

# Elementy grafiki komputerowej. Elementy algorytmów rastrowych

Aleksander Denisiuk  
Uniwersytet Warmińsko-Mazurski  
Olsztyn, ul. Słoneczna 54  
[denisjuk@matman.uwm.edu.pl](mailto:denisjuk@matman.uwm.edu.pl)

# *Elementy algorytmów rastrowych*

- ❖ Rasteryzacja odcinka
- ❖ Rasteryzacja okręgu
- ❖ Rasteryzacja elipsy
- ❖ Rasteryzacja krywej
- ❖ Wypełnienie obszaru

Najnowsza wersja tego dokumentu dostępna jest pod adresem

<http://wmii.uwm.edu.pl/~denisjuk/uwm>

# Rasteryzacja odcinka

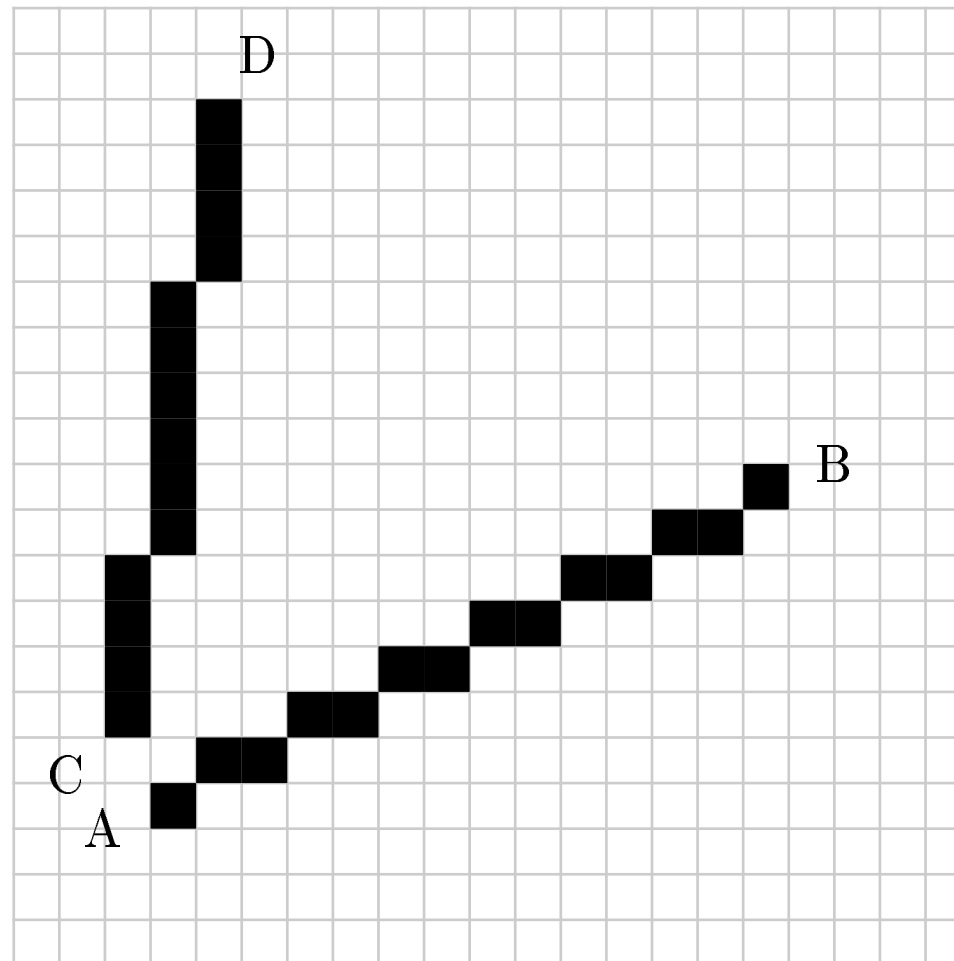
❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru



# Założenia

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

- $(x, y)$  — współrzędne „abstrakcyjne”, liczby rzeczywiste
- $(i, j)$  — współrzędne ekranowe, liczby całkowite
- $x_2 > x_1, y_2 \geq y_1$
- $y_2 - y_1 \leq x_2 - x_1$
- **Zaokrąglenie:**  $i_1 = \text{round}(x_1), i_2 = \text{round}(x_2), j_1 = \text{round}(y_1), j_2 = \text{round}(y_2)$

# Algorytm

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

- $y(i) = j_1 + \frac{i-i_1}{i_2-i_1}(j_2 - j_1)$
- $j = \text{round}(y)$
- Kod:

**Wejście:**  $(i_1, j_1)$  — początek odcinka,  $(i_2, j_2)$  — koniec odcinka,  $i_2 > i_1, j_2 \geq j_1, j_2 - j_1 \leq i_2 - i_1$

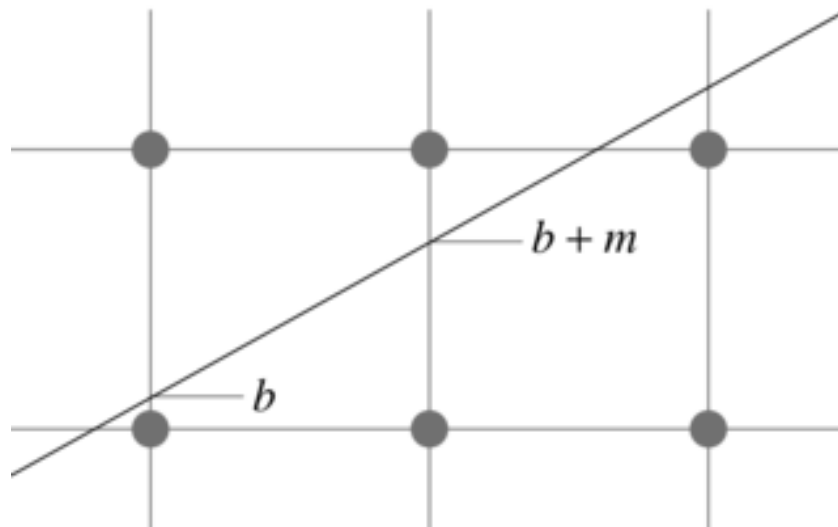
**Wynik:** Odcinek został wyświetlony

```
 $m \leftarrow \frac{j_2 - j_1}{i_2 - i_1}$   
writePixel( $i_1, j_1$ )  
 $y \leftarrow j_1$   
for  $i = i_1 + 1$  to  $i_2$  do  
     $y \leftarrow y + m$   
     $j \leftarrow \text{round}(y)$   
    writePixel( $i, j$ )  
end for
```

# Kumulacja przyrostu $y$

## ❖ Rasteryzacja odcinka

- ❖ Rasteryzacja okręgu
- ❖ Rasteryzacja elipsy
- ❖ Rasteryzacja krzywej
- ❖ Wypełnienie obszaru



- na każdym kroku do przyrostu  $y$  dodaje się  $m$
- przechodzimy o jeden piksel w górę, jeżeli przyrost przekroczy  $1/2$

# Algorytm 2

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

**Wejście:**  $(i_1, j_1)$  — początek odcinka,  $(i_2, j_2)$  — koniec odcinka,  $i_2 > i_1, j_2 \geq j_1, j_2 - j_1 \leq i_2 - i_1$

**Wynik:** Odcinek został wyświetlony

$$m \leftarrow \frac{j_2 - j_1}{i_2 - i_1}$$

$$b \leftarrow 0$$

writePixel( $i_1, j_1$ )

$$j \leftarrow j_1$$

**for**  $i = i_1 + 1$  **to**  $i_2$  **do**

$$b \leftarrow b + m$$

**if**  $b > \frac{1}{2}$  **then**

$$j \leftarrow j + 1$$

$$b \leftarrow b - 1$$

**end if**

writePixel( $i, j$ )

