

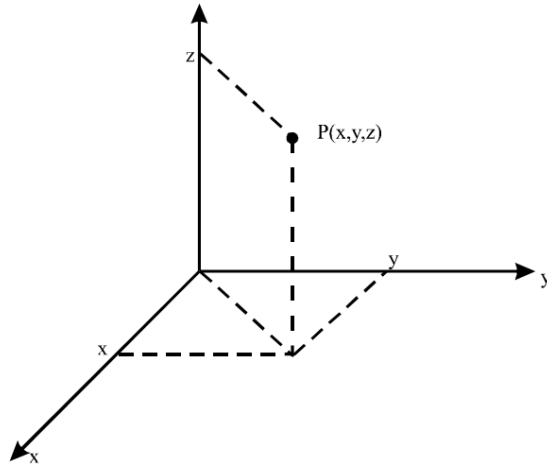
# GRAKO: PODSTAWY GRAFIKI 3W

---

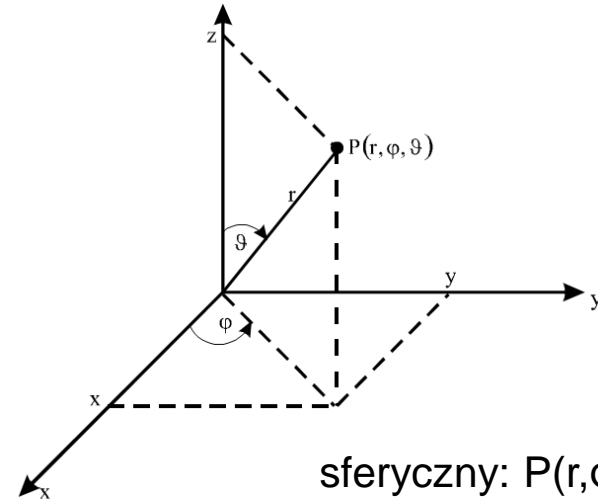
- Przekształcenia 3W
- Modelowanie
- Metody renderingu
- Przykłady



# Układy współrzędnych



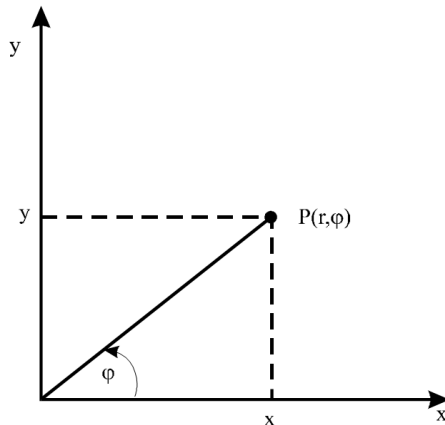
kartezjański:  $P(x, y, z)$



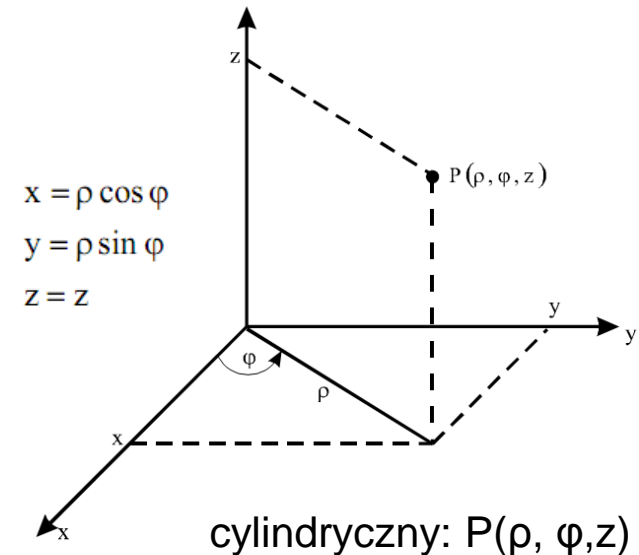
sferyczny:  $P(r, \varphi, \vartheta)$

$$\begin{aligned}x &= r \sin \vartheta \cos \varphi \\y &= r \sin \vartheta \sin \varphi \\z &= r \cos \vartheta\end{aligned}$$

$$\begin{aligned}x &= r \cos \varphi \\y &= r \sin \varphi \\r &= \sqrt{x^2 + y^2} \\ \varphi &= \arctan \frac{y}{x}\end{aligned}$$



biegunowy:  $P(r, \varphi)$

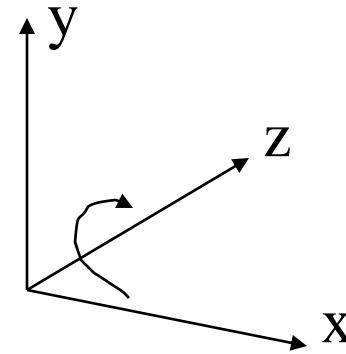
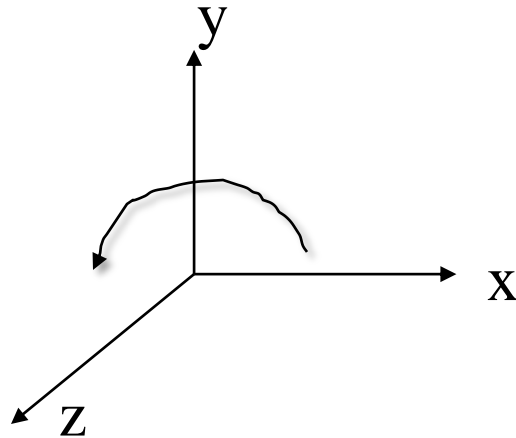


cylindryczny:  $P(\rho, \varphi, z)$

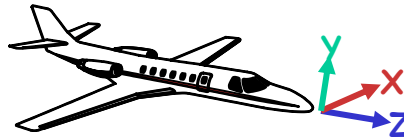
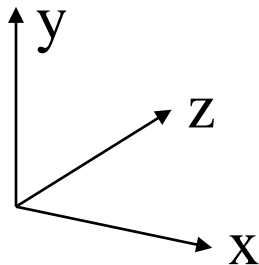
$$\begin{aligned}x &= \rho \cos \varphi \\y &= \rho \sin \varphi \\z &= z\end{aligned}$$

# Układy współrzędnych - rozszerzenia

- Lewoskrętny i prawoskrętny



- Globalny i lokalny (związany z obiektami)



# Transformacje w 3W

- Analogicznie jak w 2W z wykorzystaniem rachunku macierzowego (translacja, skalowanie itd...)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

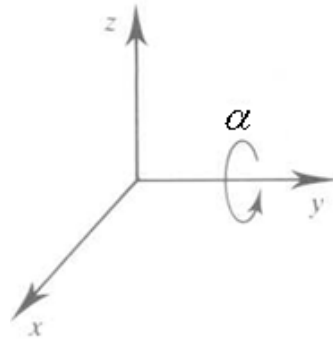
*przesunięcie*                      *skalowanie*

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

*obrót względem Ox*

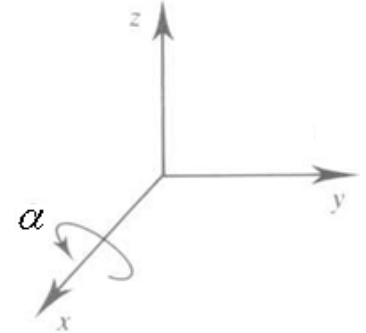
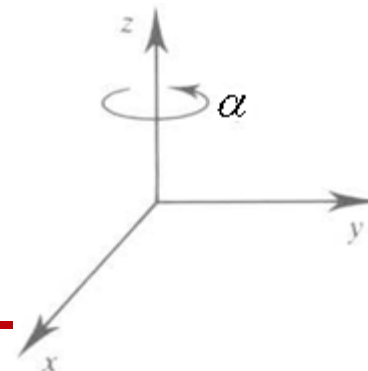
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

