СПОСОБЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ О ДВИЖУЩИХСЯ ОБЪЕКТАХ В ЗАДАЧАХ АНАЛИЗА ВОЗДУШНОЙ ОБСТАНОВКИ

М.И.Васюхин
Институт Кибернетики имени В.М.Глушкова НАН Украины

Abstract

Wasuhin M. Ways of a data representation about mobile objects in problems of the analysis of air situation. *The algorithms of problem solving for the analysis of air situation are reviewed. The effective algorithms are offered. The method of the analysis presenting a variation of methods of binary looking up, arbors of splittings and cards of Veronese is offered. The efficiency of offered means and methods is rotined.*

Для быстрого решения задач поиска движущихся объектов в процессе анализа воздушной обстановки, часто приходится решать задачи, требующие применения поиска в геометрической области [1]. При этом реальные задачи допускают вариацию, как по типу подсчитываемых объектов, так и по размерам и форме области, в которой этот подсчет ведется. Главным условием остается простота задания такой области. Под простотой понимается, что область представляет собой геометрический примитив: полуплоскость, круг, параллелепипед, симплекс.При этом она может быть получена из них небольшим количеством применений оверлейных операций: объединения и пересечения однотипных примитивов.

Целью данной работы является разработка способов представления данных о движущихся объектах в задачах анализа воздушной обстановки. Здесь воздушные объекты представлены в виде точек, а под пространством, в котором производится поиск, подразумевается воздушное пространство или же его двухмерная проекция с возможными дополнительными координатами, указывающими на тип самолета (свой, чужой; грузовой, пасажирский и т.д.). Целью алгоритма является нахождение такой структуры входных данных об объектах, чтобы для любой задаваемой области из нашего пространства время нахождения данных об объектах из этой области было минимальным. Структуры данных для таких алгоритмов и рассматриваются в этой статье.

Напомним, что предложенная структура данных, основанная на идее построения графа без циклов, с весами в вершинах, равными суммарному весу листьев из этой вершины, называется в этой работе деревом разбиений.

Усовершенствованные способы разбиения при построении деревьев разбиений

Поскольку мы имеем дело с динамическими точечными объектами, которыми являются в нашем случае самолеты, классический алгоритм построения дерева разбиений [1] можно улучшить, используя следующий общий подход.

Сначала дискретное множество точек P заменяется непрерывным всюду положительным распределением массы μ. Каждая точка заменяется маленьким плотным шаром радиуса ε весом $1-\varepsilon$ и туманностью веса ε, простирающейся в бесконечность (рисунки 1 и 2). Очевидно, что при достаточно маленьком ε хорошая схема разбиения для μ подойдет и для множества P. Для обеспечения вывода и визуализации данных на экране коллективного пользования в действующей системе выгодно выбирать центр шара не в точке текущих координат движущегося самолета, а в точке, экстраполированной на следующий период обзора.

Таким образом, предложенная схема является универсальной в том смысле, что она без изменений может быть перенесена в произвольное пространство.

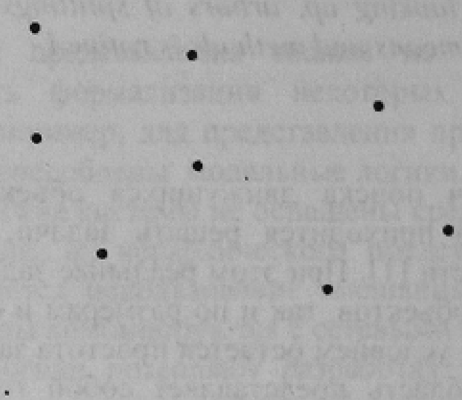

Рис. 1 - Точки на плоскости

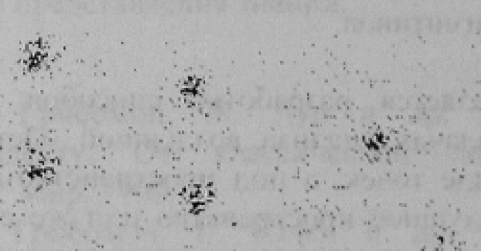

Рис. 2 - «Размазанные» точки

Построение схемы разбиения проведем индукцией по размерности пространства.

