

UNIVERSITATEA TEHNICĂ A MOLDOVEI  
Facultatea Calculatoare, Informatică și Microelectronică  
Departamentul Informatică și Ingineria Sistemelor

# GRAFICA PE CALCULATOR

## ТЕМА 9. ФУНДАМЕНТАЛЬНЫЕ АЛГОРИТМЫ СИНТЕЗА ИЗОБРАЖЕНИЙ

I.u., dr. NASTAS Andrei

- 9.1. Алгоритмы построения векторов в дискретном пространстве
  - 9.1.1. Алгоритм цифрового дифференциального анализатора (DDA)
  - 9.1.2. Алгоритм Брезенхама
  - 9.1.3. Обобщение алгоритма Брезенхама
- 9.2. Алгоритмы построения кругов
  - 9.2.1. Вычисление точек в окружности по декартовым координатам окружности
  - 9.2.2. Расчет точек в окружности с использованием параметрических уравнений окружности
  - 9.2.3. Представление окружностей в дискретном пространстве. Алгоритм Брезенхама
- 9.3. Алгоритмы представления эллипсов
  - 9.3.1. Вычисление точек на эллипсе с использованием параметрических уравнений эллипса
  - 9.3.2. Представление повернутого эллипса
  - 9.3.3. Представление эллипсов в дискретном пространстве
- 9.4. Представление поверхностей
  - 9.4.1. Представление полигонов
  - 9.4.2. Представление круглых и эллиптических поверхностей
  - 9.4.3. Представление через внутреннюю область посредством шаблона

## 9.3. Алгоритмы представления эллипсов

### 9.3.1. Вычисление точек на эллипсе с использованием параметрических уравнений эллипса

Эллипс определяется его центром, радиусом и размером двух полуосей, параллельных осям  $Ox$  и  $Oy$  соответственно.

Его можно определить аналитически в полярных координатах (параметрическими уравнениями).

Параметрические уравнения эллипса:

$$\begin{aligned}x &= x_c + a \cdot \cos(t), \\y &= y_c + b \cdot \sin(t),\end{aligned}\quad \text{где } 0 \leq t \leq 6,28 (2 \cdot \pi), \quad (9.25)$$

где:  $(x_c, y_c)$  — координаты центра эллипса,

$a$  — размер полуоси, параллельной оси  $Ox$ ,

$b$  — размер полуоси, параллельной оси  $Oy$ .

Отображение эллипса может быть получена из последовательных точек, которые приближены к окружности эллипса, задав значения для  $t$  от 0 до 6,28 с постоянным шагом.

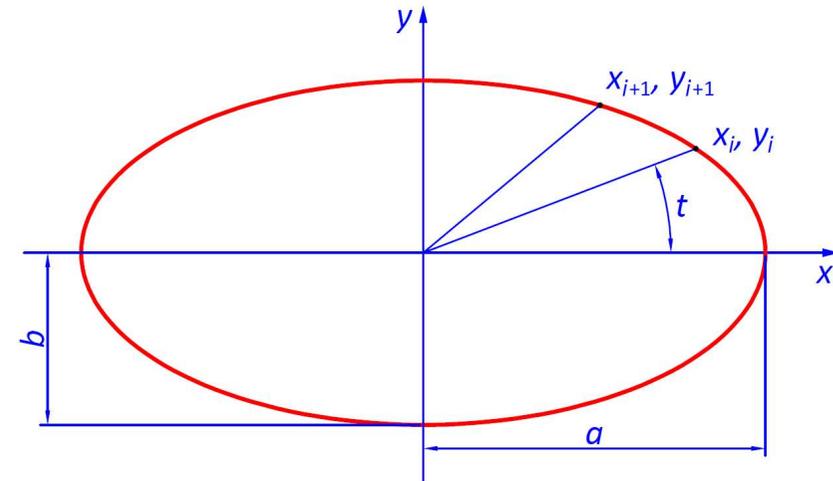


Рис. 9.7. Расчет точек на эллипсе

## 9.3.1. Вычисление точек на эллипсе с использованием параметрических уравнений эллипса

Пусть  $(x_i, y_i)$  последняя точка, вычисленная по окружности эллипса:

$$\begin{aligned}x_i &= xc + a \cdot \cos(t), \\y_i &= yc + b \cdot \sin(t).\end{aligned}\tag{9.26}$$