*obrót względem Oy*



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

*obrót względem Oz*

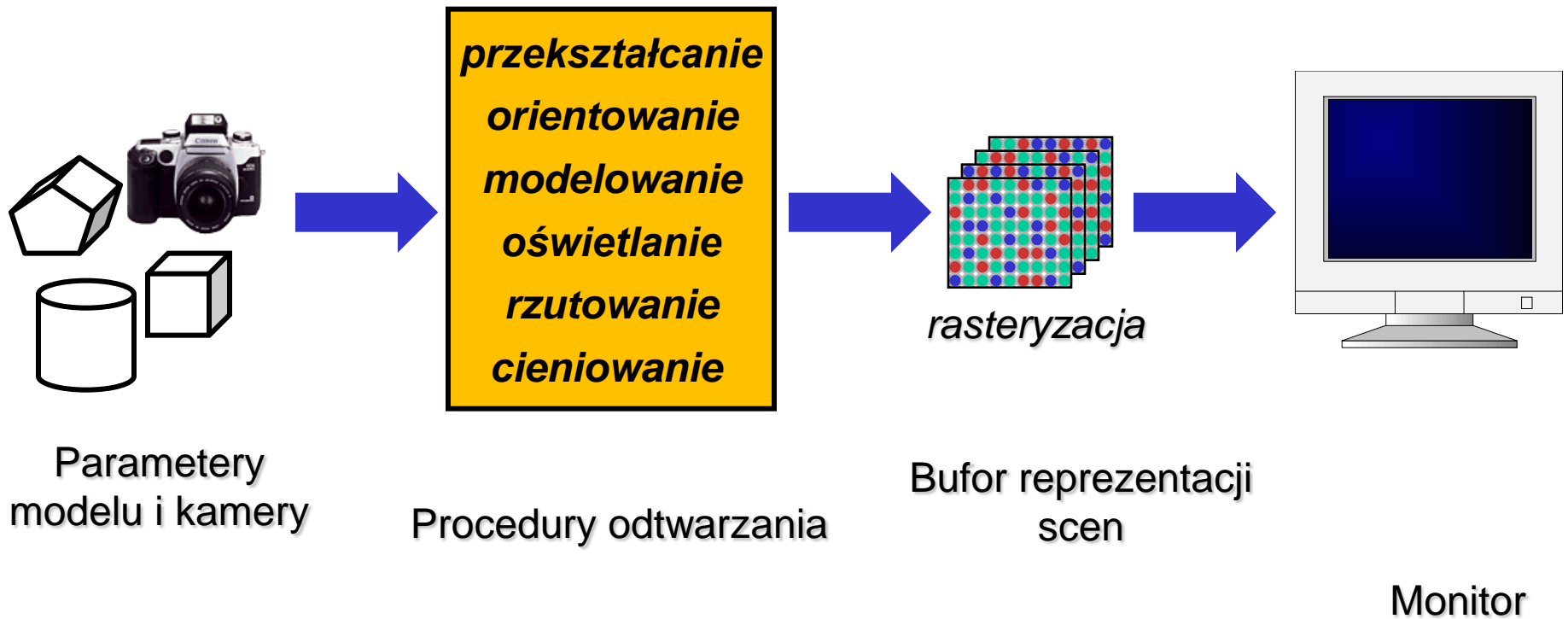


# Podstawowe koncepcje

---

- Ogólna metoda: generowanie płaszczyznowych (2W) obrazów na podstawie komputerowej reprezentacji świata 3W - modeli przestrzennych
  - Realizacja:
    - *modelowanie* (kształt, właściwości, relacje, treści i semantyka): tworzenie i reprezentacja geometrii obiektów 3W (rzeczywistych, przybliżonych)
    - *odtworzenie* (światło, perspektywa, kamera, realia): generacja 2W obrazów obiektów i scen przestrzennych (na podstawie modeli 3W)
    - *animacja*: (ruch) opis zmian obiektów w czasie (klatki dyskretnych chwil czasowych) z dopasowanymi formami odtwarzania
-

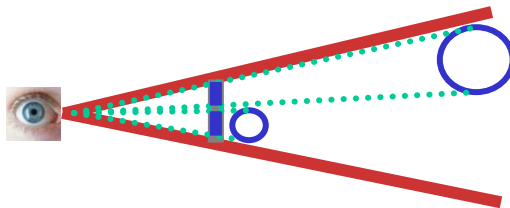
# Odtwarzanie (rysowanie-rendering) scen 3W



# Rendering wykorzystuje przekształcenia względem różnych układów współrzędnych

---

- Przekształcenia - do modelowania obiektów
- Przekształcenia - do obserwacji (ruch kamery)
- Przekształcenia - do rzutowania (zmiana typu kamery)



# Przekształcenia w odtwarzaniu 3W



Geometria  
obiektu

*Modelowanie*

- Obiekty i elementy sceny opisane w 3W układzie współrzędnych rzeczywistych

Obliczenia  
światła

- Oświetlanie
- Cieniowanie zgodne z modelem oświetlenia

*Obserwacja*

- Elementy sceny w 3W układzie współrzędnych obserwatora lub kamery (ustawienie sceny względem kamery)

