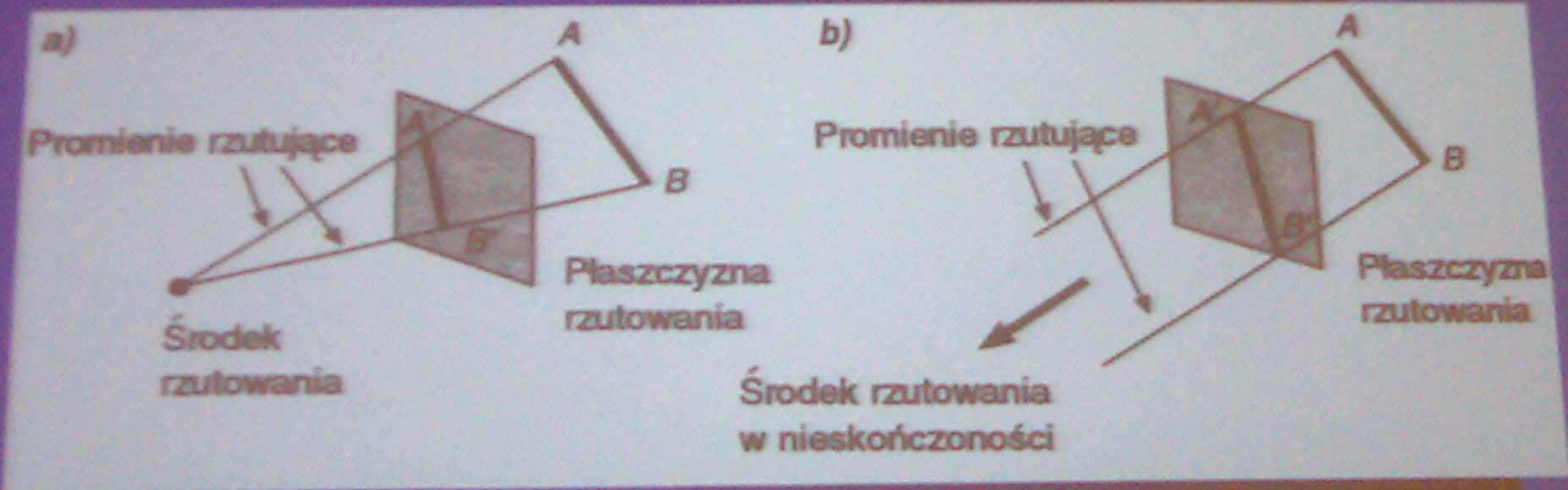
wyklady_do_kolos2

Rzutowanie

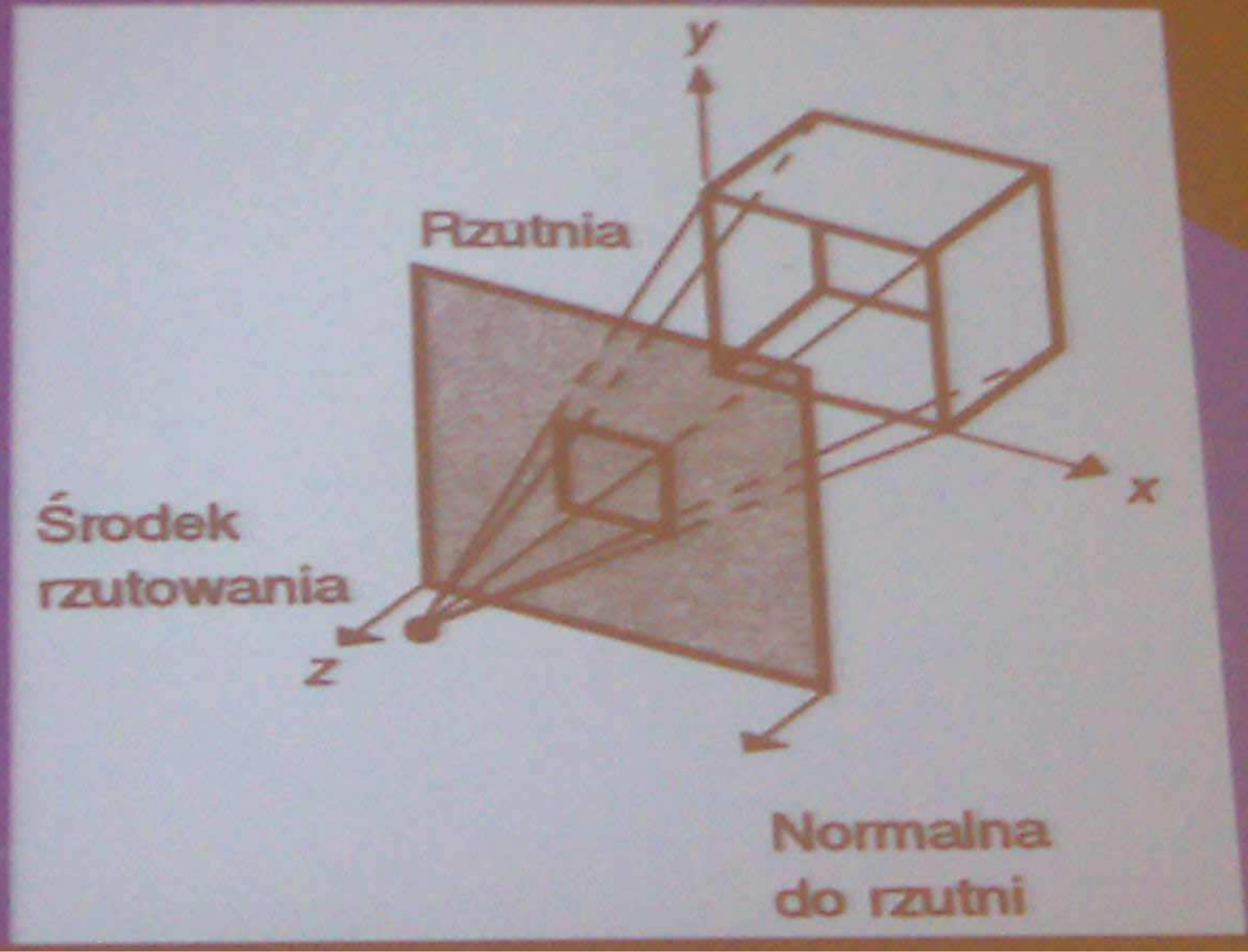
Rzutowanie to przekształcenia punktów z n-wymiarowej przestrzeni, do przestrzeni o wymiarze mniejszym niż n

- 3D → 2D
- Rzutowanie planarne
- Rzut równoległe
- Rzuty perspektywiczne



Proces rzutowania

- Rodzaj rzutu
- Parametry rzutu
  - Środek rzutowania
  - Rzutnia
- Obcinanie 3D
- Rzutowanie
- Kamera
  - Parametry kamer
- Bryła widzenia

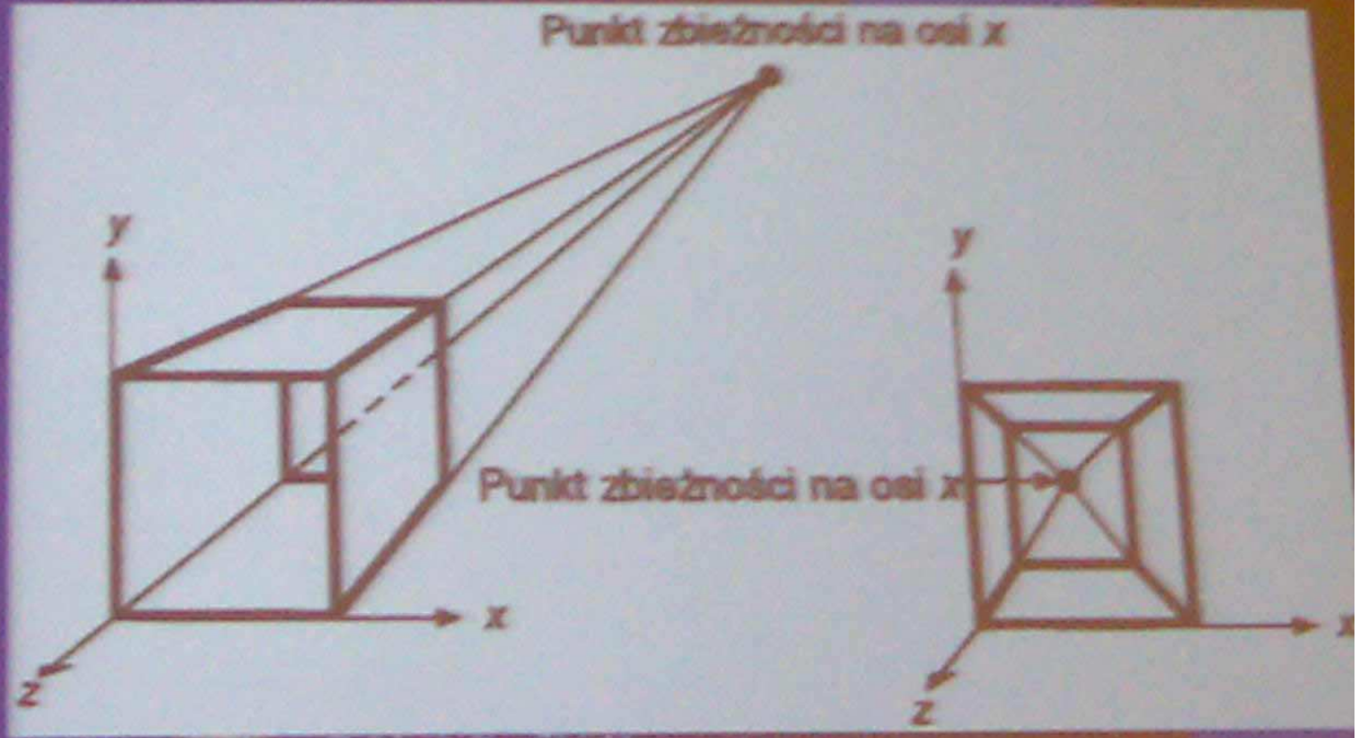


Rzuty perspektywiczne

- *Punkt zbieżności* – miejsce w którym zbiegają się rzuty zbioru linii równoległych
- Osiowe punkty zbieżności

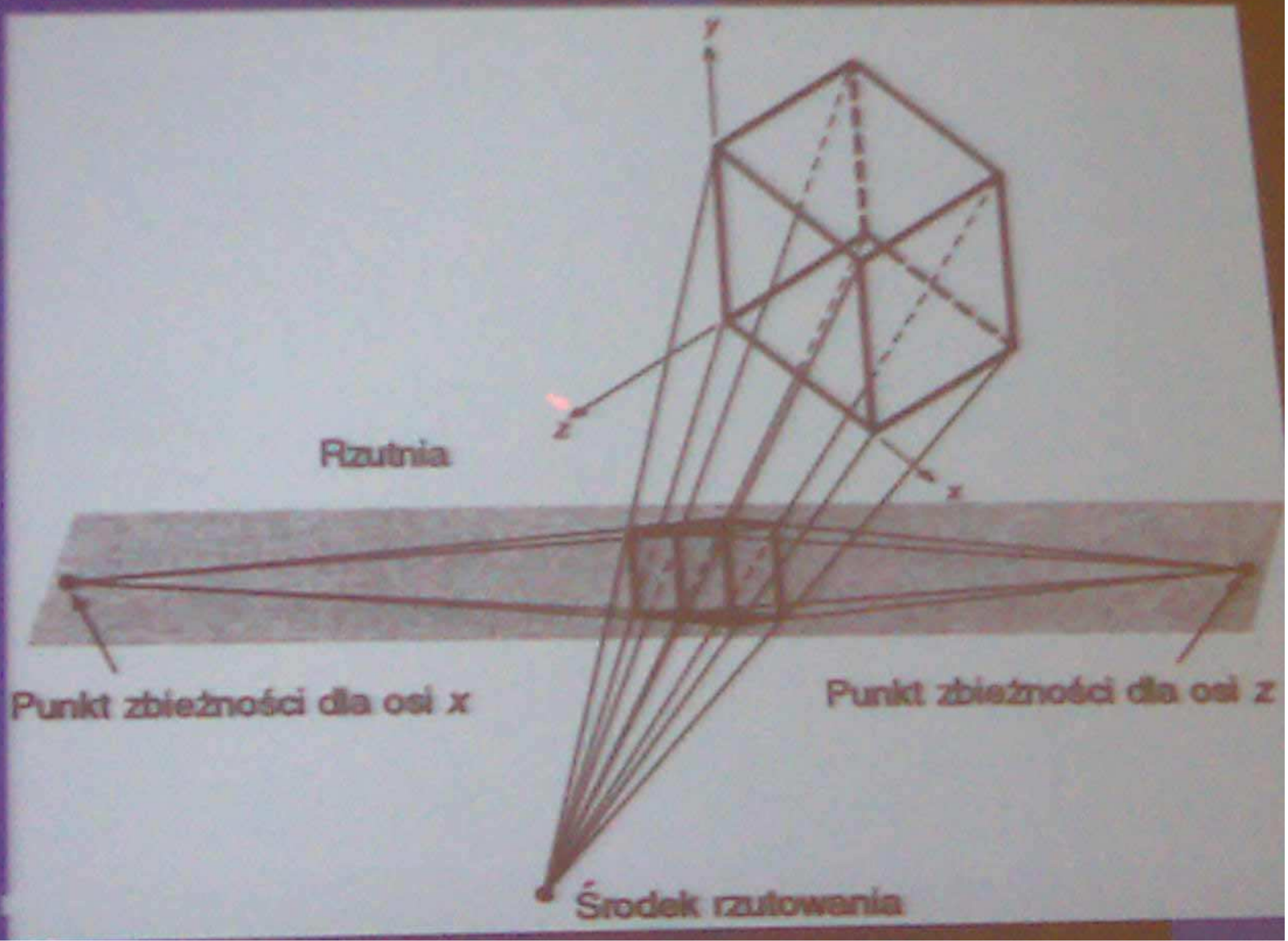
Jednopunktowy rzut perspektywiczny

- Punkt zbieżności linii równoległych do osi z
- Punkt zbieżności może być traktowany jak widok linii równoległej do osi z biegnącej od oka obserwatora

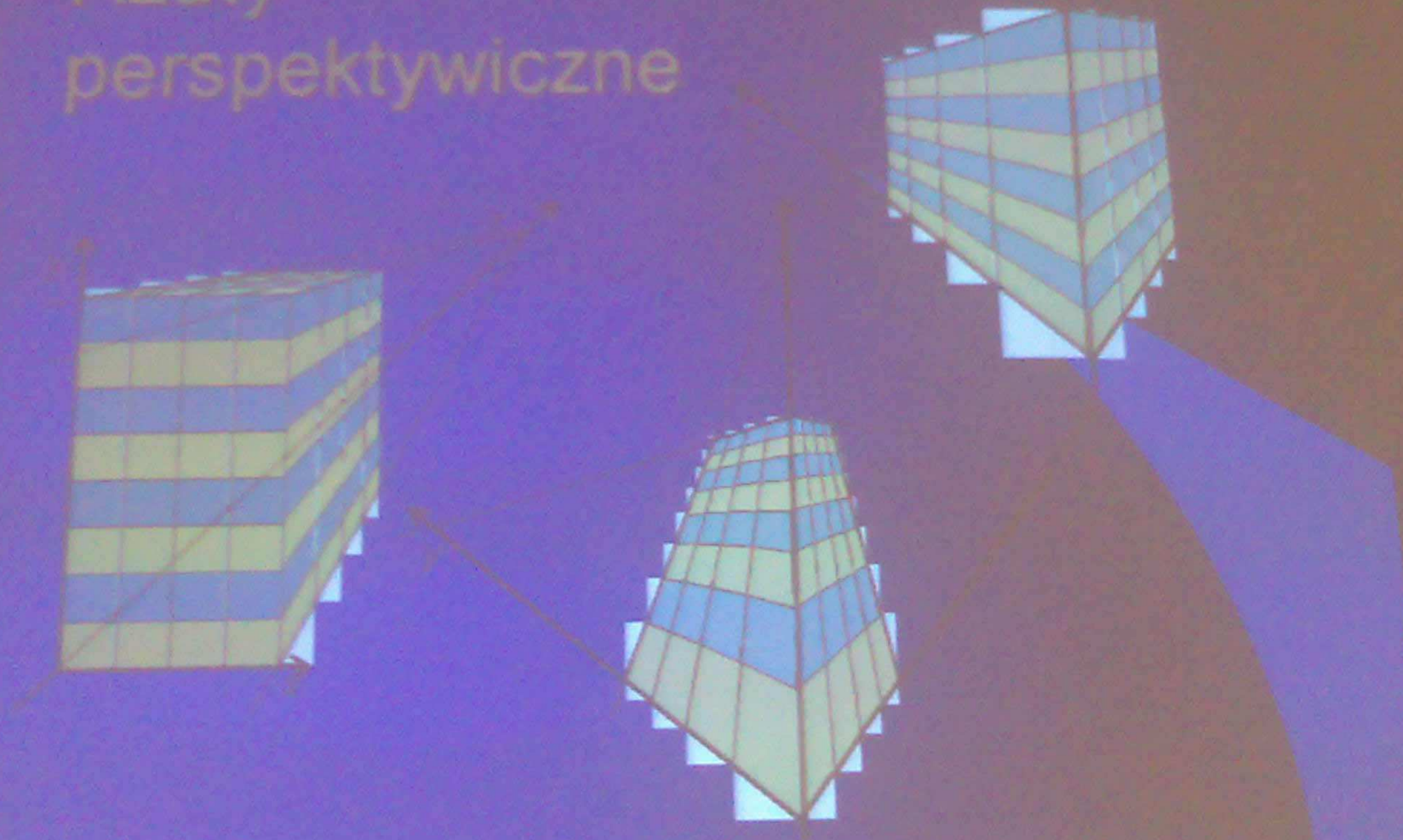


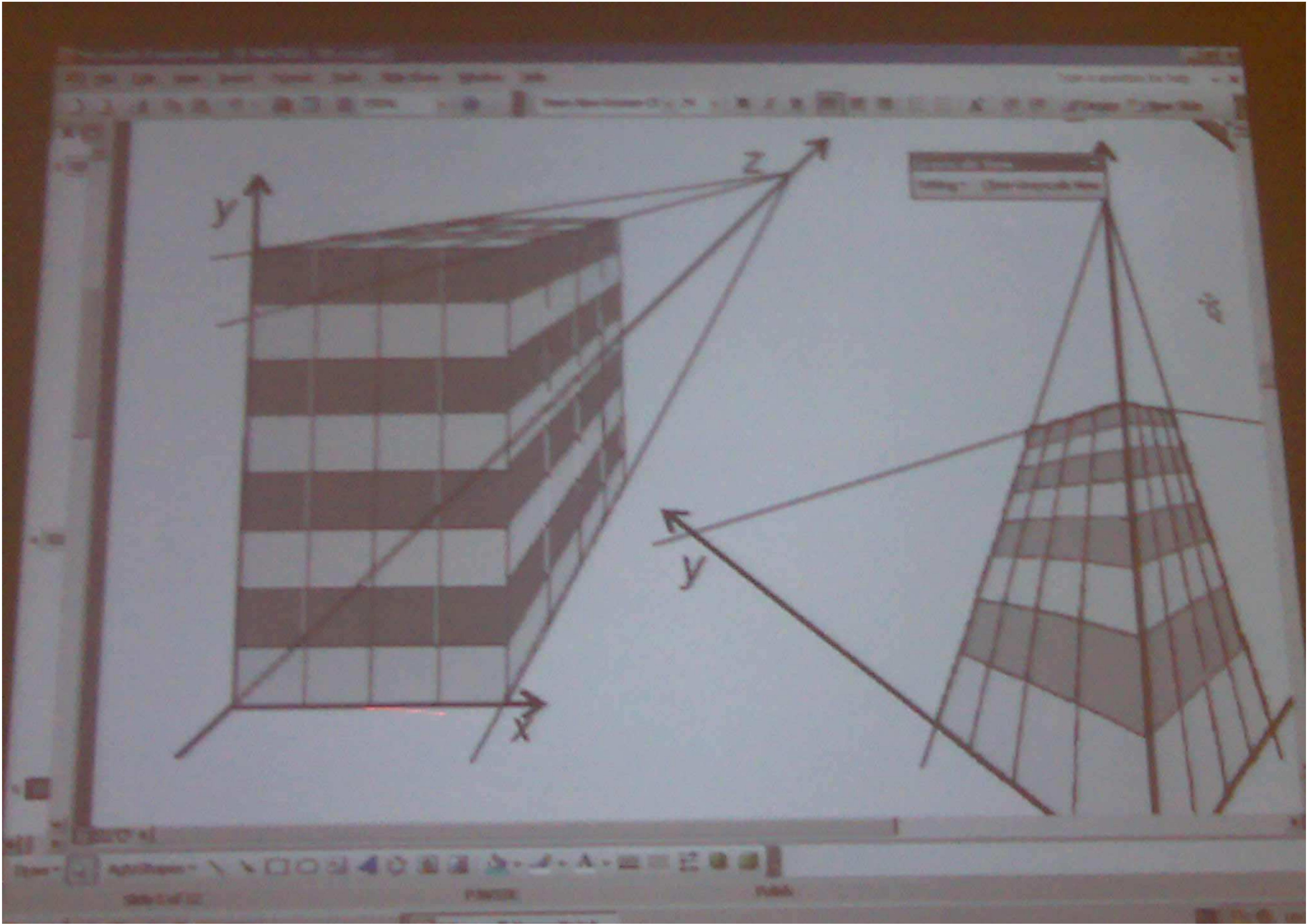
Rzuty perspektywiczne dwupunktowe

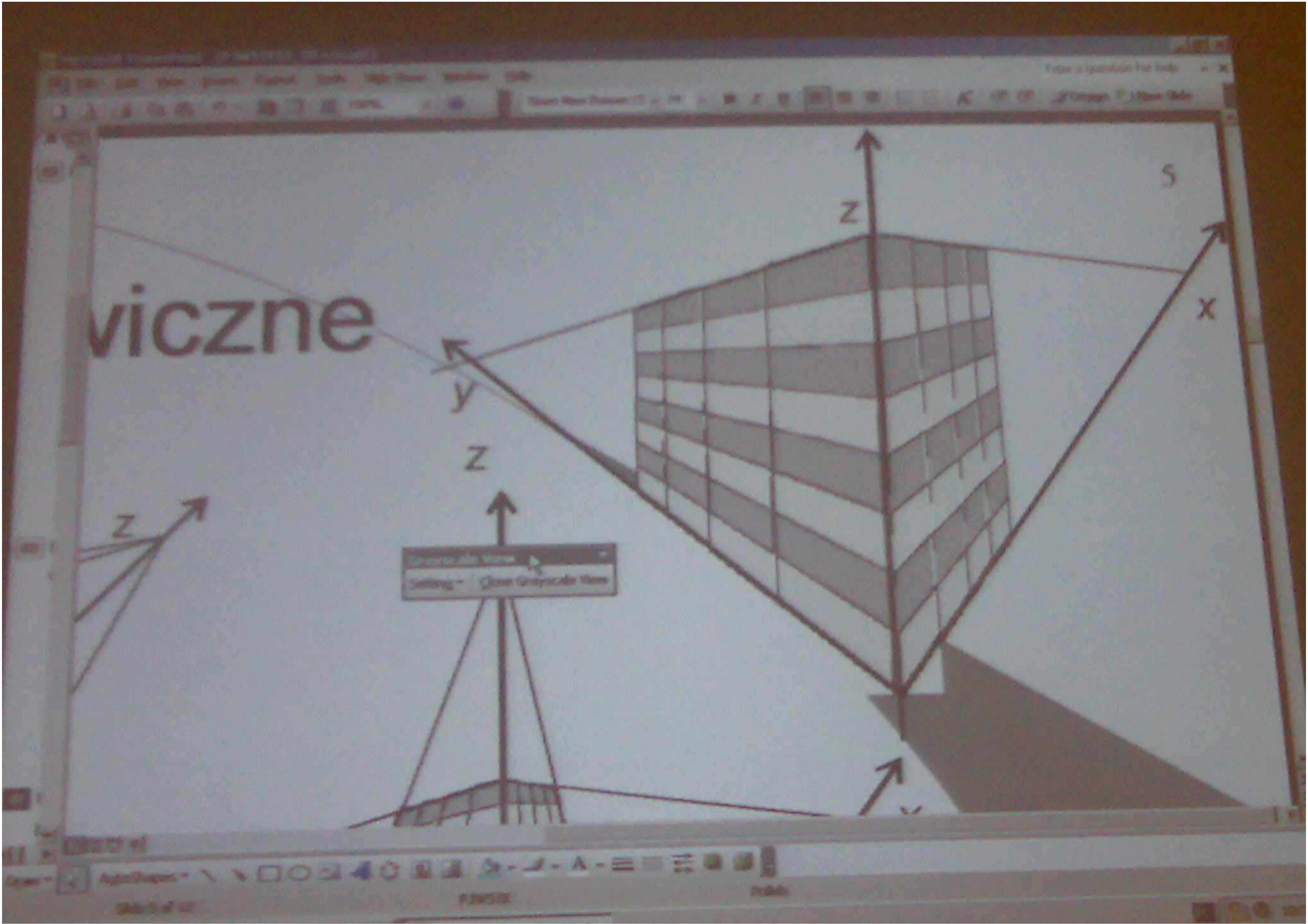
- Na linii horyzontu znajdują się dwa punkty zbieżności dla prostych równoległych do osi x i osi z
- Oś pionowa. bez punktu zbie-żności

