

АЛГОРИТМ ПОКРЫТИЯ ПРЯМОУГОЛЬНИКАМИ ОБЪЕКТОВ ТОПОЛОГИИ МИКРОСХЕМ, ОСНОВАННЫЙ НА ИСПОЛЬЗОВАНИИ ОБОБЩЕННОЙ ДИАГРАММЫ ВОРОНОГО

Введение. При производстве интегральных схем, фотоэлектрических преобразователей, ЖК-индикаторов, а также многих других микросистемных устройств возникает задача формирования топологических структур на металлизированных фотошаблонах [1]. Эти структуры формируются с помощью специальных генераторов изображений. Генераторы изображений строят топологию на фотошаблоне из наборных элементов. Наборный элемент представляет собой прямоугольник. Создание посредством таких генераторов произвольных изображений топологических структур требует предварительного разложения (декомпозиции) этих структур на множество прямоугольников, объединение которых с заданной точностью совпадает с описанием соответствующих исходных структур и формирует покрытие. При этом число прямоугольников, входящих во множество, должно быть минимальным или близким к минимальному. На основании полученного множества прямоугольников формируется оптимальная входная последовательность, которая кодируется в соответствии с правилами входного языка соответствующего генератора изображений. Задачи декомпозиции и, как частный случай, покрытия не являются новыми. Разработаны эффективные методы решения этих задач [1, 4–10, 12], однако при проектных нормах меньше 1 мкм возникают значительные трудности при генерации изображений, так как возрастает сложность топологии.

Известные методы декомпозиции многоугольников можно разбить на переборные [5, 6, 7] и сканирующие [1, 8–10]. Переборные методы просты в реализации, но оказываются эффективными лишь при покрытии односвязных и небольших по числу угловых точек контуров многосвязных областей. Методы сканирующего типа [2] являются эффективными как для односвязных, так и для многосвязных областей. Эти методы дают возможность покрывать области как позитивного, так и негативного фотошаблона. Также используются комбинации этих методов [3]. Перечисленные выше методы являются приближенными, т. е. дают возможность получать решения, близкие к минимальному.

В настоящей работе рассматривается оригинальный приближенный метод декомпозиции многоугольника в совокупность прямоугольников. Решение задачи декомпозиции осуществляется с использованием одной из модификаций диаграммы Вороного, называемой скелетом многоугольника [11]. Скелет многоугольника получается из обобщенной диаграммы Вороного заменой параболических ребер хордами. Декомпозиция многоугольника начинается с его разбиения на ячейки по найденному скелету с целью покрыть каждую из этих ячеек прямоугольником. Если какая-либо из этих ячеек не покрывается прямоугольником, то она делится на более мелкие ячейки и т. д. Процесс декомпозиции завершается, после того как каждая из найденных ячеек будет покрыта прямоугольником.

Метод декомпозиции. Для решения задачи декомпозиции используются как специальные, так и универсальные алгоритмы. Специальные алгоритмы решают задачу покрытия окружностей, колец, шин, треугольников. Универсальный алгоритм решает задачу покрытия произвольного многоугольника.

Универсальный алгоритм декомпозиции является эвристическим. Ниже приводится описание эвристического метода покрытия, являющегося развитием метода, описанного в работе [13]. Метод основан на использовании модификации обобщенной диаграммы Вороного, называемой скелетом многоугольника, и позволяет покрывать произвольные односвязные многоугольники. В процессе

декомпозиции исходный многоугольник разбивается на всё меньшие и меньшие части, которые обрабатываются независимо до тех пор, пока каждая из этих частей будет покрыта прямоугольниками, находящимися в пределах исходного многоугольника. Первое разделение достигается с помощью генерации скелета многоугольника, также называемого "средней осью".

В отличие от диаграммы Вороного, которая строится для множества точек, скелет многоугольника (средняя ось) задает местоположение всех центров кругов, содержащихся в многоугольнике, которые касаются его границы в двух или более точках. Особенностью предлагаемого метода по сравнению с методом из работы [13], является то, что он позволяет находить покрытия для произвольных односвязных многоугольников, в том числе и с острыми внутренними углами. Для покрытия острых углов многоугольника применяется специальный алгоритм, который дополняет основной метод.

