GRAKO: PODSTAWY GRAFIKI 2W

Wstęp do renderingu

- Wizualizacja 2W
  - ograniczenia rastra
  - linia w grafice rastrowej
  - łuk w rastrze
- Spójność obiektów
- Obcinanie

Wstęp do modelowania

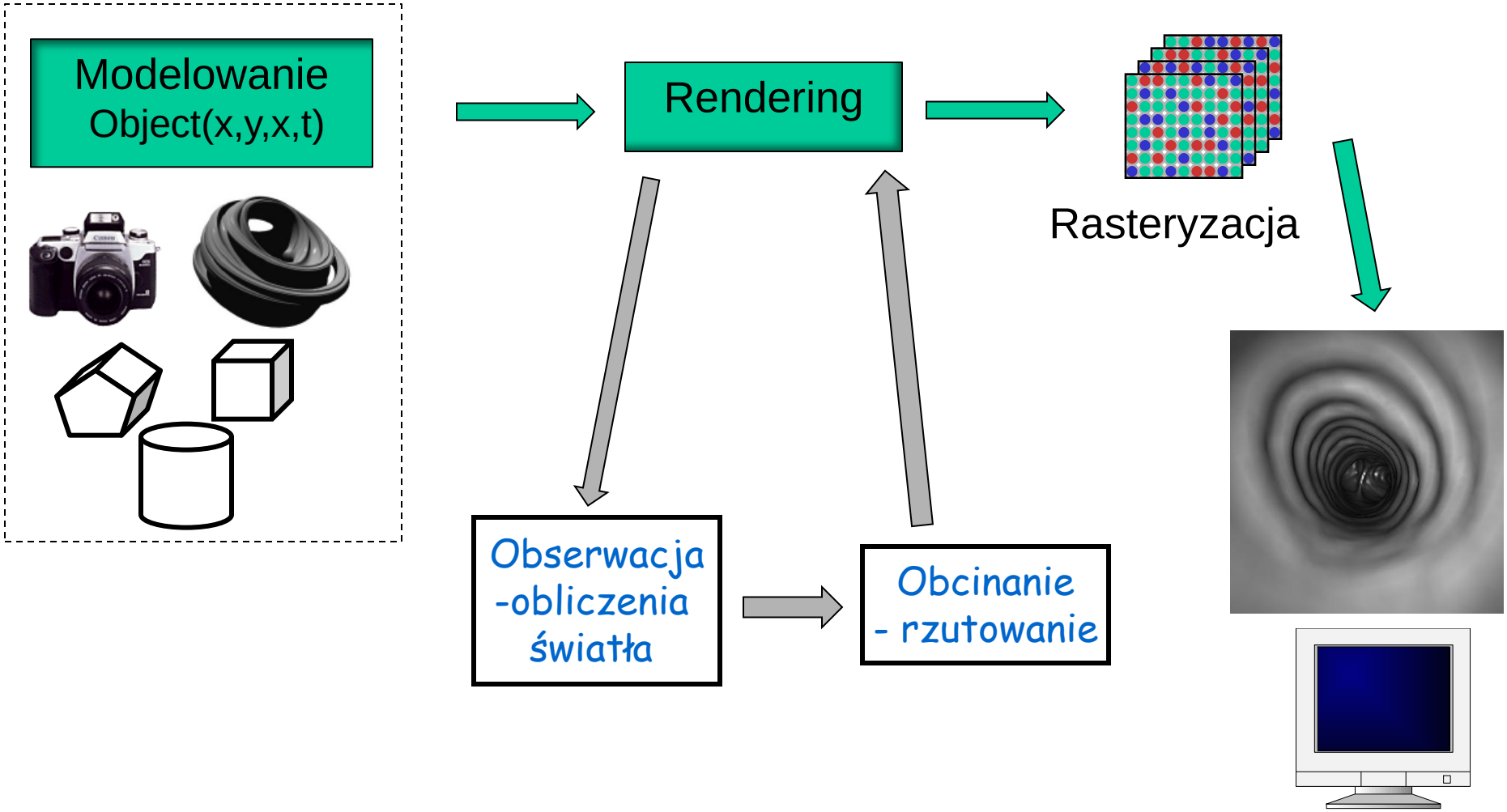
- Obiekty i transformacje 2W

Niektóre wykorzystane tutaj materiały pochodzą z:

http://wazniak.mimuw.edu.pl/index.php?title=Grafika_komputerowa_i_wizualizacja

http://www.zsk.ict.pwr.wroc.pl/zsk/dydaktyka/gk/

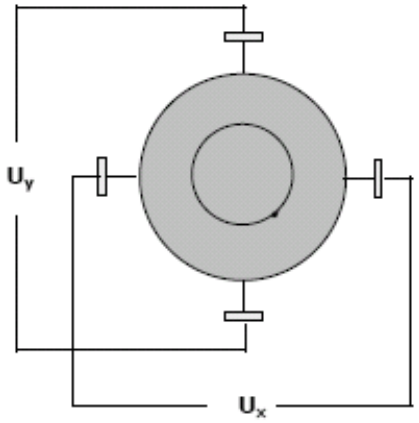
Schemat działań graficznych

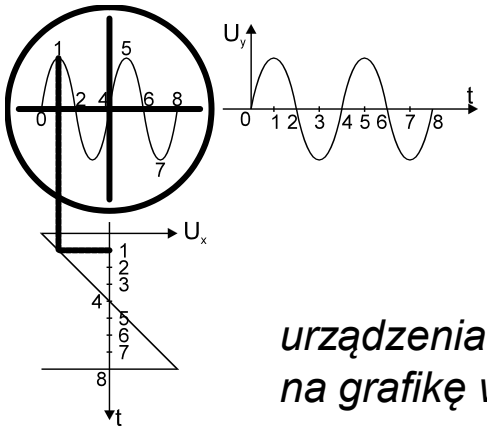


DWIE GRAFIKI

Reprezentacja obrazu - wizualizacja

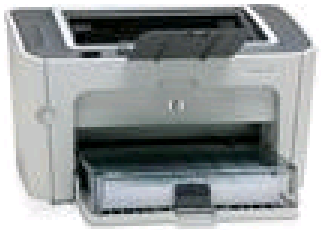
- Wektorowa (obiektowa) – prymitywy geometryczne tworzą obiekty o określonych cechach (kolor, tekstura, powierzchnia), opisanych za pomocą formuł matematycznych





urządzenia podatne na grafikę wektorową

- Rastrowa (pikselowa) – z podziałem na dyskretną siatkę ‘kolorowanych’ elementów



urządzenia zorientowane na obraz rastrowy

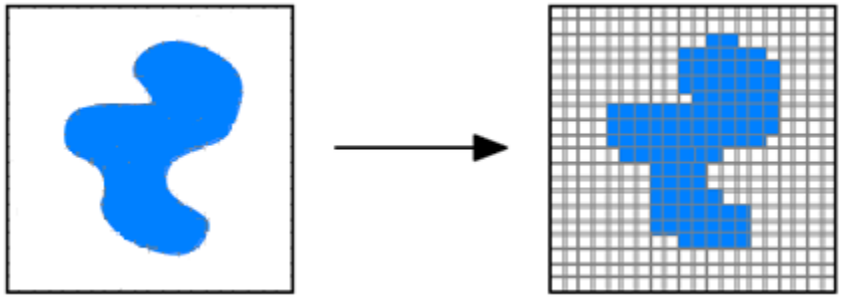
Reprezentacja wyświetlana

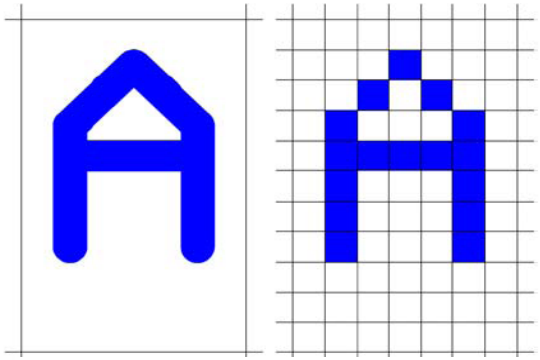
- Wektorowa
  - Skalowalność
  - Rozdzielczość ‘nieograniczona’
  - Linie – ciągłe, dowolnie cienkie
  - Złożoność obrazu – ograniczona (niekiedy) czasem poświaty
  - Wypełnienie konturów – trudne
  - Brak koloru
  - Drogi sprzęt

- Rastrowa
  - Brak skalowalności
  - Rozdzielczość ograniczona
  - Linie – schodkowe
  - Złożoność obrazu – bardzo duża
  - Wypełnienie konturów – łatwe
  - Kolor ‘dowolny’
  - Tani sprzęt

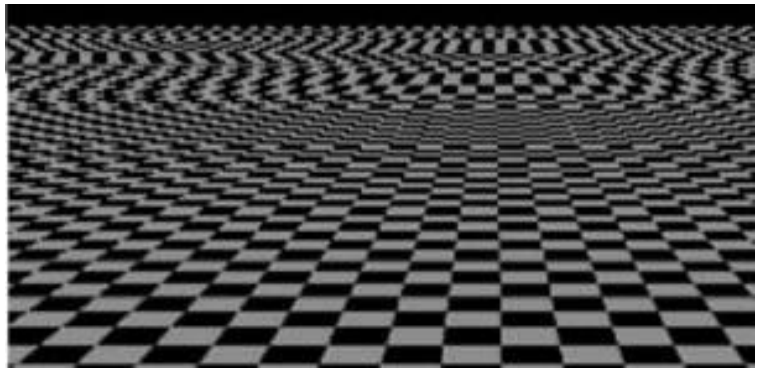
Grafika rastrowa – $R^2 \rightarrow N^2$

