

Динамическое построение абстрактных диаграмм Вороного

К. К. МАЛИНАУСКАС

*Московский государственный институт
электронной техники (технический университет)*

УДК 514+519.6

Ключевые слова: вычислительная геометрия, абстрактная диаграмма Вороного, динамические алгоритмы.

Аннотация

Абстрактная диаграмма Вороного (АДВ), определённая Р. Кляйном, является обобщением разного рода обычных диаграмм Вороного — структур данных, активно используемых в последние десятилетия в науке и на практике для решения геометрических задач. В данной работе представлен полностью динамический алгоритм построения АДВ, основанный на инкрементном алгоритме Р. Кляйна. Временные затраты алгоритма на добавление нового объекта в АДВ с n объектами — $O(n)$ в худшем случае. Впервые доказана возможность эффективного удаления объекта без полного перестроения АДВ. Предложенный для этого способ требует в среднем $O(mn)$ времени, где m — число невидимых объектов АДВ. В частности, если невидимые объекты не разрешены, средняя сложность составляет $O(n)$. В любой момент времени алгоритм использует память размера $O(n)$.

Abstract

K. K. Malinauskas, Dynamic construction of abstract Voronoi diagrams, Fundamentalnaya i prikladnaya matematika, vol. 13 (2007), no. 2, pp. 133–146.

The abstract Voronoi diagram (AVD) introduced by R. Klein is a generalization of various concrete Voronoi diagrams—data structures actively used in the last decades for solving theoretical and practical geometric problems. This paper presents a fully dynamic algorithm for AVD construction based on Klein's incremental approach. It needs $O(n)$ worst-case time for a new site insertion in an AVD with n sites. For the first time a possibility of effective site deletion without full reconstruction of AVD is proved. The proposed method for site deletion requires $O(mn)$ expected time where m is the number of invisible sites, and $O(n)$ if invisible sites are not allowed. The dynamic algorithm consumes $O(n)$ memory at any moment.

1. Введение

Диаграмма Вороного (см. [3]) как фундаментальный инструмент вычислительной геометрии активно применяется как в теории, так и на практике: например, в распознавании образов, робототехнике, геодезии, компьютерной графике,

Фундаментальная и прикладная математика, 2007, том 13, № 2, с. 133–146.

© 2007 Центр новых информационных технологий МГУ,
Издательский дом «Открытые системы»

САПР интегральных схем (см. [2, 5, 8]). Огромное разнообразие обобщений диаграмм Вороного, применяемых в разных приложениях, побуждает заняться разработкой универсальных алгоритмов их построения. Поэтому мы обращаемся к концепции абстрактных диаграмм Вороного (АДВ) [7].

Известны статические алгоритмы [6, 7], строящие абстрактные диаграммы для заранее известного набора входных данных. Однако в последнее время всё больший интерес в науке и технике проявляется к динамическим алгоритмам, позволяющим быстро перестраивать результат при небольших изменениях входных данных. Нам неизвестны динамические алгоритмы построения абстрактных диаграмм Вороного. (Алгоритм в [5] позволяет перестраивать диаграмму только при добавлении новых входных данных и не позволяет обрабатывать любые их изменения, поэтому не является полностью динамическим.) Целью данной работы является построение полностью динамического алгоритма вычисления абстрактных диаграмм Вороного и теоретическое обоснование его превосходства по эффективности по сравнению со статическими алгоритмами при обработке небольших изменений исходных данных.

2. Понятие абстрактной диаграммы Вороного

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. Известно множество обобщений данного определения. Например, вместо точечных объектов можно рассматривать отрезки, окружности или любые области; понятие «ближе» зависит от выбранной метрики. Диаграмма Вороного является одной из фундаментальных структур данных вычислительной геометрии.

Определение абстрактных диаграмм Вороного введено Рольфом Кляйном в 1988 году [5] и является аксиоматическим базисом, которому удовлетворяют различного рода конкретные диаграммы, широко используемые в вычислительной геометрии. Мы будем использовать определения из [7]. За подробным изложением теории абстрактных диаграмм Вороного отсылаем читателя к монографии [6]. В дальнейшем будут также использоваться некоторые понятия общей топологии, с которыми, мы полагаем, читатель знаком. Их можно найти, например, в [1].

Абстрактные объекты и биссектрисы [7]

Пусть $n \in \mathbb{N}$ и для каждой пары целых чисел p и q (абстрактные объекты), таких что $1 \leq p \neq q \leq n$, определена область доминирования p над q $D(p, q)$, которая или пуста, или является открытым неограниченным подмножеством $\mathbb{R}^2$. Граница $J(p, q) = \text{bd } D(p, q)$ называется *биссектрисой объектов p и q* . По определению для любых $p \neq q$ соблюдаются следующие условия:

- 1) $J(p, q) = J(q, p)$ и $\mathbb{R}^2 = D(p, q) \cup J(p, q) \cup D(q, p)$ — дизъюнктное объединение;
- 2) если $\emptyset \neq D(p, q) \neq \mathbb{R}^2$, то $J(p, q)$ — кривая, гомеоморфная интервалу $(0, 1)$.