Obcinanie

- Określenie zakresu sceny, która mieści się w widoku

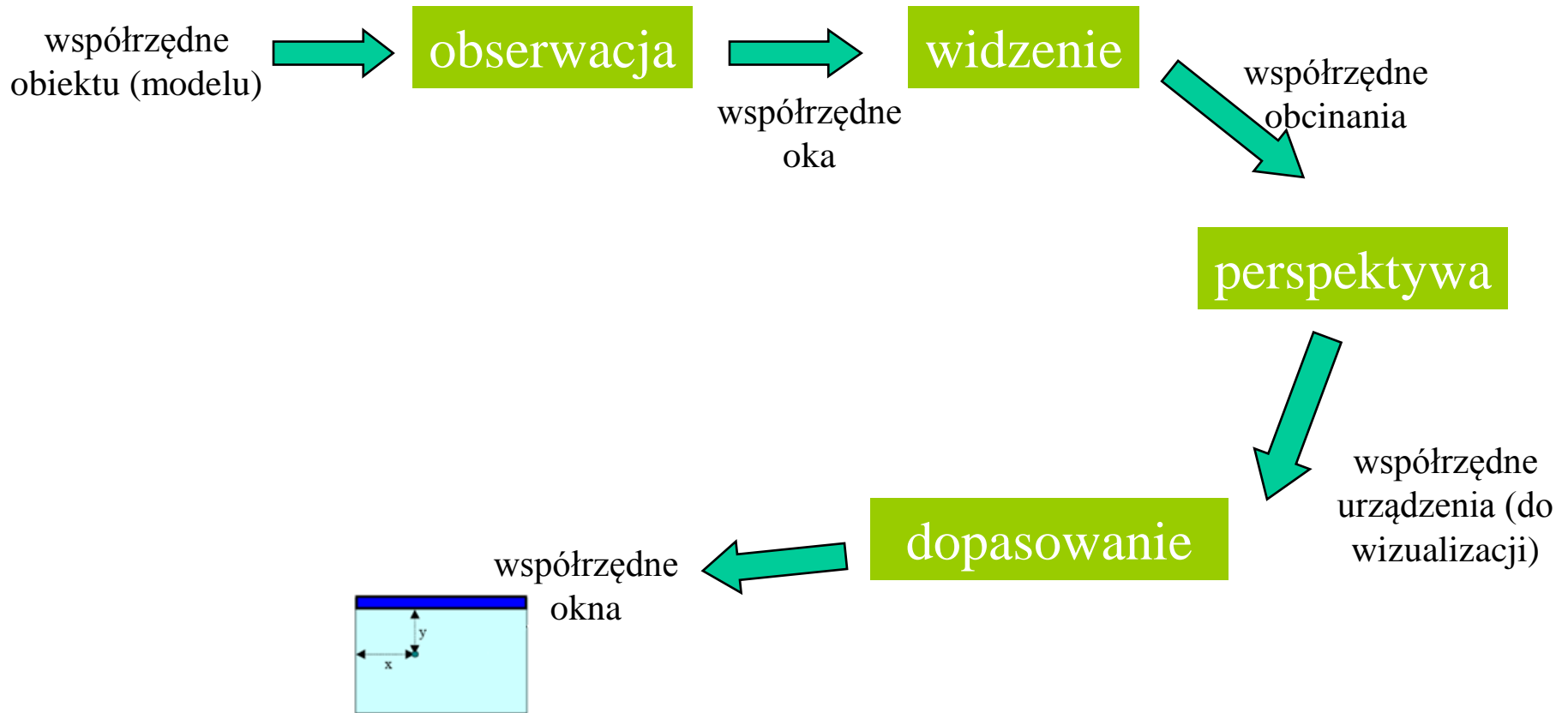
*Rzutowanie*

- Obraz sceny w 2W współrzędnych ekranu (widoku)

transformacje



# Przekształcenia układów współrzędnych



---

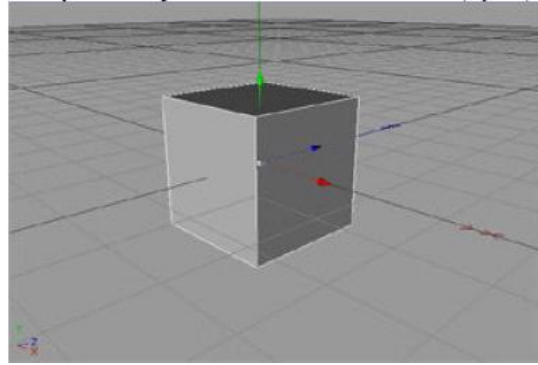
# MODELE OBIEKTÓW

---

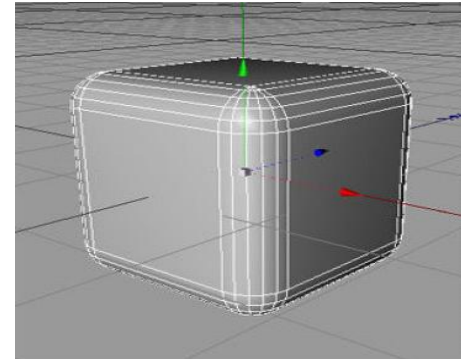
# Przykład tworzenia modelu na podstawie zestawu prymitywów



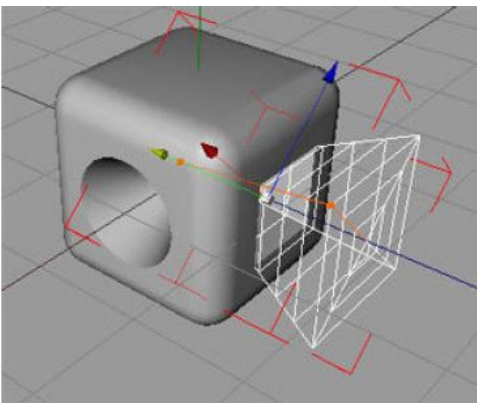
Modelowana scena naturalna



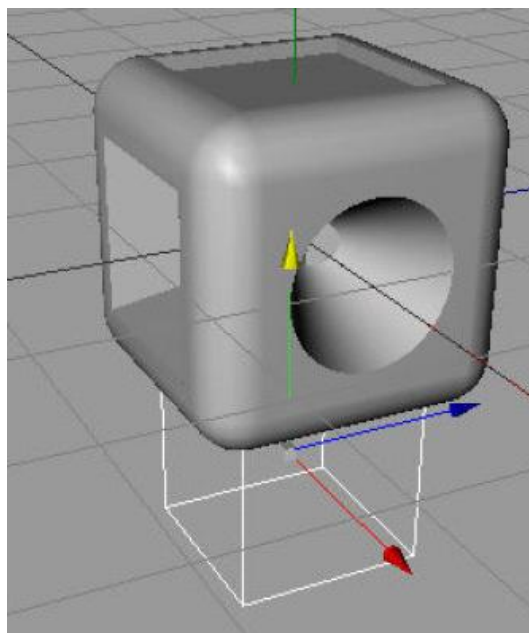
Bierzemy sześcian



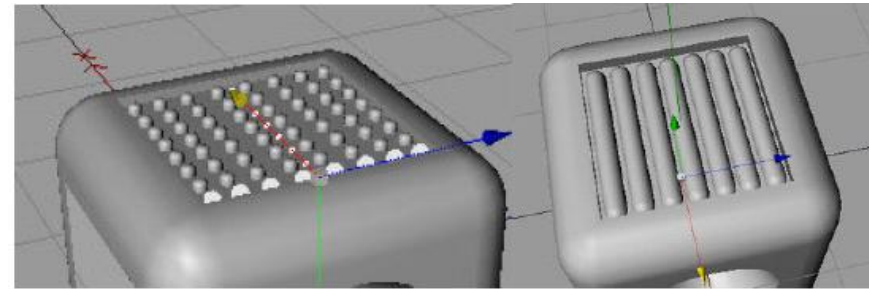
Filtrem zaokrąglamy rogi



Wcięcie koliste i kwadratowe – przecięcie z pomocniczymi figurami



Wgłębienia

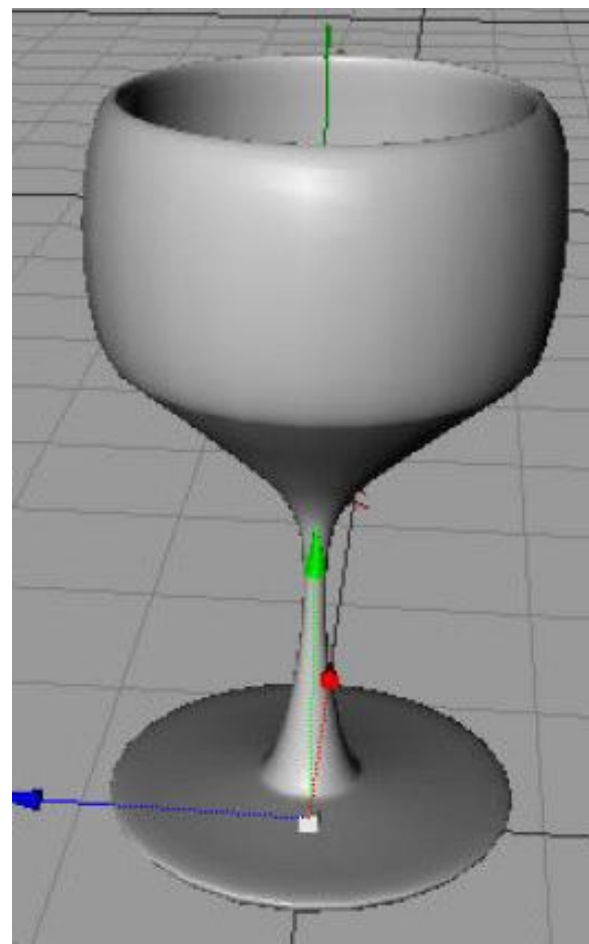


Mniejsze figury jako dodatki itd..

