

УДК 621.372.512

DOI: 10.22213/2413-1172-2024-1-89-101

Эффективные алгоритмы настройки дискретных согласующих антенных устройств ДКМВ-диапазона с использованием быстрого поиска геометрической близости точек

А. А. Извольский, кандидат технических наук, 18-й Центральный научно-исследовательский институт Министерства обороны Российской Федерации, Москва, Россия

В настоящее время интерес к вычислительным алгоритмам настройки дискретных силовых контуров согласующих антенных устройств не угасает. Вместе с тем большое количество публикаций на эту тему посвящено частным эмпирическим алгоритмам и их реализации на конкретных аппаратных платформах, не затрагивая вопросы поиска принципиально новых решений. Стремление повысить достоверность описания трансформирующих свойств дискретных силовых контуров согласующих антенных устройств на рабочих частотах определило метод их моделирования в виде матричных структур. Подобные подходы жестко связаны со структурным построением силового контура и требуют большого объема матричных вычислений с комплексными числами.

Показано, что техника дискретных согласующих антенных устройств ДКМВ-диапазона относится к такой области знаний, в которой возникающие задачи могут быть решены с помощью алгоритмов вычислительной геометрии и структур данных. Предложен легкорезализуемый вычислительный алгоритм настройки дискретного контура согласующих антенных устройств с использованием метода быстрого поиска геометрической близости точек, лишенный указанных выше недостатков. В вычислительной геометрии концепция естественных соседей связана с такими понятиями, как диаграмма Вороного, триангуляция Делоне и различными древовидными структурами данных: KD-дерево, квадродерево, дерево покрытий (cover tree), VP-дерево (vantage-point tree; дерево с точкой обзора) и др. Применение перечисленных методов является перспективным для реализации вычислительных процедур настройки дискретных согласующих антенных устройств. Однако их исполнение на микроконтроллерах – довольно сложная задача, требующая высокой квалификации программиста.

В статье обсуждается доступный и эффективный подход с использованием предварительной сортировки данных и метода «перевернутого списка». Это один из самых простых с точки зрения реализации методов. Он существенно лучше метода полного перебора, хотя и не может конкурировать с методами, использующими пространственные индексные структуры. Показан положительный эффект от применения описанных алгоритмов настройки согласующих антенных устройств ДКМВ-диапазона, в особенности, для сеансов связи с неопределенными заранее рабочими частотами, и, несомненно, эффективность методов быстрого поиска геометрической близости точек можно считать доказанной.

Ключевые слова: согласующее антенное устройство, дискретная вычислительная геометрия, диаграмма Вороного, триангуляция Делоне, древовидные структуры данных, методы поиска соседей.

Где просто, там ангелов до ста, а где мудрено, там ни одного.

Преподобный Амвросий Оптинский

Введение

Существует много областей знаний, в которых возникают геометрические задачи, решаемые с помощью геометрических алгоритмов и структур данных. Техника дискретных согласующих антенных устройств (САУ) ДКМВ-диапазона принадлежит их числу. В работе «Некоторые практические приложения матричной модели дискретно пере-

страиваемых согласующих антенных устройств» (Извольский А. А. (2023) // Вестник ИжГТУ имени М. Т. Калашникова. Т. 26, № 3) предложен переборный способ настройки САУ, основанный на обработке больших объемов данных (сортировке и поиске). С целью ограничения размерности базы данных способ следует применять к силовым контурам с числом разрядов до 16...20.

Из названия статьи следует, что в основу предлагаемого метода настройки положена дискретная вычислительная геометрия (Марк де Берг, Отфрид Чеонг, Марк ван Кревельд, Марк Овермарс. Вычислительная геометрия. Алгоритмы и приложения. 3-е изд. / пер. с англ. А. А. Слинкина. М. : ДМК Пресс, 2017). Центральным понятием геометрии и топологии является метрическое пространство – множество вместе со способом измерения расстояния между его элементами. Возникает закономерный вопрос, как же правильно выбрать метрическое пространство – диаграмму Смита с полярными координатами или комплексную плоскость с декартовыми координатами. В случае выбора диаграммы Смита работать придется с дробно-линейными (комплексными) преобразованиями плоскости, и потребуются умение записывать метрику на ней в комплексной форме. В случае конформно-евклидовых метрик это оказывается вполне возможным, но не совсем удобным. Известно, что дробно-линейное преобразование сохраняет конформно-евклидов вид метрики (Мищенко А. С., Фоменко А. Т. Краткий курс дифференциальной геометрии и топологии. М. : Физматлит, 2004).

Интуитивно хорошим выбором для дальнейшего анализа будет комплексная плоскость входных импедансов антенны. В прямоугольной системе координат каждая точка на плоскости однозначно определяется двумя числовыми координатами, которые являются расстояниями до точки от двух перпендикулярных осей, измеренными в одинаковых единицах (Ом) и имеющих одинаковый масштаб. Оси координат образуют ортонормированный базис пространства. Элементами этого пространства будут точки, и расстояние между любыми двумя точками $A = x_1 + jy_1$ и $B = x_2 + jy_2$ равно

$$\text{dist}(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = |A - B|.$$

Суть способа настройки заключается в следующем. Для всех возможных комбинаций органов настройки силового контура любой структуры и сложности с небольшим шагом изменения частоты (к примеру, 50...100 кГц) с помощью векторного анализатора цепей могут быть рассчитаны или измерены S -параметры с опорным импедансом $Z_0 = W = 50$ Ом.

В комплексной плоскости импедансов антенны $Z_A(\omega_p) = R_A(\omega_p) + jX_A(\omega_p)$ для заданного значения внутреннего импеданса источника $Z_r = W$ по формулам

$$Z_u = R_u + jX_u = \frac{(Ap_0 + B)(\bar{C}\bar{p}_0 + \bar{D}) - A\bar{C}p^2}{|Cp_0 + D|^2 - |C|^2 p^2} -$$

координаты центра окружности;

$$Rad = \frac{\rho |AD - BC|}{|Cp_0 + D|^2 - |C|^2 p^2} - \text{радиус окружности,}$$

где p_0 – координаты центра окружности; ρ – радиус (модуль коэффициента отражения на входе САУ) на диаграмме Вольперта – Смита;

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} W & W \\ -1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -S_{11} \\ S_{22} & -\Delta S \end{bmatrix};$$

$\Delta S = S_{11}S_{22} - S_{12}S_{21}$ – коэффициенты матрицы, могут быть рассчитаны координаты центра Z_u и радиус окружности Rad , соответствующие допустимому уровню рассогласования $|K_U| = \rho$.

Полученные данные, сведенные в массив, могут быть сохранены в специальном запоминающем устройстве контроллера управления САУ. Важно заметить, что предложенный переход от коэффициентов матриц к координатам центров окружностей и их радиусов позволил уменьшить объем сохраняемых данных.