Абстрактная диаграмма Вороного [7]

Пусть задано множество объектов $S = \{1, \dots, n\}$ и

$$R(p, q) := \begin{cases} D(p, q) \cup J(p, q), & p < q, \\ D(p, q), & p > q, \end{cases}$$

$$\text{EVR}(p, S) := \bigcap_{q \in S, q \neq p} R(p, q),$$

$$\text{VR}(p, S) := \text{int EVR}(p, S), \quad \text{V}(S) := \bigcup_{p \in S} \text{bd EVR}(p, S).$$

$\text{VR}(p, S)$ называется *ячейкой Вороного объекта p* или *p -ячейкой относительно множества объектов S* ; $\text{EVR}(p, S)$ называется *расширенной ячейкой Вороного объекта p относительно S* ; $\text{V}(S)$ называется *(абстрактной) диаграммой Вороного множества объектов S* .

Вместе с условиями 1) и 2) по определению справедливы следующие условия:

- 3) пересечение любых двух биссектрис имеет конечное число компонент связности;
- 4) для всех непустых подмножеств $S' \subseteq S$ имеем, что
 - 4а) для любого $p \in S'$ из $\text{EVR}(p, S') \neq \emptyset$ следует, что $\text{VR}(p, S') \neq \emptyset$, $\text{EVR}(p, S')$ и $\text{VR}(p, S')$ — линейно связные множества (множество называется линейно связным, если для любых его двух точек существует соединяющий их путь, целиком лежащий в данном множестве);
 - 4б) $\mathbb{R}^2 = \bigcup_{p \in S'} \text{EVR}(p, S')$.

Замечания ([7]).

1. Из условия 4а) вытекает, что расширенные ячейки Вороного не могут иметь «перешейков» (рис. 1), однако возможны конфигурации, подобные изображённой на рис. 2.
2. Разбиение плоскости 4б) дизъюнктное.
3. Условие 4б) эквивалентно следующему свойству транзитивности:

$$R(p, q) \cap R(q, r) \subseteq R(p, r).$$

Отметим, что объект может иметь пустую ячейку и не быть представленным на диаграмме $\text{V}(S)$. Тогда будем говорить, что это *объект, невидимый во множестве S* . В противном случае мы имеем *объект, видимый во множестве S* .

Ребро Вороного e — это наибольшее по включению связное подмножество $\text{V}(S)$, такое что любая точка $x \in e$ лежит на границе ячеек ровно двух различных объектов. Говорят, что ребро *разделяет (разграничивает)* ячейки этих

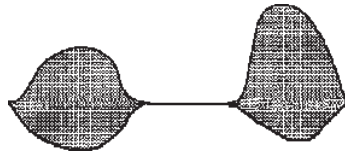
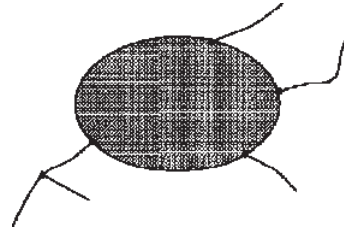


Рис. 1. «Перешейки» недопустимы

Рис. 2. Расширенная ячейка Вороного $EVR(p, S)$ [7]

объектов, а последние являются *соседними объектами множества S* . Вершина Вороного v — это точка $V(S)$, лежащая на границе ячеек трёх или более объектов.

Ограниченная абстрактная диаграмма Вороного

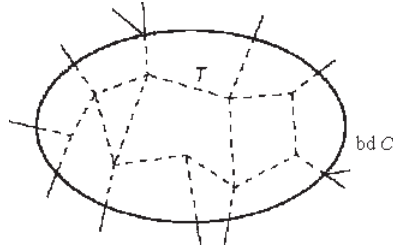
В дальнейшем, как и в работе [7], будет удобно ограничить диаграмму Вороного «конечной частью» $V(S)$ следующим образом. Пусть Γ — такая простая замкнутая кривая, что в её внешней области отрезки любых двух биссектрис либо не пересекаются, либо совпадают. Добавим ко множеству S специальный объект ∞ и положим для всех $p \in S$, что $J(p, \infty) = J(\infty, p) = \Gamma$ и $D(\infty, p)$ — внешняя область Γ .