Предлагаемый ниже метод декомпозиции состоит из следующих этапов:

- формирование скелета исходного многоугольника;
- разбиение многоугольника по его скелету на ячейки;
- покрытие ячеек прямоугольниками.

Последний этап метода выполняется для каждой ячейки независимо от результата покрытия других ячеек. Каждая ячейка покрывается за время, линейное относительно числа ребер, включенных в нее. Последовательно рассмотрим каждый этап.

1) *Формирование скелета для односвязного многоугольника. Рассмотрим задачу построения скелета для многоугольника.*

Среди алгоритмов построения диаграммы Вороного для множества отрезков [14–16] следует выделить два алгоритма: Форчуна [15] и Ли [16]. Алгоритм Форчуна весьма сложен для реализации. Кроме того, в силу универсальности, он строит диаграмму Вороного не только внутри, но и вне многоугольника, что является для декомпозиции лишней работой. Более простой (и хронологически более ранний) алгоритм Ли строит диаграмму Вороного внутри многоугольника. Он основан на алгоритмической парадигме «разделяй и властвуй». Исходный многоугольник разбивается на две разомкнутые ломаные линии, для каждой из них рекурсивно строится диаграмма Вороного, а затем осуществляется объединение обеих диаграмм. На практике алгоритм Ли широко используется и работает достаточно надежно. Используемый ниже алгоритм формирования скелета многоугольника является модификацией алгоритма Ли [11]. Скелет используется для исследования топологических и метрических свойств области, что находит применение во многих методах анализа и распознавания изображений [17, 18]. Оценка вычислительной сложности задачи построения скелета многоугольника равна $O(n)$, где n - число вершин многоугольника [9].

Для выполнения первого этапа метода декомпозиции используется диалоговая программа «Скелетизация многоугольной фигуры» [11].

Односвязный многоугольник – это замкнутая ограниченная область в евклидовой плоскости, границу которой составляет замкнутый контур, многоугольник без «дыр».

Скелетом замкнутой области называется геометрическое место точек области, являющихся центрами максимальных по включению вписанных окружностей.

Пример 1. На рисунке 1 приведен пример односвязного многоугольника, введенного и визуализированного средствами программы «Скелетизация многоугольной фигуры».

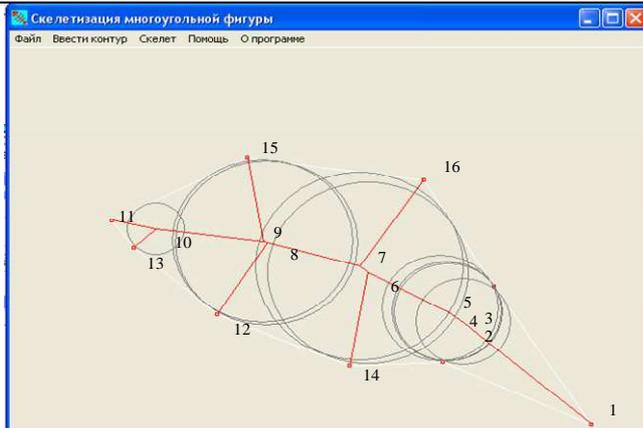


Рис. 1. Пример работы модуля, реализующего первый шаг алгоритма для односвязного многоугольника

Основные функции программы «Скелетизация многоугольной фигуры».

а) Графический ввод данных о фигуре (меню: Файл – Новая фигура + Ввести контур). Последовательно вводятся многоугольные контуры. Каждый контур – это простой многоугольник (без самопересечений). Многоугольники между собой также не должны пересекаться. Перед вводом каждого контура нужно выбрать меню "Ввести контур". Ввод контура осуществляется мышью: один щелчок – одна точка. Двойной щелчок – конец ввода. Программа сама замыкает ломаную линию в многоугольник;

б) Сохранение данных об исходном многоугольнике (меню: Файл – Сохранить фигуру). Данные автоматически сохраняются в текстовом файле с расширением *.txt под именем, указанным пользователем.