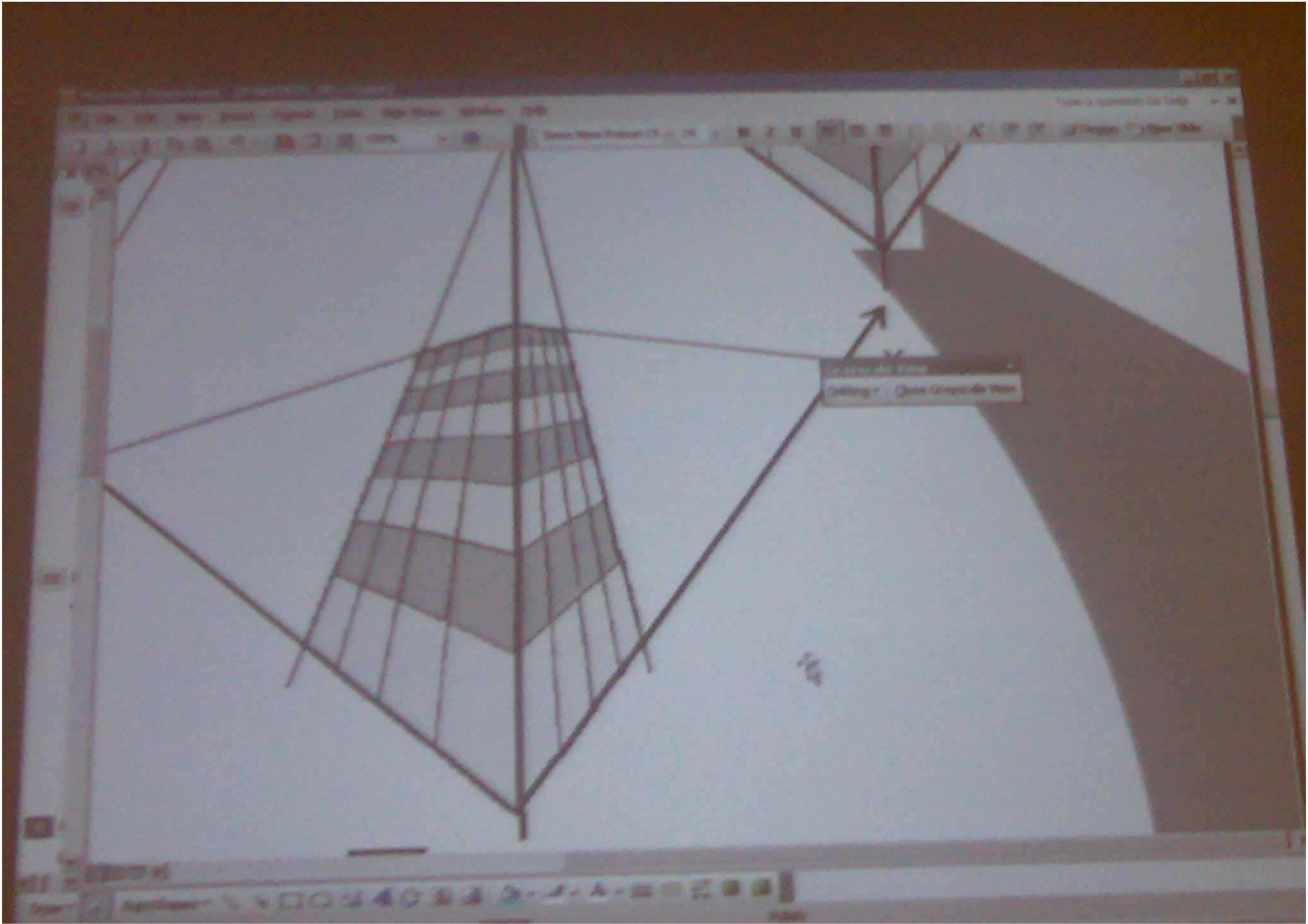


Rzuty perspektywiczne



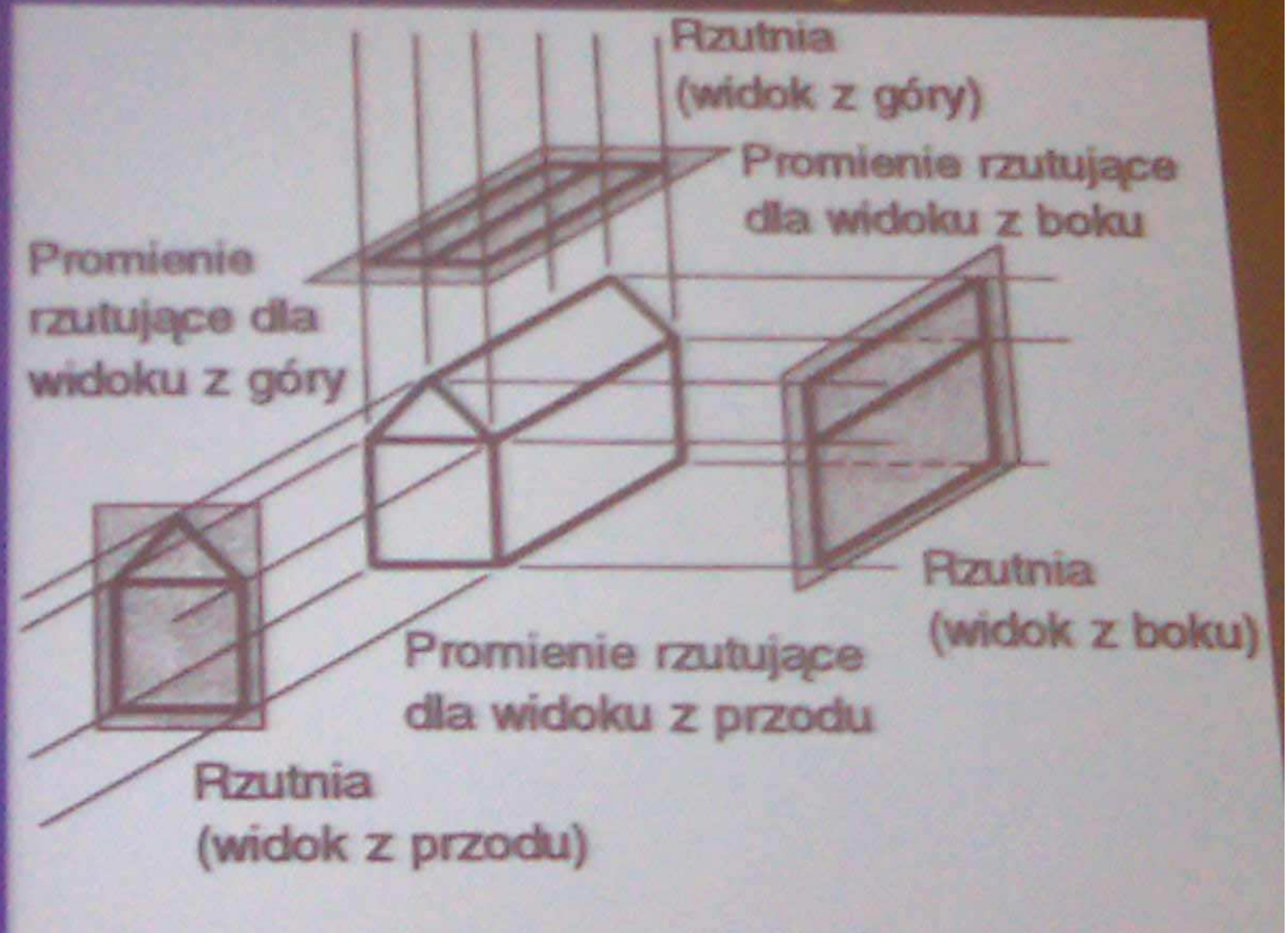

y
z
x
y


5
z
viczne
x
y
z
z



Rzuty równoległe prostokątne

Rzuty równoległe
- Prostokątne
  - Przedni
  - Górny
  - Boczny
  - Aksonometryczny
- Ukośne



Konstrukcja trzech rzutów prostokątnych

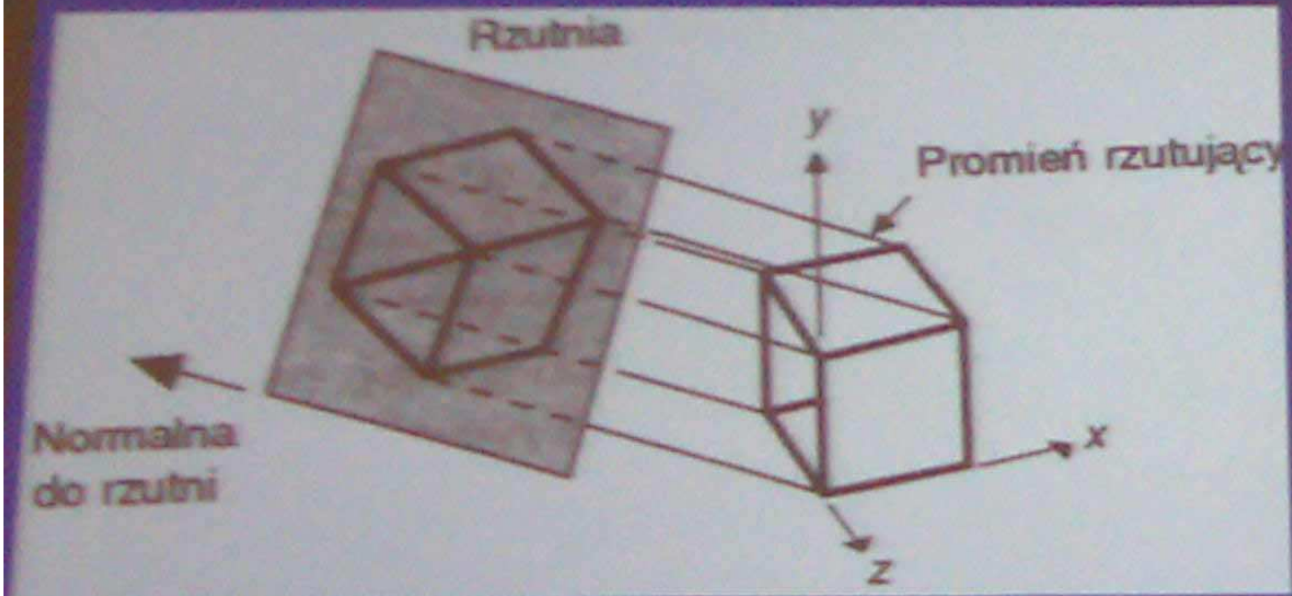
Zachowuje
- równoległość prostych
- Stosunek długości odcinków równoległych

Rzuty równoległe (2)

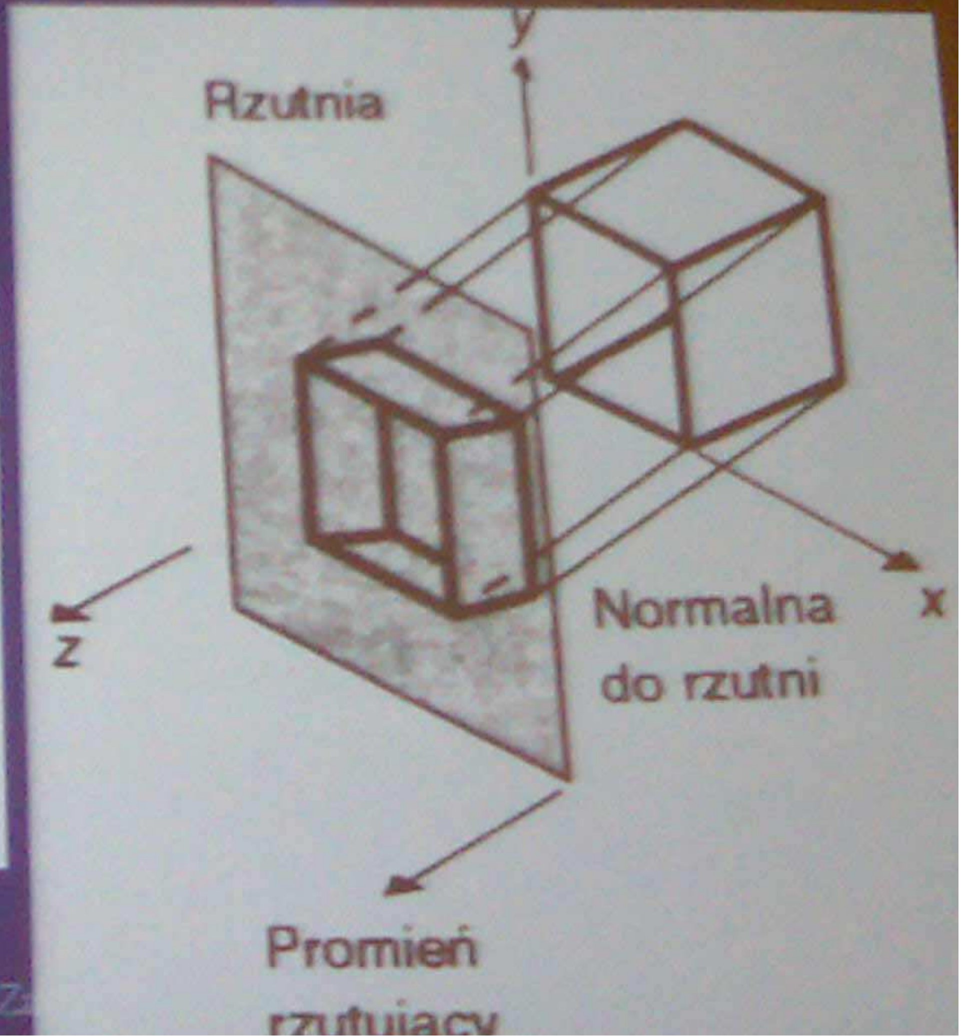
Rzuty prostokątny aksometryczny
(rzutnia nie jest prostopadła do osi głównej)

- Rzut izometryczny
  - normalna do rzutni tworzy identyczne kąty z osiami układu współrzędnych

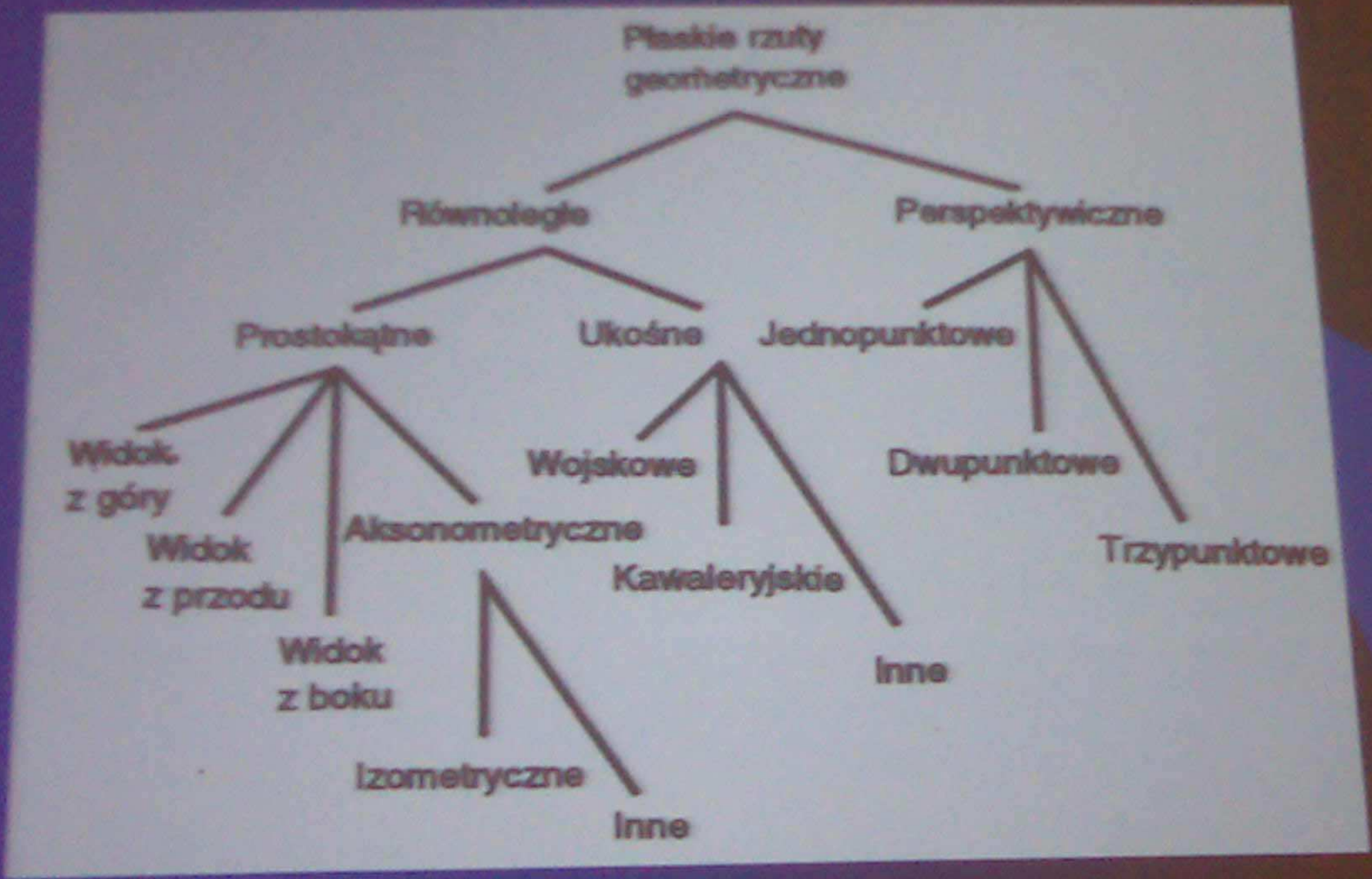
Rzut ukośny

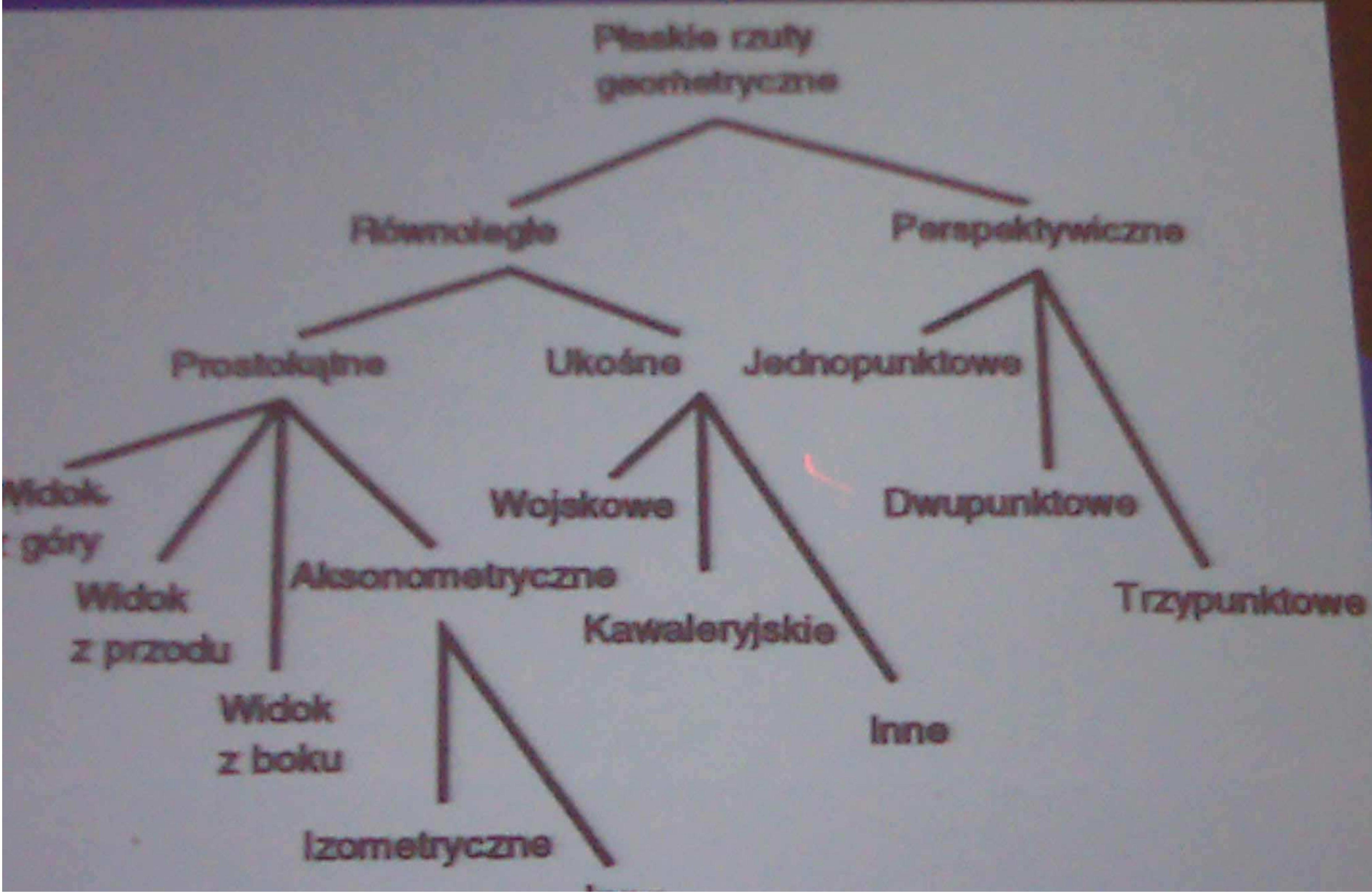


Konstrukcja rzutu izometrycznego dla sześcianu jednostkowego



Klasyfikacja rzutów



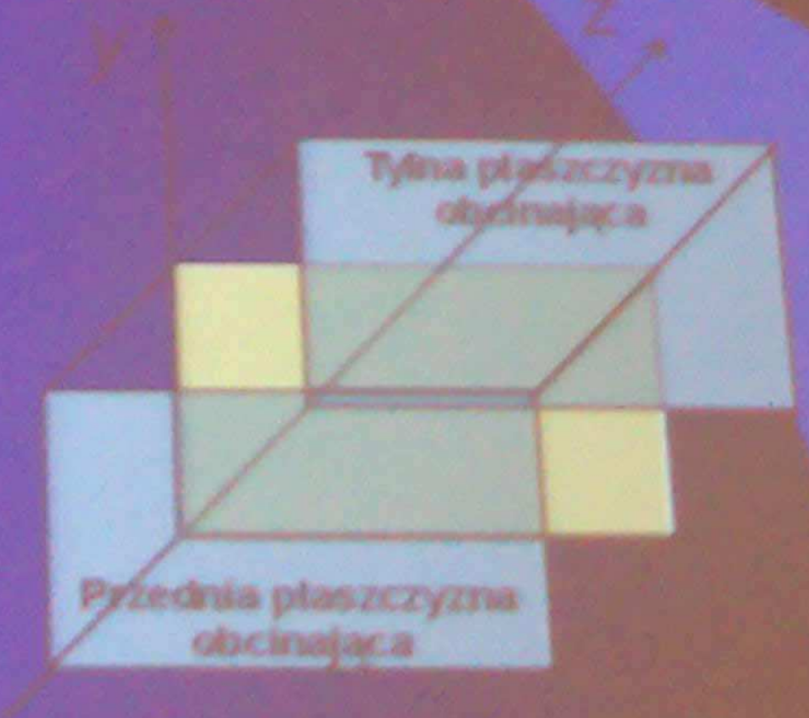

Płaskie rzuty geometryczne
Równoległe
Perspektywiczne
Prostokątne
Ukośne
Jednopunktowe
Widok z góry
Widok z przodu
Widok z boku
Aksonometryczne
Izometryczne
Wojskowe
Kawaleryjskie
Inne
Dwupunktowe
Trzypunktowe