- Położenie rysowanych figur (obiektów) jest ograniczone zwykle prostokątną strukturą pikseli
- Cechy
  - Zależność od urządzeń, położenia
  - Znaczenie rozmiaru piksela
- Ograniczenia:
  - Brak skalowania
  - Linia schodkowa (brak gładkości – dyskretyzacja kształtu)
  - Operacje rastrowe (filtracje)
  - Aliasing
  - Złożoność
  - Zmiana barwy





wektorowo *rastrowo*



efekt aliasingu

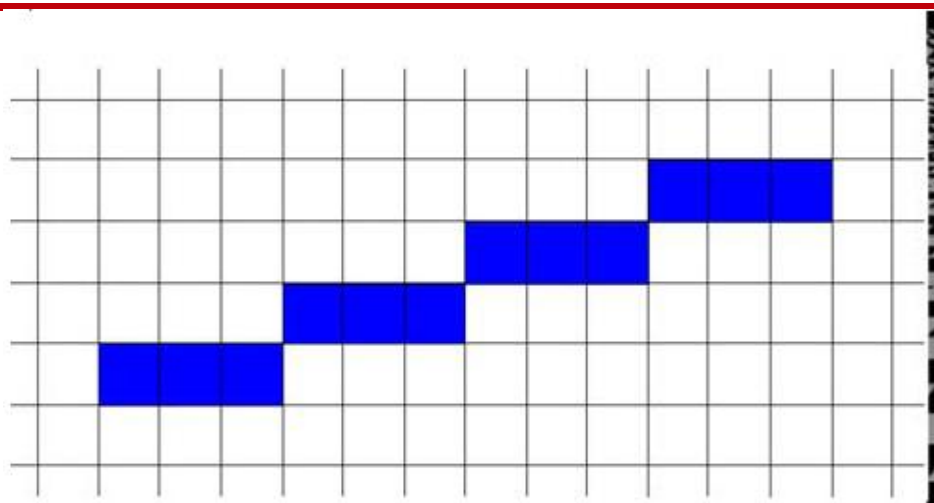
Grafika rastrowa

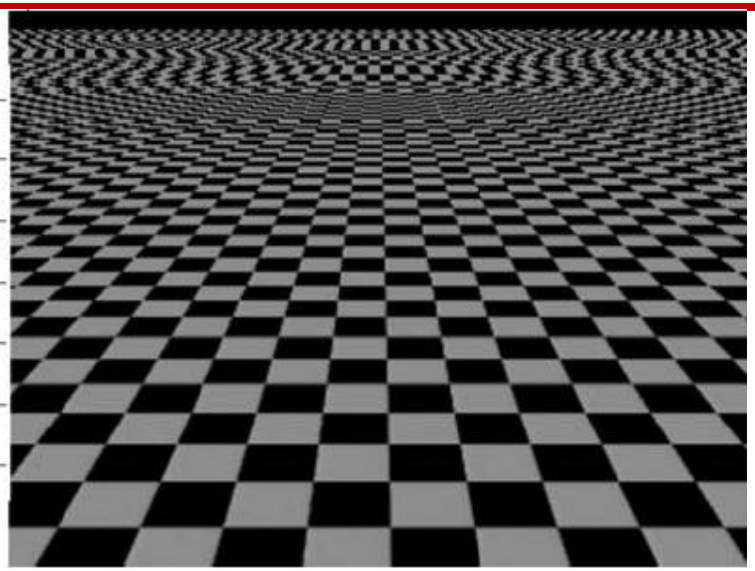
- Metody rasteryzacji:
  - Zwiększanie rozdzielczości matrycy pikselowej
  - Rozmywanie obiektów o wysokogradientowych krawędziach, korekcja koloru ukośnych odcinków, pogrubianie linii
  - Ważenie jasności (piksel świeci proporcjonalnie do rozmiaru pokrycia przez dany obiekt)

  oraz

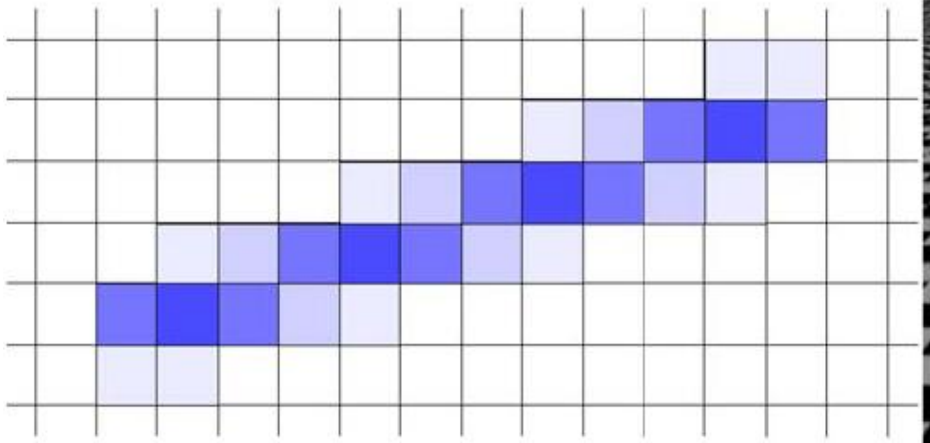
  - Przyspieszanie obliczeń
  - Projektowanie (myślenie) wektorowe

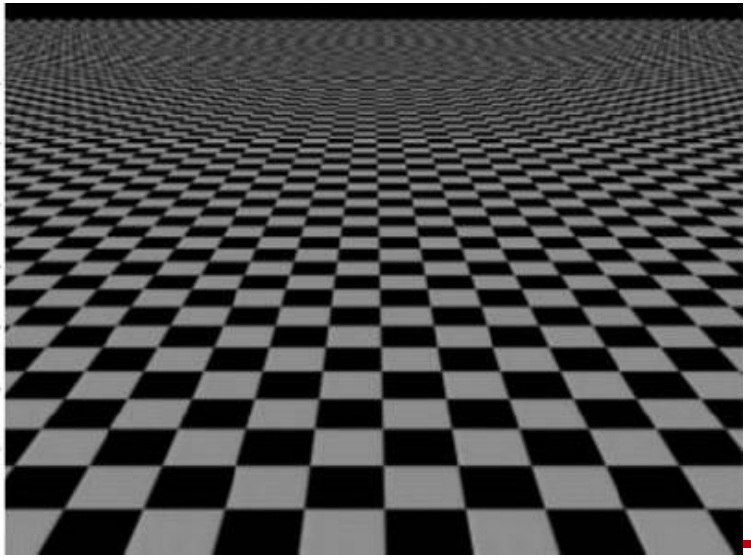
Metoda: antyaliasing





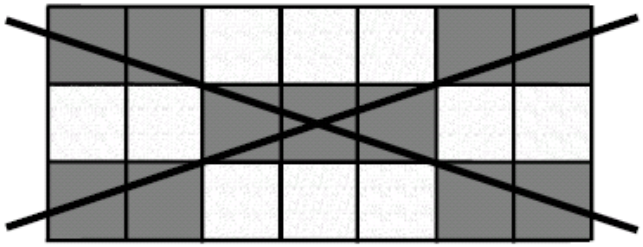
Rozmycie ostrych krawędzi poprawia widziany efekt



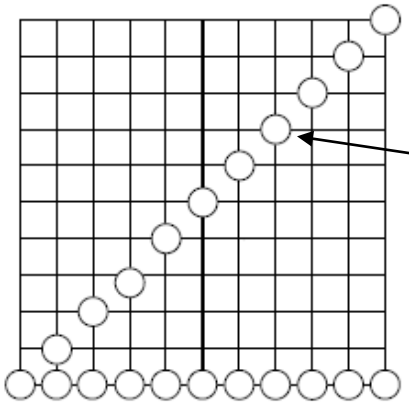


Problemy: linie

- Przecięcie dwóch linii



- Różna jasność linii

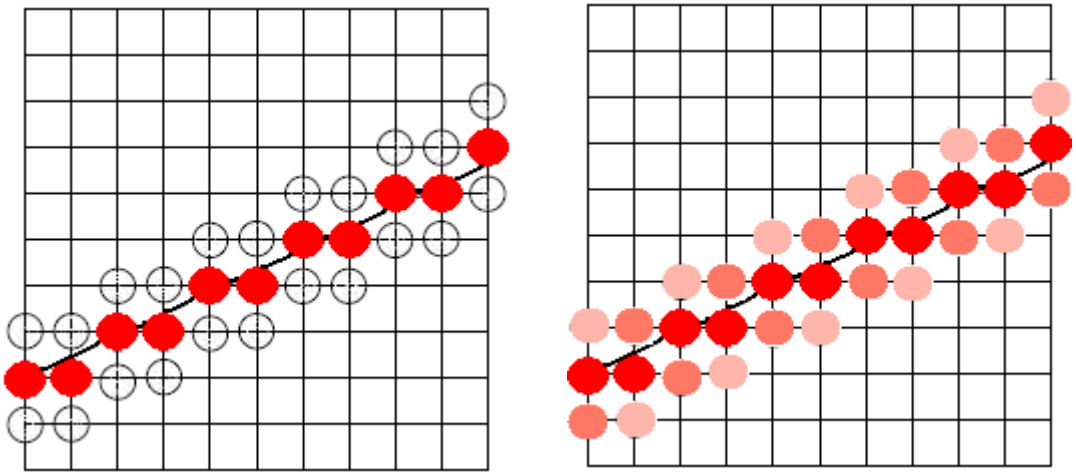


ciemniejszy odcinek – rozjaśniamy!