Для построения схемы разбиения нужно разбить рассматриваемое пространство на некоторое количество областей, с которыми мы соотносим вершины нашего дерева. По заданному μ строим центр масс $C(\mu)$ и множество $\Pi(\mu)$ (Рис. 3), которое состоит из

конечного числа кусков гиперплоскостей, удаление которого приведет к желаемым областям.
Для $d=1$ $C(\mu)$ — единственная точка, разделяющая μ, и $\Pi(\mu)=\{C(\mu)\}$. Пусть $d>1$, и пусть S —гиперплоскость, перпендикулярная x_d - оси и пересекающая μ. Для каждого вектора v с положительной x^d - координатой определены два $d-1$ - мерные распределения масс на S, обозначаемые μ_v^+ и μ_v^-. Распределение масс μ_v^+ образуется проекцией распределения массы в полупространства выше S на S в направлении $-v$.

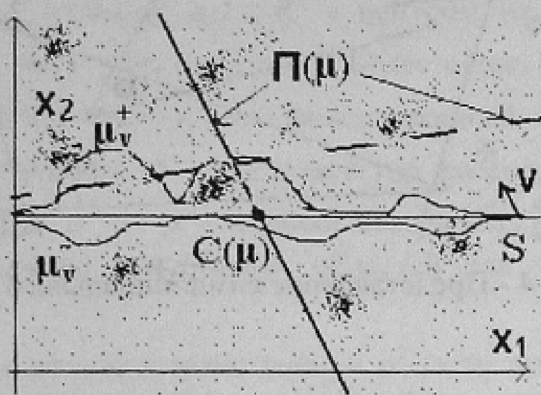

Рис. 3 - Представление усовершенствованного разбиения

Аналогично, μ_v^- — проекцией из полупространства ниже S на S в направлении v. Поскольку, гиперплоскость S можно отождествить с R^{d-1}, то точки $C(\mu_v^-)$ и $C(\mu_v^-)$ определены на S по индукции. Из общей топологии известно, что существует такое направление v, что точки $C(\mu_v^+)$ и $C(\mu_v^-)$ совпадают. Зафиксируем это направление.
Положим $C(\mu)=C(\mu_v^-)$ и
$\Pi(\mu)=S\cup\{x+tv \mid t>0, x\in\Pi(\mu_v^-)\}\cup\{x-tv \mid t>0, x\in\Pi(\mu_v^-)\}$.
Определенные точки $C(\mu)$ и гиперплоскости $\Pi(\mu)$- единственны. Более того, разбиение определяет 2^d областей, каждая из которых содержит 2^{-d} часть общей массы.
Таким образом, мы построили улучшенную схему разбиений.

Построение симплексного разбиения

В этом разделе конструкция структуры данных основана на применении симплексного разбиения. Симплексное разбиение для множества P — это совокупность пар $\Pi=\{(P_1,\Delta_1),\ldots,(P_m,\Delta_m)\}$. Множества P_i — это непересекающиеся подмножества P, образующие разбиение множества P, т.е. $P=\bigcup_{i=1}^{m}P_i$. Δ_i — это d - мерный симплекс (область, образующаяся пересечением $d+1$ плоскостью), содержащий множество P. Симплексы Δ_i, в отличие от множеств P_i, могут пересекаться. (см. рис.4).
Мы говорим, что гиперплоскость h рассекает симплекс Δ, если они пересекаются. Число пересечений симплексного разбиения Π по отношению к h определим, как число симплексов из Π, рассекаемых h, а число пересечений Π — как максимально возможное число пересечений Π относительно гиперплоскости.

