

УДК 004.932.2

**АВТОМАТИЗАЦІЯ ПОШУКУ НАЙКРАЩОГО ШЛЯХУ
НА КАРТОГРАФІЧНОМУ ЗОБРАЖЕННІ****Соколова Н.О., Волковський О.С.***Дніпропетровський національний університет ім. О.Гончара м. Дніпропетровськ**Кафедра автоматизованих систем обробки інформації**E-mail: n-olegovna@yandex.ru***Анотація**

Соколова Н.О., Волковський О.С. Автоматизація пошуку найкращого шляху на картографічному зображенні. В статті розглянута система прийняття рішень для транспортних цілей з описом алгоритмів пошуку та векторизації об'єктів доріг на картографічному зображенні та пошуку найкоротшого шляху між двома точками на цьому картографічному зображенні.

Загальна постановка проблеми. Останнім часом питанням обробки і зберігання геофізичної (і, зокрема, транспортної) інформації приділяється дуже велика увага, обумовлена значним підвищенням інтересу та попиту на різноманітні мобільні послуги. Система глобального позиціонування (GPS) та подібні до неї, надають різні сервіси використання дорожньої інформації мобільними пристроями. Кожен з них припускає наявність деякого векторизованого варіанта карти доріг, що знаходиться на сервері, програми-клієнта, та системи зв'язку між клієнтською частиною і сервером. Слабким місцем цих систем є залежність від двох факторів:

- Наявності зв'язку з сервером.
- Наявності потрібної карти на сервері.

Постановка задач дослідження. Поряд з цим в Інтернеті можна знайти чималу кількість географічних карт різноманітного змісту, які відрізняються як масштабом так й алгоритмами перетворення графічних даних в залежності від типу файлу.

Розробка програми автоматизації пошуку маршруту між двома довільними точками на довільній карті звичайного графічного формату, створивши векторну модель її доріг.

Поставлену задачу можна розбити на дві підзадачі:

1. Переведення карти у векторну форму – пошук на ній доріг і перетворення їх на сукупність зв'язаних між собою точок у просторі.
2. Пошук на створеній векторній формі карти оптимального маршруту між двома заданими користувачем точками.

Рішення задачі та результати досліджень. Винайдений алгоритм має дві особливо важливі риси – невимогливість до обчислювальних ресурсів комп'ютера та швидкий час обробки. Невимогливість означає, що алгоритм може бути реалізований під різними програмно-апаратними платформами, у тому числі й тими, в яких принципово неможливі великі потужності при обчисленні. Це, а також швидкість обробки, дозволяють використовувати його безпосередньо для оперативного обчислення у мобільних системах, не лише спеціалізованих навігаційних чи геоінформаційних, але й загального призначення, наприклад Java J2ME, Windows Mobile, та подібні. На вході алгоритму може бути будь-яке кольорове або монохромне зображення картографічного змісту, на виході – зв'язаний зважений граф із доріг, знайдених на вхідному зображенні. З точки зору мінімальності вихідних даних алгоритм не є оптимальним – це зворотній бік успішної реалізації невимогливість до процесорної потужності, і швидкодії. Ця збитковість вихідних даних гратиме одну з ключових ролей під час пошуку оптимального шляху в них.

Під перетворенням кольорового (або в градаціях сірого) зображення на бінарну карту будемо розуміти заміну на значення “1” тих його частин, які задовольняють заданим умовам, і на значення “0” - тих частин, які не задовольняють. У нашому випадку умовою відбору точки зображення буде її колірність – належність кольору поточної точки до одного з заданих користувачем кольорів. Оскільки програма не розуміє, які об'єкти, зображені на карті, є дорогами, а застосування спеціальних алгоритмів їх розпізнавання негативно вплине на швидкодію та процесороемісність програми, то перед початком роботи алгоритму користувачу буде запропоновано обрати на карті всі можливі варіанти доріг, для здобуття інформації про кольори, якими вони на поточному зображенні позначаються.

Крім того, до масиву кольорів доріг вже занесено чорний колір – стандартний колір написів-назв доріг. Алгоритм спроектовано таким чином, що при “зустрічі” його з точкою чорного кольору він окрім неї самої позначає одиницею також всі точки навколо себе в радіусі двох точок. Це дає змогу уникнути проблем, пов'язаних з неточністю обробки ділянок з написами на дорогах. Результат роботи даного етапу алгоритму – двовимірний вектор, бітова маска доріг для поточного зображення.

Наступний етап алгоритму полягає у заповненні на основі створеної бітової карти масиву вершин, який буде являти собою комп'ютерну модель графу. Звісно, під час створення будуть формуватися зв'язки між вузлами. Заповнення відбувається шляхом аналізу кожної точки отриманої бітової маски. Якщо значення точки “1”, на основі її місцезнаходження формується об'єкт-”вузол”, який заноситься до масиву вузлів – графу. Також перевіряється “околиця” поточної точки – частина бітового поля радіусом 2 точки. На всі точки-вузли, що знайдені в цій околиці, в поточний вузол додаються зв'язки до них.

Таким чином формується граф, що має в якості вузлів кожну точку “1” бітового поля. Така деталізація є надмірною для графу, тому наступним етапом стане зменшення кількості його вузлів.