**end for**

# Eliminacja liczb rzeczywistych

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

- przyrost jest wielokrotnością  $m = \frac{j_2 - j_1}{i_2 - i_1}$ :
  - ◆  $b = k \frac{j_2 - j_1}{i_2 - i_1}$
- $b < \frac{1}{2} \iff 2k(j_2 - j_1) < i_2 - i_1$ 
  - ◆ zamieniamy przyrost na przyrost całkowity
  - ◆ przyrost całkowity na każdym kroku zwiększa się o  $2\Delta j = 2(j_2 - j_1)$
  - ◆ przechodzimy na wyższy poziom w  $j$ , jeżeli przyrost całkowity przekroczy  $(i_2 - i_1)$



# Algorytm Bresenhama

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

**Wejście:**  $(i_1, j_1)$  — początek odcinka,  $(i_2, j_2)$  — koniec odcinka,  $i_2 > i_1, j_2 \geq j_1, j_2 - j_1 \leq i_2 - i_1$

**Wynik:** Odcinek został wyświetlony

$m \leftarrow 2(j_2 - j_1)$

$b \leftarrow 0$

$\text{writePixel}(i_1, j_1)$

$j \leftarrow j_1$

$P \leftarrow i_2 - i_1$

**for**  $i = i_1 + 1$  **to**  $i_2$  **do**

$b \leftarrow b + m$

**if**  $b > P$  **then**

$j \leftarrow j + 1$

$b \leftarrow b - 2P$

**end if**

$\text{writePixel}(i, j)$

**end for**

# Osiem symetrii okręgu

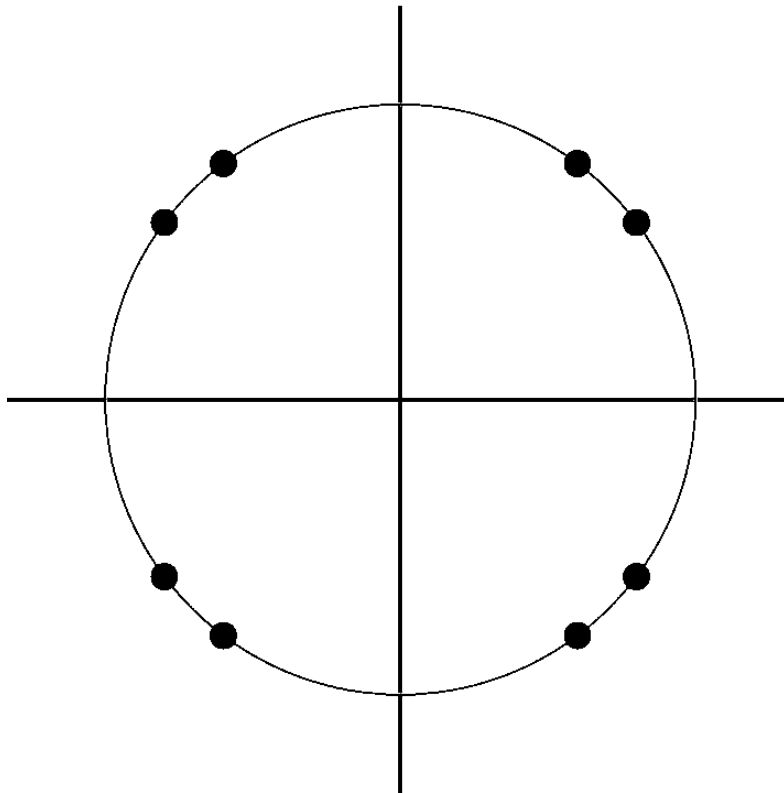
❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru



- $x^2 + y^2 = R^2$
- jeżeli  $(x, y)$  leży na okręgu, to
  - ◆  $(y, x), (x, -y), (y, -x), (-x, y), (-y, x), (-x, -y), (-y, -x)$  też leżą na okręgu

