GRAKO: PODSTAWY GRAFIKI 2W

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

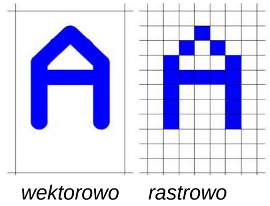
Niektóre wykorzystane tutaj materiały pochodzą z:

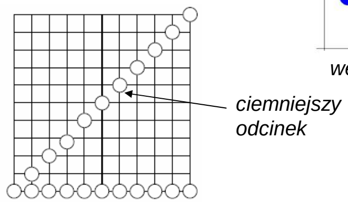
http://wazniak.mimuw.edu.pl/index.php?title=Grafika_komputerowa_i_wizualizacja

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

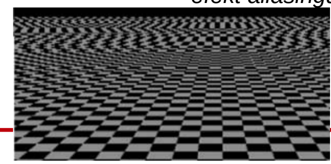
Grafika rastrowa -ograniczenia

- Położenie rysowanych figur (obiektów) jest ograniczone zwykle prostokątną strukturą pikseli
- Ograniczenia:
  - Brak skalowania
  - Linia schodkowa (brak gładkości – dyskretyzacja kształtu)
  - Aliasing
  - Złożoność
  - Zmiana barwy





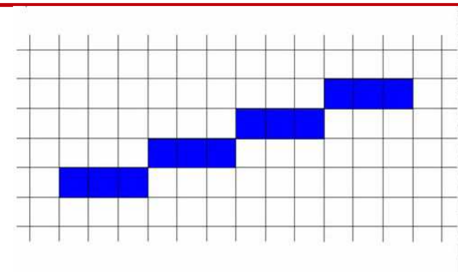
efekt aliasingu

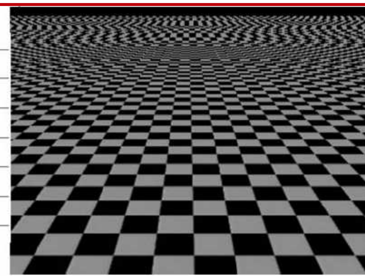


Grafika rastrowa

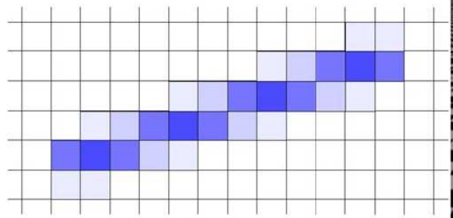
- Metody poprawy:
  - Projektowanie (myślenie) wektorowe
  - Zwiększanie rozdzielczości matrycy pikselowej
  - Rozmywanie obiektów o wysokogradientowych krawędziach, rozjaśnianie ukośnych odcinków
  - Ważenie jasności (piksel świeci proporcjonalnie rozmiaru pokrycia przez dany obiekt)
  - Przyspieszanie obliczeń