Після виконання цього етапу кількість вузлів графу зменшується в кілька разів. Це досягається за рахунок видалення окремих його вузлів, якщо відстань від кожного з них до його сусідніх вузлів не перевищує заданої величини. Цей етап працює за наступним алгоритмом:

1. Визначаємо поточний вузол
2. Знаходимо вузол, який з'єднаний з поточним вузлом
3. Проводимо всі вузли, приєднані до цього вузла. Якщо відстань до кожного з цих вузлів не перевищує допустимого значення, то кожен з цих вузлів прив'язується до поточного (у двосторонньому порядку), а зв'язки до приєданого вузла та сам приєднаний вузол видаляємо. Інакше вузол помічаємо як невидимий.
4. Шукаємо наступний приєднаний до поточного вузол. Якщо такого нема, шукаємо наступний поточний вузол графу.
5. Якщо непереглянутих вузлів у графі немає, процедура з 1 по 4 пункт повторюється згідно з встановленою кількістю ітерацій.

Результатом цього етапу буде граф, кожен вузол якого відстоїть від суміжних йому вузлів на однакову відстань. Також граф зменшиться по кількості вузлів, достатньо для роботи алгоритму пошуку оптимального шляху. Ще на цьому етапі можна провести навантаження ребер графа його довжинами, залежно від того, потребує цього задача чи ні.

Сімейство евристичних алгоритмів успішно застосовується в комп'ютерних іграх. Кожен алгоритм у своїй основі має видозмінений хвильовий алгоритм, але тут метою є або визначення відстані між двома об'єктами, або пошук шляху між ними (не найкоротший, а просто його наявність, є він, чи нема). Крім того, алгоритми розроблялись з наголосом на обминання перешкод, але вони можуть бути пристосовані для розв'язку задачі пошуку оптимального шляху.

Основна проблема, з якої стикаються алгоритми розв'язку – пошук шляху в умовах перешкод. В теорії графів і звичайного штучного інтелекту є кілька алгоритмів, які можуть вирішити проблему й складних перешкод, й зважених областей. Застосування цих алгоритмів до пошуку шляху в геометричному просторі вимагає простої адаптації: стан або вузол графів представляють об'єкт, що перебуває в певній клітці, і пересування в сусідні клітки відповідає переміщенню в сусідні стани або суміжні вузли.

Задачу пошуку на створеній векторній формі карти оптимального маршруту між двома заданими користувачем точками розв'язували за допомогою класичного алгоритму Дейкстри для проходу по графах, грані яких мають різну вагу. На кожному кроці, шукаємо неопрацьовані вузли близькі до стартового, потім переглядаємо сусідів знайденого вузла, і встановлюємо або оновлюємо відповідні відстані від старту. Цей алгоритм має переваги в порівнянні з пошуком завширшки: він бере до уваги вартість або довжину шляху й оновлює вузли, якщо до них знайдений кращий шлях. Для реалізації цього алгоритму список із чергою FIFO замінюється пріоритетною чергою, де витягнутий вузол має краще значення. В нашому випадку це найменша вартість шляху від старту.

Оскільки умовою даної задачі є застосування алгоритмів пошуку оптимального шляху до зв'язаного зваженого графу, то «чиста» реалізація алгоритму Дейкстри буде недоцільною. Втім, концепція цього алгоритму дозволяє його пристосування до будь-яких просторових форм представлення даних. У даному випадку було обрано алгоритм Дейкстри, тому що він має властивість поширюватись однаково в усі боки незалежно від знаходження кінцевої точки відносно поточної генерації хвилі. Також, через те, що вхідний граф має ребра, рівні за довжиною, відпадає потреба в їх навантаженні, оскільки за даних умов це можна зробити лише довжинами ребер. Логіка роботи алгоритму залишається при цьому незмінною: при знаходженні першою точкою згенерованої хвилі кінцевої точки найкоротший шлях вважається знайденим.

Висновки

Була розроблена програма для пошуку найкоротшого шляху між двома точками на карті, що являє собою графічний файл формату PNG. Програму створено у середовищі Eclipse 3.2, мовою Java, вона може працювати на будь-якій програмно-апаратній платформі, яка має підтримку Java Runtime Environment (JRE) версії 1.5. Програма містить реалізації двох алгоритмів: власного алгоритму пошуку та векторизації доріг на картографічному зображенні, та алгоритму Дейкстри для пошуку оптимального шляху на створеній моделі доріг. Результатом виконання є зображення шляху на вхідній карті поміткою тих доріг, через котрі проходить найкоротший маршрут.

Програму було відтестовано на платформі Windows (OS Windows XP/Windows Vista, комп'ютер PC) зі встановленим середовищем JRE 1.5. У якості вхідних картографічних зображень використовувалися скріншоти навігаційних програм-планів міст Візіком-Київ та Візіком-Дніпропетровськ.

Створену систему можна використовувати як за прямим призначенням (вирахування найкоротшого шляху між двома пунктами, використовуючи графічне зображення карти), так і розширити її можливості за рахунок перенесення її на інші стаціонарні та мобільні програмно-апаратні платформи.

Список літератури

1. Е.В. Шикин, А.В. Боресков Компьютерная графика. “Мир”, Москва, 1995.
2. Бучнев А.А., Калантаев П.А., Ким П.А., Пяткин В.П. Перспективные информационные технологии в исследовании Земли из космоса. <http://tbs.dcn-asu.ru/ipl/>
3. Б. Яне Цифровая обработка изображений “Техносфера“, 2007
4. В.Т. Фисенко В. Т. , Т.Ю. Фисенко Компьютерная обработка и распознавание изображений: учебное пособие. СПбГУ ИТМО, СП-б. 2008