

Wprowadzenie do Grafiki Komputerowej

Ryszard S. Choraś

Literatura

- J. D. Foley, A. van Dam i inni: *Wprowadzenie do grafiki komputerowej*. WNT Warszawa 1995.
- *Grafika komputerowa metody i narzędzia*, pod red. Jana Zabrodzkiego: WNT Warszawa 1994.

Grafika

- ◆ **grafika rastrowa** (grafika bitmapowa lub malarzka),
grafika wektorowa i jej specjalne rodzaje: komputerowy rysunek techniczny, CAD, Computer Aided Design;
grafika prezentacyjna;

◆ Cechy grafiki rastrowej

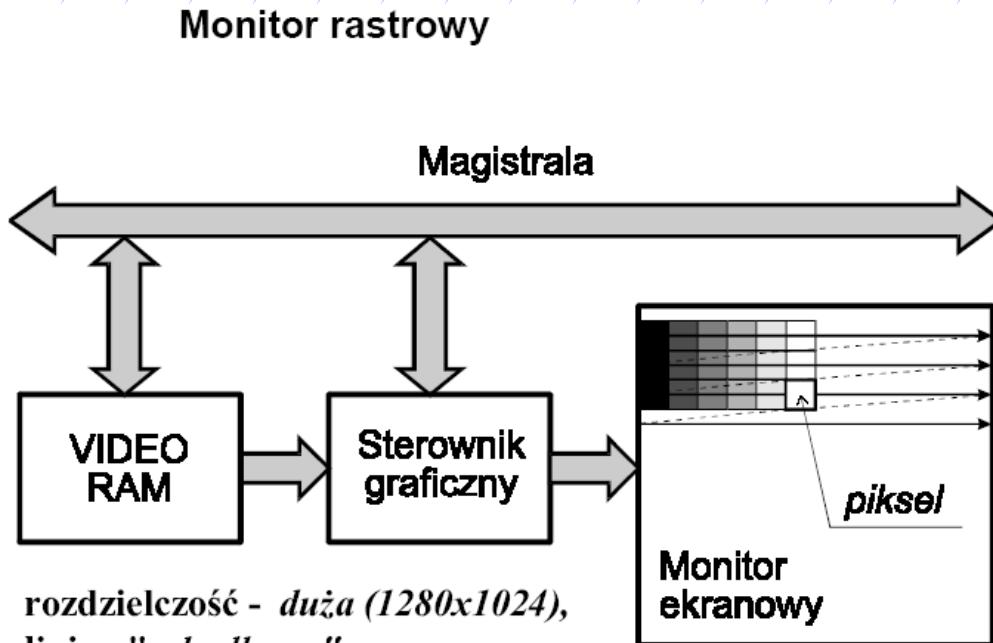
- ◆ Programy malarzkie: działają na mapach bitowych, złożonych z pikseli (pixel jest skrótem od angielskiego „picture element”), posiadających atrybuty.
- ◆ Zdjęcie legitymacyjne 3x4 cm, 300 dpi, ok. 350×470 punktów = 165 K = 0.5 MB w wiernym kolorze.
Pocztówka 10x15 cm, 300 dpi, ok. 1200×1800 pikseli =ok. 2.2 M punktów = 6.6 MB!

Grafika



- ◆ **Kolor:**
1 bit - czarnobiały,
8 bitów - pseudokolor (256 kolorów),
16 bitów - kolor pełny (32768 kolorów)
24 bity - kolor wierny, prawdziwy (16.8 M barw)
32 i 48 bitów - spotyka się w zastosowaniach profesjonalnych.
- ◆ **Mieszanie barw** - dithering (metody Bayera, Burkesa, Stuckiego ...),
- ◆ Parametry koloru: barwa, nasycenie i jasność (Hue, Saturation, Brightness), czyli "model HSB".
Podstawowe barwy RGB (czerwony, zielony, niebieski) dla monitorów
lub CMYK dla drukarek (turkusowa czyli cyjan, karmazynowa czyli
magenta i żółta oraz czarna),
Standardy Pantone oraz Trumatch.

Grafika rastrowa



- rozdzielcość - *duża* (1280×1024),
- linie - "*schodkowe*",
- złożoność obrazu - *bardzo duża*,
- wypełnianie konturów - *łatwe*,
- kolor - *szeroko dostępny*,
- cena - *dość niska*.

Grafika

◆ Podstawowe formaty grafiki rastrowej:

BMP - standard Windows, brak kompresji, używany np. do tła

TIF - Tagged File Image Format, różne warianty, kompresja bez strat LZW, używany przez faksy

GIF - Graphics Interchange Program, popularny, dobra kompresja dla koloru 1, 2, 4, 8-bitowego

JPG - Joint Photographic Experts Group, prawie najlepsza kompresja (nieodwracalna) dla wiernego koloru

PCX - 256 kolorów, kompresja do 3 razy, stracił na popularności

- Zdjęcie w formacie XGA, 1024x768 piksele, ma:

BMP - 2304 kB,

TIF bez kompresji - 2307 kB, z kompresją zip - 1133 kB, LZW - 1515 kB

PCX - 1543 kB,

PNG - 650 kB,

GIF 256 kolorów - 291 kB,

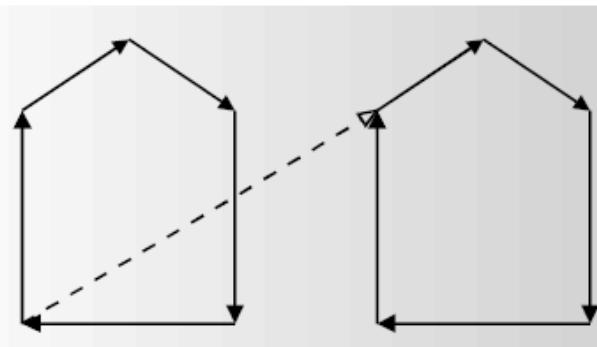
JPG - jakość 80% - 72 kB, jakość 60% - 57 kB (stopień kompresji zależy od rodzaju zdjęcia).

Grafika wektorowa

Monitor wektorowy

$$u_x = U_0 \cos \omega t$$

$$u_y = U_0 \sin \omega t$$



- rozdzielcość - *bardzo duża (4096x4096)*,
- linie - *ciągłe, cienkie*,
- złożoność obrazu - *ograniczona czasem poświaty ekranu*,
- wypełnianie konturów - *trudne*,
- kolor - *brak*,
- cena - *bardzo wysoka*.

Grafika



- ◆ **Cechy grafiki wektorowej** zorientowana obiektowo.
Punkty kontrolne określające kształt krzywych, **krzywe Beziera**; odtworzenie krzywej wymaga rozwiązania układu równań i obliczenia położenia wszystkich jej punktów.
Ścieżka: zbiór punktów wyznaczających linię na rysunku.
Kontur figury - ścieżka zamknięta.
Warstwy - połączone grupy obiektów
Obiekty zamknięte są w obwiedniach nazywanych **kopertami**.
Nałożenie koperty na jakiś obiekt może go zdeformować dopasowując do kształtu koperty, np. do gumowej powierzchni nadmuchanego balonu.
Wypełnienia: jednolite, gradientowe czyli tonalne (ciągłe przejście od jednej barwy do drugiej w sposób liniowy, radialnie lub stożkowo), sztafury (wypełnienia kreskowane lub powtarzającym się wzorem) oraz tekstury (podobne do naturalnych lub sztucznych materiałów takich jak włókna tkaniny, powierzchnia cegły czy chmury).
Tryb konturowy (wireframe), bez wypełnień i tekstur.

Grafika



Formaty graficzne:

EPS (Encapsulated Postscript, najczęstszy format, ale niezbyt dobrze realizowany przez większość programów, EPS dopuszcza nagłówki przechowujące obraz w formie rastrowej (zwykle w formacie TIF) przy niskiej rozdzielczości),

WMF (Windows MetaFile), tworzony przez narzędzia do rysowania np. w MS-Office

CDR (Corel Draw),

CGM (Computer Graphics Metafile, czyli metaplik grafiki komputerowej),

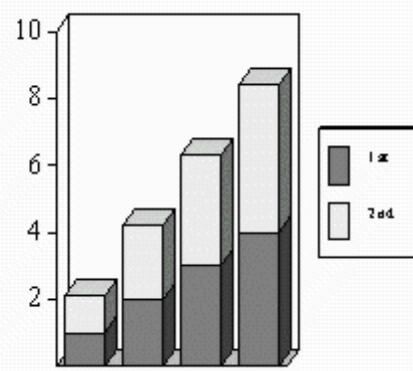
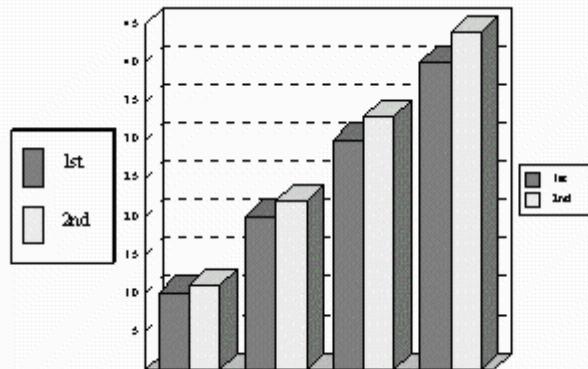
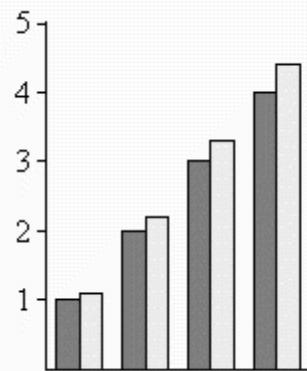
DRW (Micrografx Designer), PIC, PICT,

DXF (głównie do CAD, rysunków technicznych),

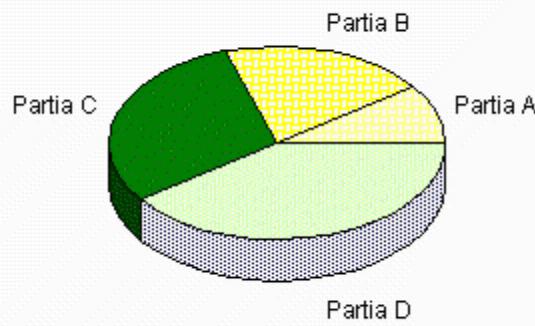
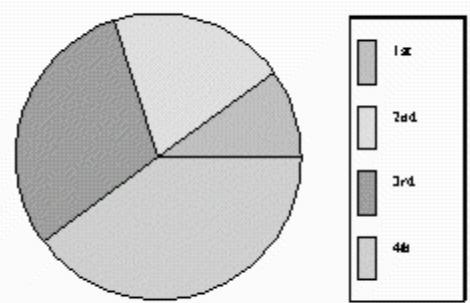
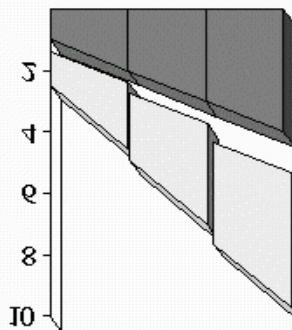
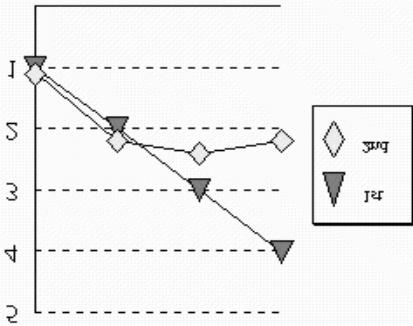
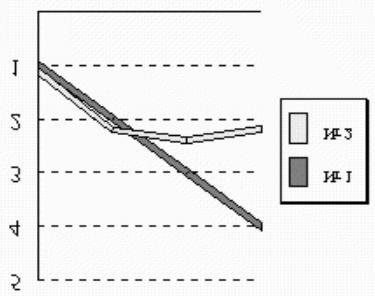
HPGL - do ploterów

Grafika

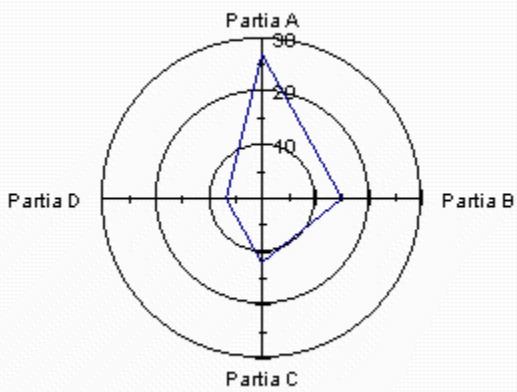
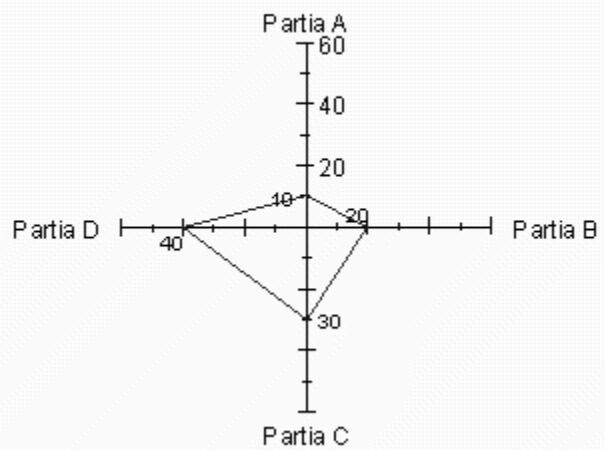
◆ **Grafika prezentacyjna** uproszczona grafika wektorowa.



Grafika



Grafika



Grafika

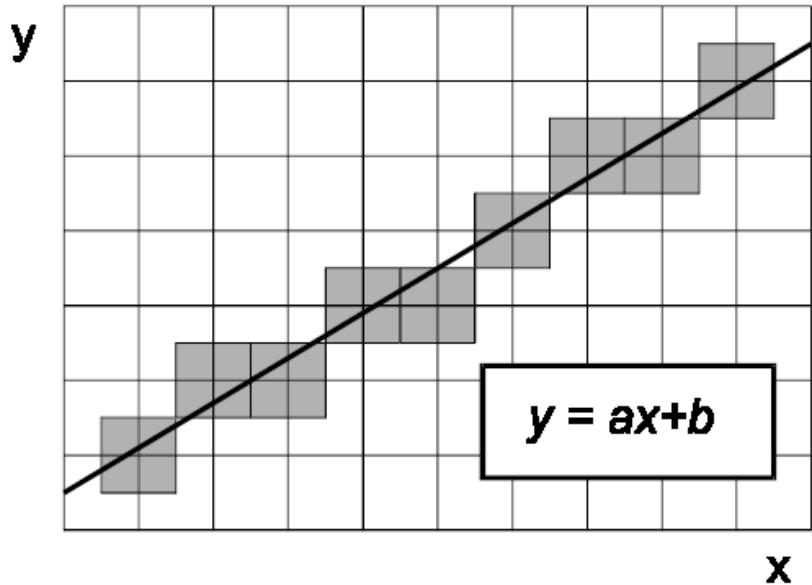
- ◆ **Komputerowo wspomagane projektowanie**
CAD, Computer Aided Design
CAE, Computer Aided Engineering,
CAM, Computer Aided Machinery
CIM, Computer Integrated Manufacturing
- ◆ Wyspecjalizowane programy do wspomagania projektowania w różnych dziedzinach:
SPICE, program do projektowania programowalnych układów logicznych;
PADS, symulator analogowych układów elektronicznych;
SUSIE, symulator działania złożonych układów logicznych;
EAGLE, program do projektowania obwodów drukowanych, ma polską instrukcję i pracuje nawet na PC-XT.
- ◆ Komputerowa animacja

Transformacje geometryczne w przestrzeni 2-D

- translacja,
- skalowanie,
- obrót.

Wizualizacja 2-D

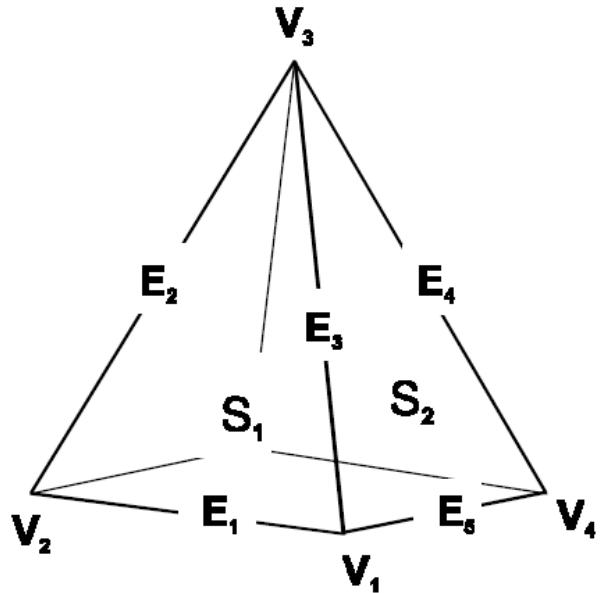
- przekształcenia układów współrzędnych,
- obcinanie.



Grafika 3-D Modele obiektów 3-D

- modele analityczne,
- modele szkieletowe,
- powierzchnie aproksymujące (Beziera, B-spline, NURBS),
- konstrukcyjna geometria bryłowa (*solid modeling*),
- drzewa ósemkowe,
- inne.

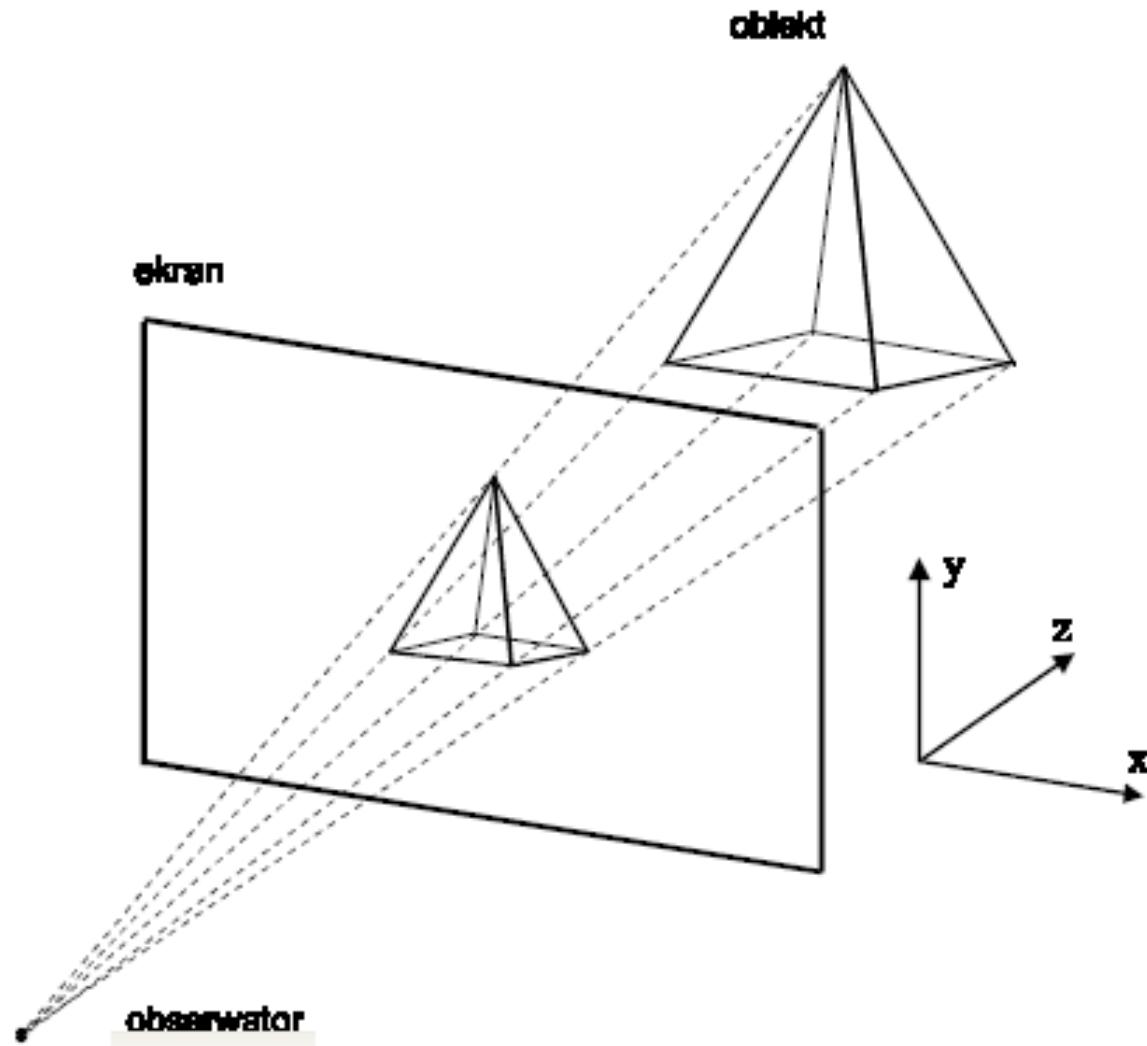
Model szkieletowy



Rzutowanie

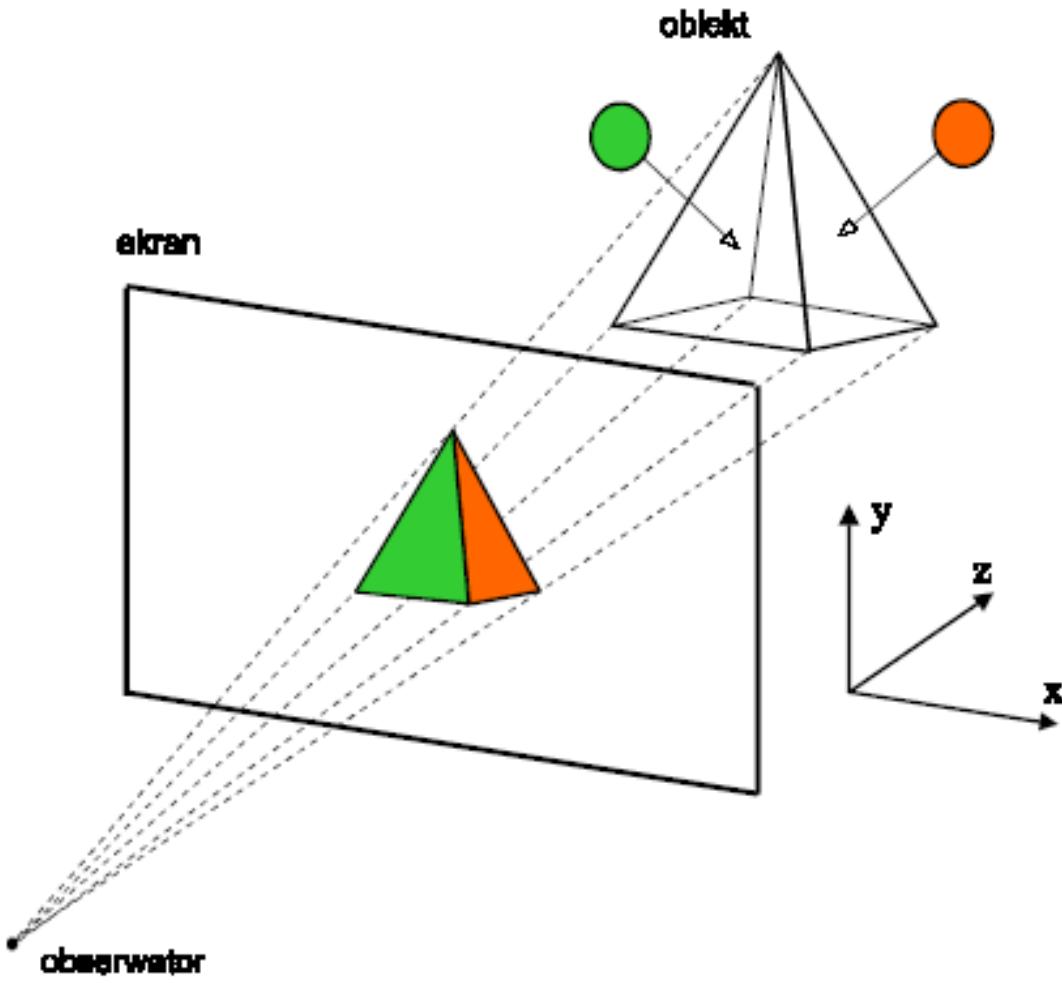
- rzut ukośny,
- rzut perspektywiczny

Rzut perspektywiczny



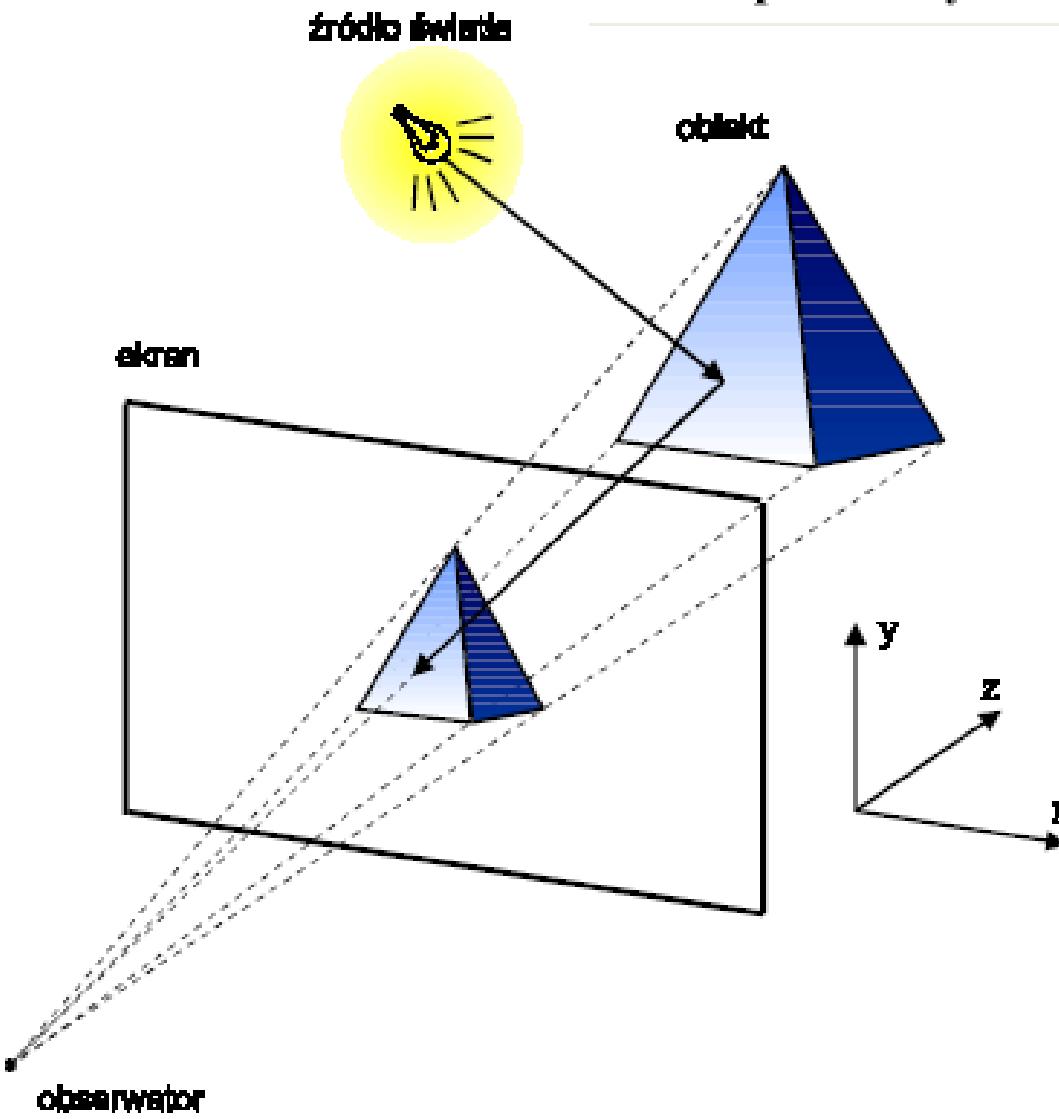
Usuwanie elementów niewidocznych

- sortowanie ścian,
- algorytm z-bufora.