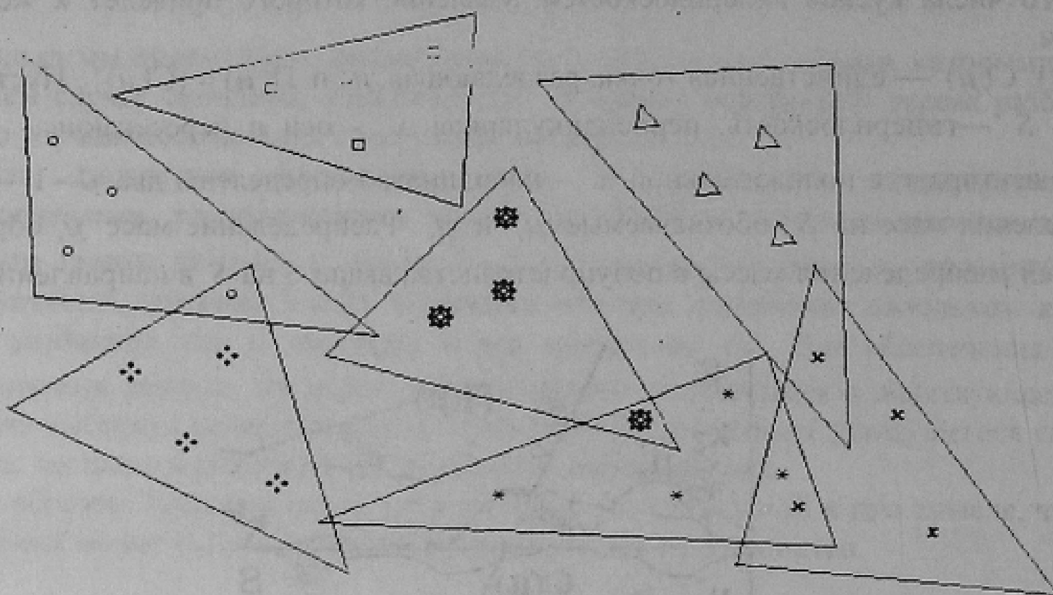

Рис. 4 - Представление симплициального разбиения

Первый шаг построения — это выбор подходящего конечного "тестового множества" H гиперплоскостей. Множество выбирается таким образом, что если число пересечений симплициального разбиения Π относительно произвольной гиперплоскости $h \in H$ ограничено некоторым числом k, то число пересечений Π относительно любой гиперплоскости есть $O(k + r^{1-1/d})$.

Потом пошагово строится симплексное разбиение Π, одна пара (P_i, Δ_i) за один раз. Предположим, что пары с (P_1, Δ_1) по (P_i, Δ_i) уже построены и $|P_j| = \lceil n/r \rceil$, $j = 1, \ldots, i$. Тогда множество еще не обработанных точек $\widetilde{P}_i = P \setminus (P_1 \cup \ldots \cup P_i)$ состоит из $n_i = n - i\lceil n/r \rceil$ точек. Для гиперплоскости $h \in H$ положим $k_i(h)$ — число симплексов среди $\Delta_1, \ldots, \Delta_i$, рассекаемых h, и назначим каждому $h \in H$ вес $\omega_i(h) = 2^{k_i(h)}$.

Здесь нам понадобится также понятие вырезки. Под вырезкой понимается конечное множество симплексов Ξ такое, что их внутренности не пересекаются, и все вместе они образуют покрытие всего пространства R^d. Вырезка Ξ называется $(1/r)$ - вырезкой для H, если нет симплекса, внутренность которого пересекается более чем n/r гиперплоскостями из H. Кроме того, если дана неотрицательная весовая функция $w: H \to R$, то можно определить $(1/r)$ - вырезку как такую, что сумма весов гиперплоскостей, пересекаемых симплексом, не превышает $1/r$ - части общего веса гиперплоскостей.