# Przykład modelowania obrysem

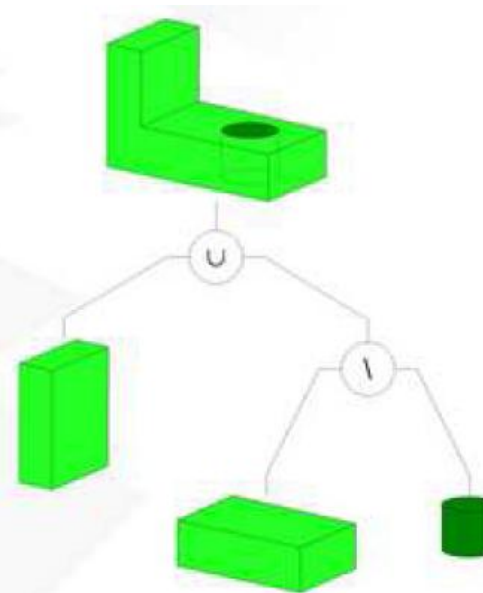
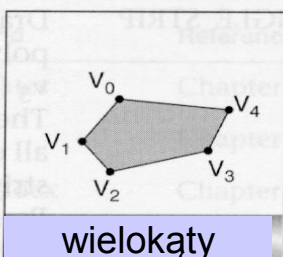
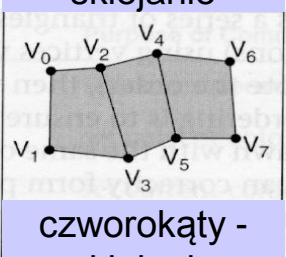
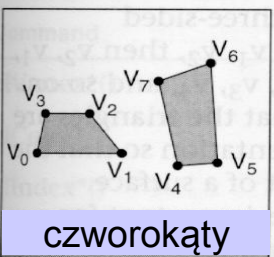
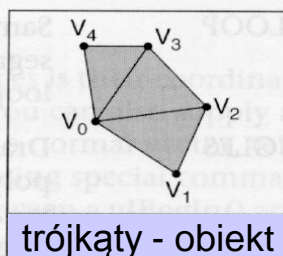
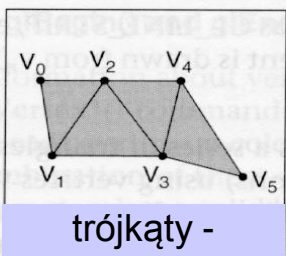
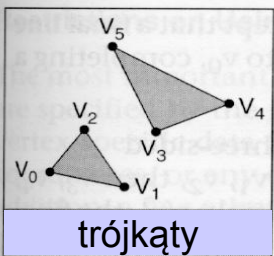
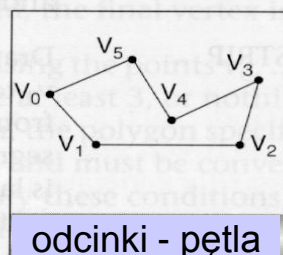
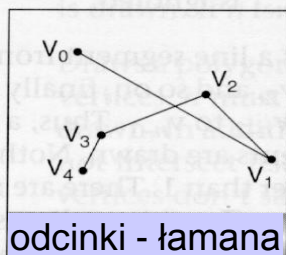
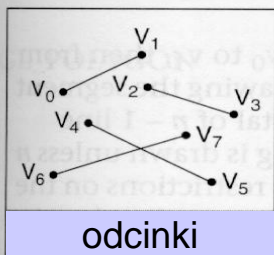
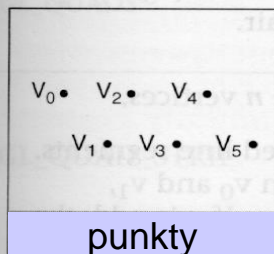


Kreślenie ręczne



Obracamy line renderujemy,  
cieniuujemy

# Modelowanie obiektów - prymitywy



*składanie obiektów z prymitywów*

- Cone
- Cube
- Cylinder
- Disc
- Plane
- Polygon
- Sphere
- Torus
- Capsule
- Oil Tank
- Tube
- Pyramid
- Platonic
- Figure
- Landscape
- Relief

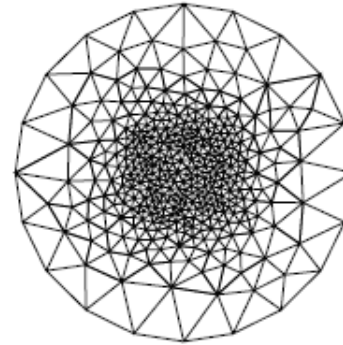
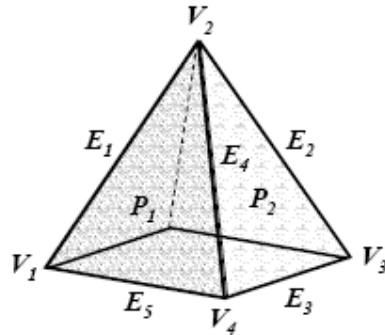
# Modele szkieletowe

## 1. Modele szkieletowe (*wire frame*)

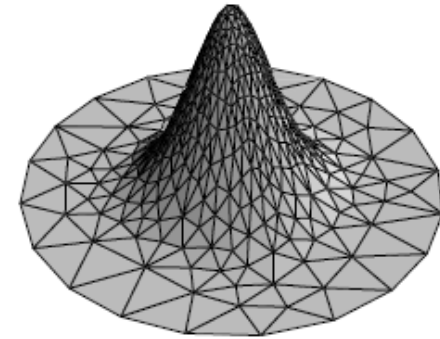
$V_1, V_2, \dots$  - wierzchołki (*vertex*)

$E_1, E_2, \dots$  - krawędzie (*edge*)

$P_1, P_2, \dots$  - ściany (*polygon surface*)



podział dziedziny parametrycznej

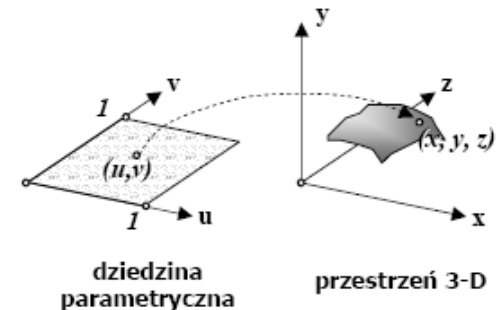


model szkieletowy powierzchni

**Model szkieletowy** - zbiór wierzchołków, krawędzi i ścian (wieloboków) połączonych tak, że każda krawędź jest wspólna przynajmniej dla dwóch ścian.

Najczęściej przetwarzane są w równania powierzchni zapisane w postaci parametrycznej.

$$\begin{aligned}x &= f_x(u, v) \\y &= f_y(u, v) \\z &= f_z(u, v)\end{aligned} \quad 0 \leq u, v \leq 1$$