Следующая точка  $(x_{i+1}, y_{i+1})$ , удалена от текущей на угловой шаг (обозначенный через pas):

$$\begin{aligned}x_{i+1} &= xc + a \cdot \cos(t + pas), \\y_{i+1} &= yc + b \cdot \sin(t + pas).\end{aligned}\tag{9.27}$$

Обозначив:

$$\begin{aligned}dx &= \cos(t) \text{ и } dy = \sin(t), \\c &= \cos(pas) \text{ и } s = \sin(pas).\end{aligned}$$

И зная, что:

$$\begin{aligned}\cos(t + pas) &= \cos(t) \cdot \cos(pas) - \sin(t) \cdot \sin(pas), \\ \sin(t + pas) &= \cos(t) \cdot \sin(pas) + \sin(t) \cdot \cos(pas).\end{aligned}$$

А

$$\begin{aligned}dx' &= \cos(t + pas) = dx \cdot c - dy \cdot s, \\dy' &= \sin(t + pas) = dy \cdot c + dx \cdot s.\end{aligned}\tag{9.28}$$

Заменив их в отношениях (9.27) для  $x_{i+1}$  și  $y_{i+1}$ , получаем:

$$\begin{aligned}x_{i+1} &= xc + a \cdot dx' = xc + a \cdot (dx \cdot c - dy \cdot s), \\y_{i+1} &= yc + b \cdot dy' = yc + b \cdot (dy \cdot c + dx \cdot s).\end{aligned}\tag{9.29}$$

## 9.3.2. Представление повернутого эллипса

Представленный ранее алгоритм может быть использован для генерации эллипсов, имеющих оси, параллельные осям 2D декартовой системы координат.

Пусть  $R$  — эллипс с центром в начале системы координат и полуосями  $a$ ,  $b$ , повернутыми на угол  $u$ , относительно осей координат.

Пусть  $E$  эллипс с центром в начале и полуосями  $a$  и  $b$ , наложенными на оси системы координат.

Точки на эллипсе  $R$  можно получить, повернув точки на эллипсе  $E$  вокруг начала координат на угол  $u$ .

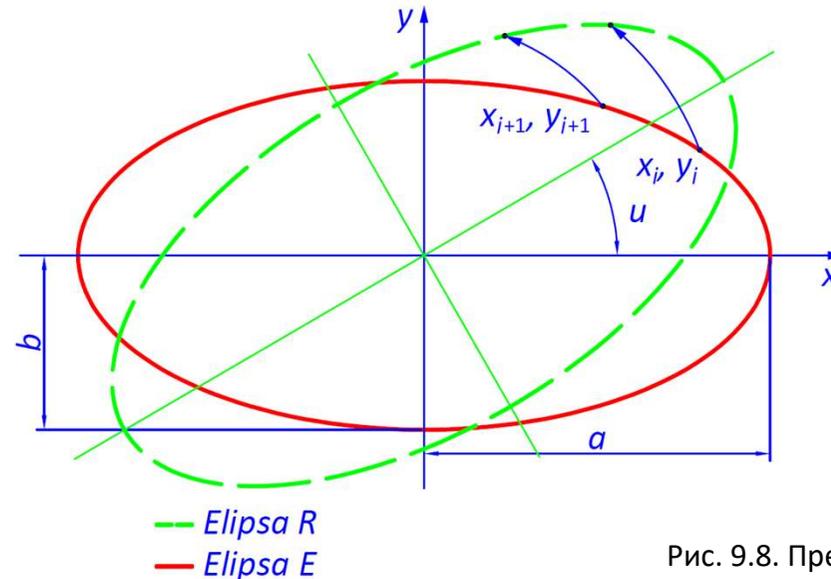


Рис. 9.8. Представление повернутого эллипса

## 9.3.2. Представление повернутого эллипса

Если центр эллипса  $R$  находится в  $(x_c, y_c)$ , затем после поворота к каждой точке применяется перенос на  $(x_c, y_c)$ .