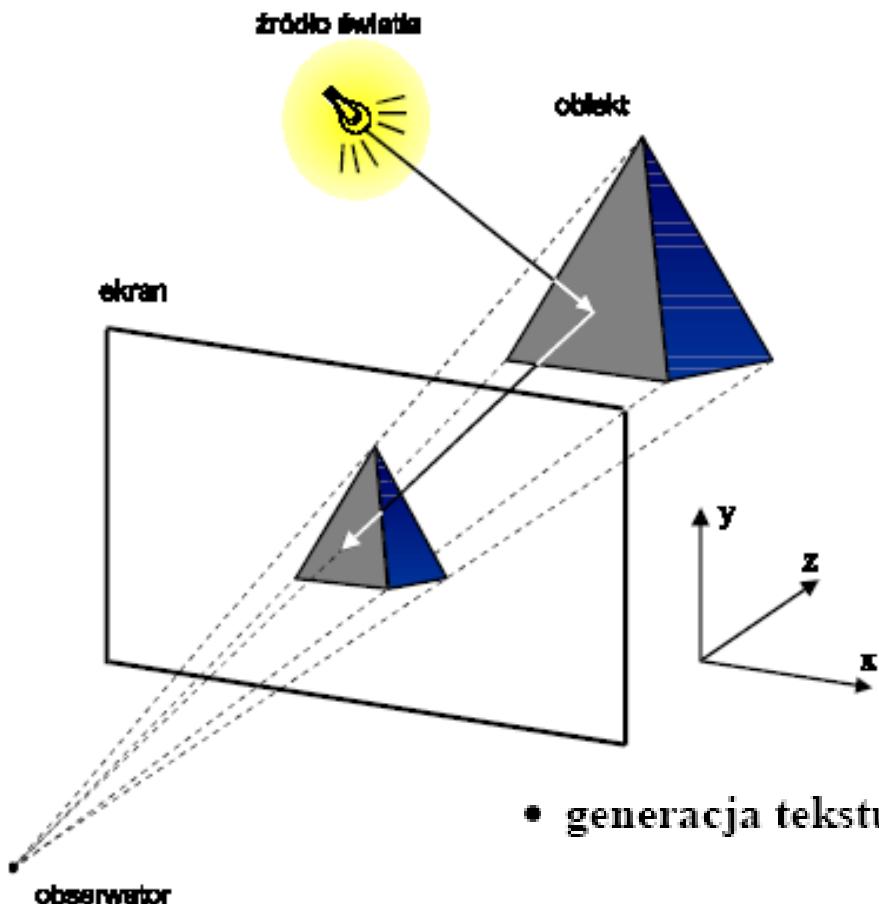


Oświetlanie obiektów

- powierzchnie lustrzane,
- powierzchnie matowe,
- ciała przezroczyste.



Teksturowanie obiektów

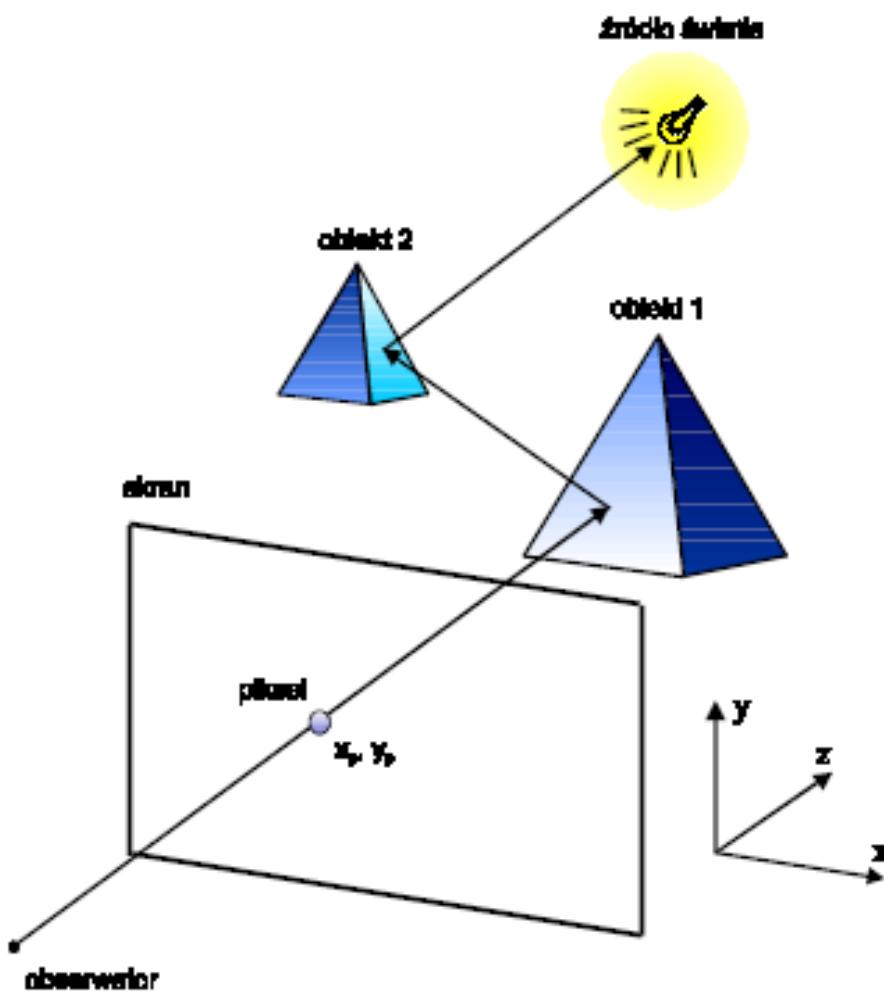


- generacja tekstur,
- nakładanie tekstur na ściany obiektów i generacja ich obrazu,
- problemy wynikające z niezgodności gęstości tekstuury i gęstości jej obrazu .

Generacja realistycznych obrazów scen 3-D

- metoda śledzenia promieni (*ray tracing*),
- metoda energetyczna (*radiosity*).

Metoda śledzenia promieni



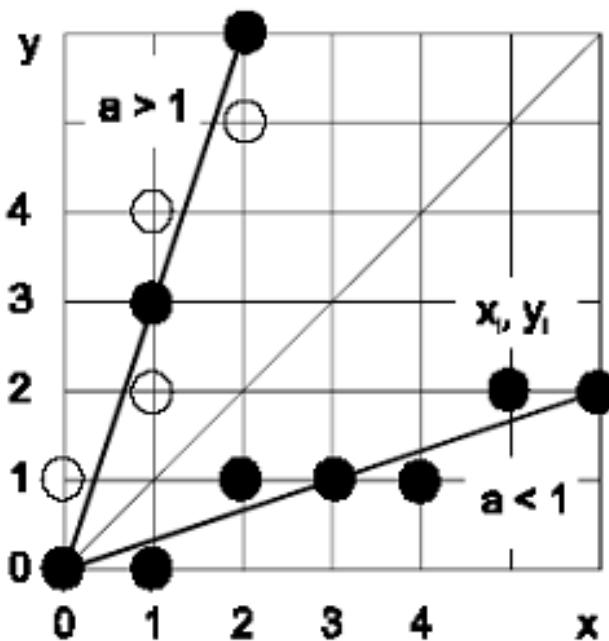
Rysowanie tworów pierwotnych w przestrzeni dyskretnej

Rysowanie odcinka prostej

$$y = ax + b$$

$$0 \leq a \leq 1$$

$$b = 0$$



Założenia

- Jeżeli $a > 1$ analizujemy równanie

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

i rysujemy punkty (y_i, x_i) .

- Jeżeli $b \neq 0$, rysujemy punkty $(x_i, y_i + \text{round}(b))$.

Odcinek rysujemy od punktu $(0, 0)$ do punktu (x_c, y_c) .

Algorytm 1 (bezpośrednie wykorzystanie równania prostej)

Krok 0

Rysujemy punkt ($0, 0$).

Krok i - ty

Dla $x_i = i$ obliczamy $y_i = y_{i-1} + a$ i rysujemy punkt $(x_i = i, \text{round}(y_i))$.

Ocena złożoności rysowania punktu:

- jedno mnożenie zmiennoprzecinkowe,
- jedno zaokrąglenie.

Algorytm 2 (przyrostowy)

Algorytm DDA (Digital Differential Analyzer)

$$y_{i-1} = ax_{i-1}$$

$$y_i = ax_i$$

$$y_i = a(x_{i-1} + 1)$$

$$y_i = ax_{i-1} + a = y_{i-1} + a$$

Krok 0

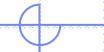
Rysujemy punkt (0, 0).

Krok i - ty

Obliczamy $y_i = y_{i-1} + a$ i rysujemy punkt
 $(x_i = i, \text{round}(y_i))$.

Ocena złożoności rysowania punktu:

- jedno dodawanie zmiennoprzecinkowe,
- jedno zaokrąglenie.



Algorytm 3

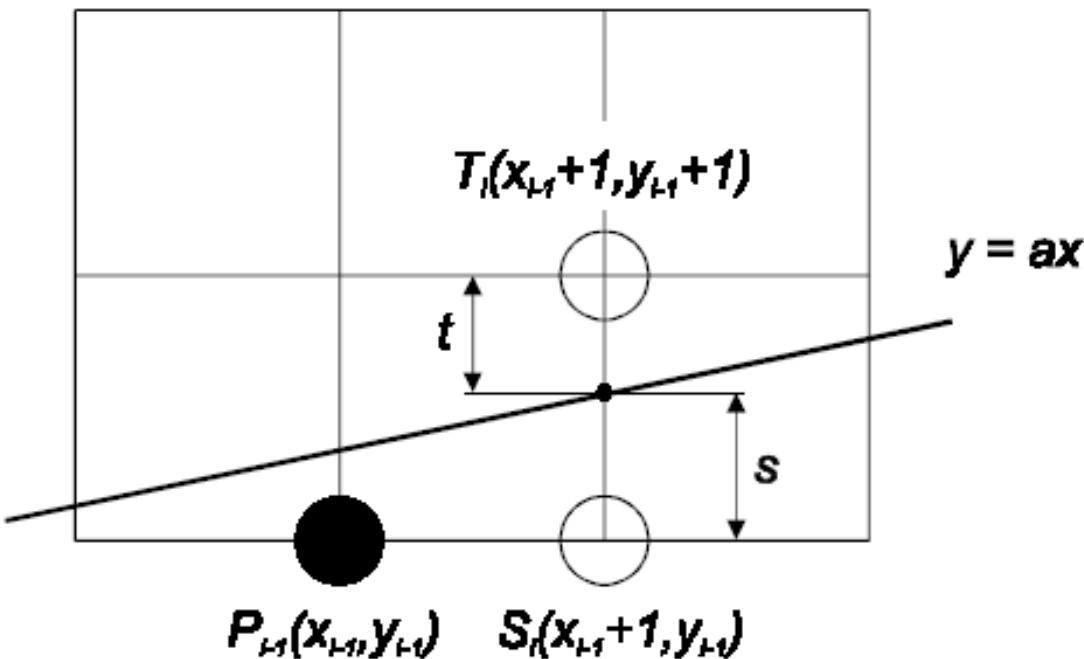
Algorytm Bresenhama

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

Założenie dodatkowe:

$$a = \frac{dy}{dx}$$

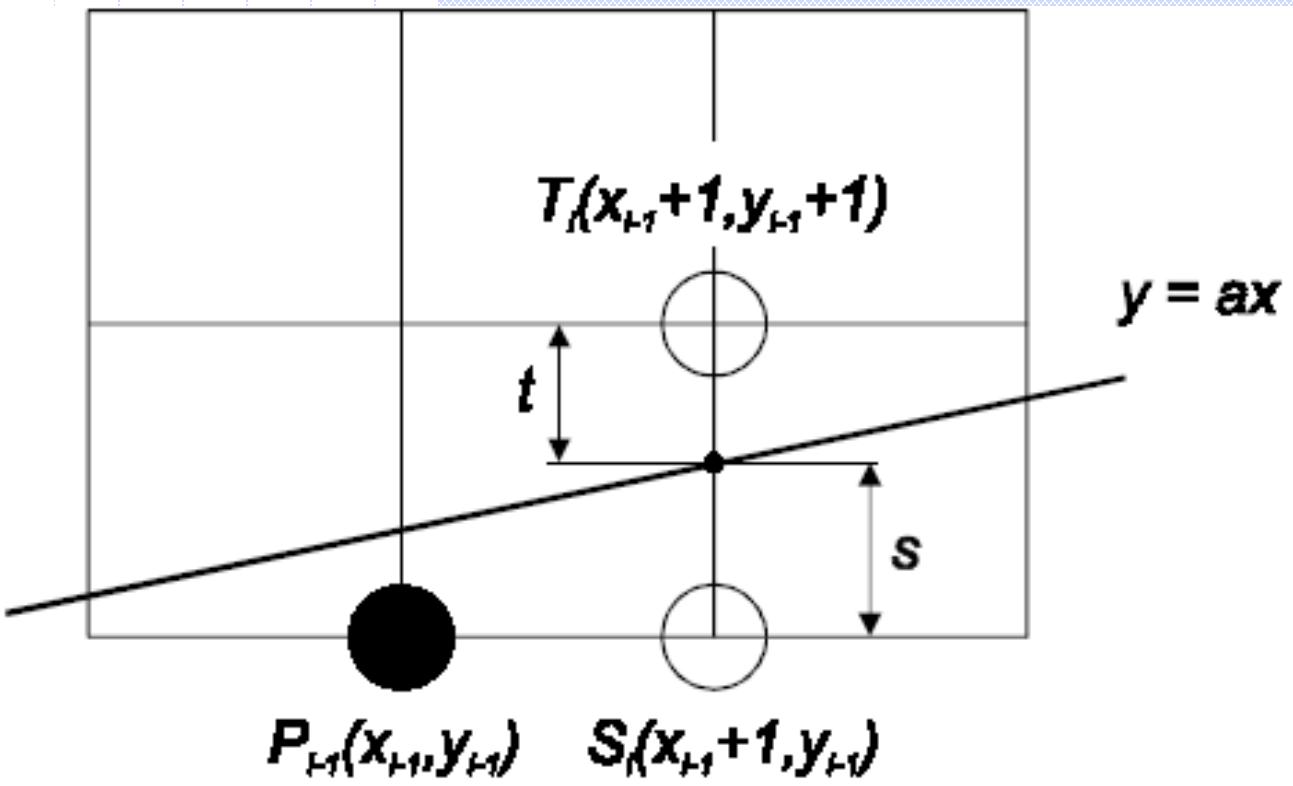
$$dx, dy \in N$$



Punkt P_{i-1} został wyznaczony.

Należy podjąć decyzję co do wyboru następnego punktu:

- Jeżeli $s < t$ ($(s - t) < 0$) to wybieramy punkt S_i ,
- Jeżeli $s \geq t$ ($(s - t) \geq 0$) to wybieramy punkt T_i .



Z rysunku

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

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

a więc,

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

Inaczej kryterium wyboru punktu wyraża się jako

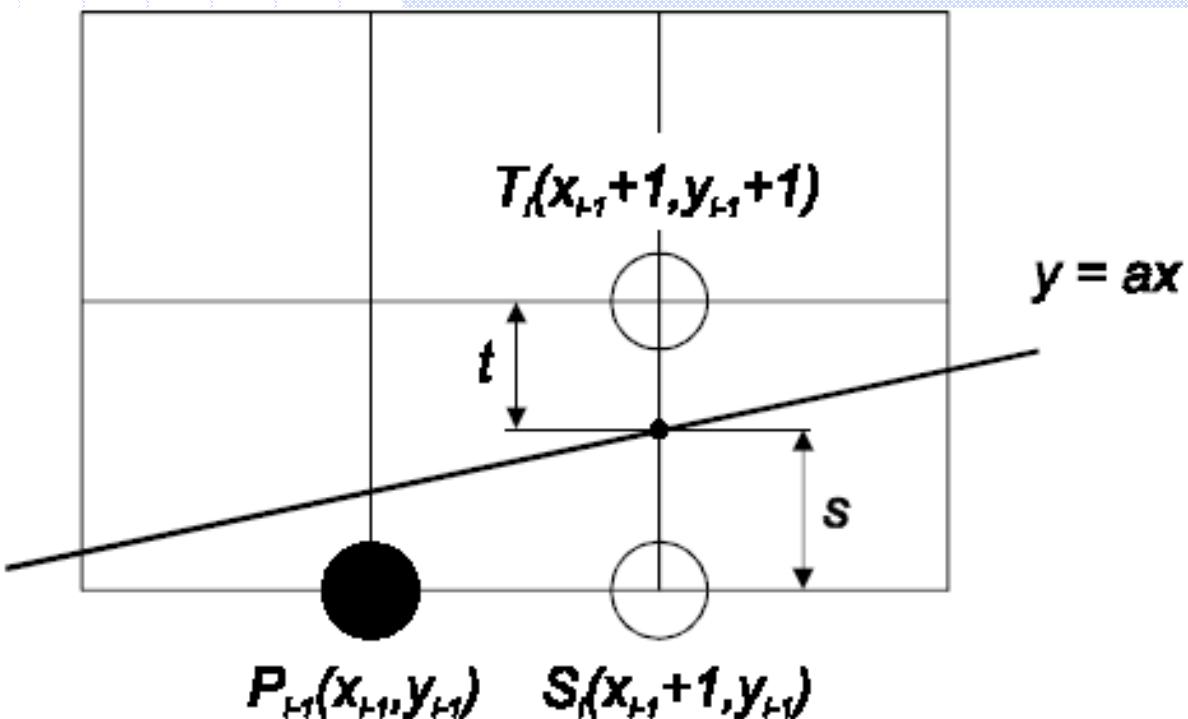
$$dx(s-t) = 2(x_{i-1}dy - y_{i-1}dx) + 2dy - dx.$$

Po podstawieniu $d_i = dx(s-t)$ uzyskuje się

$$d_i = 2x_{i-1}dy - 2y_{i-1}dx + 2dy - dx. \quad (1)$$

Zwiększając w formalny sposób indeksy ($i+1 \rightarrow i$) otrzymuje się

$$d_{i+1} = 2x_idy - 2y_idx + 2dy - dx \quad (2)$$



Odejmując stronami od równania (2) równanie (1) uzyskuje się związek między wartościami kryterium wyboru dla i - tego i i+1- go kroku

$$d_{i+1} - d_i = 2dy(x_i - x_{i-1}) - 2dx(y_i - y_{i-1})$$

Wiadomo, że $x_i - x_{i-1} = 1$, więc

$$d_{i+1} = d_i + 2dy - 2dx(y_i - y_{i-1}) \quad (3)$$

1. Jeżeli $d_i \geq 0$ ($s \geq t$) to wybiera się T_i , wtedy
 $y_i = y_{i-1} + 1$ oraz

$$d_{i+1} = d_i + 2(dy - dx)$$

2. Jeżeli $d_i < 0$ ($s < t$) to wybiera się S_i , wtedy
 $y_i = y_{i-1}$ oraz

$$d_{i+1} = d_i + 2dy$$

3. Podstawiając do (1) $(x_0, y_0) = (0, 0)$, uzyskuje się kryterium wyboru dla kroku 1

$$d_1 = 2dy - dx$$

Krok 0

Rysujemy punkt (0, 0) i obliczamy

$$d_1 = 2dy - dx$$

Krok i - ty

Dla kryterium d_i , obliczonego w kroku poprzednim jeżeli:

1. $d_i \geq 0$ rysujemy T_i (punkt "wyższy")
i obliczamy

$$d_{i+1} = d_i + 2(dy - dx)$$

2. $d_i < 0$ rysujemy S_i (punkt "równy")
i obliczamy

$$d_{i+1} = d_i + 2dy$$

Ocena złożoności rysowania punktu:

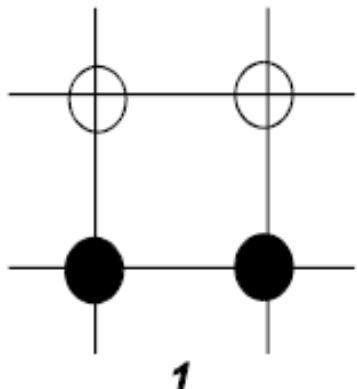
- jedno dodawanie stałoprzecinkowe,
- jedno mnożenie przez 2 (przesunięcie),
- jedno porównanie,

Algorytm 4 Algorytm z podwójnym krokiem

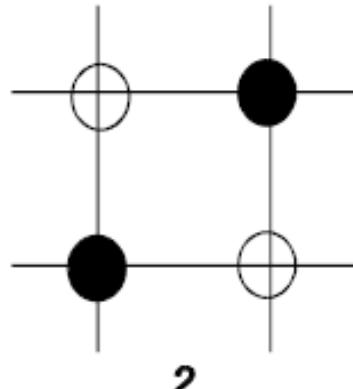
Wu X., Ronke J.G, *Double-Step Incremental Generation of Lines and Circles*, CVGIP, (37), 1987.

Idea algorytmu:

- W algorytmie Bresenhama dokonuje się w każdym kroku wyboru następnego punktu.
Możliwe są dwa przypadki.

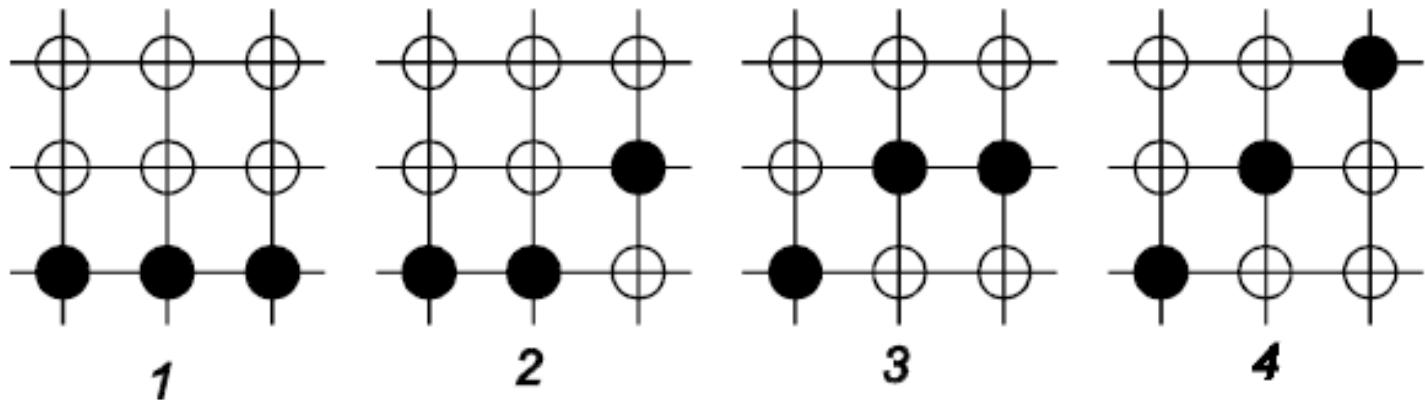


1



2

- W algorytmie z podwójnym krokiem wybiera się następną parę punktów.
Możliwe są cztery przypadki.



Rysowanie okręgu



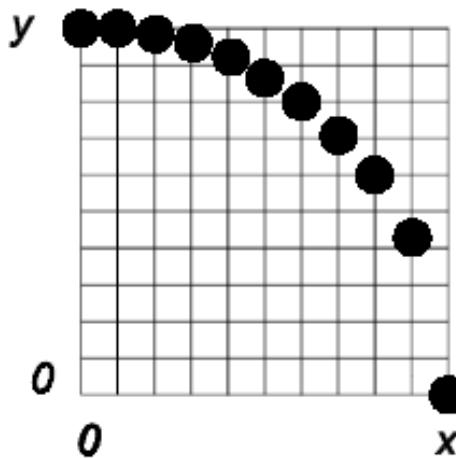
Założenia

- środek okręgu w punkcie (0, 0),
- promień okręgu wynosi r .