Пусть теперь Ξ_i — это $(1/t)$ - вырезка для H и весов ω_i, т.е. сумма весов гиперплоскостей, пересекаемых произвольным симплексом из Ξ_i, не превышает $(1/t)$ - часть общего веса всех гиперплоскостей. Параметр t выберем как можно больше, но так, чтобы общее число симплексов из Ξ_i не превышало $r(n_i/n)$. Тогда можно найти симплекс из Ξ_i, содержащий не более n/r точек из $\widetilde{P}_i$. Этот симплекс обозначим Δ_{i+1}.

Что касается множества P_{i+1}, то выберем его как некоторые $\lceil n/r \rceil$ точек из $\widetilde{P}_i$, которые содержатся в Δ_{i+1}.

После этого стандартным способом создается дерево разбиений.

Поиск в общих областях

В предыдущем разделе рассмотрен поиск в области, ограниченной гиперплоскостью. Однако часто требуется произвести запрос для более общей области с нелинейными, изогнутыми границами. Обычно такие области определяются конъюнкцией не более чем k полиномиальных неравенств максимальной степени D, где k и d — константы. Остановимся на областях, определенных одиночным полиномиальным неравенством. Области, описанные конъюнкцией нескольких неравенств, могут быть обработаны применением многоуровневой структуры данных, которая будет описана ниже.

На плоскости круговой диск $C = C(a_1, a_2, a_3) \subset R^2$ с центром (a_1, a_2) и радиусом a_3 описывается неравенством $(x_1 - a_1)^2 + (x_2 - a_2)^2 \le a_3^2$. Более обще, рассмотрим области вида $R_f(a) = \{x \in R^d : f(x, a) \ge 0\}$, где f — фиксированный полином от $d + p$ переменных $x_1, \dots, x_d, a_1, \dots, a_p$. Полином f определяет тип рассматриваемых областей (диски, конусы, цилиндры,...), и a — p - мерный вектор-параметр, определяющий область данного типа. Множество допустимых областей для такой проблемы — $\Re_f = \{R_f(a) : a \in R^p\}$.

Преобразуем области из $\Re_f$ в полупространственной области пространства большей размерности, нежели изначальное. Используем линеаризацию области, т.е. отображение φ, которое получается введением одной новой координаты t_i для каждого одночлена от переменных $x_1, \dots, x_d$, встречающихся в полиноме f. Линеаризация, возникающая таким образом, известна в алгебраической геометрии как карта Веронезе. Линеаризация позволяет свести проблему поиска в криволинейной области в плоскости с точечным множеством Р к проблеме полупространственного поиска в некотором пространстве более высокой размерности с множеством точек $\varphi(P)$.

Рассмотрим пример с дисками. Здесь

$$f(x_1, x_2, a_1, a_2, a_3) = a_3^2 - (x_1 - a_1)^2 - (x_2 - a_2)^2 = a_3^2 - a_1^2 - a_2^2 - 2a_1x_1 + 2a_2x_2 - x_1^2 - x_2^2$$

Отображение $\varphi : R^2 \to R^4$ определим равенством $\varphi(x_1, x_2) = (x_1, x_2, x_1^2, x_2^2)$. Тогда для каждого диска $C = C(a_1, a_2, a_3)$ существует полупространство $H = H(a_1, a_2, a_3)$ в R^4 такое, что $C = \varphi^{-1}(H)$ или, другими словами, точки плоскости, отображенные φ на полупространство H, являются точками диска С. Легко видеть, что подходящее полупространство — это $H = \{(t_1, t_2, t_3, t_4) \in R^4 : 2a_1t_1 + 2a_2t_2 - t_3 - t_4 + a_3^2 - a_1^2 - a_2^2 \ge 0\}$.