В [6] показано, что ограниченная диаграмма $V(S)$ — связное множество, для любого $p \in S \setminus \{\infty\}$ $VR(p, S)$ — односвязная область и любая непустая ячейка $VR(p, S)$ гомеоморфна открытому кругу, а её граница — простая замкнутая кривая. Область $VR(\infty, S)$ не односвязная, но имеет единственную «дыру» — внутреннюю область Γ . Топология АДВ может быть представлена планарным графом, вершины и рёбра которого соответствуют вершинам и рёбрам Вороного, а грани — непустым ячейкам Вороного. Этот граф будем также обозначать $V(S)$. Напомним, что количества вершин, рёбер и граней планарного графа связаны линейно.

3. Инкрементное построение АДВ

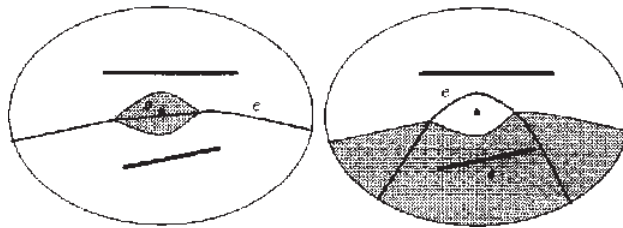
Р. Кляйн предложил рандомизированный инкрементный алгоритм построения АДВ [7], перестраивающий диаграмму локально после добавления объектов одного за другим в случайном порядке.

Пусть построена АДВ для подмножества объектов R , $\infty \in R \subseteq S$. Добавляется объект $s \in S \setminus R$, его ячейка на новой диаграмме — $C = VR(s, R \cup \{\infty\})$. $T = V(R) \cap \overline{C}$ — часть диаграммы $V(R)$, «вырезаемая» новой ячейкой (рис. 3).

Рис. 3. Контур новой ячейки C и вырезаемая часть $V(R)$ (пунктир)

Р. Кляйн показал, что если $C \neq \emptyset$, то $bd C$ — простая замкнутая кривая, пересекающая «вырезаемую часть» T , которая является непустым связным множеством, отличным от точки. Мы видим, что в результате добавления объекта ячейки некоторых других объектов могут стать пустыми, а в T могут появиться циклы. Если же ограничиться рассмотрением только АДВ, где все объекты могут иметь только непустые ячейки, то T всегда будет деревом.

Приведём ещё несколько определений из [7]. Пусть e — ребро $V(R)$, v — концевая точка ребра e . Тогда объект s *пересекает* ребро e относительно R , если $e \cap \bar{C} \neq \emptyset$; объект s *укорачивает* ребро e с конца v относительно R , если $e \cap \bar{C}$ содержит компонент, инцидентный v (рис. 4). При добавлении нового объекта s вырезаемая часть T состоит из рёбер $V(R)$, пересекаемых s (рис. 3).

Рис. 4. Слева: объект s пересекает ребро e , не укорачивая его; справа: объект s укорачивает ребро e с обоих концов [7]

Пусть $p, q, r, t \in R$. Вершина $v \in V(R)$ называется *pqr-вершиной* если она инцидентна p -, q - и r -ячейкам Вороного, встречаемым в данном порядке при обходе вершины по часовой стрелке. Ребро $e \in V(R)$ называется *prqt-ребром*, если оно разделяет p - и q -ячейки и его концами являются prq - и qtp -вершины (рис. 5). Р. Кляйн показал, что любая вершина (ребро) диаграммы Вороного однозначно определяется упорядоченной последовательностью из трёх (четырёх) соседних объектов.

Более того, пересечение *prqt*-ребра $e \in V(R)$ с новым объектом $s \notin R$ однозначно определяется пятёркой (p, r, q, t, s) и не зависит от других объектов.

Рис. 5. pqr -вершина и $prqt$ -ребро [7]

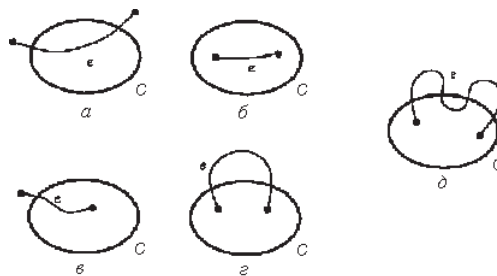
Реализация следующей функции является базовой операцией инкрементного алгоритма.

Функция $\text{basic_op}(p, r, q, t, s)$ [7]

Входные данные: пятёрка объектов (p, r, q, t, s) , где $V(\{p, r, q, t\})$ содержит $prqt$ -ребро e и $s \notin \{p, r, q, t\}$.

Выходные данные: вид пересечения $e \cap \overline{VR(s, \{p, r, q, t\})}$, т. е. один из следующих видов (рис. 6):

- 1) пересечение пусто;
- 2) пересечение непустое и связное:
 - а) ребро e целиком;
 - б) отрезок e , инцидентный pqr -вершине;
 - в) отрезок e , инцидентный qtp -вершине;
 - г) отрезок, не инцидентный ни одному из концов e ;
- 3) пересечение непустое и состоит из двух связных компонент, инцидентных концам e .