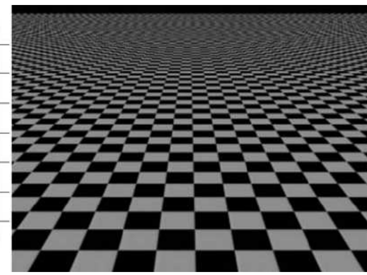
Antyaliasing



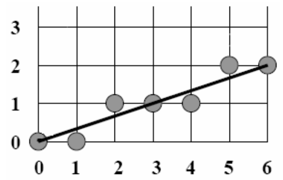


Rozmycie ostrych krawędzi poprawia widziany efekt





Rysowanie odcinka – złożoność obliczeniowa



y=ax

- $y=[a\cdot x]$

- mnożenie rzeczywistoliczbowe plus zaokrąglanie

lub

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

- dodawanie rzeczywistoliczbowe plus zaokrąglanie

Rysowanie odcinka – algorytm Bresenhama

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

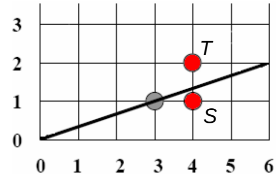
    $d_0=2\cdot(y_2-y_1)-(x_2-x_1)$

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

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

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

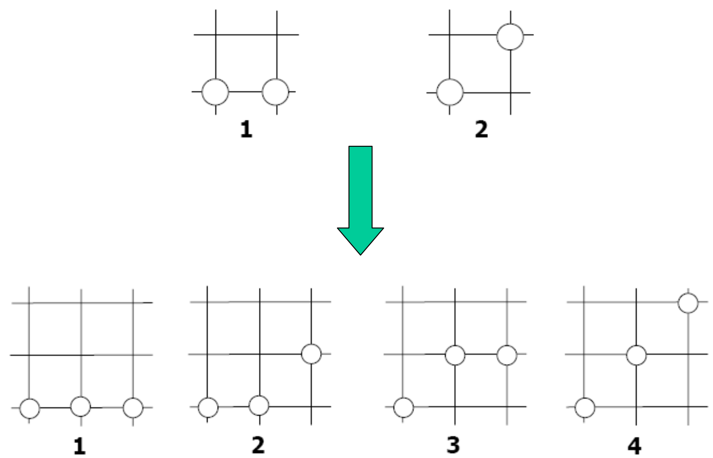
    w p.p. $(x_{i+1},y_{i+1})=T$; $d_{i+1}=d_i+p_2$;



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

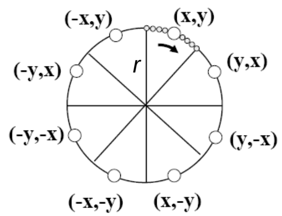
złożoność: dodawanie całkowitoliczbowe plus określanie znaku

Modyfikacja algorytmu Bresenhama



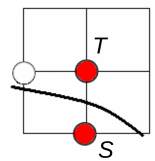
Algorytm z podwójnym krokiem

Rysowanie okręgu – algorytm Bresenhama



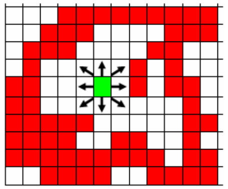
- Stałoprzecinkowy
- Założenia: zaczynamy od punktu $(0,r)$ i rysujemy 1/8 okręgu (resztę przez symetrię)
- Metoda
  - $p_1=3-2r$
  - w kolejnym kroku:

    jeśli $(p_i<0)$ to $(x_{i+1},y_{i+1})=T$; $p_{i+1}=pi+4x_{i-1}+6$;
  - 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

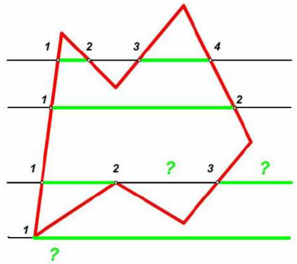
- Przez spójność (rozrost regionu)



- Przez kontrolę parzystości

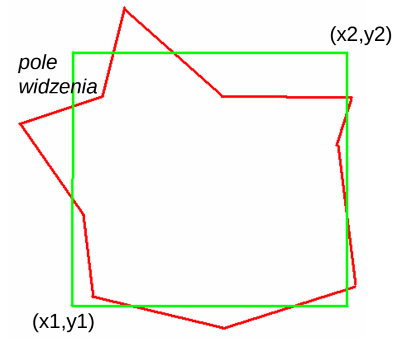
parzyste – wejście do obiektu

nieparzyste – wyjście z obiektu



przypadki szczególne – ekstrema z analizą końców odcinków, odcinki poziome

Obcinanie obiektu do widocznego fragmentu

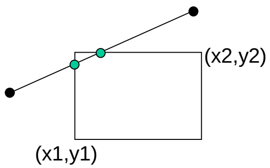


$$x1 < x < x2 \quad \wedge \quad y1 < y < y2$$

Konwersja do bufora wyświetlania i usunięcie wszystkich pikseli nie mieszczących się z widoku

Obcinanie odcinków

- Sprawdzanie końców odcinków:
  - oba wewnątrz widoku: odcinek zostawiamy
  - jeden na zewnątrz: przycinamy do okna widoku
  - oba końce na zewnątrz - rozstrzyganie: odrzucamy odcinek czy przycinamy?
    - obliczanie punktów przecięcia z widokiem i sprawdzanie, czy należą do widoku



