УДК 004.651.4 (045)

В.А. Бородін, к.т.н., доц.

ГНУЧКИЙ МЕТОД ПОШУКУ В ІНТЕРАКТИВНИХ БАЗАХ ДАНИХ

Запропоновано швидкий та гнучкий метод регіонального пошуку для інтерактивних баз даних, Цей метод пропонується використовувати в авіаційних комплексах інтерактивної взаємодії.

In the paper the quick and flexible range searching method for interactive data bases creation is proposed. This method is proposed for using in interactive aviation complexes.

Вступ

У багатьох прикладних інтерактивних комплексах, що є, наприклад, в комплексах, які спостерігають за літаками в аеропортах, а також для забезпечення найбільш швидкого доступу при оптимізації запитів до баз даних доводиться вирішувати задачі, що використовують апарат пошуку в геометричній області, зокрема регіонального пошуку [1; 2].

Формалізація задачі регіонального пошуку [1–3] може бути подана так.

Нехай $\Re$ — система підмножин d-вимірного Евклідового простору, які є областями.

При цьому задана множина P точок у R^d. Потужність цієї множини позначимо через n. Кожній точці відповідає деяка вага $w_i \in R$.

Базову задачу пошуку в геометричній області поставимо так.

Кожній точці надана деяка вага. Розробити ефективний алгоритм, який за заданою областю $G \in \Re$ визначає суму ваги всіх точок з P, що лежать у G.

Вибравши відповідні ваги точок, можна вирішити задачі знаходження кількості точок з P, що лежать в G, а також виведення списку точок, які належать цій області.

Аналіз досліджень і публікацій

На практиці, під точками розуміється набір координат, який характеризує об'єкти, над якими здійснюється пошук (літаки, записи в базі даних).

Під простором, в якому проводять пошук, мають на увазі двовимірну (або d-вимірну) проекцію простору системи з можливими додатковими потрібними координатами.

Основна мета методів геометричного пошуку – знаходження такої структури вхідних даних про об'єкти, щоб для будь-якої заданої області з нашого простору час знаходження відповідної ваги точок, що знаходяться в цій області був оптимальним.

Під оптимальністю розуміють найменший час пошуку для будь-яких вибраних областей і координат точок [3–6].

Серед задач пошуку на площині слід виділити такий важливий окремий випадок, як пошук в області, що має прямокутну форму, зі сторонами, паралельними осям координат. Ця задача зветься також регіональним пошуком [4;7].

Саме до такого формулювання задачі зводяться задачі пошуку в базах даних.

Метою цієї роботи є побудова алгоритму, який вирішує задачу регіонального пошуку на площині досить швидко, не використовуючи додаткові ресурси пам'яті. Під пам'яттю розуміють довжину побудованої структури даних.

Серед відомих методів пошуку, які вирішують цю задачу найбільш оптимально, є метод побудови дерев регіонів Бентлі [4]. Цей метод ґрунтується на побудові дерева пошуку, що сортує дані за кожною з координат у спеціальне дерево відрізків.

Для пошуку у d-вимірному посторі ця структура витрачає для n точок час $O(\log^d n)$ операцій – (час виконання цієї дії), використовуючи при цьому $O(n \log^{d-1} n)$ пам'яті та витрачаючи $O(n \log^{d-1} n)$ операцій на попередню обробку. Під $\log$ розуміють двійковий логарифм.

Цей метод доволі зручний у програмуванні, та він має певні недоліки крім того, що невідомо, чи він оптимальний [5].

Зокрема, програмна реалізація цього методу доволі складна, перебудова її структури в разі зміни даних про об'єкти пошуку потребує суттєвої кількості додаткових операцій. Водночас жорстка структура дерева, яке використовується для пошуку, може бути не дуже ефективною у випадку, коли потрібно скоротити кількість витраченої пам'яті з незначним збільшенням часу пошуку.

Така проблема особливо актуальна для використання цих методів для створення бази даних, які працюють в інтерактивному режимі, або в режимі реального часу.

Мета роботи − розроблення такої структури дерева пошуку, яка дозволяє керувати розміром структури даних у разі незначної зміни швидкості пошуку.

Побудова структури даних для швидкого пошуку

Структуру даних для швидкого пошуку можна створити так. Зафіксуємо заздалегідь число $k \in N$, $k \geq 4$. Розіб'ємо множину P точок на k майже рівних частин, які можна назвати стовпчиками, (тобто різниця потужностей будь-якої пари цих множин не перевищує одиниці) таким чином, щоб в першій із цих множин містились $\lceil n/k \rceil$ точок з найменшими x-координатами, потім виділимо точки серед тих, що залишились, з найменшими x-координатами і так далі.

Скобками $]x[$ тут і надалі позначатимемо найбільше ціле число, що більше за x.

Для кожної з утворених множин точок знову проведемо поділ на майже рівні множини, які назвемо рядками, де виділимо так само точки з найменшими, але вже y-координатами.

Отже, маємо k^2 майже рівних частин.

Для кожної з цих частин запам'ятуємо дерево бінарного пошуку за x-координатою та дерево пошуку за y-координатою.

Далі для кожної з утворених множин точок аналогічно проведемо ділення на k^2 майже рівних частин, для яких запам'ятовуємо дерева пошуку за кожною з координат та знову проведемо подальший поділ на k^2 майже рівних частин, доки кількість точок у кожній частині стане менше двох.

Глибина рівнів розбиття для цієї структури буде, очевидно:

$$]\log_{k^2} n[= \left\rceil \frac{\log n}{2 \log k} \right\lceil .$$

Таким чином, довжина структури даних буде $n]\log n / \log k[$.