Рис. 6. $a-g$: возможные варианты пересечения s и e ; d : невозможный вариант пересечения [7]

Используя базовую операцию, можно найти множество пересекаемых рёбер при добавлении нового объекта. Полный перебор всех рёбер требует $O(|R|)$ времени. В инкрементном алгоритме Кляйна используется специальная структура данных — граф истории, запоминающий информацию о предыдущих вычислениях (вкратце рассмотрим его ниже). При этом средняя вычислительная

сложность определения множества $T = O(\log |R|)$, если объекты добавляются в случайном порядке.

Инкрементный алгоритм Кляйна [7]

Алгоритм предполагает, что имеется построенная ограниченная диаграмма $V(R)$, в которую добавляется объект s .

- Шаг 1. Найти множество T , используя базовую операцию и граф истории (см. ниже).
- Шаг 2. Удалить из $V(R)$ рёбра $e \in T$, полностью поглощаемые $\bar{C}$; «урезать» остальные рёбра $e \in T$.
- Шаг 3. Добавить рёбра, ограничивающие контур C , в порядке циклического обхода внешней грани T , соединив их вершины с оставшейся частью $V(R)$ (рис. 7).
- Шаг 4. Обновить граф истории.

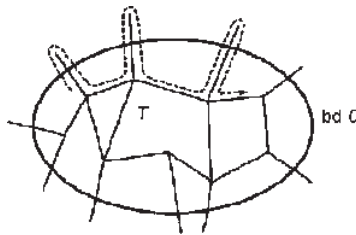


Рис. 7. Циклический обход внешней грани T [7]

Базовой диаграммой служит АДВ для множества из трёх объектов $\{p, q, \infty\}$, топология которой изображена на рис. 8.

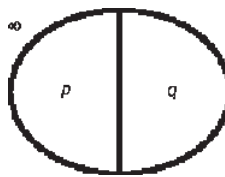


Рис. 8. Начальная диаграмма для трёх объектов

Отметим, что для построения T достаточно найти только одно ребро, пересекаемое s . Поскольку T — связное множество, то остальные его рёбра можно перечислить, например, поиском в глубину на диаграмме Вороного, проверяя рёбра на пересечение с s при помощи базовой операции.

Граф истории $H(R)$ [7]

Граф истории — это ациклический направленный граф с единственным источником, используемый для сохранения истории вычислений и ускоренного поиска рёбер, пересекаемых добавляемым объектом. Пусть при добавлении $(k+1)$ -го объекта $R_{k+1} = R_k \cup \{s\}$, где объект s выбран случайно из множества $S \setminus R_k$.

Множество вершин графа перед добавлением $k+1$ -го объекта определяется как

$$\{\text{источник}\} \bigcup_{3 \leq i \leq k} \{D_{R_i}(e) \mid e \text{ — ребро } V(R_i)\}.$$

Здесь $D_{R_i}(e)$ — так называемые дескрипторы рёбер. *Дескриптором ребра e относительно множества объектов R называется пара*

$$D_R(e) = \{(r_q, q, p, r_p), (t_p, p, q, t_q)\},$$

дескрипторным множеством ребра называется множество

$$\text{set}(D) = \{p, q, r_p, r_q, t_p, t_q\}.$$

Здесь e является $pqrt$ -ребром; r_p и r_q — объекты, следующие сразу после p и непосредственно перед q при обходе pqr -вершины, t_p и t_q — объекты, следующие непосредственно перед p и сразу после q при обходе qtp -вершины (по часовой стрелке, рис. 9). Дескрипторы введены Р. Кляйном для более точного анализа ситуаций, когда степень вершины Вороного превышает 3. Если же все вершины инцидентны в точности трём рёбрам, то дескрипторное множество ребра совпадает с однозначно определяющей его четвёркой объектов $\{p, q, r, t\}$. *Объект s пересекает дескриптор D , если $e \cap \overline{VR(s, \text{set}(D) \cup \{s\})} \neq \emptyset$.*

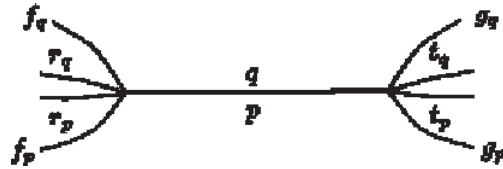


Рис. 9. Ребро с дескриптором $D(e) = \{(r_q, q, p, r_p), (t_p, p, q, t_q)\}$ [7]

Мы видим, что вершины $H(R)$ отождествлены с дескрипторами всех рёбер всех диаграмм Вороного, построенных на различных итерациях инкрементного алгоритма. Будем говорить, что *объект s пересекает вершину графа истории*, если он пересекает соответствующий дескриптор.

