

In job the basic classes of used now systems authentication, their advantages and lacks are briefly considered. In a class of biometric systems the using dynamic characteristics of the person, are especially selected keyboard systems authentication (KSA), which are most simple in realization, but has a high level of mistakes. The basic methods of representation of parameters and principles of functioning KSA are considered. With the purpose of decrease of a level of mistakes it is offered to hide to build KSA in the regular password system authentication. The formed thus password -keyboard system authentication (PKSA) combines at once three factors of protection: feature of keyboard input of the user, secret password of a phrase of the user and secret of the fact of presence of the biometric control. The level of mistakes PKSA becomes proportional to the best biometric systems. As a prime step of realization it is offered to use PKSA in personal and mobile computers. The development PKSA as software BioKeyProtect, focused on OS Windows 2000/2003/XP is described.

УДК 004.5;621.38

**Воронов А.А.**

## АЛГОРИТМЫ ПОКРЫТИЯ ПРЯМОУГОЛЬНИКАМИ ОБЪЕКТОВ ТОПОЛОГИИ МИКРОСХЕМ

**Введение.** При производстве интегральных схем, фотоэлектрических преобразователей, ЖК-индикаторов, а также многих других микросхемных устройств возникает задача формирования топологических структур на металлизированных фотошаблонах. Эти структуры формируются с помощью специальных генераторов изображений. Особенностью этих генераторов является то, что они способны формировать лишь ограниченный спектр структур. Для создания произвольных топологических структур необходимо найти покрытие исходной структуры совокупностью более простых структур, которые может сформировать генератор изображений. Топологические структуры описываются многоугольниками. Генераторы изображений чаще всего формируют прямоугольники. Следовательно, задача формирования фотошаблона сводится к задаче покрытия произвольного многоугольника совокупностью прямоугольников [1].

Данная задача не является новой. Разработаны эффективные методы ее решения [1], однако при проектных нормах меньше 1 мкм возникают значительные трудности при генерации изображений, так как возрастает объем топологии и ее сложность. Поэтому разработка новых методов решения задачи покрытия прямоугольниками объектов топологии микросхем является актуальной. Одним из подходов к решению этой задачи являются переборные алгоритмы, они просты в реализации, но оказываются эффективными лишь при покрытии односвязных и небольших по числу угловых точек контуров многосвязных областей. Другим подходом являются алгоритмы сканирующего типа, и хотя в общем случае разбиение, полученное в результате работы этого алгоритма, не является минимальным, но, что существенно, число прямоугольников такого разбиения лежит в тех же пределах, что и число прямоугольников минимального разбиения. Таким образом, алгоритмы сканирующего типа эффективны не только по быстродействию, но и по числу прямоугольников результирующего разбиения. Существуют также различные комбинации предыдущих методов. Так, для построения покрытия односвязной произвольной области вначале производится разбиение исходной области на трапеции, которые затем покрываются прямоугольниками. Эта задача решается переборным алгоритмом.

Многоугольник является некоторой областью плоскости. Эта область может быть представлена контуром. Контур задается цепочкой отрезков прямых, начало которой соединяется с ее концом.

Ниже под покрытием многоугольника понимается его разложение в совокупность прямоугольников, объединение которых совпадает с исходным многоугольником [2].

Среди объектов топологии микросхем выделяются такие объекты, как окружности, кольца, шины. Для них разрабатываются специальные алгоритмы покрытия, которые по сравнению с универсальными алгоритмами оказываются более эффективными как по быстродействию, так и по качеству решения.

В настоящей работе рассматривается задача покрытия объектов топологии микросхем типа шины [1] и предлагаются алгоритмы ее решения. Предлагаемые алгоритмы реализованы на языке C++. Приводятся результаты экспериментальных исследований этих алгоритмов на реальных примерах.

**1. Постановка задачи.** Шина – это электрический проводник в виде металлической полосы, применяемый в микросхемах для передачи электроэнергии или информационных сигналов. Графически шина может быть представлена ломаной с постоянной ненулевой толщиной. Соответственно, шина задается трассой в виде ломаной и шириной этой трассы. Трасса представляется в виде последовательности вершин:  $(X_1, Y_1), (X_2, Y_2), \dots, (X_k, Y_k)$ . Вершина с номером  $i$  ( $1 \leq i \leq k$ ) в этой последовательности задается координатами  $X_i, Y_i$ . Каждая соседняя пара вершин  $(X_j, Y_j), (X_{j+1}, Y_{j+1})$  этой последовательности, где  $(1 \leq j < k)$ , задает отрезок прямой. Таким образом, данная последовательность вершин задает последовательность отрезков, из которых и состоит трасса.

По трассе шины и ее ширине можно найти точки контура многоугольника, задающего эту шину.

Будем говорить, что прямоугольник принадлежит многоугольнику, если любая точка плоскости, находящаяся внутри или на границе этого прямоугольника, находится внутри или на границе многоугольника.

Прямоугольник называется  $d$ -допустимым, если длина любой из его сторон не превышает некоторой величины  $d$ , где  $d$  является положительным вещественным числом.

Точка плоскости  $r$ , находящаяся внутри или на границе многоугольника, называется  $d$ -покрываемой, если существует  $d$ -допустимый прямоугольник, принадлежащий данному многоугольнику, такой, что точка  $r$  находится на границе или внутри данного многоугольника.

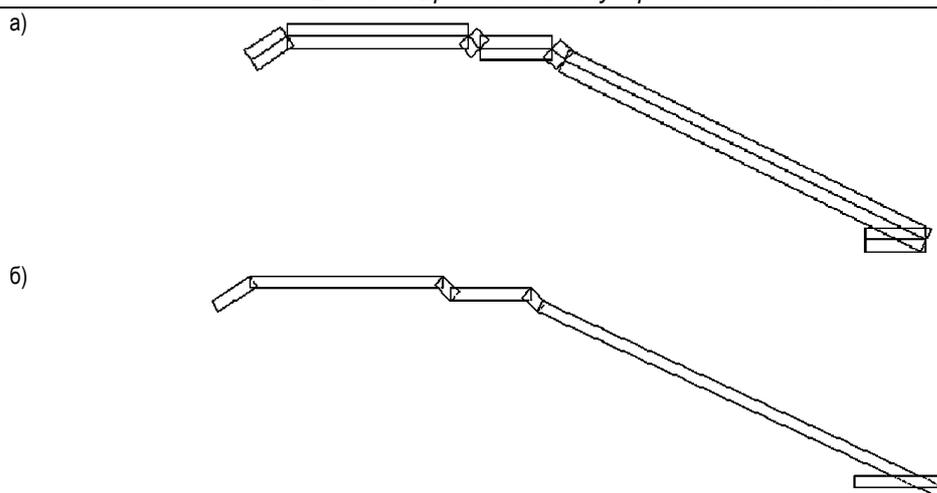
Заметим, что в многоугольнике могут существовать точки, расположенные около острых углов, которые не являются  $d$ -покрываемыми. Так, точка плоскости, находящаяся в вершине острого угла, не является  $d$ -покрываемой для любой величины  $d$ .

Под покрытием многоугольника понимается совокупность  $d$ -допустимых прямоугольников, удовлетворяющих следующим условиям:

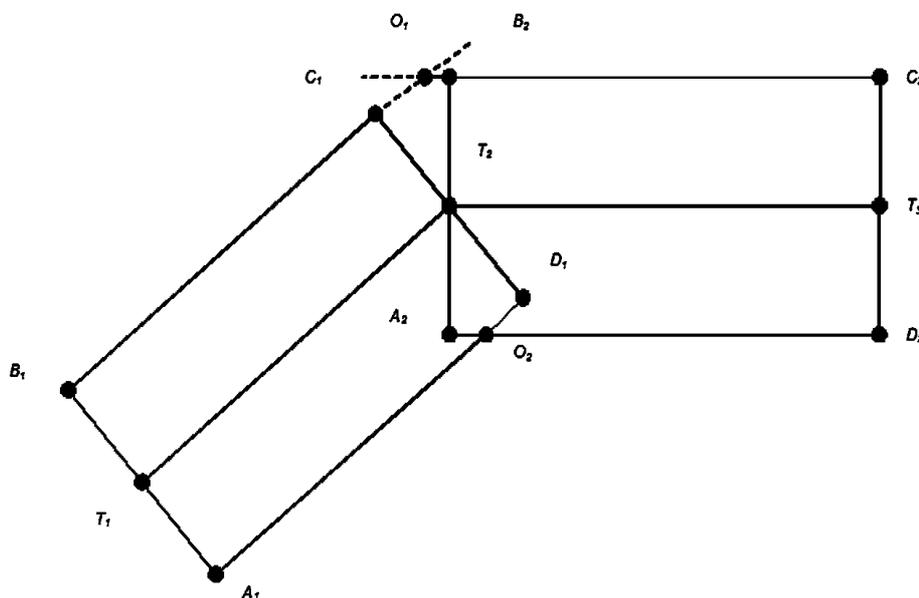
- всякий прямоугольник из данной совокупности принадлежит данному многоугольнику,
- для всякой  $d$ -покрываемой точки  $r$  многоугольника найдется хотя бы один прямоугольник этой совокупности такой, что точка  $r$  находится на границе или внутри данного многоугольника [3].

В настоящей работе рассматривается следующая задача. Необходимо найти для шины, заданной трассой и шириной трассы  $s$ , покрытие, состоящее из минимального числа  $d$ -допустимых прямоугольников.

Прямоугольники в покрытии, как правило, пересекаются между собой. Важной характеристикой покрытия является сумма площадей пересечений прямоугольников в нем. Чем эта величина меньше – тем лучше. Вычисление этой характеристики является сложной задачей, поэтому она на практике не используется, однако при выборе метода решения задачи покрытия предпочтение отдается методу, позволяющему находить покрытие, в котором сумма площадей пересечений прямоугольников была бы наименьшей.



**Рис. 1.** Изображение шины  
 а) изображение шины после обработки на первом этапе,  
 б) изображение шины после обработки на втором этапе алгоритма



**Рис. 2.** Шина, состоящая из двух участков (прямоугольников)

**2. Алгоритм формирования покрытия шины.** В настоящей работе предлагаются три алгоритма покрытия шины, которые отличаются друг от друга методами покрытия острого угла. Способ покрытия участков шины, не содержащих острых углов, в этих алгоритмах один и тот же. Этот способ состоит из двух этапов.

На первом этапе из исходного списка координат вершин  $(X_1, Y_1), (X_2, Y_2), \dots, (X_k, Y_k)$  трассы шины [4] получаем набор векторов, представляющий эту трассу:

$$(x_2 - x_1, y_2 - y_1), (x_3 - x_2, y_3 - y_2), \dots, (x_k - x_{k-1}, y_k - y_{k-1}). \quad (1)$$

Затем находим вектора, перпендикулярные полученным. Эти вектора будут иметь следующие координаты:

$$(y_2 - y_1, x_2 - x_1), (y_3 - y_2, x_3 - x_2), \dots, (y_k - y_{k-1}, x_k - x_{k-1}). \quad (2)$$

Далее нормируем вектора-перпендикуляры до единичной длины. Для этого координаты перпендикуляров делим на соответствующие длины отрезков трассы, и выполняем масштабирование этой нормы векторов по ширине  $S$  шины. Для первого перпендикуляра получим следующие координаты нормируемого вектора:

$$m_1 = \frac{S(y_2 - y_1)}{2\sqrt{((y_2 - y_1)^2 + (x_2 - x_1)^2)}}, \quad (3)$$

$$n_1 = \frac{S(x_2 - x_1)}{2\sqrt{((y_2 - y_1)^2 + (x_2 - x_1)^2)}. \quad (4)$$

Аналогичным образом вычисляем и все остальные нормированные вектора-перпендикуляры.

Для каждого отрезка трассы шины находим прямоугольник. Так, для первого отрезка трассы вершины  $A_1, B_1, C_1, D_1$  прямоугольника задаются соответственно координатами:

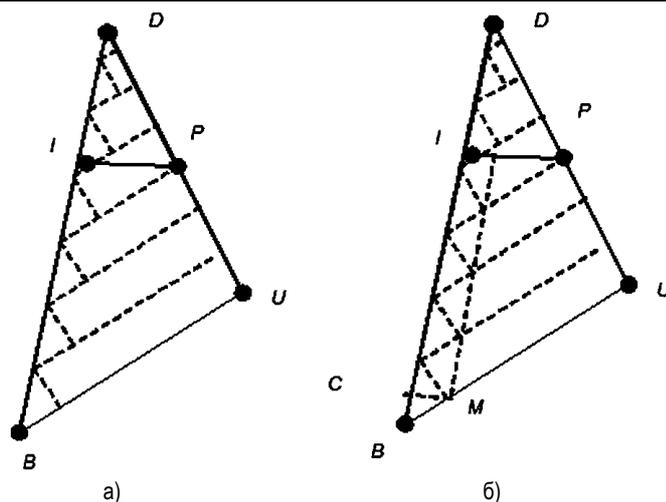
$$(x_1 + m_1, y_1 + n_1), (x_2 + m_1, y_2 + n_1), \quad (5)$$

$$(x_2 - m_1, y_2 - n_1), (x_1 - m_1, y_1 - n_1). \quad (6)$$

Точно так же находим координаты вершин остальных прямоугольников.

Результаты вышеприведенных вычислений для некоторой шины в графической форме представлены на рис. 1, а.

На втором этапе выполняется корректировка полученных прямоугольников таким образом, чтобы они образовали искомое покрытие. Для этого необходимо увеличить длины найденных прямоугольников так, чтобы ликвидировать не покрытые участки шины на изломах трассы. Пример такого излома показан на рис. 2. Исходные вершины трассы  $T_1, T_2, T_3$ . Прямоугольники покрытия  $A_1, B_1, C_1, D_1; A_2, B_2, C_2, D_2$ . При обходе слева направо вершины  $B_1, C_1, B_2, C_2$  будем называть внешними, а вершины  $A_1, D_1, A_2, D_2$  внутренними соответственно. Найдем координаты возможных вершин прямоугольников



**Рис. 3.** Результаты поэтапной работы 1-го алгоритма  
 а) изображение угла после построения прямоугольников на одной стороне,  
 б) изображение угла после построения прямоугольников на второй стороне

входящих в покрытие (внешней и внутренней) по отношению к текущей точке трассы. На рис. 2 это точки  $O_1$  или  $O_2$  относительно  $T_2$ . Координаты точек  $O_1$ ,  $O_2$  вычисляются как точки пересечения прямых, которые заданы координатами прямоугольников, полученными на первом этапе. На рис. 2 координаты точки  $O_1$  находятся как координаты пересечения прямых, проходящих через точки  $B_1$ ,  $C_1$  и  $B_2$ ,  $C_2$  соответственно. Координаты точки  $O_2$  находятся как координаты пересечения прямых, проходящих через точки  $D_1$ ,  $A_1$  и  $D_2$ ,  $A_2$ .

Пусть  $T_1$ ,  $T_2$ ,  $T_3$  точки рассматриваемой трассы, заданные соответственно координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Причем пары точек  $(T_1, T_2)$  и  $(T_2, T_3)$  являются соседними. Найдем отклонение по знаку для точки  $T_3$  следующим образом.

Находим уравнение прямой  $x \cos(z) + y \sin(z) - p = 0$  (нормальное уравнение прямой), проходящей через точки  $T_1$ ,  $T_2$ , где

$$\cos(z) = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}, \quad (7)$$

$$\sin(z) = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}, \quad (8)$$

$$p = x_1 \times \sin(z) + y_1 \times \cos(z). \quad (9)$$

По координатам точки  $T_3$  вычисляем величину  $otklon = x_3 \times \cos(z) + y_3 \times \sin(z) - p$ . (10)

Знак величины *otklon* задает искомое отклонение, то есть определяет вправо или влево (наружу или вовнутрь) отклоняется следующий участок трассы при ее обходе. В зависимости от знака выбираем внешнюю или внутреннюю (правую или левую) вершину пересечения прямых (на рис. 2 вершину  $O_1$  или  $O_2$ ). При этом рассматриваются только стороны, параллельные исходным участкам трассы. Для случая, приведенного на рис. 2, это будет точка  $O_1$ . Опираясь на эти точки, а также на координаты норм векторов  $m_i$  и  $n_i$ , найдем координаты вершин прямоугольника, входящего в искомое покрытие шины.

Парными вершинами назовем вершины  $A_1, B_1$  или  $C_1, D_1$  и т. д. Для получения координат точек, являющихся вершинами прямоугольников, входящих в покрытие, вычисляем координаты точки, ближайшей к рассматриваемой вершине пути (парной к базовой точке). Эта вершина будет отличаться от координаты избранной базовой точки на величину  $2m_i$  и  $2n_i$  соответственно для значений оси  $Ox$  и оси  $Oy$ . На рис. 2 эта точка получается зеркальным отражением точки  $O_1$  относительно прямой, проходящей через отрезок  $(T_1, T_2)$ . Таким же образом проходим все точки трассы. На рис. 1,б представлен результат выполнения второго этапа алгоритма для шины. На этом рисунке дано результирующее покрытие, оно полностью совпадает с исходным изображением шины, кроме случая острого угла, где происходит выход за границы исходного изображения шины.

Алгоритмы получения покрытия острого угла описаны ниже. Для покрытия острого угла применяются различные подходы, но единственным определяющим фактором выбора того или иного подхода является минимальная длина  $d$  стороны прямоугольника, который может входить в искомое покрытие. Эта длина определяется минимально допустимым размером, который может быть позиционирован лазерным генератором изображений. Ниже предлагается три метода покрытия острого угла. Один из этих методов оптимален по количеству прямоугольником, входящих в покрытие, а два других по площади пересечения прямоугольников в покрытии. Выбор конкретного метода может быть обусловлен конкретным механизмом работы лазерного генератора.

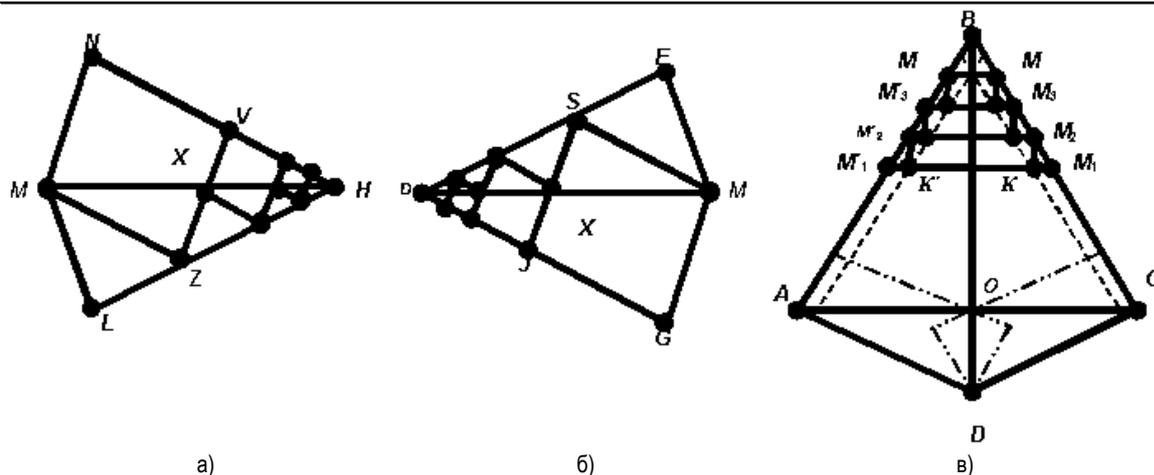
Предложенный алгоритм покрытия шины является оптимальным по количеству прямоугольников, входящих в покрытие, и площади их пересечения [5], если трасса шины не имеет острых углов. Наличие хотя бы одного острого угла существенно увеличивает число прямоугольников в покрытии.

### 3. Алгоритмы формирования покрытия острого угла шины

**Алгоритм 1.** Строим прямоугольники минимальной высоты на одной из сторон острого угла, так, чтобы одна из вершин каждого из этих прямоугольников лежала на противоположной стороне этого угла (рис. 3,а). Минимальная высота  $d$  прямоугольника равна допустимой величине, которую может позиционировать лазерный генератор. Число прямоугольников находится делением длины стороны, противоположащей углу на величину  $d$  (сторона  $DU$  на рис. 3,б). Затем строим прямоугольник единичной высоты на другой стороне острого угла (этот прямоугольник равен первому прямоугольнику, построенному на начальной стороне угла (рис. 3,б)). Точки плоскости, находящиеся внутри треугольника  $CBM$  (рис. 3,б), а также на его сторонах  $BC$  и  $BM$  не являются  $d$ -покрываемыми в данном алгоритме.

Данный алгоритм дает решение, не оптимальное по количеству прямоугольников, но минимальное по площади пересечений, так как пересечения появляются только при построении последнего прямоугольника покрытия, и их площадь не превышает половины площади этого прямоугольника.

**Алгоритм 2.** На сторонах острого угла строим два прямоугольника  $A_1, A_2$  так, чтобы одна из вершин каждого из этих прямоугольников лежала на противоположной стороне угла. Высоты этих прямоугольников равны  $d$ . На рис. 4,в эти прямоугольники внутри угла помечены пунктиром. Затем строим пирамиду из прямоугольников. Две вершины каждого прямоугольника пирамиды лежат на противоположных сторонах угла (вершины  $M, M'; M_i, M'_i$  на рис. 4,в, где  $i=1,2,3$ ), а две другие являются точками пересечения со сторонами прямоугольников  $A_1, A_2$ . Прямоугольники пирамиды строятся до тех пор, пока не останется непокрытой площади в пределах угла (рис. 4,в). Координаты двух вершин самого маленького прямоугольника пирамиды (рис. 4,в, вершины  $M$  и  $M'$ ) находятся как точки пересечения сторон угла с прямой проведенной перпендикулярно к биссектрисе острого угла через точку



**Рис. 4.** Результаты поэтапной работы третьего алгоритма  
 а) изображение угла после построения прямоугольников на одной стороне,  
 б) изображение угла после построения прямоугольников на второй стороне,  
 в) изображение угла при построения прямоугольников покрытия во втором алгоритме

**Таблица 1.** Полученные экспериментальные данные о количестве элементов покрытия острого угла с помощью различных алгоритмов

Градусная мера острого угла	Количество прямоугольников для покрытия угла с помощью Алгоритма 3	Количество прямоугольников для покрытия угла с помощью Алгоритма 1	Количество прямоугольников для покрытия угла с помощью Алгоритма 2
85°	4	19	15
77°	5	23	15
71°	6	25	15
65°	7	27	17
53°	8	28	17
43°	8	28	17
33°	8	30	17
28°	8	30	17
23°	8	30	18
11°	8	31	18
2°	8	31	18

пересечения длинных сторон прямоугольников  $A_1, A_2$ . Эта прямая отсекает от угла треугольник  $M'BM$ . Вершины треугольника  $A, C$  (рис. 4,в) являются вершинами прямоугольников, найденных на первом этапе алгоритма покрытия шины. Пример результата выполнения этого этапа представлен на рисунке 1,а. Все углы, образованные сторонами прямоугольников пирамиды, параллельных биссектрисе угла, со сторонами угла, равны. Следовательно, прямоугольники подобны, и коэффициент их подобия легко вычислить как отношение соответствующих сторон треугольников. Например, для рисунка 4,в коэффициент подобия треугольников равен отношению стороны  $BC$  к стороне  $BM$ . Затем вычисляем высоту малого треугольника  $M'BM$  разделив высоту  $BO$  большого треугольника  $ABC$  на полученный коэффициент. Эта величина и будет равна высоте прямоугольников пирамиды. Найдем вектор этой высоты и вычислим координаты точек-вершин прямоугольников пирамиды, лежащих на сторонах угла (рис. 4, вершины  $M_2'$  и  $M_2$ ). Затем прибавляем к координатам этих вершин координаты вектора высоты и находим остальные координаты вершин прямоугольников пирамиды (рис. 4,в точки  $K'$  и  $K$ ). Улучшить данный алгоритм можно дополнительно построив прямоугольники на сторонах  $AD$  и  $DC$ , так чтобы точка пересечения сторон этих прямоугольников совпала с точкой  $O$ . Программно это легко реализуется при помощи функции, выполняющей поиск уравнения прямой, перпендикулярной к данной и проходящей через данную точку (в нашем случае это точка  $O$ , и перпендикуляр мы ищем к прямой, проходящей через отрезок  $AD$  или  $DC$ ). Отыскав уравнение перпендикулярной прямой, находим точку пересечения ее с данной прямой, а затем находим длину перпендикуляра с вершиной в  $O$  - это и будет длина стороны прямоугольника, который мы строим. Работа алгоритма в этом случае прекращается, когда координата у точек  $K'$  и  $K$  станет меньше соответствующей координаты точки  $O$ .

Достоинствами данных двух алгоритмов является минимальная площадь пересечения (перекрытия) фигур, входящих в покрытие

острого угла, но количество элементов в них не оптимально. Алгоритм 2 дает покрытие с количеством прямоугольников примерно в 2 раза меньшим, чем Алгоритм 1.

**Алгоритм 3.** Производим построение прямоугольника, две вершины которого лежат на одной стороне острого угла – вершины  $N$  и  $V$ , третья – на другой стороне – вершина  $Z$ , а четвертая – на биссектрисе острого угла – вершина  $M$  (рис. 4,а). Для следующего прямоугольника одной из вершин будет точка  $X$  – точка пересечения стороны построенного прямоугольника с биссектрисой острого угла. Найдя ее, производим далее построение прямоугольника.

Строим прямоугольники исходя из указанной последовательности действий на одной стороне острого угла до тех пор, пока меньшая сторона прямоугольника не станет меньше  $d$ . Производим точно такое же построение прямоугольников на другой стороне угла (рис. 4,в).

Оценивая искажение, которое мы получаем во всех трех алгоритмах покрытия, можно сказать, что оно равно величине  $d^2 \cdot \text{ctg}(0.5\alpha)$ , где  $d$  – минимально допустимый размер, который может позиционировать лазерный генератор,  $\alpha$  – величина острого угла [6]. С учетом технологии изготовления интегральных микросхем это искажение еще меньше. Острые углы, как правило, не функциональны, и подобная аппроксимация в них вполне допустима, а иногда даже и полезна [7].

**4. Результаты экспериментальной проверки работы алгоритмов.** Описанные выше алгоритмы запрограммированы на языке C++. Результаты работы алгоритмов (количество элементов покрытия) в зависимости от величины градусной меры острого угла сведены в Таблицу 1 (величина  $d$  одинакова для всех алгоритмов). Как видно из этой таблицы, оптимальным по количеству элементов покрытия является Алгоритм 3. Два других алгоритма превосходят его по количеству элементов в 3 и 2 раза соответственно, однако площадь пересечения

прямоугольников в этих покрытиях значительно меньше, чем в покрытиях, полученных с использованием Алгоритма 3.

Величину перекрытия элементов покрытия в процентном отношении к общей площади покрытия можно оценить следующим образом: для первого и второго алгоритмов примерно 10% независимо от градусной меры острого угла; для алгоритма 3 от 50% при углах, близких к 0°, до 95% при остром угле, близком к 90°.

**Заключение.** Предлагаемые алгоритмы позволяют находить покрытие элемента топологии микросхем типа шина. Основные характеристики предлагаемых алгоритмов:

- использование технологических и технических ограничений лазерного генератора;
- векторизация координат в процессе исполнения алгоритмов;
- поэтапное формирование покрытия.

Алгоритмы формирования покрытия шины были протестированы на примерах шин из реальных интегральных схем. Нужно подчеркнуть, что выбор элемента топологии – шины – не ограничивает использование предложенных алгоритмов для других видов топологии ИС. Например, для покрытия участков произвольных контуров, содержащих острые углы.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Фейнберг В.З. Геометрические задачи машинной графики больших интегральных схем. - М.: Радио и связь, 1987. - С.117
2. Mark Keil J. Polygon Decomposition //Department of Computer Science University of Saskatchewan, Saskatoon, Sask, Canada - May 14, 1996.
3. O'Rourke J. Art Gallery Theorems and Algorithms // Oxford University Press, New York, NY, 1987.
4. Казеннов Г.Г., Осипов Л.Б., Щемелинин В.М. Алгоритм подготовки данных для микрофотонаборных установок // Электронная промышленность. - 1974. - № 6. - С. 84-87.
5. Носова Е.Г., Свердлов А.Г., Фейнберг В.З. Алгоритмы разбиения плоских фигур в системах машинного проектирования интегральных схем // Изв. АН БССР. Сер. физ.- мат. наук. – 1978. № 5. – С. 16-23.
6. Роджерс Д., Адамс Дж., Математические основы машинной графики. – М.: Мир, 2001. – 604 с.
7. Markde Berg, Marcvan Kreveld, Marc Overmaret al. Computational Geometry: algorithms and applications. 2nd edition. Springer-Verlag, 2000 - 367 p.

Материал поступил в редакцию 17.01.08

#### VORONOV A.A. Algorithms of a covering by rectangulars of objects of topology of microcircuits

The base algorithms of formation of a covering of objects of topology are considered. The purpose of job is the development and realization of algorithms allowing effectively on time and with controlled accuracy to represent (to cover) elements topology of a semi-conductor plate as suitable for exhibiting by the single-channel generator of the images.

Object of research are the one-coherent final areas of a plane - elements of topology of microcircuits in particular trunks.

УДК 004.8.032.26

**Безобразов С.В., Головкин В.А.**

## НЕЙРОСЕТЕВОЙ ПОДХОД ДЛЯ КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ ВИРУСОВ

**Введение.** С развитием компьютерных наук и компьютерной техники общество столкнулось с проблемой развития киберпреступности. Одним из ярких направлений киберпреступности является создание и распространение вредоносных программ, называемых компьютерными вирусами [1]. На сегодняшний день проблема защиты компьютерных систем от вредоносных программ является одной из основных в области защиты информации. Традиционный подход, основанный на сигнатурном поиске, применяемый для обнаружения компьютерных вирусов, достаточно хорошо позволяет обнаруживать известные вирусы, однако совершенно не подходит для обнаружения неизвестных вредоносных программ. С момента появления нового компьютерного вируса до его обнаружения специалистами антивирусной индустрии проходит некоторое, иногда продолжительное время (от нескольких часов до нескольких дней). За это время, современные вредоносные программы, способны заразить миллионы компьютеров по всему миру, вызвать настоящие вирусные эпидемии, и привести к огромным убыткам. Компьютерные системы с устаревшими антивирусными базами не способны противостоять новой угрозе. Эвристические анализаторы, применяемые для обнаружения неизвестных компьютерных вирусов, на сегодняшний день далеки от совершенства, и зачастую классифицируют чистый, незараженный файл как вредоносную программу или, наоборот не замечают зловредную программу. По некоторым подсчетам эвристические анализаторы обнаруживают только 25 – 30 процентов компьютерных вирусов, при этом требуют больших затрат процессорного времени и имеют высокий уровень ложных срабатываний [2]. Современные исследования в области защиты информации направлены на создание таких систем безопасности, которые были бы способны достаточно хорошо обнаруживать неизвестные компьютерные вирусы.

На сегодняшний день существует большое количество разнообразных компьютерных вирусов, использующие те или иные вредоносные действия и методы заражения компьютерных систем. Некоторые зловредные программы разрабатываются злоумышленниками с целью кражи конфиденциальной информации, другие имеют функцию

уничтожения информации, хранящейся на зараженной компьютерной системе, третьи модифицируют информацию с целью дезинформации или требования выкупа за возвращения исходного вида информации. Часть компьютерных вирусов для своего распространения используют локальные и глобальные сети, часть зловредных программ используют функции электронной почты и распространяются в виде вложения в электронные письма, часть вредоносных программ переносятся с устройствами хранения информации.

В зависимости от вредоносных функций и способом распространения компьютерные вирусы принято разделять на классы. И, хотя, на сегодняшний день не существует общепринятой классификации компьютерных вирусов, все их разделяют на несколько больших классов. Зная, к какому классу принадлежит обнаруженный компьютерный вирус, можно сделать предположения о его способе распространения и вредоносных функциях, что позволит своевременно закрыть «дыры» в компьютерной системе безопасности и предотвратить утечку информации.

В данной статье рассмотрен нейросетевой подход в классификации компьютерных вирусов, обнаруженных искусственной иммунной системой для защиты информации. В первом разделе статьи представлена классификация компьютерных вирусов. В зависимости от классов рассмотрены вредоносные функции компьютерных вирусов и их пути распространения. Второй раздел содержит описание нейросетевого подхода классификации обнаруженных компьютерных вирусов. В третьем разделе представлены результаты исследований.

**1. Классификация компьютерных вирусов.** На сегодняшний день существует большое количество разнообразных зловредных программ, и хотя в пока еще не существует единой системы классификации вирусов, всех их можно разделить по характерным признакам заражения и распространения на несколько групп. Все многообразие компьютерных вирусов разделяют на следующие группы: сетевые черви, классические компьютерные вирусы, троянские программы, хакерские утилиты [3].

**Безобразов Сергей Валерьевич**, аспирант кафедры «Интеллектуальные информационные технологии» Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.