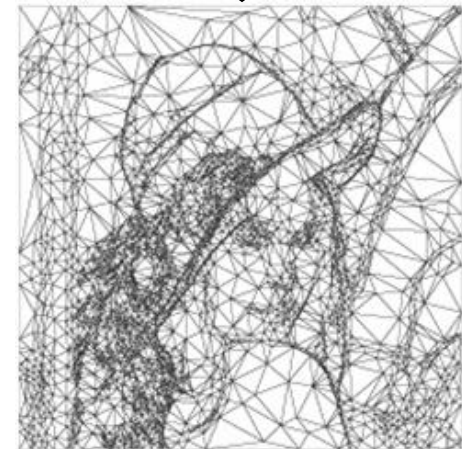
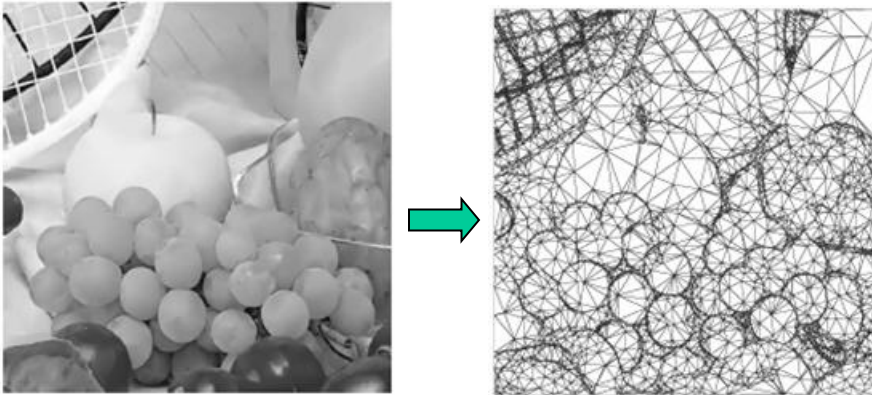
Tworzenie modelu szkieletowego powierzchni polega na:

- podziale dziedziny parametrycznej na wieloboki,
- obliczeniu dla wierzchołków wieloboków punktów w 3-D,
- połączeniu wyliczonych punktów w wieloboki w 3-D.

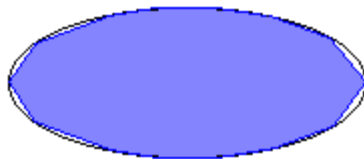


# Modelowanie szkieletowe

- Modele siatek podziału dziedziny parametrycznej

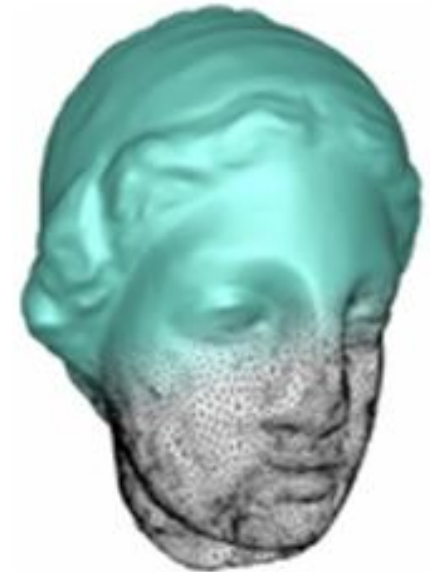
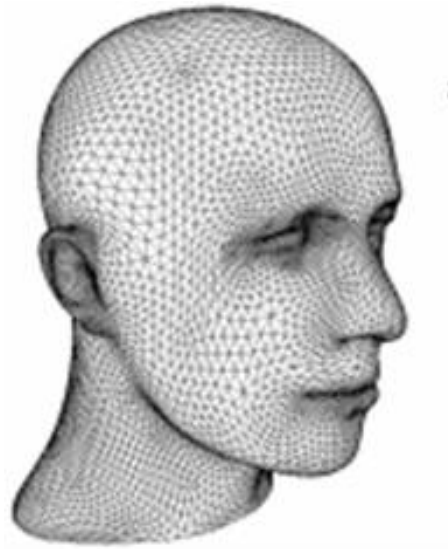
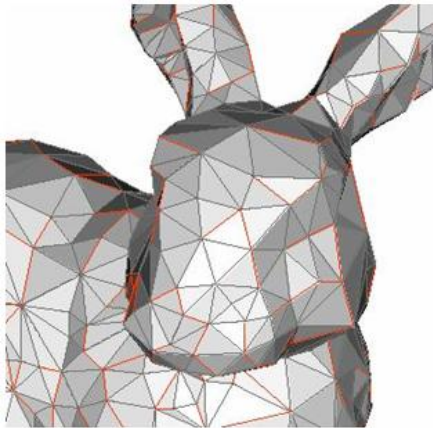


- Rasteryzacja wielokątowa



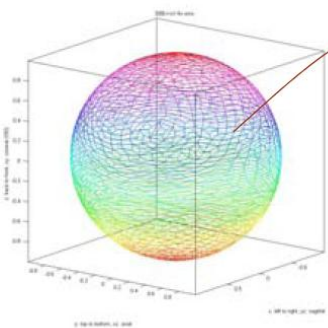
# Przykładowe modele obiektów

---



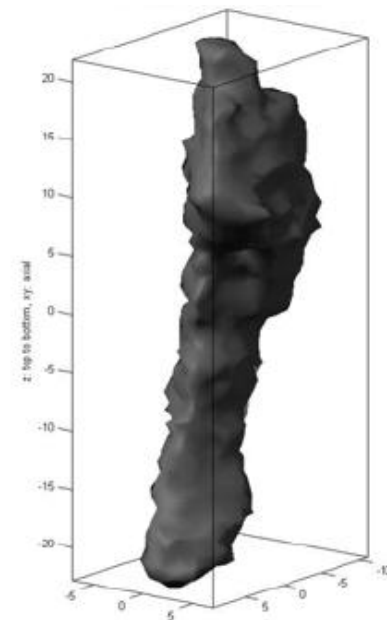
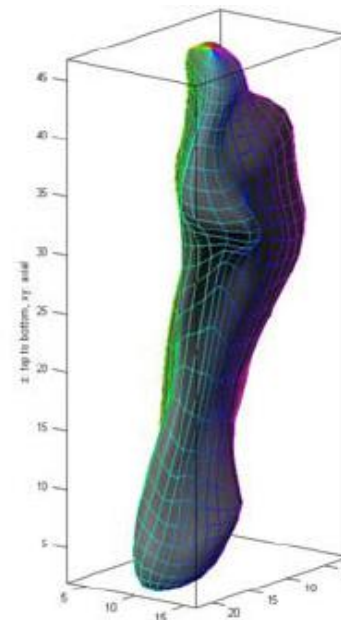
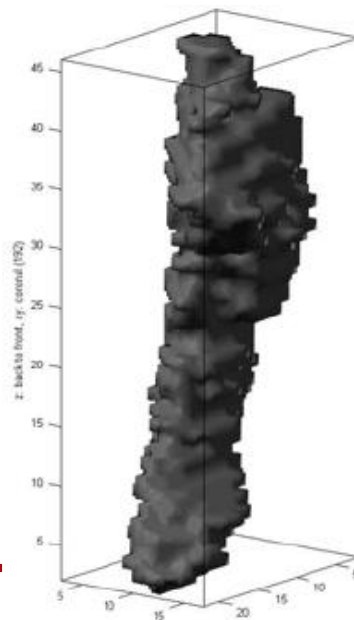
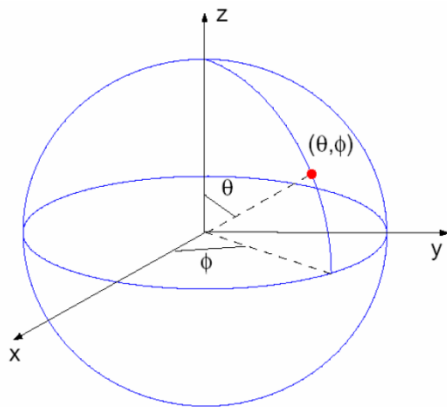
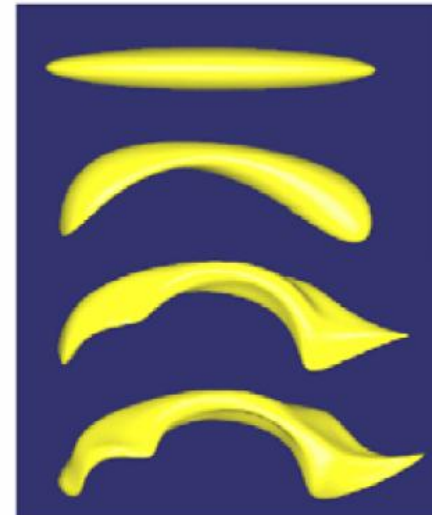
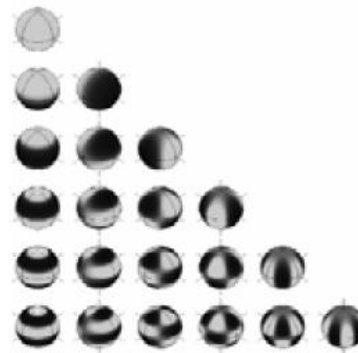
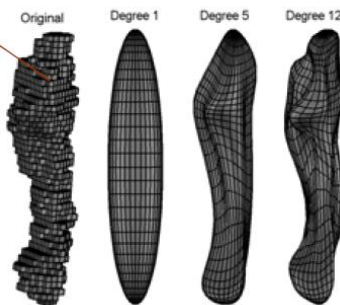