# Wybór następnego piksela

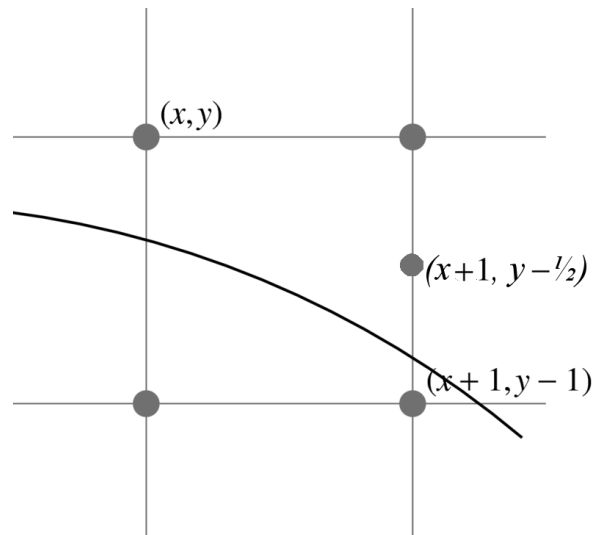
❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru



- Zaczynamy od wierzchołka  $(0, R)$
- Analizujemy  $f(x, y) = 4((x + 1)^2 + (y - \frac{1}{2})^2 - R^2)$ 
  - ◆ jeżeli  $f(x, y) > 0$  to przechodzimy w prawo i w dół
  - ◆ jeżeli  $f(x, y) < 0$  to przechodzimy tylko w prawo
- $f(x + 1, y) = f(x, y) + 8x + 12$
- $f(x + 1, y - 1) = f(x, y) + 8x - 8y + 20$
- $f(0, R) = 5 - 4R$

# Algorytm

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

**Wejście:** Środek okręgu jest w  $(0, 0)$ , promień  $R \in \mathbb{N}$

**Wynik:** Okrąg został wyświetlony

$i \leftarrow 0, j \leftarrow R, f \leftarrow 5 - 4R$

**writePixel**( $i, j$ )

**while**  $i < j$  **do**

**if**  $f > 0$  **then**

$f \leftarrow f + 8i - 8j + 20$

$j \leftarrow j - 1$

**else**

$f \leftarrow f + 8i + 12$

**end if**

$i \leftarrow i + 1$

**writePixel**( $i, j$ )

**end while**

# Rasteryzacja elipsy

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

- Zaczynamy od wierzchołka  $(0, b)$
- Analizujemy  $f(x, y) = 4a^2b^2 \left( \frac{(x+1)^2}{a^2} + \frac{(y-\frac{1}{2})^2}{b^2} - 1 \right)$ 
  - ◆ jeżeli  $f(x, y) > 0$  to przechodzimy w prawo i w dół
  - ◆ jeżeli  $f(x, y) < 0$  to przechodzimy tylko w prawo
- $f(x + 1, y) = f(x, y) + 8b^2x + 12b^2$
- $f(x + 1, y - 1) = f(x, y) + 8b^2x - 8a^2y + 12b^2 + 8a^2$
- $f(0, b) = 4b^2 - 4a^2b + a^2$

# Zmiana kierunku

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

- Jeżeli  $b^2x > a^2y$ , to zmienia się kierunek rasteryzacji
- Analizujemy  $g(x, y) = 4a^2b^2 \left( \frac{(x+\frac{1}{2})^2}{a^2} + \frac{(y-1)^2}{b^2} - 1 \right)$ 
  - ◆ jeżeli  $g(x, y) > 0$  to przechodzimy w prawo i w dół
  - ◆ jeżeli  $g(x, y) < 0$  to przechodzimy tylko w dół
- $g(x, y) = f(x, y) - 4b^2x - 3b^2 - 4a^2y + 3a^2$
- $g(x, y - 1) = g(x, y) - 8a^2y + 12a^2$
- $g(x + 1, y - 1) = g(x, y) + 8b^2x - 8a^2y + 8b^2 + 12a^2$

# Algorytm

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

**Wejście:** Środek elipsy jest w  $(0, 0)$ , promienie  $a, b \in \mathbb{N}$

**Wynik:** Elipsa została wyświetlona

$i \leftarrow 0, j \leftarrow b, f \leftarrow 4b^2 - 4a^2b + a^2$

writePixel( $i, j$ )

**while**  $b^2i < a^2j$  **do**

**if**  $f > 0$  **then**

$f \leftarrow f + 8b^2i - 8a^2j + 12b^2 + 8a^2$

$j \leftarrow j - 1$

**else**

$f \leftarrow f + 8b^2i + 12b^2$

**end if**

$i \leftarrow i + 1$

  writePixel( $i, j$ )

**end while**

# Algorytm. Zmiana kierunku

- ❖ Rasteryzacja odcinka
- ❖ Rasteryzacja okręgu
- ❖ Rasteryzacja elipsy
- ❖ Rasteryzacja krywej
- ❖ Wypełnienie obszaru

```

$$g \leftarrow f - 4b^2i - 3b^2 - 4a^2j + 3a^2$$
while  $j > 0$  do  
  if  $g \leq 0$  then  
     $g \leftarrow g + 8b^2i - 8a^2j + 8b^2 + 12a^2$   
     $i \leftarrow i + 1$   
  else  
     $g \leftarrow g - 8a^2j + 12a^2$   
  end if  
   $j \leftarrow j - 1$   
  writePixel( $i, j$ )  
end while
```



# Rasteryzacja krywej

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

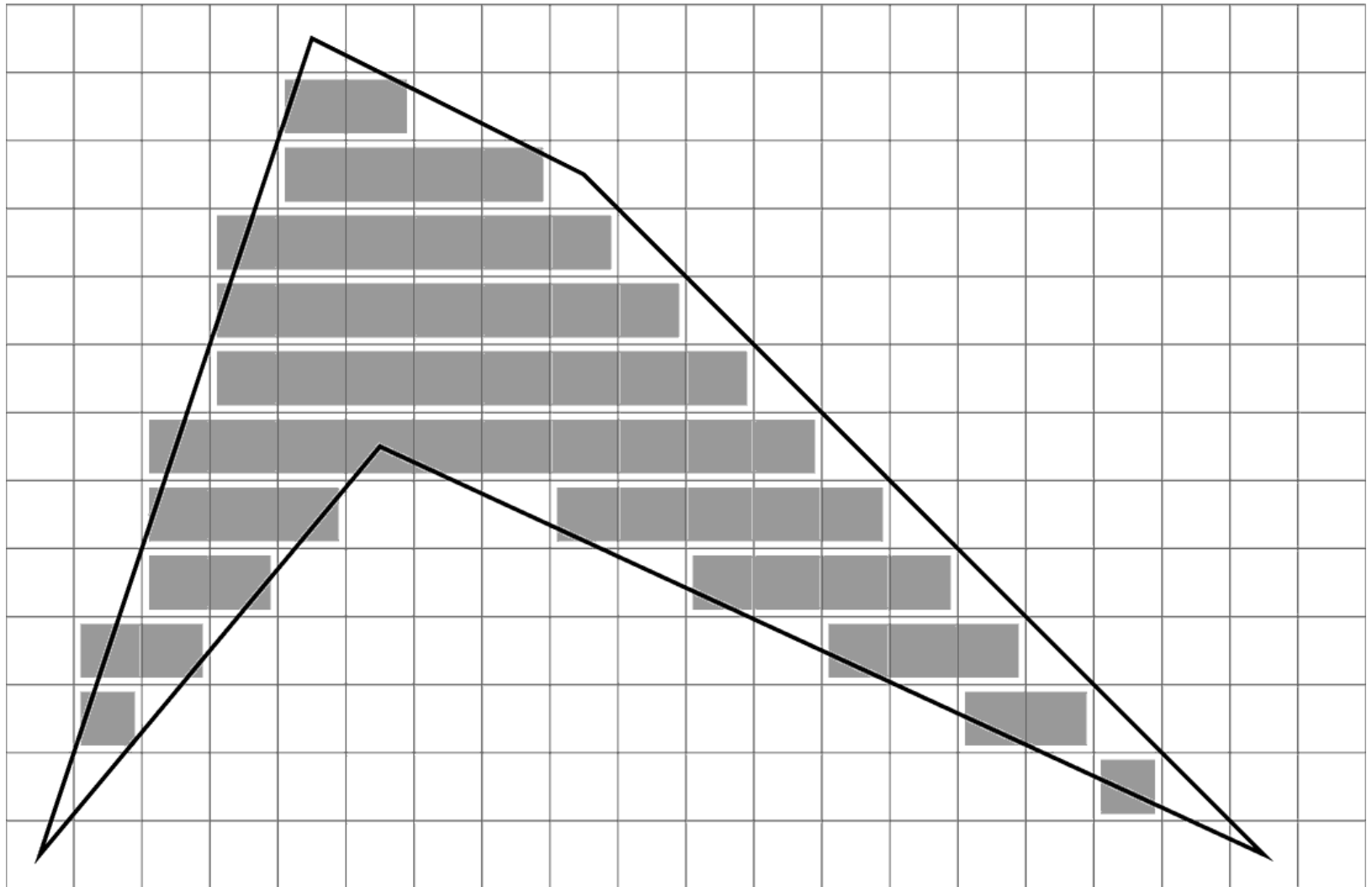
- Przybliżenie przez łamaną

- Metoda Eulera dla równania

$$\begin{cases} \dot{x} = f_1(x, y), \\ \dot{y} = f_2(x, y), \\ x(0) = x_0, \quad y(0) = y_0. \end{cases}$$

# Wypełnienie wieloboku

- ❖ Rasteryzacja odcinka
- ❖ Rasteryzacja okręgu
- ❖ Rasteryzacja elipsy
- ❖ Rasteryzacja krywej
- ❖ Wypełnienie obszaru



# Przeглядanie liniami poziomymi (Scanline interpolation)

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

**Wejście:** lista krawędzi wieloboku  $\{ [(x_i, y_i), (x_{i+1}, y_{i+1})] \}$ ,  
 $i = 0, \dots, n, x_n = x_0, y_n = y_0$

**Wynik:** wypełniono wnętrze wieloboku

uporządkuj wierzchołki w krawędziach aby  $y_i < y_{i+1}$ , usuń krawędzie poziome

uporządkuj krawędzie w kolejności rosnących  $y_i$

$TAK \leftarrow \emptyset$  (Tabela Aktywnych Krawędzi)

$y \leftarrow y_i$  pierwszej krawędzi

**repeat**

$TAK \leftarrow TAK \cup \{ \text{krawędzie, których pierwszy koniec jest na linii } y \}$

Opracowanie poziomu  $y$

$y ++;$

$TAK \leftarrow TAK \setminus \{ \text{krawędzie, których drugi koniec jest na linii } y \}$

**until**  $TAK = \emptyset$

# Opracowanie poziomu $y$

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru

**Wejście:**  $TKA \neq \emptyset$  (zawiera parzystą ilość elementów)

**Wynik:** wypełniony poziom  $y$

**for all** krawędzi z  $TKA$  **do**

Oblicz współrzędną  $x$  punktu przecięcia z linią poziomą  $y$

**end for**

Posortuj  $TKA$  w kolejności rosnących współrzędnych  $x$  punktów przecięcia

**for all** kolejnych par krawędzi z  $TKA$  **do**

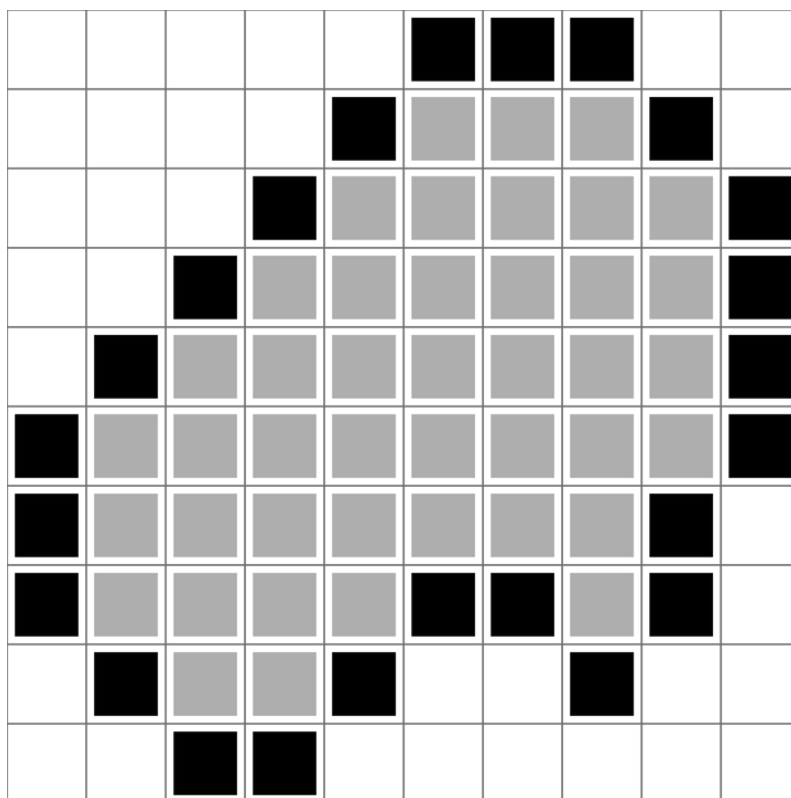
rysuj odcinek poziomy na linii  $y$ , między ich punktami przecięcia z linią  $y$ ;

**end for**

# Wypełnianie przez zalewanie

- ❖ Rasteryzacja odcinka
- ❖ Rasteryzacja okręgu
- ❖ Rasteryzacja elipsy
- ❖ Rasteryzacja krywej
- ❖ Wypełnienie obszaru

- obszar jest czterospójny
- brzeg obszaru jest ośmiospójny



# Procedura rekurencyjna

- ❖ Rasteryzacja odcinka
- ❖ Rasteryzacja okręgu
- ❖ Rasteryzacja elipsy
- ❖ Rasteryzacja krywej
- ❖ Wypełnienie obszaru

**Wejście:** punkt  $(i, j)$  zawiera się w obszarze

**Wynik:** zamalowany cały obszar

**if** niezamalowany wewnętrzny piksel  $(i, j)$  **then**

Zamaluj  $(i, j)$

Wypełnij poczynając z  $(i - 1, j)$

Wypełnij poczynając z  $(i, j - 1)$

Wypełnij poczynając z  $(i + 1, j)$

Wypełnij poczynając z  $(i, j + 1)$

**end if**

# Stos zawieszonych zadań

**Wejście:** punkt  $(i, j)$  zawiera się w obszarze

**Wynik:** zamalowany cały obszar

Stos  $S \leftarrow \emptyset$

zamaluj  $(i, j)$ ;  $S \leftarrow S \cup (i, j)$

**while**  $S \neq \emptyset$  **do**

$S \leftarrow S \setminus (i, j)$

**if** niezamalowany wewnętrzny piksel  $(i - 1, j)$  **then**

zamaluj  $(i - 1, j)$ ;  $S \leftarrow S \cup (i - 1, j)$

**end if**

**if** niezamalowany wewnętrzny piksel  $(i, j - 1)$  **then**

zamaluj  $(i, j - 1)$ ;  $S \leftarrow S \cup (i, j - 1)$

**end if**

**if** niezamalowany wewnętrzny piksel  $(i + 1, j)$  **then**

zamaluj  $(i + 1, j)$ ;  $S \leftarrow S \cup (i + 1, j)$

**end if**

**if** niezamalowany wewnętrzny piksel  $(i, j + 1)$  **then**

zamaluj  $(i, j + 1)$ ;  $S \leftarrow S \cup (i, j + 1)$

**end if**

**end while**

❖ Rasteryzacja odcinka

❖ Rasteryzacja okręgu

❖ Rasteryzacja elipsy

❖ Rasteryzacja krywej

❖ Wypełnienie obszaru