Пусть:

$$\begin{aligned}x' &= a \cdot dx', \\y' &= b \cdot dy'.\end{aligned}\tag{9.30}$$

текущая точка на эллипсе  $E$ , где  $[dx', dy']$  были получены путем применения формулы повторения (9.28).

Обозначив:

$$\begin{aligned}dx_1 &= a \cdot dx', \\dy_1 &= b \cdot dy'.\end{aligned}\tag{9.31}$$

Соответствующая точка на эллипсе  $R$ :

$$\begin{aligned}x_r &= x_c + dxr, \\y_r &= y_c + dyr.\end{aligned}\tag{9.32}$$

где

$$\begin{aligned}dxr &= dx_1 \cdot \cos(u) - dy_1 \cdot \sin(u), \\dyr &= dx_1 \cdot \sin(u) + dy_1 \cdot \cos(u).\end{aligned}\tag{9.33}$$

### 9.3.3. Представление эллипсов в дискретном пространстве

Следующий алгоритм начинается с уравнения, соответствующего эллипсу с центром в начале и осями, выровненными с осями 2D декартовой системы координат.

*Эллипс пересекает ось Oх в  $a$  и  $-a$ , и оси Oу в  $b$  и  $-b$  (рисунок 9.9).*

Считается эллипс, определенный уравнением по умолчанию:

$$f(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 + a^2 \cdot b^2 = 0 \quad (9.34)$$

Алгоритм генерирует дискретное приближение эллипса в первом квадранте (т.е. в октантах 1 и 2), зная, что остальные точки на эллипсе могут быть получены симметрией.

Также для эллипса с центром в  $(x_c, y_c)$  применяется перенос на  $(x_c, y_c)$  (рисунок 9.10).

Алгоритм заключается в выборе между точками в дискретном пространстве относительно расстояний, на которых они расположены от теоретического эллипса.

Эти расстояния измеряются горизонтально в первом октанте и вертикально во втором октанте.

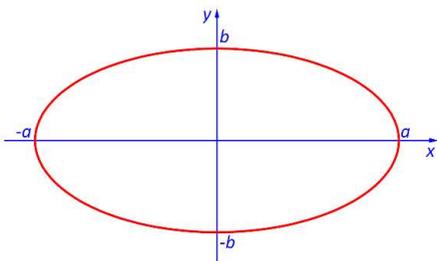


Рис. 9.9. Пересечение эллипса с осями

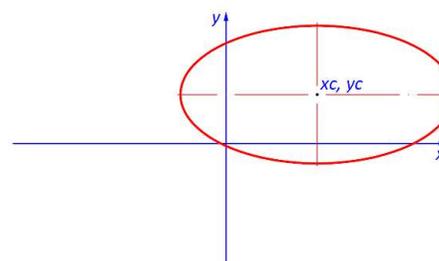


Рис. 9.10. Перенос эллипса

## 9.3.2. Представление повернутого эллипса

Пусть  $P_1$  и  $P_2$  являются двумя смежными точками дискретного пространства, между которыми вы должны выбрать.

$P_1$  находится снаружи эллипса, а  $P_2$  внутри.

Определяется, какая из них ближе к эллипсу по положению относительно точки находящееся на середине расстояния между двумя точками.