# Przykłady modeli



$$(\theta, \phi) \leftrightarrow (x, y, z)$$

Area-preserving mapping



---

# WYGŁADZANIE MODELI

---

# Uzupełnienie: równanie parametryczne prostej

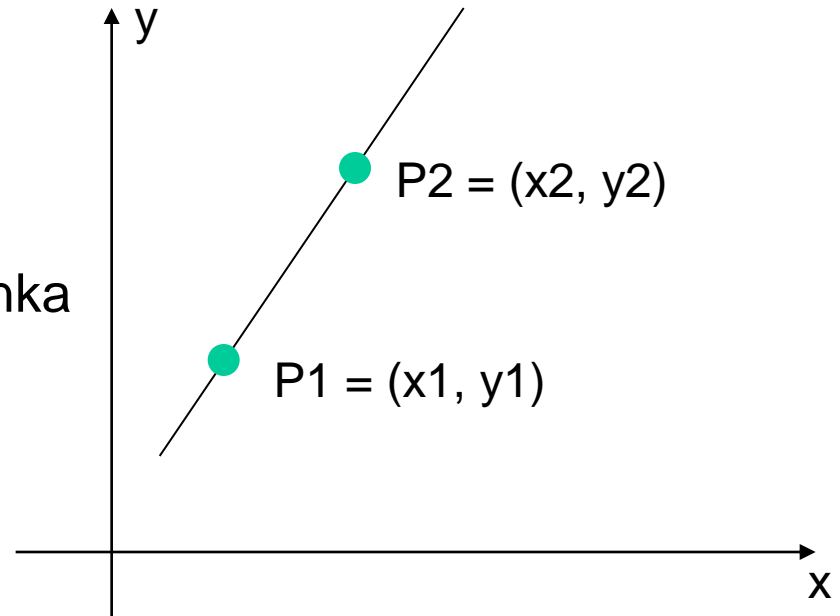
- Dane:  $P1 = (x1, y1)$  and  $P2 = (x2, y2)$

$$x = x1 + t(x2 - x1)$$

$$y = y1 + t(y2 - y1)$$

- 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:

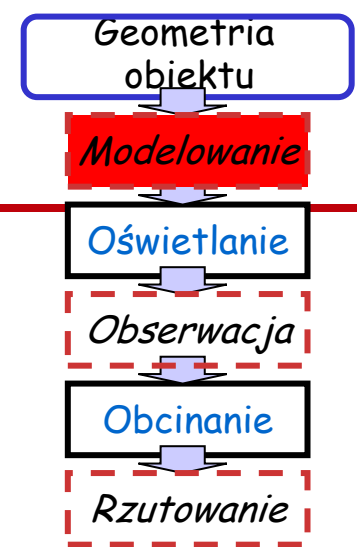
$$x = x1 + t(x2 - x1)$$

$$y = y1 + t(y2 - y1)$$

$$z = z1 + t(z2 - z1)$$

- Równania parametryczne są wykorzystywane przy tworzeniu modeli bardziej złożonych obiektów

# Modelowanie – krzywe parametryczne



- Krzywe 3 stopnia

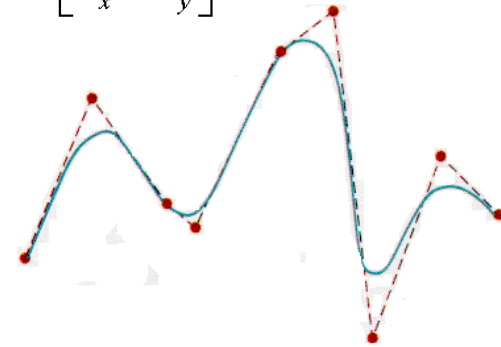
$$x = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y = a_y t^3 + b_y t^2 + c_y t + d_y$$

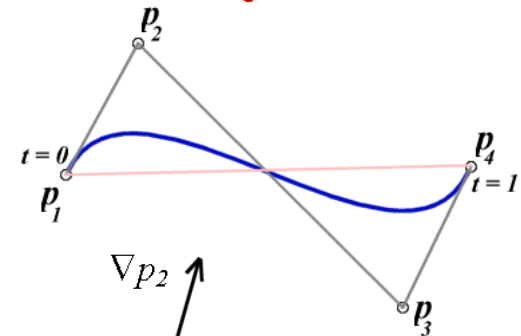
$$[x \ y] = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix}$$

- Krzywe sklejane – 4 punkty kontrolne (lokalne)

- ciągłość C1 i C2 na łączeniach
- aproksymacja



- Beziery - dwa punkty końcowe i dwa (globalne) inne punkty definiujące styczne w punktach końcowych



- Hermita – dwa punkty i wektory (globalne) styczne w tych punktach

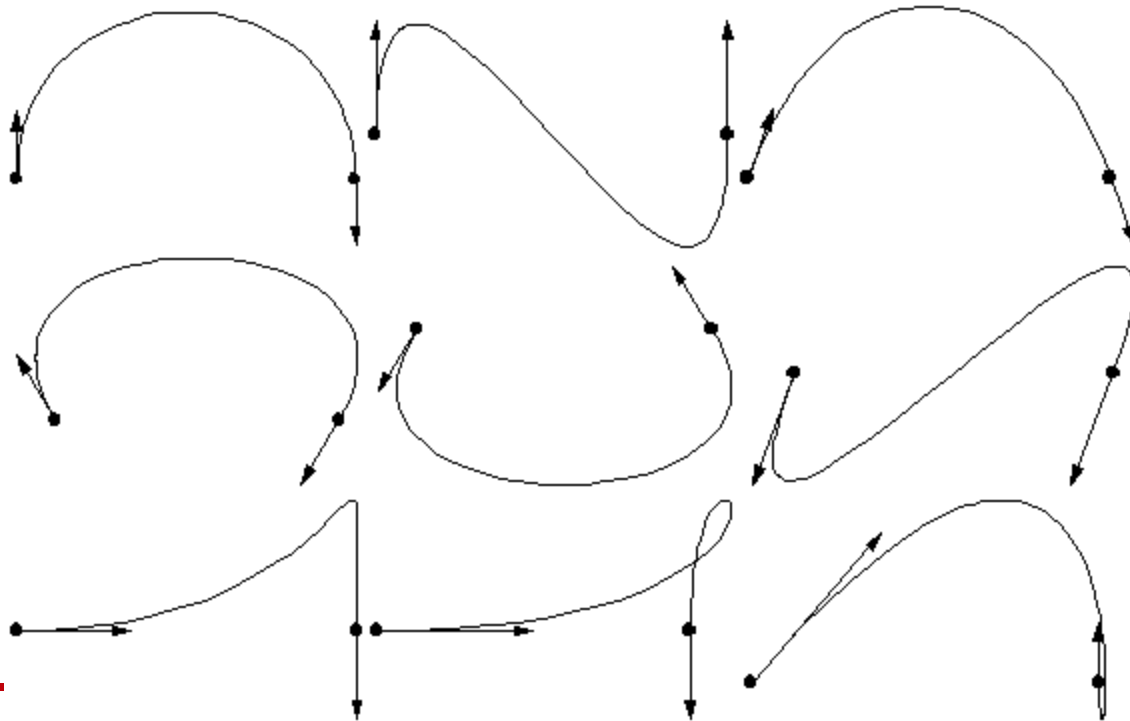


# Krzywe Hermita

## ■ Liczenie

$$[x \quad y] = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \frac{dx_1}{dt} & \frac{dy_1}{dt} \\ \frac{dx_2}{dt} & \frac{dy_2}{dt} \end{bmatrix}$$