$\sqrt{2}$ raza dłuższy odcinek – piksele powinny świecić $\sqrt{2}$ raza jaśniej

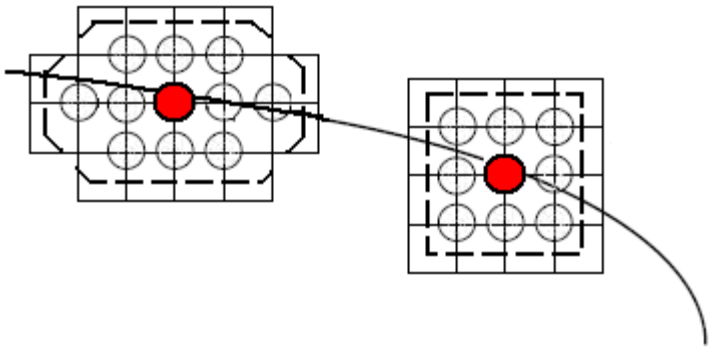
Antidotum: pogrubianie linii

- Powielanie i wprowadzanie tonów pośrednich



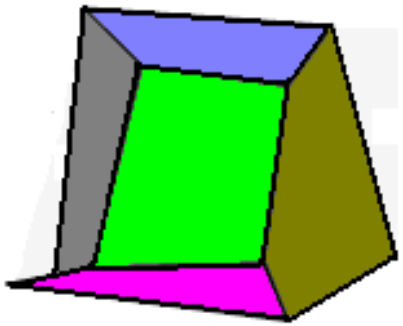
w kolumnie lub wierszu zależnie od nachylenia

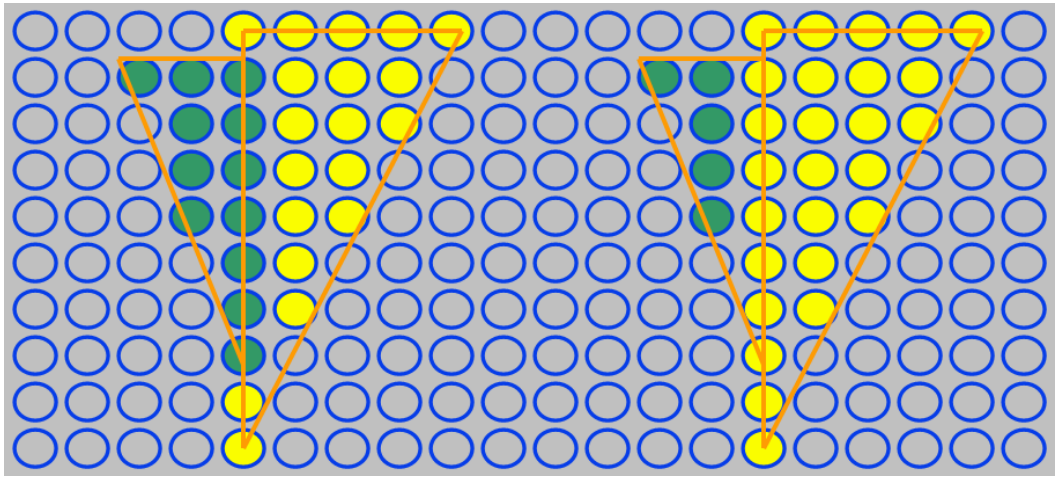
- Powielanie adaptacyjne - dynamiczny pędzel (pióro) z powielaniem pikseli zależnie od nachylenia krzywej



Metody: brzegi figur

- Wypełnianie brzegów wielokątów (reguła przynależności brzegu)



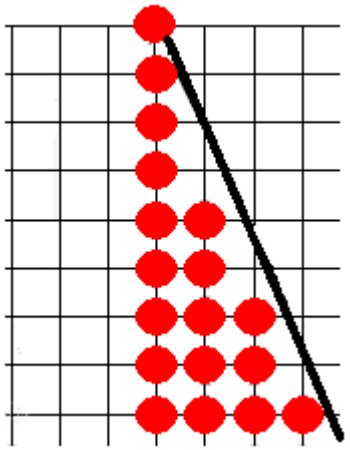


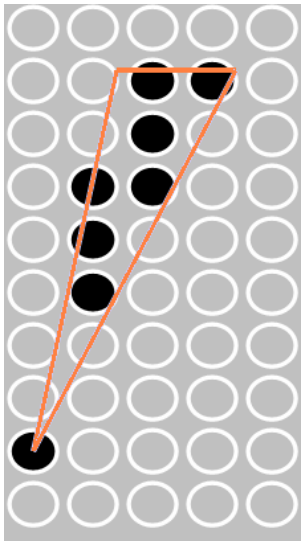
kolorujemy jedynie wewnątrz figur (nie na brzegach), następnie rysujemy lewy brzeg oraz dolny brzeg wielokątów

lepiej: wielotonowe barwy pośrednie

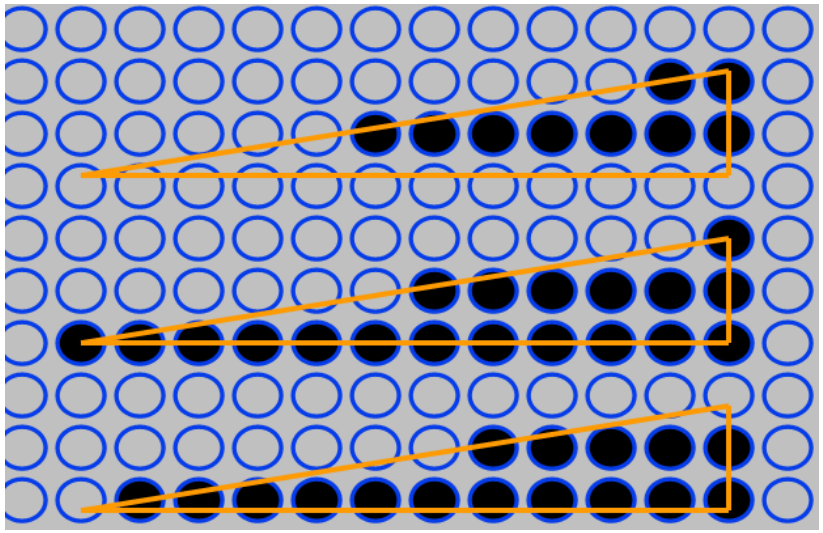
Problemy: zmienne kształty figury

- Kliny, drzazgi – różne kształty zależnie od położenia (antidotum: wypełnianie wielotonowe, zwiększanie rozdzielczości)





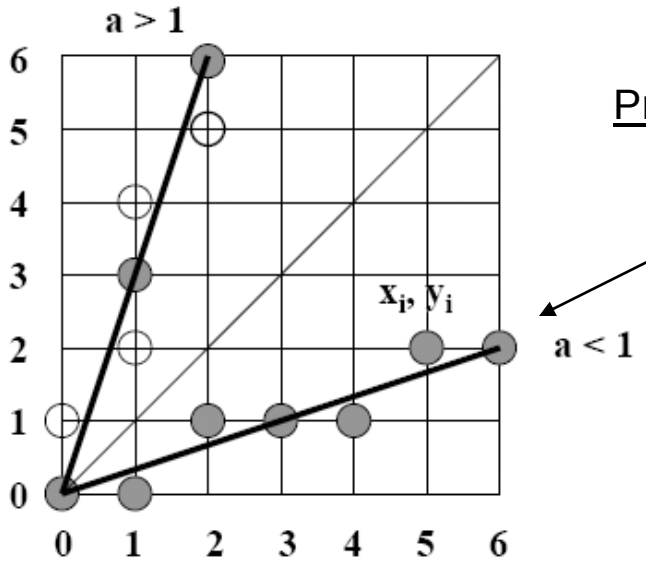
przesuwający się trójkąt



Jasność pikseli jest proporcjonalna do powierzchni piksela pokrytej daną figurą

RYSOWANIE PRYMITYWÓW - ZŁOŻONOŚĆ OBLICZENIOWA

Problem redukcji złożoności obliczeniowej – rysowanie odcinka



Procedura wyznaczania:

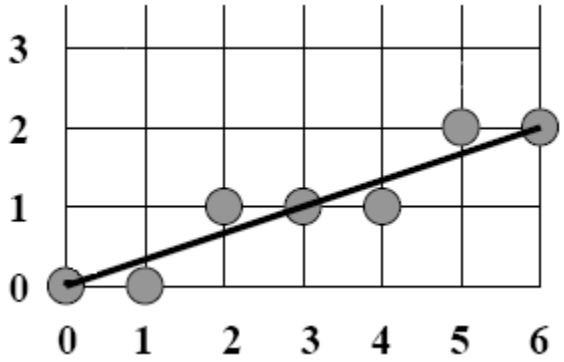
$$y = ax + b$$

$$0 \le a \le 1$$

$$b = 0$$

dla $a > 1$ liczymy $y = \frac{1}{a}x$

i symetrycznie odbijamy



y=ax

Koszt:

• y= **[**a·x**]** - mnożenie rzeczywistoliczbowe plus zaokrąglanie

lub

• $\Delta y = a \cdot \Delta x \rightarrow y_{i+1} = [y_i + a]$ bo $\Delta x = 1$

- dodawanie **rzeczywistoliczbowe** plus zaokrąglanie

Rysowanie odcinka – algorytm Bresenhama

- Założenia:
  - $x_{i+1}=x_i+1$, $0 \le a \le 1$
  - operacje tylko stałoprzecinkowe
- Metoda:
  - Inicjujemy zmienne pomocnicze opisujące wyznaczany odcinek o początku (x_1,y_1) i końcu (x_2,y_2)

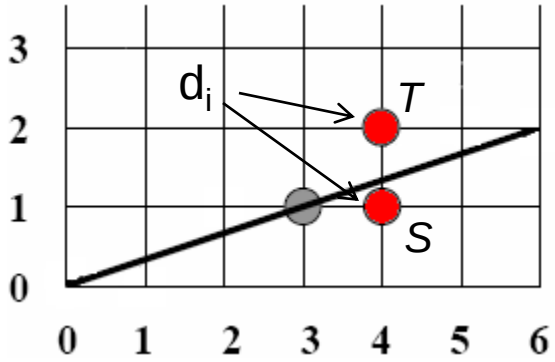
    $d_0=2\cdot(y_2-y_1)-(x_2-x_1)$

    $p_1=2\cdot(y_2-y_1)$

    $p_2=2\cdot[(y_2-y_1)-(x_2-x_1)]$
  - Na podstawie d_i wybieramy punkt *S* lub *T* (rysunek) bliższy teoretycznej prostej (odcinkowi) i aktualizujemy zmienną d_{i+1} *w* każdym kolejnym kroku