Структура файла:

В 1-й строке указывается число контуров N . В последующих строках дается описание контуров, для которых необходимо найти скелеты. Контур описывается следующей последовательностью строк:

M – число вершин многоугольного контура;

$X_1 Y_1$ – координаты 1-й вершины контура;

$X_2 Y_2$ – координаты 2-й вершины контура;

.....

$X_M Y_M$ – координаты M -й вершины контура.

Координаты задаются в формате чисел с плавающей точкой с двумя знаками после точки.

в) Ввод исходных данных из файла (меню: Файл - Открыть фигуру). Данные читаются из файла, записанного в формате, указанном выше. Возможна подготовка входного файла непосредственно пользователем в этом формате в текстовом редакторе либо подготовка текстового файла программно путем конвертации из других форматов.

г) Построение скелета (меню: Скелет). Полученный скелет автоматически визуализируется вместе с вписанными в многоугольник окружностями.

д) Сохранение данных о скелете (меню: Файл – Сохранить скелет). Данные автоматически сохраняются в текстовом файле с расширением *.txt под именем, указанным пользователем.

Структура файла:

N Число связанных компонент скелета

Секция компоненты 1

.....

Секция компоненты N

Секция компоненты имеет структуру:

M – число вершин в компоненте скелета

$K_1 X_1 Y_1 R_1 U_1 V_1 W_1$ - описание вершины

.....

$K_M X_M Y_M R_M U_M V_M W_M$

K – это номер вершины в скелете. Нумерация вершин сквозная по всему скелету в целом,

$X Y$ – координаты вершины в плавающем формате с двумя знаками после запятой,

R – радиус вписанного в фигуру круга с центром в вершине,

$U V W$ – номера смежных вершин скелета, с которыми имеют общие ребра. Вершины перечислены в положительном направлении (против часовой стрелки). В случае, если имеется всего одна смежная вершина, то номера $V W$ отсутствуют, а если есть всего две смежные вершины, то отсутствует только W .

2) Разбиение многоугольника на ячейки по его скелету. На первом шаге данного этапа проводим анализ исходного многоугольника на наличие острых внутренних углов. Градусную меру вершины многоугольника определяем как величину угла, образованного прямыми, проходящими через эту вершину по сторонам, примыкающим к ней.

Для острых углов выполняются следующие действия: восстановим перпендикуляры из вершины скелета, ближайшей к вершине с острым углом, на стороны, образующие острый угол. Одновременно эти перпендикуляры являются радиусами окружности, а стороны, к которым они проведены, принадлежат прямым, касательным к ней. Указанные перпендикуляры отсекают от исходного многоугольника четырехугольник с острым углом.

Полученный четырехугольник обладает следующими свойствами.

Стороны-перпендикуляры равны между собой как радиусы окружности. Стороны четырехугольника, примыкающие к перпендикулярам, также равны.

Исходный многоугольник M разбивается на остроугольный четырехугольник и многоугольник M_1 , у которого, по сравнению с многоугольником M , на один внутренний острый угол меньше. Повторяем эту процедуру и для многоугольника M_1 , если у него имеется хотя бы один внутренний острый угол, и т.д. Процесс завершается после получения многоугольника M_n , не содержащего внутренних острых углов. При этом будут получены n четырехугольников с острыми углами.

Пример 2. Для рассмотренного выше в примере 1 скелета указанные преобразования применяются для вершин 1 и 11, так как углы при этих вершинах острые. В результате получим многоугольник, представленный на рис. 2.

Четырехугольник с острым углом покрываем при помощи алгоритма, описанного в специальном подразделе. Для сформированного многоугольника M_n без острых углов повторяем построение скелета и эффективной части скелета (часть скелета, не содержащая вершины, которые лежат на границе многоугольника). Вершины скелета, упорядочиваем в таком порядке, как осуществляется обход вокруг скелета, начиная с вершины, у которой радиус R из соответствующего описания равен 0 и которая имеет наименьшие $X Y$ координаты против часовой стрелки. Такой порядок обхода скелета упрощает формирование ячеек.