Если средняя точка находится внутри эллипса, то точка  $P_1$  ближе к эллипсу, чем  $P_2$ , и наоборот.

Вычисляется значение функции  $f$  в средней точке  $(x_m, y_m)$ :

– для первого октанта,

$$(x_m, y_m) = (x_i - 0,5, y_i + 1), \quad (9.35)$$

– для второго октанта,

$$(x_m, y_m) = (x_i - 1, y_i + 0,5). \quad (9.36)$$

Значение  $f(x_m, y_m)$  может означать:

> 0, если  $(x_m, y_m)$  точка находится снаружи эллипса;

< 0, если  $(x_m, y_m)$  точка находится внутри эллипса;

= 0, если  $(x_m, y_m)$  точка находится на эллипсе.

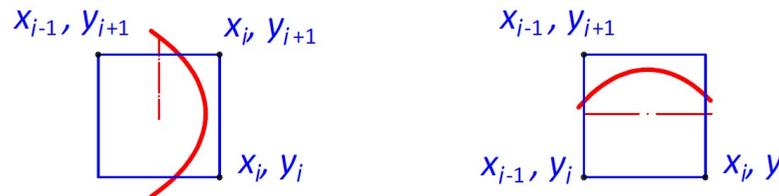


Рис. 9.11. Выбор точек для эллипса

## 9.3.2. Представление повернутого эллипса

Далее изучается ситуация, когда необходимо определить точки перехода от одного октанта к другому.

Наклон кривой, определенной в уравнении (9.34), задается отношением:

$$\frac{dx}{dy} = -\frac{fx}{fy} = -\frac{2 \cdot b^2 \cdot x}{2 \cdot a^2 \cdot y}, \quad (9.37)$$

где  $fx$  и  $fy$  являются частными производными функции  $f$  по отношению к  $x$  и  $y$ .

Переход от октанта 1 к октанту 2 происходит, когда наклон становится больше  $-1$ :

$$\text{то есть: } -\frac{2 \cdot b^2 \cdot x}{2 \cdot a^2 \cdot y} > -1,$$

$$\text{или: } 2 \cdot b^2 \cdot x < 2 \cdot a^2 \cdot y.$$

Затем делаем следующие обозначения:

$$\begin{aligned} dx &= 2 \cdot b^2 \cdot x, \\ dy &= 2 \cdot a^2 \cdot y. \end{aligned} \quad (9.38)$$

Эти значения вычисляются повторно, обновляясь, при изменении  $x$ , соответственно  $y$ .

## 9.3.2. Представление повернутого эллипса

Значение функции  $f$  в средней точке обозначим через  $fm_i$ , и также вычисляется повторно.

Для первого рас, имеем  $x_m = a - 0,5$  и  $y_m = 1$ ,

откуда получаем  $fm_i = f(x_m, y_m) = b^2 \cdot (0,25 - a) + a^2$ .

Для точек первого октанта, заменяем в уравнении (9.34) и получаем:

$$f\left(x_i - \frac{1}{2}, y_i + 1\right) = b^2 \cdot (x_i^2 - x_i + 0,25) + a^2 \cdot (y_i^2 + 2y_i + 1) - a^2 \cdot b^2. \quad (9.39)$$

Для точек второго октанта заменяем в уравнении (9.34) и получим:

$$f\left(x_i - 1, y_i + \frac{1}{2}\right) = b^2 \cdot (x_i^2 - 2x_i + 1) + a^2 \cdot (y_i^2 + y_i + 0,25) - a^2 \cdot b^2. \quad (9.40)$$

Разница между значением функции в средней точке для второго октанта и значением для первого октанта составляет:

$$b^2 \cdot \left(-x_i + \frac{3}{4}\right) + a^2 \cdot \left(-y_i - \frac{3}{4}\right).$$

## 9.4. Представление поверхностей

### 9.4.1. Представление полигонов

Поверхности могут быть определены тремя способами:

- по контуру – известен цвет пикселей, составляющих контур поверхности и одна внутренняя точка;
- через внутреннюю область – известен цвет пикселей и одна внутренняя точка;
- геометрически – известны координаты точек, геометрически определяющих контур

поверхности; в этом случае контур представляет собой некий многоугольник.