Процесс настройки САУ на рабочей частоте ω_p по результатам однократного измерения входного импеданса антенны сводится к поиску лучшего варианта комбинации дискретных органов согласования силового контура. Условием настройки САУ будет попадание точки, отображающей входной импеданс антенны на рабочей частоте $(R_A + jX_A)$, внутрь рассчитанной окружности, соответствующей некоторой комбинации M силового контура. Математически это соответствует ситуации, когда расстояние между точкой импеданса антенны и центром окружности меньше радиуса окружности:

$$\begin{aligned} \text{dist}(Z_A(\omega_p), Z_u(M)) &= \\ &= \sqrt{(R_A(\omega_p) - R_u(M))^2 - (X_A(\omega_p) - X_w(M))^2}; \\ \text{dist}(Z_A(\omega_p), Z_u(M)) &\leq Rad(M). \end{aligned}$$

При количестве органов согласования в силовом контуре более двух можно найти несколько вариантов настройки с близкими характеристиками. При сохранении в массиве данных информации об относительных потерях в контуре может быть осуществлен выбор варианта согласования с наименьшими потерями.

В основу рассуждений положены соображения о том, что всю информацию о силовом контуре САУ в диапазоне частот можно задать в дискретном виде, то есть конечным набором данных. Принимается тезис, что любую информацию в задачах обработки информации можно «подменить» дискретной информацией. Причем в нашем случае эта информация должна быть получена эмпирическим путем. В общем виде обработку информации можно представить как извлечение искомой информации из исходной. Необходимо признать, что искомая информация обязательно должна содержаться в исходной, и в принципе никакой новой информации в результате обработки и поиска нет.

Цель исследования – разработать вычислительный алгоритм настройки САУ, не содержащий вычислений в традиционном смысле слова, но осуществляющий быстрый целенаправленный поиск нужной информации о настройке в базе данных большого объема по результатам измерения входного импеданса антенны.

Поскольку речь идет об автоматическом поиске, исходная информация должна быть представлена в специальной форме, которая упростит процесс ее обработки. Однако это можно проделать, только затратив некоторый ресурс.

Хранение исходной информации

Исходная информация хранится в базе данных. База данных – это совокупность информации, организованная так, чтобы доступ к ней для контроллера САУ был удобен. База данных в нашем случае статическая, не претерпевает изменений и оформлена в виде таблиц. Для понимания нужно ясно представлять ключевые термины из организации базы данных, в частности следующие фундаментальные концепции: запись, поле, индекс и запрос.

Рабочий диапазон САУ разбит на множество фиксированных частот. Для каждого дискретного значения частоты своя пара таблиц. В терминах баз данных каждая такая строка называется записью. Каждая запись содержит несколько типов сведений о силовом контуре САУ на дискретной частоте. Разные виды сведений называются полями. Поля соответствуют колонкам. Первая таблица содержит три различных поля: номер точки n , координаты центра окружности $R_{ц}$, $X_{ц}$ (два поля). Вторая таблица – четыре поля: номер точки n , квадрат радиуса окружности Rad^2 , кодовая комбинация органов настройки $Cod(n)$, относительные потери η . Первая таблица хранит данные о координатах точек и будет использоваться только для их по-

иска. Если требуется сравнить расстояние с каким-то числом, удобнее не извлекать корень и сравнивать квадрат расстояния с квадратом числа.

Для быстрой обработки запросов (поиска по базе) базы данных используется механизм индексации. Без индексов было бы сложно найти что-либо в базе из сотен тысяч записей, не говоря уже о базах из миллионов записей. Индексация позволяет контроллеру САУ выбирать записи последовательно по порядку перечисления в ключевом поле.

Постановка задачи

Перейдем к детальному рассмотрению предлагаемого способа настройки. Каждой из множества N дискретных комбинаций силового контура САУ на комплексной плоскости входных импедансов антенны соответствуют N точек импеданса антенны, которые могут быть идеально согласованы. В случае допустимого рассогласования эти точки «раздуются» в окружности, при этом произойдет смещение их центров. При определенном уровне рассогласования часть комплексной плоскости будет «покрыта» кругами без пробелов, с наложениями. Каждая точка «покрытой» области входных импедансов может быть согласована с наперед заданным качеством данным контуром САУ на рабочей частоте. Заметим, что при покрытии кругами (см. рис. 1) обязательно образуются зоны с многократным покрытием. Наличие этих зон указывает на инвариантность настройки.

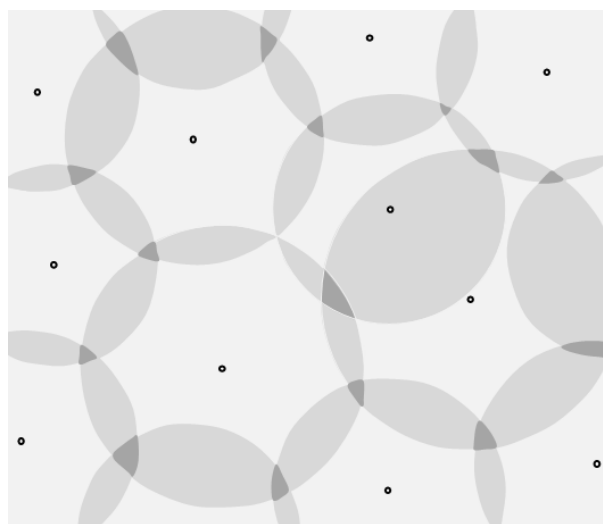


Рис. 1. Покрытие прямоугольной области кругами

Fig. 1. Covering a rectangular area with circles

Таким образом, задача настройки САУ [1] может быть сведена к статической задаче о покрытии поля (Хорьков А. В. Математические

модели, алгоритмы и программы оптимизации многократного покрытия ограниченных множеств : дис. ... канд. техн. наук. Казань, 2019). Для эффективного выполнения задачи поиска на покрытом поле необходимо его правильно разделить на зоны.

Аналогия с геоинформационными системами

Можно провести следующую аналогию с картами геоинформационных систем [2] и автомобильными навигаторами. Известно, что в навигаторах карта хранится в виде нескольких тематических слоев, представляющих, например, дороги, населенные пункты, магазины, заправочные станции, рестораны, отели и др. Выделение отдельных слоев позволяет сосредоточиться на информации одного типа. С помощью специального приемника спутниковой навигации определяется точное расположение автомобиля с привязкой к электронной карте.

Геоинформационная система весьма полезна, поскольку она достаточно детализирована. Доскональная карта Европы содержит гигантский объем данных. В каждый момент времени отображать нужно только малую часть карты. Однако система должна уметь эту часть найти: зная прямоугольную область, или окно, система должна определить, какая часть карты (дороги, города и др.) находится в этом окне, и отобразить ее. Это называется оконным запросом.