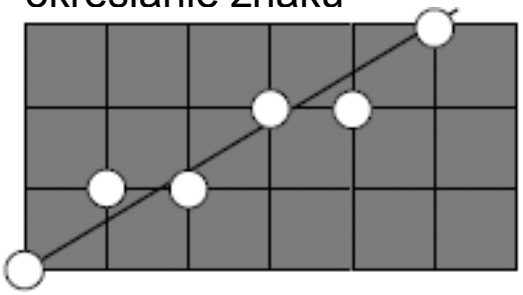
    jeśli $(d_i<0)$ to $(x_{i+1},y_{i+1})=S$; $d_{i+1}=d_i+p_1$;

    w p.p. $(x_{i+1},y_{i+1})=T$; $d_{i+1}=d_i+p_2$;

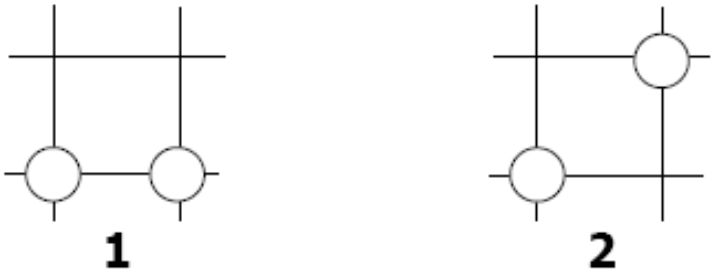


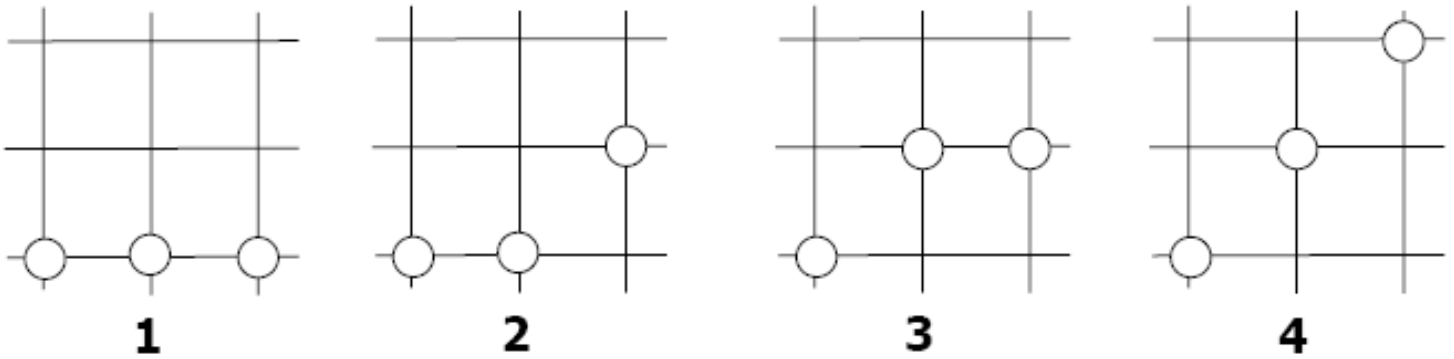
z punktów S i T wybieramy bliższy odcinkowi, czyli w tym przypadku S

złożoność: dodawanie całkowitoliczbowe plus określanie znaku



Modifikacja algorytmu Bresenhama





Algorytm z podwójnym krokiem – wybiera się od razu parę punktów

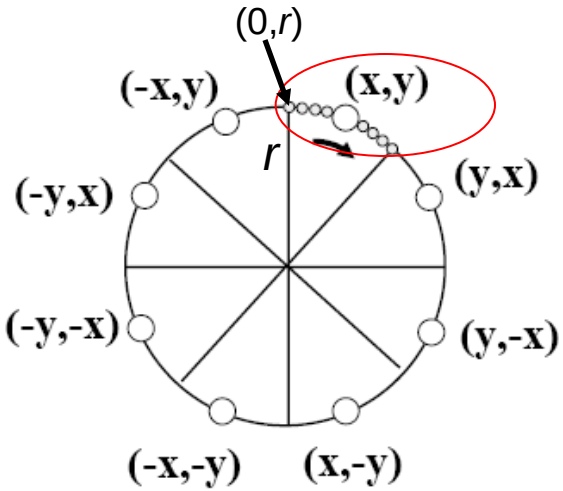
Rysowanie okręgu – algorytm Bresenhama

Duża złożoność liczenia ze wzoru na okrąg lub r. parametrycznych:

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

$$x = r\cos\alpha$$

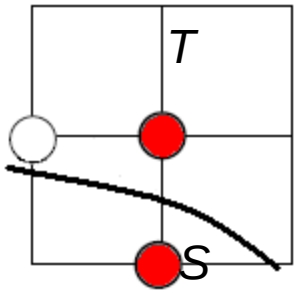
$$y = r\sin\alpha, \quad 0 \le \alpha < 2\pi$$



Algorytm Bresenhama dla okręgu:

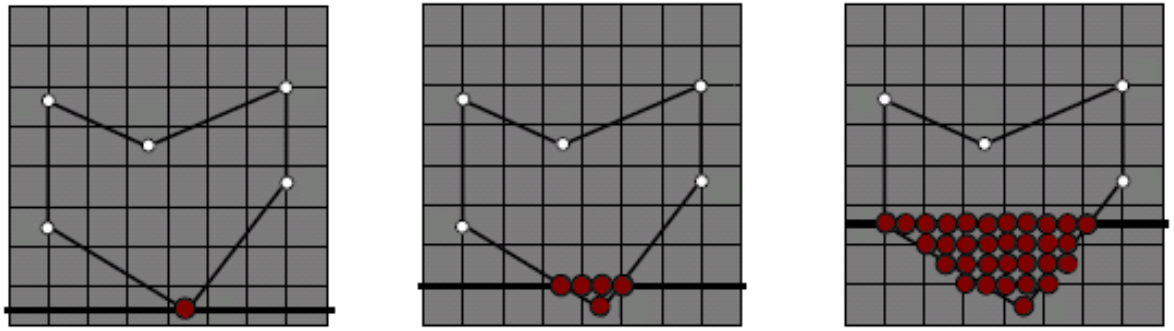
- Stałoprzecinkowy
- Założenia: zaczynamy od punktu (0,*r*) i rysujemy 1/8 okręgu (resztę przez symetrię)
- Metoda
  - $p_1=3-2r$
  - w kolejnym kroku:

    jeśli ($p_i<0$) to (x_{i+1},y_{i+1}) =*T*; $p_{i+1}=p_i+4x_{i-1}+6$;
  - w p.p. (x_{i+1},y_{i+1}) =S; $p_{i+1}= p_i+4(x_{i-1}-y_{i-1})+10$



Wypełnianie obszarów

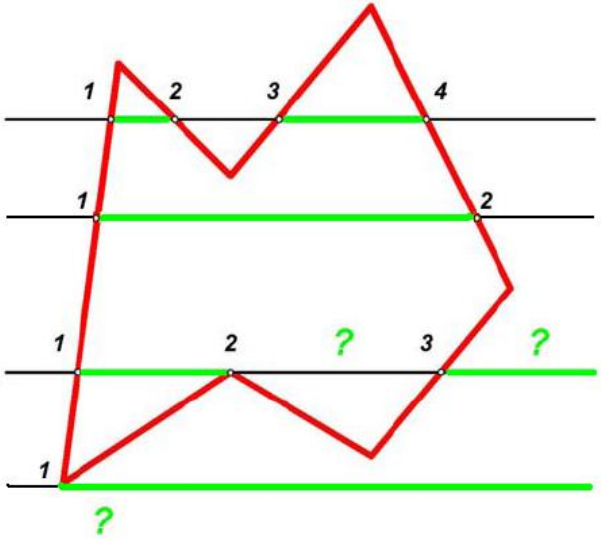
- Za pomocą linii poziomych (wypełniamy kolorem linia po linii)



- Wykorzystuje się kontrolę parzystości

<u>nieparzyste</u> – wejście do obiektu, czyli malujemy!

<u>parzyste</u> – wyjście z obiektu, czyli nie malujemy!



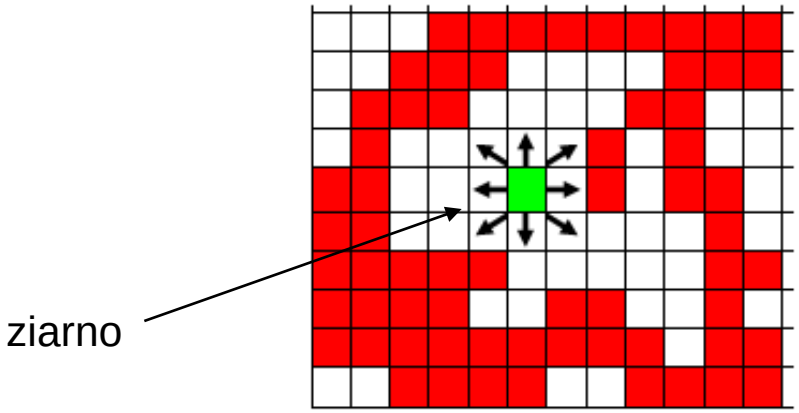
<u>Modyfikacja:</u>

a) Uwzględnienie końców przecinanej krawędzi (jedynie leżących po dwóch stronach)
b) pomijanie krawędzi poziomych (logiczne zlepianie jej końców)

przypadki szczególne – ekstrema z analizą końców odcinków, odcinki poziome

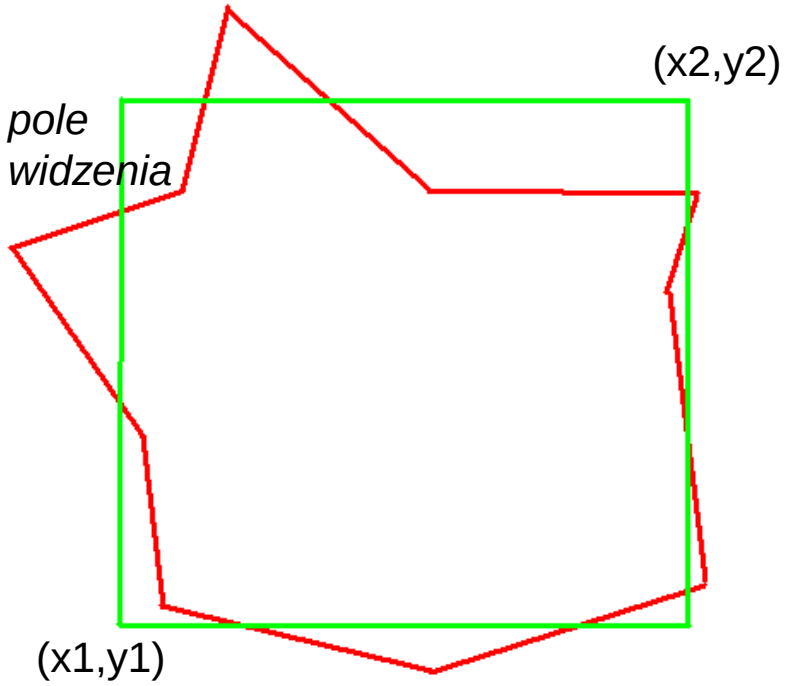
Wypełnianie obszarów

- Przez spójność (rozrost regionu)