Algorytm 1 (bezpośrednie użycie równania okręgu)

$$x^2 + y^2 = r^2$$

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





Algorytm 2 (równanie parametryczne)

$$\begin{aligned}x(\phi) &= r \cos \phi & 0 \leq \phi \leq 2\pi \\y(\phi) &= r \sin \phi\end{aligned}$$

Rysujemy punkty:

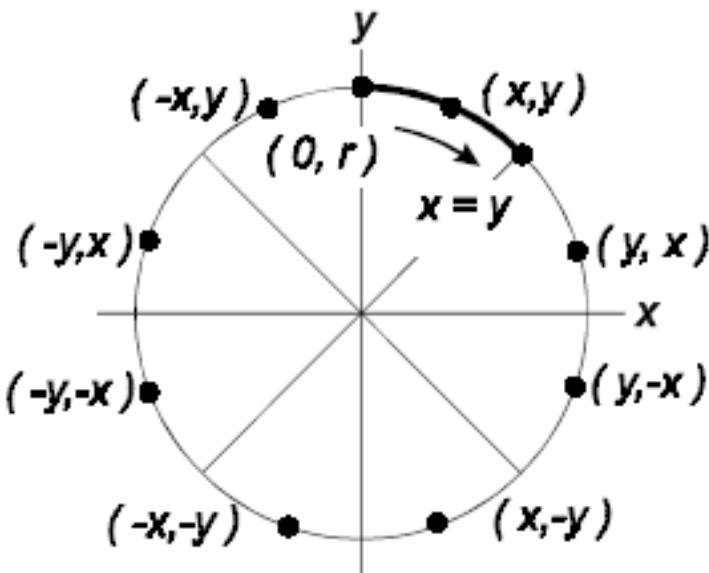
$$(\text{round}(x(i \cdot \Delta\phi)), \text{round}(y(i \cdot \Delta\phi))) \quad i = 0, 1, 2, \dots$$

$$\Delta\phi = f(r) = ?$$

Algorytm 3 Algorytm Bresenhama

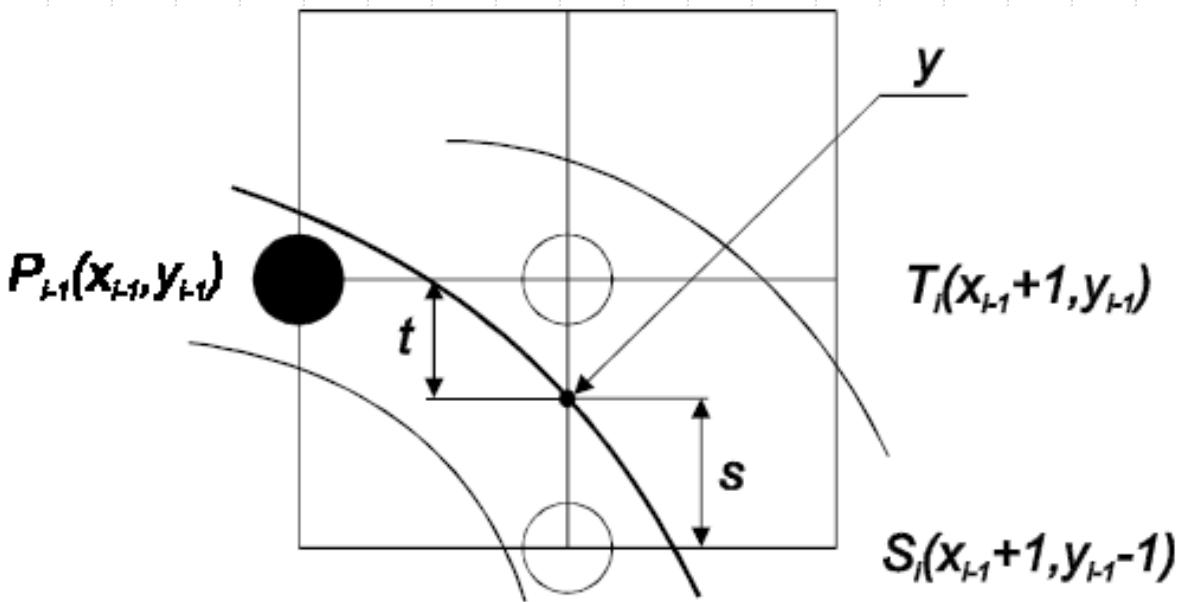
Bresenham J.E., *A Linear Algorithm for Incremental Digital Display of Arcs*, Communications of ACM, 20, 1977.

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.

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

Punkt P_{i-1} został wyznaczony.

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

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

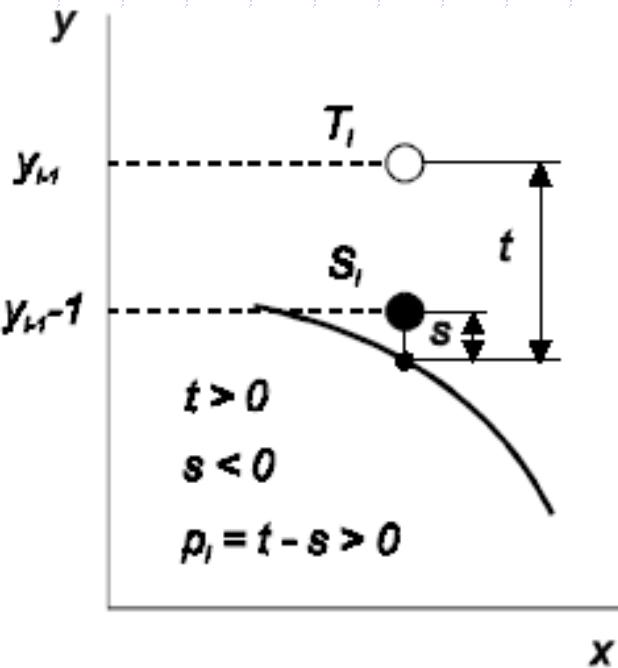
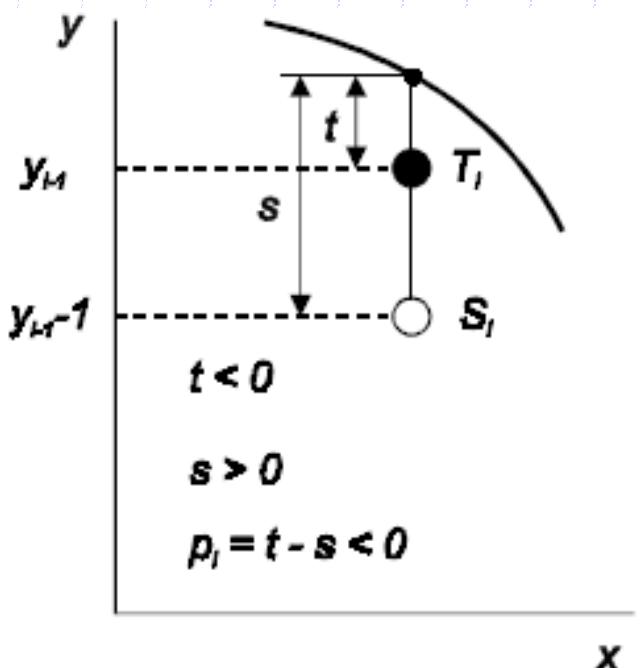
$$s = y^2 - (y_{i-1} - 1)^2 = r^2 - (x_{i-1} + 1)^2 - (y_{i-1} - 1)^2$$

Kryterium decyzyjne dla i-tego kroku wyraża się jako.

$$p_i = t - s$$

Należy podjąć decyzję co do wyboru następnego punktu:

- **Jeżeli $p_i < 0$ ($s > t$) to wybieramy punkt T_i ,**
- **Jeżeli $p_i \geq 0$ ($s \leq t$) to wybieramy punkt S_i .**



Wiadomo, że $p_i = t - s$ więc

$$p_i = 2(x_{i-1} + 1) + y_{i-1}^2 + (y_i - 1)^2 - 2r^2 \quad (1)$$

Zwiększać w formalny sposób indeksy ($i+1 \rightarrow i$) otrzymuje się

$$p_{i+1} = 2[(x_{i-1} + 1) + 1]^2 + y_i^2 + (y_i - 1)^2 - 2r^2 \quad (2)$$

Odejmując stronami od równania (2) równanie (1) i wykonując proste przekształcenia uzyskuje się związek między wartościami kryterium wyboru dla i -tego i $i+1$ -go kroku w postaci

$$p_{i+1} = p_i + 4x_{i-1} + 6 + 2(y_i - y_{i-1}) - 2(y_i - y_{i-1})$$

1. Jeżeli $p_i < 0$ to wybiera się T_i , wtedy $y_i = y_{i-1}$ czyli

$$p_{i+1} = p_i + 4x_{i-1} + 6$$

2. Jeżeli $p_i \geq 0$ to wybiera się S_i , wtedy $y_i = y_{i-1} - 1$ czyli

$$p_{i+1} = p_i + 4(x_{i-1} - y_{i-1}) + 10$$

3. Podstawiając do (1) $(x_0, y_0) = (0, r)$, uzyskuje się kryterium wyboru dla kroku 1

$$p_1 = 3 - 2r$$

Krok 0

Rysujemy punkt $(0, r)$ i obliczamy

$$p_1 = 3 - 2r$$

Krok i - ty

Dla kryterium p_i obliczonego w kroku poprzednim jeżeli:

1. $p_i < 0$ rysujemy T_i (punkt "równy")
i obliczamy

$$p_{i+1} = p_i + 4x_{i-1} + 6$$

2. $p_i \geq 0$ rysujemy S_i (punkt "niższy")
i obliczamy

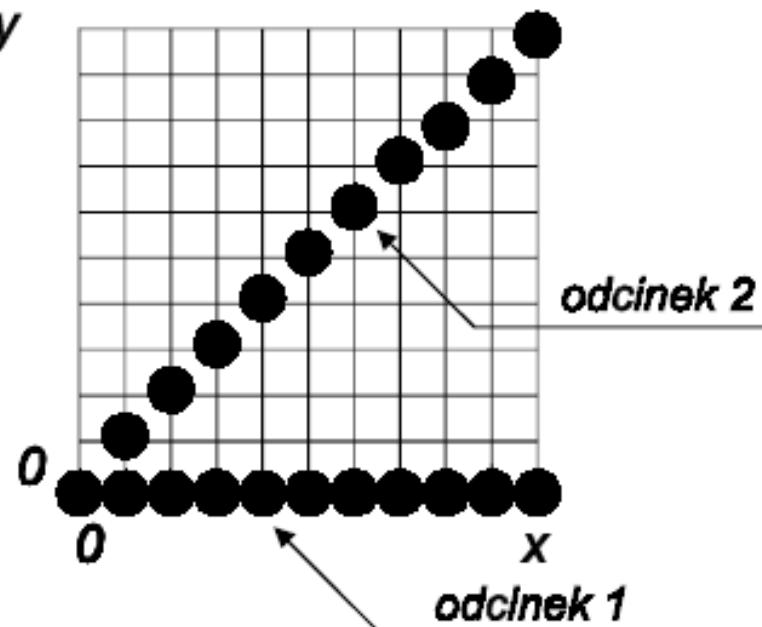
$$p_{i+1} = p_i + 4(x_{i-1} - y_{i-1}) + 10$$

Krok i-ty powtarzamy aż osiągniemy $x = y$

Rysowanie prymitywów - zagadnienia specjalne

Zmiana jasności odcinka w funkcji nachylenia

Przykład:



- *odcinek 2* jest $\sqrt{2}$ razy dłuższy od *odcinka 1*,
- punkty *odcinka 2* należy wyświetlić $\sqrt{2}$ razy jaśniej niż punkty *odcinka 1*.

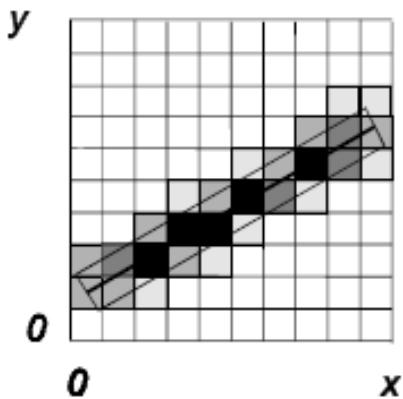
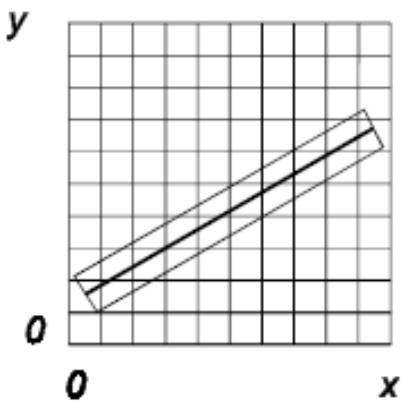
Usuwanie zakłóceń

Aliasing - zniekształcenie obrazu powstałe w wyniku próbkowania z małą częstotliwością obrazu o wysokiej częstotliwości.

Antialiasing - usuwanie zniekształceń obrazu powstających w wyniku aliasingu.

Metody eliminacji zniekształceń aliasingowych

- zwiększenie częstotliwości próbkowania,
- próbkowanie bezwagowe,



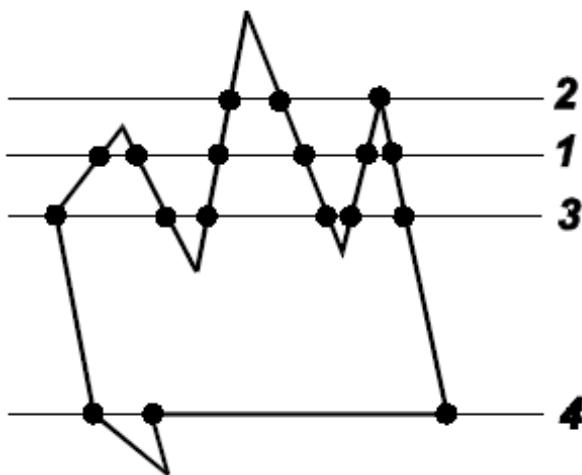
jasność piksela jest proporcjonalna do pokrytej przez piksel powierzchni prostokąta

Wypełnianie konturów

Wypełnianie wieloboków

Zasada parzystości:

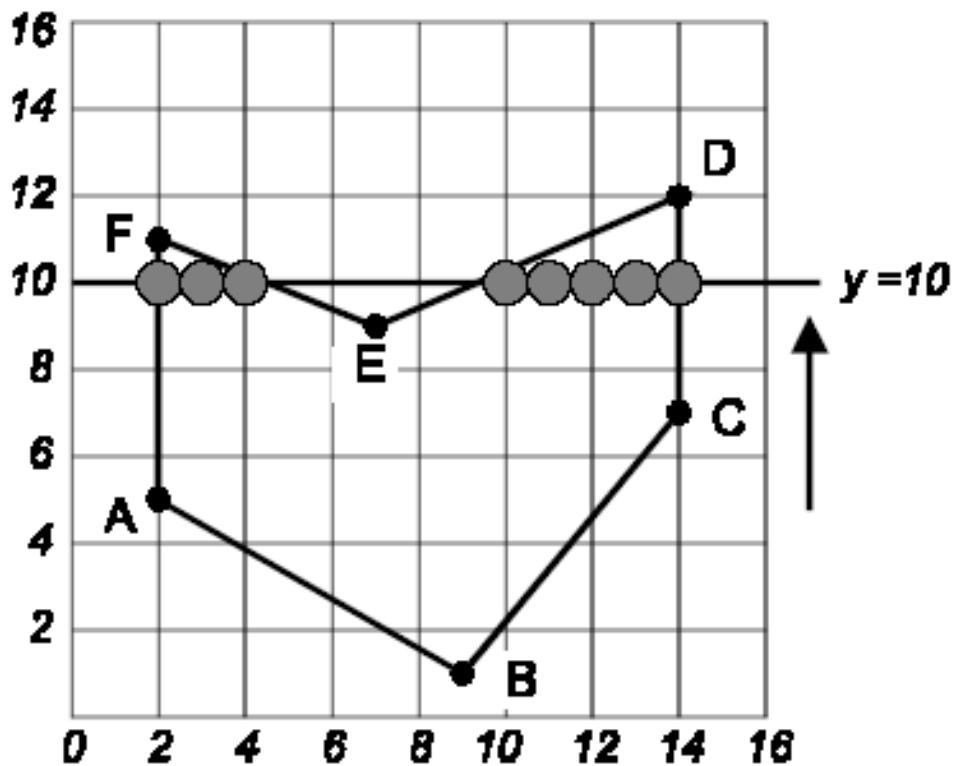
Prosta nie przechodząca przez wierzchołek przecina wielobok parzystą ilość razy.



- 1 - prosta nie przechodzi przez wierzchołek, liczba przecięć wynosi 6.
- 2 - prosta przechodzi przez wierzchołek, liczba przecięć wynosi 3.
- 3 - prosta przechodzi przez wierzchołek, liczba przecięć wynosi 6.
- 4 - prosta pokrywa się z krawędzią poziomą.

Algorytm przeglądania linii (*scan line algorithm*)

Dany jest wielobok (bez krawędzi poziomych) opisany jako zbiór krawędzi .



Opis wieloboku:

Krawędź	y_{min}	y_{max}	x_{min}	x_{max}
AB	1	5	9	2
BC	1	7	9	14
CD	7	12	14	14
DE	9	12	7	14
EF	9	11	7	2
FA	5	11	2	2

Zasada wypełniania:

W przeglądanej linii należy wypełniać punkty pomiędzy poszczególnymi parami krawędzi .

Na rysunku dla $y = 10$ między FA - EF i DE - CD.

Opis algorytmu

Krok 0

Utworzyć globalną tablicę krawędzi (ET).

Krok 1

Ustawić y na najmniejszej wartości współrzędnej y z globalnej tablicy krawędzi (ET), czyli y dla pierwszej niepustej grupy krawędzi.

Krok 2

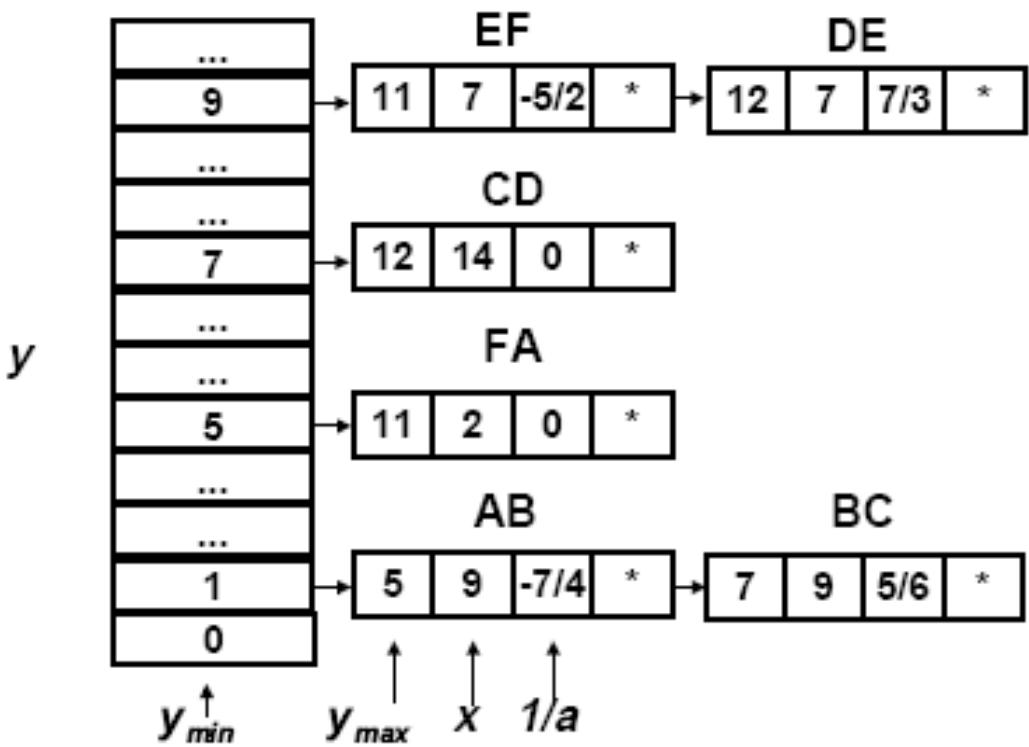
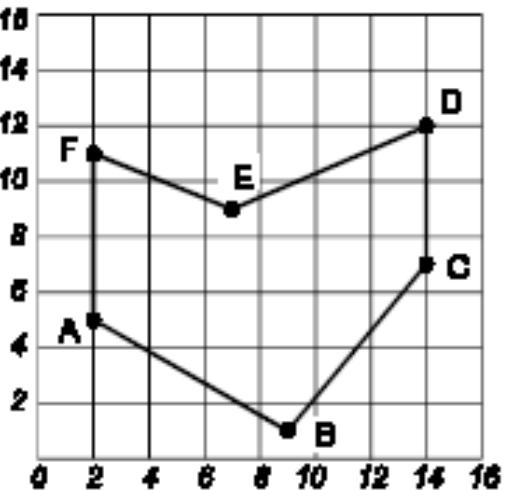
Wyzerować aktywną tablicę krawędzi (AT).

Krok 3

Powtarzać tak długo, dopóki tablica globalna (ET) i tablica aktywna (AT) nie będą puste.

- Przenieść z grupy y tablicy globalnej (ET) do tablicy aktywnej (AT) te krawędzie, dla których $y_{min} = y$ i posortować je ze względu na x.
- Wypełnić piksele w linii y, wykorzystując pary x z tablicy aktywnej (AT).
- Usunąć z tablicy aktywnej (AT) te krawędzie, dla których $y = y_{max}$.
- Zwiększyć y o 1 (następna linia).
- Dla każdej pary krawędzi, która nie jest pionowa wyliczyć i wstawić do tablicy aktywnej (AT) nowe wartości x.

Krok 0 - Tworzenie globalnej tablicy krawędzi (ET)



Kolejność porządkowania: $x_{min}y_{max}$

- minimum y_{min} - przypisanie do grupy
- minimum x_{min} - porządkowanie w grupie
- minimum y_{max} - porządkowanie w grupie
- minimum x_{max} - porządkowanie w grupie

Krok 1

Ustawiamy y na najmniejszej wartości współrzędnej y z globalnej tablicy krawędzi.

$$y = 1$$

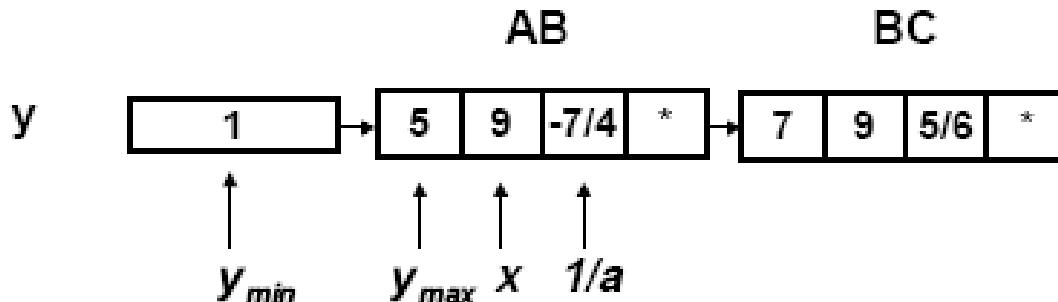
Krok 2

Zerujemy aktywną tablicę krawędzi (AT).

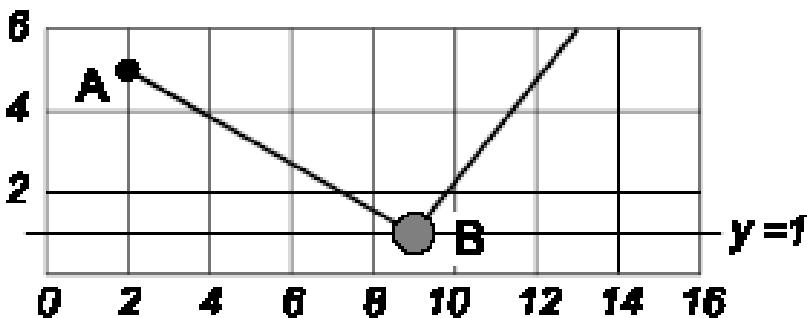
Krok 3

Przenosimy z grupy y tablicy globalnej (ET) do tablicy aktywnej (AT) te krawędzie, dla których $y_{min} = y$ i sortujemy krawędzie ze względu na x .

(AT)



- Wypełniamy piksele w linii y , wykorzystując pary x z tablicy aktywnej (AT) i zasadę parzystości.



- Usuwamy z tablicy aktywnej (AT) te krawędzie, dla których $y = y_{max}$, (brak takich krawędzi)
- Zwiększamy y o 1 ($y = 2$).

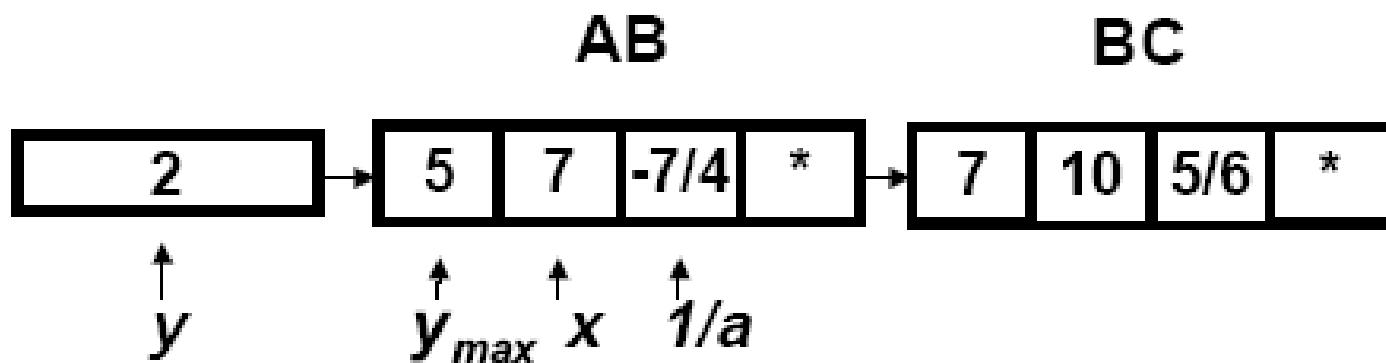
- Dla każdej pary krawędzi, która nie jest pionowa wyliczamy i wstawiamy do tablicy aktywnej (AT) nowe wartości x .