Bryła widzenia

Stożek widzenia, piramida widzenia

- Przednia płaszczyzna obcinania
  - Płaszczyzna rzutowania
  - Ekran (obcinanie góra, dół, lewo, prawo)
- Tylna płaszyczyzna obcinania

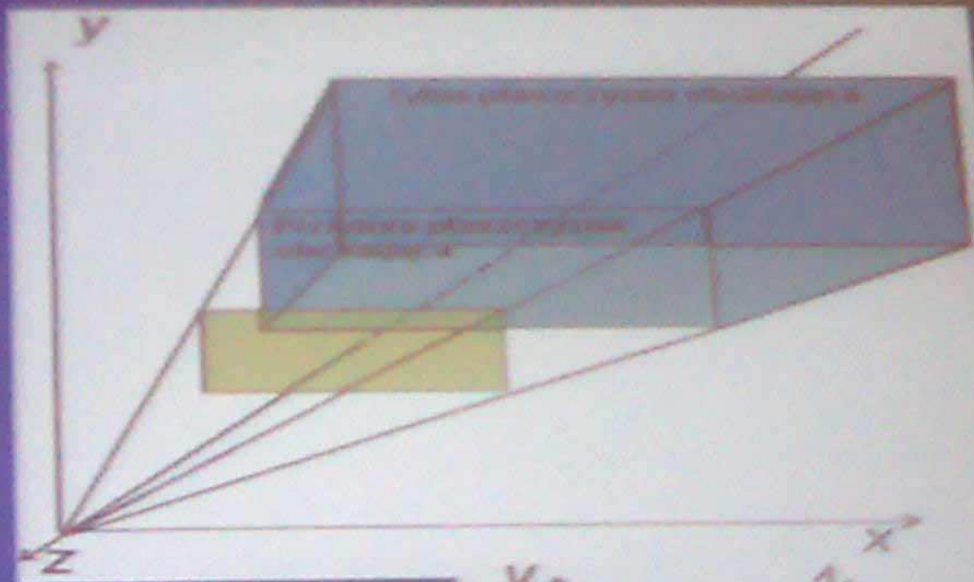
W rzucie równoległym bryłą widzenia jest równoległobok



Bryła widzenia dla rzutu perspektywicznego

Punkt zbieżności (oko obserwatora)

Obcinanie przez bryłę widzenia



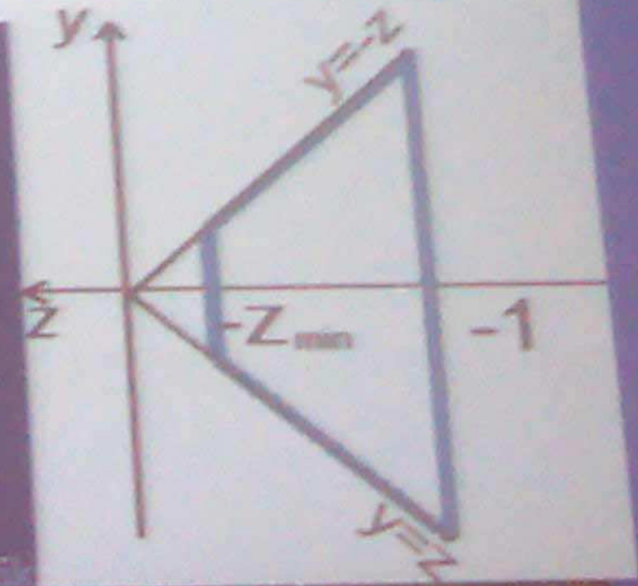
Kanoniczne bryły widzenia

- Rzut równoległy

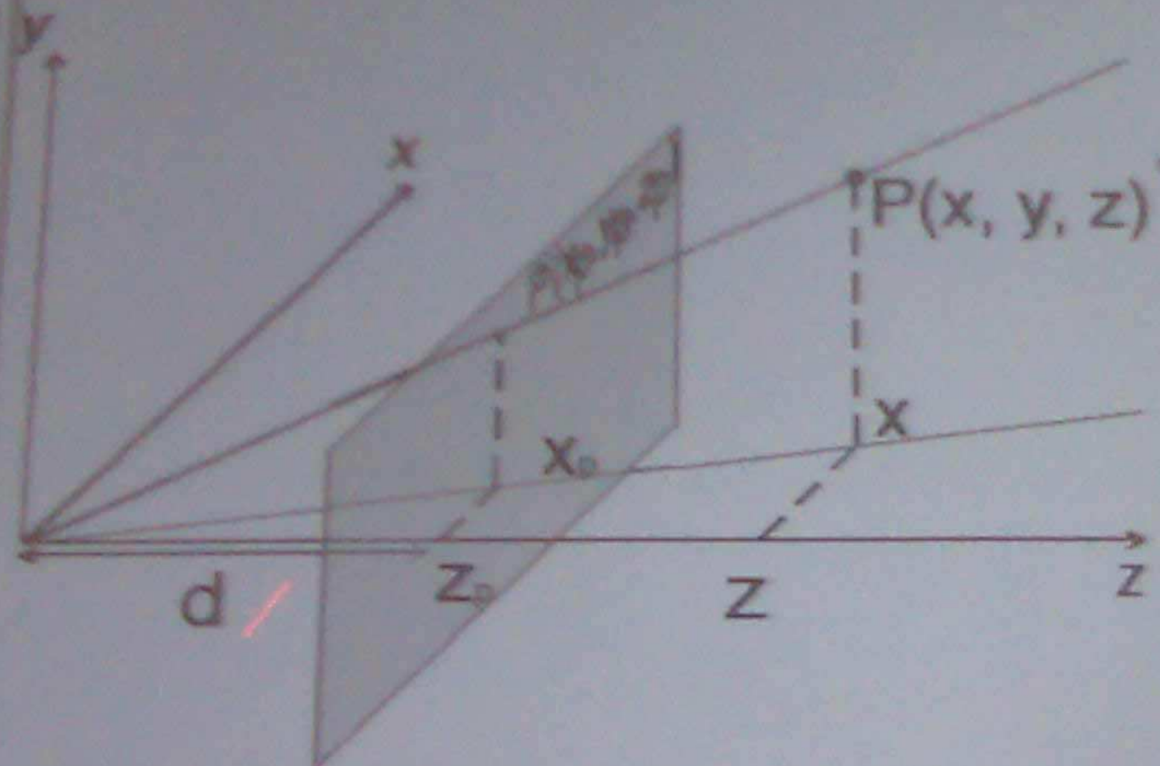
  $x=-1,\ x=1,\ y=-1,\ y=1,\ z=0,\ z=-1$

- Rzut perspektywiczny

  $x=z,\ x=-z,\ y=z,\ y=-z,\ z=z_{max},\ z=-1$



Przekształcenia macierzowe rzutu



$$\frac{x_p}{d} = \frac{x}{z} \qquad \frac{y_p}{d} = \frac{y}{z}$$

$$x_p = \frac{x}{\frac{z}{d}} \qquad y_p = \frac{y}{\frac{z}{d}}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_{per} \bullet P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ \frac{z}{d} \end{bmatrix}$$

Po normalizacji (tak by W = 1)

$$\begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ \frac{z}{z/d} = d \\ 1 \end{bmatrix}$$

Reprezentacja brył i powierzchni

- modelowanie istniejących obiektów
  - dobrze znany wygląd obiektu
  - brak jednoznacznego (matematycznego) opisu powierzchni
  - wymaganie nieskończenie wiele punktów do opisu
  - *Aproksymacja* obiektu
- obiekty nierzeczywiste
  - swoboda w reprezentacji obiektu

Podstawowe reprezentacje powierzchni

- Siatki wielokątowe
- Powierzchnie parametryczne
- Powierzchnie drugiego stopnia

Siatki wielokątowe

Siatka wielokątowa to zbiór, połączonych płaskich powierzchni ograniczonych przez łamane zamknięte.

- Reprezentacja obiektów
- Powierzchnie krzywoliniowe reprezentowane jedynie z pewnym przybliżeniem

Reprezentacja siatek wielokątowych

- Reprezentacja bezpośrednia

  $P = (V_1, V_2, \ldots, V_n) = ((x_1,y_1,z_1), (x_2,y_2,z_2), \ldots, (x_n,y_n,z_n))$

- Wskaźniki na listę wierzchołków

  $V = (V_1, V_2, \ldots, V_n)$

  $P_1 = (1,2,4)$; $P_2 = (4,2,3)$

- Lista krawędzi wielokąta

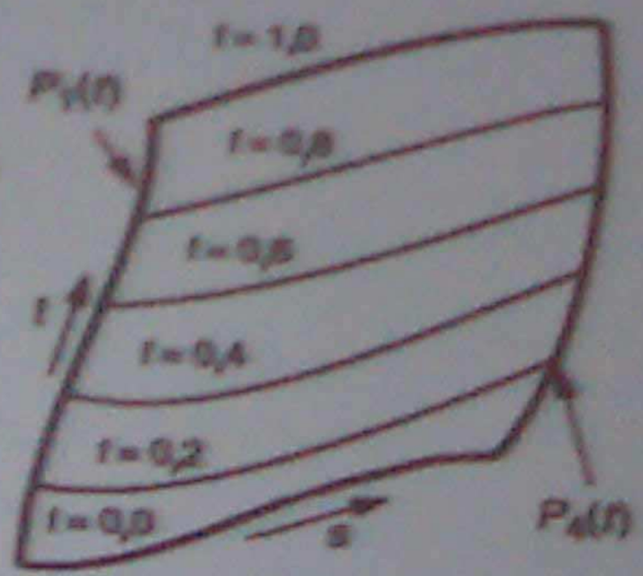
  $V = (V_1, V_2, \ldots, V_n)$

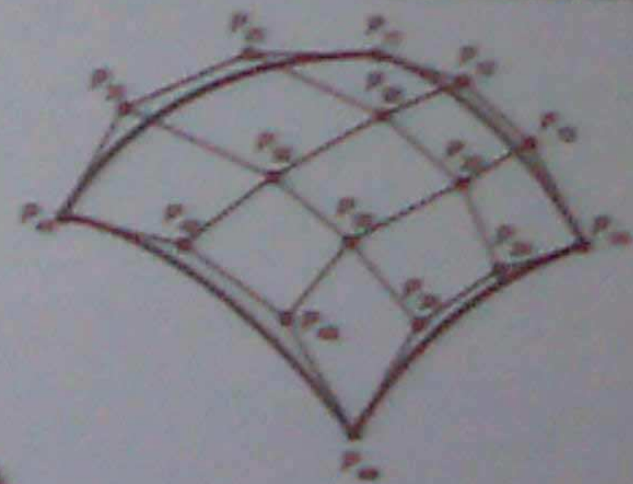
  $E_1 = (V_1, V_2, P_1, P_2)$; $E_2 = (V_2, V_3, P_2, 0)$; ...

  $P_1 = (E_1, E_4, E_5)$; $P_2 = (E_2, E_3, E_4)$;

Powierzchnie parametryczne

- Wielomianowe krzywe parametryczne trzeciego stopnia $Q(t) = (f_x(t), f_y(t), f_z(t))$
- Dobór współczynników
- Parametryczne wielomianowe płaty powierzchni
  - Współrzędne punktu powierzchni określane są poprzez dwa parametry
  - $Q(s, t) = (f_x(s,t), f_y(s,t), f_z(s,t))$
  - Brzegi są krzywymi parametrycznymi





Powierzchnie drugiego stopnia

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

Jeśli $f(x,y,z)$ jest wielomianem drugiego stopnia, to Mówimy o powierzchniach drugiego stopnia.

$$f(x,y,z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k$$

$$P^T \bullet Q \bullet P = 0$$

$$Q = \begin{bmatrix} a & d & f & g \\ d & b & e & h \\ f & e & c & j \\ g & h & j & k \end{bmatrix} \qquad P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Powierzchnie drugiego stopnia(cd)

Zalety reprezentacji

- Obliczanie normalnej do powierzchni
- Testowanie czy punkt leży na powierzchni
- Łatwe obliczanie z dla danych współrzędnych punktu (x, y)
- Obliczanie przecięcia jednej powierzchni z drugą

Zadanie

Wiedząc, że powierzchnia drugiego stopnia opisana jest równaniem macierzowym $P^T \cdot Q \cdot P = 0$. Jaka bryła reprezentowana jest dla

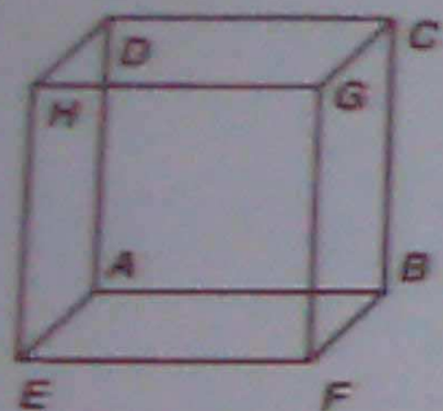
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Reprezentacja brył

Jeśli powierzchnie 3D opisują brzeg zamkniętego obszaru (o określonej objętości) mówimy o *modelowaniu brył*

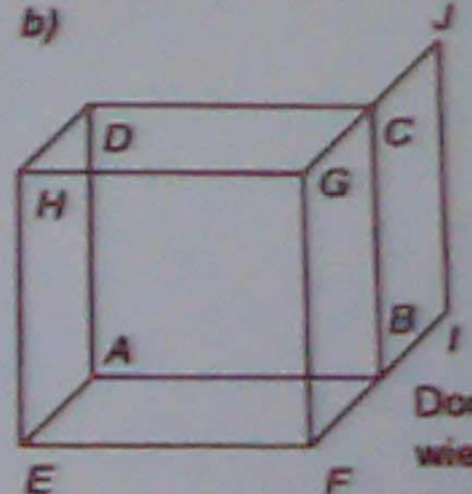
- Wnętrze obiektu
- Kolizje obiektów
- Modelowanie własności fizycznych itp.

a)



Wierzchołki		Odcinki
A	(0, 0, 0)	*AB*
B	(1, 0, 0)	*BC*
C	(1, 1, 0)	*CD*
D	(0, 1, 0)	*DA*
E	(0, 0, 1)	*EF*
F	(1, 0, 1)	*FG*
G	(1, 1, 1)	*GH*
H	(0, 1, 1)	*HE*
		AE
		BF
		CG
		DH

b)



Dodatkowe wierzchołki		Dodatkowe odcinki
I	(1, 0, −1)	*BI*
J	(1, 1, −1)	*IJ*
		JC

Własności dobrej reprezentacji brył

- *Domena reprezentacji* – możliwość przedstawienie możliwie wielu obiektów fizycznych
- *Niedwuznaczność, kompletność* – wiemy dokładnie co jest reprezentowane
- *Unikatowość* umożliwia kodowanie dowolnej bryły tylko w jeden sposób (sprawdzanie czy dwa obiekty są równe)
- *Dokładność* – reprezentacja bez przybliżania
- *Poprawność* – tylko prawidłowe obiekty
- *Domknięcie* – operacje na poprawnych bryłach powinny dawać poprawne bryły (obrotu, skalowania i inne np. operacje boolowskie)
- *Zwartość, oszczędność* i *efektywność*

Bryły – operacje boolowskie

Zbiór operacji boolowskich

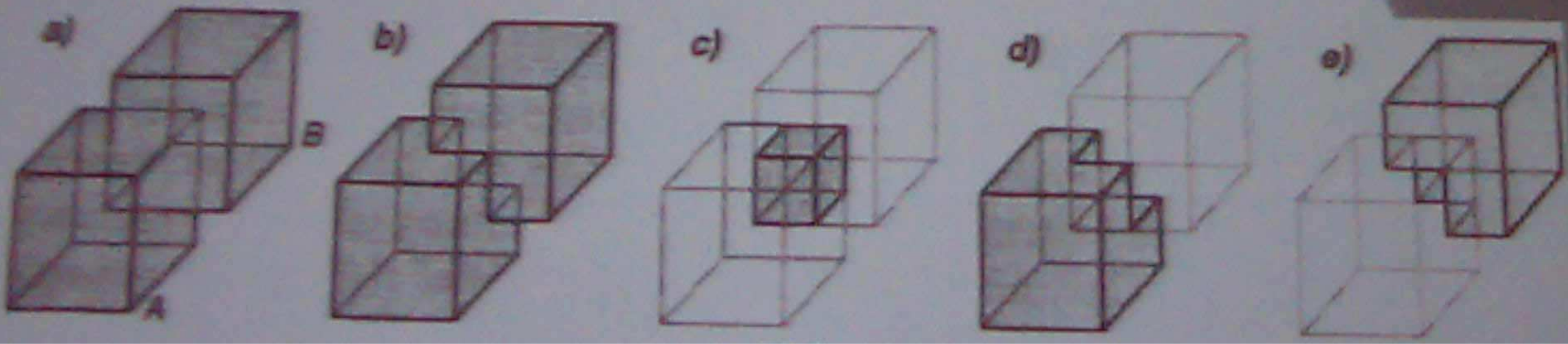
- Suma
- Różnica
- Przecięcie

W wyniku tych operacji m powstać bryły, odcinki punkty.

Are you sure to power off ?

Press <Power> again to power off

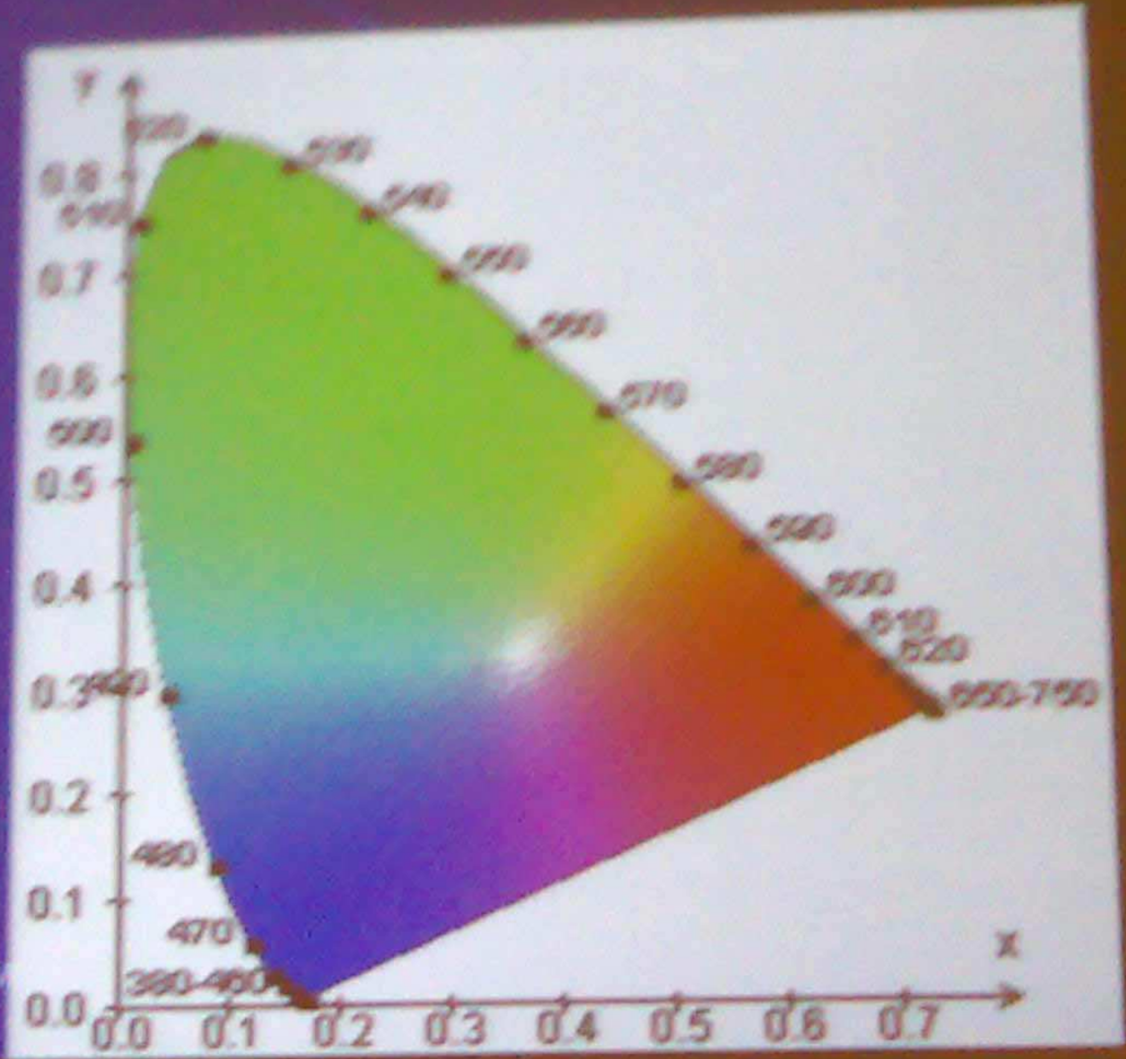
Regularyzow konanie operacji na bryłach daje bryłę.