- Warunek rozrostu (brak koloru, podobieństwo kolorów)

Obcinanie obiektu do widocznego fragmentu



$$x1 < x < x2 \quad \wedge \quad y1 < y < y2$$

Model: wielokąt określają jego wierzchołki, odcinki zaś ich końce

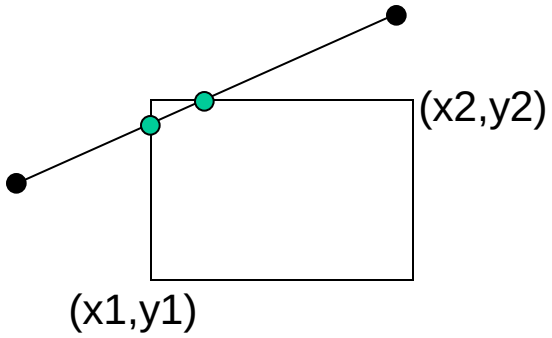
Konwersja do bufora wyświetlania (piksel po pikselu) i usunięcie wszystkich pikseli nie mieszczących się z widoku – rozwiązanie?

Obcinanie odcinków

- Sprawdzanie końców odcinków:
  - oba wewnątrz widoku: odcinek zostawiamy!

$$x_1 \le x_p \le x_2 \quad \wedge \quad y_1 \le y_p \le y_2$$

  - jeden na zewnątrz: przycinamy do okna widoku (obliczamy drugą współrzędną przycięcia)
  - oba końce na zewnątrz - rozstrzyganie: odrzucamy odcinek czy przycinamy?
    - obliczanie punktów przecięcia z widokiem i sprawdzanie, czy mają punkty wspólne z widokiem

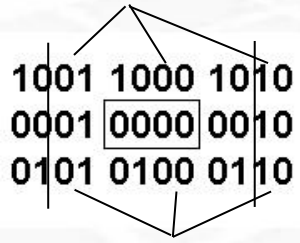


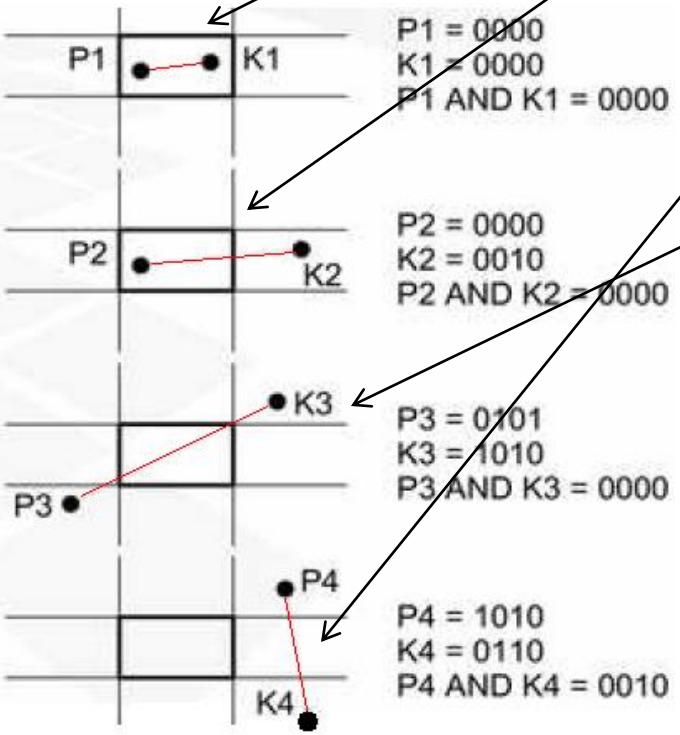
Obcinanie odcinka (przyspieszenie)

algorytm Cohena-Sutherlanda

-dzielimy na regiony z kodem binarnym

-końcom odcinka przypisujemy kody regionów



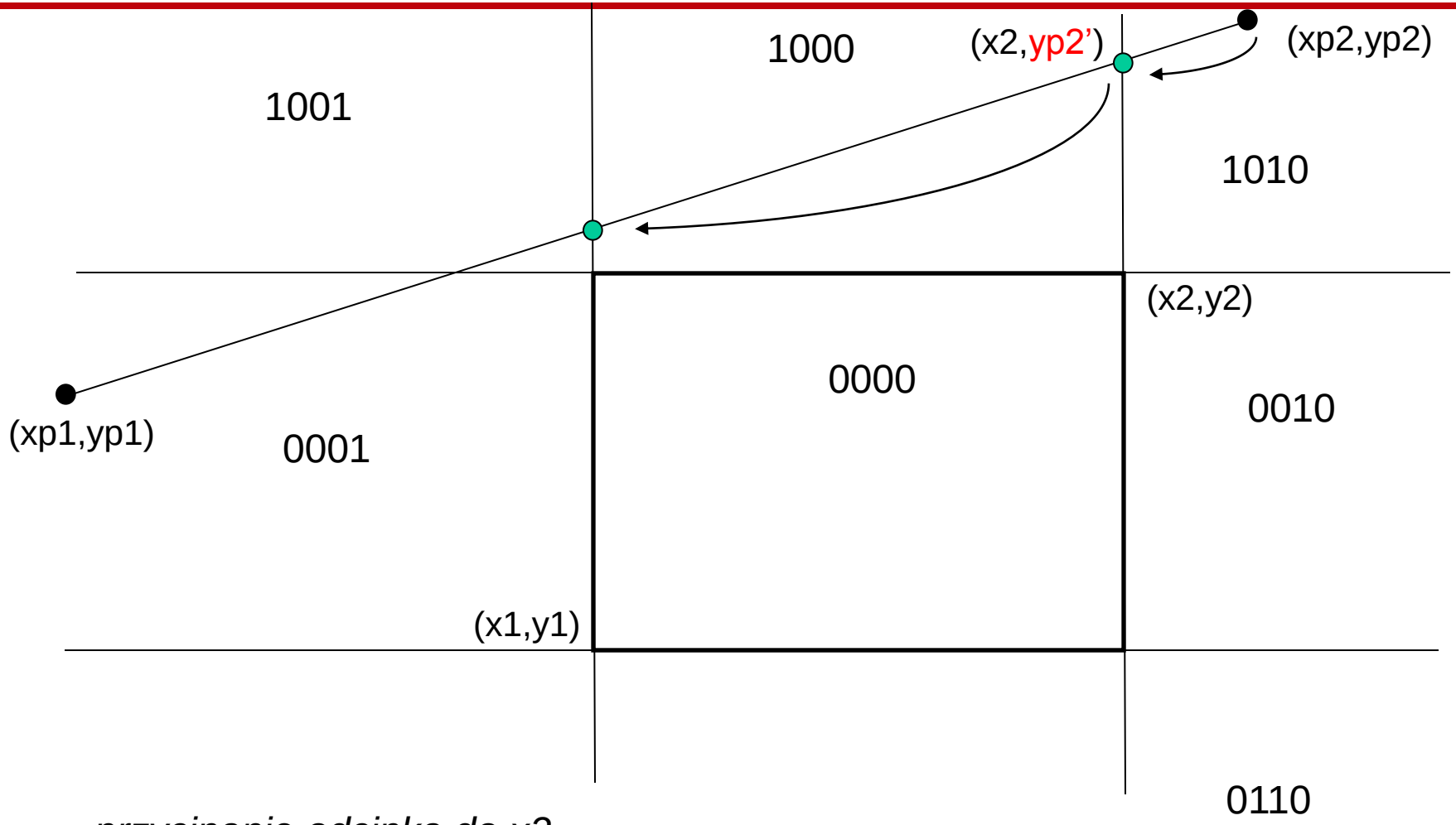


- Sprawdzanie końców odcinków:
  - oba wewnątrz widoku: odcinek zostawiamy
    - oba końce 0000
  - jeden na zewnątrz (AND końców daje 0): przycinamy do okna widoku
  - oba końce na zewnątrz - rozstrzyganie: odrzucamy odcinek czy przycinamy?
    - AND końców daje wynik niezerowy – odrzucamy odcinek zewnętrzny
    - AND końców daje 0 – sytuacja niejednoznaczna

    - obliczanie punktów przecięcia z widokiem i sprawdzanie, czy należą do widoku:

      przycięcie odcinka do najbliższej prostej ograniczającej widok (według ustalonej kolejności), określenie nowego regionu końca odcinka i powtórzenie procesu sprawdzania końców (maksimum 4 kroki iteracji)

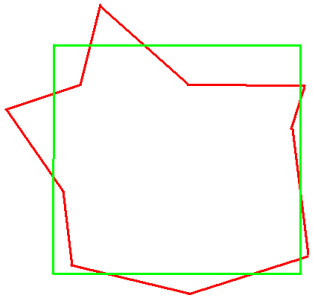
Algorytm Cohena-Sutherlanda (wyjaśnienie)