## ■ Rysowanie



# Krzywe Bezierra

$$[x \ y] = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

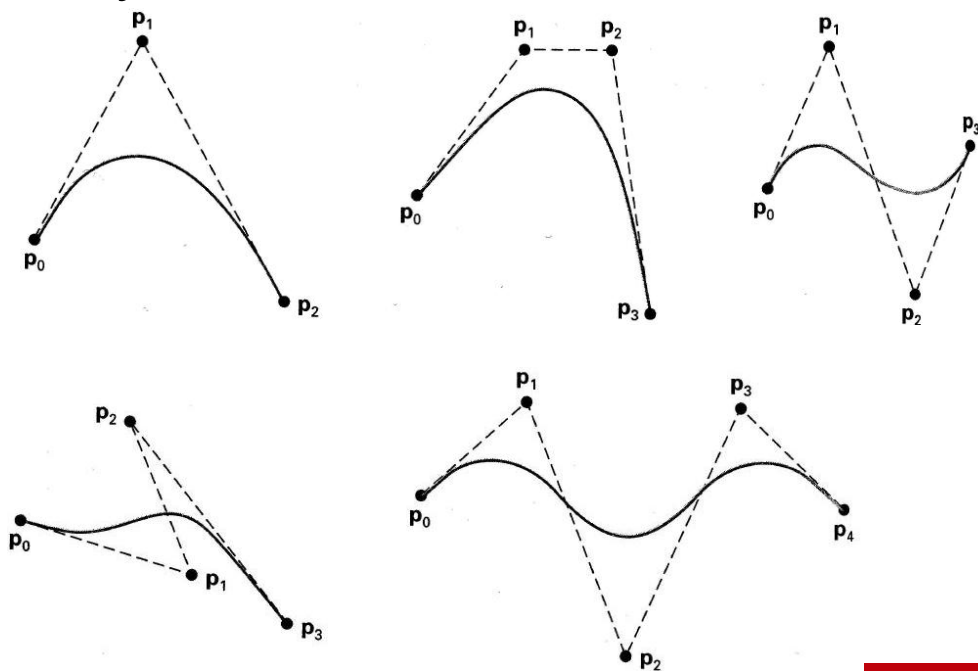
## ■ Liczenie

$$\begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

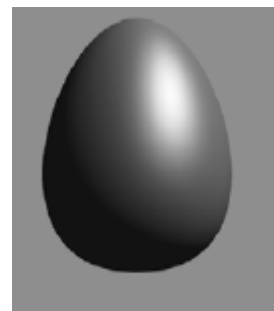
$$p(t) = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

wielomiany *Bernsteina*

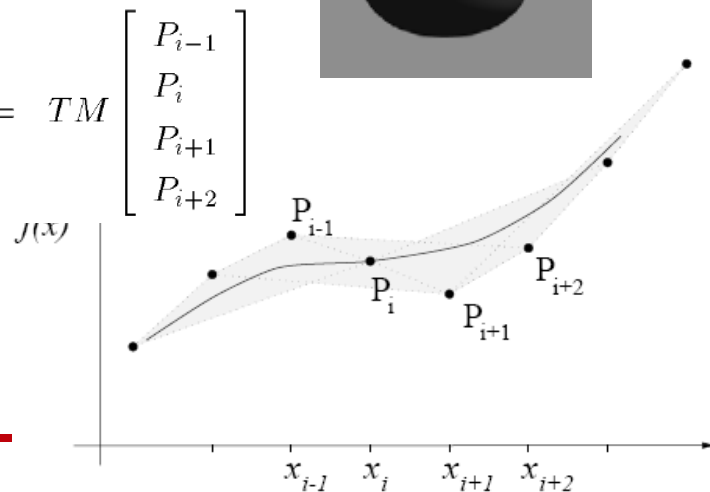
## ■ Rysowanie



efekt

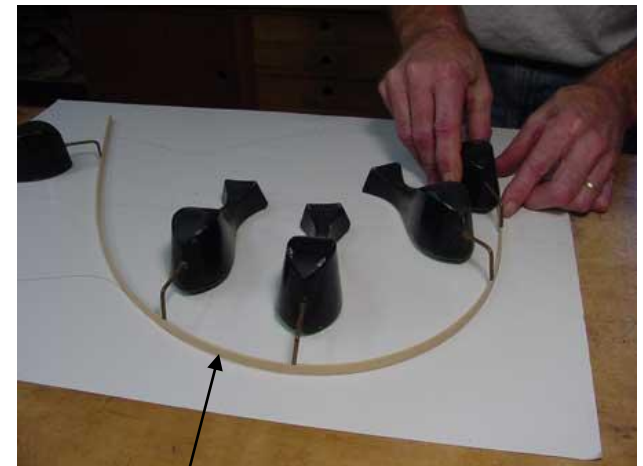
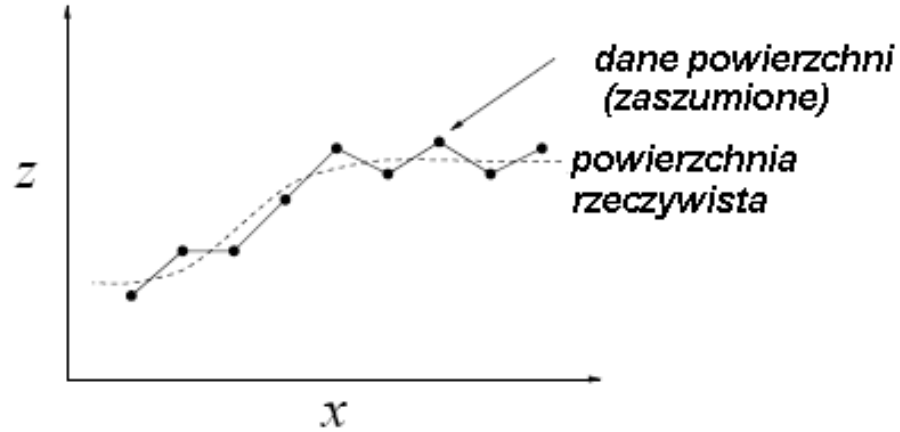