Однако, в случае с дисками, общий метод является немного расточительным в смысле размерности пространства отображения. Известна линеаризация дисков на плоскости в пространстве размерности 3. Поскольку каждый диск может быть получен как вертикальная проекция пересечения некоторого полупространства в R^3 с параболоидом $z = x^2 + y^2$ на плоскость $z = 0$, то лучшая линеаризация получается отображением плоскости на параболоид при помощи отображения $(x, y) \mapsto (x, y, x^2 + y^2)$. В общем случае линеаризация минимальной размерности для конкретного полинома f может быть эффективно найдена решением некоторой системы линейных уравнений.

Порядок сложности алгоритмов, созданных таким образом для проблем поиска с областями из $\Re_f$, зависит от d, размерности пространства, содержащего множество точек, в то время как пространственные требования для логарифмического времени запроса зависят главным образом от p. числа параметров. определяющих область. Для поиска в дисковой области на плоскости мы имеем $d = 2$. и время запроса для линейного пространства — приблизительно $n^{1-1/2} = \sqrt{n}$; число параметров — $p = 3$. поэтому память. близкая к $O(n^3)$. гарантирует логарифмическое время запроса.

Многоуровневые структуры данных и их использование в алгоритмах поиска в геометрической области

Определим на одном простом и важном примере многоуровневую структуру данных.
Пусть $S = \{s_1, ..., s_n\}$ — множество сегментов на плоскости. создать структуру данных. которая быстро вычисляет число сегментов S. пересеченных невертикальной линией запроса h.
Пусть a_i. b_i — концы сегмента s_i. A обозначает множество всех точек a_i. и B — множество всех точек b_i. Следует подсчитать число отрезков s_i. концы a_i которых находятся выше h. а концы b_i — ниже h. Противоположный случай решается симметрично.
Рассмотрим некоторое дерево разбиения для множества А. для определенности. пусть оно будет деревом разбиений. основанным на симплексном разбиении. При использовании такого дерева разбиений мы можем определить за $\sqrt{n}$ время число точек A. лежащих в полуплоскости R над h. Вес каждой точки из $A \cap R$ подсчитывается из нескольких посещенных узлов дерева. В каждом таком узле находим симплексы соответствующего симплексного разбиения. находящиеся полностью внутри R. и подсчитываем их веса.
Для каждого узла дерева разбиений будем называть классы симплексного разбиения. сохраненные в нем. каноническими множествами. Дерево разбиений обеспечивает разбиение множества $A \cap R$ приблизительно на $\sqrt{n}$ канонических множеств различных размеров. Общее число канонических множеств в дереве разбиений — $O(n)$. и сумма их размеров легко оценивается $O(n \log n)$.
Чтобы решить проблему с сегментами. увеличим дерево разбиений множества A следующим образом. Для каждого канонического множества $M \subset A$ создадим дерево разбиений для множества $M' = \{b_i : a_i \in M\}$. и сохраним его в соответствующем узле дерева разбиений для A. (Дерево разбиений для A названо первичным или первого уровня: деревья для множеств M' названы вторичными или второго уровня.)
Как применить полученную структуру данных к проблеме подсчета сегментов? Сначала выразим множество $A \cap R$ как непересекающееся объединение некоторых канонических подмножеств $M_1, ..., M_m$. и затем для каждого такого M_i используем соответствующее вторичное дерево разбиений для подсчета точек из M_i'. находящихся ниже линии h. Слагая вместе все эти числа по всем каноническим множествам M. мы получаем требуемое число сегментов s_i с a_i выше h и b_i ниже h.
Для данной двухуровневой структуры данных требуется только $O(n \log n)$ пространство и время запроса остается близким к $\sqrt{n}$.
Только что использованный принцип достаточно универсален. Обычно. он применим каждый раз. когда запрос есть конъюнкция нескольких условий (или геометрически.

пересечение нескольких областей), и для каждого условия (области) мы уже имеем подходящую эффективную структуру данных.