Sposób obliczania nowych wartości x :

a)
$$x(y) = x(y - 1) + \frac{1}{a} \quad (\text{algorytm DDA}),$$

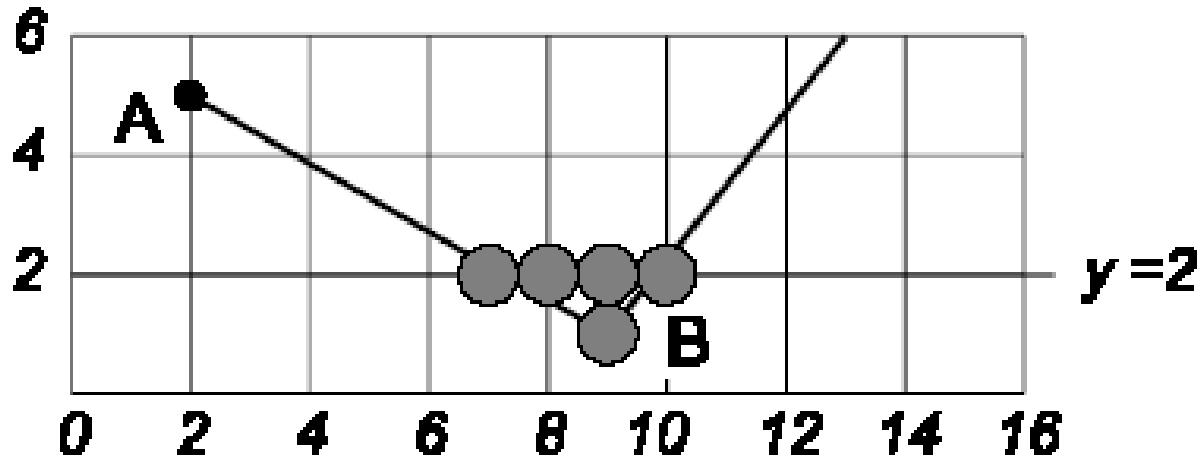
b) algorytm Bresenhama,

(AT)



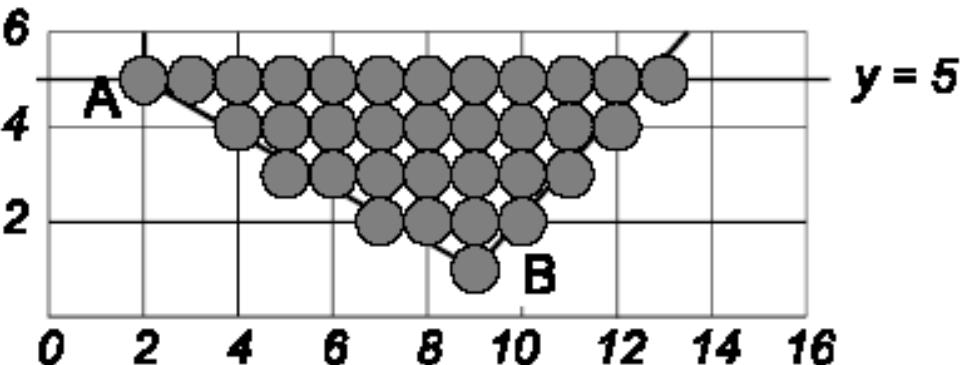
Koniec pierwszego przebiegu dla kroku 3

W kolejnym przebiegu uzyskamy rysunek :



Postępujemy analogicznie aż do osiągnięcia $y = 5$

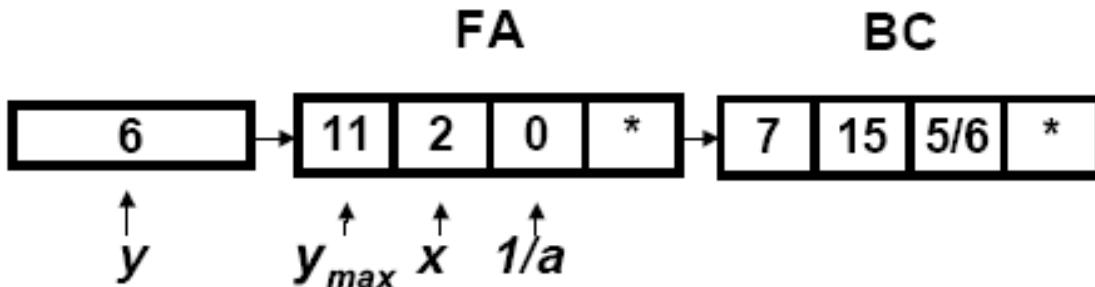
Dla $y = 5$ rysunek wygląda następująco:



Dla krawędzi AB zachodzi $y = y_{max}$ (koniec krawędzi).
Usuwamy AB z tablicy aktywnej (AT).

Przenosimy do tablicy aktywnej z tablicy globalnej
(ET) krawędź FA, dla której $y_{min} = y$.

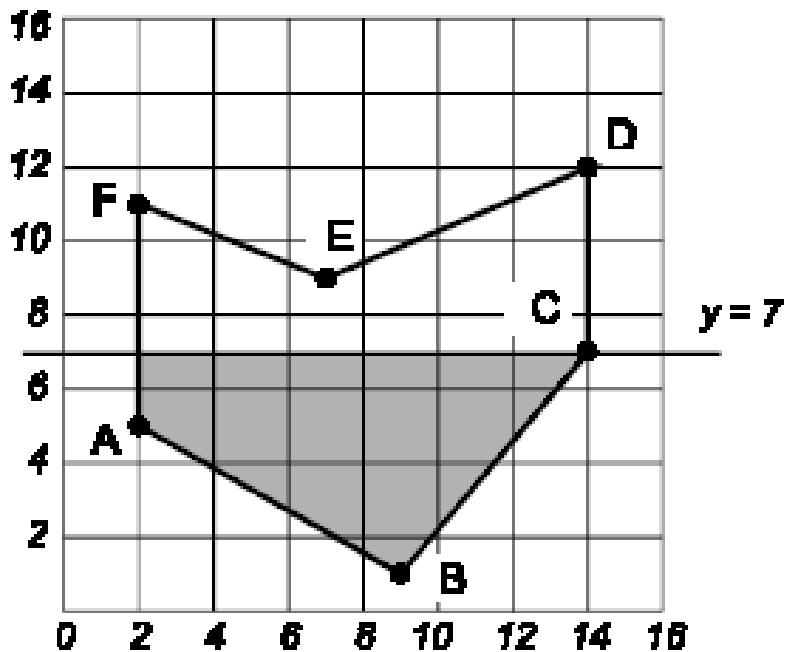
Porządkujemy tablicę (AT) ze względu na x.
Tablica aktywna przybiera postać:



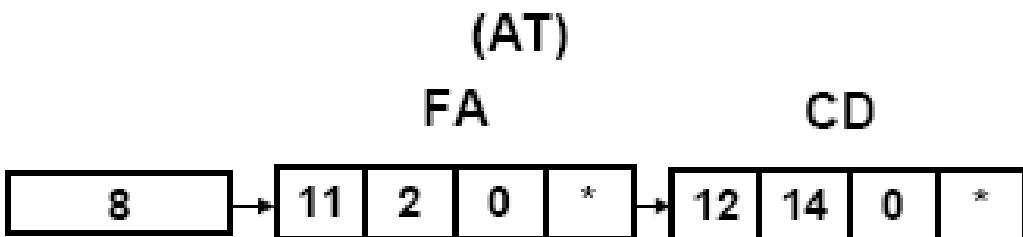
Kontynuujemy wypełnianie aż osiągniemy koniec krawędzi występującej w tablicy aktywnej (AT), lub początek krawędzi z tablicy globalnej (ET).

W rozważanym przypadku dla $y = 7$, osiągamy koniec krawędzi BC.

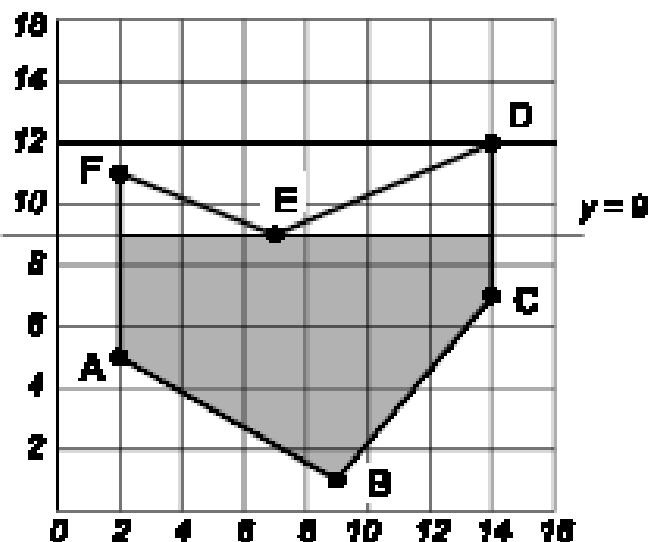
Uzyskany efekt wypełnienia pokazuje rysunek.



Po usunięciu z tablicy aktywnej (AT) krawędzi BC,
zwiększeniu y i przeniesieniu z tablicy globalnej (ET)
krawędzi CD tablica aktywna przyjmuje postać:

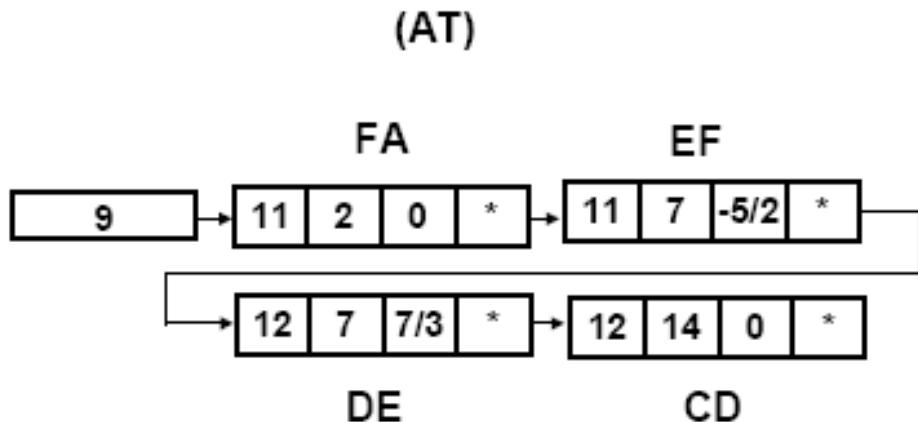


Wypełniając dalej, osiągamy wartość $y = 9$. Rysunek wygląda następująco:

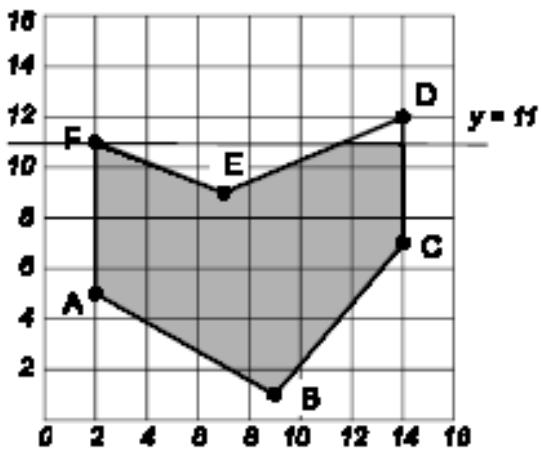


Stwierdzamy, że dla $y = 9$ para krawędzi EF i DE spełnia warunek $y_{min} = y$.

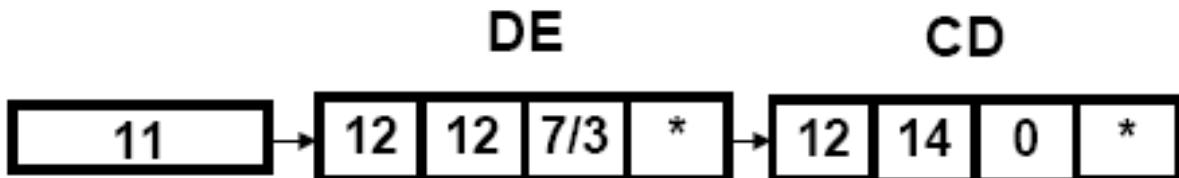
W związku z tym przenosimy krawędzie EF i DE do Tablicy aktywnej (AT), która po uporządkowaniu elementów ze względu na X przybiera postać:



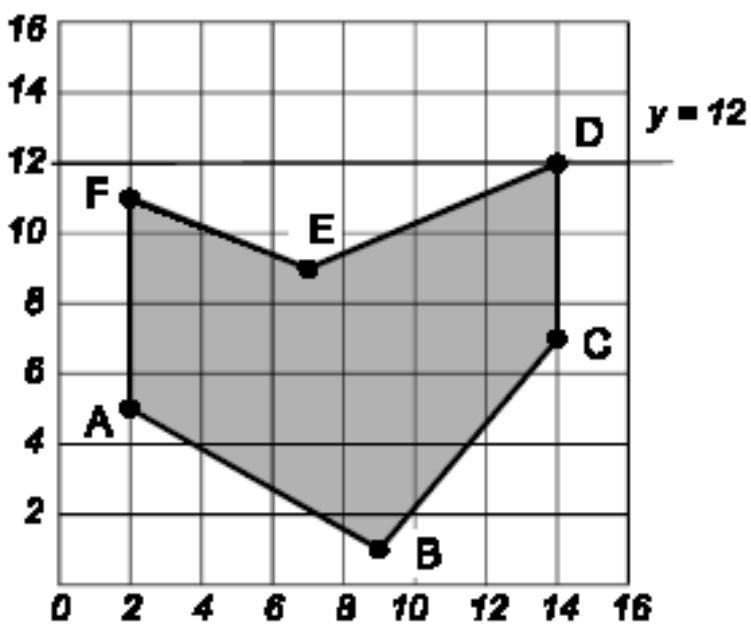
Tablica globalna (ET) staje się pusta. Kontynuujemy wypełnianie między parami krawędzi z tablicy (AT), aż do osiągnięcia $y = 11$. Uzyskujemy efekt jak



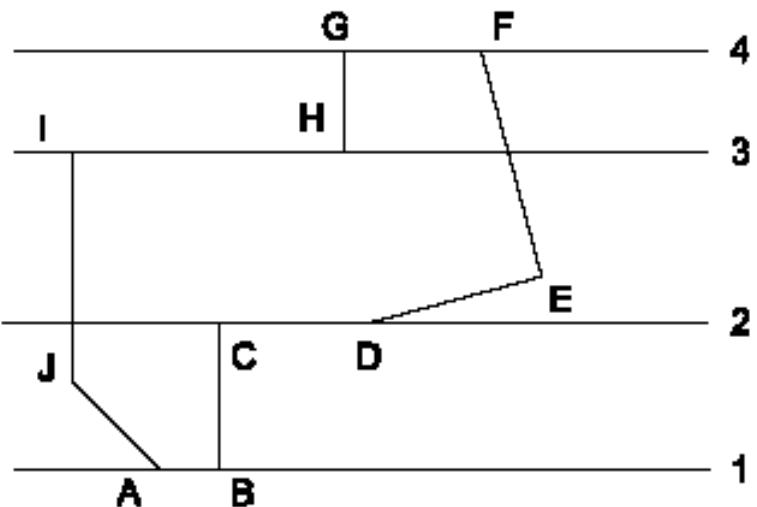
Usuwamy z tablicy (AT) parę krawędzi FA, EF.
(AT)



Wypełniamy następną linię i osiągamy końce krawędzi DE i CD. Usuwany krawędzie z tablicy. Tablica (AT) jest już pusta, czyli kończymy wypełnianie. Ostateczny efekt pokazuje rysunek.



Problem krawędzi poziomych:



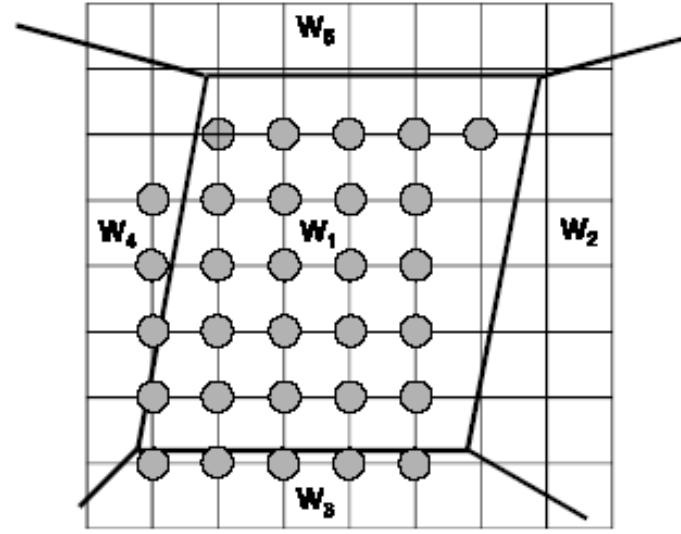
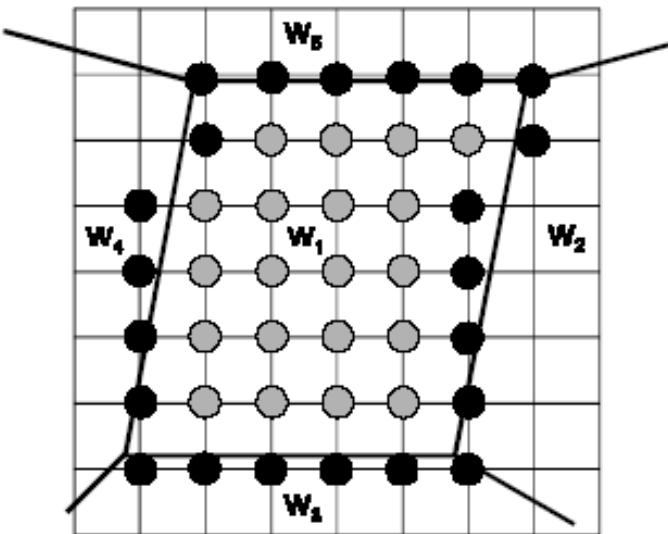
Rozwiązanie:

Pominięcie krawędzi poziomych w tablicy (AT).

Przykłady:

- 1 - w tablicy (AT) znajdzie się AJ i BC,
- 2 - w tablicy (AT) znajdzie się IJ i DE,
- 3 - w tablicy (AT) znajdzie się IJ i EF,
- 4 - w tablicy (AT) znajdzie się GH i FE,

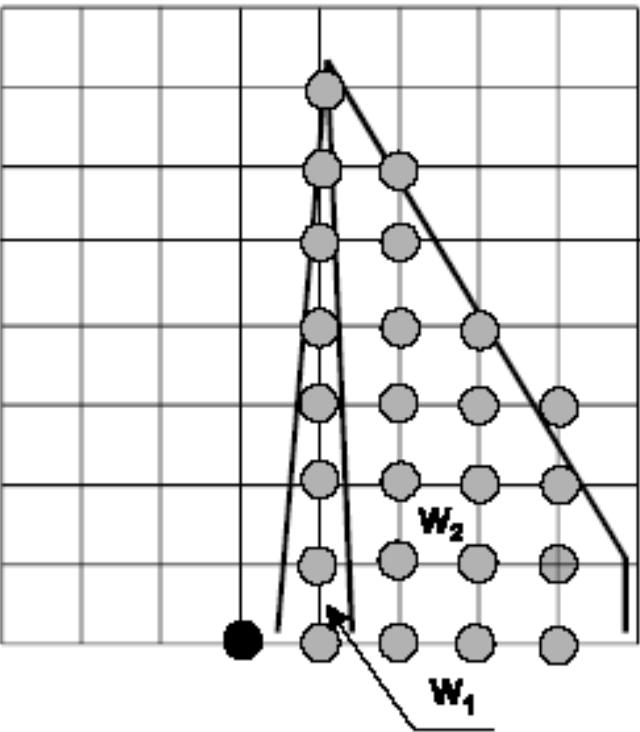
Problem brzegu wieloboku:



Rozwiązanie:

- rysować piksele leżące wewnątrz wieloboku, ale nie na brzegu,
- rysować piksele należące do lewej krawędzi,
- rysować piksele należące do dolnej krawędzi.

Problem wieloboków "bardzo wąskich":



Zastosowano poprzednio opisaną konwencję rysowania.

Obraz wieloboku w_1 składa się tylko z jednego punktu.

Brak zadawalającego rozwiązania przy tym sposobie rysowania.

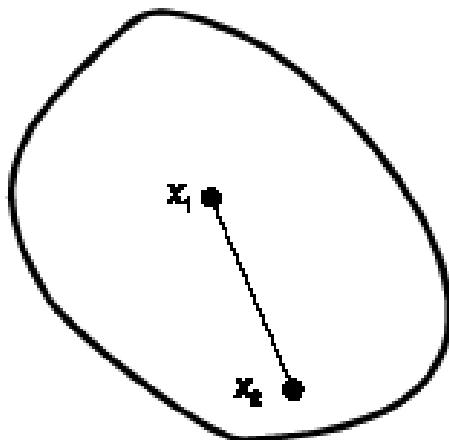
Należy zastosować wypełnianie wielotonowe.

Wypełnianie konturu zadanego w postaci obrazu

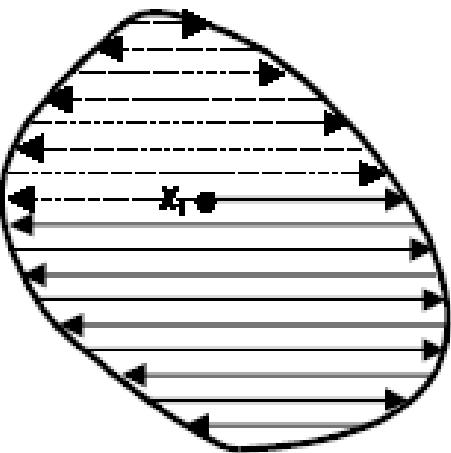
Dany jest kontur i punkt leżący wewnątrz konturu.

Kontur wypukły

Dla dowolnej pary punktów wewnętrznych odcinek, którego końcami są te punkty leży w całości wewnątrz konturu.



Algorytm wypełniania konturu wypukłego



- wypełniamy w linii poczynając od punktu startowego, aż do prawej granicy konturu,
- znajdujemy "niższy" punkt konturu i wypełniamy linię do lewej granicy konturu,
- powtarzamy tak długo, dopóki możemy znaleźć punkt "niższy",
- wracamy do punktu startowego i kontynuujemy proces poruszając się "w górę".

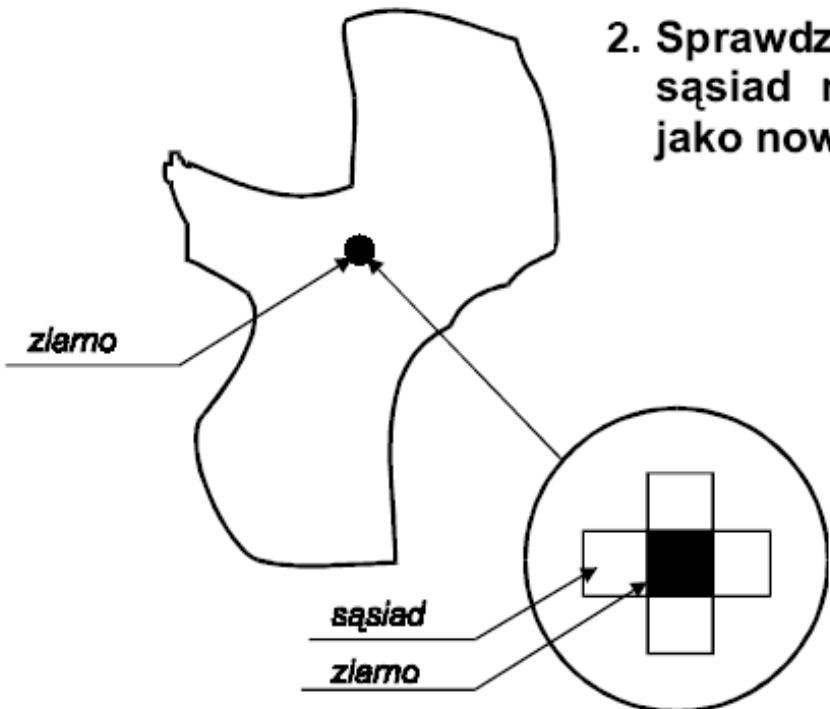
Kontur niewypukły

Wypełnianie przez spójność:

Algorytm:

1. Wypełniane jest ziarno,

2. Sprawdzani są kolejno sąsiedzi, jeżeli sprawdzany sąsiad nie należy do konturu, przyjmowany jest jako nowe ziarno i następuje powrót do punktu 1.



