Науково-практичний звіт на тему

НАЙКОРОТШІ ШЛЯХИ НА МНОЖИНІ ПЕРЕШКОД У 2D

В. В. Точаненко, студент 3 курсу, групи ІПС-31

Анотація. У роботі запропоновано метод знаходження найкоротшого шляху між двома точками на двовимірній множині із урахуванням перешкод. Ці перешкоди є полігонами. Для знаходження шляху використовується алгоритм Дейкстри (пошук на графі видимості). Апроксимація ламаної відбувається за допомогою побудови сплайну Ерміта.

1 Вступ

Постановка проблеми. Робота розв'язує проблему знаходження оптимального шляху на множині перешкод – полігонів. Задача цього алгоритму полягає в тому, щоб знайти найкоротший шлях між двома точками, що оминає усі дані перешкоди.

Ця задача застосовується в будівництві міст, у робототехніці для обчислення оптимального шляху, у відеоіграх для визначення найкоротшого шляху для пересування транспорту, у обчисленні оптимального шляху для прокладання електромереж, та інше.

Аналіз останніх досліджень. Ця задача розглядається у багатокутній області із Евклідовою метрикою. Для вирішення цієї задачі існують і інші алгоритми, але вони використовуються на інших метриках.

Задача побудови найкоротшого шляху між двома точками на множині перешкод у багатокутній області із Евклідовою метрикою полягає у побудові графу видимості. Алгоритм працює за час $O(E + n\log(n))$, пошуку найкоротшого шляху у графі за допомогою алгоритму Дейкстри за час $O(E + n\log(n))$ і апроксимації отриманого шляху побудовою сплайну Ерміта. Для побудови графу видимості будемо перебирати всі вершини графу, перевіряючи чи перетинає поточний відрізок якесь ребро будь-якої перешкоди. Час роботи такого алгоритму $O(n^3)$. Отримати кращу складність роботи алгоритму досить важко, адже у найгіршому випадку нам все одно доведеться розглянути усі можливі відрізки і сторони полігонів-перешкод. Є декілька більш швидких методів. Метод Мітчела має складність $O(n^{\frac{3}{2}} + \varepsilon)$, метод Хешберга і Сурі має складність $O(n\,log(n))$, метод Лі - $O(n^2\log(n))$.

Метод графа видимості для пошуку найкоротшого шляху на Евклідовій площині був запропонований Нільсом Нільсоном у 1969 році.

Коли граф видимості побудований, для розв'язання задачі пошуку найкоротшого шляху можна використовувати декілька підходів:

1. Алгоритм Дейкстри. Він був запропонований у 1959 році Едсгером Дейкстрою для знаходження найкоротших шляхів у зваженому графі. Складність такого алгоритму $O(n^2)$

2. Алгоритм покрокового переглядання усіх шляхів і початкової до кінцевої вершин. Він зупинить свою роботу коли будуть пройдені усі можливі шляхи і з них обереться найменший. Складність такого алгоритму завелика.
3. Алгоритм Лі. Заснований на пошуку в ширину. Він працює на графах із ребрами, що мають одиничну довжину, тому ребра повинні мати цілу довжину більшу за 0. Ребра з довжинами більшими за 1 можна розбити на одиничні ребра. Такий алгоритм має асимптотику $O(n^2)$.

Останнім кроком буде апроксимація ламаної. Існує багато методів побудови апроксимації ламаної:

1. Інтерполяція Лагранжа та Ерміта. Поліном буде мати вигляд $y = \sum_{i=1}^{n} a_i x^i$, де a_i – коефіцієнт поліному, а n – порядок поліному.
2. Параметричний запис кривої у формі Фергюсона: $x = x(u);\ y = y(u);\ z = z(u); u \in I$
3. Апроксимація ламаної методом Безьє. Поліном буде мати вигляд $\sum_{i=0}^{n} \overline{P}_i J_{ni}(t)$, де $\overline{P}_i$ – радіус-вектор точок на ламаній, а $J_{ni}(t)$ – апроксимуючі многочлени Бернштейна.