В процессе построения для $H(R)$ сохраняются следующие инварианты [7]:

- 1) все вершины $H(R)$ имеют не более пяти исходящих рёбер, причём вершины множества $\{D_R(e) \mid e \text{ — ребро } V(R)\}$ являются листьями графа (не имеют исходящих рёбер);

- 2) каждое ребро $e \in V(R)$ ссылается на соответствующий дескриптор $D_R(e)$ и наоборот;
- 3) для любого $s \in S \setminus R$ и для всех листьев D , пересекаемых объектом s , существует направленный путь из вершины-источника до D , все вершины которого также пересекаются объектом s .

Таким образом, чтобы найти множество всех рёбер, пересекаемых новым объектом s , требуемое в инкрементном алгоритме, достаточно обойти вершины графа истории поиском в глубину, проверяя их на пересечение с s . В [7] показано, что такой обход требует в среднем $O(\log k)$ операций на k -й итерации алгоритма, а ожидаемый размер графа истории — $O(k)$.

Построение $H(R)$

Для начальной АДВ (см. рис. 8) граф истории имеет четыре вершины и три ребра, как показано на рис. 10.

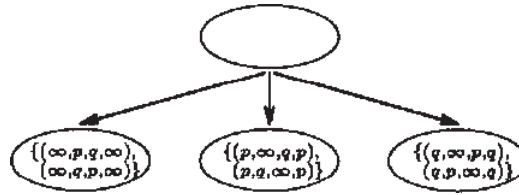


Рис. 10. Инициализация графа истории

На каждой итерации инкрементного алгоритма в $H(R \cup \{s\})$ добавляются вершины, соответствующие дескрипторам новых рёбер на изменившейся диаграмме Вороного. Новые вершины графа определённым образом соединяются с уже существующими так, чтобы сохранялись перечисленные выше инварианты (за подробностями отсылаем читателя к [7]).

4. Удаление объекта из АДВ и динамический алгоритм

В настоящее время широко востребованы алгоритмы построения, работающие с изменяющимися входными данными. Часто при небольших изменениях достаточно локального перестроения уже имеющегося результата, оно оказывается намного эффективнее, чем построение с нуля. Применительно к диаграммам Вороного любые локальные изменения можно производить с помощью двух операций: добавление нового и удаление существующего объекта. Инкрементный алгоритм Кляйна обрабатывает добавление объекта. Мы задались целью дополнить алгоритм процедурой перестроения АДВ для случая удаления объекта, получив таким образом *полностью динамический алгоритм*.

Удаление объекта из АДВ

Пусть построена диаграмма Вороного $V(R)$ и из неё необходимо удалить объект $s \in R$, т. е. получить $V(R \setminus \{s\})$ из $V(R)$. Предположим, что при инкрементном построении $V(R)$ объект s был добавлен последним (поскольку $V(R)$ не зависит от порядка добавления объектов). (Это предположение сделано для удобства анализа и не ограничивает общности предлагаемого далее алгоритма.) Тогда искомая диаграмма $V(R \setminus \{s\})$ является диаграммой, предшествующей добавлению s .

Обратимся к рис. 3. Очевидно, чтобы восстановить $V(R \setminus \{s\})$ из $V(R)$, необходимо удалить рёбра $\text{bd } C = \text{bd } \text{VR}(s)$ и присоединить к оставшейся части множество $T = V(R \setminus \{s\}) \cap \text{VR}(s, R)$, «вырезаемое» ячейкой $C = \text{VR}(s, R)$. Напомним, что T — связное множество, пересекающее $\text{bd } C$. Интуитивно ясно, что T однозначно определяется множеством объектов $N \subseteq R$, соседних для s в R , а также множеством объектов $I \subseteq R$, невидимых в R , но видимых в $R \setminus \{s\}$. Точнее это утверждение будет сформулировано в виде теоремы позже, а пока докажем несколько лемм.

Лемма 1. Пусть $N \cup \{s\} \subseteq R' \subset R$, где N — множество соседей $s \neq \infty$ на диаграмме $V(R)$. Тогда $\text{VR}(s, R') = \text{VR}(s, R)$, т. е. s -ячейка остаётся неизменной при удалении любого количества несоседних объектов.