Wykres chromatyczności

Międzynarodowej Komisji Oświetlenia CIE

Fikcyjne barwy podstawowe X, Y, Z umożliwiają określenie wszystkich barw widzialnych



Inne modele barw

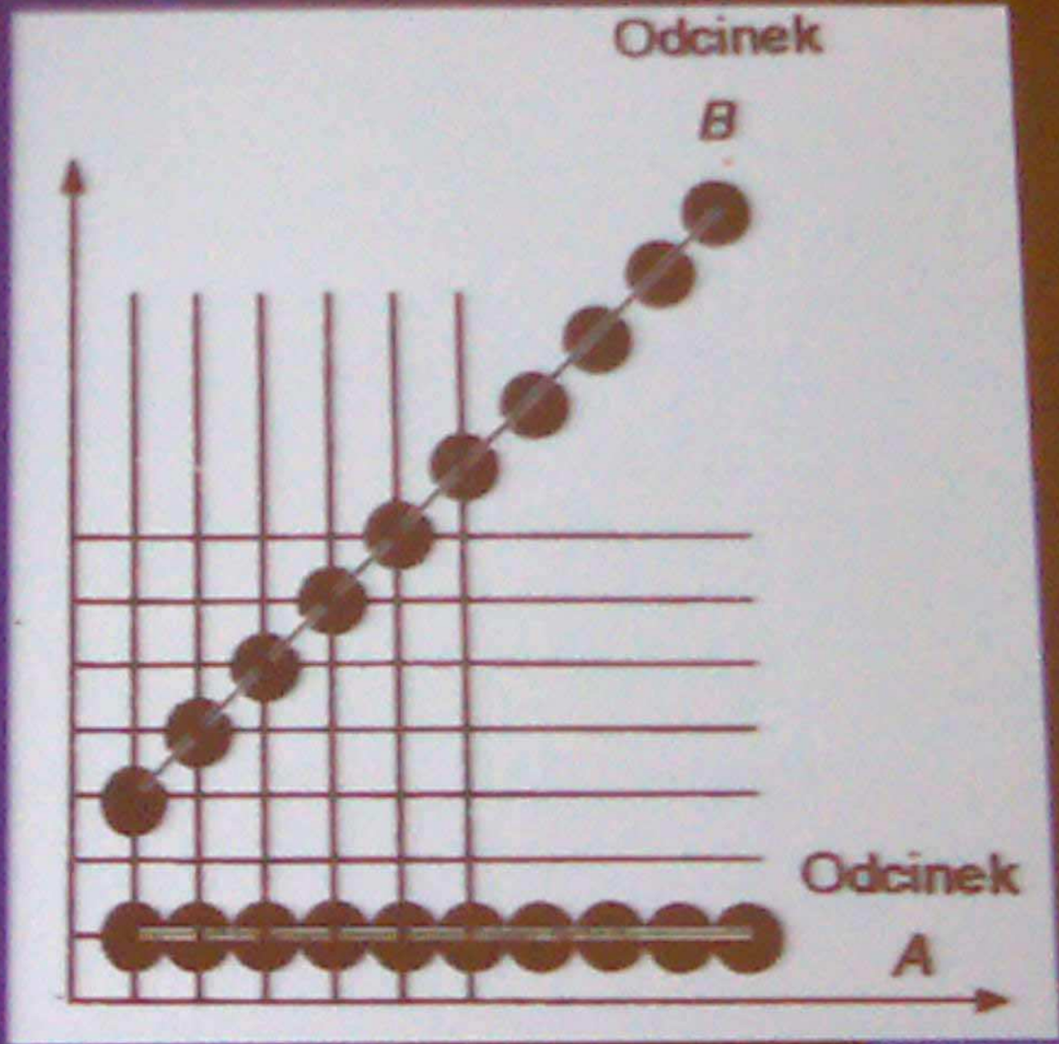
- Modele telewizyjne YIQ, YUV
- Modele percepcyjnie równoważne
  - CIE L*u*v – dla urządzeń emitujących
  - CIE L*a*b – dla określenia koloru obiektu (światło odbite)
- *Profil urządzenia* (ICC) określa sposób przeliczania barwy z modelu odniesienia (np. Lab CIE) do modelu urządzenia

Inne modele barw

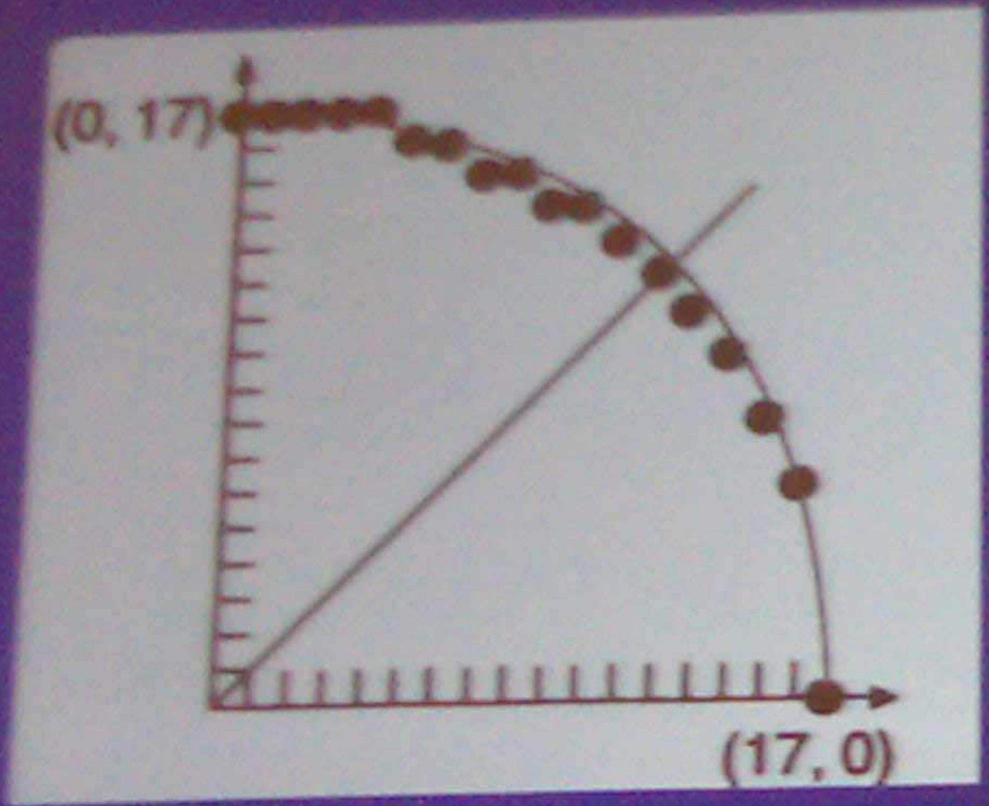
- Modele telewizyjne YIQ, YUV
- Modele percepcyjnie równoważne
  - CIE L*u*v – dla urządzeń emitujących
  - CIE L*a*b – dla określenia koloru obiektu (światło odbite)
- *Profil urządzenia* (ICC) określa sposób przeliczania barwy z modelu odniesienia (np. Lab CIE) do modelu urządzenia

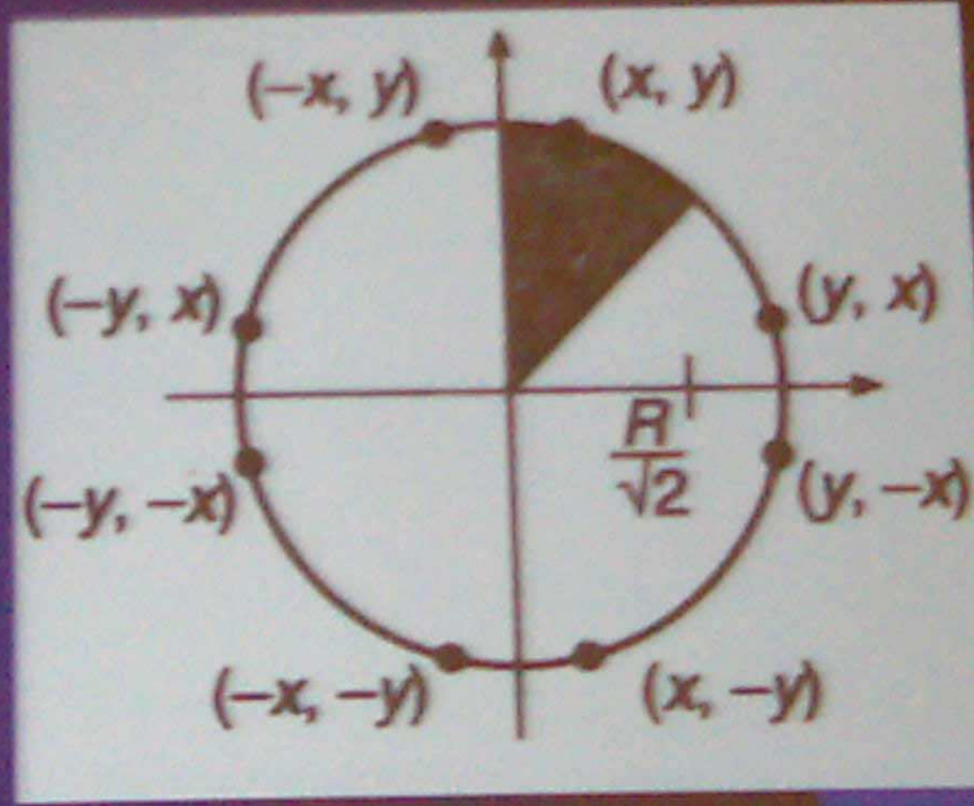
Problemy

- kierunek rysowania
- obcinanie
- jasność odcinka
- łamane



Rysowanie okręgów (1)

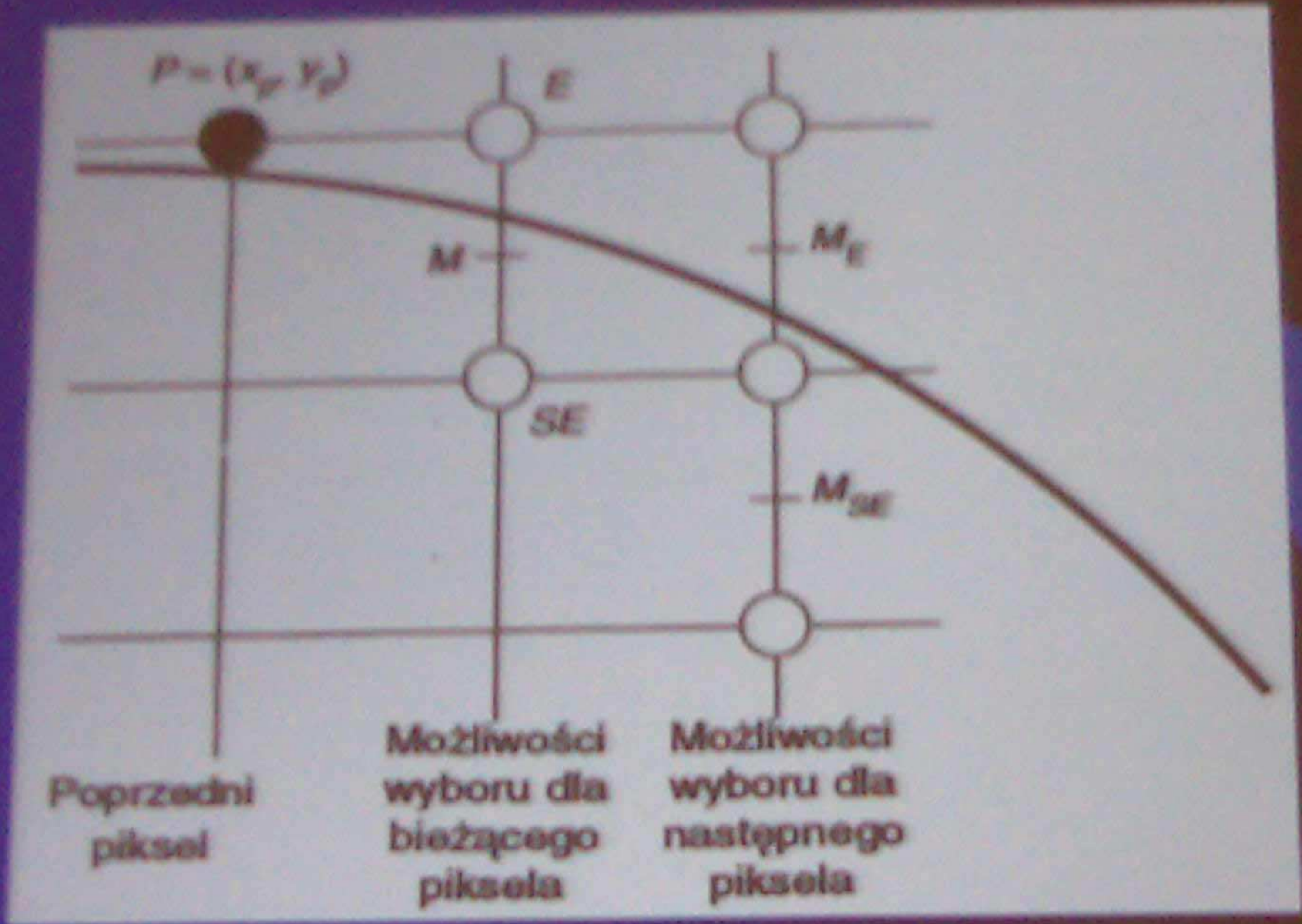




$$x^2 + y^2 = R^2$$

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

Rysowanie okręgów (2)



$$F(x,y) = x^2 + y^2 - R^2$$

Obliczanie zmiennych decyzyjnych (1)

$$d = F(M) = F(x_p+1,\ y_p-1/2)$$
$$= (x_p+1)^2 + (y_p-1/2)^2 - R^2$$

Jeśli E (to M przesuwa się w prawo o 1)

$$d_{new} = F(x_p+2,\ y_p-1/2)$$
$$= (x_p+2)^2 + (y_p-1/2)^2 - R^2$$
$$= (x_p^2+4x_p+4) + (y_p-1/2)^2 - R^2$$
$$= (x_p^2+2x_p+1) + 2x_p+3 + (y_p-1/2)^2 - R^2$$
$$= \underline{(x_p+1)^2} + 2x_p + 3 + \underline{(y_p-1/2)^2} - \underline{R^2} +$$
$$= d + 2x_p + 3$$

Obliczanie zmiennych decyzyjnych (2)

<u>Jesli SE</u> (to M przesuwa się w prawo o 1 i w dół o 1)

$$
\begin{aligned}
d_{new} &= F(x_p+2,\ y_p - 3/2)\\
&= (x_p+2)^2 + (y_p-3/2)^2 - R^2\\
&= (x_p^2 + 4x_p + 4) + (y_p^2 - 3y_p + 9/4) - R^2\\
&= (x_p+1)^2 + 2x_p + 3 + (y_p^2 - y_p + 1/4) - 2y_p + 8/4 - R^2\\
&= \underline{(x_p+1)^2} + 2x_p + 3 + \underline{(y_p-1/2)^2} - 2y_p + 2 - R^2\\
&= d + 2x_p - 2y_p + 5
\end{aligned}
$$

Obliczenia wartości startowych

- Punkt startu (x_p , y_p) = (0, R)

$$d = F(x_p + 1, y_p - 1/2) = F(1, R - 1/2)$$
$$= 1^2 + (R - 1/2)^2 - R^2$$
$$= 1^2 + (R^2 - R + 1/4) - R^2$$
$$= 5/4 - R$$

void MidCircle(int R)

```
{ int x, y;
  float d;
  x = 0;
  y = r;
  d = 5.0 / 4 - R;
  CirclePoints(x, y);
  while (y > x) {
    if (d < 0) {      /* piksel E */
      d += x * 2.0 + 3;
      x++;
    } else {          /* piksel SE */
      d += (x - y)*2.0 + 5;
      x++;
      y--;
    }
    CirclePoints(x, y);
  }
}
```

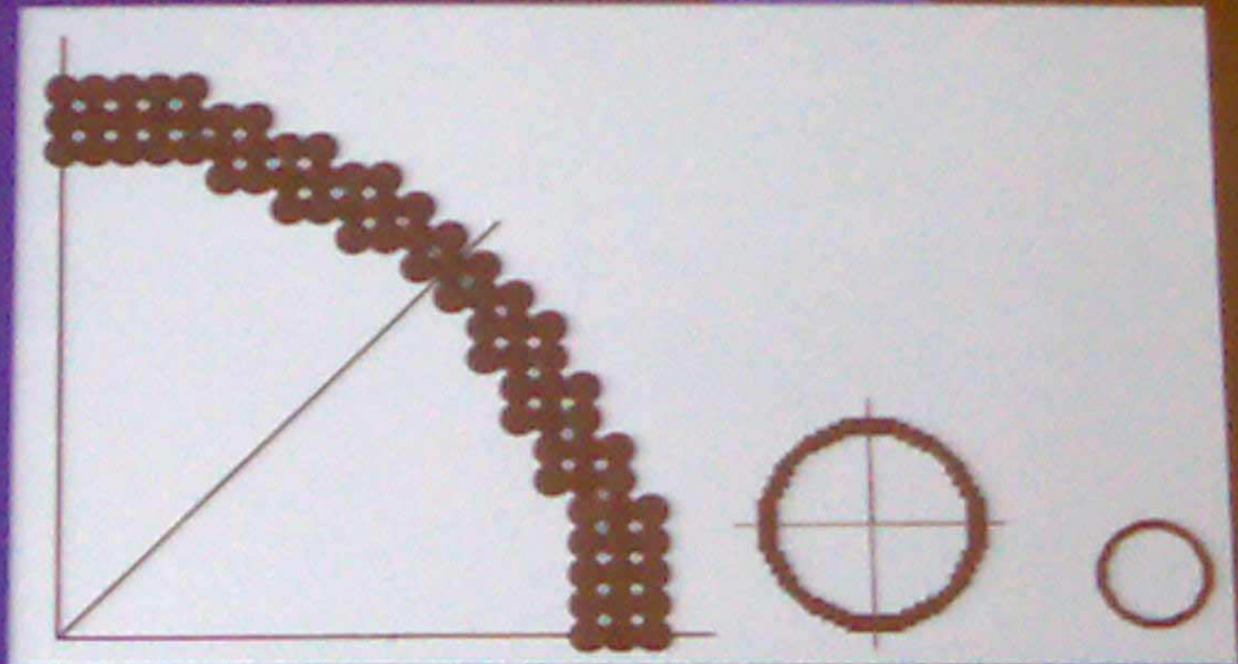
void MidCircle(int R)

```
{ int x, y;
  float d;
  x = 0;
  y = r;
  d = 5.0 / 4 - R;
  CirclePoints(x, y);
  while (y > x) {
    if (d < 0) {        /* piksel E */
      d += x * 2.0 + 3;
      x++;
    } else {            /* piksel SE */
      d += (x - y)*2.0 + 5;
      x++;
      y--;
    }
    CirclePoints(x, y);
  }
}
```

Rysowanie okręgów

Modyfikacje algorytmu

- zmiana środka okręgu
- aspekt monitora



Inne zagadnienia

- kierunek rysowania
- pogrubianie linii
- styl linii

Przykład

Narysować okrąg o środku w punkcie (0,0) i promieniu R = 6:

$d = 5/4 - 24/4 = -19/4$

<u>(0,6)</u> $d<0$ to E

$d = d + 2x+3 = -19/4 + 12/4 = -7/4$

$x = 1$

<u>(1,6)</u> $d<0$ to E

$d = -7/4 + 8/4 + 12/4 = 13/4$

$x = 2$

<u>(2,6)</u> $d>0$ to SE

$d = d + 2(x-y) + 5 = 13/4 - 32/4 + 20/4 = 1/4$

$x = 3; y = 5$

<u>(3,5)</u> $d > 0$ to SE

$d = 1/4 - 8/4 + 20/4 = 13/4$

$x = 4; y = 4$

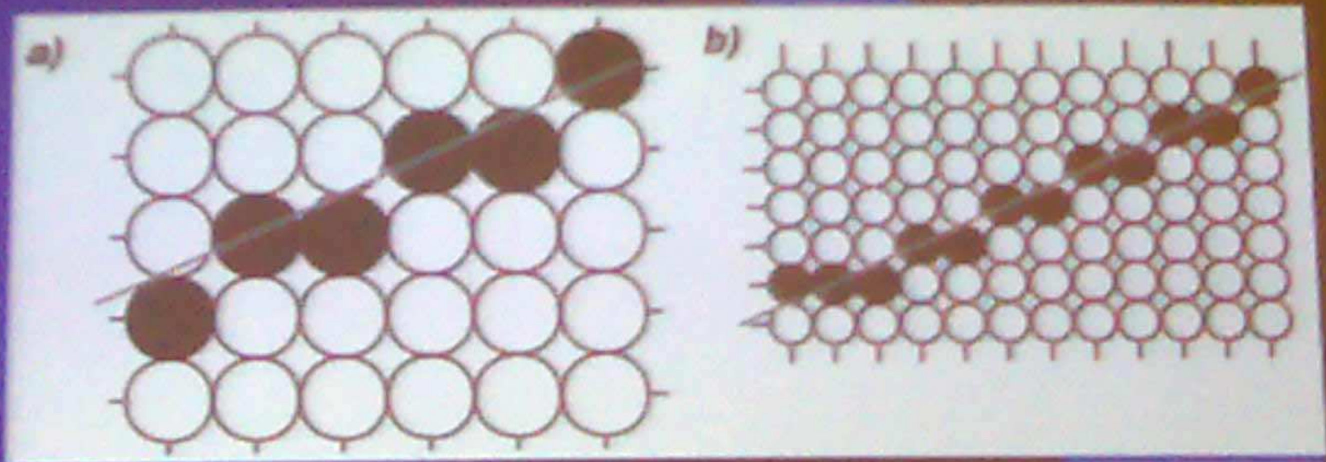
<u>(4,4)</u>

Usuwanie zakłóceń

Rysowanie w przestrzeni dyskretnej powoduje powstanie w obrazie zakłóceń (*Aliasing*)

Metody odkłócania (*Antyaliasing*)

- zwiększenie rozdzielczości
  - Zwiększenie rozmiaru pamięci obrazu



  - Zwiększenie czasu rysowania prymitywu
  - Zwiększenie pasma pamięci i pasma monitora

Sprawy organizacyjne

Krzysztof Gracki
k.gracki@ii.pw.edu.pl
tel. (22) 6605031

- regulamin przedmiotu
- kolokwia 7 XI, 19 XII
- egzamin 23 I
- ocena końcowa

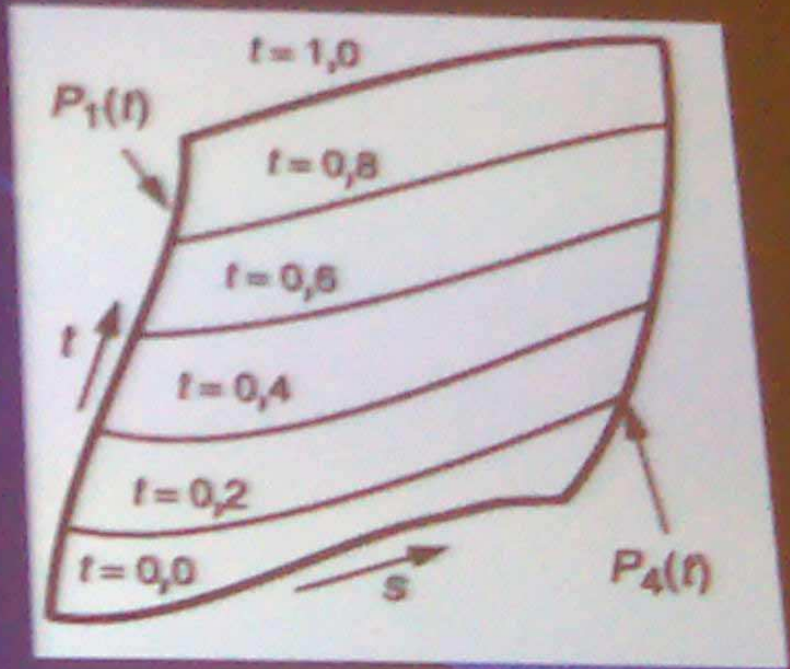
- Laboratorium
  Zbigniew Szymański
  z.szymanski@ii.pw.edu.pl
- Regulamin
  - Corel
  - 3DStudioMax