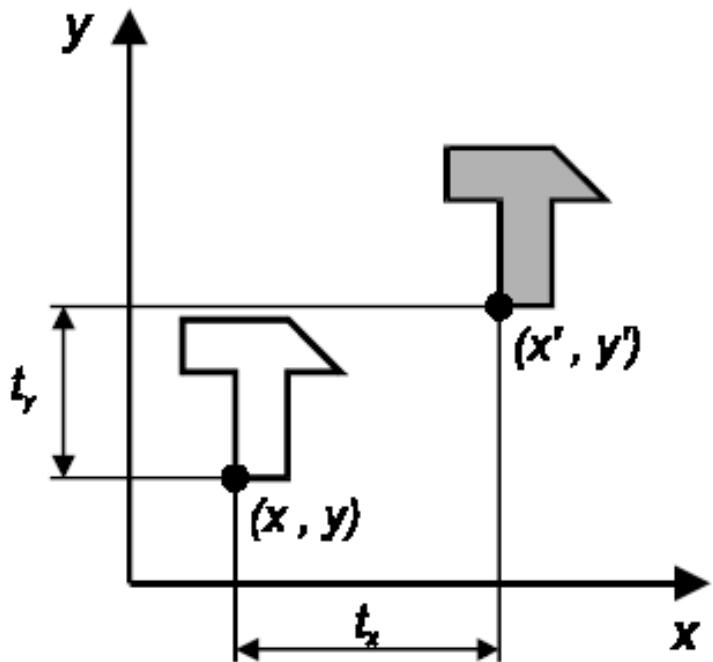
Transformacje 2-D

Transformacje w przestrzeni 2-D

Transformacje elementarne

- **przesunięcie,**
- **zmiana skali,**
- **obrót wokół środka układu współrzędnych.**

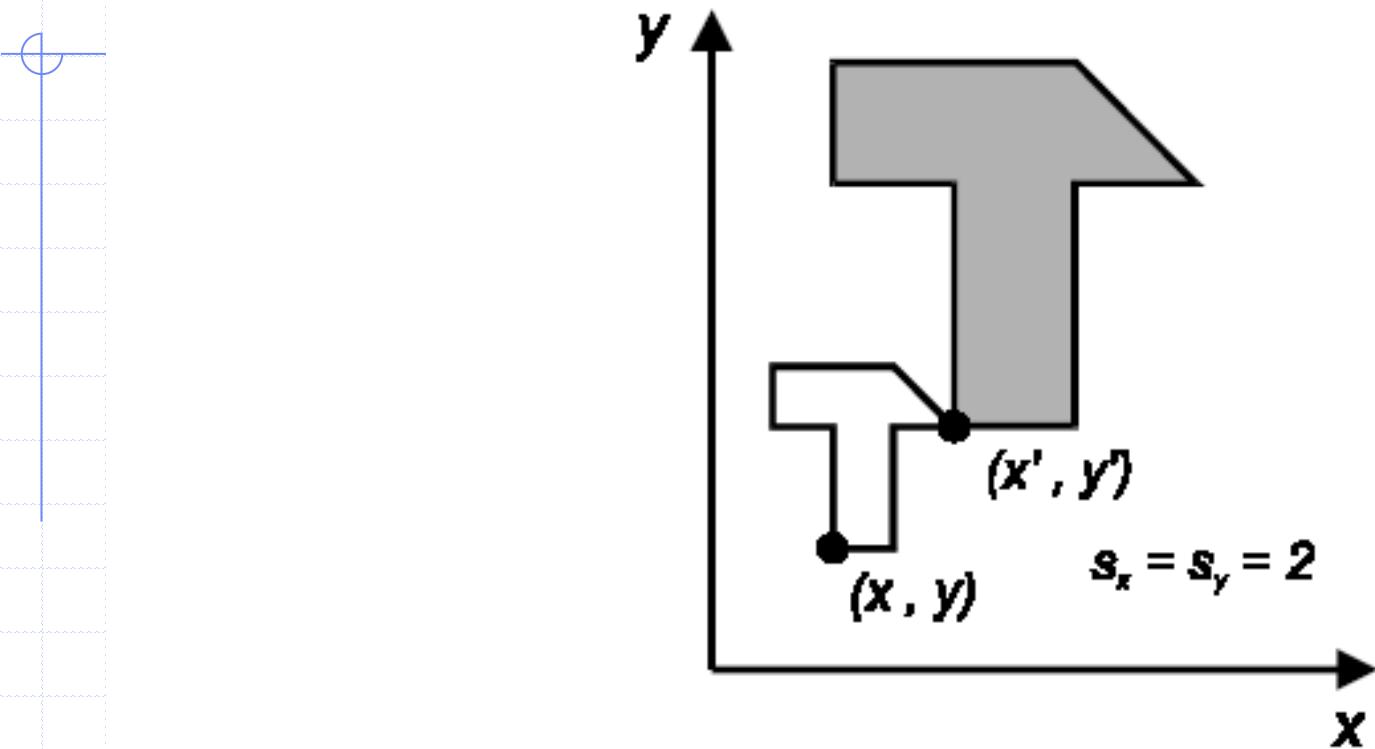
Przesunięcie (translation):



$$x' = x + t_x$$

$$y' = y + t_y$$

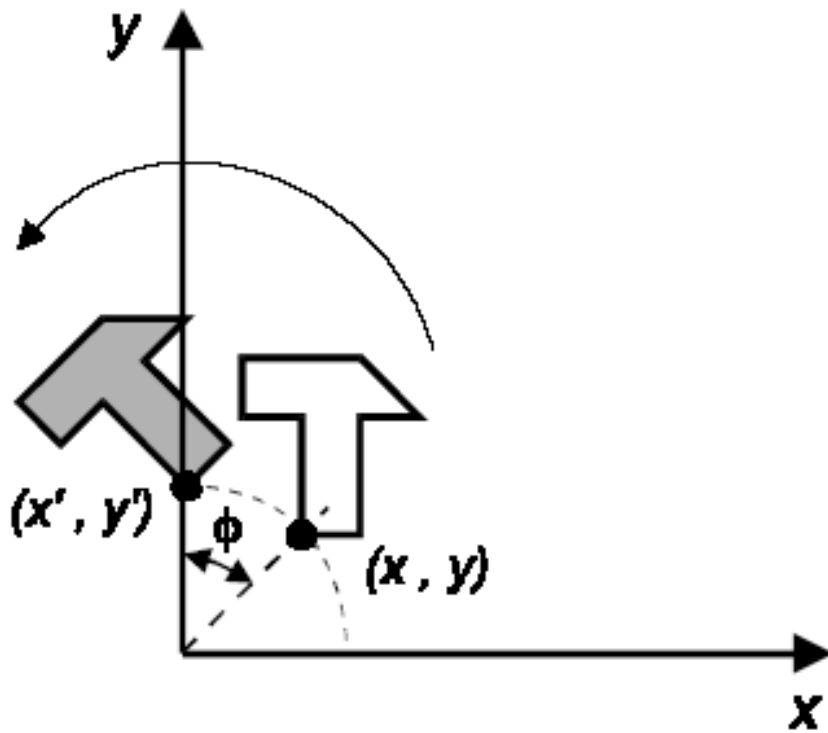
Zmiana skali (scaling):



$$x' = x \cdot s_x$$

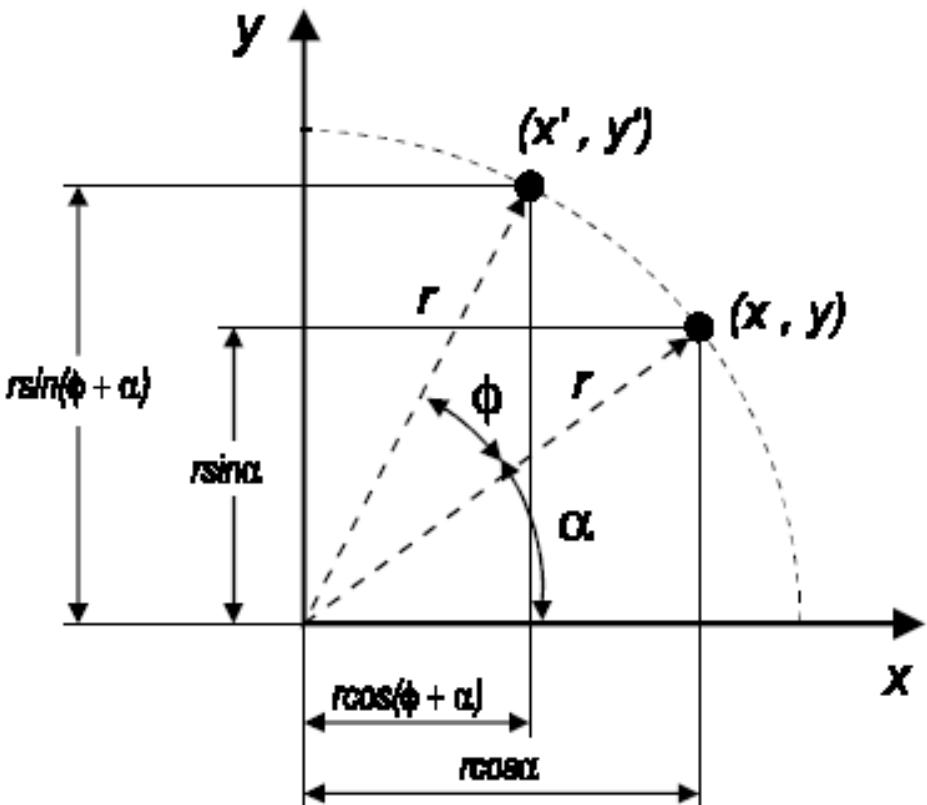
$$y' = y \cdot s_y$$

Obrót wokół środka układu współrzędnych (rotation):



$$x' = x \cos \phi - y \sin \phi$$

$$y' = y \cos \phi + x \sin \phi$$



$$x' = r \cos(\alpha + \phi) = r \cos \alpha \cos \phi - r \sin \alpha \sin \phi$$

$$y' = r \sin(\alpha + \phi) = r \sin \alpha \cos \phi - r \cos \alpha \sin \phi$$

$$x = r \cos \alpha. \quad v = r \sin \alpha$$

$$x' = x \cos \phi - y \sin \phi$$

$$y' = y \cos \phi + x \sin \phi$$

Współrzędne jednorodne (homogeneous coordinates)

August Ferdynand Möbius (1790 - 1868)

$$(x, y) \rightarrow [x \ y \ 1]$$

$$(x', y') \rightarrow [x' \ y' \ 1]$$

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix}$$

Przesunięcie:


$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

macierz przesunięcia

$$T(t_x, t_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

$$[x' \ y' \ 1] = [x \ y \ 1] \cdot T(t_x, t_y)$$

Zmiana skali:

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

macierz zmiany skali

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[x' \ y' \ 1] = [x \ y \ 1] \cdot S(s_x, s_y)$$

Obrót wokół środka układu współrzędnych:



$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

macierz obrotu

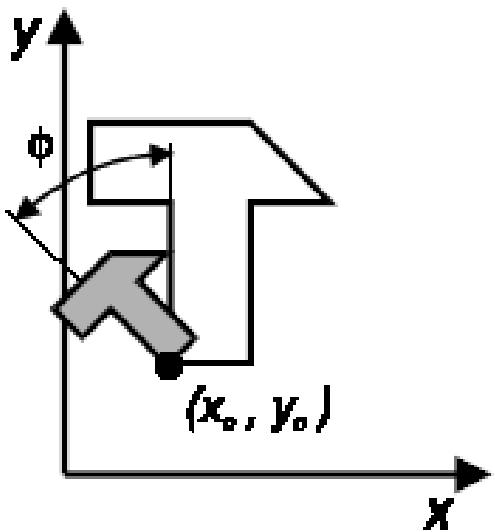
$$R(\phi) = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[x' \ y' \ 1] = [x \ y \ 1] \cdot R(\phi)$$

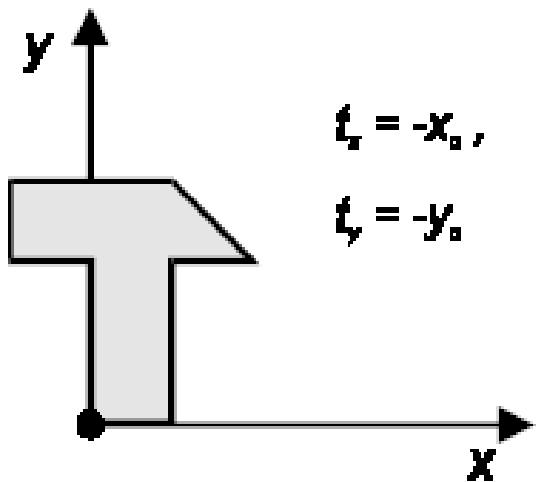
Składanie transformacji

Przykład:

Obrócić obiekt wokół punktu (x_c, y_c) pomniejszając obiekt dwukrotnie.



1. Przesunięcie obiektu, tak aby punkt (x_c, y_c) znalazła się w punkcie $(0, 0)$.



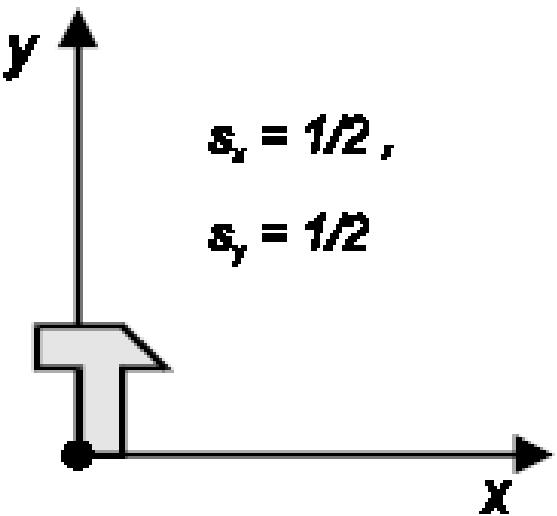
$$p = [x \quad y \quad 1]$$

$$p' = [x' \quad y' \quad 1]$$

$$p' = p \cdot T(-x_c, -y_c)$$

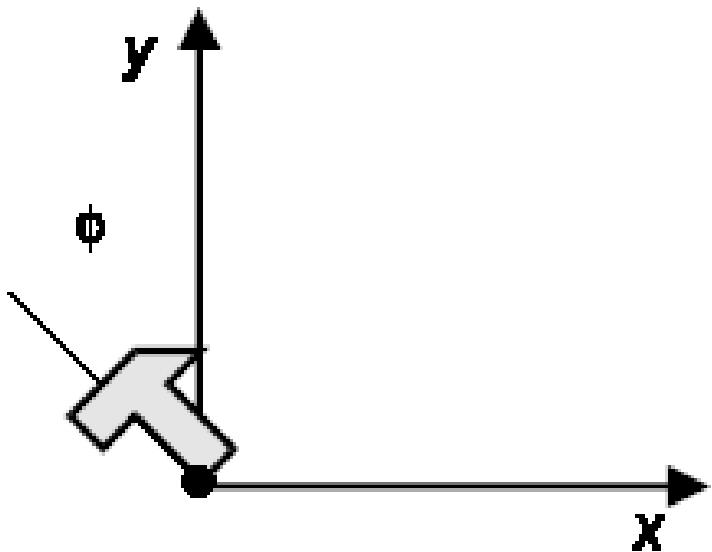
2. Przeskalowanie obiektu z parametrami skalowania

$$s_x = 1/2, \ s_y = 1/2.$$



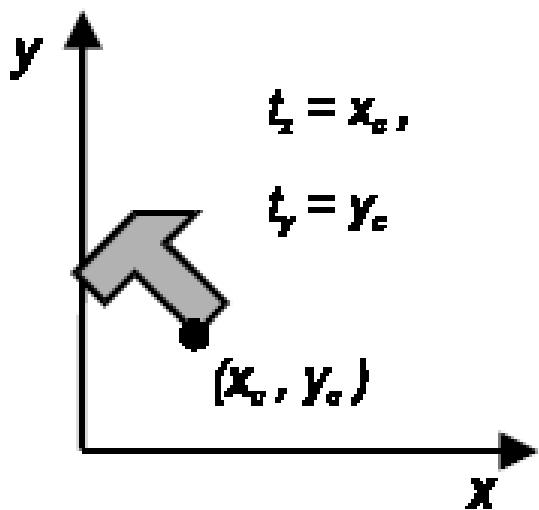
$$p' = p \cdot T(-x_c, -y_c) \cdot S(s_x, s_y)$$

3. Obrócenie obiektu wokół środka układu współrzędnych o kąt ϕ



$$\mathbf{p}' = \mathbf{p} \cdot T(-x_c, -y_c) \cdot S(s_x, s_y) \cdot R(\phi)$$

4. Przesunięcie obiektu, tak aby punkt $(0, 0)$ znalazł się w punkcie (x_c, y_c) .



$$p' = p \cdot T(-x_c, -y_c) \cdot S(s_x, s_y) \cdot R(\phi)T(x_c, y_c)$$

Transformacja ogólna:

$$M = T(-x_c, -y_c) \cdot S(s_x, s_y) \cdot R(\phi)T(x_d, y_d)$$

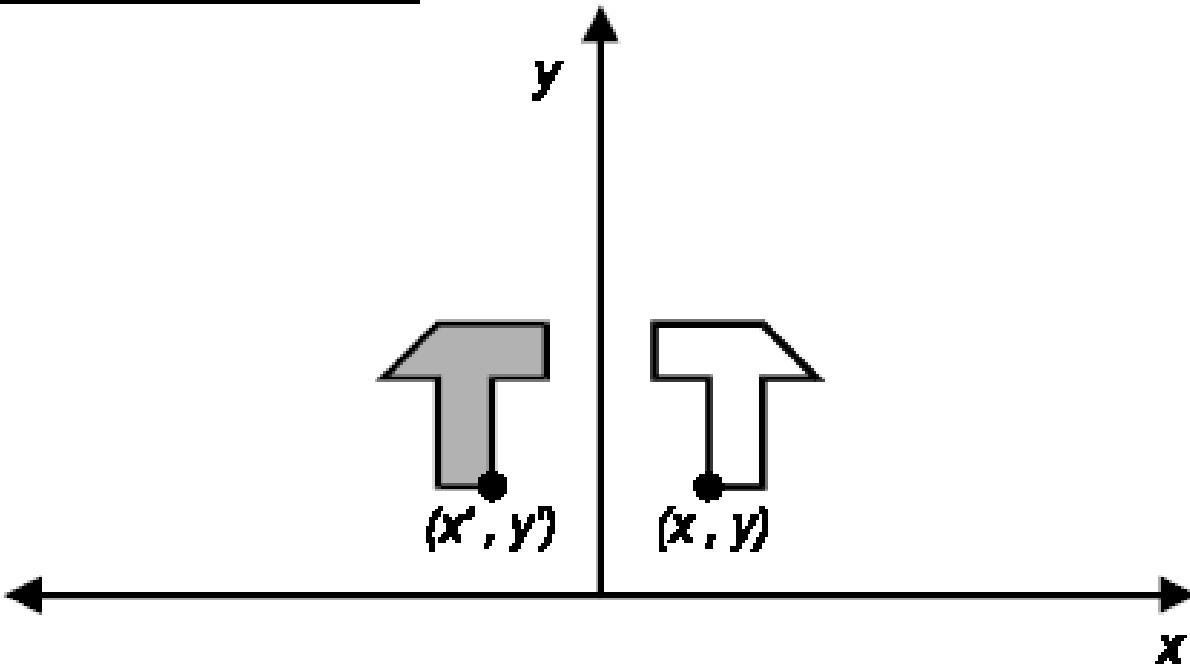
$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix}$$

obliczanie nowych współrzędnych punktu:

- 4 dodawania zmiennoprzecinkowe,
- 4 mnożenia zmiennoprzecinkowe.

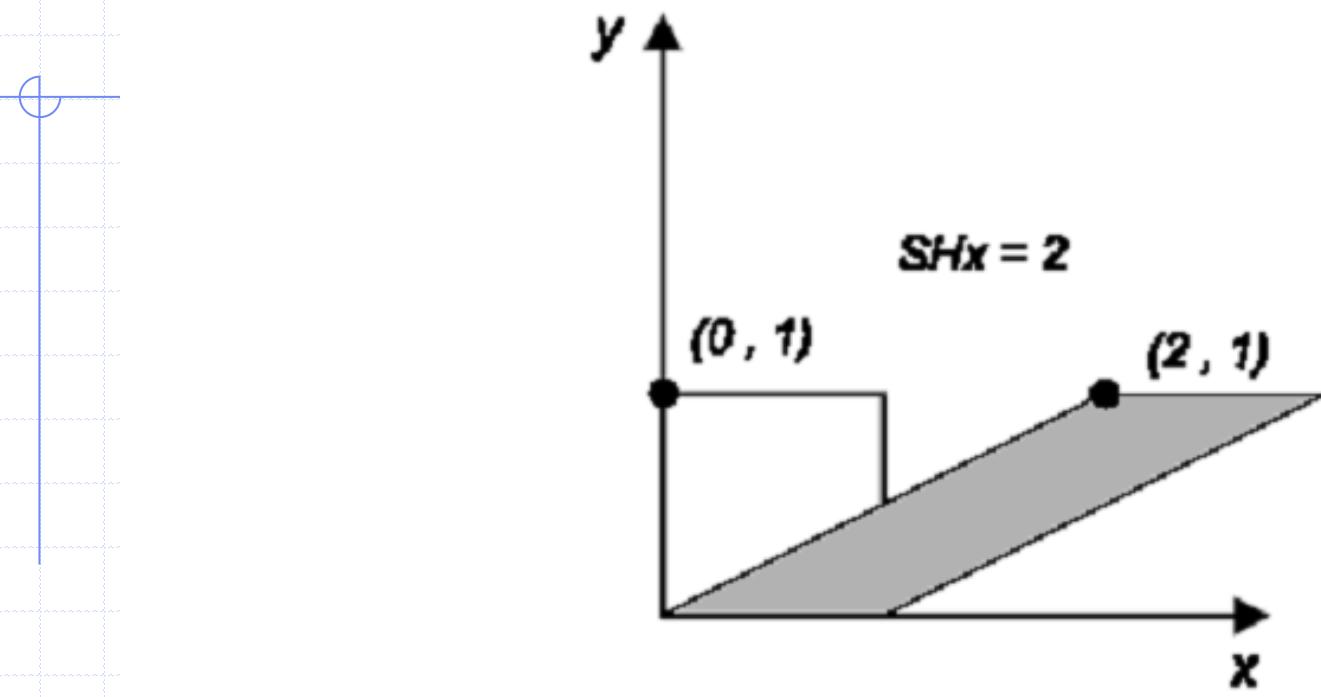
Inne transformacje

Odbicie (reflection):



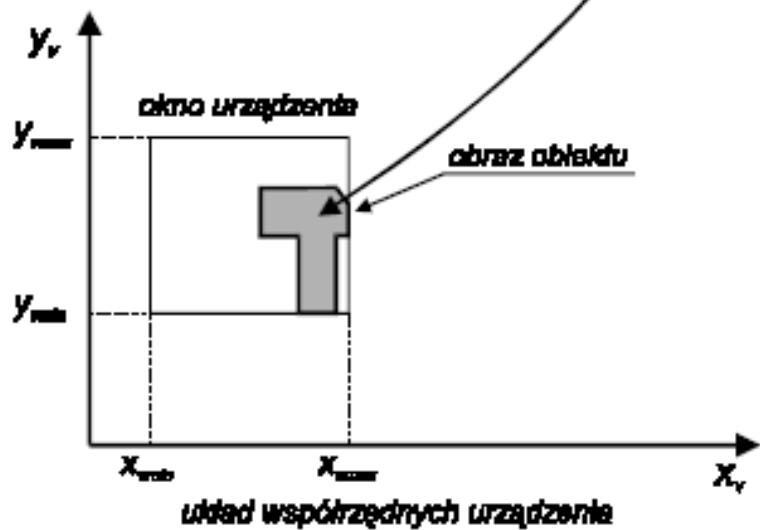
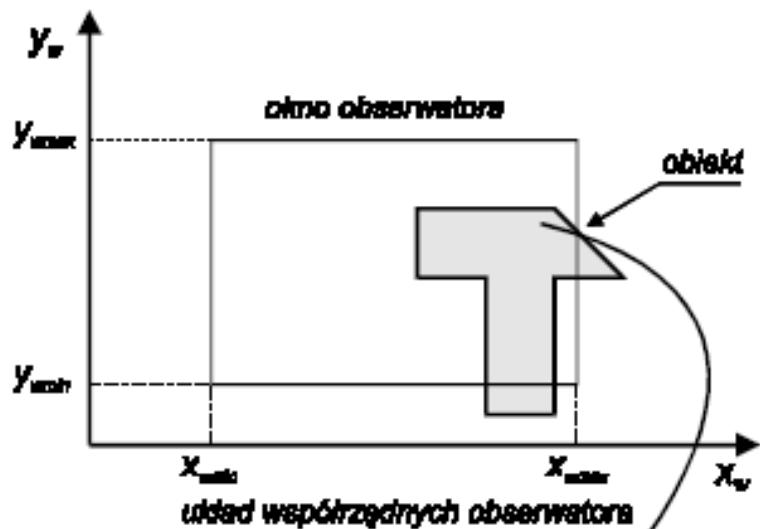
$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ścinanie (shear):



$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ SHx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ogólna procedura wizualizacji 2-D



- układ współrzędnych obserwatora (*world coordinates*),
- okno obserwatora (*window*),
- układ współrzędnych urządzenia (*device coordinates*),
- okno urządzenia (*viewport*),

Przebieg wizualizacji 2-D

1. Zdefiniować obiekt w układzie współrzędnych obserwatora .
2. W układzie współrzędnych obserwatora określić okno obserwatora.
3. W układzie współrzędnych urządzenia określić okno urządzenia.
4. Zmodyfikować opis obiektu usuwając te elementy , które znajdują się poza oknem obserwatora (obcinanie).
5. Przetransformować opis obiektu z wnętrza okna obserwatora do wnętrza okna urządzenia, stosując transformację

$$p_v = p_w \cdot T(-x_{w\min}, -y_{w\min}) S(s_x, s_y) T(x_{v\min}, y_{v\min})$$

gdzie

$$p_v = [x_v \quad y_v \quad 1] \quad p_w = [x_w \quad y_w \quad 1]$$

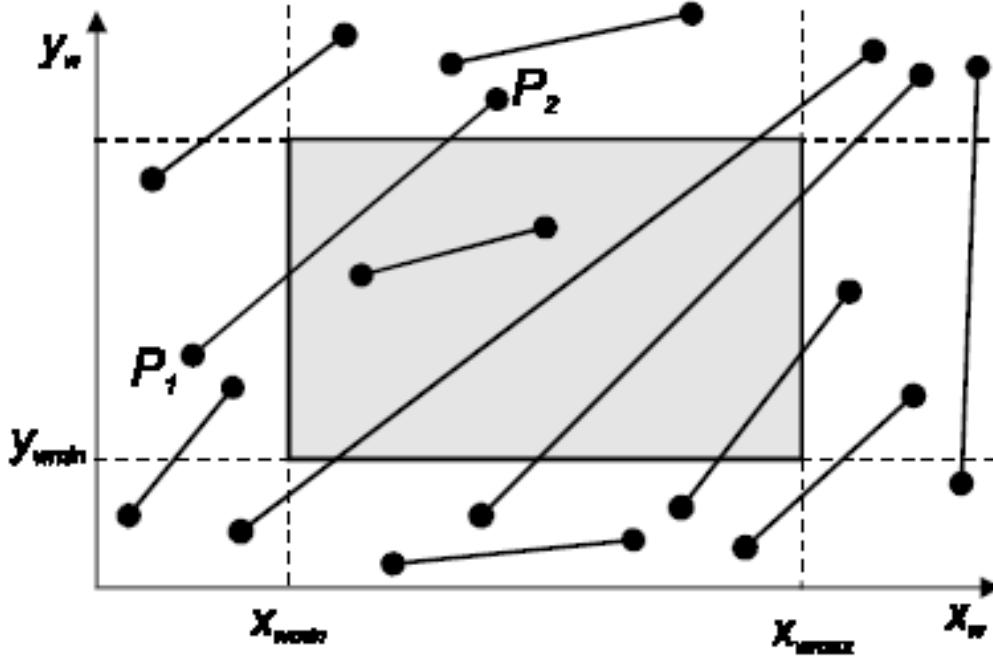
$$s_x = \frac{x_{v\max} - x_{v\min}}{x_{w\max} - x_{w\min}} \quad s_y = \frac{y_{v\max} - y_{v\min}}{y_{w\max} - y_{w\min}}$$

i narysować obraz obiektu.