При значительном объеме данных, с которым приходится иметь дело в дискретных САУ, проверять для каждого элемента «частотной» карты, попадает ли он в окно, очевидно, бессмысленно. Нужно сохранить карту в какой-то структуре данных, которая позволит быстро находить часть внутри заданного окна. В оконных запросах пространство поиска двумерное.

Комплексная плоскость входных импедансов антенны для фиксированной рабочей частоты, зарегушированная окружностями, соответствующими всем комбинациям дискретных органов силового контура, по сути является картой САУ. При использовании такой карты часто возникает запрос о локализации точки: имея карту и координаты точки q , найти область на карте, содержащую q . Конечно, карта есть не что иное, как разбиение плоскости на области, то есть планарное разбиение.

При каждом измерении входного импеданса антенны на заданной частоте (для каждой частоты своя карта САУ) поступают такие запросы. Отвечать на них необходимо быстро в реальном

времени. Значит, нужна удобная структура данных, поддерживающая частые запросы о локализации точки и поиске ее соседних точек.

Базовые определения по вопросу поиска ближайших соседей

1. Задача эффективного по времени поиска k ближайших соседей [3, 4] состоит в создании такой структуры из множества точек S , в которой можно быстро найти k ближайших соседей из множества S для данной точки запроса q (Гусев Д. И. Алгоритм поиска ближайших соседей // Программные продукты и системы. 2012. № 3).

Введем базовые определения.

Точка запроса – точка, для которой необходимо найти ближайшего соседа из множества точек S , будет обозначаться как q : $q \in R^2$.

Истинный ближайший сосед. Пусть дано множество S из N точек, векторное пространство $V = R^2$ с определенной нормой $dist(p, q) = |p - q|$ и точка запроса $q \in V$, тогда точка p – истинный ближайший сосед точки q тогда и только тогда, когда $dist(p, q) = |p - q| = \min\{|p' - q|; p' \in S \setminus q\}$.

Меры эффективности алгоритмов

Исследовать алгоритм можно с разных точек зрения. Очевидно, что алгоритм должен решать задачу правильно. Второе свойство алгоритмов – эффективность, или время работы. Желательно, чтобы алгоритм работал быстро, но существуют теоретические пределы быстродействия алгоритма, зависящие от задачи. Помимо правильности и времени работы при рассмотрении алгоритмов зачастую следует принимать во внимание организацию данных и конкретную реализацию.

Известны две фундаментальные меры эффективности алгоритмов. Первая определяет время счета алгоритма, вторая – пространство, обычно совпадающее с объемом памяти, использованной алгоритмом. В монографии Препарата Ф., Шеймос М. «Вычислительная геометрия: Введение» (М. : Мир, 1989) на стр. 19 приведены оценки сложности геометрических вычислений, аппарат обозначений которых мы и заимствуем. Общепринято оценивать сложность комбинаторных геометрических алгоритмов для задач данного размера для худшего случая.

При анализе действий по поиску данных в специально подготовленных структурах сосредоточимся на двух оценках:

- сколько времени необходимо в худшем случае для ответа на один запрос;

- сколько памяти необходимо для хранения структуры данных.

На времени, необходимом для организации данных перед поиском, фокусироваться не будем. Корректировать (добавлять, удалять) данные нет необходимости.

Краткий обзор известных методов поиска ближайших соседей

Диаграмма Вороного

Диаграмма Вороного (или двойственная ей триангуляция Делоне) является одной из фундаментальных структур данных вычислительной геометрии. Под классической диаграммой Вороного для набора точечных объектов на плоскости понимается разбиение плоскости на

ячейки Вороного, каждая из которых является геометрическим местом точек, расположенных ближе к данному объекту, чем к остальным. Подробное изложение теории диаграмм Вороного можно найти в [5–9].

На рисунке 2 показан пример построения диаграммы Вороного.

Для построения диаграммы Вороного существует множество методов. Самый быстрый по скорости работы и относительно простой в реализации – метод Стивена Фортуна (Fortune S. J. A sweep line algorithm for Voronoi diagrams // J. Algorithmica, 1987, No. 2, pp. 153–174), его временная сложность составляет $O(N \log N)$.

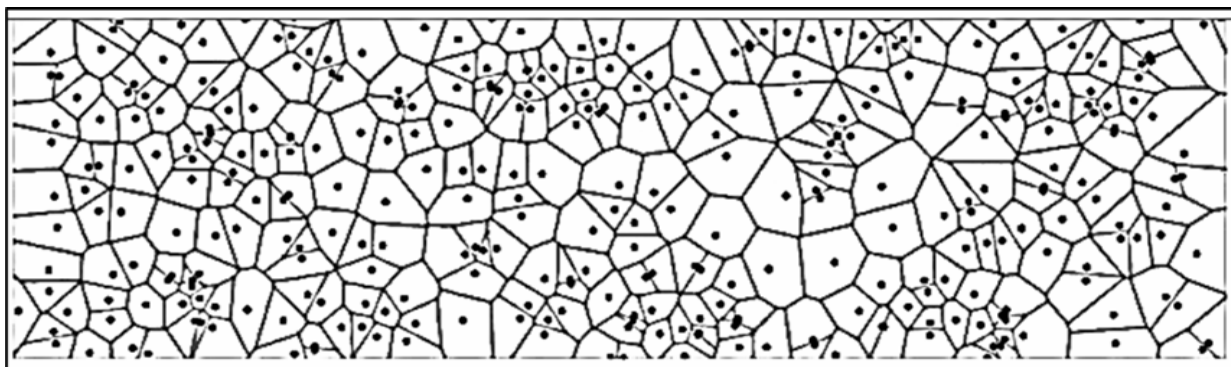


Рис. 2. Иллюстрация построения диаграммы Вороного

Fig. 2. Illustration of building a Voronoi diagram

Природа геометрических задач привела к созданию специальных необычных структур данных, одна из которых называется реберный список с двойными связями (РСДС) (de Berg M., van Kreveld M. Computational Geometry. Algorithms and Applications. Second, Revised Edition. Berlin: Springer-Verlag, 2000). Диаграмма Вороного представляет собой сложную структуру данных, хранящую всю необходимую информацию для решения множества задач о близости.

Трудности использования на практике диаграммы Вороного

Исследование диаграммы Вороного в настоящей работе мотивировано желанием повысить быстродействие настройки САУ. Вместе с тем важно учитывать следующие обстоятельства.

В нашем случае пространство является конечномерным евклидовым пространством, каждая точка является точкой, существует конечное число точек, и все они различны, при этом ячейки Вороного являются выпуклыми многогранниками. Однако в общем случае ячейки Вороного могут быть невыпуклыми и даже несо-

единенными. Ребра между ячейками Вороного представляют собой отрезки, лучи или линии на плоскости, а вершины между ячейками – точки. Ребра и вершины могут не существовать, если соответствующее пересечение ячеек Вороного пусто. Иногда ребро может вырождаться в точку, а несколько вершин могут совпадать друг с другом. Это происходит, когда плоскость содержит четыре и более точек, лежащих на одной окружности.

Вместе с тем для эффективного программного представления диаграммы Вороного требуются довольно сложные структуры данных. Трудоемкость этих представлений связана не только с наборами данных, но и необходимостью поддерживать отношения смежности между элементами разных типов представления. Например, ребро должно быть связано с предыдущим и следующим ребрами на границе ячейки; кроме того, ребро должно хранить по крайней мере одну ссылку на ячейку, заключенную в границу [10; <https://www.codeproject.com/Articles/882739/Simple-Approach-to-Voronoi-Diagrams>, 2020].

К сожалению, высокая трудность построения и обслуживания этих структур данных является серьезным препятствием для массового использования преимуществ этого мощного математического замысла на практике.

Геометрический поиск (виды и методы)

Задача поиска ближайшего соседа в маломерном векторном пространстве хорошо изучена. Существует большое количество алгоритмов. В статье не рассматриваются алгоритмы, которые основаны на дискретном метрическом пространстве, а также алгоритмы, используемые для поиска среди очень большого количества данных, когда используется активность чтения и записи с жесткого диска, так как данный объем информации не удастся хранить в оперативной памяти из-за их низкой эффективности на данных небольшого размера ($N \leq 10^6$).

Метод полного перебора (brute force)

Полный перебор, или линейный поиск, – наиболее простой в реализации алгоритм, который просматривает каждый элемент множества и проверяет, удовлетворяет ли тот заданному условию. Важным достоинством является отсутствие необходимости в подготовительных операциях, алгоритм может работать сразу на множестве элементов. Для решения поставленной задачи метод неприемлем в силу линейной сложности, $O(N)$ – не лучший алгоритм обработки больших массивов.

На ранних стадиях разработки алгоритма САУ разумно рассмотреть и реализовать вариант полного перебора точек множества. Он основан на простом алгоритме, который вычисляет расстояние между точкой запроса и каждой точкой множества и находит точку с минимальным расстоянием.

Метод сетки

Неэффективность метода полного перебора может быть устранена с помощью популярного сеточного метода (<https://helenkapatsa.ghost.io/alghoritm-poiska-po-sietkie/>). При простейшем варианте этого метода строятся квадратные ячейки одинакового размера. В каждой сеточной ячейке хранится список содержащихся в ней точек. Алгоритм поиска вычисляет расстояние между точкой запроса (незакрашенная точка) и точками в серых ячейках (рис. 3).

Поиск начинается с ячейки, содержащей точку запроса, и продолжается в ячейках, смежных с посещенными ячейками. Поиск прекращается, когда расстояние между точкой запроса и ближайшей непросмотренной ячейкой становится

больше минимального расстояния между точкой запроса и точкой в подмножестве обработанных данных.

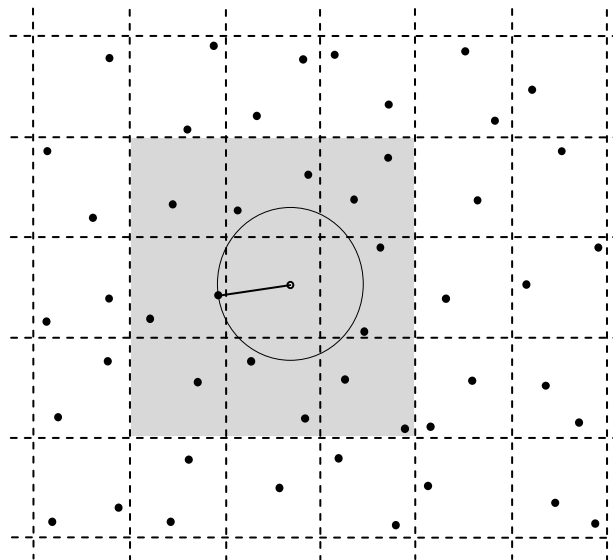


Рис. 3. Иллюстрация метода сетки

Fig. 3. Illustration of the grid method

Метод сетки значительно проще, чем метод, основанный на пространственной структуре данных, но он не является полностью тривиальным и имеет ряд ограничений. Этот метод обеспечивает наилучшую производительность, когда точки равномерно распределены по ячейкам. Рациональный размер ячейки зависит от плотности точек. Производительность метода снижается, когда в сетке слишком много пустых ячеек или когда среднее количество точек в одной ячейке становится слишком большим.

Важным этапом при реализации алгоритма быстрого поиска методом сетки является определение оптимального числа разбиений – такого разбиения области на клетки, при котором время вычислений минимально. Для двумерного случая известны такие подходы, когда область считается прямоугольником со сторонами, параллельными осям координат.

Метод KD-дерева (KD-tree)

Чтобы устранить вычислительную неэффективность подхода грубой силы, были изобретены различные древовидные структуры данных. KD-дерево, или K-мерное дерево, – структура данных, служащая для организации точек в K-мерном пространстве. Является частным случаем бинарного дерева поиска. Разбиение пространства происходит по одной из координатных осей так, чтобы точка разбиения была посередине между наиболее удаленными объектами вдоль этой оси. Разбиение происходит ре-

курсивно с циклическим чередованием координатных осей [11; <https://en.wikipedia.org/wiki/Quadtree>]. Является наиболее простым видом дерева, поиск эффективней линейного, сложность алгоритма поиска ближайших соседей для

двухмерного случая составляет $O(2\sqrt{N})$ и может быть уменьшена.

Пример карты KD-дерева представлен на рисунке 4 – запрос к круговому диапазону.

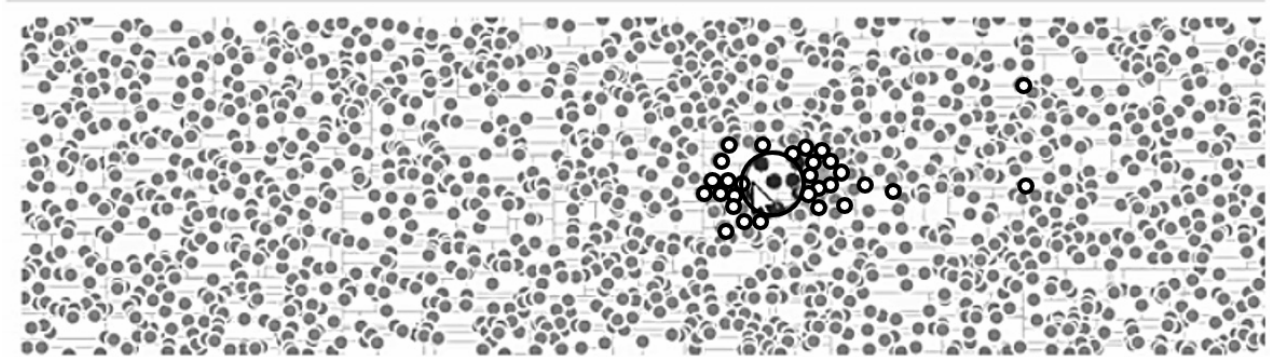


Рис. 4. Иллюстрация метода KD-дерева

Fig. 4. Illustration of KD-tree method

Квадродерево

Деревья квадрантов используются для рекурсивного разбиения двухмерного пространства по 4 квадранта. Области представляют собой квадраты, прямоугольники или имеют произвольную форму. Англоязычный термин *Quad Tree* был придуман Рафаэлем Финкелем и Джоном Бентли в 1974 году (Raphael Finkel and Bentley J. L. Quad Trees: A Data Structure for Retrieval on Composite Keys. Acta Informatica, 1974, vol. 4, no. 1, pp. 1-9).

Основное преимущество этой структуры данных заключается в адаптивности: она обеспечивает хорошую производительность при вставке, удалении и поиске. Quad tree рекурсивно делит плоскость на четыре квадранта, как показано на диаграмме рисунка 5.

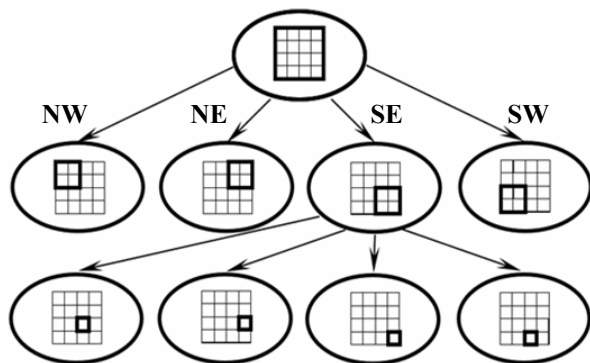


Рис. 5. Принцип рекурсивного деления 2D-пространства

Fig. 5. Principle of recursive vision of 2D-space

Основным достоинством квадродеревьев является возможность с их помощью описывать сколь угодно сложные пространственные области при незначительных затратах памяти ЭВМ. Иерархическая организация структуры квадродеревьев позволяет осуществлять эффективный доступ к данным.

Известны несколько разных видов квадродеревьев: региональные (Region), точечные (Point), точка-регион (PR Quad Tree), ребро (Edge Quad Tree). Подробное описание можно найти в книге Сарилы Хар-Пеледа (2011) «Проект по математической генеалогии». В интернете имеются реализации на языках Java, Python, C++, C#, которые поддерживают статические/динамические данные и обычные пространственные запросы [12–14].

Известно, как построить алгоритм поиска в круговом и прямоугольном диапазонах [15]. Пример показан на рисунке 6.

Стоимость построения точечного квадродерева из случайного набора точек составляет $O(N \log_4 N)$. Простой поиск в сбалансированном точечном квадродереве имеет временную сложность $O(N \log_4 N)$, в худшем случае, когда на каждом уровне дерева находится только один узел, сложность равна $O(N)$. Показано, что стоимость поиска в диапазоне $O(N^{1/2})$ в худшем случае (Bentley et al., 1977; Lee and Wong, 1977). В работе Lee and Wong (1977) также показано, что поиск k точек занимает время $O(kN^{1/2})$ в худшем случае.

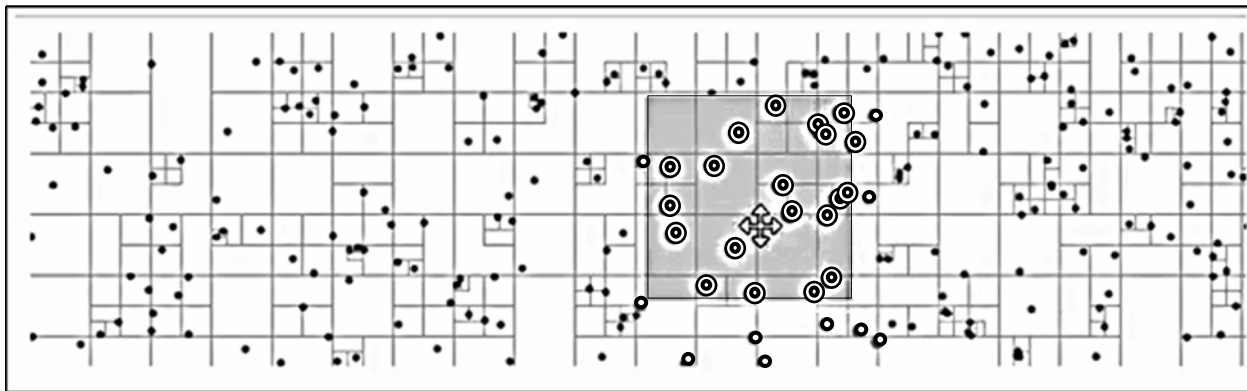


Рис. 6. Иллюстрация поиска в прямоугольном диапазоне в структуре квадродерева

Fig. 6. Illustration of rectangular range search in a Quad Tree structure

Дерево покрытий (cover tree)

Дерево покрытий – древовидная структура данных (дерево), специально разработанная для ускорения поиска ближайшего соседа. Дерево можно рассматривать как иерархию, верхний уровень которой содержит корневую точку, а нижний – все точки в метрическом пространстве. Каждому уровню C соответствует целое число i , которое уменьшается на единицу в каждом нижнем уровне. Каждый уровень C в дереве покрытий имеет три важных свойства:

- вложенность, $C_i \subseteq C_{i-1}$;
- покрытие, для каждой точки $p \in C_{i-1}$ существует точка $p \in C_i$ такая, что расстояние от p до q меньше или равно 2^i , и ровно одна такая точка q является предком точки p ;
- разделение, для всех точек $p, q \in C_i$ расстояние от p до q больше или равно 2^i .

Подробности этого метода может найти в работах Alina Beygelzimer, Sham Kakade, John Langford (2006) “Cover Trees for Nearest Neighbor”; Пролубников А. В. (2017) «Об одном подходе к решению задачи о покрытии с интегральными весами и его вычислительной сложности».

VP-дерево (vantage-point tree, дерево с точкой обзора)

VP-дерево можно построить для объектов из метрического пространства. Из первоначального множества берется одна из точек (опорная точка) и выбирается радиус R для этой точки. Остальные точки делятся на два подмножества – с расстоянием меньше R до опорной точки и расстоянием больше R . В каждом из получившихся подмножеств выбирается следующая опорная точка и новый радиус, и т. д., пока количество элементов в каждом из оставшихся подмножеств не станет меньше определенного порогового значения.

Опорные точки и радиусы сфер (на плоскости окружностей) разбиения выбираются так, чтобы дерево получилось максимально сбалансированным. В отличие от KD-дерева, которое применимо только для точек из R^k , VP-дерево может быть использовано для поиска ближайших объектов из любого метрического пространства.

Пример карты VP-дерева представлен на рисунке 7.



Рис. 7. Иллюстрация метода VP-дерева

Fig. 7. Illustration of the Vantage-Point Tree method

Работа с точками происходит только с помощью функции расстояния, алгоритм с отдельными координатами точек не используется, поэтому размерность пространства и его замкнутость не имеют значения (Yianilos P.N. (1993) Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces).

Сложность алгоритма построения дерева $O(N \log N)$, алгоритма поиска – $O(\log N)$ [16, 17].

Применение методов KD-дерева, квадродерева, дерева покрытий и VP-дерева является перспективным для реализации вычислительных процедур настройки дискретных САУ.

Метод редких точек

Идея алгоритма заключается в хранении списка объектов, имеющих «редкие» координаты. В нашем случае также метод неприемлем в силу невозможности выделить «редкие» координаты в метрическом пространстве с числовыми координатами. Сложность алгоритма поиска ближайших соседей составляет $O(N)$ [18].

Таким образом, для реализации вычислительных процедур настройки дискретных САУ наиболее перспективными являются методы VP-дерева, KD-дерева, квадродерева и дерева покрытий. Построение каждой из этих структур на микроконтроллере является сложнейшей задачей, требующей высокой квалификации программиста. Для преодоления этой сложности рассмотрим простой подход, не требующий построения специальной структуры данных.

Идея простого варианта

Идея практического варианта по замене диаграммы Вороного заключается в снижении сложности представления за счет хранения и использования минимально возможного количества наборов данных. В предлагаемом минимальном варианте частотная карта будет представлена только набором узлов (центров окружностей) с радиусами окружностей. Вторая процедура этого подхода – алгоритм поиска ближайших соседей, который решает задачу определения местоположения точки и нахождения соседей в ее окрестности. Если поиск ближайшего соседа будет максимально упрощен, предлагаемый подход обеспечит основное преимущество такого представления «карты» – низкую стоимость разработки и обслуживания.

Метод инвертированного списка

В этой работе мы сосредоточимся на методе инвертированного списка [19]. С точки зрения реализации метод простой; он несколько сложнее, чем метод полного перебора, однако проще, чем сеточный метод и пространственное индексирование. С точки зрения производительности этот метод превосходит метод перебора, хотя он не может конкурировать с методами, использующими пространственные индексные структуры.

Метод инвертированного списка основан на упорядоченном наборе данных. Для набора двумерных точек это требование подразумевает, что набор узлов должен быть заранее отсортирован по координате R (или X -координате) точки, что значительно сокращает диапазон поиска.

Как показано на рисунке 8, порядок позволяет алгоритму избежать поиска в регионах, которые довольно удалены от точки запроса.

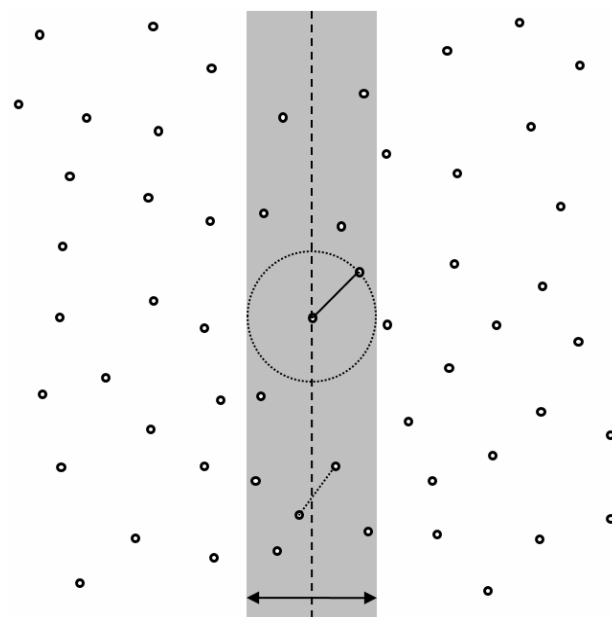


Рис. 8. Поиск ближайшего соседа в упорядоченном по одной координате наборе точек

Fig. 8. Finding the nearest neighbor in a set of points ordered by one coordinate

Координата R точки запроса пересекает отрезок пунктирной линии между двумя начальными точками прямого и обратного последовательных поисков. Серая область содержит минимальное подмножество точек, которые должен посетить алгоритм.

Таким образом, алгоритм поиска ближайшего соседа состоит из двух этапов. Во-первых, алгоритм находит начальную позицию поиска, заданную точкой запроса в наборе точек. Эта позиция указывает на две точки, координаты R которых образуют диапазон, содержащий координату R точки запроса. При этом множество точек разбивается на два подмножества. Затем начальная позиция используется для запуска двух последовательных поисков в возрастающем и убывающем порядке точек. Поиск перемещается вперед или назад, и вычисляется расстояние между точкой запроса и текущей точкой подмножества. Каждый поиск прекращается, когда

абсолютное значение разницы между координатами R точки запроса и текущей точкой становится больше вычисленного минимального расстояния от точки запроса до точки в обрабатываемом подмножестве.

Принцип, лежащий в основе методов ближайшего соседа, состоит в том, чтобы найти предопределенное количество точек, ближайших по расстоянию к новой точке. При таком поиске требуется найти k точек в каждом направлении (возрастающем и убывающем порядке точек), затем из объединенного списка, содержащего $2k$ точек, после сортировки их по дистанции выбрать k ближайших точек.

Вычислительная сложность алгоритма

Производительность этого алгоритма определяется вторым этапом обработки. В среднем время выполнения каждого последовательного поиска пропорционально квадратному корню из числа точек. Поиск начальной позиции более эффективен. В упорядоченном наборе точек в худшем случае время выполнения логарифмическое.

По сравнению с поиском методом перебора обсуждаемый алгоритм обеспечивает повышение производительности на порядки. Например, когда количество точек близко к 1000000, этот алгоритм более чем в 100 раз быстрее, чем линейный поиск.

Важным преимуществом обсуждаемого алгоритма поиска ближайшего соседа является эффективная поддержка запросов прямоугольного диапазона. Его можно использовать для поиска всех соседей узла и построения его ячейки за один проход алгоритма.

На сортировку массива будет потрачено $O(N \log N)$, а на поиск – $O(N)$. На практике оценка на поиск оказывается гораздо меньше. Положительным моментом является то, что массив не претерпевает никаких изменений, и сортировку можно осуществить заранее.

Предложенный алгоритм настройки САУ оказывается очень удобным при работе в сеансах связи с заранее неизвестными рабочими частотами [20] по сравнению, например, с алгоритмом, описанным в [21]. В заданном диапазоне частот необходимо произвести настройку на всех дискретных значениях частоты, на которых измеряются S -параметры всех возможных комбинаций органов настройки силового контура. Информацию о параметрах настройки блока САУ на всех частотах диапазона удобно хранить в энергонезависимой памяти контроллера.

Таким образом, процесс обучения САУ осуществляется без процедуры перебора органами

настройки, сохраняя тем самым ресурс коммутационных элементов. Алгоритм позволяет для каждой частоты найти несколько вариантов настройки с заданным качеством по согласованию и выбрать лучший, например, по коэффициенту полезного действия. Процесс обучения САУ при настройке на 100 частотах займет около 2 секунд.

Выводы

В рамках настоящей статьи выполнена экспериментальная проверка и оценка эффективности предложенного алгоритма, реализованного на языке C++. Результаты экспериментов подтвердили ожидаемую высокую производительность нового алгоритма, а также его простоту исполнения. В качестве наполнения дальнейших работ видится повышение скорости поиска при переходе к VP-дереву, а также проведение исследований в области комбинированных алгоритмов.

Показан положительный эффект от реализации предложенных алгоритмов настройки САУ ДКМВ-диапазона, в особенности для сеансов связи с неопределенными заранее рабочими частотами; эффективность методов быстрого поиска геометрической близости точек для данного применения доказана.

Выбор рационального алгоритма для двумерного пространства $D=2$ (плоскости) для набора данных с размерностью $N=2^{16} \dots 2^{20}$ (65536...1048576) весьма сложен и зависит от ряда факторов:

- размерность набора данных;
- структура данных в смысле разреженности;
- число запрашиваемых ближайших соседей k .

Заметим, что для каждой частоты при наличии одной антенны требуется осуществить один запрос с числом $k=10 \dots 12$.

Для плоскости асимптотически оптимальное решение ведет к диаграмме Вороного (Arya S., Mount D.M., Netanyahu N.S., Silverman R., Wu A.Y. (1998) An optimal algorithm for approximate nearest neighbor searching in fixed dimensions), что на практике является медленным решением, так как имеет большую константу перед оценкой.

Для реализации вычислительных процедур настройки дискретных САУ наиболее перспективными являются методы KD-дерева, VP-дерева, квадродерева и дерева покрытий. Построение каждой из этих структур на микроконтроллере является трудной задачей, требующей высокой квалификации программиста.

Для преодоления этой сложности предложен простой подход с использованием метода перевернутого списка. Он существенно лучше метода

полного перебора, хотя и не может конкурировать с методами, использующими пространственные индексные структуры.

В завершение следует отметить, что VP-деревья представляют собой необычную структуру разделения двоичного пространства; они обеспечивают некоторые преимущества по сравнению со своими аналогами – KD-деревьями и квадродеревьями – и сокращают продолжительность поиска.

Библиографические ссылки

1. Лемперт А. А., Казаков А. Л., Ле. К. М. О задаче многократной упаковки кругов в ограниченное множество // *Динамические системы, оптимальное управление и математическое моделирование*. 2019. С. 397–399.
2. Сяо Н. Алгоритмы ГИС / пер. с англ. А. А. Слинкина. М.: ДМК Пресс, 2021. – 328 с.
3. Дронов С. В. Структура коллектива ближайших соседей в семействе разбиений конечного множества // *Прикладная дискретная математика*. 2020. № 47. С. 5–15.
4. Yury A Malkov and Dmitry A Yashunin (2018) Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
5. Киреев Т. Ф., Булгакова Г. Т. Построение диаграммы Вороного с ограничениями на плоскости // *Вычислительные технологии*. 2019. Т. 24, № 4. С. 28–37.
6. Совертков П. И. Диаграмма Вороного на плоскости с различными метриками // *Математика для школьников*. 2023. № 3. С. 8–20.
7. Авсеева О. В., Воротилина А. И. Реализация алгоритма построения диаграммы Вороного // *Математика, информационные технологии, приложения*. 2023. С. 97–101.
8. Чумаченко А. А., Шадричева М. С. Триангуляция Делоне // *Алея науки*. 2017. Т. 1, № 12. С. 413–415.
9. Получение данных о территориальных объектах на основе алгоритма триангуляции Делоне / Е. Н. Ким, В. В. Яворский, Е. Г. Ключева, М. М. Есмагамбетова // *Труды университета*. 2022. № 3 (88). С. 314–320.
10. Осипов А. С. О нечетких алгоритмах ближайших соседей // *Труды научно-исследовательского института системных исследований Российской академии наук*. 2018. Т. 8, № 5. С. 153–161.
11. Маньяков Ю. А., Сорокин А. И. Метод сегментации изображений на основе квадродерева // *Системы и средства информатики*. 2020. Т. 30, № 4. С. 102–112.
12. Park K. (2019) A hierarchical binary quadtree index for spatial queries. *Wireless Networks*, vol. 25, no. 4, pp. 1913–1929.
13. Lipp M., Helmig R. (2020) A locally-refined locally-conservative quad tree finite-volume staggered-

grid scheme. *Fluid Mechanics and its Applications*, vol. 121, pp. 149–159.

14. Pochet A., Celes W., Lopes H., Gattass M. (2017) A new quad tree-based approach for automatic quadrilateral mesh generation. *Engineering with Computers*, vol. 33, no. 4, pp. 275–292.

15. Paoli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright (2020) Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

16. Могилко А. А. Параллельный алгоритм поиска ближайшей точки в радиусе // *Наука и образование*. 2019. № 11. С. 363–382.

17. Мясников Е. В. Выбор опорных точек при построении VP-деревьев для поиска дубликатов на цифровых изображениях // *Информационные технологии и нанотехнологии*. 2021. С. 23–29.

18. Ефимчик А. А. Анализ алгоритмов поиска в больших массивах данных // *Вестник современных исследований*. 2018. № 12.1 (27). С. 537–541.

19. Scholer H.E., Williams J. Yiannis and J. Zobel (2022) Compression of inverted indexes for fast query evaluation: Proc. of the ACM-SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, August 2022.

20. Николашин Ю. Л., Будко П. А., Жуков Г. А. Основные направления модернизации декаметровых систем связи // *Техника средств связи*. 2019. № 1 (145). С. 13–25.

21. Уткина А. В. Алгоритм работы антенно-согласующих устройств в сеансах связи, где частоты заранее не определены // *Техника радиосвязи*. 2019. Вып. 1 (40). С. 38–45.

References

1. Lempert A.A., Kazakov A.L., Le K.M. (2019) [On the problem of multiple packing of circles into a bounded set]. *Dinamicheskie sistemy, optimal'noe upravlenie i matematicheskoe modelirovanie*, pp. 397–399 (in Russ.).
2. Sjaio H. (2021) [GIS Algorithms]. Moscow, DMC Press Publ., 328 p. (in Russ.).
3. Dronov S.V. (2020) [Structure of a collective of nearest neighbors in a family of partitions of a finite set]. *Prikladnaja diskretnaja matematika*, no. 47, pp. 5–15 (in Russ.).
4. Yury A Malkov and Dmitry A Yashunin (2018) Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
5. Kireev T.F., Bulgakova G. T. (2019) [Constructing a Voronoi diagram with plane constraints]. *Vychislitel'nye tehnologii*, vol. 24, no. 4, pp. 28–37 (in Russ.).
6. Sovertkov P.I. (2023) [Voronoi diagram on the plane with various metrics]. *Matematika dlja shkol'nikov*, no. 3, pp. 8–20 (in Russ.).

7. Avseeva O.V., Vorotilina A.I. (2023) [Implementation of the algorithm for constructing a Voronoi diagram]. *Matematika, informacionnye tehnologii, prilozhenija*, pp. 97-101 (in Russ.).
8. Chumachenko A.A., Shadricheva M.S. (2017) [Trianguljacija Delone]. *Aleja nauki*, vol. 1, no. 12, pp. 413-415 (in Russ.).
9. Kim E.N., Javorskij V.V., Kljueva E.G., Esma-gambetova M.M. (2022) [Obtaining data on territorial objects based on the Delaunay triangulation algorithm]. *Trudy universiteta*, no. 3, pp. 314-320 (in Russ.).
10. Osipov A.S. (2018) [On fuzzy nearest neighbor algorithms]. *Trudy nauchno-issledovatel'skogo instituta sistemnyh issledovanij Rossijskoj akademii nauk*, vol. 8, no. 5, pp. 153-161 (in Russ.).
11. Man'jakov Ju.A., Sorokin A.I. (2020) [Quadtree based image segmentation method]. *Sistemy i sredstva informatiki*, vol. 30, no. 4, pp. 102-112 (in Russ.).
12. Park K. (2019) A hierarchical binary quadtree index for spatial queries. *Wireless Networks*, vol. 25, no. 4, pp. 1913-1929.
13. Lipp M., Helmig R. (2020) A locally-refined locally-conservative quad tree finite-volume staggered-grid scheme. *Fluid Mechanics and its Applications*, vol. 121, pp. 149-159.
14. Pochet A., Celes W., Lopes H., Gattass M. (2017) A new quad tree-based approach for automatic quadrilateral mesh generation. *Engineering with Computers*, vol. 33, no.4, pp. 275-292.
15. Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright (2020) Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261-272.
16. Mogilko A.A. (2019) [Parallel algorithm for finding the nearest point in a radius]. *Nauka i obrazovanie*, no. 11, pp. 363-382 (in Russ.).
17. Mjasnikov E.V. (2021) [Selecting reference points when constructing VP-trees for searching for duplicates in digital images]. *Informacionnye tehnologii i nanotehnologii*, pp. 23-29 (in Russ.).
18. Efimchik A.A. (2018) [Analysis of search algorithms in large data sets]. *Vestnik sovremennyh issledovanij*, no. 12.1, pp. 537-541 (in Russ.).
19. Scholer H.E., Williams J. Yiannis and J. Zobel (2022) Compression of inverted indexes for fast query evaluation: Proc. of the ACM-SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, August 2022.
20. Nikolashin Ju.L., Budko P.A., Zhukov G.A. (2019) [The main directions of modernization of the decimeter communication system]. *Tehnika sredstv svyazi*, no. 1, pp. 13-25 (in Russ.).
21. Utkina A.V. (2019) [Algorithm for operation of antenna-matching devices in communication sessions where frequencies are not predetermined]. *Tehnika radiosvyazi*, iss. 1, pp. 38-45 (in Russ.).

Advanced Algorithms for Configuring Discrete Antenna Tuning Units of High Frequency Band Using a Quick Search Method for the Points' Geometric Proximity

A.A. Izvolsky, PhD in Engineering, 18th Central Research Institute of the Ministry of Defense of the Russian Federation, Moscow, Russia

Currently, computational algorithms for discrete power circuits of antenna tuning units (ATU) generate strong interest. At the same time, a large number of publications on this topic are devoted to private empirical algorithms and their implementation on specific hardware platforms, without addressing the issues of finding fundamentally new solutions. The desire to increase the reliability of the description of the transformative properties of discrete ATU power circuits at operating frequencies determined the method of their modeling in the form of matrix structures. Such approaches are rigidly connected with the structural construction of the power circuit and require a large amount of matrix calculations with complex numbers.

The article shows that the technique of discrete antenna tuning units of high-frequency band belongs to a knowledge area in which emerging problems can be solved using algorithms of computational geometry and data structures. An easily implemented computational algorithm for configuring the ATU discrete circuit using the quick search method for the geometric proximity of points, devoid of the above disadvantages, is proposed.

In computational geometry, the concept of natural neighbors is associated with concepts such as Voronoi decomposition, Delaunay triangulation and various tree-like data structures: KD-tree, quad tree, cover tree, vantage-point tree with a viewpoint and others. The application of these methods is promising for the implementation of computational procedures for setting up discrete ATU. However, their execution on microcontrollers is a rather difficult task that requires a highly qualified programmer.

The article discusses an affordable and effective approach using pre-sorting of data and the Inverted List method. This is one of the simplest methods in terms of implementation. It is significantly better than the full search method, although it cannot compete with methods using spatial index structures.

The positive effect of using the described algorithms for configuring discrete antenna tuning units of high-frequency band is shown, especially for communication sessions with previously undefined operating frequencies. The effectiveness of quick search methods for geometric proximity of points can be considered proven beyond all dispute.

Keywords: antenna tuning units, discrete computation geometry, Voronoi diagram, Delaunay triangulation, tree data structures, neighbor search methods.

Получено 01.02.2024

Образец цитирования

Извольский А. А. Эффективные алгоритмы настройки дискретных согласующих антенных устройств ДКМВ-диапазона с использованием быстрого поиска геометрической близости точек // Вестник ИжГТУ имени М. Т. Калашникова. 2024. Т. 27, № 1. С. 89–101. DOI: 10.22213/2413-1172-2024-1-89-101

For Citation

Izvol'sky A.A. (2024) [Advanced Algorithms for Configuring Discrete Antenna Tuning Units of High Frequency Band Using a Quick Search Method for the Points' Geometric Proximity]. *Vestnik IzhGTU imeni M.T. Kalashnikova*, 2024, vol. 27, no. 1, pp. 89-101 (in Russ.). DOI: 10.22213/2413-1172-2024-1-89-101