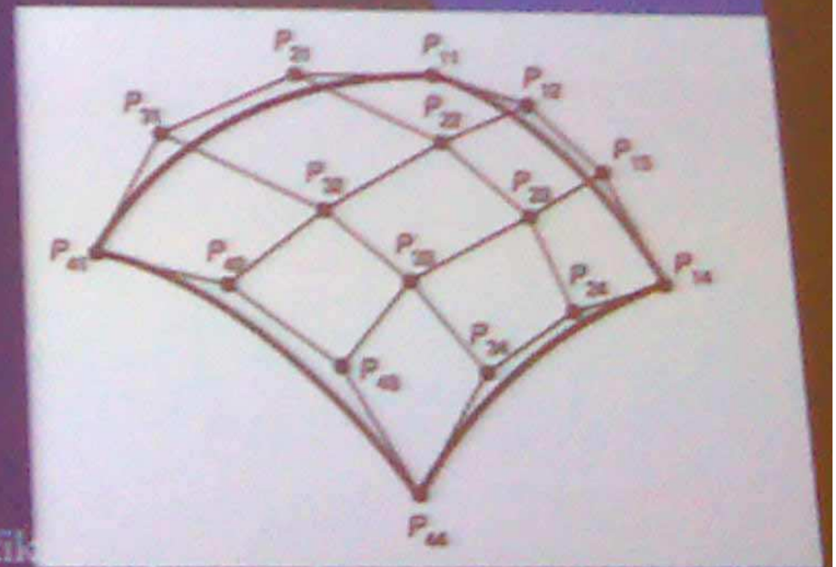
Jakość obrazu a percepcja człowieka

- Człowiek jest w stanie rozróżniać szczegóły widziane w obrębie kąta o wartości około 1 minuty kątowej.
- Dla typowej odległości obserwatora od ekranu monitora: około 0,5 metra, maksymalna sensowna rozdzielczość waha się w przedziale 100-200 punktów (pikseli) na centymetr
- Współczesne (standardowe) monitory mają 30-50 pkt cm
- Monitory o rozdzielczości pikselowej 4096 × 3072 praktycznie zapewniają maksymalna jakość obrazu

Powierzchnie parametryczne

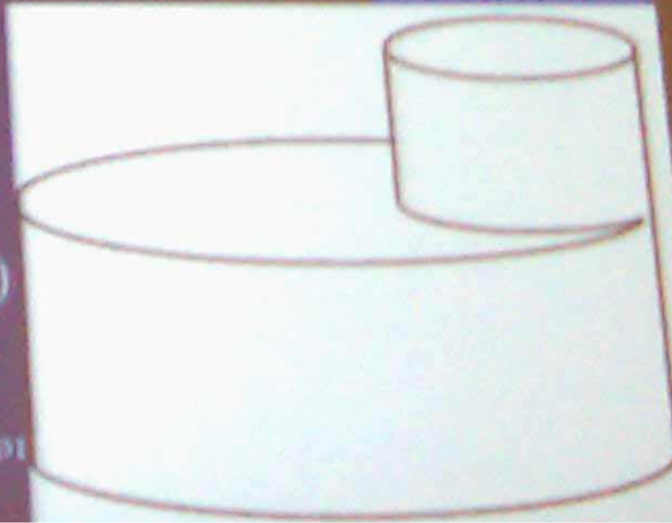
- Wielomianowe krzywe parametryczne trzeciego stopnia $Q(t) = (f_x(t), f_y(t), f_z(t))$
- Dobór współczynników
- Parametryczne wielomianowe płaty powierzchni
  - Współrzędne punktu powierzchni określane są poprzez dwa parametry
  - $Q(s, t) = (f_x(s,t), f_y(s,t), f_z(s,t))$
  - Brzegi są krzywymi parametrycznymi





Sposoby reprezentacji brył

- Kopiowanie prymitywów
  - Systemy CAD
  - Biblioteki gotowych parametryzowanych elementów
  - Standardowe wymiary, normy
- Reprezentacje z przesuwaniem (zagarnianie przestrzeni)
  - Przesuwanie obiektu wzdłuż trajektorii
  - Przesunięcia obrotowe
  - Przesunięcia ogólne
- Reprezentacja brzegowa (b-rep)
  - Opis obiektu powierzchniami ograniczającymi
  - Powierzchnie płaskie (np. triangulacja)
  - Powierzchnie krzywoliniowe



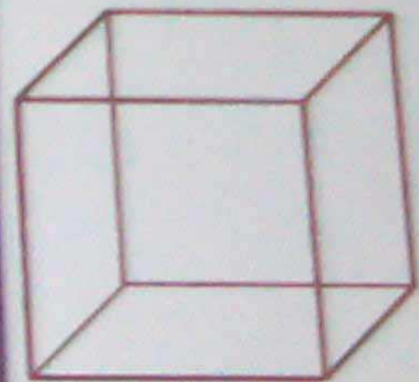
Reguła Eulera

Wielościan prosty to taki, który da się przekształcić w kulę

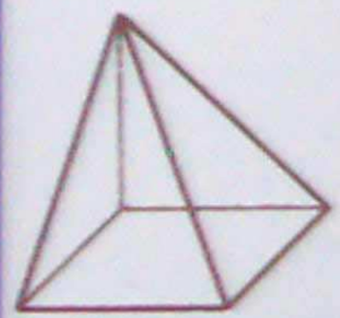
Reprezentacja brzegowa wielościanu prostego spełnia równanie Eulera

$V - E + F = 2$

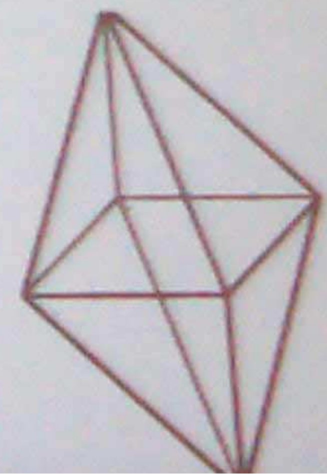
V - liczba wierzchołków
E - liczba krawędzi
F - liczba ścian



V=8
E=12
F=6



V=5
E=8
F=5



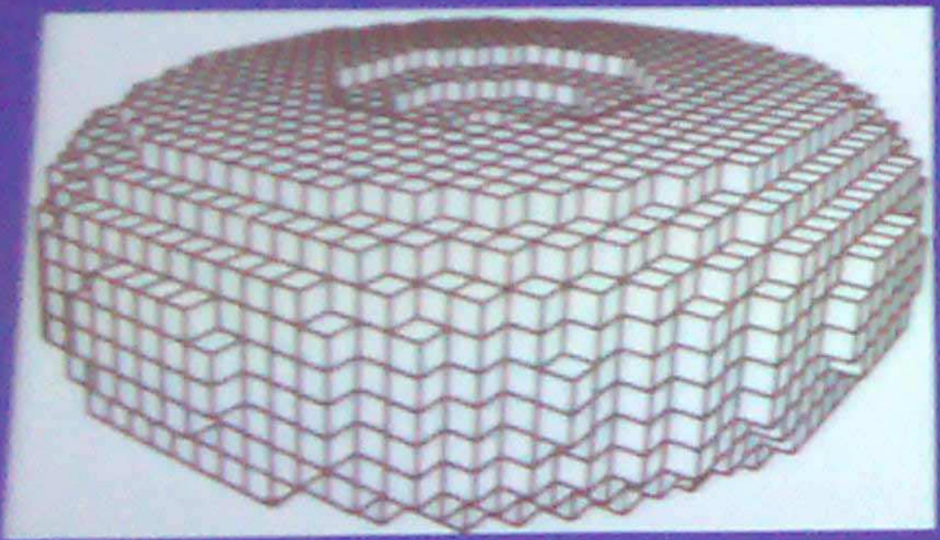
V=6
E=12
F=8

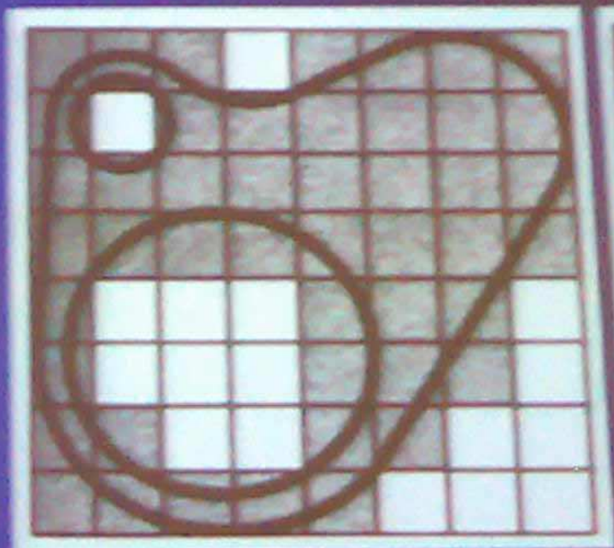
Sposoby reprezentacji brył (c.d.)

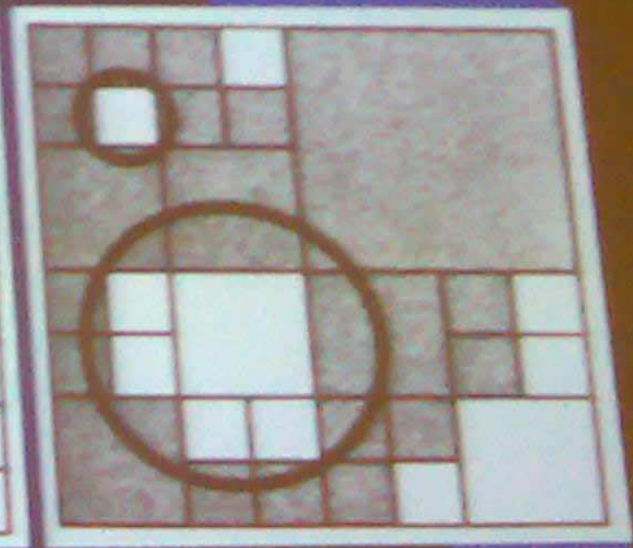
- Reprezentacja z podziałem przestrzennym

Bryła jest dekomponowana na zbiór prostszych nie przecinających się brył.

- Dekompozycja na przylegające komórki
- Reprezentacja wokselowa (identyczne komórki)
- Drzewa ósemkowe



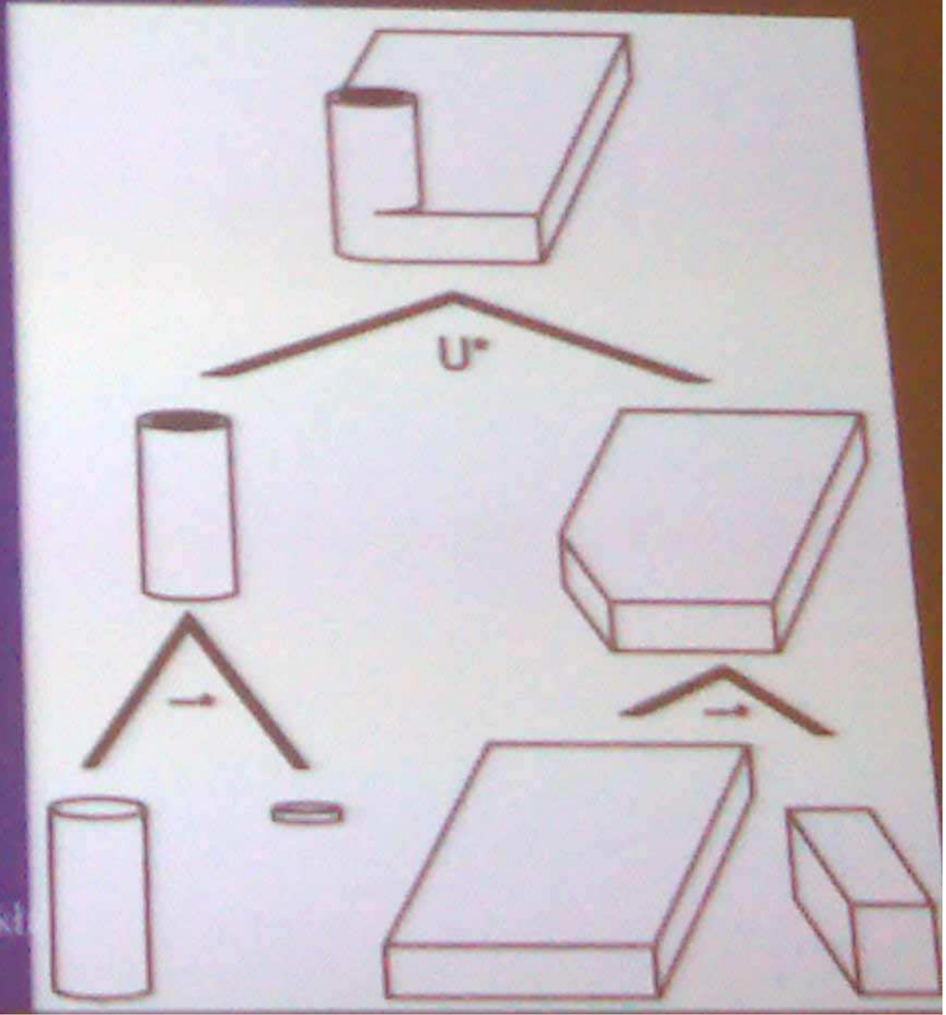




Sposoby reprezentacji brył (c.d.)

- Konstruktywna geometria brył (CSG)

  Łączenie prostych prymitywów za pomocą regularyzowanych operatorów boolowskich (włączonych do reprezentacji)



- Inne
  - Metakule
  - Systemy cząstek
  - Modele fraktalne
  - L-Systemy

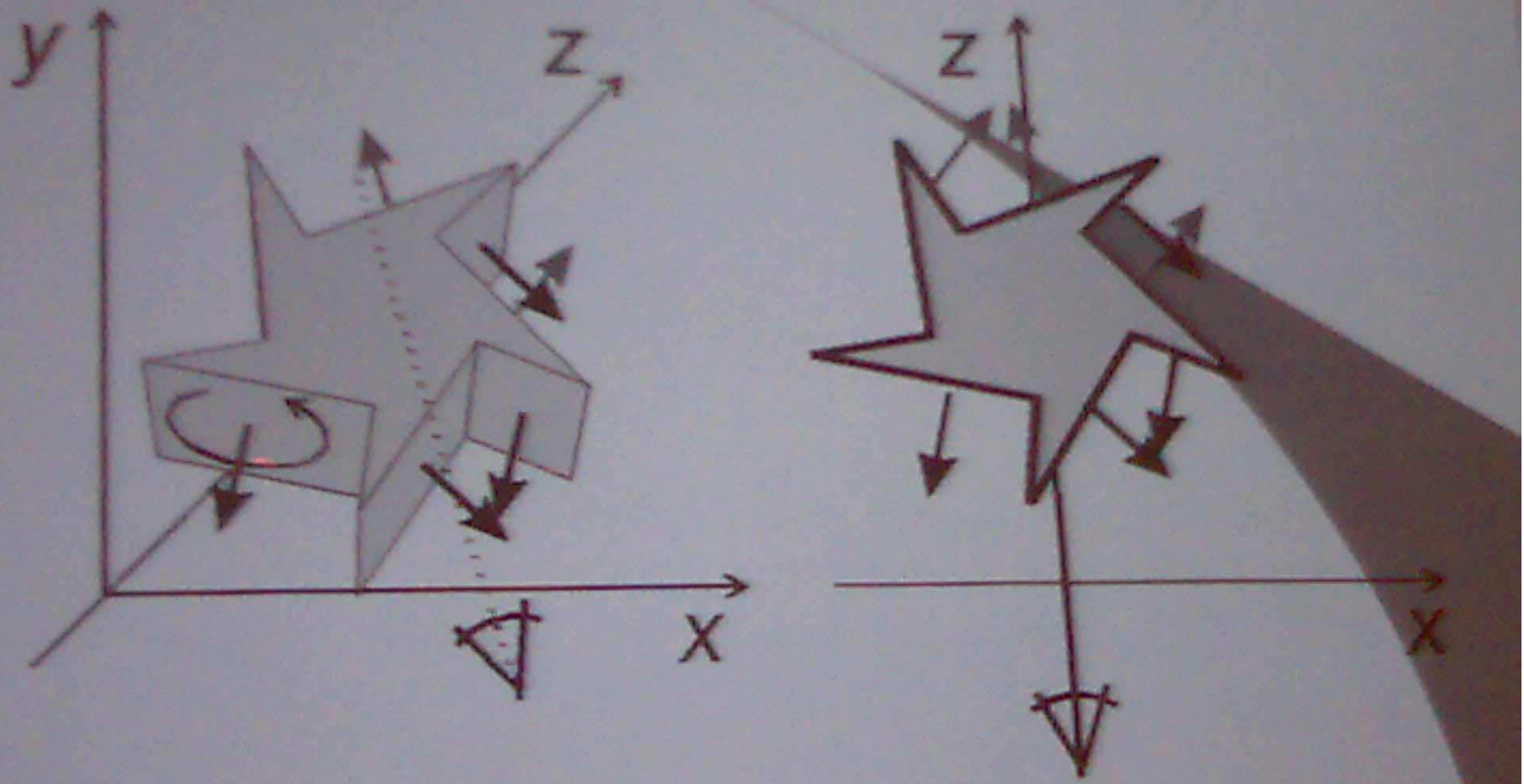
Porównanie reprezentacji

- Dokładność:
  - Podział przestrzenny
  - B-rep (wielokątowa)
  - CSG (bryły gładkie)
  - B-rep (powierzchnie krzywoliniowe)
- Dziedzina:
  - Kopiowanie prymitywów
  - Przesuwanie
  - Podział przestrzeni
  - B-rep (teoretycznie)
- Unikatowość:
  - Drzewa ósemkowe
  - Metody wokselowe
- Poprawność
  - B-rep
  - CSG (mało sprawdzeń)
  - Wokselowa
- Domknięcie
  - Kopiowanie prymitywów
- Efektywność
  - Modele nieprzetworzone np. CSG
  - Modele przetworzone np. wokselowe

Porównanie reprezentacji

- Dokładność:
  - Podział przestrzenny
  - B-rep (wielokątowa)
  - CSG (bryły gładkie)
  - B-rep (powierzchnie krzywoliniowe)
- Dziedzina:
  - Kopiowanie prymitywów
  - Przesuwanie
  - Podział przestrzeni
  - B-rep (teoretycznie)
- Unikatowość:
  - Drzewa ósemkowe
  - Metody wokselowe
- Poprawność
  - B-rep
  - CSG (mało sprawdzeń)
  - Wokselowa
- Domknięcie
  - Kopiowanie prymitywów
- Efektywność
  - Modele nieprzetworzone np. CSG
  - Modele przetworzone np. wokselowe

Wybieranie ścian tylnych



Własności wyświetlanych scen

- Spójność obiektów
  Dla rozłącznych obiektów możemy porównywać obiekty a nie wszystkie ich ściany
- Spójność ścian
- Spójność krawędzi
- Spójność powierzchni
- Spójność głębokości
- Spójność ramek

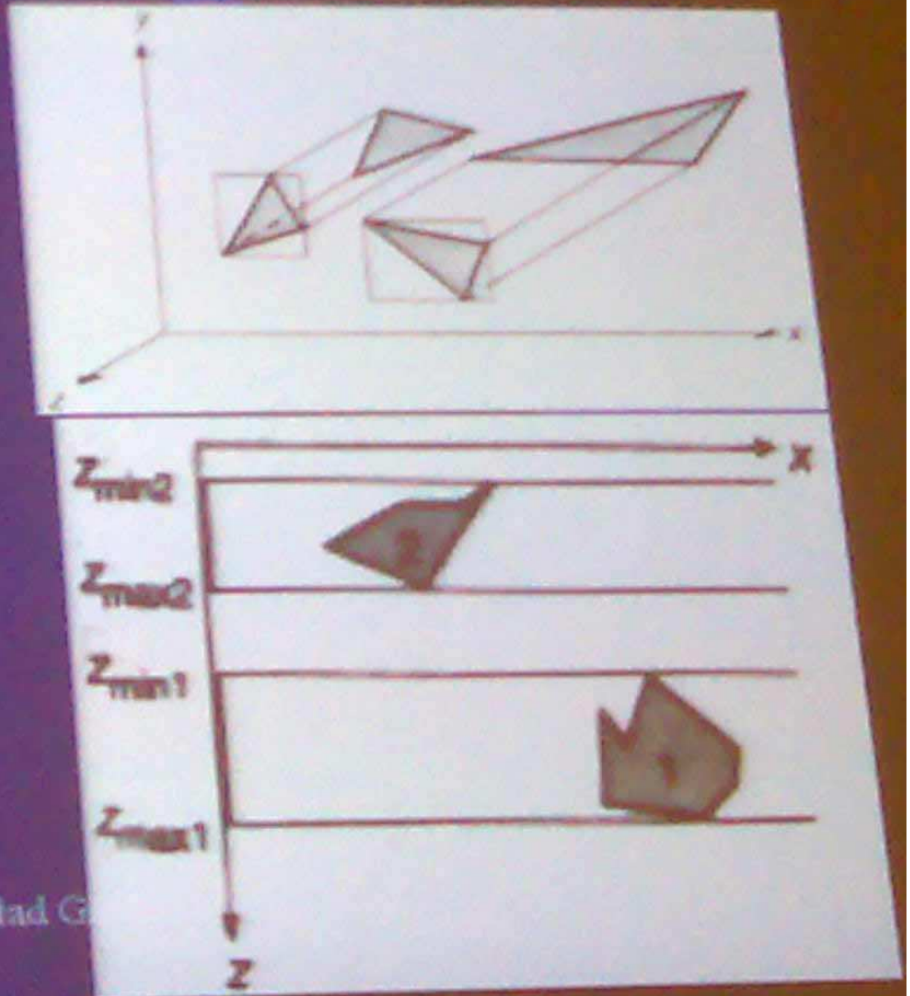
Algorytm Malarski (sortowanie ścian)

Problemy związane z zaslanianiem obiektów (lub jego fragmentów) może być traktowane jako zadanie sortowania ścian.

Tworząc obraz wystarczy posortować ściany względem odległości od obserwatora i rysować je (wypełnione wielokąty) zaczynając od ściany położonej najdalej.