Кількість операцій для побудови цієї струкури $n \log n]\log_k n[$, оскільки на кожному рівні потрібно $2n \log n$ операцій для побудови бінарного дерева пошуку.

Пошук відбувається так. Для двовимірної задачі регіонального пошуку область задають двома точками, які є заданими координатами (a,b), (c,d). Отже, прямокутник має координати вершин (a,b), (a,d), (c,b), (c,d).

На першому рівні розбиття ці точки потрапляють не більше ніж у чотири множини розбиття.

Знайти ці множини можна так. Оскільки в створеній структурі є впорядковане бінарне дерево пошуку за x-координатою, то можна за $2]\log k[$ операцій визначити, в які підмножини з першого розбиття на k підмножин (в яких двох чи одному стовпчиках) потряпляють ці точки.

Далі потрібно визначити, в яких рядках містяться вершини прямокутника пошуку, а також де проходять його вертикальні та горизонтальні сторони.

Для того, щоб знайти вершини прямокутника, потрібно здійснити пошук серед точок, які належать знайденим рядкам. Для цього достатньо здійснити пошук серед найбільших y-координат у кожному з рядків цих стовпчиків. Оскільки таких координат k, то всі вершини можна знайти за час $4]\log k[$.

Знаючи множини точок (рядки), до яких належать ці вершини, автоматично можна визначити не більше $k-2$ рядків, через які проходять вертикальні сторони прямокутника пошуку у двох або одному стовпчику, де містяться вершини прямокутника.

Для того, щоб визначити рядки, через які проходять горизонтальні сторони прямокутника пошуку, потрібно для кожного з не більш ніж $k-2$ стовпчиків, що лежать між тими, де містяться вертикальні сторони, провести пошук для найбільших y-координат у кожному з рядків цих стовпчиків, щоб визначити, які з цих рядків перетинаються з прямими $y = c$, $y = d$.

Для цього потрібно виконати $2(k-2)]\log k[$ операцій, оскільки бінарний пошук проводиться для k чисел, які можна легко виділити зі структури бінарних дерев за кожною множиною. Аналогічно будують структуру пошуку й для багатовимірного простору.

Оцінювання швидкості пошуку

Подальший пошук проводять так. Знаходять множини (рядки), що цілком розміщуються у прямокутнику пошуку. Оскільки під час створення структури даних було створено бінарне дерево пошуку, то вже відома сума ваг точок для цих рядків і можна її просто додати до шуканої суми ваг.

Для рядків, що не містять вершин прямокутника пошуку та які мають перетин з вертикальними сторонами прямокутника пошуку, потрібно

виконати пошук у бінарному дереві за x-координатою з підрахунком та додаванням потрібної суми до підсумкової аналогічно для тих рядків, які містять вершини прямокутника пошуку та мають перетин з горизонтальними сторонами прямокутника пошуку, виконують пошук за y-координатою.

Отже, маємо ще $4(k-2)]\log(n/k^2)[$ операцій та не більше $(k-2)^2$ операції додавання до підсумкової суми.

Для тих рядків, що містять вершини прямокутника пошуку, потрібно перейти на наступний рівень структури даних та виконати алгоритм, ідентичний попередньому рівню.

Так продовжується доти доки алгоритм не переходить на фінальний рівень (одна або дві точки), для якого достатньо провести чотири операції порівняння для визначення існування точок в області пошуку.

Звідси отримуємо, що для виконання пошуку потрібно операцій не більше

$$2k\log n + 2(k-2)\log^2 n/\log k .$$

Отже, побудований метод для обирання малого k близький до параметрів методу пошуку Бентлі.

Зокрема, при $k=4$ кількість операцій для пошуку дорівнює $8\log n + 2\log^2 n$ та довжина структури даних становить $n]\log n[/2$.

У випадку $k=\sqrt{n}$ маємо, що довжина структури даних дорівнюватиме $2n$, тобто буде строго лінійною від кількості даних, а кількість операцій під час пошуку буде не більше $4\sqrt{n}\log n - 8\log n$.

Цей метод зручний у випадку, коли потрібно зменшити розмір структури даних, наприклад, для зменшення використання ресурсів оперативної пам'яті комплекса.

Висновки

Запропонований метод створення бази динамічних даних для швидкого пошуку інформації важливий для комплексів реального часу та інтерактивних систем.

Розроблену схему побудови бази даних швидкого пошуку можна використовувати для створення баз динамічних даних в авіаційних комплексах інтерактивної взаємодії.

Література

1. *Agarval P.K., Erickson J.* Geometric range searching and its relevativs. Advances in Discrete and Computational Geometry. − American Mathematical Society Press, 1999. – P. 1–56 .

2. *Agarval P.K.* Range searching. – Computing Surveys, 1996. – №5. − P. 1–31.

3. *Overmars M.H.* The Design of Dynamic Data stractures. − Springer-Verlag, LNCS 156, 1983.

4. *Препарата Ф., Шеймос М.* Вычислительная геометрия: введение. – М.: Мир, 1989. – 478 с.

5. *Кнут Д.* Искусство программирования для ЭВМ. В 7 т. − Т.3. Сортировка и поиск. – М.: Мир, 1978. – 844 с.

6. *Васюхин М.И., Бородин В.А.* Методы реализации алгоритма поиска объектов в прямоугольной области при анализе воздушной обстановки // Математичні машини і системи. – 2001. – № 1–2. – С.100–105.

7. *Бородин В.А., Васюхин М.И., Креденцар С.М.* Алгоритм поиска в прямоугольной области для геоинформационных комплексов реального времени // Вестн. ХГТУ. – 2006. – № 1. – С. 50–60.

Стаття надійшла до редакції 05.11.08.