Рассмотрим проблему поиска в области в абстрактной форме. Каждой задаче с точечным множеством $P=\{p_1,\dots,p_n\}\subset R^d$ и с множеством $\Re$ допустимых областей поставим в соответствие систему множеств $(P,\wp)$. Здесь $\wp$ — система всех подмножеств P, определенных областями из $\Re$, т.е. $\wp=\{P\cap R \mid R\in\Re\}$. В дальнейшем можно работать только с этой системой множеств, сведя, таким образом, геометрическое содержание проблемы к минимуму.

В большинстве алгоритмов поиска в области поступают следующим образом. Определяется другая система $\sum(P)$ подмножеств P, так называемых канонических подмножеств, и дается правило, как представить каждую область $R\in\wp$ как непересекающееся объединение канонических множеств из $\sum(P)$. Назовем эту пару, систему множеств $\sum(P)$ плюс правило, как разложить множества $\wp$, схемой разложения для $(P,\wp)$.

Такая схема разложения может быть превращена в структуру данных для поиска в области (при условии, что разложения могут быть найдены эффективно). Достаточно только сохранить общий вес точек для каждого канонического подмножества.

Для деревьев разбиений канонические подмножества — это точечные множества, расположенные в областях схемы разложения в узлах дерева разбиения. Например, для одномерной структуры данных для поиска в области с интервалами канонические подмножества — это в точности канонические интервалы, и каждый интервал может быть разбит на $O(\log n)$ канонических интервалов.

В последующем необходимо, чтобы схема разложения действовала не только на системе $(P,\wp)$, но также и на системах, порожденных подмножествами P. Это означает, что для подмножества $P'\subset P$ мы также имеем схему разложения для $(P',\{P'\cap R: R\in\wp\})$. Однако, к счастью, это имеет место для геометрических проблем, поскольку, если мы можем построить схему разложения для множества, мы можем также построить ее для его подмножеств. Такая схема разложения будет называться наследственной схемой.

Теперь пусть P будет множеством базисных объектов (не обязательно точек: в начальном примере этого раздела это были сегменты). Пусть $\wp_1$, $\wp_2$ — две системы множеств P, и предположим, что мы имеем схему разложения D_1 для $(P,\wp_1)$ и наследственную схему разложения D_2 для $(P,\wp_2)$. В примере с сегментами $\wp_1$ могут быть все подмножества вида $\{s_i\in P: a_i\in H\}$ для некоторой полуплоскости H, и аналогично $\wp_2=\{s_i\in P: b_i\in H\}$ для некоторой полуплоскости H. Схема разложения для $(P,\wp_1)$ получается из дерева разбиений на множестве A для a_i-концов, и схемы разложения для $(P,\wp_2)$ — дерева разбиений для b_i-концов.

Следует получить схему разложения для более сложной системы $(P,\wp)$, где $\wp$ состоит из всех множеств вида $R_1\cap R_2$, где $R_1\in\wp_1$ и $R_2\in\wp_2$. В примере с сегментами, диапазоны, которые нас интересуют, действительно имеют такую форму, а именно, множество сегментов, a-концы которых находятся в верхней полуплоскости H_1, и b_i-концы — в нижней полуплоскости H_2.

Названо то, что получилось, составной схемой разложения D_1 и D_2 и обозначим ее $D=D_1\circ D_2$. Для разложения множества $R=R_1\cap R_2\in\wp$ мы сначала раскладываем.

используя D_1, множество $R_1 \in \wp_1$ на канонические подмножества $C_1, C_2, \ldots, C_m \in \sum_1(P)$. Затем для каждого C_i рассматриваем схему разложения D_2, действующей на подсистеме $\wp_2$, индуцированной C_i, и разбиваем множество $C_i \cap R_2$ на канонические множества $C_{i1}, C_{i2}, \ldots, C_{ik_i} \in \sum_2(C_i)$. Объединение этих совокупностей для всех $i = 1,2,\ldots,m$ дает нам разложение $R_1 \cap R_2$. Канонические множества $\sum$ в результирующей схеме разложения D будут таким образом всеми каноническими множествами из $\sum_2(C)$ для некоторого $C \in \sum_1(P)$.