Efektywne obliczanie zasłaniania

- Jeśli na rzutni prostokąty ograniczające się nie przecinają to żadna z dwóch brył nie zaslania drugiej
- *Testowanie minmax*
  Jeśli wzdluż osi z (kierunku patrzenia) wszystkie wierzcholki pierwszej bryly leżą przed wszystkimi wierzcholkami drugiej, to druga nie może zasłaniać pierwszej



Algorytm sortowania ścian

1. Wybieramy ścianę **P** leżącą najdalej obserwatora (o największej współrzędnej z)
2. Jeśli z-ograniczenia **P** i pozostałych ścian (**Q**$_i$) są rozłączne to **P** nie może zasłaniać żadnej ściany
   - Rysujemy i wypełniamy **P** i rozpatrujemy pozostałe ściany (pkt1.)
3. Jeśli nie (z-ograniczenia ściany **P** i **Q** przecinają się) to sprawdzamy :

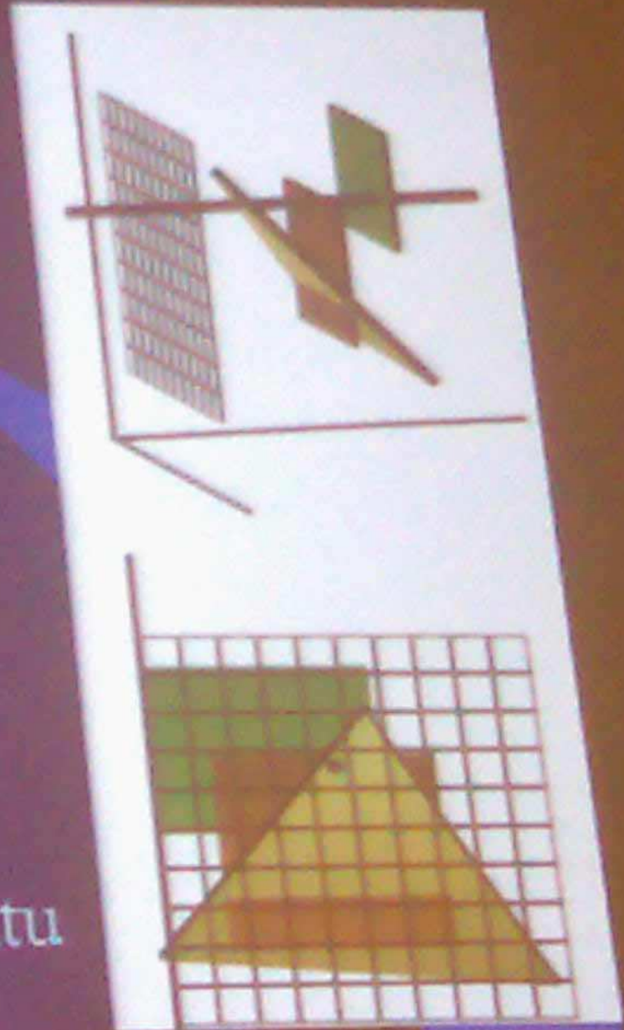
A) czy są rozłączne x-ograniczenia

B) czy są rozłączne y-ograniczenia

Algorytm Z-Bufora (bufora głębokości)

Dla każdego piksela rzutu (oprócz koloru) należy przechowywać współrzędną z narysowanego wielokąta

- Pamięć obrazu przechowuje wartości barw
- *Z-bufor* z zawiera informacje o odległości obiektu, którego barwa zapamiętana jest w pamięci obrazu.



Jeśli współrzędna z rozpatrywanego punktu (x, y) nie jest dalej od obserwatora niż punkt który został zapamiętany w buforach to barwa i głębokość zapisywana jest do pamięci.

Algorytm

Wartości w z-buforze $\in \langle 0, z_{max} \rangle$

- Dla wszystkich (x, y) $Z_{buf}[x, y] := 0$;
- Dla każdego wielokąta {

  Dla każdego piksela rzutu wielokąta {

  z := wartość współrzędnej z wielokąta dla piksela (x,y)

  if (z >= $Z_{buf}[x,y]$) {

  $Z_{buf}[x,y]$ = z; WritePiksel(x, y, kolor wielokąta)

  }}}

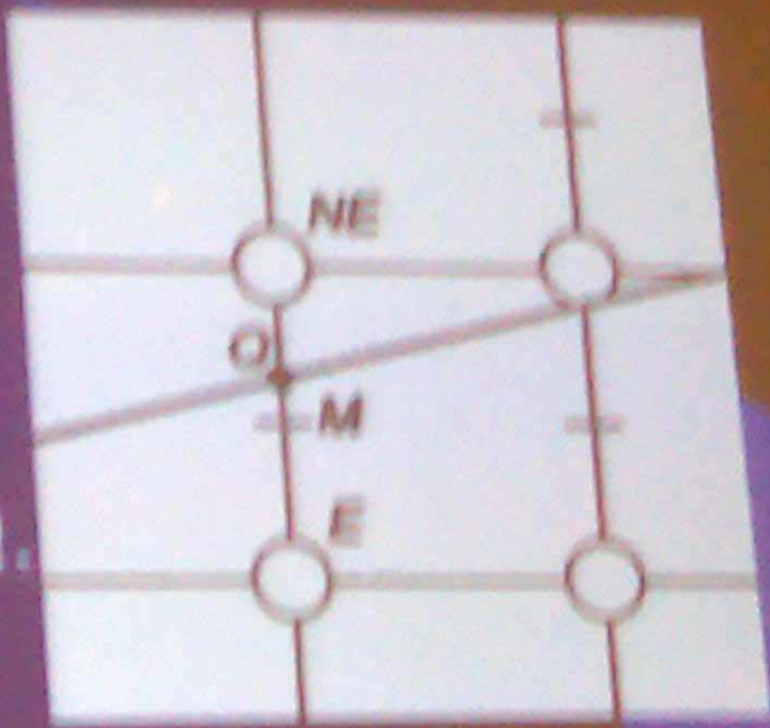
Zalety algorytmu

- Nie jest potrzebne wstępne sortowanie
- Łatwość implementacji (sprzętowej i programowej)
- Dowolna kolejność przeglądania wielokątów

Usuwanie zakłóceń(2)

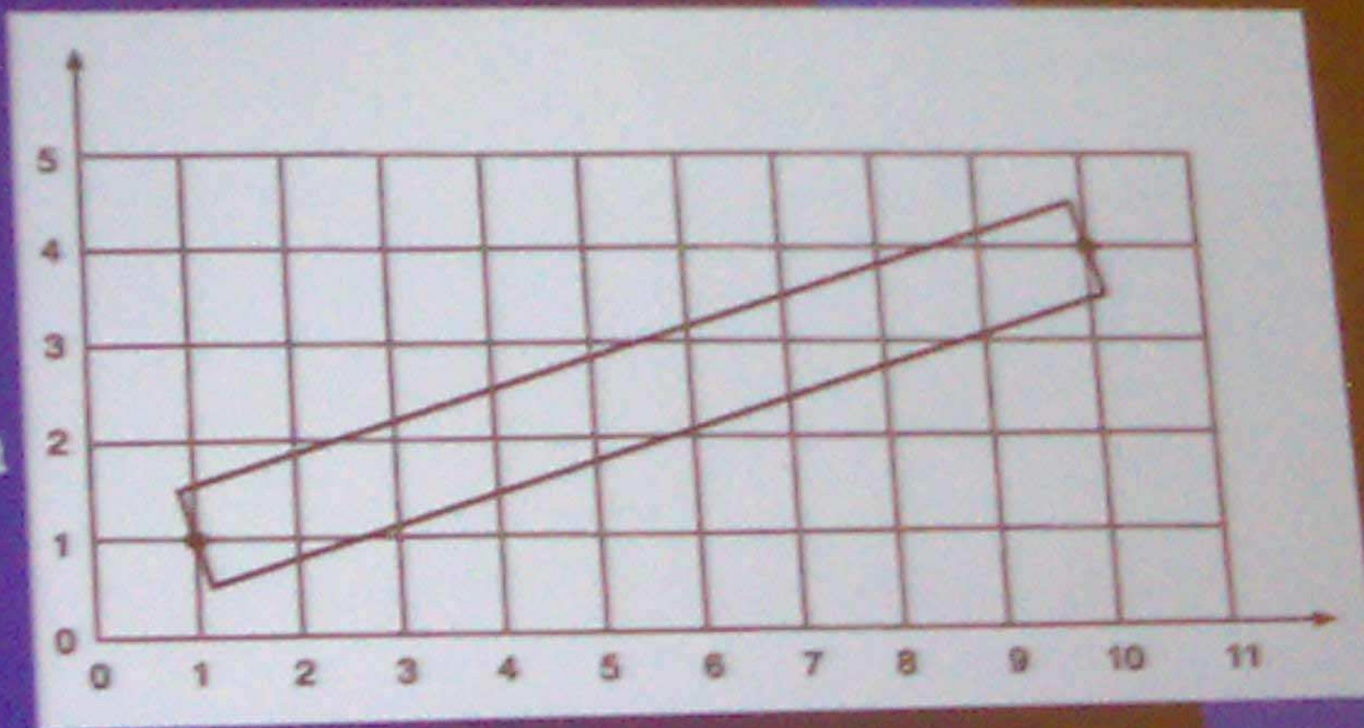
Metoda z dwoma pikselami w kolumnie

- Piksele leżą najbliżej idealnej linii
- Odcień piksela NE zależy od długości odcinka NE-Q, odcień piksela E od Q-E
- Suma odcieni pikseli NE i E jest stała



Próbkowanie powierzchni

- Kwadratowe piksele
- Odcinek ma określoną szerokość
- Odcinek ma wnosić pewien udział w zaczernienie piksela
- Wymagana jest wielobitowa reprezentacja piksela
- Podczas rysowania ustawianych jest kilka pikseli



Bezwagowe próbkowanie powierzchni

Jasność zaznaczonego piksela jest proporcjonalna do powierzchni zakrytej przez odcinek

Obliczanie jasności piksela

- Nadpróbkowanie (Oversampling)

Możliwe jest rysowanie linii o grubości mniejszej niż jeden piksel

Nadpróbkowanie (*oversampling*)

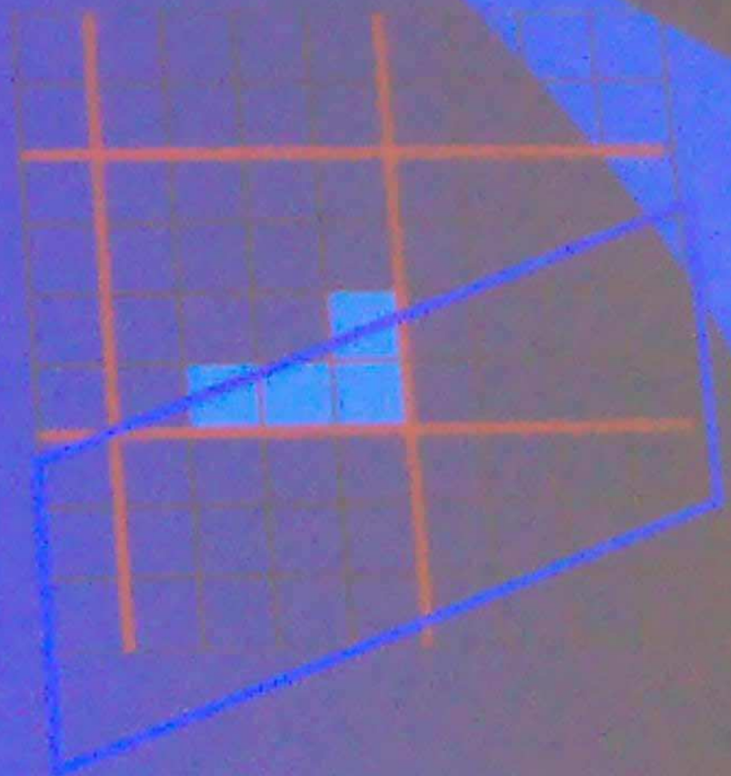
Odcinek o grubości jednego piksela rysujemy tak jakby był odcinkiem złożonym z pewnej liczby mniejszych pikseli

O jasności piksela decyduje liczba zakrytych pikseli

- Przykład

Piksel złożony z 16 „małych pikseli"

Jasność piksela = 4/16 I_{max} = 1/4 I_{max}

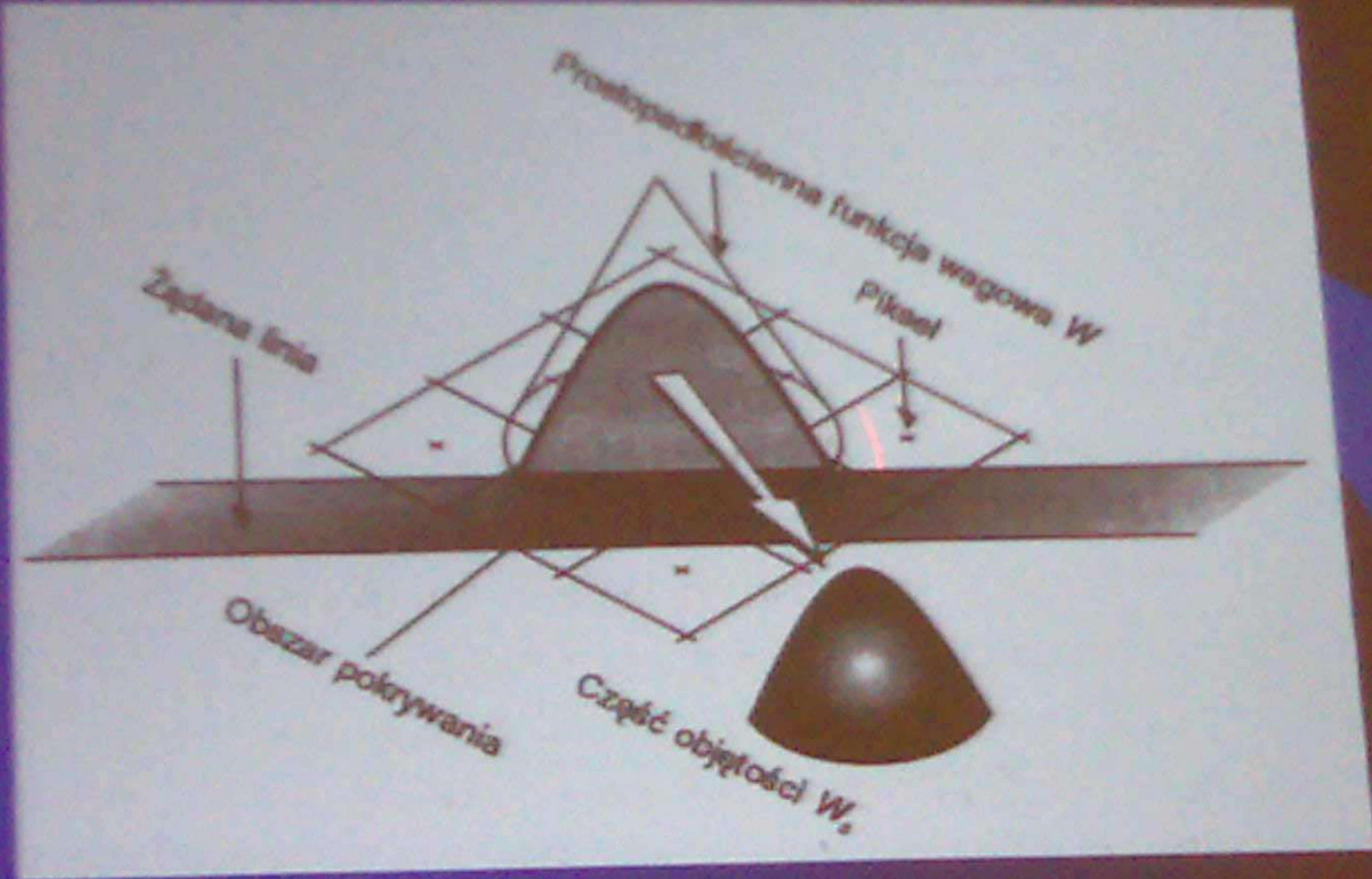


Właściwości bezwagowego próbkowania powierzchni

- jasność piksela zależy od odległości odcinka od środka piksela
  - im odcinek jest dalej, tym jego wpływ na jasność jest mniejszy
- odcinek nie wpływa na jasność piksela jeśli go nie przecina
- takie samo pole wnosi równą jasność
  - np. poziome odcinki o grubości ułamkowej wyglądają tak samo

Filtr stożkowy

- waga zależy liniowo od odległości od środka obszaru

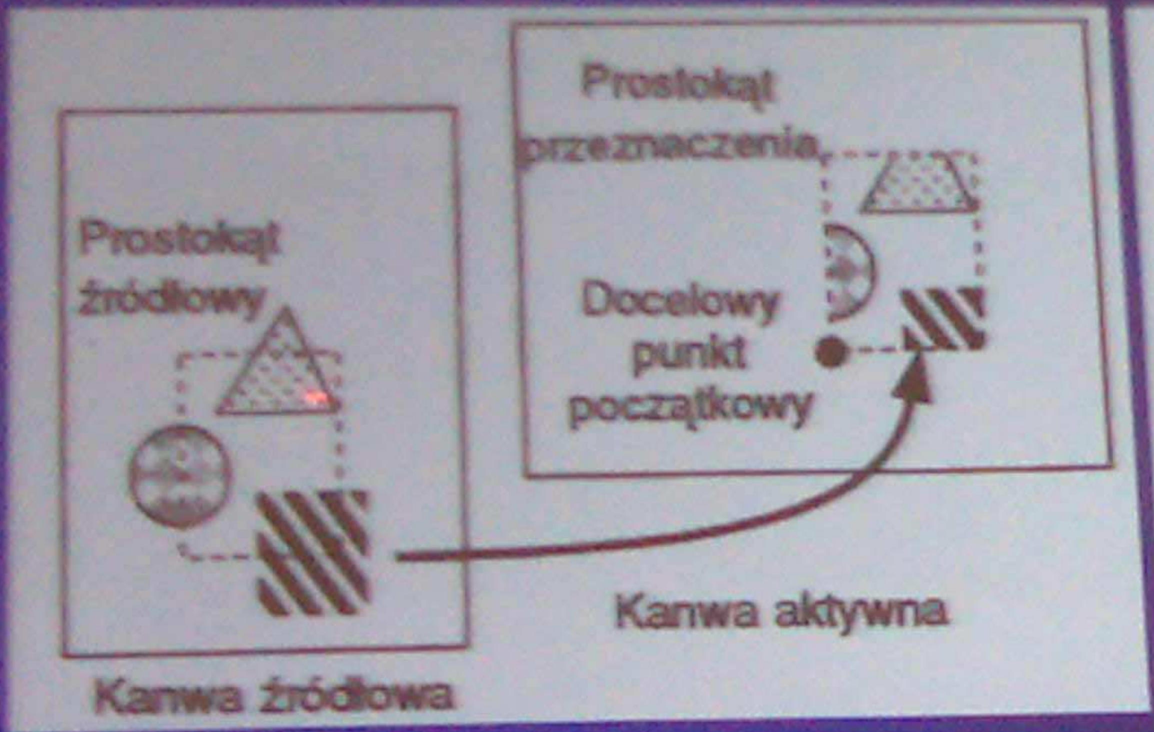


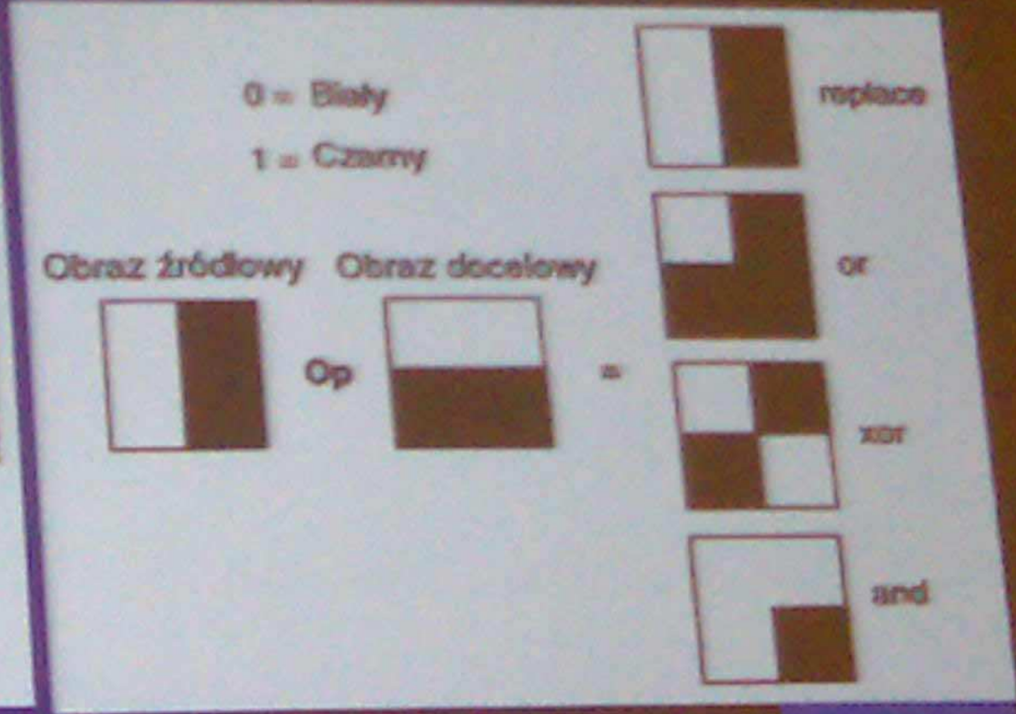
Filtr stozkowy dla kołowego piksela o średnicy dwóch skoków siatki

Mapy bitowe

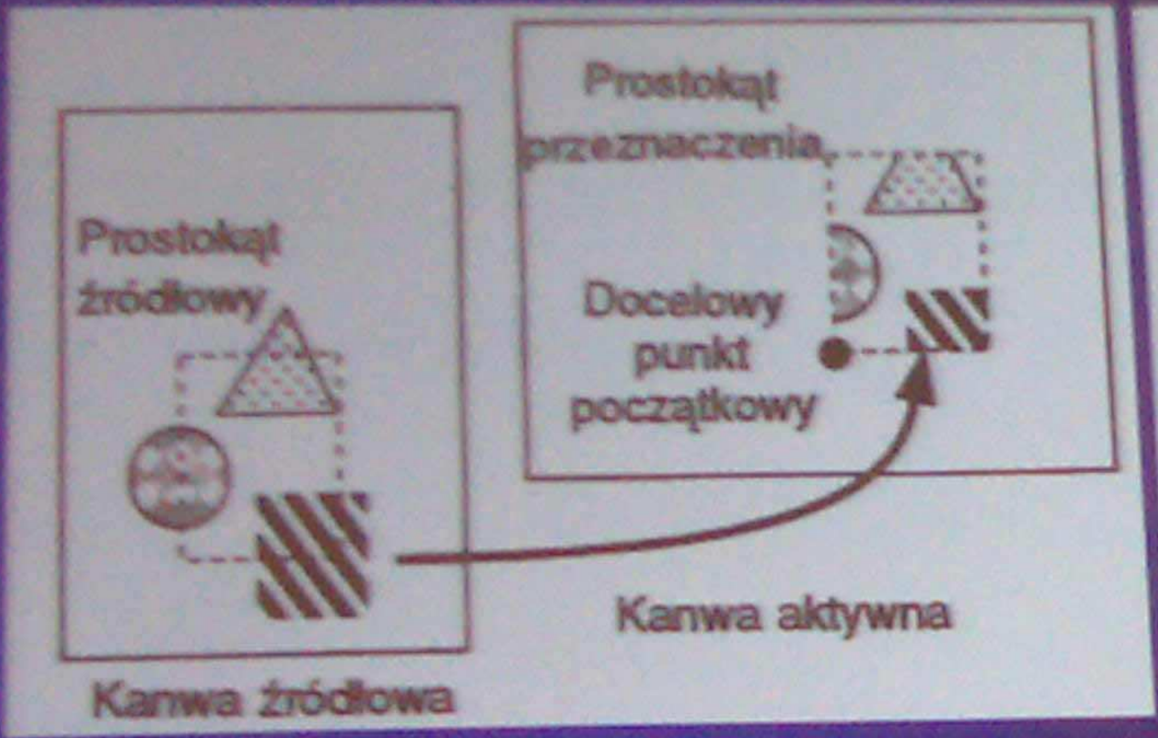
- Kanwa
- Układy współrzędnych
- Atrybuty kanwy (kontekst kanwy)
  - pisak (kolor, kształt)
  - wypełnienie
- Kopiowanie
- Maski
- Przyslanianie
- Przesuwanie/Skalowanie