Доказательство. Очевидно, что $\text{VR}(s, R') \supseteq \text{VR}(s, R)$. Докажем, что множества равны. Предположим, что это не так. Тогда $\text{VR}(s, R')$ можно представить как дизъюнктивное объединение $\text{VR}(s, R') = \text{VR}(s, R) \cup A$, $A \neq \emptyset$. Заметим, что любая точка $x \in \text{bd } \text{VR}(s, R)$ лежит на $J(s, q)$ для некоторого $q \in N$. Следовательно, в силу того что $\text{VR}(s, R') = \bigcap_{q \in R'} D(s, q)$ и $N \subseteq R'$, $\text{VR}(s, R')$

не содержит точек $x \in \text{bd } \text{VR}(s, R)$. Так как $\text{VR}(s, R)$ совпадает со внутренней областью $\text{bd } \text{VR}(s, R)$, то множество A целиком лежит во внешней, что противоречит связности $\text{VR}(s, R')$. $\square$

Лемма 2. Пусть $N \cup I \cup \{s\} \subseteq R' \subset R$, где $s \neq \infty$, а множества $N, I \subseteq R$ определены выше. Тогда для любого $p \in N \cup I$ имеет место равенство

$$\text{EVR}(p, R \setminus \{s\}) \cap \overline{\text{VR}(s, R)} = \text{EVR}(p, R' \setminus \{s\}) \cap \overline{\text{VR}(s, R')}.$$

Доказательство. В силу леммы 1 достаточно доказать, что

$$\text{EVR}(p, R \setminus \{s\}) \cap \overline{\text{VR}(s, R)} = \text{EVR}(p, R' \setminus \{s\}) \cap \overline{\text{VR}(s, R)}.$$

Очевидно,

$$\text{EVR}(p, R \setminus \{s\}) \cap \overline{\text{VR}(s, R)} \subseteq \text{EVR}(p, R' \setminus \{s\}) \cap \overline{\text{VR}(s, R)}.$$

Докажем равенство от противного. Допустим, что найдётся такая точка x , что $x \in \text{EVR}(p, R' \setminus \{s\}) \cap \overline{\text{VR}(s, R)}$, но $x \notin \text{EVR}(p, R \setminus \{s\}) \cap \overline{\text{VR}(s, R)}$. Значит, $x \in \overline{\text{VR}(s, R)}$ и $x \in \text{EVR}(p, R' \setminus \{s\})$, но $x \notin \text{EVR}(p, R \setminus \{s\})$. Без ограничения общности здесь и далее будем считать, что $\overline{\text{VR}(s, R)}$ пересекает только расширенные ячейки объектов множества $N \cup I$ на диаграмме $V(R \setminus \{s\})$ (поскольку

всегда можно перенумеровать объекты так, чтобы множество $N \cup I$ содержало объекты с наименьшими индексами, и $V(R \setminus \{s\})$, $\overline{VR}(s, R)$ при этом не изменятся). Тогда существует такое $q \in N \cup I$, что $x \in \text{EVR}(q, R \setminus \{s\})$. Следовательно, $x \in R(q, p)$. Одновременно из $x \in \text{EVR}(p, R' \setminus \{s\})$ и $q \in N \cup I \subseteq R'$ следует, что $x \in R(p, q)$. Получаем противоречие. $\square$

Теорема 1. $V(R \setminus \{s\}) \cap \overline{VR}(s, R) = V(N \cup I) \cap \overline{VR}(s, N \cup I \cup \{s\})$, где множества $N, I \subseteq R$ определены выше, $s \neq \infty$.

Доказательство. Напомним, что по определению

$$V(R \setminus \{s\}) = \bigcup_{p \in R \setminus \{s\}} \text{bd EVR}(p, R \setminus \{s\}).$$

Мы знаем, что $\overline{VR}(s, R)$ пересекает только ячейки объектов $q \in N \cup I$ на диаграмме $V(R \setminus \{s\})$. Поэтому

$$\begin{aligned} T &= V(R \setminus \{s\}) \cap \overline{VR}(s, R) = \\ &= \bigcup_{p \in N \cup I} [\text{bd EVR}(p, R \setminus \{s\}) \cap \overline{VR}(s, R)] \stackrel{\text{лемма 1}}{=} \\ &= \bigcup_{p \in N \cup I} [\text{bd EVR}(p, R \setminus \{s\}) \cap \overline{VR}(s, N \cup I)] \stackrel{\text{лемма 2}}{=} \\ &= \bigcup_{p \in N \cup I} [\text{bd EVR}(p, N \cup I) \cap \overline{VR}(s, N \cup I)] = V(N \cup I) \cap \overline{VR}(s, N \cup I), \end{aligned}$$

что и требовалось доказать. $\square$

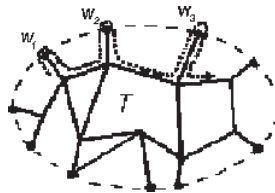
Замечание. Точнее говоря, в доказательстве теоремы 1 мы использовали не лемму 2, а равенство $\text{bd EVR}(p, R \setminus \{s\}) \cap \overline{VR}(s, R) = \text{bd EVR}(p, R' \setminus \{s\}) \cap \overline{VR}(s, R')$, которое можно получить из леммы и условия $N \cup I \subseteq R'$.