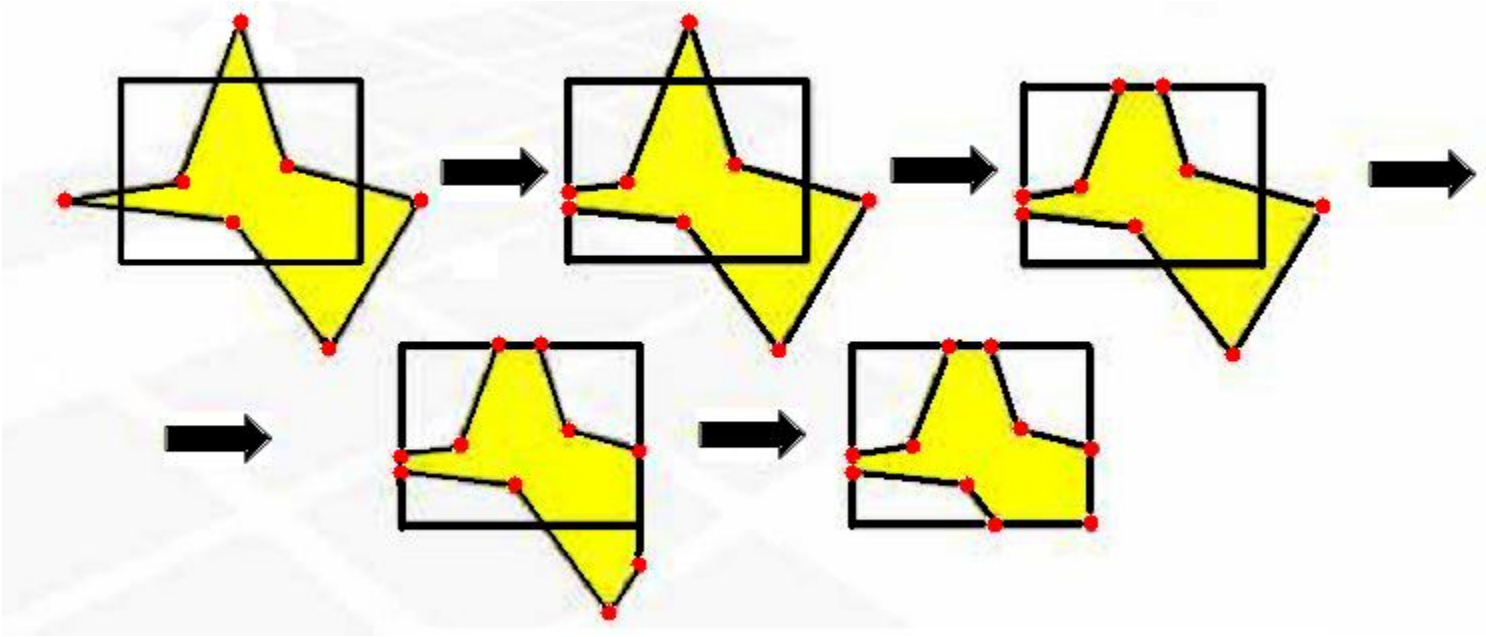


przycinanie odcinka do x2

yp2' = yp1 + a(x2– xp1), a=(yp2-yp1)/(xp2-xp1)

Powracamy: obcinanie wielokąta



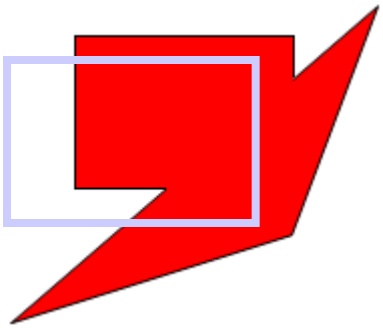


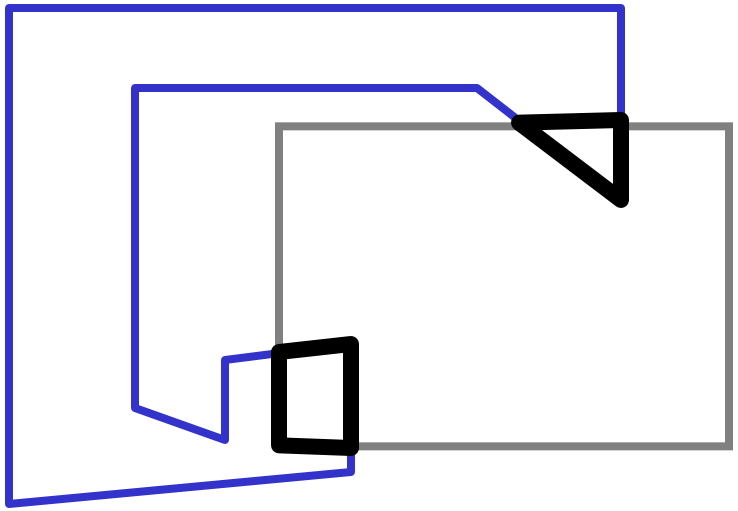
Algorytm Sutherlanda- Hodgmana

Rysunek z wazniaka

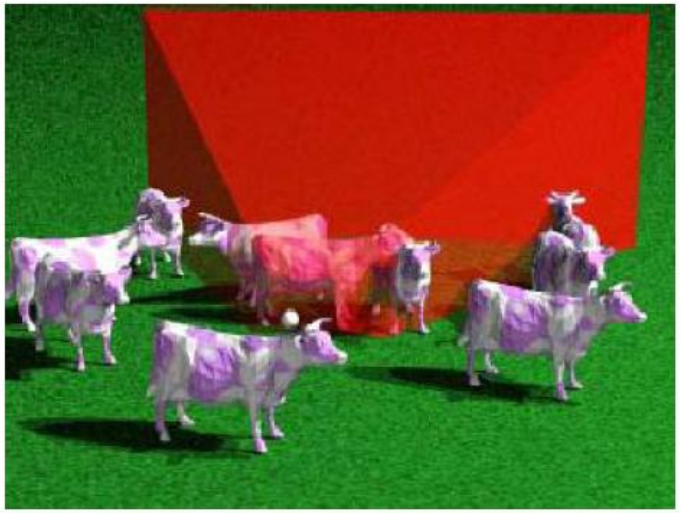
Przycinanie nie jest łatwe

- Z jednego wielokąta – kilka





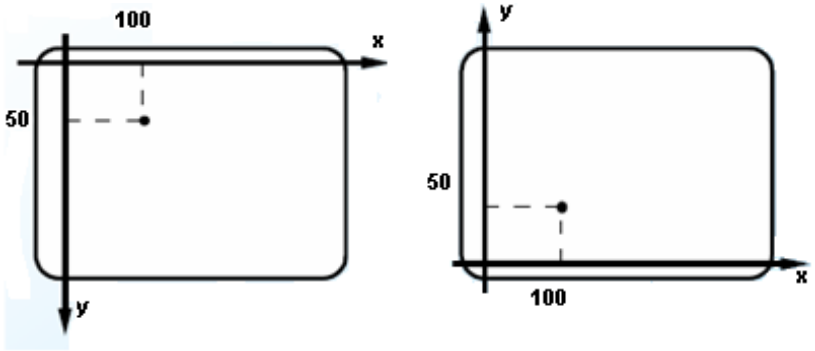
- W przestrzeni 3W



PRZEKSZTAŁCENIA

Przekształcenia 2W – układy współrzędnych

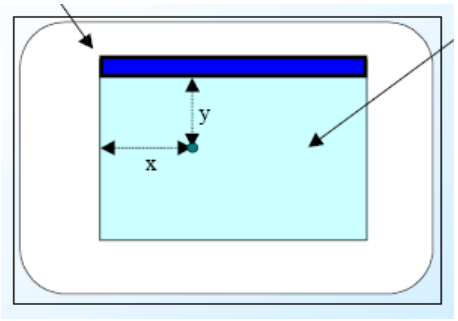
- konwencje adresowania pikseli



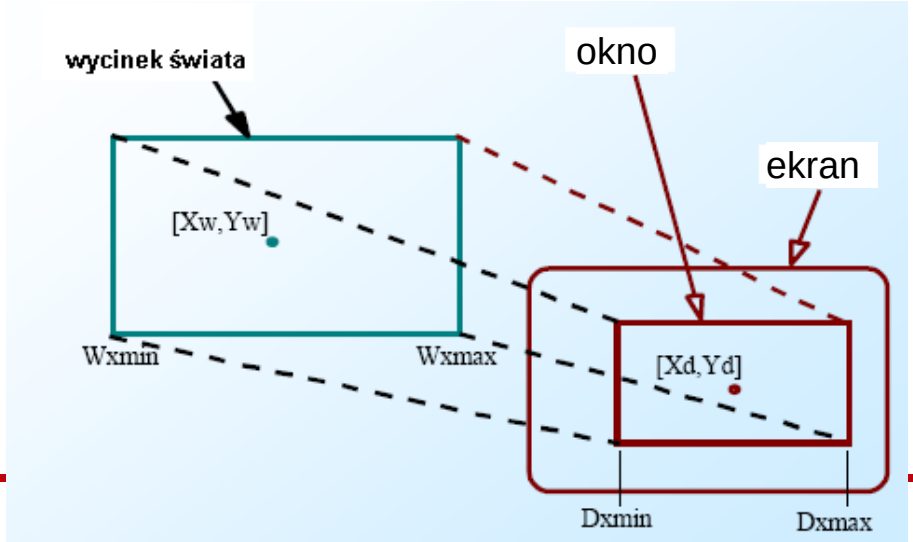
- okno do wyświetlania obrazu

monitor

okno do rysowania

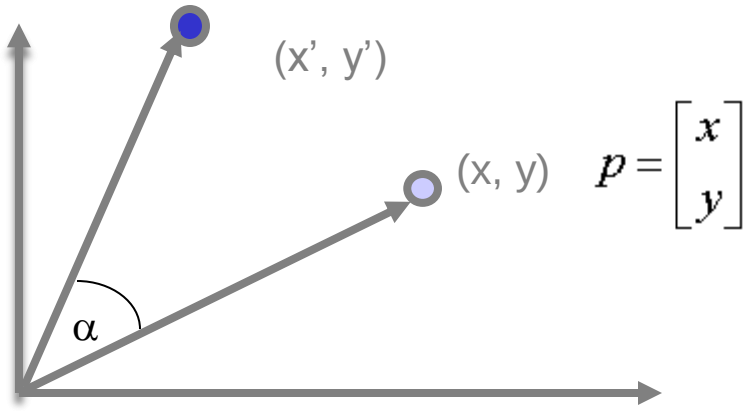


- normalizacja



Przekształcenia 2W

- Podstawowe elementy (prymitywy) to punkty, odcinki, wielokąty
- Relacje w budowaniu treści : piksele → prymitywy → obiekty
- Łatwość reprezentacji obiektów w 2W
- W przekształceniach wykorzystuje się działania ma macierzach – od pikseli do obiektów



$$p = \begin{bmatrix} x \\ y \end{bmatrix}$$

obracamy punkt końcowy odcinka o kąt α:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Przestrzeń afiniczna