Obcinanie (clipping)

Obcinanie odcinka - algorytm Cohena i Sutherlanda

W przestrzeni obserwatora dany jest zbiór odcinków. Każdy odcinek opisany jest przez punkt początkowy i końcowy.



Kodowanie obszarów w przestrzeni obserwatora:

1001	1000	1010
<i>okno obserwatora</i>		
0001	0000	0010
0101	0100	0110

bit4, bit3, bit2, bit1

bit1 = 1 - na lewo od okna obserwatora,
bit2 = 1 - na prawo od okna obserwatora,
bit3 = 1 - w dół od okna obserwatora,
bit4 = 1 - w górę od okna obserwatora.

Krok 1

Dla każdego punktu końcowego odcinka obliczyć różnice współrzędnych punktu końcowego i granic okna obserwatora

$$\alpha_1 = x - x_{w\ min}$$

$$\alpha_2 = x_{w\ max} - x$$

$$\alpha_3 = y - y_{w\ min}$$

$$\alpha_4 = y_{w\ max} - y$$

Krok 2

Zakodować wszystkie punkty końcowe odcinków według reguły:

jeżeli $\alpha_i < 0$ to $biti = 1$
jeżeli $\alpha_i \geq 0$ to $biti = 0$

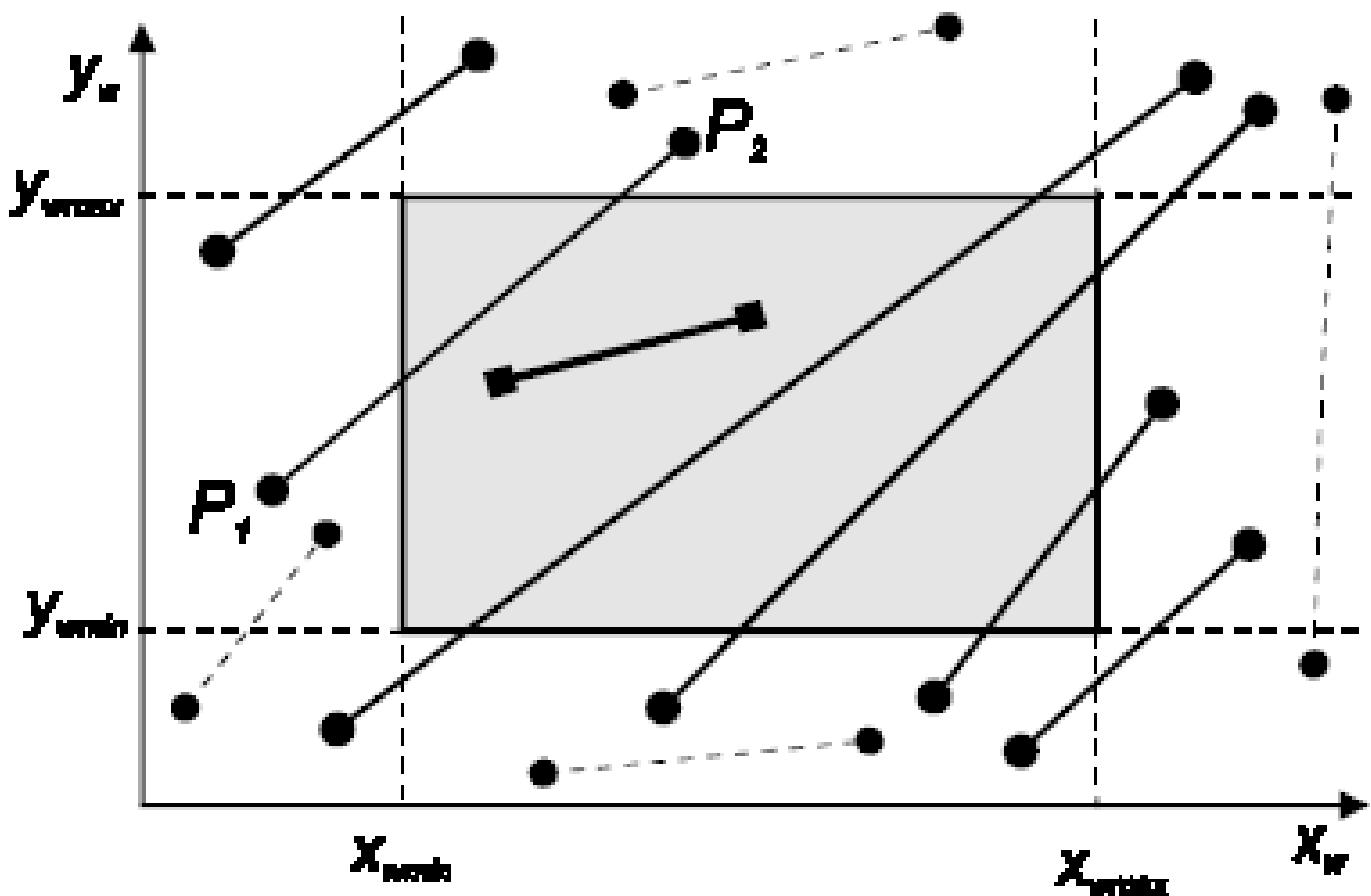
Krok 3

Sprawdzić kody par punktów końcowych dla wszystkich odcinków.

Jeżeli:

1. $\text{kod}P_1 = \text{kod}P_2 = 0000$ - odcinek leży całkowicie wewnętrz okna obserwatora
2. $\text{bit}_i \text{ dla } P_1 = \text{bit}_i \text{ dla } P_2 = 1$ - odcinek leży całkowicie na zewnątrz okna obserwatora

- pozostawić odcinki leżące wewnętrz okna,
- usunąć odcinki leżące na zewnątrz okna,
- jeśli wyczerpano w ten sposób wszystkie odcinki zakończyć algorytm,
- w przeciwnym przypadku wykonać krok 4.



Krok 4

Dla pozostałych odcinków, których punkty końcowe leżą na lewo, lub na prawo od granicy okna obliczyć nowe współrzędne tych punktów według wzorów:

$y = y_{w \min}$ dla punktów leżących na lewo,

$y = y_{w \max}$ dla punktów leżących na prawo,

$$x = x_k + (y - y_k) / m$$

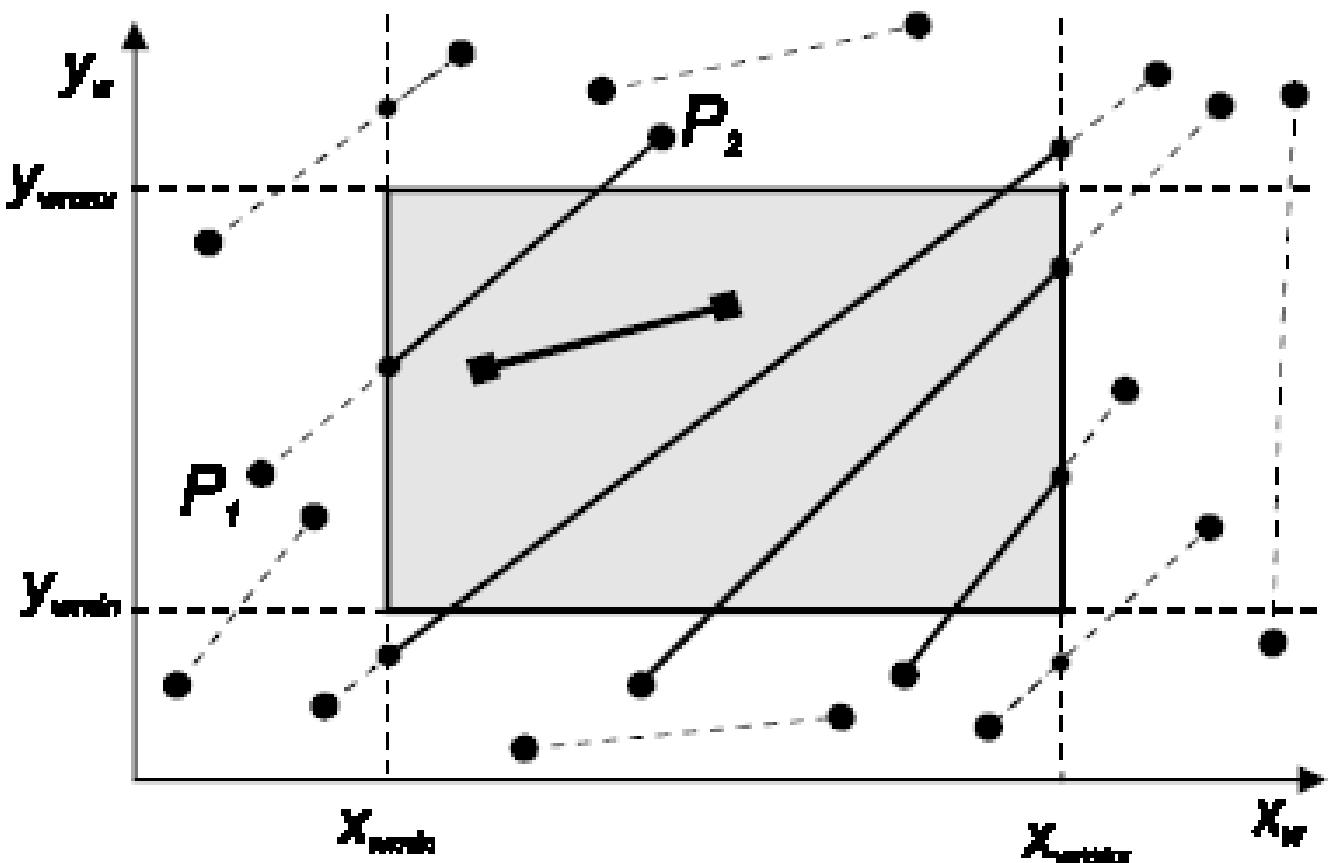
gdzie

x, y - nowe współrzędne punktu końcowego,

x_k, y_k - poprzednie współrzędne punktu.

Zakodować nowe punkty końcowe według reguły opisanej w kroku 2.

Powtórzyć krok 3.



Dla pozostałych odcinków, których punkty końcowe leżą pod, lub ponad granicami okna obliczyć nowe współrzędne tych punktów według wzorów:

$y = y_{w \min}$ dla punktów leżących poniżej okna,

$y = y_{w \max}$ dla punktów leżących powyżej okna,

$$x = x_k + (y - y_k) / m$$

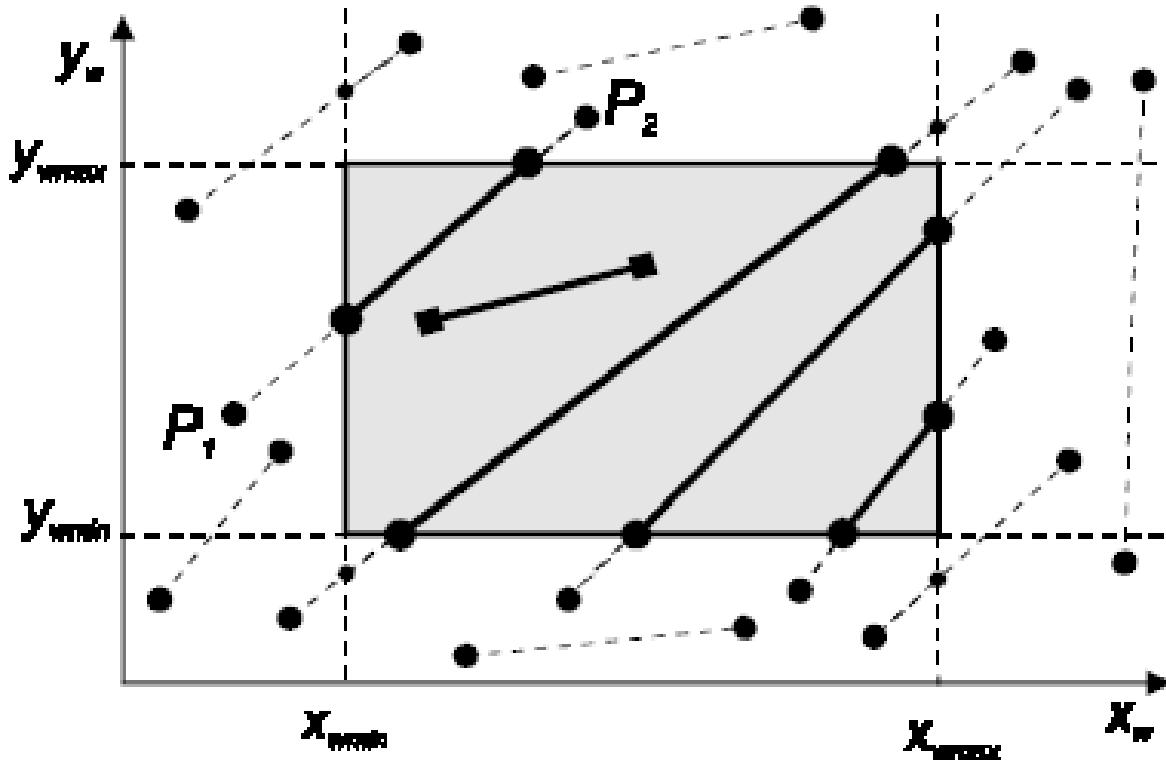
gdzie

x, y - nowe współrzędne punktu końcowego,
 x_k, y_k - poprzednie współrzędne punktu.

Zakodować nowe punkty końcowe według reguły opisanej w kroku 2.

Powtórzyć krok 3.

Ostateczny efekt obcinania:



GRAFIKA 3-D

Modele obiektów 3-D

Klasyfikacja modeli

- Modele opisujące powierzchnię
 - *siatka wieloboków*
 - *bezpośredni opis równaniem powierzchni*
 - *wielomianowe krzywe parametryczne*
 - *parametryczne wielomianowe płyty powierzchni*
 - *modele fraktalne*
- Modele opisujące bryłę
 - *operacje boolowskie*
 - *reprezentacje z przesuwem*
 - *reprezentacje brzegowe*
 - *reprezentacja z podziałem przestrzennym*
 - *konstrukcyjna geometria bryłowa*

Siatka wieloboków (*polygon surfaces*)

Siatka wieloboków - zbiór wierzchołków, krawędzi i wieloboków połączonych tak, że każda krawędź jest wspólna przynajmniej dla dwóch wieloboków.

Problemy:

1. Jak zbudować siatkę ?
2. Jak efektywnie zapisać siatkę ?

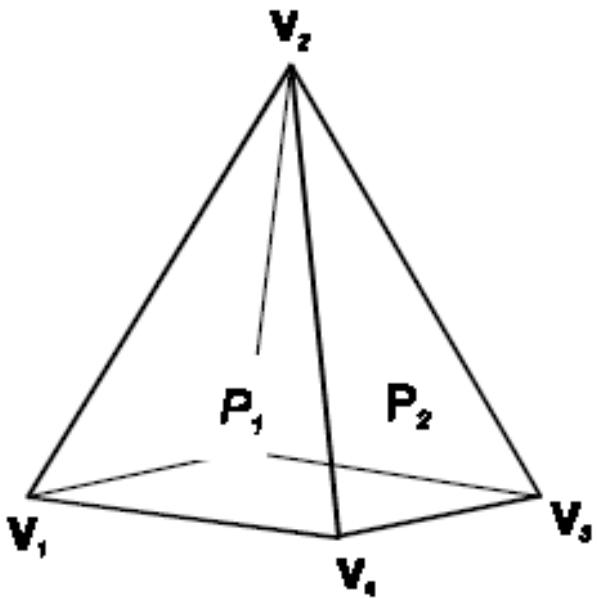
Metody tworzenia siatek wieloboków

- budowa siatki przez przetworzenie zadanego matematycznego opisu modelowanej powierzchni
- interakcyjne sposoby tworzenia siatek wieloboków wykorzystujące realny model trójwymiarowy bryły, urządzenia do wskazywania punktu w przestrzeni i odpowiednie algorytmy łączące wskazane punkty krawędziami

Metody reprezentacji siatek wieloboków

- **reprezentacja bezpośrednia**
- **reprezentacja przy pomocy wskaźników na listę wierzchołków**
- **reprezentacja przy pomocy wskaźników na listę krawędzi**

Reprezentacja bezpośrednią



Każdy wielobok opisany jest listą współrzędnych wierzchołków.

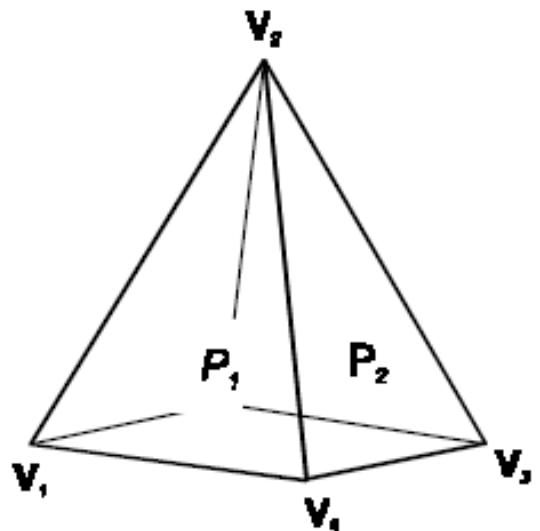
$$\mathbf{V} = ((x_1, y_1, z_1), \dots, (x_4, y_4, z_4)) = (V_1, V_2, V_3, V_4)$$

$$P_1 = (E_1, E_4, E_5) = (1, 4, 5)$$

Własności opisu:

- wierzchołki na listach wieloboków powtarzają się
- modyfikacja opisu jest trudna (przesunięcie wierzchołka wymaga znalezienia wszystkich wieloboków wspólnych dla wierzchołka i modyfikacji odpowiedniej trójki współrzędnych)
- przy rysowaniu krawędzie wspólne dla dwóch wieloboków będą rysowane dwukrotnie

Reprezentacja przy pomocy wskaźników na listę wierzchołków



Każdy wierzchołek zapisany jest na liście wierzchołków.

$$V = ((x_1, y_1, z_1), \dots, (x_4, y_4, z_4)) = (V_1, V_2, V_3, V_4)$$

Wieloboki zapisywane są jako listy wskaźników na elementy listy wierzchołków.

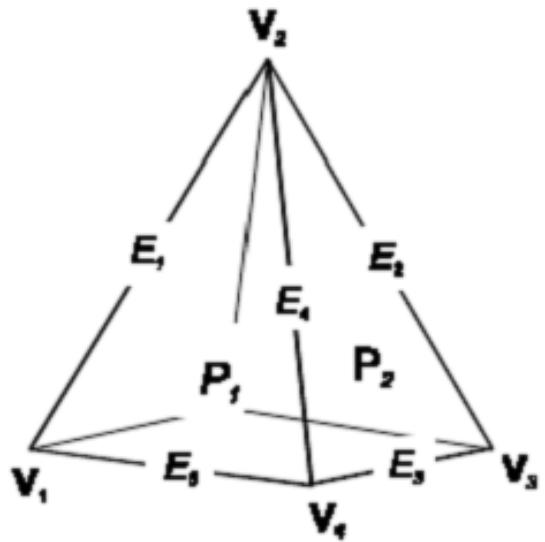
$$P_1 = (E_1, E_4, E_5) = (1, 4, 5) ; P_2 = (2, 3, 4)$$

Własności opisu:

- **wierzchołek jest zapisywany tylko jeden raz**
- **modyfikacja współrzędnych wierzchołka jest łatwa**
- **trudno znaleźć wieloboki o wspólnej krawędzi**
- **przy rysowaniu krawędzie wspólne dla dwóch wieloboków będą rysowane dwukrotnie**
- **trudno wypełniać obrazy wieloboków**



Reprezentacja przy pomocy wskaźników na listę krawędzi



Każdy wierzchołek zapisany jest na liście wierzchołków.

$$\mathbf{V} = ((x_1, y_1, z_1), \dots, (x_4, y_4, z_4)) = (V_1, V_2, V_3, V_4)$$

Tworzona jest lista krawędzi.

$$E_1 = (V_1, V_2, P_1, \lambda) = (1, 2, 1, 0)$$

$$E_2 = (V_2, V_3, P_2, \lambda) = (2, 3, 2, 0)$$

$$E_3 = (V_3, V_4, P_2, \lambda) = (3, 4, 2, 0)$$

$$E_4 = (V_4, V_2, P_1, P_2) = (4, 2, 1, 2)$$

$$E_5 = (V_4, V_1, P_1, \lambda) = (4, 1, 1, 0)$$

Wieloboki zapisywane są jako listy wskaźników na elementy listy krawędzi.

$$P_1 = (E_1, E_4, E_5) = (1, 4, 5)$$

$$P_2 = (E_2, E_3, E_4) = (2, 3, 4)$$

Własności opisu:

- **wierzchołek jest zapisywany tylko jeden raz**
- **modyfikacja współrzędnych wierzchołka jest łatwa**
- **łatwo znaleźć wieloboki o wspólnej krawędzi**
- **przy rysowaniu krawędzie wspólne dla dwóch wieloboków będą jeden raz**
- **łatwo wypełniać obrazy wieloboków**
- **brak bezpośredniej informacji, która krawędź łączy się z wierzchołkiem**

Elementy rachunku wektorowego:

wektor - $P_i = [x_i \quad y_i \quad z_i]$

Suma wektorów:

$$P_1 + P_2 = [x_1 + x_2 \quad y_1 + y_2 \quad z_1 + z_2]$$

Iloczyn wektora i liczby:

$$aP_i = [ax_i \quad ay_i \quad az_i]$$

Iloczyn skalarny:

$$P_1 \cdot P_2 = x_1x_2 + y_1y_2 + z_1z_2$$

Iloczyn wektorowy:

wersory (wektory jednostkowe)



$$u_x = [1 \ 0 \ 0]$$

$$u_y = [0 \ 1 \ 0]$$

$$u_z = [0 \ 0 \ 1]$$

iloczyn wektorowy

$$\begin{aligned} P_1 \times P_2 &= \det \begin{bmatrix} u_x & u_y & u_z \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{bmatrix} = \\ &= [y_1 z_2 - z_1 y_2, \ z_1 x_2 - x_1 z_2, \ x_1 y_2 - y_1 x_2] \end{aligned}$$

Równanie płaszczyzny wyznaczanej przez wierzchołki wieloboku , wektor normalny dla tej płaszczyzny:

Równanie płaszczyzny:

$$Ax + By + Cz + D = 0$$

1. Obliczenie wektora normalnego $[A \ B \ C]$

Dla trzech wierzchołków wieloboku P_1, P_2, P_3 obliczyć:

$$[A \ B \ C] = (P_2 - P_1) \times (P_3 - P_1)$$

Jeżeli $[A \ B \ C] = [0 \ 0 \ 0]$ to wierzchołki są współliniowe i nie określają płaszczyzny. Należy w takim przypadku wybrać inne wierzchołki i ponownie obliczyć wektor normalny.

2. Obliczenie współczynnika D

Obliczone wcześniej współczynniki A, B, C i współrzędne dowolnego wierzchołka wstawić do równania płaszczyzny i wyliczyć D .

← Dla różnych trójk wierzchołków możemy otrzymać różne równania płaszczyzny w przypadku gdy wielobok nie jest płaski.

Aby otrzymać wielobok lub układ wieloboków płaskich można:

1. Podzielić wielobok na mniejsze wieloboki płaskie.
2. Zmodyfikować współrzędne wierzchołków tak, aby nowe wierzchołki leżały na płaszczyźnie możliwie najbliższej (w sensie pewnego kryterium) poprzednim wierzchołkom.

