_i)$ [2]

1. Timothy M. Chan. "Optimal output-sensitive convex hull algorithms in two and three dimensions". Discrete and Computational Geometry, Vol. 16, pp.361–368. 1996.

2. Препарата Ф., Шеймос М. Раздел 2.2: Задачи локализации точек. // Вычислительная геометрия: введение. — Москва: Мир, 1989.

Построение графа видимости для полученных h отрезков

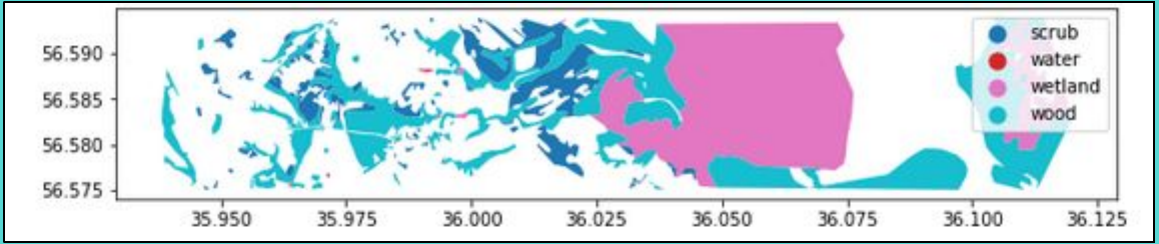
1) Алгоритм разделения окружности на num_angle углов и запоминания фиксированного количества расстояний до отрезка, каждое из которых соответствует определенному углу наклона. Точность построения - π/num_angle. Погрешность ≤ 5° незаметна для человека. Данный параметр также используется для упрощения получаемого графа. Временная сложность - O(h), однако константа является большой.
2) Алгоритм вращающейся заметающей прямой - O(h log h). На практике предпочтителен, так как работает быстрее.

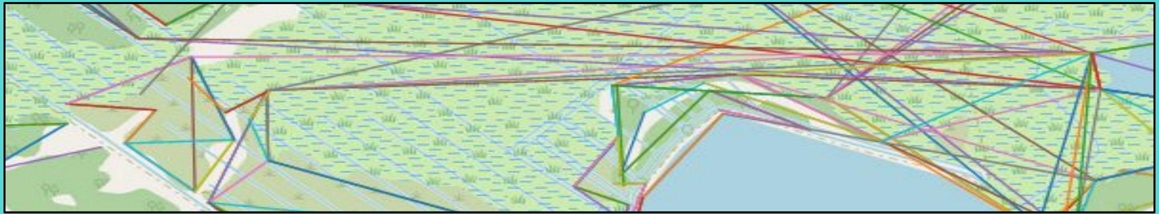
Итоговая сложность алгоритма

1) Нахождение 2h опорных точек: $O(\sum^{h}_{i=1} (\log n_i + f(n_i)))$, где $f(n_i) = \log(n_i)$, если $q \notin CH(P_i)$, $f(n_i) = n_i^2$, если $q \in CH(P_i)$. Из того, что $\sum^{h}_{i=1} (\log n_i) < h * \log(\sum^{h}_{i=1} n_i) = \log n$ следует, что минимальная сложность алгоритма - $O(h \log n)$, максимальная - $O(n^2)$. Память линейна. Однако не существует конфигурации, при которой $\forall q, \forall P_i \rightarrow q \in CH(P_i)$. На практике количество таких точек $q << n$. Сложность $O(h \log n)$ достигается.
2) Построение графа видимости для отрезков - $O(h \log h)$

Итоговая сложность: $O(h \log n)$. Граф является оптимизированным.

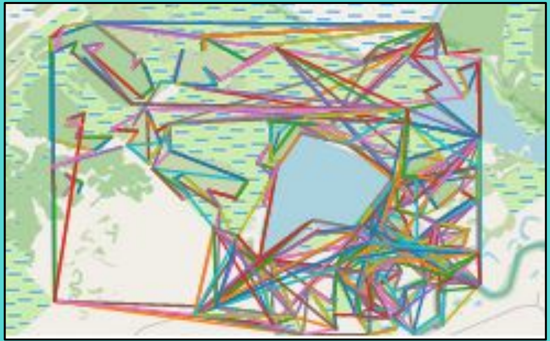
Примеры работы

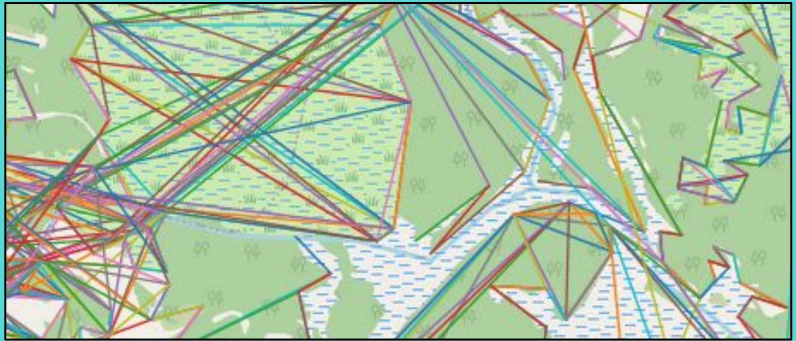


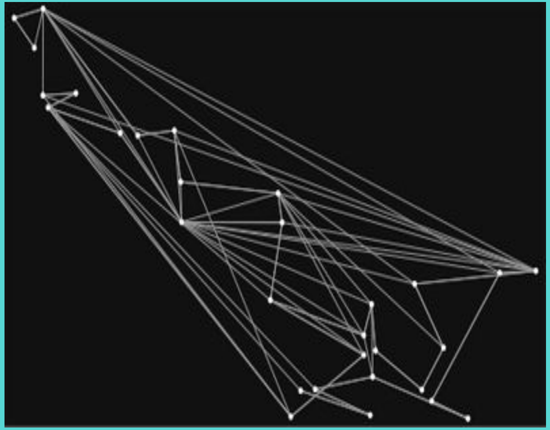


www.openstreetmap.org

Примеры работы







Спасибо за внимание!