Кожен із цих методів має свою специфіку і майже однакові за часом роботи у $O(n)$. Виберемо будь-який з них. Нехай обраним методом апроксимації ламаної буде сплайн Ерміта, що використовує інтерполяційні поліноми Ерміта.

Новизна та ідея. В розглядуваній роботі запропоновано підхід вирішення задачі знаходження найкоротшого шляху між двома точками на множині перешкод у багатокутній області із Евклідовою метрикою таким чином: побудова графу видимості, знаходження алгоритмом Дейкстри найкоротший шлях між двома точками, а потім апроксимація отриманої ламаної сплайном Ерміта.

Мета статті. Розробити програмний додаток для знаходження найкоротшого шляху між двома точками на множині перешкод у багатокутній області із Евклідовою метрикою.

2 Основна частина

Постановка задачі. Нехай задані початкові точки A та B, деяка множина полігонів H, де H_i – i-тий поліном, що складається із k_i точок. Необхідно побудувати сплайн Ерміта на основі найкоротшого шляху від точки A до точки B, оминаючи усі полігони.

Означення 1. *Три точки a, b, c будемо називати розташованими за годинниковою стрілкою **(поворот праворуч)**, якщо для кожної послідовної пари крапок $[a, b], [b, c], [c, b]$ третя точка розташована справа від прямої, що проходить через перші дві точки. Три точки будуть розташовані проти годинникової стрілки **(поворот ліворуч)** у іншому випадку. Якщо вони розташовані на одній прямій, то такі точки є колінеарними.*

Означення 2. *Нехай маємо два відрізки, кожен з котрих задається двома точками: $[p_1 q_1]$для першого відрізку і $[p_2 q_2]$ для другого. Дані два відрізки пересікаються тоді і тільки тоді, коли виконуються такі два вирази:*

- *Трійки крапок $[p_1 q_1 p_2]$ та $[p_1 q_1 q_2]$ мають різні повороти*
- *Трійки крапок $[p_2 q_2 p_1]$ та $[p_2 q_2 q_1]$ мають різні повороти*

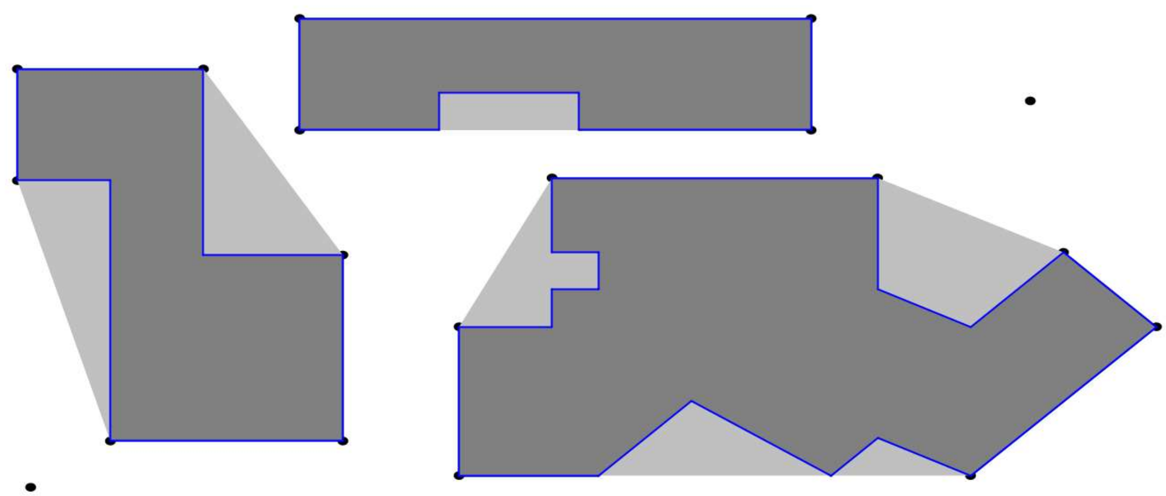
2.1 Побудова опуклих оболонок

Перш ніж будувати граф видимості, необхідно побудувати опуклу оболонку для кожного полігону. Алгоритмів для вирішення такої задачі досить велика кількість, тому можна обрати будь-який. Виберемо алгоритм Грехема для побудови опуклої оболонки скінченної множини точок. Час роботи такого алгоритму $O(n \log(n))$.