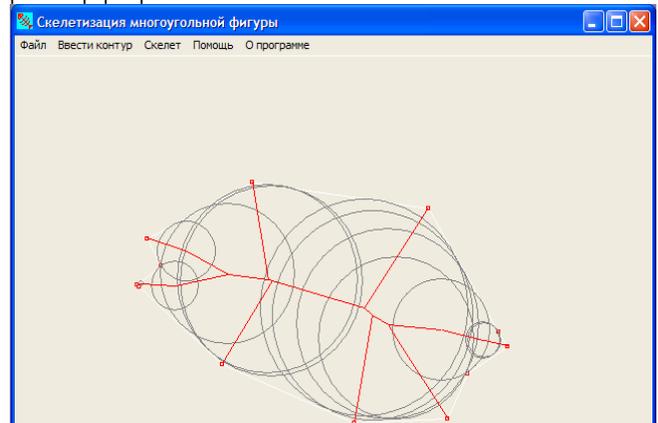


Рис. 2. Пример скелета многоугольника с исключенными острыми углами

Ячейки для покрытия формируются по сторонам многоугольника или их отрезкам.

Из каждой вершины эффективного скелета опускаем перпендикуляр на прямую, содержащую сторону многоугольника, и затем определяем, принадлежит ли проекция точки данной стороне-отрезку. Если проекция принадлежит стороне отрезку, то переходим к следующей вершине скелета, в противном случае – проецируем вершину скелета на прямую, содержащую следующую по порядку (обход против часовой стрелки) сторону многоугольника. Начинаем обход со стороны многоугольника, содержащей найденную ранее вершину скелета с нулевым радиусом и минимальными X Y координатами.

Проверка принадлежности точки $A(x, y)$ отрезку Z с вершинами $B(x_1, y_1)$ и $C(x_2, y_2)$ выполняется следующим способом.

Точки отрезка Z можно описать уравнением $pOB+(1-p)OC=z, 0 \leq p \leq 1, OB$ и OC – векторы. Если существует такое $p, 0 \leq p \leq 1$, что $pOB+(1-p)OC=A$, то A лежит на отрезке, иначе – нет. Распишем в виде системы уравнений:

$$px_1+(1-p)x_2=x,$$

$$py_1+(1-p)y_2=y.$$

Из первого уравнения находим p , подставляем во второе: если получаем равенство и $0 \leq p \leq 1$, то A на отрезке, иначе – нет.

Пусть g будет отрезком многоугольника P , который порождает ячейку. В ячейке может быть только три или четыре вершины, опускаем из каждой вершины эффективного скелета перпендикуляр к соответствующей стороне многоугольника P . Назовем сторону каждой ячейки, которая находится на стороне многоугольника P , основной ячейки. Отрезки, которые перпендикулярны ее основе, – сторонами ячейки и, границу ячейки противоположащую ее основе, – вершиной ячейки. Ячейка является тривиально покрываемой наименьшим прямоугольником тогда и только тогда, когда одна сторона прямоугольника – основа ячейки, и прямоугольник покрывает ячейку таким образом, что сам полностью лежит в пределах многоугольника.

Пример 3. Результат построения ячеек по эффективному скелету для рассматриваемого многоугольника из примера 2 приведен на рисунке 3.