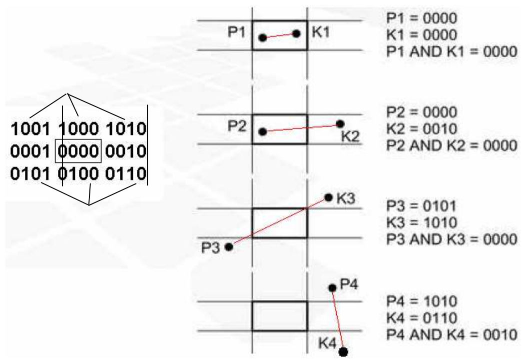
$$x1 < x < x2 \quad \wedge \quad y1 < y < y2$$

Obcinanie odcinka (przyspieszenie)

<u>algorytm Cohena-Sutherlanda</u>

-dzielimy na regiony z kodem binarnym

-końcom odcinka przypisujemy kody regionów

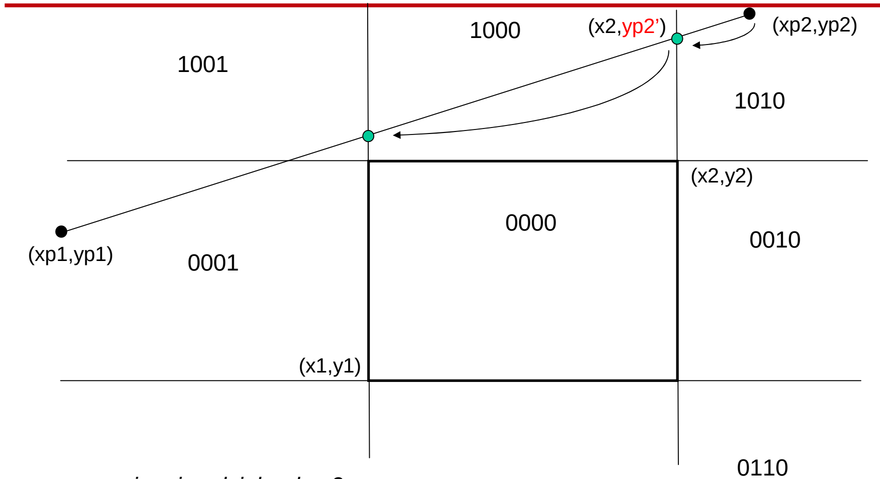


- Sprawdzanie końców odcinków:
  - oba wewnątrz widoku: odcinek zostawiamy
    - oba końce 0000
  - jeden na zewnątrz - AND końców daje 0: przycinamy do okna widoku
  - oba końce na zewnątrz - rozstrzyganie: odrzucamy odcinek czy przycinamy?
    - AND końców daje wynik niezerowy – odrzucamy odcinek zewnętrzny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

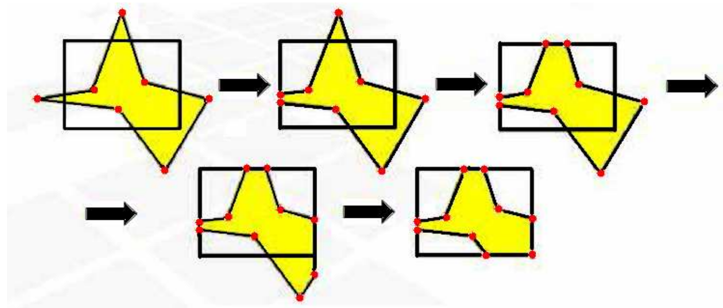
Algorytm Cohena-Sutherlanda (wyjaśnienie)