Bezpośredni opis równaniem powierzchni

Postać uwikłana równania powierzchni

$$f(x, y, z) = 0$$

5.3.1. Powierzchnie drugiego stopnia (kwadryki)

Kwadryka opisana jest równaniem

$$\begin{aligned} f(x, y, z) = & Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz + \\ & + 2Fxz + 2Gx + 2Hy + 2Jz + K = 0 \end{aligned}$$

lub

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & J \\ G & H & J & K \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$$



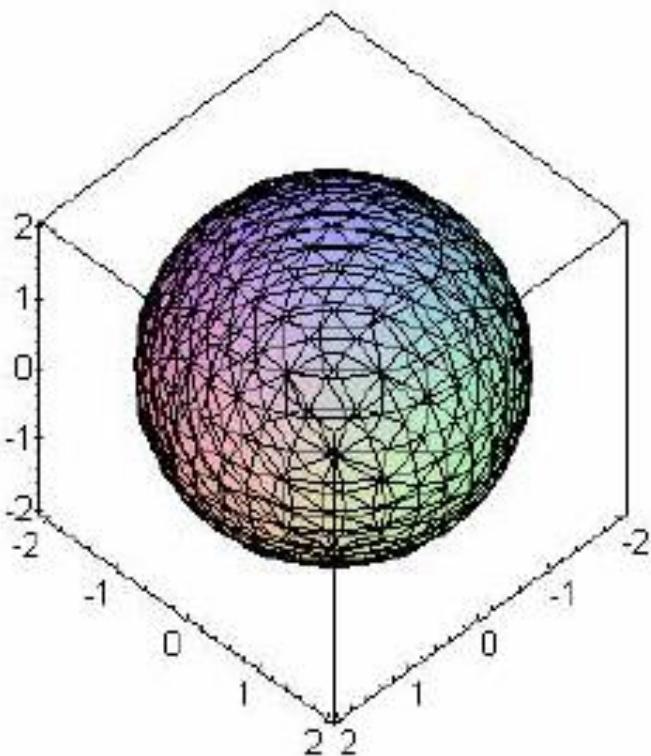
1. Płaszczyzna

$$A = B = C = \dots = F = 0$$

$$f(x, y, z) = 2Gx + 2Hy + 2Jz + K = 0$$

2. Sfera

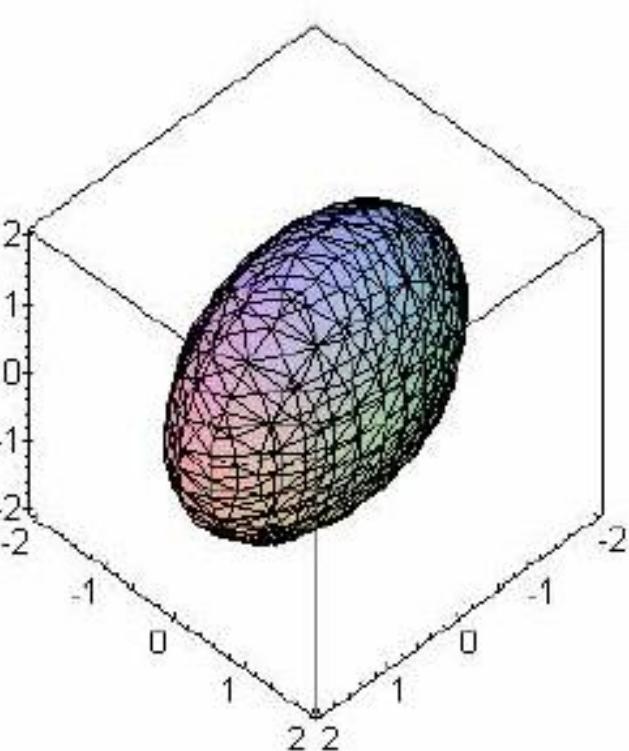
$$f(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0$$



$$r = 2$$

3. Elipsoida

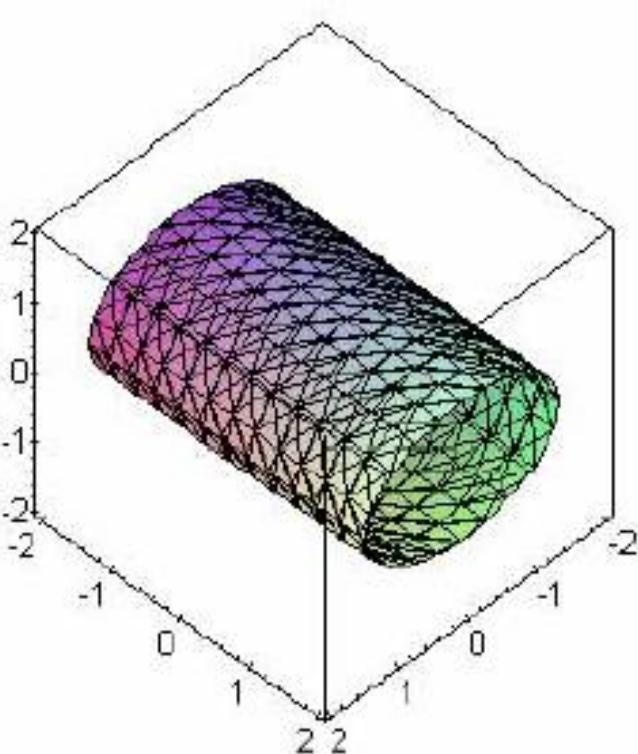
$$f(x, y, z) = \frac{x^2}{a^2} + \frac{z^2}{c^2} - 4fy = 0$$



$$a = 2, b = 1, c = 2$$

4. Cylinder

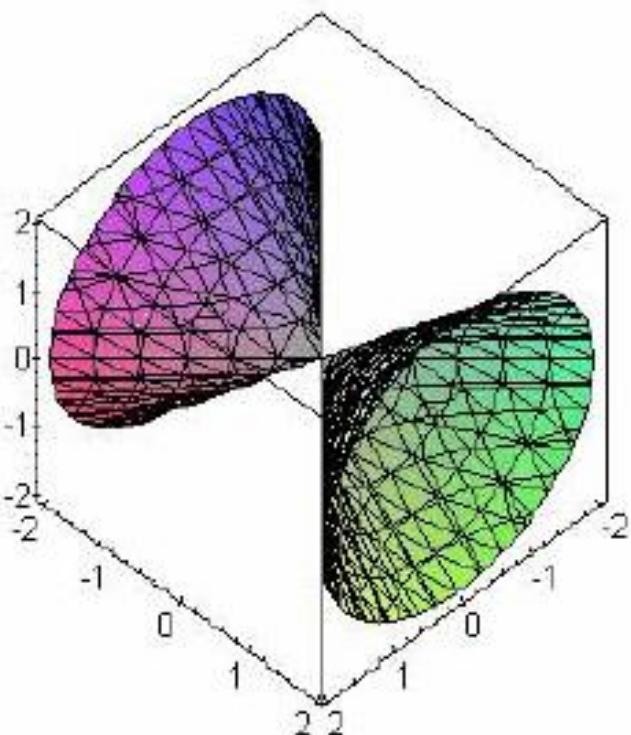
$$f(x, y, z) = \frac{x^2}{a^2} - \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$$



$$a = c = 1, b = 1 / \sqrt{2}$$

5. Stożek

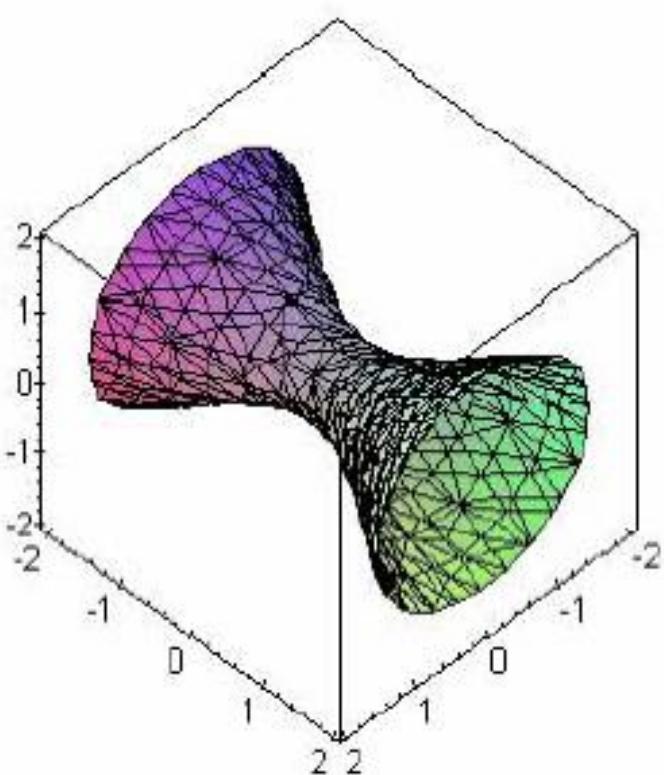
$$f(x, y, z) = \frac{x^2}{a^2} + \frac{z^2}{c^2} - 4fy = 0$$



$$a = 1, \quad c = 1$$

6. Hiperboloida

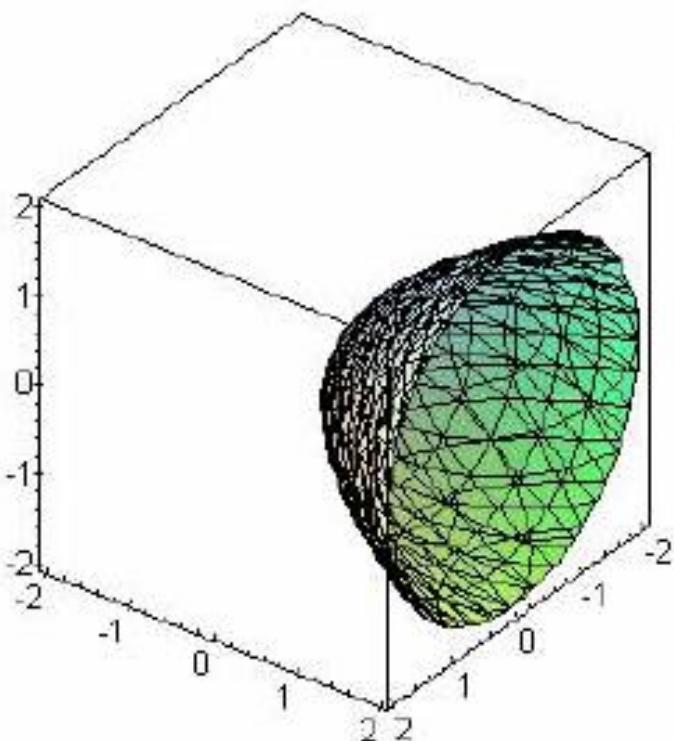
$$f(x,y,z) = \frac{x^2}{a^2} - \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$$



$$a = c = 1, b = 1/\sqrt{2}$$

7. Paraboloida

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{z^2}{c^2} - 4fy = 0$$



$$a = c = 1/\sqrt{2}, \quad f = 1$$

Zalety stosowania kwadryk w grafice komputerowej:

- Łatwe obliczanie wektora normalnego do powierzchni


$$\vec{N} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{bmatrix}$$

Wektor normalny można wyznaczyć analitycznie.

- Łatwe obliczanie punktów przecięcia powierzchni z prostą (ważne w algorytmach Ray - tracing).
- Łatwe testowanie czy dany punkt leży na powierzchni (podstawić dany punkt do równania i sprawdzić czy wynik jest bliski zera).
- Łatwe obliczanie z dla danych x, y (ważne w algorytmach realizujących usuwanie powierzchni niewidocznych)
- Stosunkowo łatwe obliczanie przecięcia jednej kwadryki z drugą.

Wada opisu uwikłanego powierzchni:

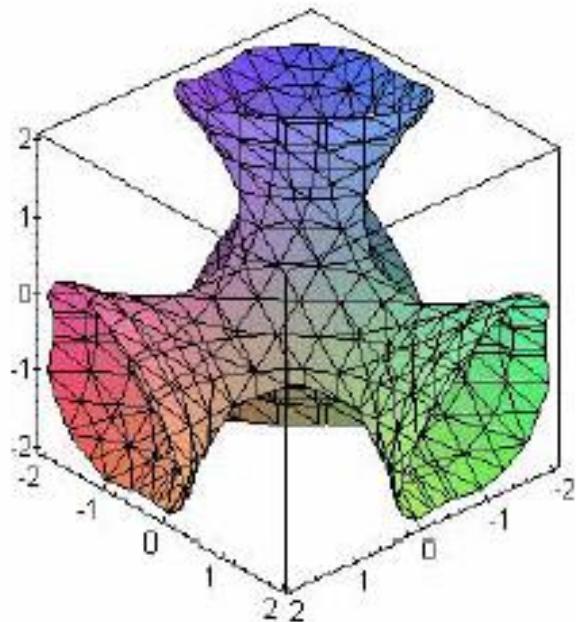
- Trudna generacja punktu leżącego na powierzchni
- Trudna generacja fragmentu powierzchni np. połowy czy ćwiartki sfery

Powierzchnia trzeciego stopnia

Przykład:



$$f(x, y, z) = x^3 + y^3 + z^3 - (x + y + z)^3 = 0$$



Parametryczny opis powierzchni

$$x(u,v) = f_x(u,v) \quad u_{\min} \leq u \leq u_{\max}$$

$$y(u,v) = f_y(u,v) \quad v_{\min} \leq v \leq v_{\max}$$

$$z(u,v) = f_z(u,v)$$



Przykłady:

1. Sfera o promieniu r i środku $(0, 0, 0)$

$$x(u,v) = r \sin(\pi u) \cos(2\pi v) \quad 0 \leq u \leq 1$$

$$y(u,v) = r \sin(\pi u) \sin(2\pi v) \quad 0 \leq v \leq 1$$

$$z(u,v) = r \cos(\pi u)$$

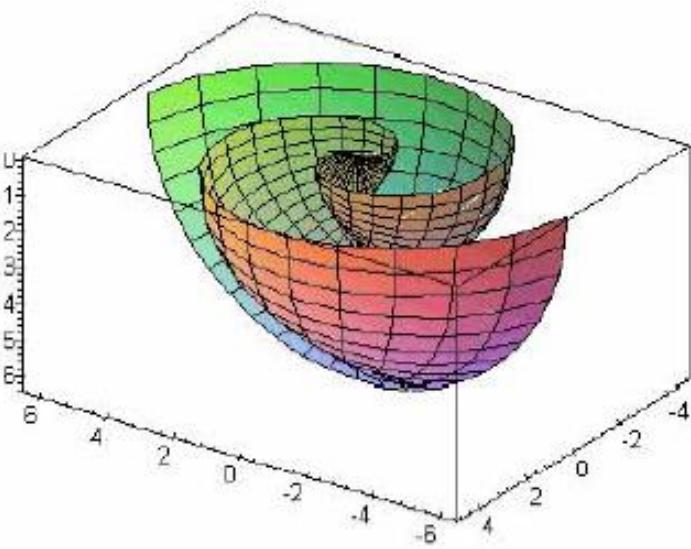
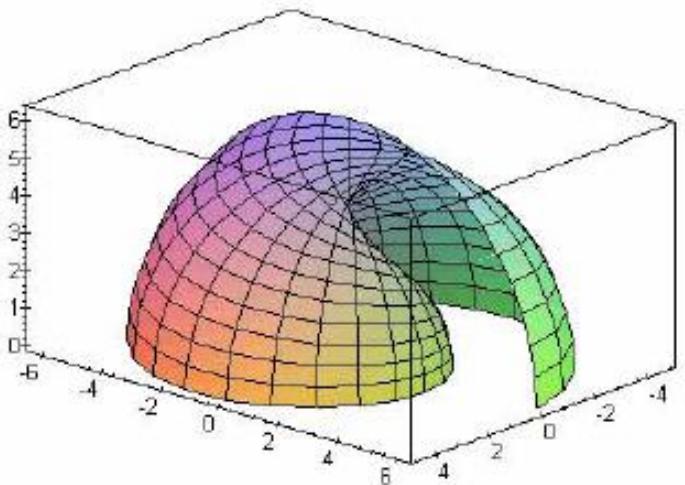
- Łatwa generacja punktu leżącego na powierzchni (dla pary parametrów u, v łatwo jest wyznaczyć $x(u,v), y(u,v), z(u,v)$).
- Łatwa generacja fragmentu powierzchni (np. prowadzimy obliczenia punktów dla przedziałów $0 \leq u \leq 1/2, 0 \leq v \leq 1/2$ i uzyskujemy tylko część powierzchni sfery).

2. Powierzchnia o równaniu

$$x(u, v) = u \sin(u) \cos(v) \quad 0 \leq u \leq 2\pi$$

$$y(u, v) = u \cos(u) \cos(v) \quad 0 \leq v \leq \pi$$

$$z(u, v) = u \sin(v)$$



Modele obiektów 3-D

Krzywa parametryczna w przestrzeni 3-D

Równanie parametryczne krzywej:

$$x(u) = f_x(u)$$

$$y(u) = f_y(u) \quad u_{min} \leq u \leq u_{max}$$

$$z(u) = f_z(u)$$

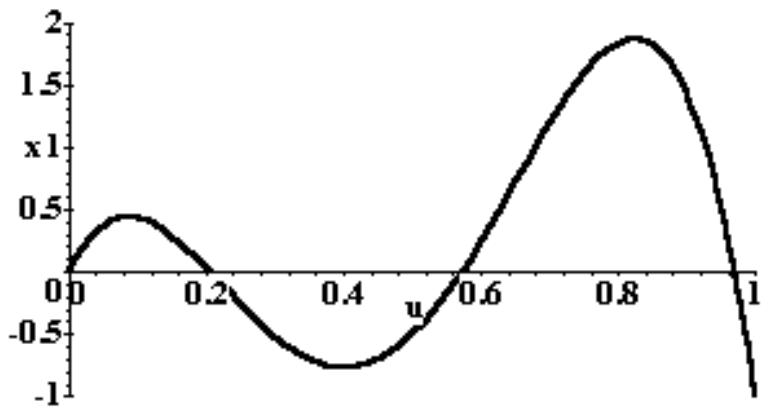
Krzywa opisana wielomianami 4-go stopnia.

$$x(u) = -100u^4 + 175u^3 - \frac{175}{2}u^2 + \frac{23}{2}u$$

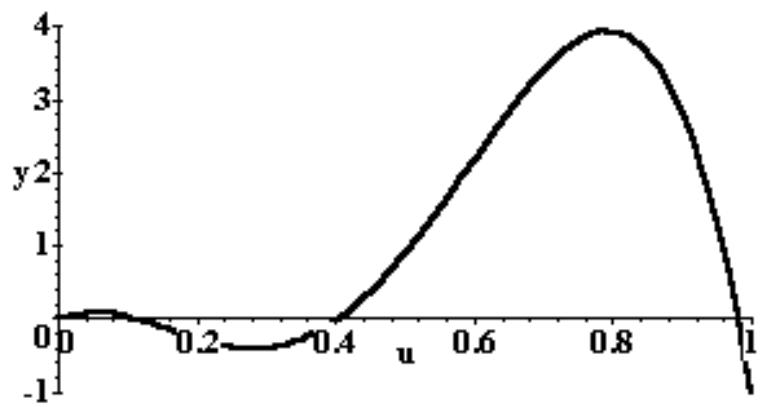
$$y(u) = -100u^4 + 150u^3 - \frac{500}{9}u^2 + \frac{41}{9}u \quad 0 \leq u \leq 1$$

$$z(u) = 0$$

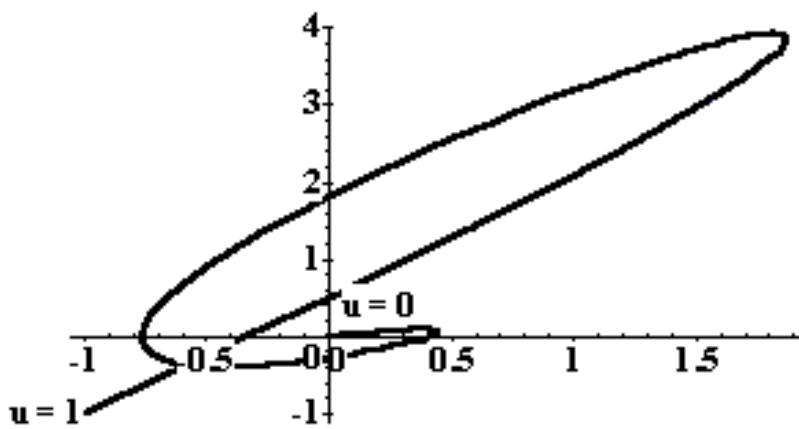
Jeśli $z(u) = 0$, to krzywa leży na płaszczyźnie $x-y$.



funkcja $x(u) = f_x(u)$



funkcja $y(u) = f_y(u)$



krzywa

$$\begin{aligned}x(u) &= f_x(u) \\y(u) &= f_y(u)\end{aligned}$$

$$0 \leq u \leq 1$$

Problemy :

1. Jaki wybrać stopnie wielomianów ?

- niski stopień - mała elastyczność**
- wysoki stopień - trudne do opanowania oscylacje funkcji i wzrost złożoności obliczeniowej**

2. Jak dobierać współczynniki wielomianów, aby w przewidywalny sposób wpływać na kształt modelowanej krzywej ?

Parametryczne krzywe trzeciego stopnia

Ogólna postać równań dla krzywej trzeciego stopnia

Krzywa zadana jest równaniem:

$$Q(u) = [x(u) \quad y(u) \quad z(u)]^T$$

gdzie

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$

$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \quad 0 \leq u \leq 1$$

$$z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$$

Jeżeli

$$U = [u^3 \quad u^2 \quad u \quad 1]^T$$

oraz

$$C = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

to

$$Q(u) = [x(u) \quad y(u) \quad z(u)]^T = C \cdot U$$

Krzywa Hermite'a

Charles Hermite (1822 - 1901)

Krzywa określona jest przez cztery ograniczenia:

- dwa punkty krzywej (początkowy i końcowy),
- dwa wektory styczne w tych punktach.

Jak współczynniki równania krzywej zależą od ograniczeń?

Można napisać, że

$$Q(u) = C \cdot U = G \cdot M \cdot U$$

lub inaczej

$$Q(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = [G_1 \ G_2 \ G_3 \ G_4] \begin{bmatrix} m_{11} & m_{21} & m_{31} & m_{41} \\ m_{12} & m_{22} & m_{32} & m_{42} \\ m_{13} & m_{23} & m_{33} & m_{43} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

M - macierz bazowa Hermite'a

G - macierz geometrii (opisuje wymienione wcześniej ograniczenia)

$$G = [G_1 \ G_2 \ G_3 \ G_4] = \begin{bmatrix} P_{0x} & P_{1x} & R_{0x} & R_{1x} \\ P_{0y} & P_{1y} & R_{0y} & R_{1y} \\ P_{0z} & P_{1z} & R_{0z} & R_{1z} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix}$$

gdzie

$(P_{0x}, P_{0y}, P_{0z}), (P_{1x}, P_{1y}, P_{1z})$ - współrzędne punktów początkowego i końcowego krzywej,

$(R_{0x}, R_{0y}, R_{0z}), (R_{1x}, R_{1y}, R_{1z})$ - współrzędne końców wektorów stycznych w punktach, początkowym i końcowym.

Celem rozważań jest wyznaczenie macierzy M .

Pamiętając, że

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x = G_x \cdot M \cdot [u^3 \ u^2 \ u \ 1]^T$$

oraz

$$x'(u) = G_x \cdot M \cdot [3u^2 \ 2u \ 1 \ 0]^T$$

uwzględniamy nałożone wcześniej ograniczenia

$$x(0) = P_{0x} = G_x \cdot M \cdot [0 \ 0 \ 0 \ 1]^T$$

$$x(1) = P_{1x} = G_x \cdot M \cdot [1 \ 1 \ 1 \ 1]^T$$

$$x'(0) = R_{0x} = G_x \cdot M \cdot [0 \ 0 \ 1 \ 0]^T$$

$$x'(1) = R_{1x} = G_x \cdot M \cdot [3 \ 2 \ 1 \ 2]^T$$

Sformułowane wyżej cztery równanie zapisane w postaci macierzowej przyjmuje formę

$$[P_{0x} \ P_{1x} \ R_{0x} \ R_{1x}] = G_x = G_x \cdot M \cdot \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Aby równanie było spełnione musi zachodzić

$$M = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

Równanie krzywej Hermite'a można więc zapisać jako

$$Q(u) = [x(u) \ y(u) \ z(u)]^T = G \cdot M \cdot U$$

lub w postaci rozwiniętej

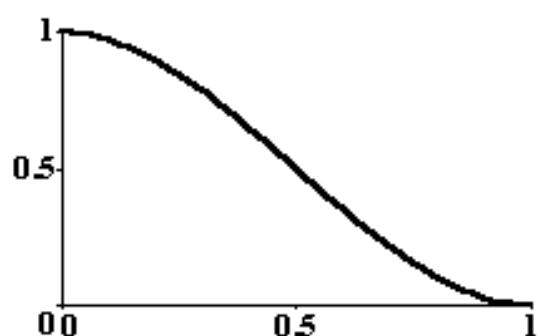
$$\begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \end{bmatrix} (2u^3 - 3u^2 + 1) + \begin{bmatrix} P_{1x} \\ P_{1y} \\ P_{1z} \end{bmatrix} (-2u^3 + 3u^2) + \\ + \begin{bmatrix} R_{0x} \\ R_{0y} \\ R_{0z} \end{bmatrix} (u^3 - 2u^2 + u) + \begin{bmatrix} R_{1x} \\ R_{1y} \\ R_{1z} \end{bmatrix} (u^3 - u^2)$$

Wielomiany w nawiasach noszą nazwę funkcji bazowych Hermite'a.

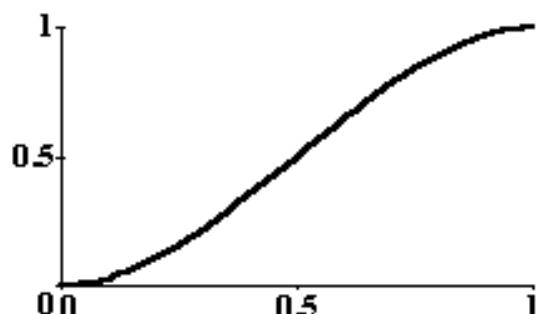
Krzywą można zapisać jako kombinację liniową wektorów składowych macierzy geometrii G i funkcji bazowych Hermite'a $B(u)$.

$$Q(u) = P_0B_0(u) + P_1B_1(u) + R_0B_2(u) + R_1B_3(u)$$

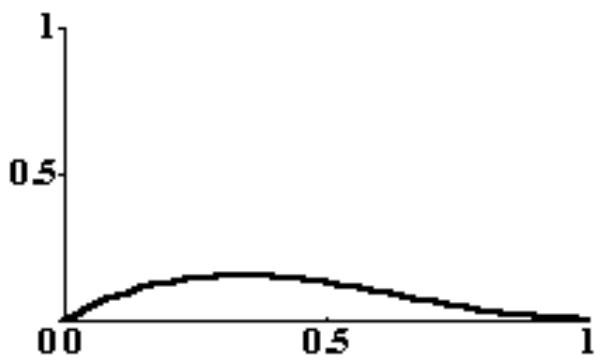
Poszczególne wielomiany bazowe mają następujący przebieg.