1) Для того, чтобы сгенерировать контурно-определенную область, необходимо, чтобы контур уже был вписан в растровую память.

Начиная с известной внутренней точки, меняется цвет всех внутренних пикселей контура в цвет в соответствии с шаблоном, который задан.

2) Генерация области, определяемой внутренней областью, подразумевает наличие в растровой памяти поверхности.

Начиная с известной внутренней точки, меняется цвет всех пикселей поверхности в цвете при помощи шаблона, который задается.

Существует два типа алгоритмов, используемых для создания контурных поверхностей и определяемых через внутреннюю область:

- *алгоритмы, основанные на проверке цвета точек, прилегающих к текущей точке;*
- *алгоритмы, основанные на пересечении растровых линий, пересекающих поверхность.*

Fig. 9.13. Exemple de colorare cu conexiuni de ordinul 4 și 8

## 9.4. Представление поверхностей

### 9.4.1. Представление полигонов

А) Поверхности, определяемые контуром или через внутреннюю область, могут быть:

- связанные 4-го порядка (рисунок 9.12, а),
- или связанные 8-го порядка (рисунок 9.12, б).

Поверхность получаемая связями 4-го порядка, это когда каждый пиксель поверхности может быть достигнут, начиная с внутренней точки, комбинацией смещений в 4 направлениях: влево, вправо, сверху, снизу.

Поверхность получаемая связями 8-го порядка, это когда каждый пиксель поверхности может быть достигнут, начиная с внутренней точки, комбинацией перемещений в 8 направлениях – горизонтально, вертикально и по диагоналям.

Примеры:

На рисунке 9.13 представлены поверхности:

- а – с всеобщей окраской,
- б - с частичной окраской, связанные 8-го порядка,
- в - с частичным окрашиванием, связанные 4-го порядка.

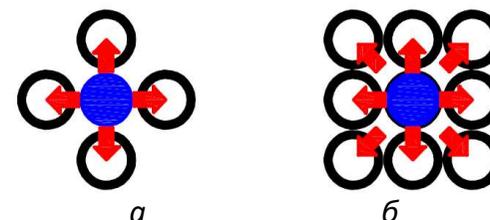


Рис. 9.12. Поверхности определяемые контуром

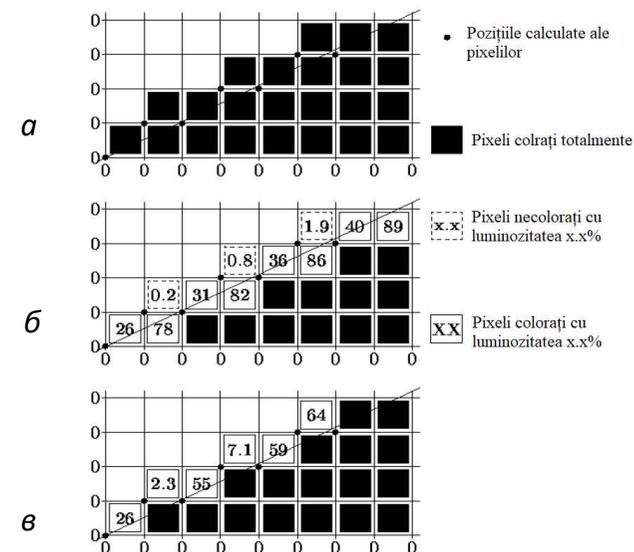


Рис. 9.13. Окрашенные поверхности связанные 8-го порядка и 4-го порядка

## 9.4. Представление поверхностей

### 9.4.1. Представление полигонов

*Б) Алгоритм требует резервирования большого пространства для хранения программ.*

Поэтому алгоритмы, тестирующего луча строка за строкой, предпочтительны для определения поверхностей через контур или через внутреннюю область.