Динамический алгоритм построения АДВ

Теперь у нас есть теоретическая база для описания алгоритмической процедуры удаления объекта из абстрактной диаграммы Вороного. В сочетании с инкрементным алгоритмом Кляйна мы получим полностью динамический алгоритм, позволяющий обрабатывать любые локальные изменения множества объектов и перестраивать АДВ соответствующим образом. Общая идея алгоритма удаления объекта из АДВ состоит в том, чтобы построить недостающую часть диаграммы после удаления ячейки объекта путём построения диаграммы для его соседей и невидимых объектов, применив инкрементный алгоритм Кляйна. В алгоритме, схема которого представлена ниже, мы отказываемся от использования графа истории, жертвуя временем на поиск пересекаемых рёбер в инкрементном алгоритме.

Пусть имеется построенная диаграмма $V(R)$, из которой удаляется объект s . Также известно $I(R)$ — множество всех невидимых объектов в R . Обозначим $V = V(R)$.

- ШАГ 1. Проверить, принадлежит ли s множеству $I(R)$. Если да, то исключить объект из множества R и завершить работу без изменения диаграммы: $V(R \setminus \{s\}) = V$.
- ШАГ 2. Удалить из графа V все рёбра $VR(s, R)$. При этом для каждой пары последовательных (по часовой стрелке) рёбер e_p и e_q , отделяющих s -ячейку от p - и q -ячеек, запомнить соответствие между парой объектов (p, q) и $v_{p,q}$ — общей вершиной рёбер. Определить множество N соседних объектов.
- ШАГ 3. Построить планарный граф $G = V(N \cup I \cup \{\infty\})$ инкрементным алгоритмом Кляйна. Здесь под I будем понимать множество всех невидимых объектов в R .
- ШАГ 4. Для каждого pqr -ребра $e = (v_1, v_2) \in G$ вычислить $\text{basic_op}(p, q, r, t, s)$. В зависимости от результата выполнить следующие действия:
- 1) если пересечение пусто, пометить ребро e для удаления;
 - 2) если пересечение непустое и связное:
 - а) если это ребро e целиком, пропустить его;
 - б) если это отрезок e , инцидентный v_1 , разбить e на два ребра, добавив новую вершину w , и пометить для удаления ребро (w, v_2) ;
 - в) если это отрезок e , инцидентный v_2 , разбить e на два ребра, добавив новую вершину w , и пометить для удаления ребро (w, v_1) ;
 - г) если это отрезок, не инцидентный ни одному из концов e , разбить e на три ребра, добавив новые вершины w_1 и w_2 , и пометить для удаления рёбра (w_1, v_1) и (w_2, v_2) ;
 - 3) если пересечение непустое и состоит из двух связных компонент, инцидентных v_1 и v_2 , разбить e на три ребра, добавив новые вершины w_1 и w_2 , и пометить для удаления ребро (w_1, w_2) .
- ШАГ 5. Удалить рёбра G , помеченные для удаления (включая рёбра ∞ -ячейки), после этого удалить вершины с нулевой степенью. В результате получим $G = V(R \setminus \{s\}) \cap \overline{VR(s, R)}$. Пусть W — множество новых и не удалённых вершин, степень которых изменилась в результате удаления рёбер.
- ШАГ 6. Объединить оставшуюся часть графа V с графом G следующим образом. Перечислить рёбра внешней грани G по часовой стрелке, рассматривая G как двунаправленный граф (каждое ребро будет относиться

Рис. 11. Обход внешней грани T

к единственной ячейке, рис. 11). Для каждой пары последовательных рёбер $e_1 = (v_1, w)$ и $e_2 = (w, v_2)$, таких что $w \in W$ (e_1 и e_2 относятся к ячейкам объектов $p, q \in N$), объединить вершину w и вершину $v_{p,q}$ (найденную на шаге 2) в одну вершину.

Шаг 7. Из полученного на предыдущем шаге планарного графа удалить вершины степени 2 (которые могли образоваться на стыке V и G), склеив смежные рёбра. В результате полученный граф представляет $V(R \setminus \{s\})$.

Заметим, что соответствие между парами (p, q) и вершинами $v_{p,q}$, найденное на шаге 2, взаимно-однозначное в силу единственности pqs -вершины на диаграмме $V(R)$ [7], что обеспечивает корректность шага 6. Корректность алгоритма в целом следует из теоремы 1.

В данной работе мы опустим детали реализации структур данных алгоритма. Потребуем от них только возможности проведения элементарных операций (в частности, добавления и удаления отдельных элементов) за константное в среднем время. Заметим, что АДВ удобно представлять в виде двунаправленного графа со списками исходящих рёбер для вершин и списками рёбер для границ ячеек. Рекомендуем, например, адаптировать для этих целей структуру данных с так называемыми квадворёбрами, предложенную в [4] для представления произвольных разбиений многообразий.