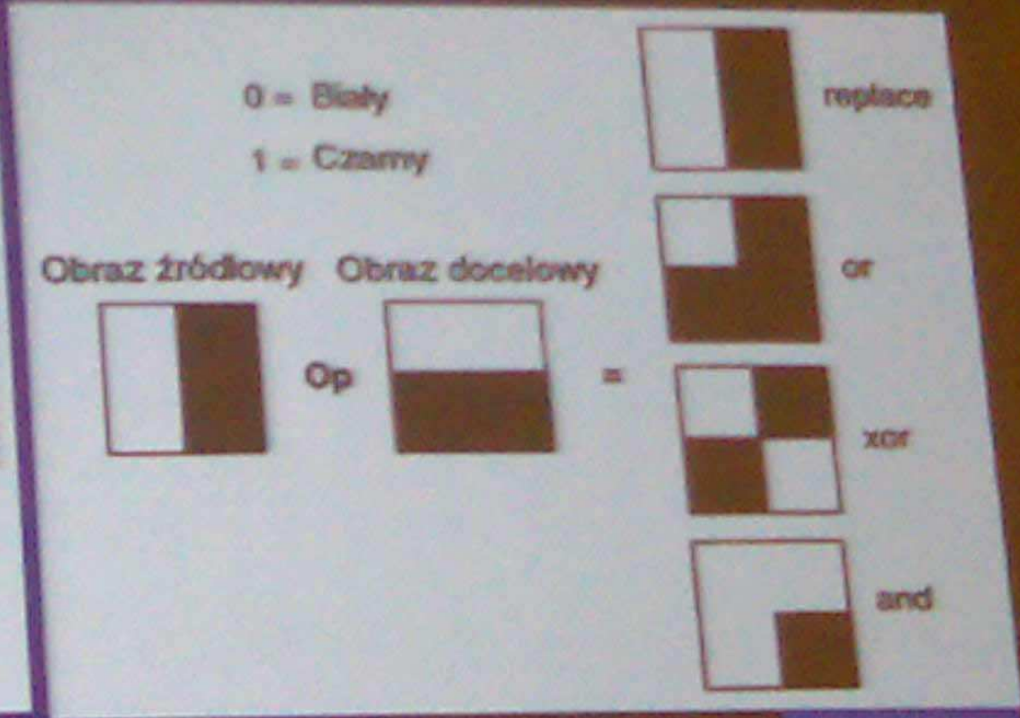
Kopiowanie map bitowych





Kopiowanie map bitowych





Rodzaje map bitowych

- 1- bitowe
- 8- bitowe odcienie szarości
- 8-bitowe z paletą kolorów (indeksowane)
- 24-bitowe RGB, Lab
- 32-bitowy CMYK, RGBA
- do 12, 14, 16 bitów/kanal

Wypełnianie wielokątów

- Wyznaczenie współrzędnych piksela do wypełnienia
- Ustalić czy i czym należy wypełnić piksel
  - Wypełnianie stałym kolorem
  - Wypełnianie tonalne
  - Wypełnianie wzorami

Wypełnianie rysunków rastrowych

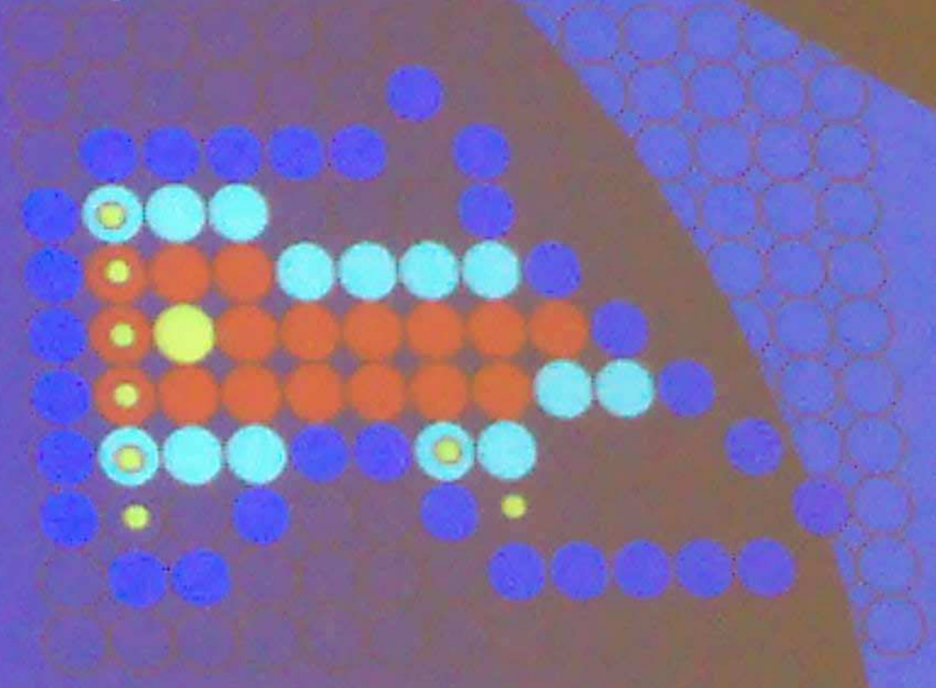
- kolor obrysu
- kolor (wzór) wypełniania

Segment to poziomy odcinek pikseli wewnątrz wielokąta

Algorytmy wypełnianie wielokątów

Rysunki rastrowe

- Wypełnianie rekurencyjne
  np. metoda z czterema sąsiadami
- Wypełnianie liniami poziomymi
  np. Algorytm Smitha



Rysunki wektorowe

- Wypełnianie prostokąta

```
for(y = ymin;   y <= ymax; y++)
    for(x = xmin;x <= xmax; x++)  /* wypełnianie
      segmentu */
       WritrePixel(x,y, color);
```

Problemy

- Wielokrotne zapisywania tego samego piksela

- Wypełnianie przez przeglądanie liniami poziomymi

Rysunki wektorowe

- Wypełnianie prostokąta

```
for(y = ymin;  y <= ymax; y++)
   for(x = xmin;x <= xmax; x++) /* wypełnianie
     segmentu */
       WritrePixel(x,y, color);
```

  Problemy
  - Wielokrotne zapisywania tego samego piksela

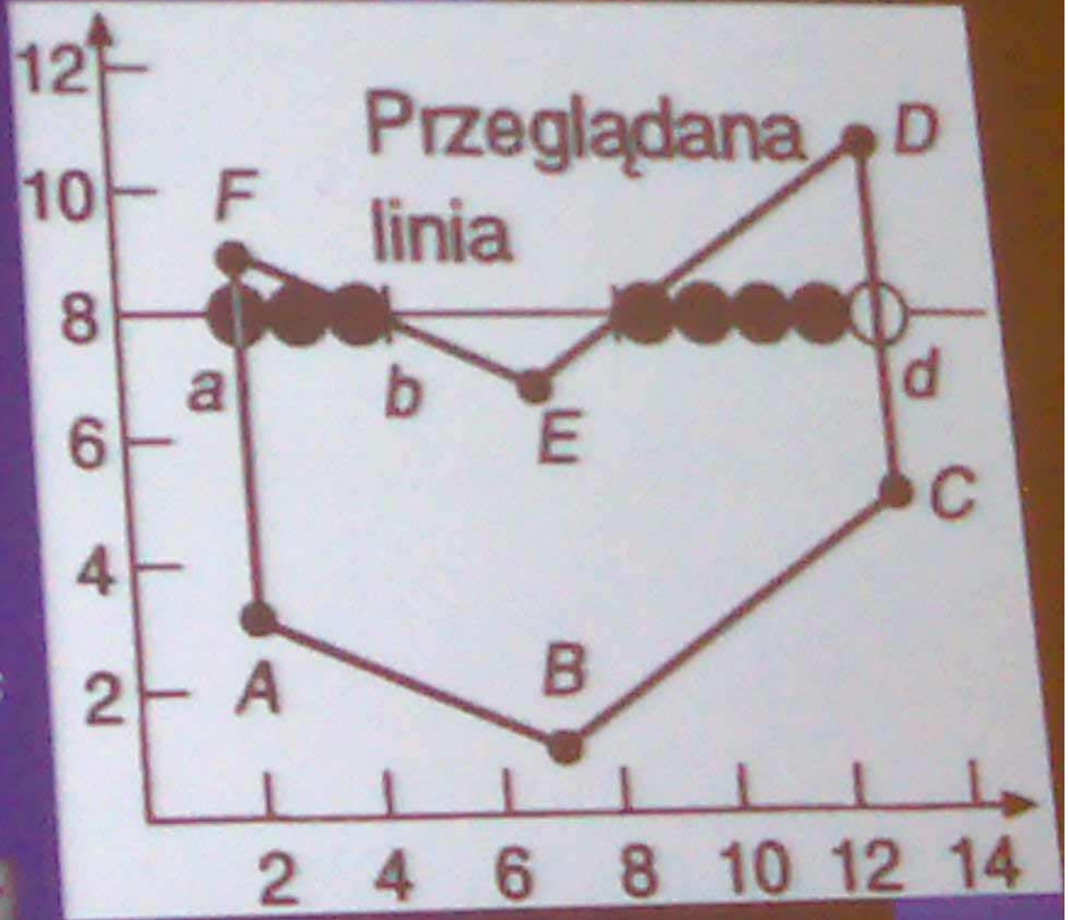
- Wypełnianie przez przeglądanie liniami poziomymi

Wypełnianie wielokątów z przeglądaniem linii

Kroki algorytmu:

for ($y = y_{min}$; $y <= y_{max}$; y++)

- wyznaczenie przecięć poziomej prostej y z krawędziami wielokąta
- posortować te punkty według współrzędnej x
- wypełnić segmenty (piksele leżące wewnątrz wielokąta) korzystając z reguły parzystości



Problemy:

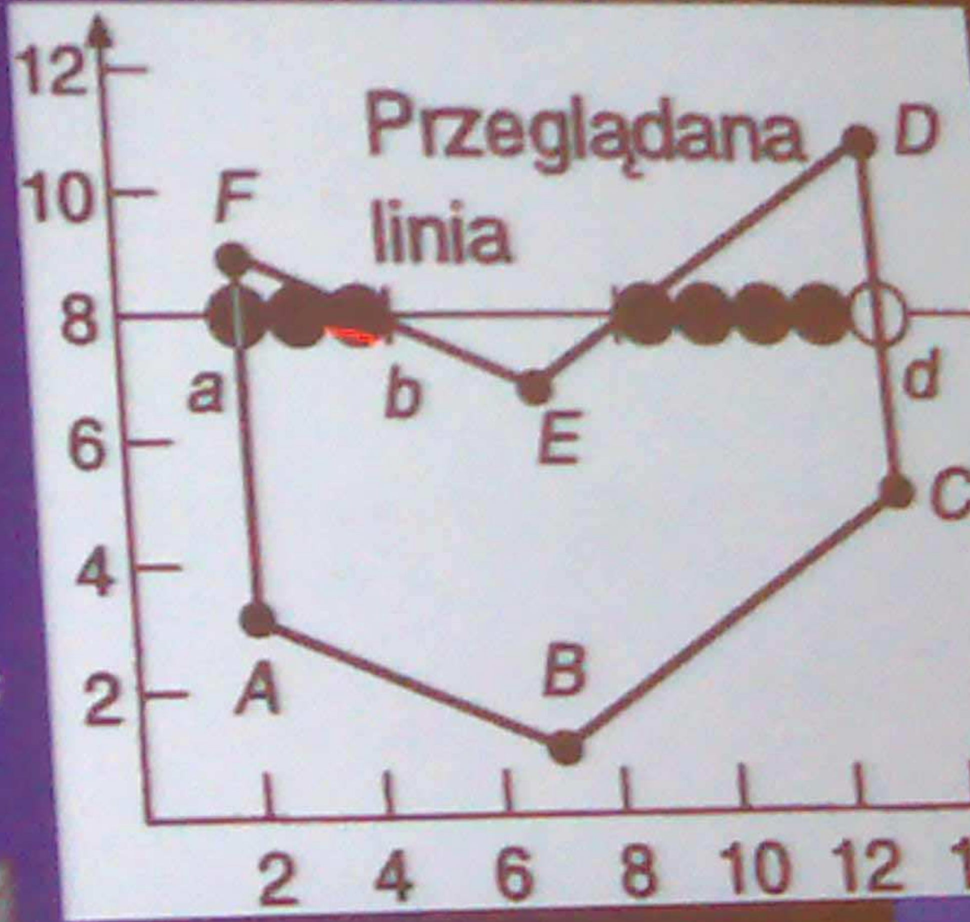
- wartości ułamkowe przecięć
- krawędzie poziome
- wspólny wierzchołek

Wypełnianie wielokątów z przeglądaniem linii

Kroki algorytmu:

for ($y = y_{min}$; $y <= y_{max}$; y++)

- wyznaczenie przecięć poziomej prostej y z krawędziami wielokąta
- posortować te punkty według współrzędnej x
- wypełnić segmenty (piksele leżące wewnątrz wielokąta) korzystając z reguły parzystości



Problemy:

- wartości ułamkowe przecięć
- krawędzie poziome
- wspólny wierzchołek

Wypełnianie bez wielokrotnej konwersji

- konwersja w niewidocznej pamięci (budowanie masek wnętrza obiektu)
- wykorzystując tę maskę wpisujemy wzór prostokątny w obszar prostokątny

1. Konwersja wielokąta – tworzenie maski bitowej
2. Zerowanie tła z wykorzystaniem maski
3. Przygotowanie mapy obiektu ze wzorem
4. Zapis mapy obiektu

Algorytmy obcinania

- Obcinanie w kanwach (operacjami Copy)
- Obcinanie w trakcie konwersji
  - obcinanie wartości krańcowych
  - warunkowy zapis piksela
  - próbkowanie co kilka pikseli
  - obcinanie złożonych prymitywów

  Efektywne gdy mało pikseli leży poza obszarem obcinania
- Obcinanie analityczne

Obcinanie analityczne

- operacje zmiennoprzecinkowe
- łatwe dla odcinków i wielokątów

<u>obcinanie prostokątem</u>	<u>daje</u>
• prostokąta	prostokąt
• wielokąta wypukłego	wielokąt wypukły
• dowolnego wielokąta	kilka wielokątów
• okręgu	łuki

Obcinanie analityczne

- operacje zmiennoprzecinkowe
- łatwe dla odcinków i wielokątów

<u>obcinanie prostokątem</u>	<u>daje</u>
• prostokąta	prostokąt
• wielokąta wypukłego	wielokąt wypukły
• dowolnego wielokąta	kilka wielokątów
• okręgu	łuki

Obcinanie analityczne

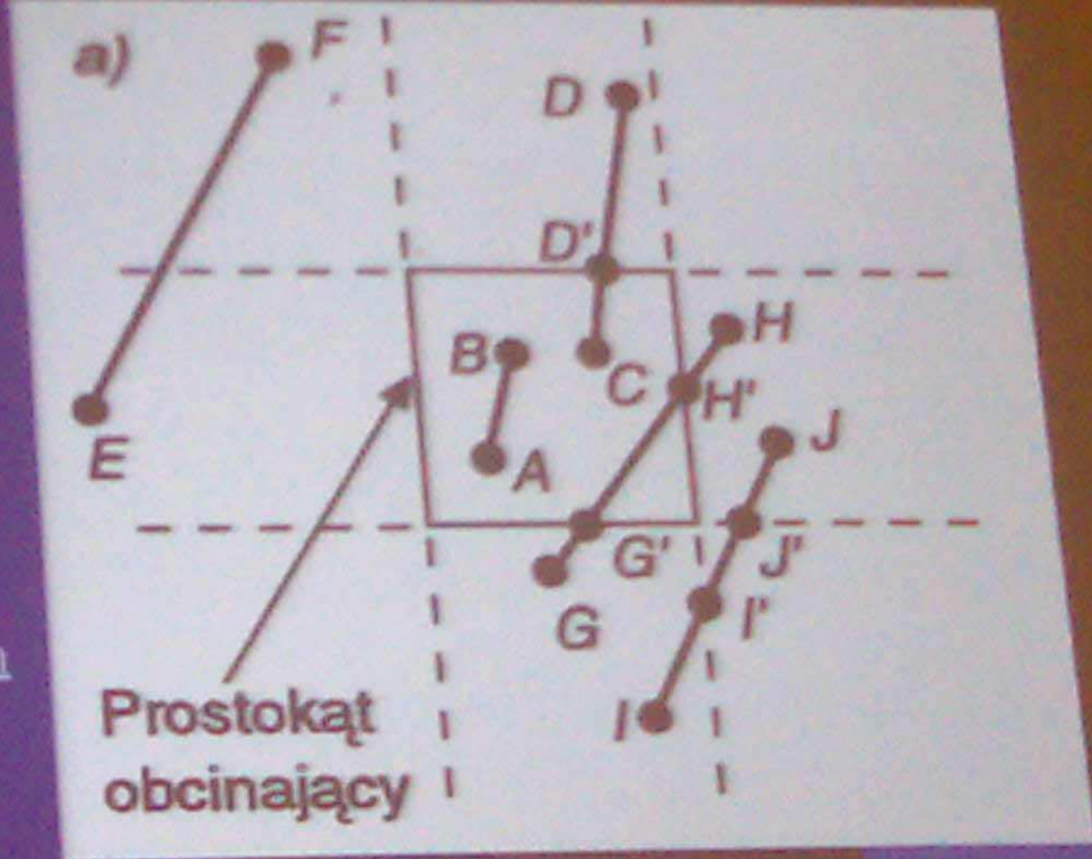
- operacje zmiennoprzecinkowe
- łatwe dla odcinków i wielokątów
- <u>obcinanie prostokątem</u> — <u>daje</u>
  - prostokąta — prostokąt
  - wielokąta wypukłego — wielokąt wypukły
  - dowolnego wielokąta — kilka wielokątów
  - okręgu — łuki

Obcinanie odcinków

- Obcinanie punktów

  $x_{min} \le x \le x_{max}$

  $y_{min} \le y \le y_{max}$
- Obcinanie odcinków z rozwiązywaniem układu równań



  - analiza punktów końcowych odcinka
  - obliczanie współrzędnych punktu przecięcia odpowiednich prostych
  - sprawdzenie czy punkt przecięcia leży na odcinku
  - sprawdzenie czy punkt przecięcia leży na krawędzi obcinającej

Przykład 1 - Obcinanie z rozwiązywaniem układu równań

Czy odcinek (0, 0) - (100, 200), będzie narysowany na monitorze o rozdzielczości 1024x800) ?

Parametryczne równanie odcinka

$$\mathbf{Q(t) = (1\text{-}t)\, P_p + t\, P_k = P_p + t\, (P_k - P_p)}$$

$x = x_0 + t(x_1 - x_0)$ $\quad (x_0, y_0) = (0, 0)$

$y = y_0 + t(y_1 - y_0)$ $\quad (x_1, y_1) = (100, 200)$

$$\left.\begin{array}{l} x = 100t \\ y = 200t \end{array}\right\} \; t \in \langle 0, 1 \rangle$$

Przecięcie z linią x = 1024

$x = 100t = 1024$

$t = 10.24 \quad t \notin \langle 0, 1 \rangle$

nie zostanie obcięty

Przecięcie z linią y = 800

$y = 200t = 800$

$t = 4 \quad t \notin \langle 0, 1 \rangle$

nie zostanie obcięty

Przykład 2 - Obcinanie z rozwiązywaniem układu równań

Czy odcinek opisany równaniem parametrycznym

$$\left.\begin{array}{l} x = 200t - 100 \\ y = 100t \end{array}\right\} t \in <0, 1>$$

zostanie obcięty prostą $x = 0$?

Przecięcie z linią $x = 0$

$x = 200t - 100 = 0$

$t = 100/200 = ½ \quad t \in <0, 1>$

Zostanie obcięty

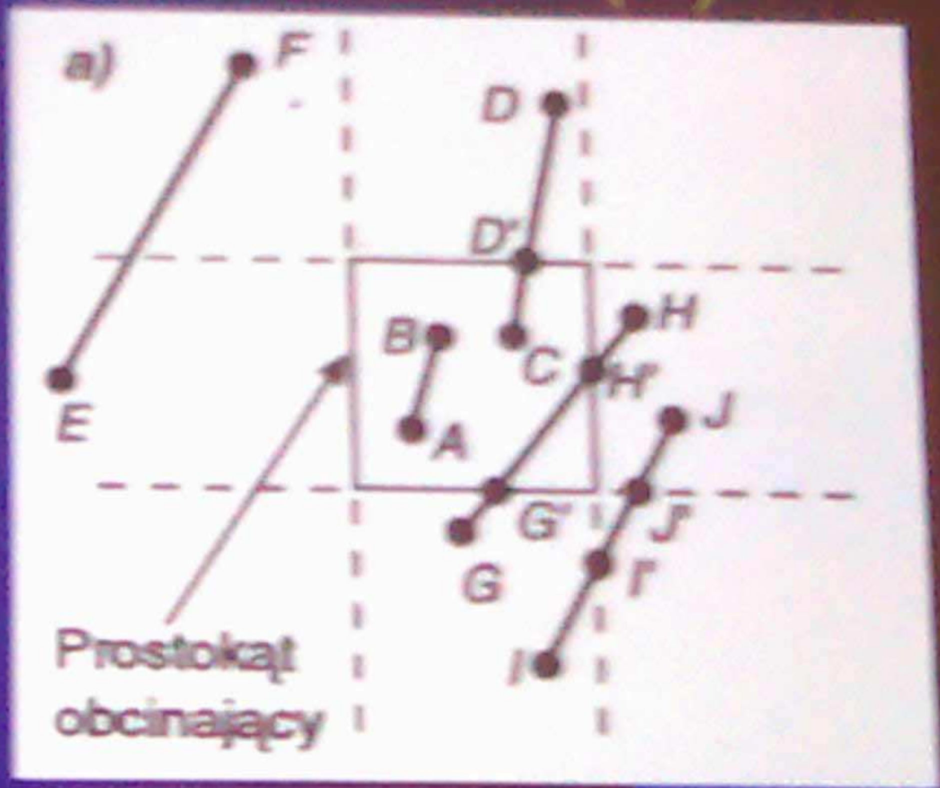
Współrzędne punktu przecięcia :

$x = 0$

$y = y(½) = 100 * ½ = 50$

Algorytm Cohena-Sutherlanda (1)

- Podstawowe kroki algorytmu
  - Akceptacja odcinka (AB)
  - Odrzucenie odcinka (EF)
  - Podział odcinka krawędzią obcinającą na dwie części, tak aby jedną można było odrzucić (iteracyjne obcinanie odcinka)



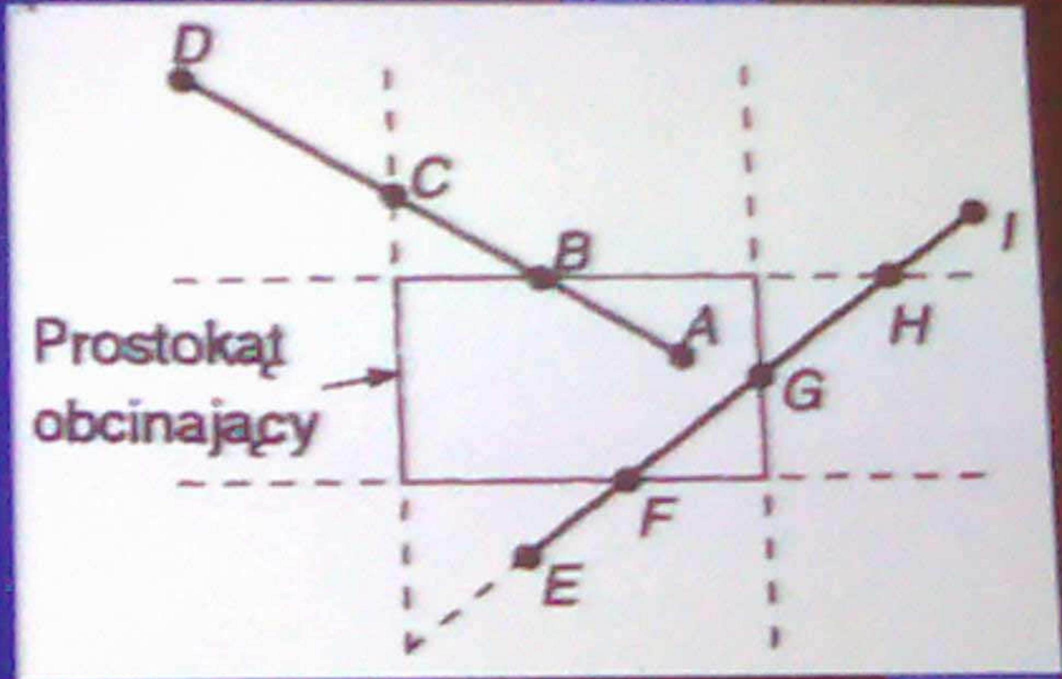
- Algorytm bardzo efektywny gdy
  - Duży prostokąt obcinający (obejmuje większość pola wyświetlania)
  - Mały prostokąt obcinający

Algorytm Cohena-Sutherlanda (3)

Podział odcinka

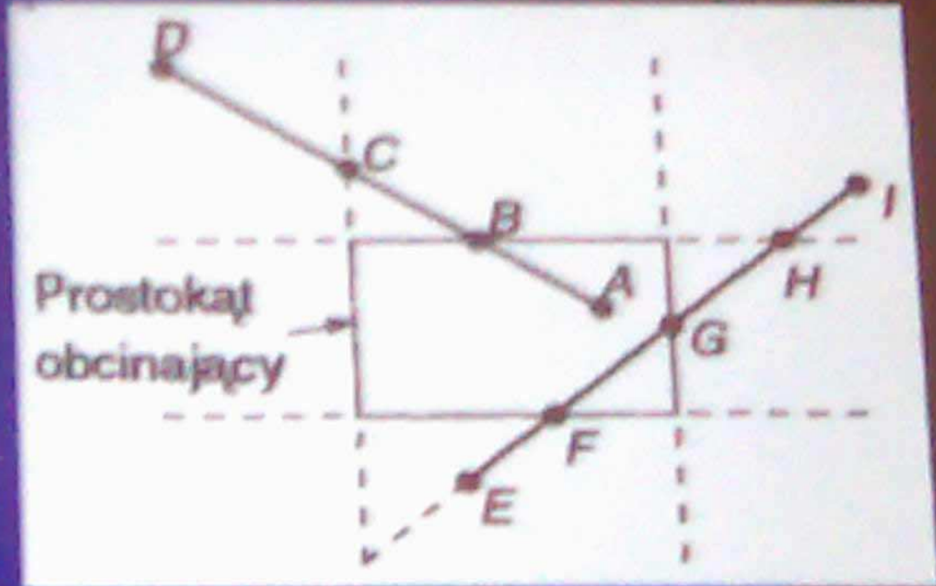
Podziału dokonujemy wykorzystując krawędź, którą odcinek przecina. W algorytmie musimy korzystać z tego samego porządku testowania np. (góra, dół, prawa, lewa)

Bity kodu odpowiadają przecinanym krawędziom



Przykład 1

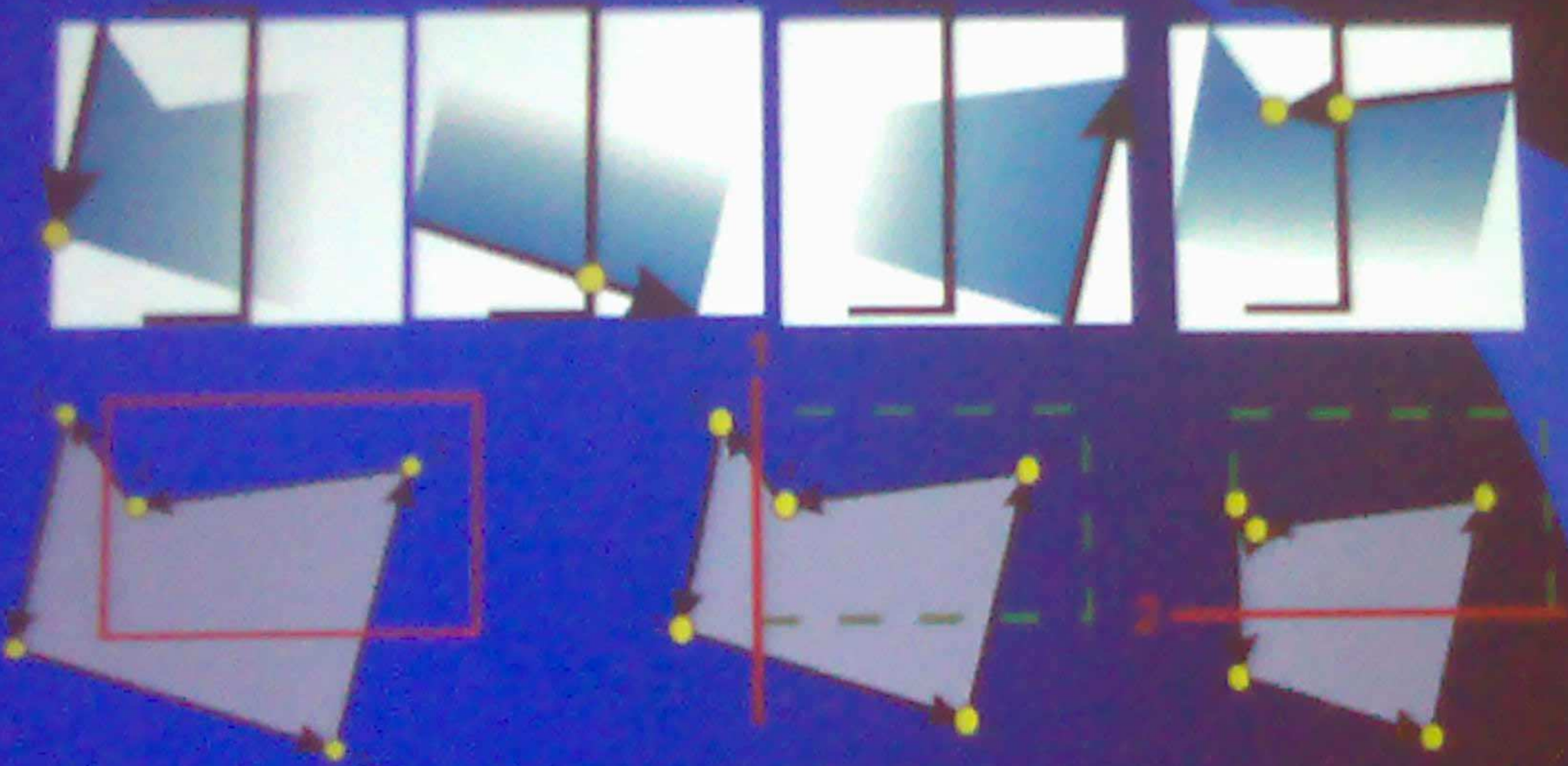
<u>Odcinek AD</u> (A - 0000, D - 1001)



- A OR D = 1001 <> 0
  Nie można zaakceptować odcinka
- A AND D = 0
  Nie można odrzucić odcinka
- Wybieramy punkt D (jako punkt zewnętrzny)
  Odcinek przecina krawędź górną i lewą (kod 1001)
- Obcinamy krawędzią górną i otrzymujemy AB
  Porządek testowania powoduje, że najpierw obcinamy krawędzią górną
- Analizujemy odcinek AB

Algorytm Sutherlanda-Hodgmana

Obcinanie wielokąta prostą obcinającą (obcinanie kolejnymi krawędziami wielokąta)



Krzywe parametryczne

- $x = f_x(t)$; $y = f_y(t)$
- funkcje liniowe $x = 20t - 5$
  $y = 10t + 100$

t	0	1	2	3	4
x	5	25	45	65	85
y	100	110	120	130	140

- funkcje nieliniowe $x = 2t^2 + 4$
  $y = t^2 + 5$

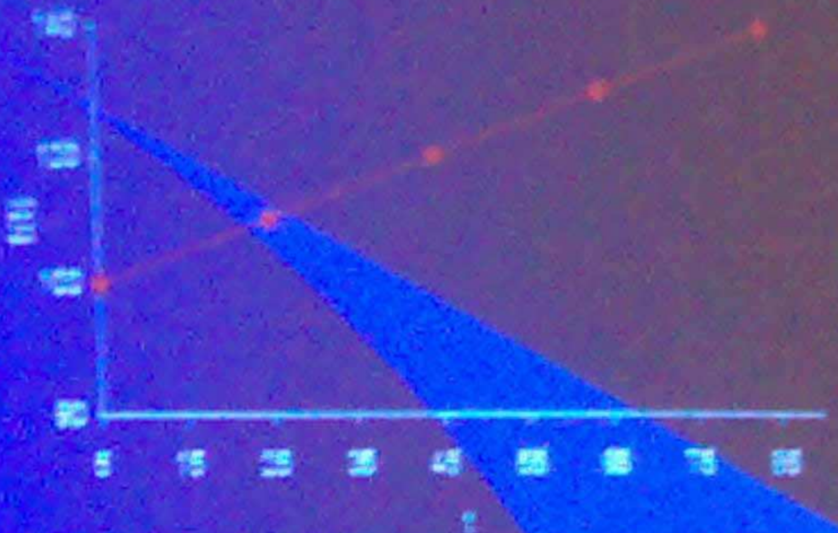
t	0	1	2	3	4
x	4	6	12	22	36
y	5	6	9	14	21

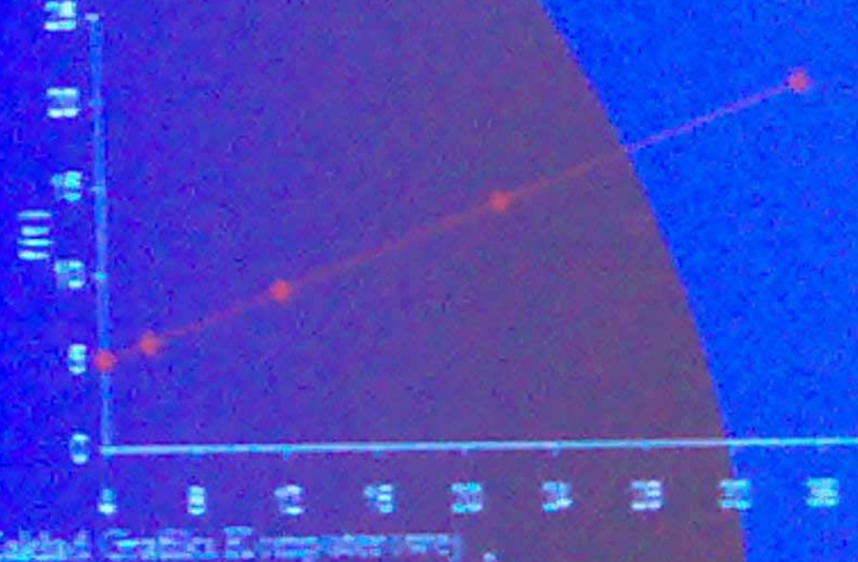
jeśli $t' \in \langle 0,1 \rangle$ $t' = 1/4 * t \Rightarrow t = 4t'$

$x' = 2(4t')^2 + 4 = 32t'^2 + 4$

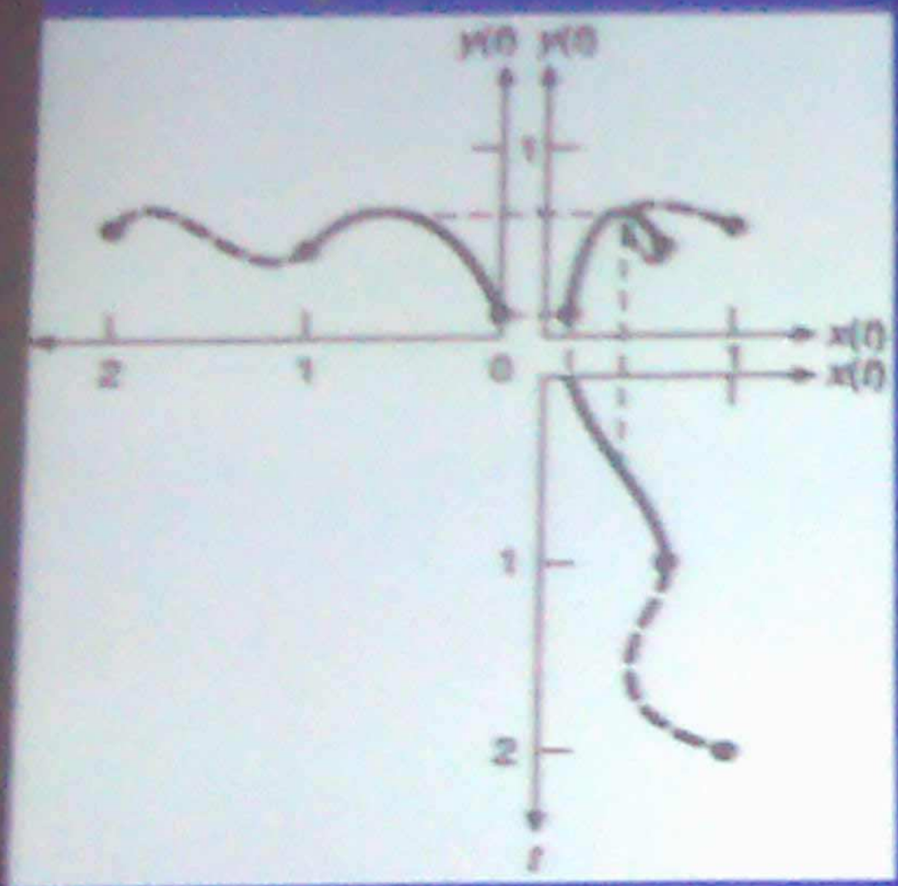
$y' = (4t')^2 + 5 = 16t'^2 + 5$

$t' = (0, 1/4, 1/2, 3/4, 1)$





Parametryczne krzywe trzeciego stopnia



- $f(t) = at^3 + bt^2 + ct^1 + d$

$$x(t) = a_x t^3 + b_x t^2 + c_x t^1 + d_x$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t^1 + d_y$$
$$z(t) = a_z t^3 + b_z t^2 + c_z t^1 + d_z$$

$Q(t) = [x(t), y(t), z(t)]$

- Reguła Hornera

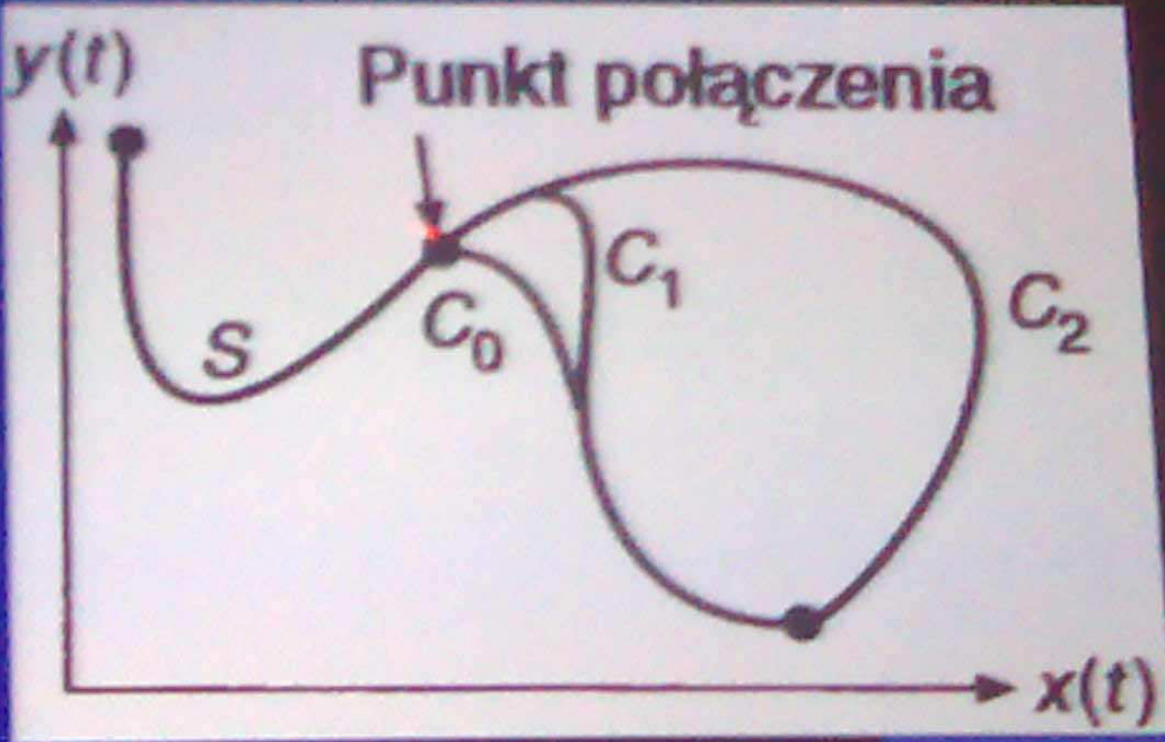
$f(t) = ((at + b)t + c)t + d$

- Wektory styczne w punkcie

$Q'(t) = [d/dt\ x(t), d/dt\ y(t), d/dt\ z(t)]$

$f'(t) = 3at^2 + 2bt + c$

Ciągłość



Ciągłość geometryczna

- G^0 połączenie segmentów
- G^1 kierunki wektorów stycznych (nachylenie segmentów) równe w punkcie połączenia

Ciągłość parametryczna C^n

pochodna jest parametrycznym wektorem stycznym

- C^1 kierunki i długości wektorów stycznych (pierwsza pochodna) są równe
- C^2 kierunki i długości wektorów drugiej pochodnej są równe

Krzywa jako kombinacja liniowa punktów (1)

$P(x,y) = P\,(\,f_x(t),\ f_y(t)\,)$ $\quad t \in \langle 0,1 \rangle$

$f(t) = at + b$

- punkt początkowy $P_p(x_p,\ y_p)$

$\underline{t = 0}$:

$P_p = (x_p = x(0),\ y_p = y(0))$;

- punkt końcowy $P_k(x_k,\ y_k)$

$\underline{t = 1}$:

$P_k = (x_k = x(1),\ y_k = y(1))$;

$$\begin{cases} x = a_x t + b_x \\ y = a_y t + b_y \end{cases}$$

$$\begin{cases} x_p = x(0) = b_x \\ x_k = x(1) = a_x + b_x = a_x + x_p \\ \\ y_p = y(0) = b_y \\ y_k = x(1) = a_y + b_y = a_y + y_p \end{cases}$$

Krzywa jako kombinacja liniowa punktów (2)

Dla punktów $P_p(x_p, y_p)$ i $P_k(x_k, y_k)$ wyznaczamy współczynniki a_x, b_x, a_y, b_y

$b_x = x_p \qquad b_y = y_p$

$a_x = x_k - x_p \qquad a_y = y_k - y_p$

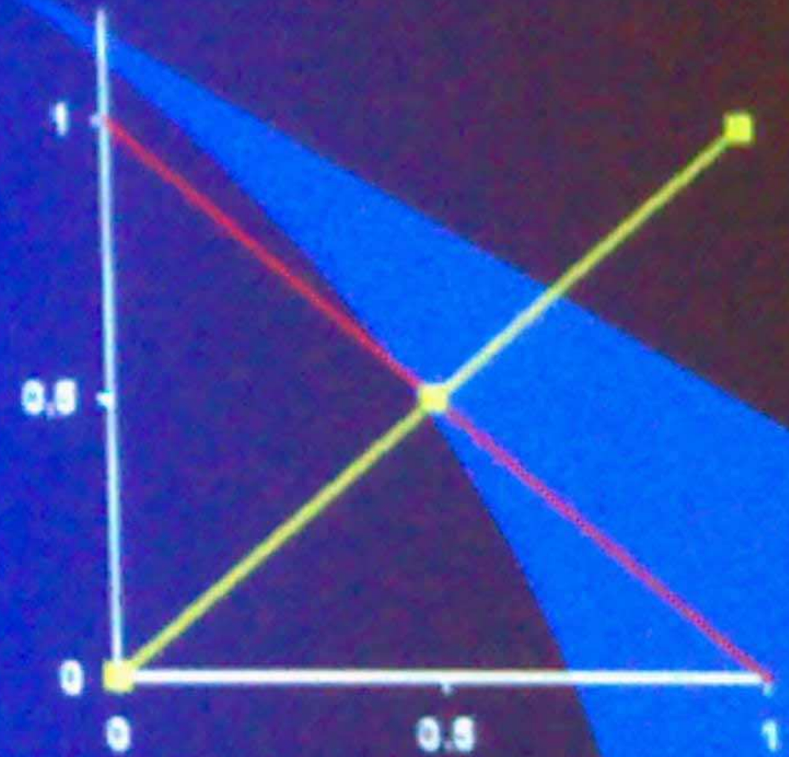
$x = a_x t + b_x = (x_k - x_p)\, t + x_p \qquad t \in \langle 0,1 \rangle$

$y = a_y t + b_y = (y_k - y_p)\, t + y_p$

$$\begin{cases} x = x_p\,(1-t) + x_k\, t \\ y = y_p\,(1-t) + y_k\, t \end{cases}$$

$$Q(t) = (1-t)\, P_p + t\, P_k$$

$$Q(t) = \Sigma_i\, W_i(t)\, P_i$$



Krzywe Béziera

Określone przez punkty końcowe (P_1, P_4) i dwa punkty kontrolne (P_2, P_3).

$$Q(t) = \sum_{i=0}^{n} B_i^n(t)\, P_{i+1} \qquad t \in \langle 0, 1\rangle$$

$B_0^3 = (1-t)^3$ $\qquad$ $B_1^3 = 3t\,(1-t)^2$

$B_2^3 = 3t^2(1-t)$ $\qquad$ $B_3^3 = t^3$

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)\, P_3 + t^3 P_4$$

- $Q(0) = P_1$
- $Q(1) = P_4$
- $Q'(0) = 3(P_2 - P_1)$
- $Q'(1) = 3(P_4 - P_3)$