Перший крок в алгоритмі – знайти точку з найменшою y-координатою. Якщо таких декілька, то обираємо серед них точку з найменшою x-координатою. Додаймо цю точку в стек.

Далі, точки мають бути відсортовані в порядку зростання кута, який вони разом з P утворюють з віссю x. Для кожної точки визначаємо чи було пересування від двох попередніх точок до цієї точки поворотом ліворуч чи поворотом праворуч. Якщо це був поворот праворуч, тоді передостання точка (від якої повертали) не є частиною опуклої оболонки і має бути видалена зі стека. Цей процес триває доти, доки останні три точки утворюють поворот праворуч. Як тільки поворот ліворуч був отриманий, алгоритм рухається до наступної точки у відсортованому масиві.

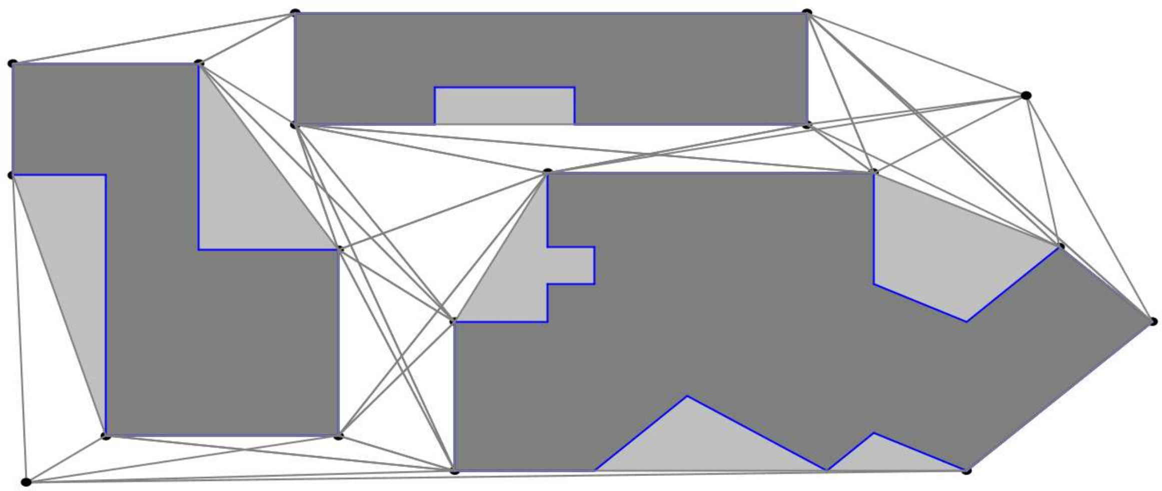
Алгоритм закінчить роботу тоді, коли наступною точкою для перегляду буде початкова точка.



Малюнок 1. Світло-синіми полігонами позначені опуклі оболонки кожного полігону. Синіми відрізками позначено грані полігонів

2.2 Алгоритм побудови графу видимості

Обираємо як початкову точку дану точку A. Проводимо відрізок, що починається у цій точці, та закінчується у першій отриманій вершині будь-якого полігону. Перевіряємо чи не перетинається цей відрізок із кожною гранню кожного полігону. Якщо перетинань не було виявлено, записуємо цей відрізок як ребро графу у зважений граф G разом із його вагою m. Таку процедуру проводимо із кожним відрізком, що починається у крапці A та закінчується у поточній вершині поточного полігону. Далі за початок відрізку беремо наступну точку і розглядаємо із усіма іншими, окрім тих, що уже були розглянуті. Таким чином отримаємо граф видимості без дублікатів ребер.