przycinanie odcinka do x2

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

Obcinanie wielokąta

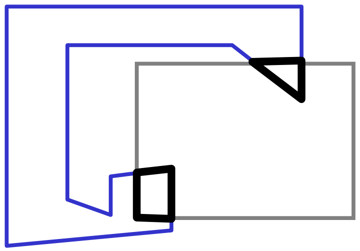


Algorytm Sutherlanda- Hodgmana

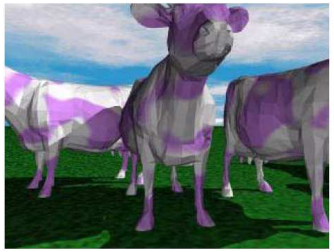
Rysunek z wazniaka

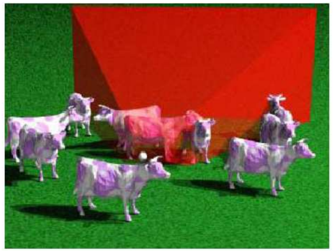
Obcinanie nie jest łatwe

- Z jednego wielokąta – kilka



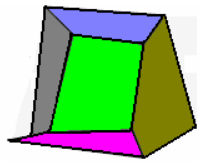
- W przestrzeni 3W

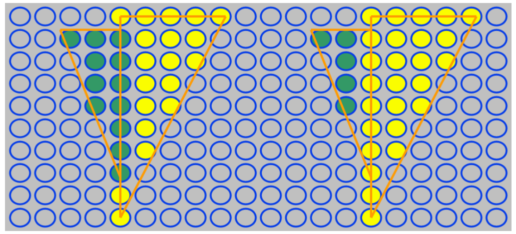




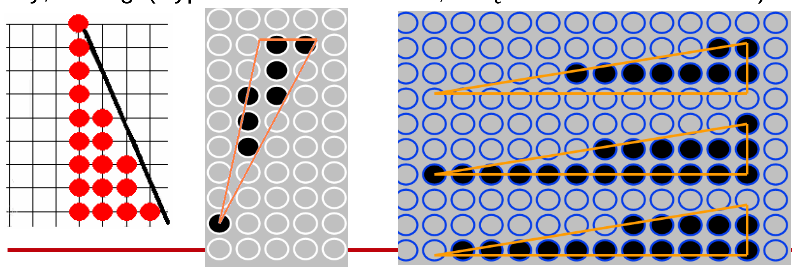
Inne problemy

- Wypełnianie brzegów wielokątów (reguła przynależności brzegu)



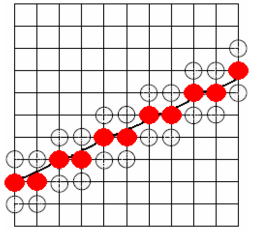


- Kliny, drzazgi (wypełnianie wielotonowe, zwiększanie rozdzielczości)



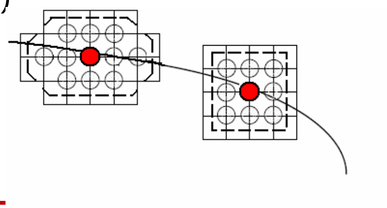
Pogrubianie linii

- Powielanie i wprowadzanie tonów pośrednich



w kolumnie lub wierszu zależnie od nachylenia a

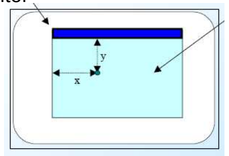
- Dynamiczny pędzel (pióro)



Przekształcenia 2W

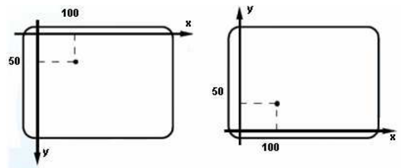
- okno do wyświetlania obrazu

monitor

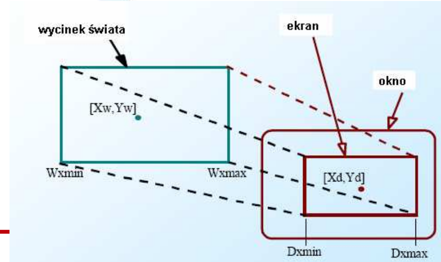


okno do rysowania

- konwencje adresowania pikseli



- normalizacja



Przekształcenia 2W

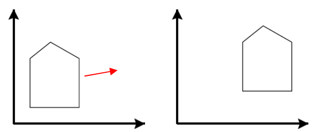
- Podstawowe elementy (prymitywy) to punkty, odcinki, wielokąty
- Relacja: piksele $\rightarrow$ prymitywy $\rightarrow$ obiekty
- Łatwość reprezentacji obiektów w 2W
- W przekształceniach wykorzystuje się działania ma macierzach