То, что было сделано в примере с сегментами, может таким образом быть переписано как композиция произвольных двух схем разложения, и затем превращено в структуру данных поиска в области предвычисления весов всех канонических множеств. Если известны параметры обеих схем разложения, необходимы лишь стандартные вычисления для получения параметров составной схемы.

Как один из основных примеров рассмотрим применение многоуровневых структур данных к поиску в симплексной области. Описанный выше метод поиска в полупространственной области дает схему разложения D, которая разбивает множество точек в полупространственном запросе на $O(\log n)$ канонических подмножеств.

Поскольку симплекс представляет собой пересечение $d+1$ полупространств, можно получить симплексную схему разложения $d+1$ кратной композицией D с самой собой, что даст требуемую структуру данных для поиска в симплексной области. Простые вычисления показывают, что возникающая в результате структура данных имеет время запроса $O(\log^{d+1} n)$ и занимает память $O(n^{d+\varepsilon})$.

Выводы

1. Показано, что в случае применения алгоритмов, использующих построение деревьев разбиений, наиболее эффективным для поиска в простых областях оказалось симплексное разбиение.

2. В случае поиска в сложных областях достаточно эффективным оказалось использование линеаризации (карты Веронезе), представленное как отображение в полупространство. При этом, для решения задач со сложным движением объектов предложены методы, основанные на применении многоуровневых структур данных.

3. Исследования показали, что решение задачи, основанное на алгоритмах быстрой сортировки и бинарного поиска, требует $O(n)$ ресурсов памяти и $O(\log n)+k$ времени, а на предварительную обработку данных необходимо $O(n\log n)$ времени. Алгоритм, использующий деревья разбиений, требует $O(n^{1-\frac{1}{d}})$ времени необходимого на поиск при требуемой памяти $O(n)$, однако при этом требуется значительное время на предварительную обработку данных, причем для построения нужной структуры данных используются алгоритмы, требующие время, зависящее полиномиально от количества входных целей. Показано, что многоуровневые системы данных имеют время запроса $O(\log^{d+1} n)$ и занимают память $O(n^{d+\varepsilon})$, при этом расходуется значительное время на предварительную обработку, соответствующее квадратичному полиному от $O(n)$.

4. Все предложенные алгоритмы являются «теоретически хорошими» с точки зрения времени их выполнения. Показано, что метод решения задачи, основанный на алгоритмах быстрой сортировки и бинарного поиска наиболее прост из всех методов с линейной памятью. А методы, использующие большую, чем память размером $O(n)$, оказываются неэффективными при большом количестве объектов. Алгоритм, основанный на использовании деревьев разбиений, занимает меньше всего времени на поиск, но требует сложной и длительной предпроцессорной обработки. Доказано, что многоуровневые системы данных наиболее универсальны, но сложны для практической реализации.

Литература

1. Васюхин М.И. Методы решения задач анализа воздушной обстановки при её представлении в реальном времени: // Збірник наукових праць / ДРОТ ЦНДІОВТ ЗС України. – випуск 6. – К.: ЦНДІОВТ ЗС України, 2000.
2. Matousek Juri. Geometric Range Searching.-ACM Computing Surveys, Vol.26., №4, December, 1994. .- p.437-457.
3. Bentley. J. L. and Saxe.J. B. Decomposable searching problems 1: Static-to-dynamic transformation. J. Alg. 1. 1980. -p. 301-358.
4. Bronnimann, H., Chazelle B., and Pach J. How hard is halfspace range searching. Discrete Comput. Geom. 10. . 1993. –p.143-155.
5. Chazelle B.A . Functional approach to data structures and its use in multidimensional searching. SIAM J. Comput. 17. . 1988. –p. 427-462.

Поступила в редакційну колегію 20.05.2000 р.