Малюнок 2. Сірими відрізками позначено граф видимості

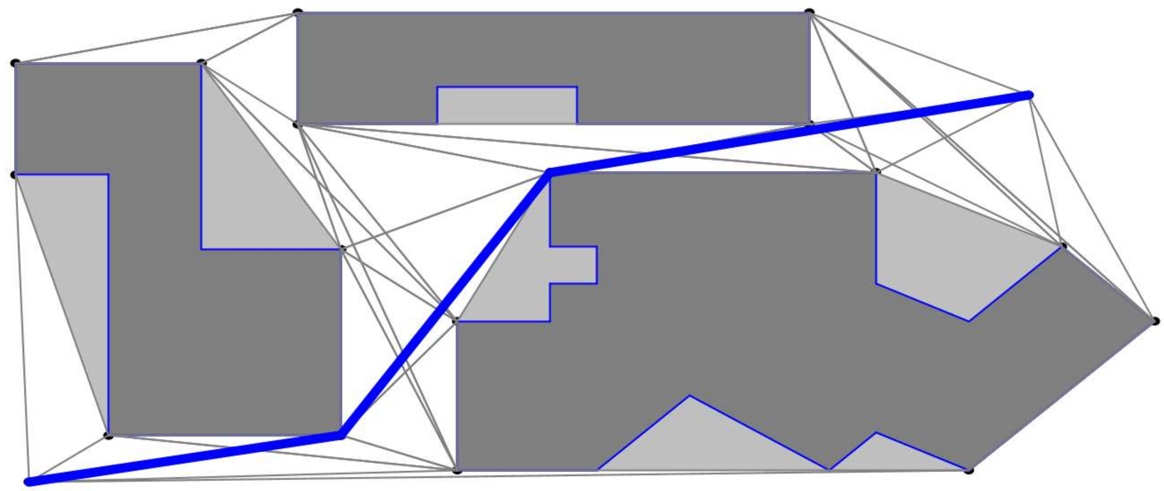
2.3 Знаходження найкоротшого шляху

Для цієї підзадачі будемо користуватись алгоритмом Дейкстри. Він має час роботи $O(V^2)$, де V – кількість ребер у графі видимості.

Спочатку маємо масив відстаней та масив позначок. На початку алгоритму відстані заповнюються великим додатнім числом (більшим максимального можливого шляху в графі), а масив позначок заповнюється нулями. Потім відстань для початкової вершини вважається рівною нулю і запускається основний цикл.

На кожному кроці циклу ми шукаємо вершину з мінімальною відстанню і прапором рівним нулю. Потім ми встановлюємо в ній позначку 1 і перевіряємо всі сусідні з нею вершини. Якщо в ній відстань більша, ніж сума відстані до поточної вершини і довжини ребра, то зменшуємо його. Цикл завершується коли позначки всіх вершин стають рівними 1.

Детальніше цей алгоритм вивчався у курсі «Алгоритми та складність» у четвертому семестрі ОС «Інженерія програмного забезпечення».



Малюнок 3. Найкоротший шлях на графі видимості

2.4 Апроксимація ламаної

Для підзадачі апроксимації ламаної будемо використовувати сплайн Ерміта.

Задано початкову точку p_0 з початковим вектором m_0 при $t = 0$ та кінцеву точку p_1 з кінцевим вектором m_1 при $t = 1$.