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

Przekształcenia afiniczne (przesunięcie, obrót, skalowanie)

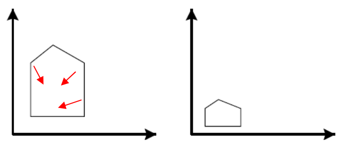
- Przesunięcie

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



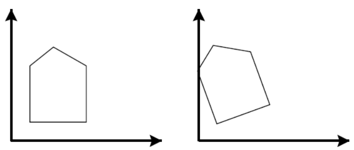
- Skalowanie

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



- Obrót

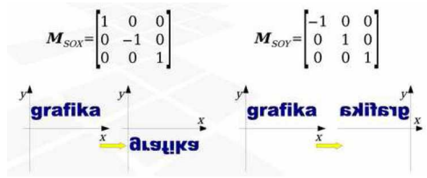
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



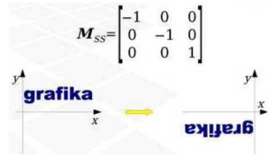
Przekształcenia afiniczne

- We współrzędnych jednorodnych (homogenicznych) 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
    $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

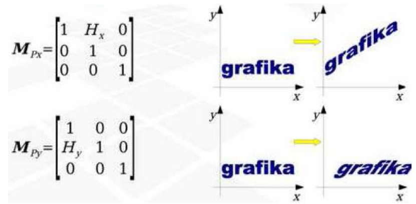
Inne przekształcenia 2W



symetria osiowa (odbicie)

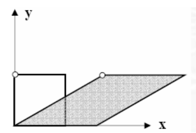


symetria środkowa



pochylenie

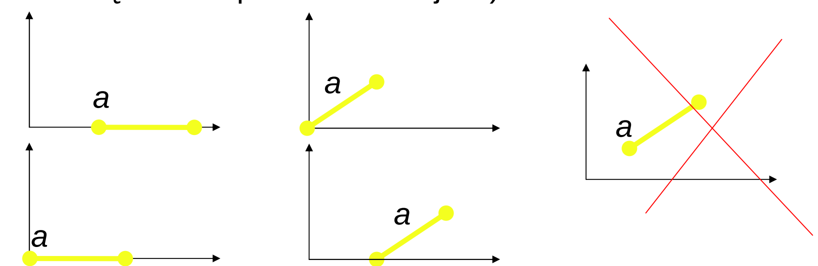
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



ścinanie

Przekształcenia jednorodne, znormalizowane

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



Istotne zasady przekształcania -cd

- Proste składanie transformacji

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

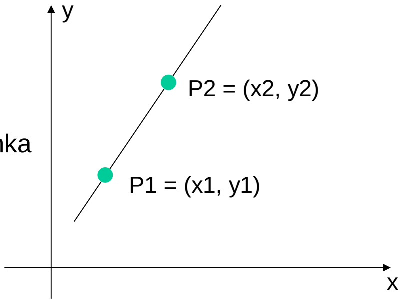
$$= \begin{bmatrix} \cos\theta & -\sin\theta & p_x(1-\cos\theta) + p_y\sin\theta \\ \sin\theta & \cos\theta & p_y(1-\cos\theta) - p_x\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

- Transformacje odwrotne

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

Równanie parametryczne prostej

- Dane: P1 = (x1, y1) and P2 = (x2, y2)
  x = x1 + t(x2 - x1)
  y = y1 + t(y2 - y1)
- Kładąc:
  - t=0, mamy (x1, y1)
  - t=1, mamy (x2, y2)
  - (0<t<1), dostajemy punkty odcinka pomiędzy (x1, y1) i (x2, y2)



- Analogicznie w 3W:
  x = x1 + t(x2 - x1)
  y = y1 + t(y2 - y1)
  z = z1 + t(z2 - z1)