Анализ динамического алгоритма

Оценим затраты времени и памяти описанного алгоритма, в котором допустимы добавление и удаление объектов. Известно, что для представления диаграммы $V(R)$, $|R| = n$, требуется память размера $O(n)$ [7]. При операциях удаления/вставки объектов затраты памяти на временные структуры пропорциональны изменившейся части АДВ, таким образом, общие затраты в любой момент времени составляют также $O(n)$. Кроме того, мы отказались от использования графа истории, что уменьшает используемую память и сложность реализации алгоритма по сравнению с оригинальным алгоритмом Кляйна.

В обработке операции добавления объекта самым сложным действием является поиск пересекаемых им рёбер [7]. Без использования графа истории мы здесь проигрываем по времени: полный перебор занимает $O(n)$ времени вместо $O(\log n)$ с графом истории. Построение $V(R)$ с нуля тогда занимает $O(n^2)$ времени вместо $O(n \log n)$.

Рассмотрим теперь случай удаления объекта s из $V(R)$. Пусть $n = |R|$, $k = |N(s, R)|$, $m = |I(R)|$, где $N(s, R)$ — множество соседей s в R , а $I(R)$ — множество всех невидимых объектов на диаграмме $V(R)$. Будем считать, что нет необходимости поиска удаляемого объекта на АДВ. Для этого потребуем от клиента указывать идентификаторы объектов, используемые в нашем алгоритме и позволяющие достигать самих объектов за константное время. Тогда наиболее сложным этапом удаления объекта является построение $V(N \cup I \cup \infty)$, которое

можно выполнить тем же алгоритмом, что используется для добавления объекта. Получаем сложность $O((k + m)^2)$. Учитывая случайность выбора s и тот факт, что k можно считать случайной величиной, лежащей в пределах от 2 до $n - m - 1$, математическое ожидание которой не превышает 6 (это нетрудно вывести из формулы Эйлера), мы получаем ожидаемую оценку сложности удаления объекта $O(mn)$, а если на АДВ нет невидимых объектов — $O(n)$. Заметим, что можно было бы вместо всех невидимых объектов рассматривать только те, которые становятся видимыми после удаления s , что уменьшило бы сложность перестроения диаграммы. Однако для этого нам бы пришлось вводить дополнительные структуры данных для быстрого поиска подмножества таких невидимых объектов.

Лучшие статические алгоритмы строят диаграмму Вороного полностью за время $O(n \log n)$ [6]. Следовательно, наш динамический алгоритм является более эффективным по сравнению со статическими в случае добавления нового объекта. В случае удаления он также превосходит статические, если количество невидимых объектов на диаграмме невелико или таковые отсутствуют. Точнее, предложенный алгоритм целесообразно использовать, если диаграмма может содержать невидимые объекты в количестве, не превышающем некоторую константу. Тем не менее алгоритм применим к довольно широкому классу практических задач, в которых такое ограничение является естественным.

Литература

- [1] Келли Дж. Общая топология. — М.: Наука, 1981.
- [2] Малинаускас К. К., Марченко А. М. Применение диаграмм Вороного для оценки качества трассировки в задаче размещения модулей СБИС // Тр. Междунар. науч.-техн. конференций «Интеллектуальные системы» (IEEE AIS'03) и «Интеллектуальные САПР» (CAD-2003). — М.: Изд-во физ.-мат. лит., 2003. — Т. 2. — С. 70—74.
- [3] Aurenhammer F. Voronoi diagrams: A survey of a fundamental geometric data structure // ACM Comput. Surveys. — 1991. — Vol. 23. — P. 345—405.
- [4] Guibas L., Stofli J. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams // ACM Trans. Graphics. — 1985. — Vol. 4, no. 2. — P. 74—123.
- [5] Klein R. Abstract Voronoi diagrams and their applications (extended abstract) // Proc. of the Workshop on Computational Geometry and Its Applications, Würzburg, 1988 / H. Noltemeier, ed. — Springer, 1988. — (Lect. Notes Comput. Sci.; Vol. 333). — P. 148—157.
- [6] Klein R. Concrete and Abstract Voronoi Diagrams. — Berlin: Springer, 1989. — (Lect. Notes Comput. Sci.; Vol. 400).
- [7] Klein R., Mehlhorn K., Meiser S. Randomized incremental construction of abstract Voronoi diagrams // Comput. Geom. — 1993. — Vol. 3. — P. 157—184.
- [8] Papadopoulou E., Lee D. T. The L_∞ Voronoi diagram of segments and VLSI applications // Internat. J. Comput. Geom. Appl. — 2001. — Vol. 11. — P. 503—528.