Для кубічного полінома та його похідної

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$p`(t) = a_1 + 2a_2 t + 3a_3 t^2$$

виразимо коефіцієнти a_0, a_1, a_2, a_3 через $p(0), p(1), p`(0), p`(1)$:

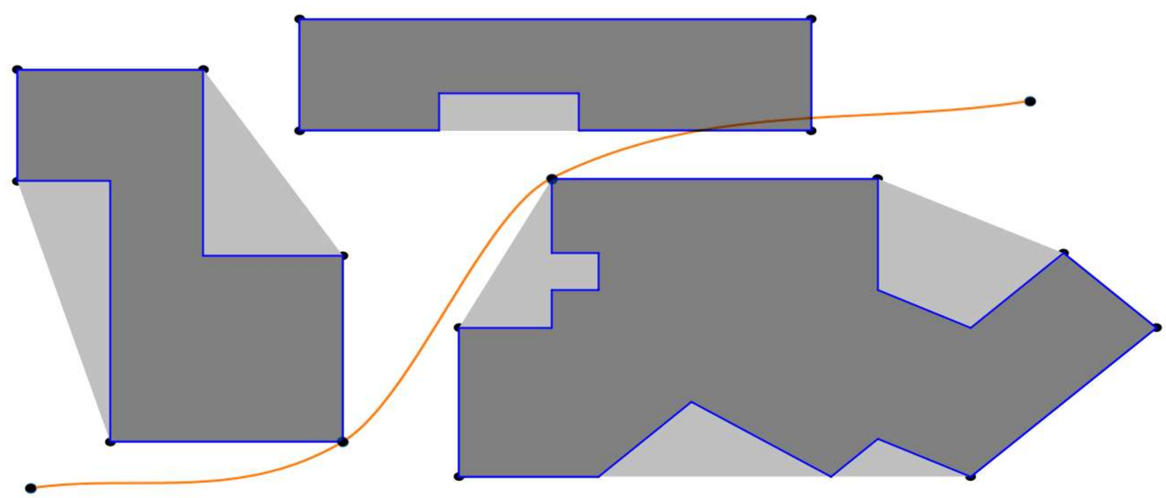
$$\begin{cases} p(0) = a_0 \\ p(1) = a_0 + a_1 + a_2 + a_3 \\ p`(0) = a_1 \\ p`(1) = a_1 + 2a_2 + 3a_3 \end{cases}$$
$$\begin{cases} a_0 = p(0) \\ a_1 = p`(0) \\ a_2 = 3\big(p(1) - p(0)\big) - 2p`(0) - p`(1) \\ a_3 = p`(0) = p`(1) - 2(p(1) - p(0)) \end{cases}$$

Підставивши значення полінома та його похідної із таблиці справа, отримаємо чотири базові ермітові поліноми:

$$\begin{cases} h_{00}(t) = 2t^2 - 3t^2 + 1 = (1-t)^2(1+2t) \\ h_{01}(t) = -2t^3 + 3t^2 = t^2(3-2t) \\ h_{10}(t) = t^3 - 2t^2 + t = t(1-t)^2 \\ h_{11}(t) = t^3 - t^2 = t\text{^}2(t-1) \end{cases}$$

Тоді інтерполяційний поліном визначається як лінійна комбінація чотирьох базових:

$$p(t) = h_{00}(t)p_0 + h_{10}(t)m_0 + h_{01}(t)p_1 + h_{11}(t)m_1, t \in [0,1]$$



Малюнок 4. Апроксимація найкоротшого шляху, що представлений ламаною

Висновок

У роботі запропоновано підхід вирішення задачі знаходження найкоротшого шляху між двома точками на множині перешкод у багатокутній області із Евклідовою метрикою таким чином: побудова графу видимості, знаходження алгоритмом Дейкстри найкоротший шлях між двома точками, а потім апроксимація отриманої ламаної сплайном Ерміта.

Джерела

1. Математическое обеспечение подсистем машинной графики и геометрического моделирования
2. Моделирование кривых в современных системах автоматизированного проектирования - Вестник Иркутского государственного технического университета
3. Аппроксимация методом Безье - Научное общество GraphiCon
4. Кубічні сплайни Ерміта – Wikipedia
5. Алгоритм Грехема - Wikipedia
6. Однорідні та неоднорідні B-Сплайни – Лекція з Математичних основ обчислювальної геометрії - Терещенко Василь Миколайович
7. Motion planning: граф видимости, дорожные карты – Дмитрий Елисеев