- Modele obiektów, figury można tworzyć w różnych przestrzeniach, ale ostateczna reprezentacja (wymagana przez systemy wizualizacji) powstaje w afinicznej przestrzeni euklidesowej
- Definicja przestrzeni afinicznej:
  - mamy pewien zbiór *E* i przestrzeń liniową *V* (inaczej wektorową, z możliwością dodawania i skalowania)
  - mamy działanie odejmowania punktów, które parze punktów $p, q \in E$ przyporządkowuje pewien wektor $v = p - q \in V$, takie że
    - dla każdego $q \in E$ i $v \in V$ istnieje dokładnie jeden punkt $p \in E$ spełniający równanie $v = p - q$
    - dla dowolnych punktów $p, q, r \in E$ zachodzi tak zwana równość trójkąta: $r - p = (r - q) + (q - p)$
  - to zbiór *E* nazywa się przestrzenią afiniczną, a przestrzeń *V* jej przestrzenią wektorów swobodnych
  - wymiar przestrzeni afinicznej jest równy wymiarowi przestrzeni *V* , zaś związki pomiędzy punktami przestrzeni *E* i wektorami swobodnymi są takie:
    - $v = p - q \in V$ — różnica punktów jest wektorem,
    - $p = q + v \in E$ — suma punktu i wektora jest punktem
  - przestrzeń afiniczna jest rzeczywista, jeśli jej przestrzeń wektorów swobodnych jest przestrzenią liniową nad ciałem liczb rzeczywistych R
  - w przestrzeni rzeczywistej możemy określić iloczyn skalarny, a przestrzeń z iloczynem skalarnym nazywa się przestrzenią euklidesową

$$\forall_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in V,\, a,b\in\mathbb{R}} \langle a\boldsymbol{x} + b\boldsymbol{y}, \boldsymbol{z}\rangle \;=\; a\langle \boldsymbol{x}, \boldsymbol{z}\rangle + b\langle \boldsymbol{y}, \boldsymbol{z}\rangle$$

Przestrzenie graficzne

- Iloczyn skalarny daje możliwość liczenia odległości punktów (długości wektora łączącego te punkty):

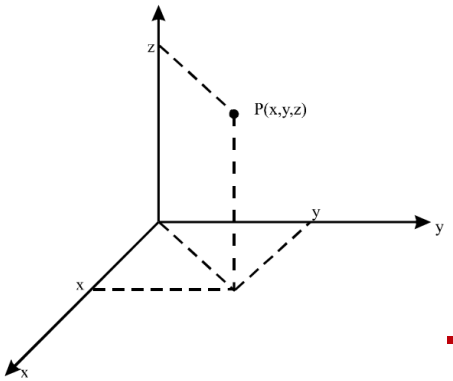
$$\rho(\boldsymbol{p},\boldsymbol{q}) = \sqrt{\langle \boldsymbol{p}-\boldsymbol{q}, \boldsymbol{p}-\boldsymbol{q}\rangle}$$

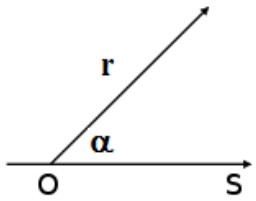
oraz kąt pomiędzy wektorami:

$$\cos\alpha = \left|\frac{\langle \boldsymbol{p}-\boldsymbol{q}, \boldsymbol{s}-\boldsymbol{t}\rangle}{\rho(\boldsymbol{p},\boldsymbol{q})\rho(\boldsymbol{s},\boldsymbol{t})}\right|$$

- W grafice wykorzystuje się rzeczywiste afiniczne przestrzenie euklidesowe dwu- i trójwymiarowe

- Wykorzystuje się zazwyczaj współrzędne kartezjańskie (prostokątne), biegunowe (odległość plus kąty w 2W) i sferyczne (odległość plus kąty w 3W)

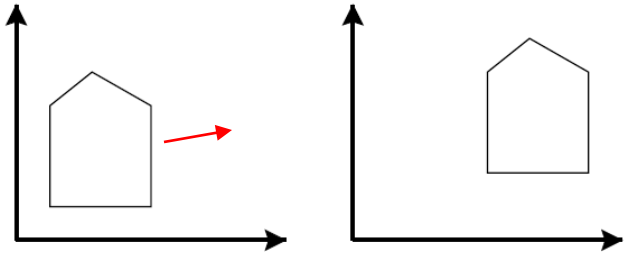




Zasadnicze przekształcenia afiniczne (przesunięcie, obrót, skalowanie)

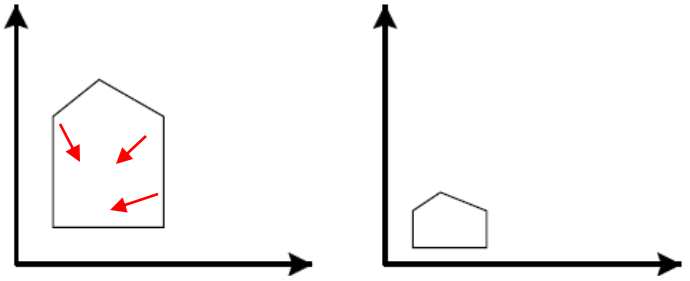
- Przesunięcie

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$



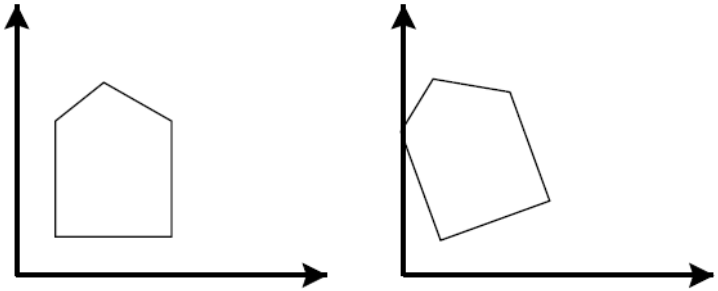
- Skalowanie

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x \cdot x \\ s_y \cdot y \end{bmatrix}$$



- Obrót

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

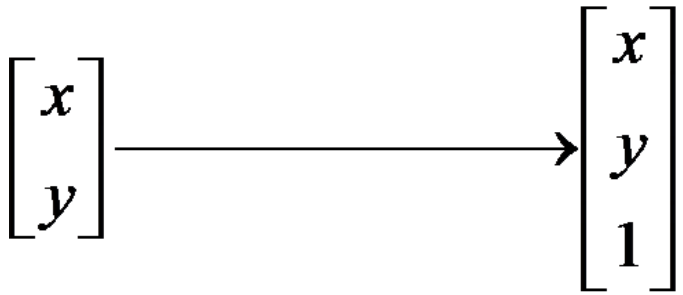


Cechy przekształceń afinicznych

- Własności figur, zachowywane przez przekształcenia afiniczne (niezmienniki afiniczne) to:
  - współliniowość i współpłaszczyznowość punktów
  - wypukłość figury
  - proporcje odległości punktów współliniowych
  - bycie trójkątem
  - bycie równoległobokiem
  - bycie elipsą
  - obrazem odcinka jest odcinek lub punkt
  - obrazem prostych równoległych są punkty albo proste równoległe

Przekształcenia jednorodne (homogeniczne), znormalizowane

- Dwuwymiarowe współrzędne za pomocą trójelementowych wektorów



Nieco zagadkowe, ale zrozumiałe i przede wszystkim korzystne

- Ujednolicony sposób opisu wszystkich przekształceń na płaszczyźnie i w 3D

Przekształcenia afiniczne

- We współrzędnych jednorodnych znormalizowanych mamy:
  - Translacja

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

  - Skalowanie

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

  - Obrót

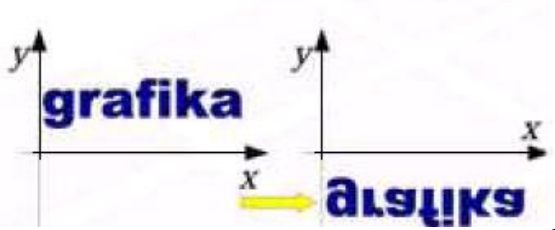
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

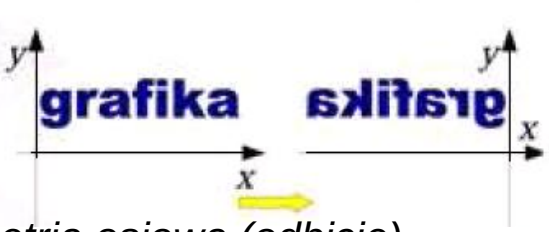
Pochodne przekształcenia afiniczne

$$M_{SOX}=\begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

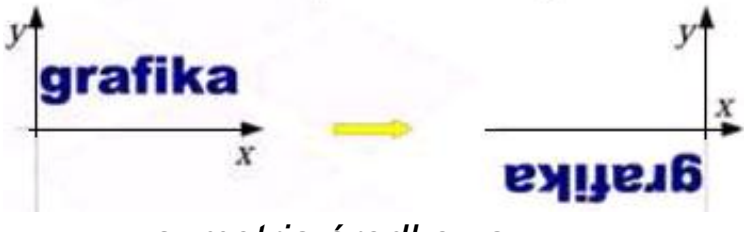
$$M_{SOY}=\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$M_{SS}=\begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}$$