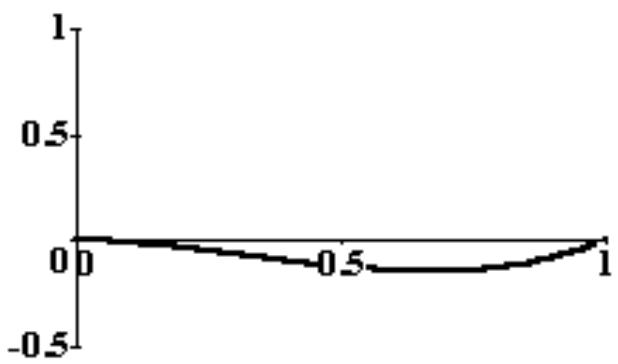
$$B_0(u) = 2u^3 - 3u^2 + 1$$



$$B_1(u) = -2u^3 + 3u^2$$



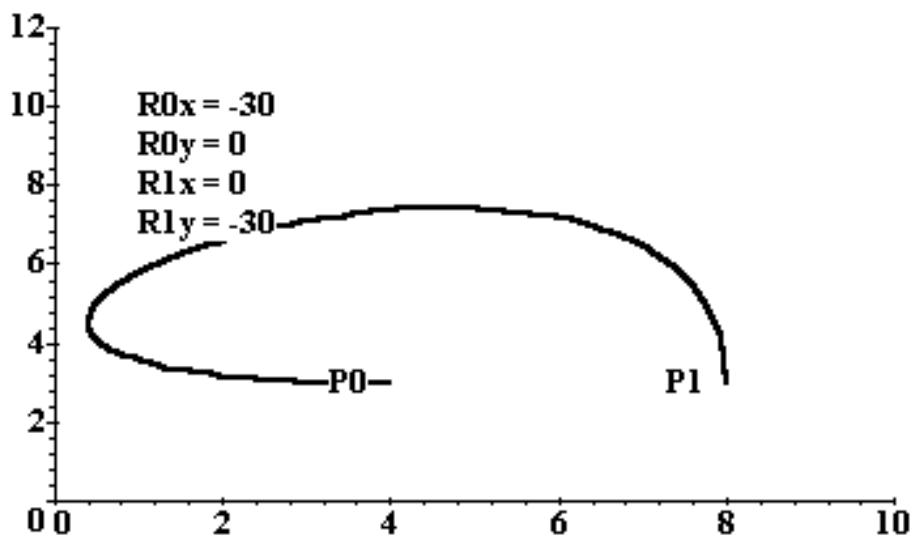
$$B_2(u) = u^3 - 2u^2 + u$$

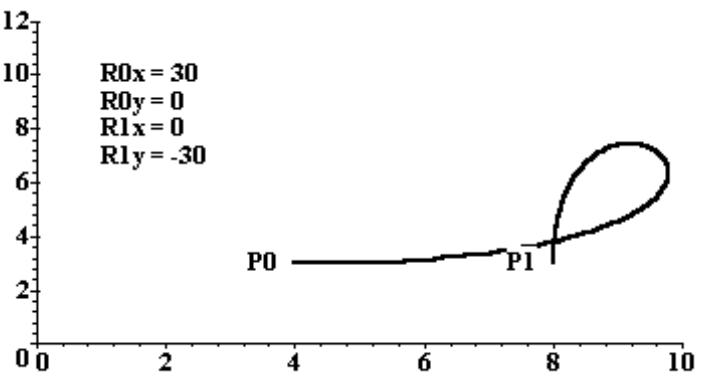
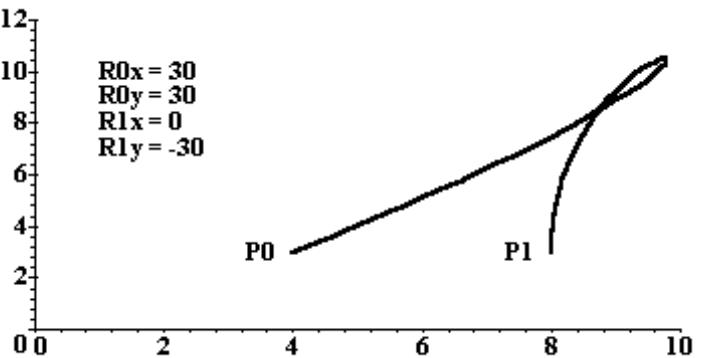
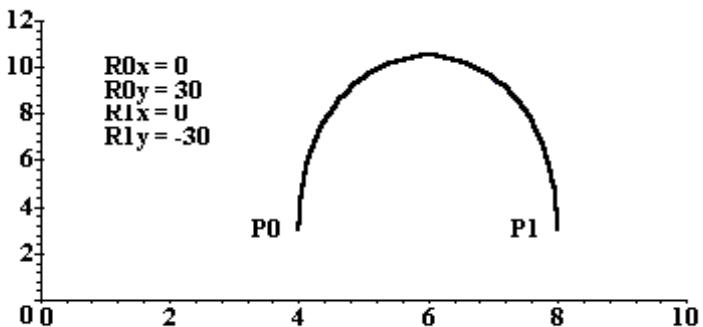
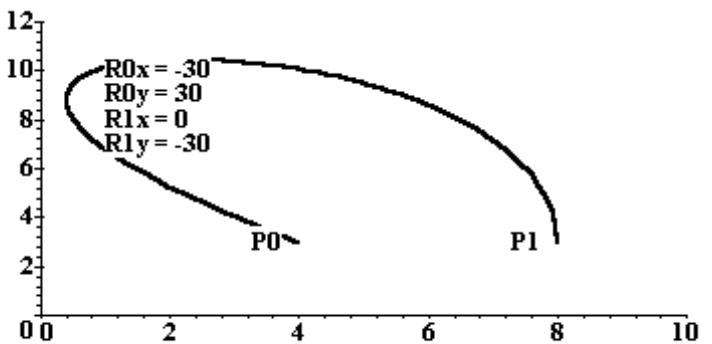


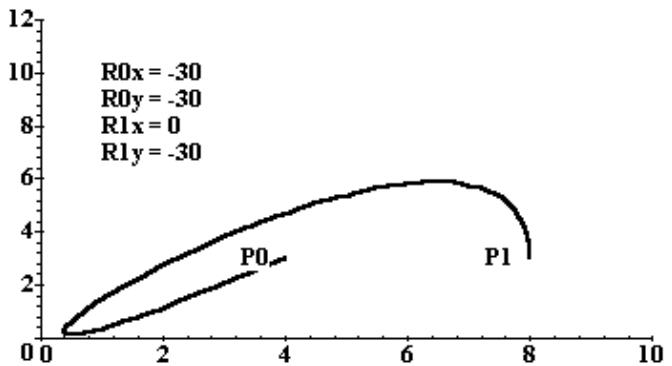
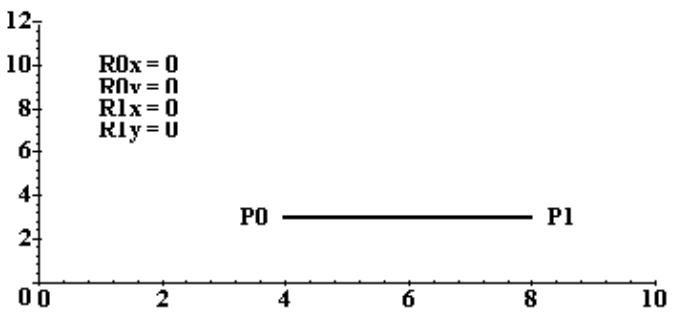
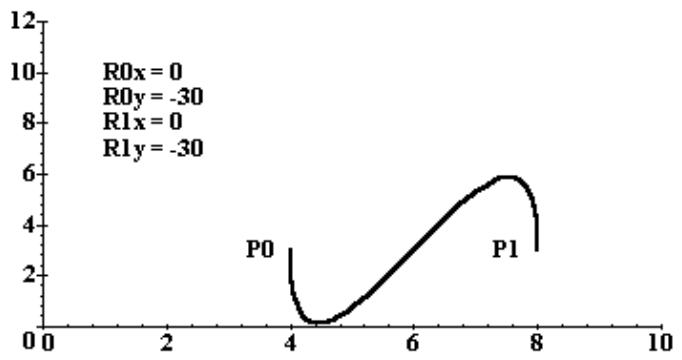
$$B_3(u) = u^3 - u^2$$

Dla krzywej o początku w punkcie (P_{0x}, P_{0y}) i końcu w (P_{1x}, P_{1y}) , sterowanie przebiegiem krzywej realizuje się przez dobór współrzędnych $(R_{0x}, R_{0y}), (R_{1x}, R_{1y})$ wyznaczających końce wektorów zaczepionych w punktach (P_{0x}, P_{0y}) i (P_{1x}, P_{1y}) .

1.







Krzywa Beziera

Bezier, P., *Numerical Control - Mathematics and Applications*, Wiley, London, 1972.

Projektowanie karoserii samochodów Renault.

Krzywa określona jest przez tzw. punkty kontrolne:

- punkt początkowy i końcowy (należą do krzywej),**
- inne punkty (które nie należą do krzywej).**

Definicja:

Dany jest zbiór $n+1$ punktów kontrolnych P_k

$$P_k = (P_{kx} \quad P_{ky} \quad P_{kz}) \quad k = 0, 1, \dots, n$$

gdzie P_{kx}, P_{ky}, P_{kz} są współrzędnymi x, y, z punktu kontrolnego.

Krzywa Beziera opisana jest układem równań parametrycznych

$$x(u) = \sum_{k=0}^n P_{kx} B_{kn}(u)$$

$$y(u) = \sum_{k=0}^n P_{ky} B_{kn}(u) \quad 0 \leq u \leq 1$$

$$z(u) = \sum_{k=0}^n P_{kz} B_{kn}(u)$$

gdzie

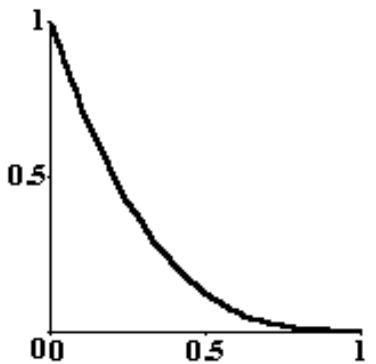
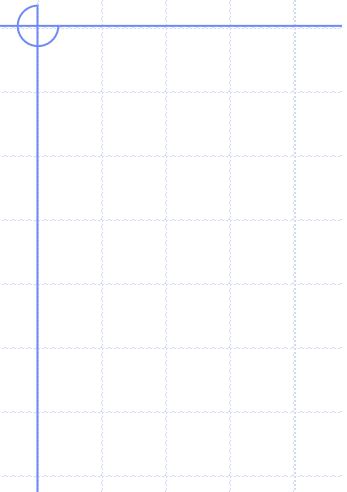
$$B_{kn}(u) = C(n, k)u^k(1-u)^{n-k}$$

przy czym

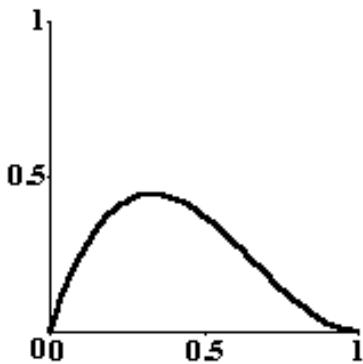
$$C(n, k) = \frac{n!}{k!(n-k)!}$$

Wielomiany $B_{kn}(u)$ noszą nazwę wielomianów Bernsteina.

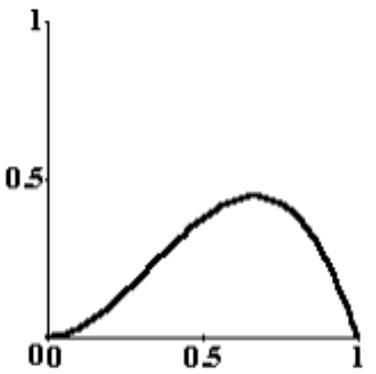
Przykładowe przebiegi wielomianów dla $n = 3$.



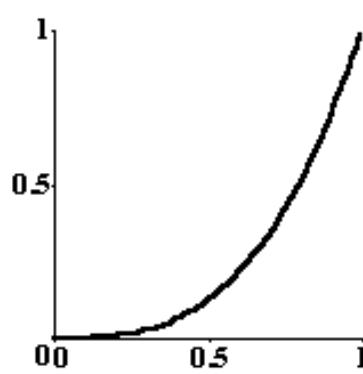
$$B_{03}(u) = (1-u)^3$$



$$B_{13}(u) = 3u(1-u)^2$$

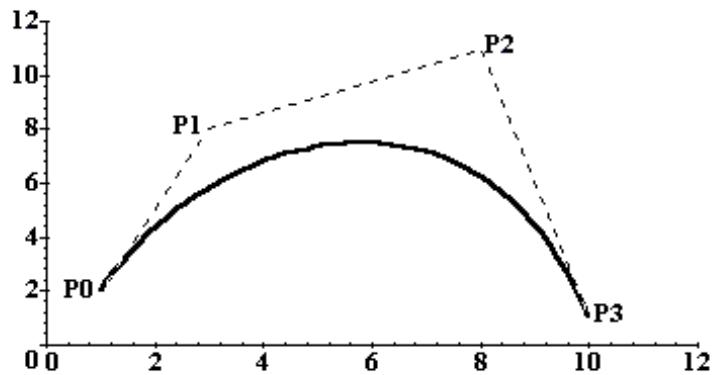
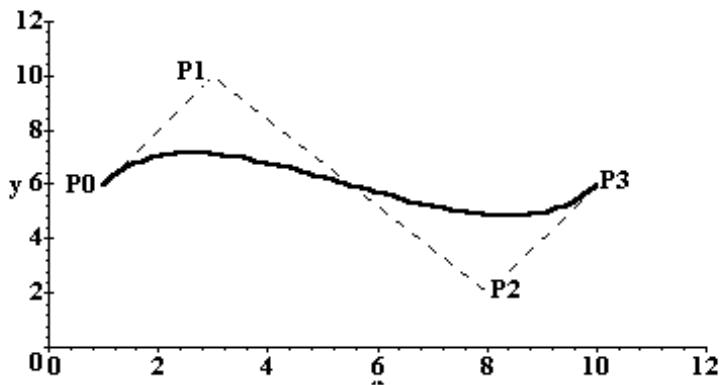
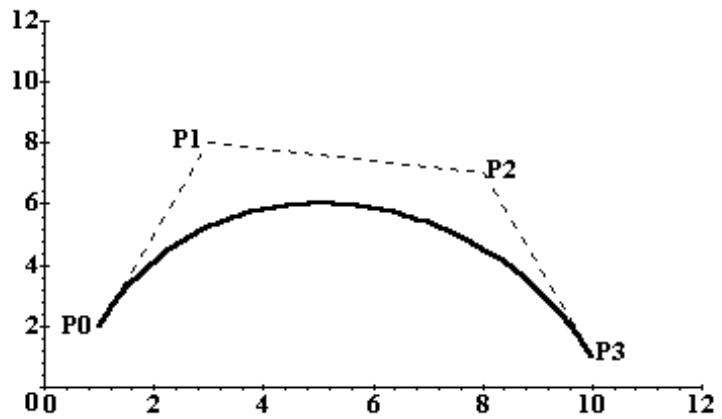


$$B_{23}(u) = 3u^2(1-u)$$

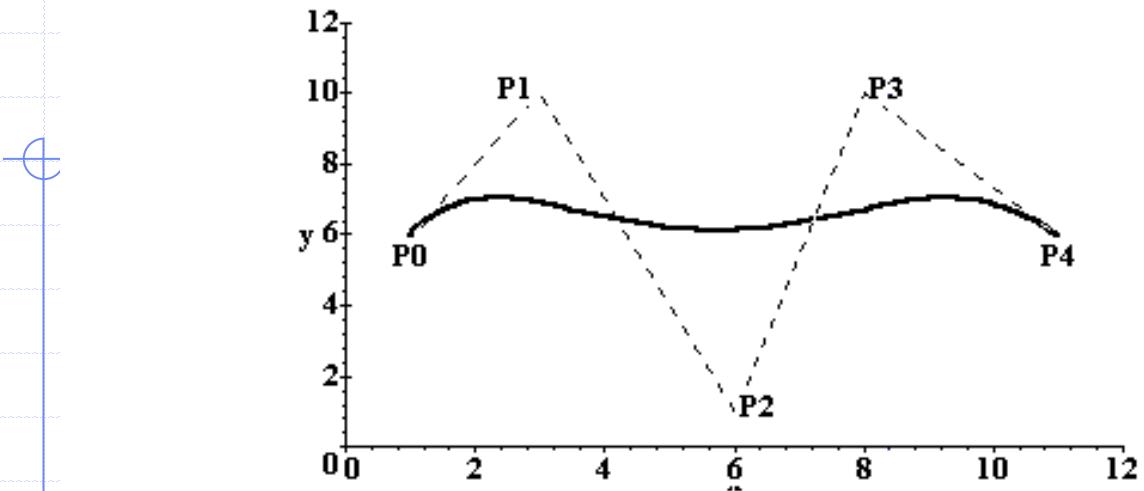


$$B_{33}(u) = u^3$$

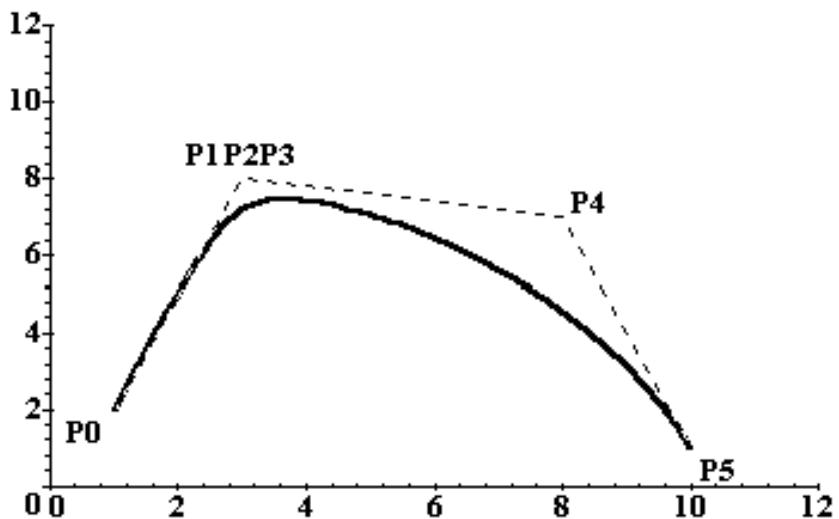
Przykłady krzywych Beziera (4 punkty kontrolne):



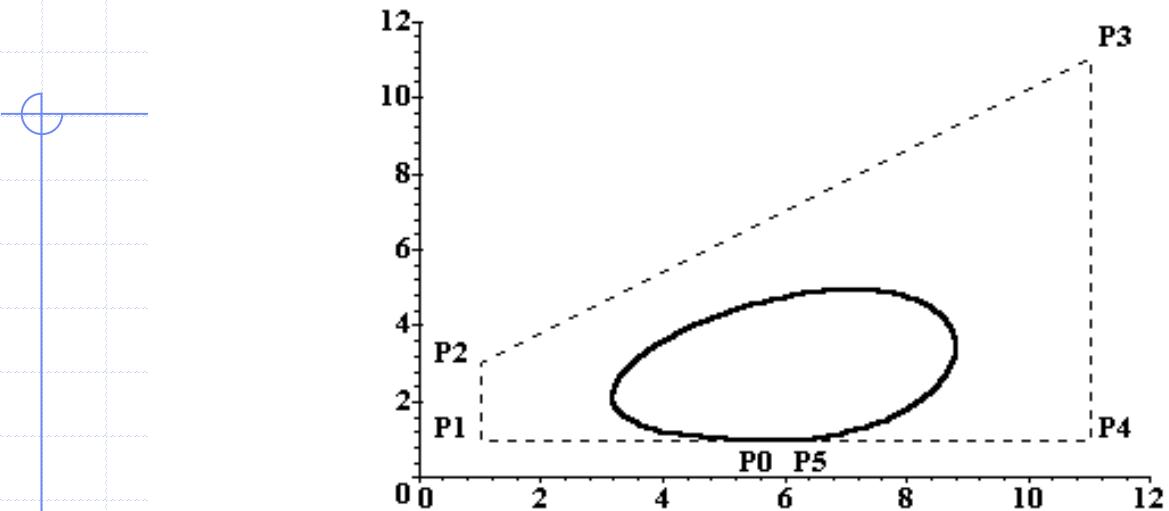
Przykład krzywej Beziera (5 punktów kontrolnych):



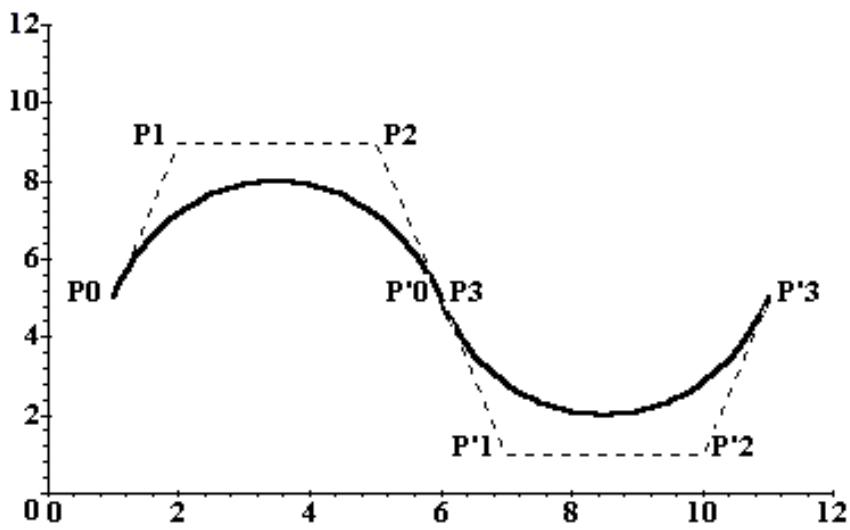
Przykład krzywej Beziera (6 punktów kontrolnych , punkty powielone)



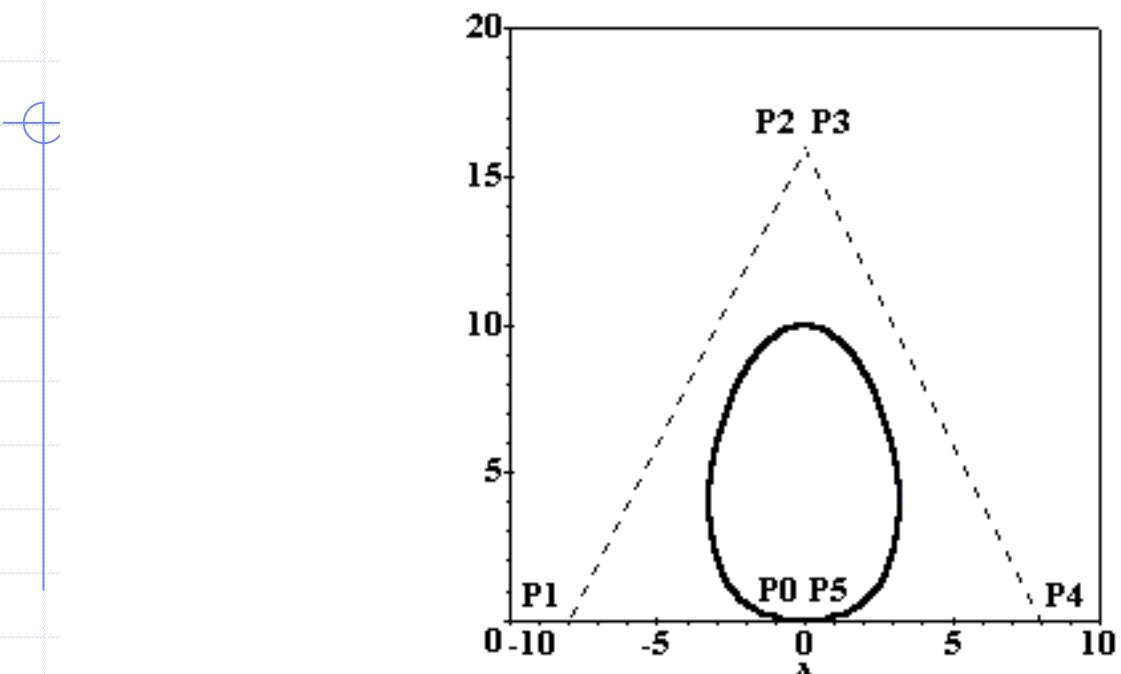
Przykład krzywej Beziera (6 punktów kontrolnych , zamknięty układ punktów)



Przykład gładkiego łączenia dwóch krzywych Beziera:



Przykład zastosowania:



Krzywa zamknięta ,jeden punkt kontrolny powielony.
Równania parametryczne krzywej mają postać:

$$x(u) = -40(u^5 - 5u^4 + 6u^3 - 4u^2 + u)$$

$$y(u) = 160(u^4 - 2u^3 + u^2)$$

$$0 \leq u \leq 1$$



Po dokonaniu obrotu krzywej wokół osi y otrzymuje się parametryczne równania powierzchni.

$$x(u, v) = -40(u^5 - 5u^4 + 6u^3 - 4u^2 + u)\cos(2\pi v)$$

$$y(u, v) = 160(u^4 - 2u^3 + u^2)$$

$$z(u, v) = -40(u^5 - 5u^4 + 6u^3 - 4u^2 + u)\sin(2\pi v)$$

$$0 \leq u \leq 1$$

$$0 \leq v \leq 1$$

Uzyskana powierzchnia wygląda następująco:

