Politechnika Krakowska
Wydział Inżynierii Elektrycznej i Komputerowej
Katedra Informatyki Technicznej

Instrukcja do ćwiczeń laboratoryjnych
z przedmiotu:

Grafika Komputerowa i Multimedia

Rasteryzacja odcinka, okręgu i elipsy

Podstawy

Grafika rastrowa aproksymuje prymitywy matematyczne (idealne), opisane przez wierzchołki siatki kartezjańskiej, za pomocą zbiorów pikseli o odpowiednim poziomie szarości lub barwie. Piksele są pamiętane w postaci mapy bitowej albo pikselowej w pamięci CPU lub w pamięci obrazu. Algorytm rysowania oblicza współrzędne pikseli, które leżą na lub blisko idealnej nieskończenie cienkiej linii nałożonej na siatkę dwuwymiarowego rastra.

Rysując na urządzeniu rastrowym przechodzimy do układu współrzędnych całkowitych (układ współrzędnych pikselowych). W takim układzie piksel będzie reprezentowany np. jako kółko o środku w punkcie (x, y) w siatce całkowitoliczbowej.

Rysując np. odcinek chcemy, aby sekwencja pikseli leżała tak blisko idealnej linii, jak to tylko możliwe, i aby była ona możliwi prosta. Rozważamy wyświetlanie rysunków o grubości jednego piksela, lecz pamiętajmy że powinno się udostępnić możliwość rysowania odcinków o grubości większej niż jeden piksel.

Rasteryzacja odcinka

Założenia:

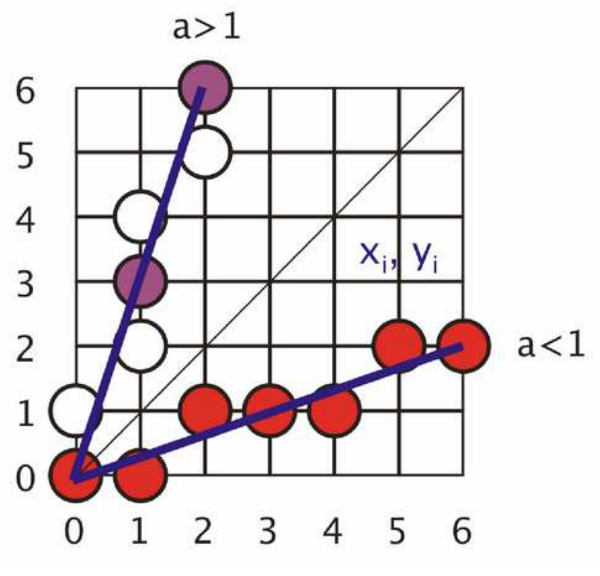
$$y = ax + b$$
$$0 \le a \le 1$$
$$b = 0$$

Jeżeli a>1, analizujemy równanie

$$y = \frac{1}{a}x$$

i rysujemy punkty $(y_i\ ,\ x_i)$

Jeżeli b≠0
rysujemy punkty $(x_i,\ y_i + \mathrm{round}(b))$



Algorytm 1 (bezpośredni)

- Krok 0
  Rysujemy punkt (0,0)

- Krok i-ty
  Dla x_i = i obliczamy $y_i = ax_i$
  i rysujemy (x_i, round (y_i))

Algorytm 2 (przyrostowy)

Algorytm DDA (Digital Differential Analizer)

- Założenia:

$$y_{i-1} = ax_{i-1} \qquad y_i = ax_i$$
$$y_i = a\,(x_{i-1} + 1)$$
$$y_i = ax_{i-1} + a = y_{i-1} + a$$

- Krok 0
  Rysujemy punkt (0,0)

- Krok i-ty
  Obliczamy $y_i = y_{i-1} + a$
  i rysujemy (x_i, round(y_i))

Algorytm 3 (Bresenhama)

Bresenham J. E., *Algorithm for Computer Control of Digital Plotter*, IBM Systems Journal, 4, 1965.

- Założenia:

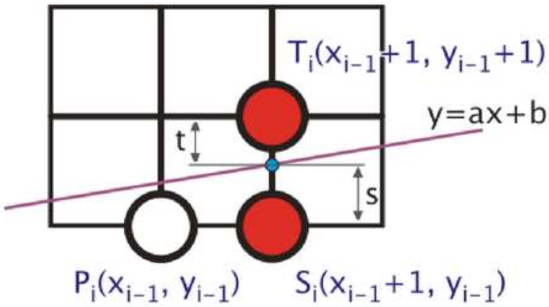
$$y = ax + b$$
$$0 \le a \le 1 \qquad b = 0$$

$$a = \frac{dy}{dx} \qquad dx, dy \in N$$

$$t = y_{i-1} + 1 - \frac{dy}{dx}(x_{i-1} + 1)$$

$$s = \frac{dy}{dx}(x_{i-1} + 1) - y_{i-1}$$

$$s - t = 2\frac{dy}{dx}(x_{i-1} + 1) - 2y_{i-1} - 1$$



- **Krok 0:**
  Rysujemy punkt (0,0) i obliczmy $d_1 = 2dy - dx$

- **Krok i-ty**
  Dla kryterium d_i obliczonego w kroku poprzednim jeżeli:
  1. $d_i \geq 0$ rysujemy T_i i obliczamy $d_{i+1}=d_i+2(dy-dx)$
  2. $d_i < 0$ rysujemy S_i i obliczamy $d_{i+1}=d_i+2dy$

Rasteryzacja okręgu

Założenia:
- środek okręgu w punkcie (0,0)
- Promień okręgu wynosi r

Algorytm 1 (bezpośredni)

$$x^2 + y^2 = r^2$$
$$y = \pm\sqrt{r^2 - x^2}$$

Algorytm 2 (równania parametryczne)

$$x(\alpha) = r \cos \alpha \qquad 0 < \alpha < 2\pi$$
$$y(\alpha) = r \sin \alpha$$

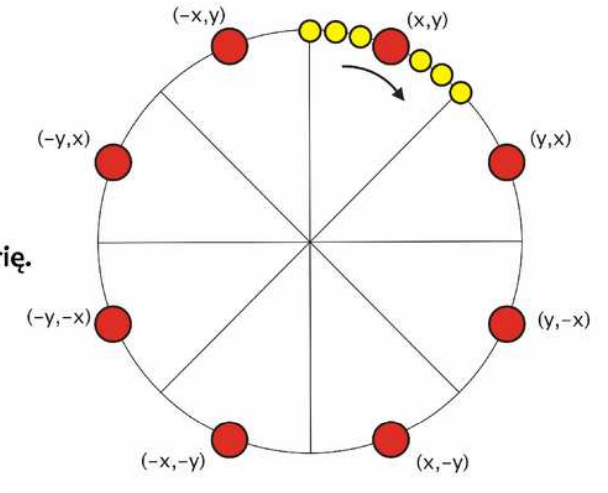
Rysujemy punkty:

$$(\mathrm{round}(x(i \cdot \Delta\alpha)), \mathrm{round}(y(i \cdot \Delta\alpha))) \qquad i = 0,1,2,..$$

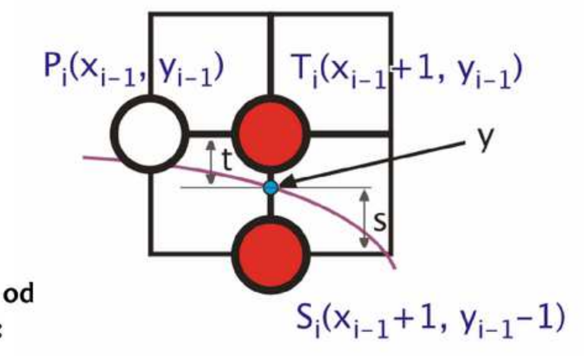
Algorytm 3 (Bresenhama)

Bresenham J. E., *A Linear Algorithm for Incremental Digital Display of Arcs*, Communications of ACM, 20, 1977

Patent USA 4 371 933 (nowsza wersja algorytmu 1983)



- Rysowana jest 1/8 okręgu od punktu (0,r).
- Aby uzyskać pełny okrąg należy punkty odpowiednio powielać wykorzystując symetrię.



Punkt $(y, x_{i-1}+1)$ określony jest równaniem:

$$y^2 = r^2 - (x_{i-1} + 1)^2$$

Zakładamy, że punkt P_{i-1} został wyznaczony

Odległość „prawdziwego" punktu okręgu $(y, x_{i-1}+1)$ od punktów siatki T_i i S_i oceniamy przy pomocy wyrażeń:

$$t = y_{i-1}^{\;2} - y^2 = y_{i-1}^{\;2} - r^2 + (x_{i-1} + 1)^2$$

$$s = y^2 - (y_{i-1} - 1)^2 = r^2 - (x_{i-1} + 1)^2 - (y_{i-1} - 1)^2$$

- Krok 0:
  Rysujemy punkt (0,r) i obliczmy $d_1 = 3 - 2r$
- Krok i-ty (do momentu gdy x>y)
  Dla kryterium d_i obliczonego w kroku poprzednim jeżeli:

  1. $d_i<0$ rysujemy T_i i obliczamy $d_{i+1} = d_i + 4x_{i-1} + 6$

  2. $d_i \geq 0$ rysujemy S_i i obliczamy $d_{i+1} = d_i + 4(x_{i-1} - y_{i-1}) + 10$

Rasteryzacja elipsy

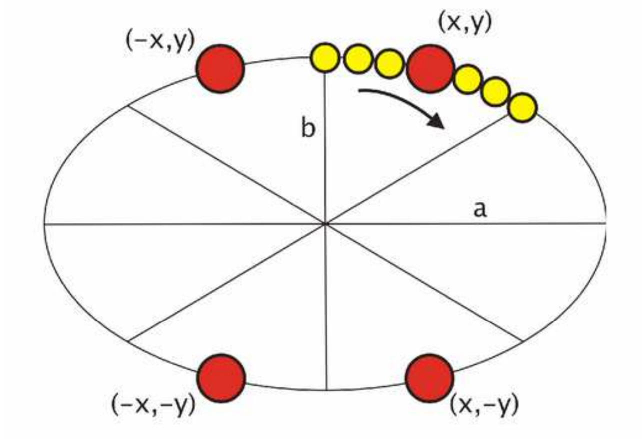
Założenia:

- **środek elipsy w punkcie (0,0)**
- **Półosie elipsy wynoszą a i b**

$$b^2x^2 + a^2y^2 = a^2b^2$$

Algorytm (Bresenhama)

- Rysowana jest 1/8 elipsy od punktu (0,b).
- W każdym kroku stawiamy cztery symetryczne punkty
- Początkową osią wiodącą jest oś OX



- **Krok 0:**
  Rysujemy punkt (0,b) i obliczmy $d_1 = 4b^2 - 4a^2b + a^2$
- **Krok i-ty (do momentu zmiany osi wiodącej - współczynnik nachylenia stycznej do elipsy równy -1 [tg 135°])**
  Dla kryterium d_i obliczonego w kroku poprzednim jeżeli:

  1. $d_i<0$ rysujemy T_i i obliczamy $d_{i+1} = d_i + 8b^2x_{i-1} + 12b^2$

  2. $d_i \geq 0$ rysujemy S_i i obliczamy $d_{i+1} = d_i + 8b^2x_{i-1} + 12b^2 - 8a^2y_{i-1} + 8a^2$

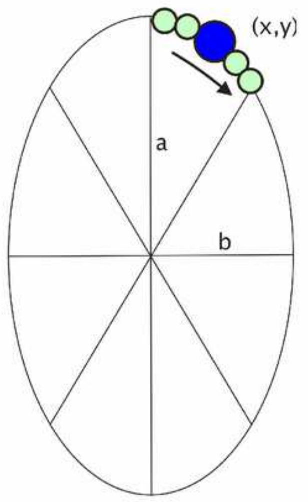
Do obliczenia punktu zmiany osi wiodącej wykorzystujemy wyrażenie:

$$x^2 = \frac{a^4}{a^2 + b^2}$$

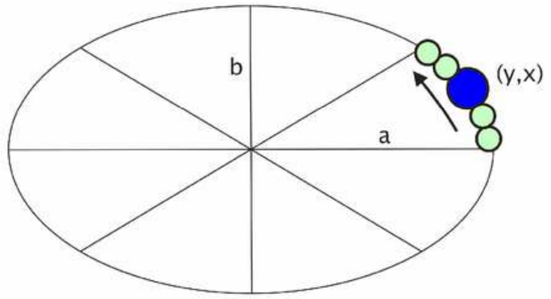
Przy zmianie osi wiodącej z OX na OY musimy także zmienić wartość zmiennej decyzyjnej, lub…

Algorytm (Bresenhama)
(po zmianie osi wiodącej)

– Jeżeli zamienimy w kreślonej elipsie półosie a z b wówczas opisanym wcześniej algorytmem wykreślimy część oznaczoną na rysunku



– Jeżeli zamienimy miejscami współrzędne x i y wówczas wykreślimy część oznaczon na rysunku



– Krok 0:
Rysujemy punkt (a,0) i obliczamy $d_1 = 4a^2 - 4b^2a + b^2$

– Krok i-ty (do momentu zmiany osi wiodącej - współczynnik nachylenia stycznej do elipsy równy -1 [tg 135°])
Dla kryterium d_i obliczonego w kroku poprzednim jeżeli:

1. $d_i < 0$ rysujemy T_i i obliczamy $d_{i+1} = d_i + 8a^2y_{i-1} + 12a^2$

2. $d_i \geq 0$ rysujemy S_i i obliczamy $d_{i+1} = d_i + 8a^2y_{i-1} + 12a^2 - 8b^2x_{i-1} + 8b^2$

Do obliczenia punktu zmiany osi wiodącej wykorzystujemy wyrażenie:

$$y^2 = \frac{b^4}{b^2 + a^2}$$