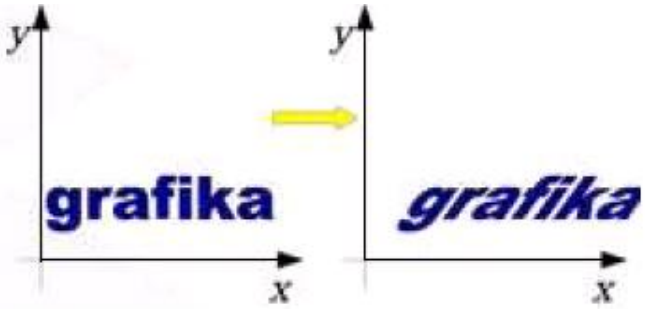


symetria osiowa (odbicie)



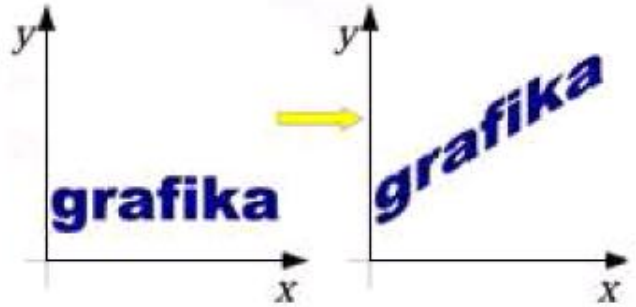
symetria środkowa

$$M_{Px}=\begin{bmatrix}1 & H_x & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

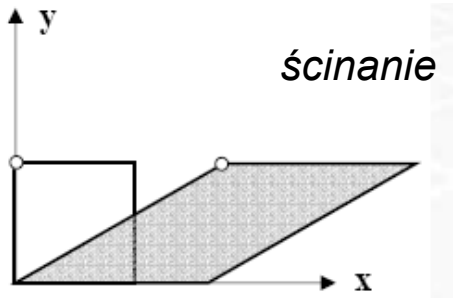


$$\begin{bmatrix}x'\\ y'\\ 1\end{bmatrix}=\begin{bmatrix}1 & h_x & 0\\ h_y & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ 1\end{bmatrix}$$

$$M_{Py}=\begin{bmatrix}1 & 0 & 0\\ H_y & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$



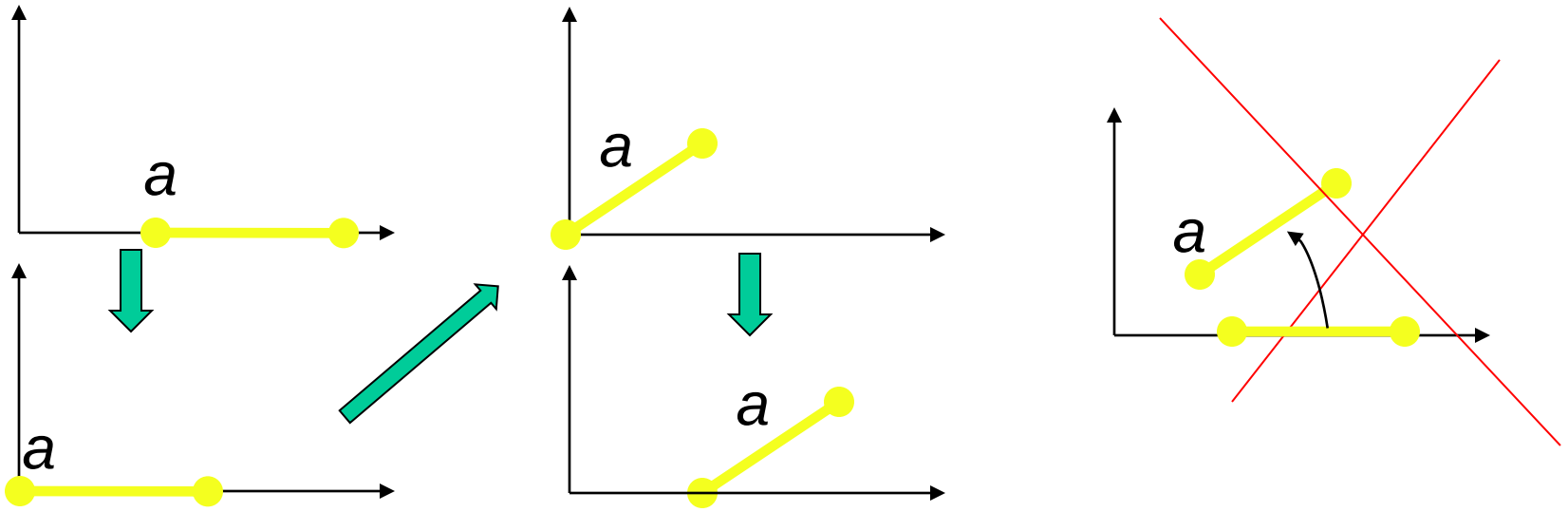
pochylenie



ścinanie

Rodzaje przekształceń jednorodnych

- Afiniczne zachowują równoległość linii
- Sztywne zachowują długości i kąty (np. translacja i obrót)
- Skalowanie jest nieizometryczne (nie zachowuje długości)
- Kolejność wykonywania przekształceń (przesunięcie do początku układu współrzędnych, skalowanie - obrót, przesunięcie w odpowiednie miejsce)



Istotne zasady przekształcania -cd

- Proste składanie transformacji

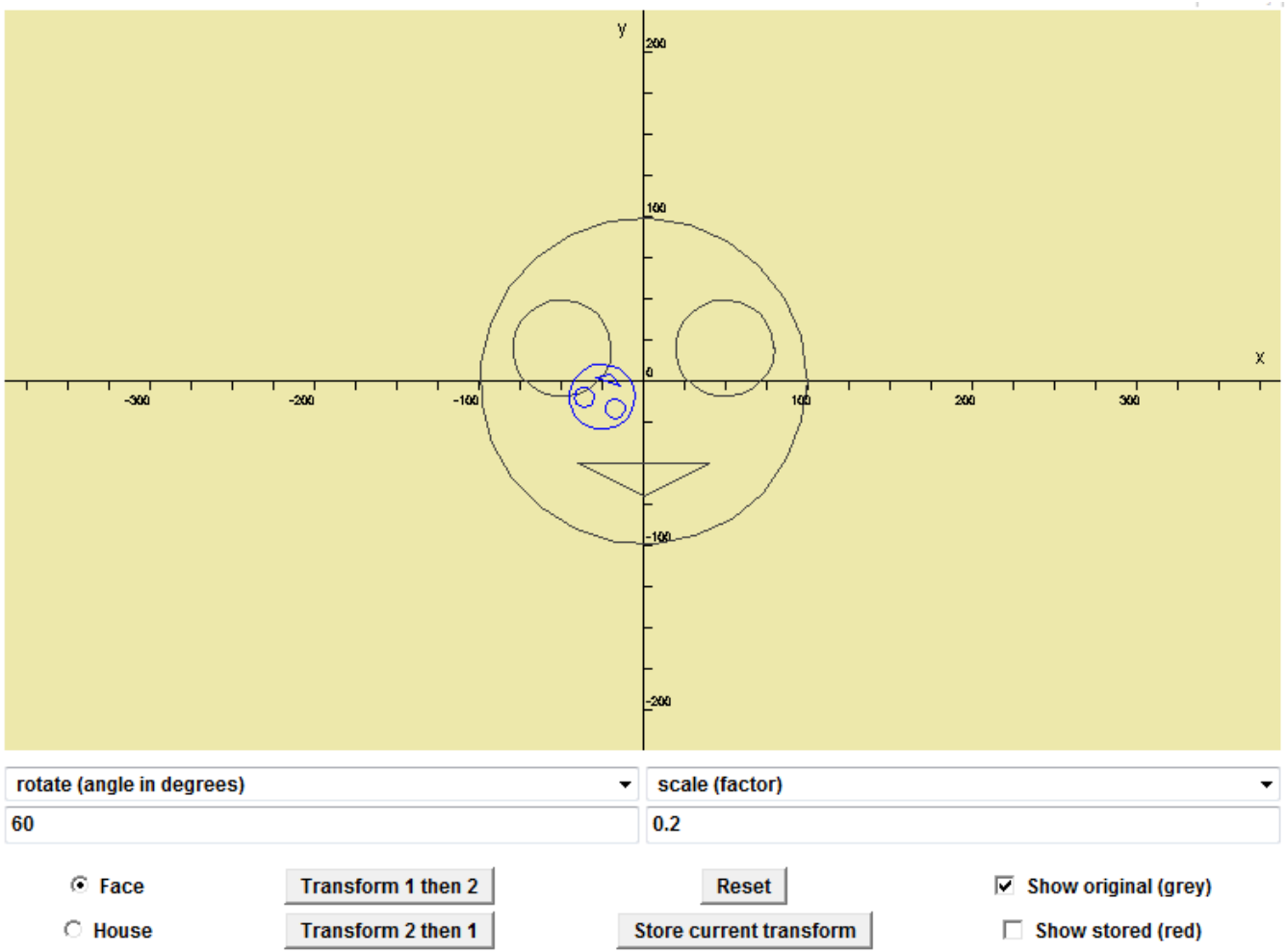
$$T(\boldsymbol{p})R(\theta)T(-\boldsymbol{p}) = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -p_x \\ 0 & 1 & -p_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & p_x(1-\cos\theta) + p_y\sin\theta \\ \sin\theta & \cos\theta & p_y(1-\cos\theta) - p_x\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

- Transformacje odwrotne

$$T^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad S^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Zabawa z apletami



http://www.cs.princeton.edu/~min/cs426/applets.html

A.Cohena-Sutherlanda