Ниже мы представляем такой пользовательский алгоритм для областей, определенными контуром. *Поверхность может быть выпуклой или вогнутой и иметь отверстия.*

Таким образом, горизонтальная линия, которая пересекает его, может пересекаться с очертаниями поверхности несколько раз.

Рассмотрим следующую поверхность:

Компактная группа пикселей, расположенных на одной линии, которые не имеют цвета обводки или цвета заливки, образует диапазон; например, точки E, F и G являются правыми концами трех интервалов на одной полосе (рисунок 9.15).

Пусть  $P(x_{int}, y_{int})$  будет данной внутренней точкой. Он является первой отправной точкой для окрашивания.

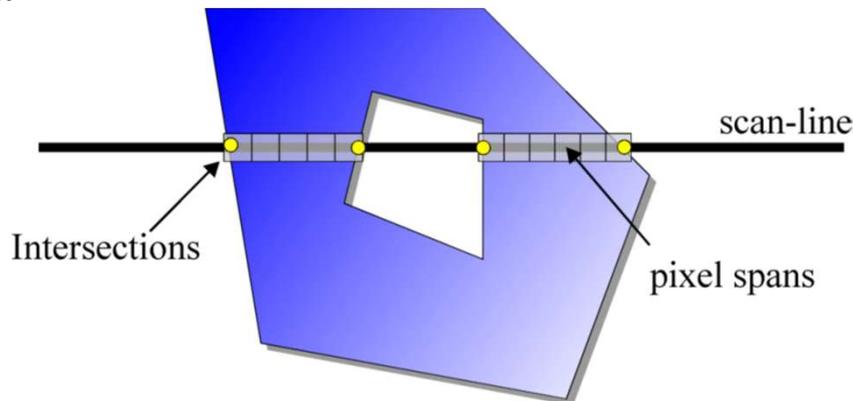


Рис. 9.14. Пример распознавания полигона

## 9.4. Представление поверхностей

### 9.4.1. Представление полигонов

В рамках алгоритма циклически выполняются следующие шаги:

1) Начиная с отправной точки,  $P(x_i, y_i)$ , раскрашиваются все пиксели справа, пока не достигнет контура, затем все пиксели слева, пока не будет достигнут контур.

Запоминаются правый конец цветовой гаммы =  $x_{\max}$ , и его левый конец =  $x_{\min}$ .

2) Рассматриваются пиксели в диапазоне  $x_{\min} - x_{\max}$  на линии над текущей линией, чтобы определить, включает ли она только контурные пиксели или уже окрашенные пиксели.

Диапазон может содержать несколько интервалов. Правый конец каждого интервала запоминается.

Такая же обработка выполняется для строки ниже текущей.

3) Определяется новая начальная точка для окрашивания  $(x_i, y_i)$  и возвращаемся к шагу 1.

Поскольку последняя точка, находится на строке ниже текущей, обработка будет продолжена с этой строки.

Выполнение алгоритма завершается, когда все точки окрашены.

Окраска многоугольника представленного на предыдущем рисунке начинается с точки I.

После окрашивания линии  $y = y_1$  будут окрашены линии в зоне 1.

На первой линии зоны 2 изменяется  $x_{\max}$ .

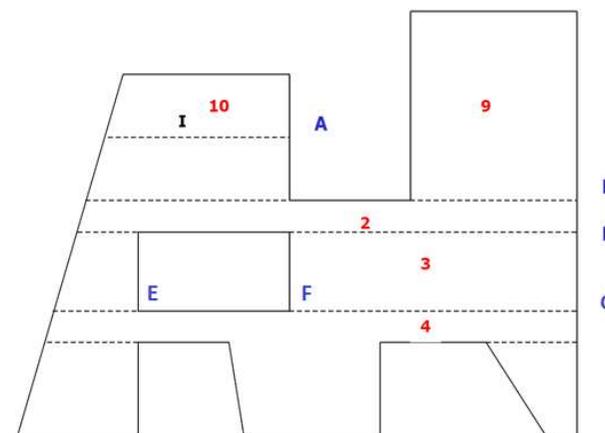


Рис. 9.15. Пример распознавания полигона

## 9.4. Представление поверхностей

### 9.4.1. Представление полигонов

При рассмотрении пикселей на линии, содержащей верхнюю сторону внутреннего прямоугольника многоугольника, сохраняются точка C, а затем точка D.

Точка D немедленно, становясь отправной точкой.

Проходя для раскраски линию  $y = y_D$ , справа налево, встретится с правой стороной прямоугольника, и изменится  $x_{\min}$ .

Окраска продолжается линиями в зоне 3, за которыми следуют линии в зоне 4 и зоне 4 и зоне 5.

При достижении нижнего предела зоны 5 точкой в верхней части является F, поэтому окрашивание продолжается с зоной 6 и так далее.

## 9.4.2. Представление круглых и эллиптических поверхностей

*Алгоритмы представления кругов и эллипсов могут быть очень легко расширены для отображения всех внутренних точек.*

Например, для каждой точки, определенной в алгоритме Брезенхама, можно провести четыре горизонтальные линии, определяемые восемью симметричными точками на окружности.

## 9.4.3. Представление через внутреннюю область посредством шаблона

*Шаблон представляет собой матрицу целых чисел, представляющих цвета.*

Представление внутренней части поверхности предполагает, что каждый элемент матрицы шаблона соответствует одному пикселю.

Шаблон наносится либо из заданной внутренней точки, либо из начала экрана, окрашивая только внутренние точки поверхности.

Очень простой метод представления внутренней области посредством шаблона основан на алгоритме раскрашивания путем штриховочных линий.

Таким образом, вместо отображения всех пикселей на штриховочном отрезке одинаковым цветом, цвет каждого пикселя будет извлечен из строки матрицы шаблона, соответствующей линии штриховки.

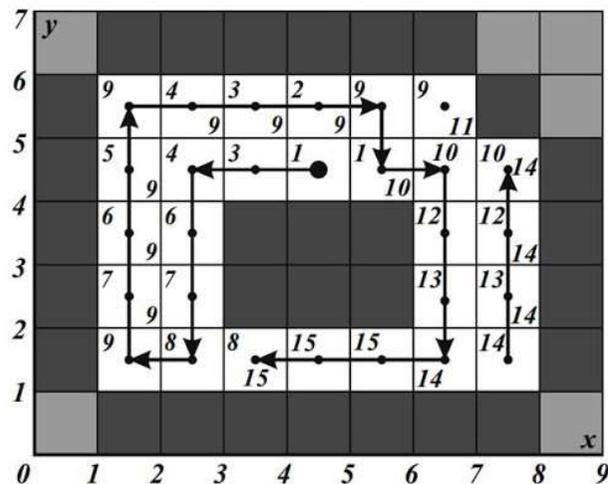


Рис. 9.16. Пример раскраски многоугольника связным методом 4-го порядка

## 9.4.3. Представление через внутреннюю область посредством шаблона

Рассмотрим случай, когда шаблон применяется из точки  $(0,0)$ .

Мы отмечаем, что  $hs$  – количество строк матрицы шаблона и  $ls$  – количество столбцов.

Затем пиксели на штриховочном отрезке  $y = yh$  будут окрашены с использованием элементов на линии  $lin = yh \% hs$  матрицы шаблона.

Пиксель с координатами  $(x, yh)$  будет окрашен с использованием значения элемента на линии  $lin$  и столбца  $x \% ls$  матрицы шаблона.

# ВОПРОСЫ