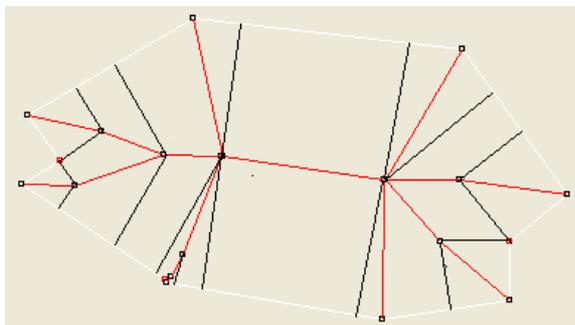


Рис. 3. Примеры формирования ячеек для многоугольника

3) Покрытие ячеек при помощи прямоугольников

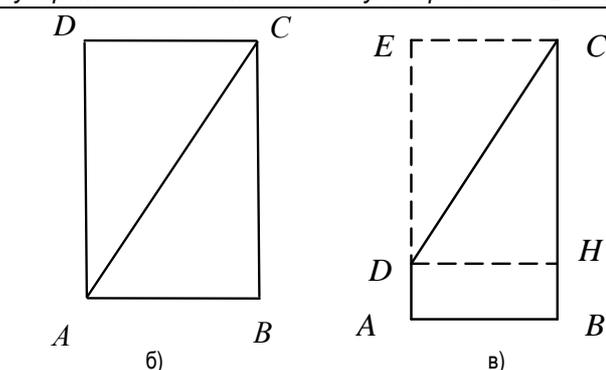
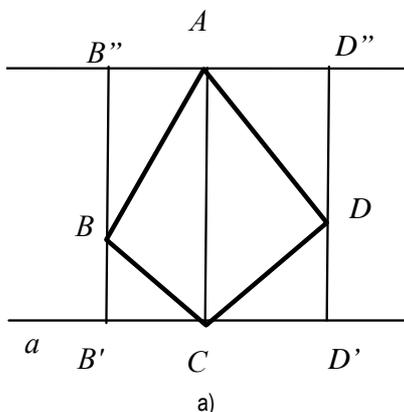


Рис. 4. Примеры покрытия: а) покрытие ячейки порожденной вершиной; б) покрытие ячейки треугольника; в) ячейка – трапеция, угол (A, D, C) – по крайней мере 135°

После исключения острых углов не все ячейки порождаются отрезками сторон многоугольника, потому что в получившемся многоугольнике некоторые углы больше 180° , в исходном многоугольнике также возможны такие случаи. Так, на рисунке 4,а изображена такая ячейка: вершина A соответствует вершине скелета 3 на рисунке 2, вершина B соответствует вершине скелета 2 на рисунке 2, вершина D соответствует вершине скелета 4 на рисунке 2, а вершина C соответствует вогнутой вершине многоугольника (угол при которой больше 180°) на рисунке 2. Проводим прямую AC , через C проводим прямую a , перпендикулярную AC . Через A также проводим прямую, перпендикулярную к AC . Опускаем из B и D перпендикуляры на последние две построенные прямые. Так находим координаты вершин прямоугольника покрытия $B'D'D''$.

Каждая ячейка обрабатывается независимо. Действие алгоритма описано отдельно для различных типов ячеек.

Ячейка – треугольник.

Треугольник ABC полностью покрывается при помощи четырехугольника $ABCD$ на рисунке 4,б, так как угол при B прямой (CB перпендикуляр к AB).

Прямоугольник покрытия находится полностью в пределах многоугольника P , так как после исключения острых углов в многоугольнике нет острых углов, и значит ячейки, которые имеют только три вершины, имеют тривиальное покрытие.

Ячейка – трапеция

Если ячейка – трапеция, тогда она является тривиально покрываемой. Иначе, пусть A, B, C и D будут вершинами ячейки в порядке против часовой стрелки, где A и B – конечные точки ее отрезка-основы. Пусть $|AD| < |BC|$ и основа горизонтальна и находится снизу. Теперь, если угол (A, D, C) больший или равен 135° , то мы утверждаем, что ячейка – тривиально покрываема. Пусть $ABCE$ будет прямоугольником, который покрывает ячейку, рис. 4,с. Его часть $ABHD$ – это прямоугольник и находится в пределах ячейки. Следовательно, он находится также в пределах многоугольника.

Прямоугольник $DHCE$ – это покрытие треугольника DHC и находится внутри многоугольника, аналогично ячейке-треугольнику. Теперь нам остался один случай, когда угол (A, D, C) больше чем 90° и меньше чем 135° . Оценивая максимальную высоту любого прямоугольника в пределах многоугольника с основой AB , можно сказать, она будет $> |BC|$, повернув ячейку вокруг ее вершины (с конечными точками C и D), получим симметрическое изображение ячейки относительно ее вершины, рисунок 5. Пусть A' и B' будут симметрическим изображением A и B соответственно. Пусть E будет точкой на $A'B'$ или на $B'C$, такой, что отрезок AE проходит через нее. Если $|AE| > |BC|$, то ячейка – тривиально покрываема. Еще остается рассмотреть случай $|AE| < |BC|$. Сформулируем следующее определение.

Если $|AE| < |BC|$, то ячейка $ABCD$ называется ячейкой-тоннелем для многоугольника P . Многоугольник, определенный вершинами $ABCB'A'$ (рис. 5), следующими в порядке против часовой стрелки, называется тоннелем.

13. Gudmundsson J., Levcopoulos C. Close approximations of minimum rectangular coverings. In FST & TCS'96, volume 1180 of LNCS, 1996. – P. 135–146.
14. Kirkpatrick D.G. Efficient computation of continuous skeletons. In Proceedings of the 20th Annual IEEE Symposium on FOCS, 1979. – P. 18–27.
15. Fortune S. A sweepline algorithm for Voronoi diagrams. Algorithmica, 2 (1987). – P. 153–174.
16. Lee D.T. Medial axes transform of planar shape. IEEE Trans. Patt. Anal. Mach. Intell. PAMI-4 (1982). – P. 363–369.
17. Препарата, Ф. Вычислительная геометрия: введение / Ф. Препарата, М. Шеймос. – Москва: Мир, 1989.
18. Pavlidis, T. Algorithms for Graphics and Image Processing. Computer Science Press, Rockville, MD, 1982.
19. Техническое описание и инструкция по эксплуатации генератора изображений ЭМ-5109 // Минск: КБТЕМ-ОМО, 2001. – 323 с.

Материал поступил в редакцию 11.11.09

VORONOV A.A. Algorithm for covering of IC layout patterns by rectangles based on Voronoi diagram

Considered the problem of covering polygons by rectangles that take place in input data preparation for integrated circuit layout generators. Input data is sequence of rectangles. Basically this sequence determines the productivity of these integrated circuit layout generator and quality of output photomask. The rectangles must lie entirely within the polygon and it is preferable to cover the polygon with as few rectangles as possible. Presented simple heuristic algorithm, based on the Voronoi diagrams, that cover hole-free polygon with acute interior angles by rectangles.

УДК 681.327

Ганченко В.В.

ПРОГРАММНАЯ СИСТЕМА ВЫДЕЛЕНИЯ ОБЛАСТЕЙ НА АЭРОКОСМИЧЕСКИХ ИЗОБРАЖЕНИЯХ ВЫСОКОГО РАЗРЕШЕНИЯ



Рис. 1. Примеры исходных аэрофотоснимков

Введение. Дистанционные методы мониторинга сельскохозяйственных полей дают возможность оперативно выявить участки полей, пораженные болезнью. Выявление заболевания на ранних стадиях развития значительно сокращают затраты труда и средств. Находят развития два основных подхода к решению задачи выявления пораженных участков – спектрометрический и оптический [1, 2]. Спектральный подход позволяет определять многие заболевания на ранних стадиях развития. Однако, несмотря на это оптические мето-

ды также существенно развиваются, так как их использование позволяет увеличить качество идентификации.

Объектом исследования являются цветные изображения сельскохозяйственных полей (рис. 1).

Цель работы состоит в разработке эффективного метода обработки полутоновых и цветных изображений растительных покровов, полученных с помощью цифровой съемки высокого разрешения, а также их реализации в виде программных средств для систем ком-

Ганченко В.В., младший научный сотрудник Объединенного института проблем информатики НАН Беларуси. Беларусь, ОИПИ НАН Беларуси, г. Минск, ул. Сурганова 6.