$$f(x) = TM \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}$$



# Modelowanie - krzywe sklejane

- wpasowanie modelu w dane rzeczywiste

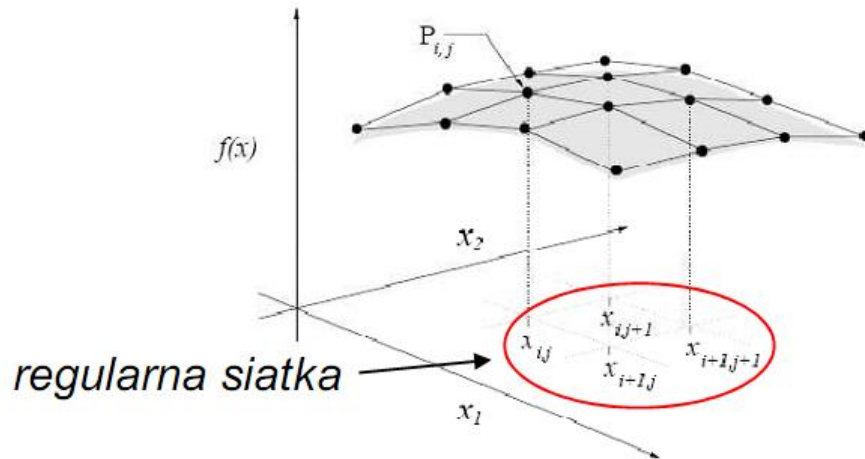


spline

# Krzywe sklejane w 3W

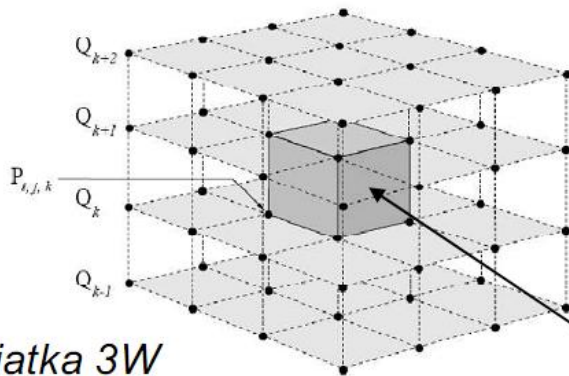
$$0 \leq x_1, x_2 \leq 1$$

- sklejanie w 2W (powierzchnie)



$$Q = \begin{bmatrix} P_{i-1,j-1} & P_{i,j-1} & P_{i+1,j-1} & P_{i+2,j-1} \\ P_{i-1,j} & P_{i,j} & P_{i+1,j} & P_{i+2,j} \\ P_{i-1,j+1} & P_{i,j+1} & P_{i+1,j+1} & P_{i+2,j+1} \\ P_{i-1,j+2} & P_{i,j+2} & P_{i+1,j+2} & P_{i+2,j+2} \end{bmatrix}$$

- sklejanie w 3W



$$f(\mathbf{x}) = RM \begin{bmatrix} SM & Q_{k-1} & M^T T \\ SM & Q_k & M^T T \\ SM & Q_{k+1} & M^T T \\ SM & Q_{k+2} & M^T T \end{bmatrix}, \quad 0 \leq x_1, x_2, x_3 \leq 1$$

$$R = \begin{bmatrix} x_3^3 & x_3^2 & x_3 & 1 \end{bmatrix}$$

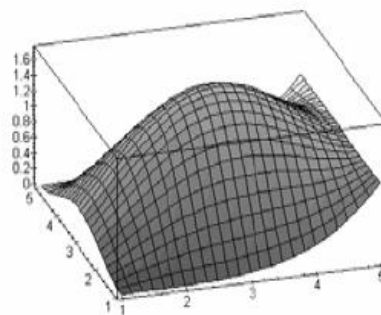
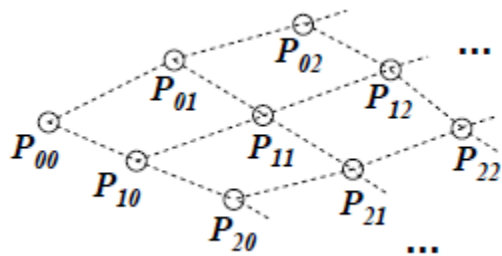
regularna siatka 3W

tutaj określamy f.sklejaną 3W

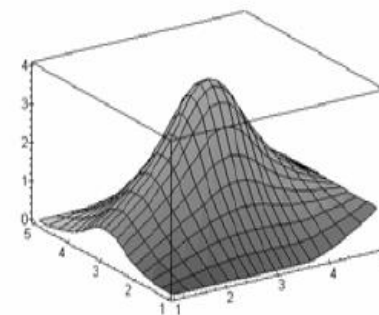


# Rozpinanie gładkich siatek – modeli powierzchni

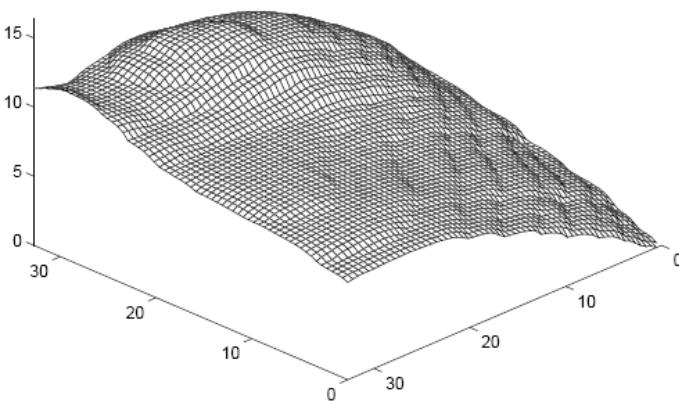
## ■ Siatki punktów kontrolnych



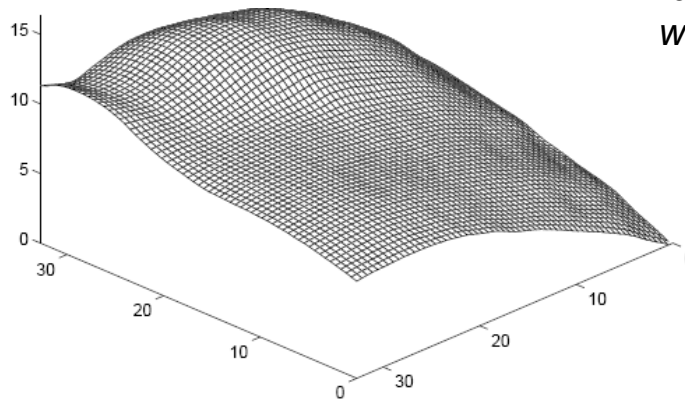
*powierzchnia Beziera*



*niejednorodna powierzchnia B-sklejana (NURBS – non-uniform rational B-spline, wprowadza się wagi punktów wiodących)*



*model liniowy*



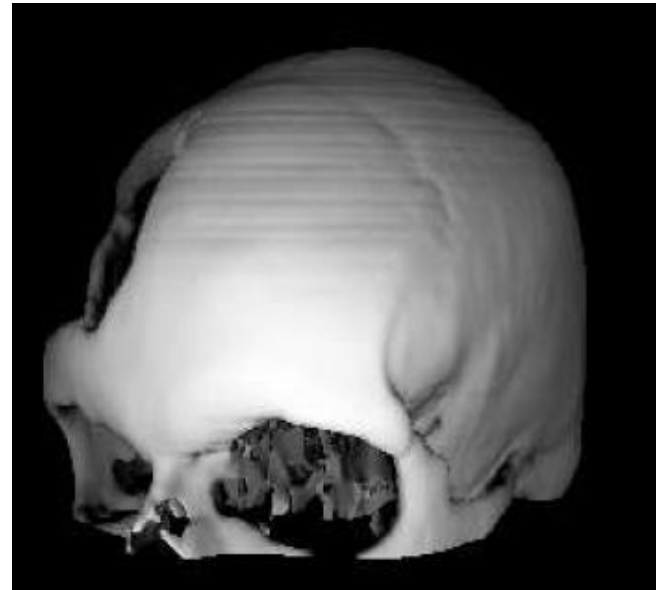
*sklejanie wielomianami stopnia 3*

# Efekty renderingu powierzchni

---

- Model krzywych sklejanych z ray-castingiem

*z różnymi gradientami*



---

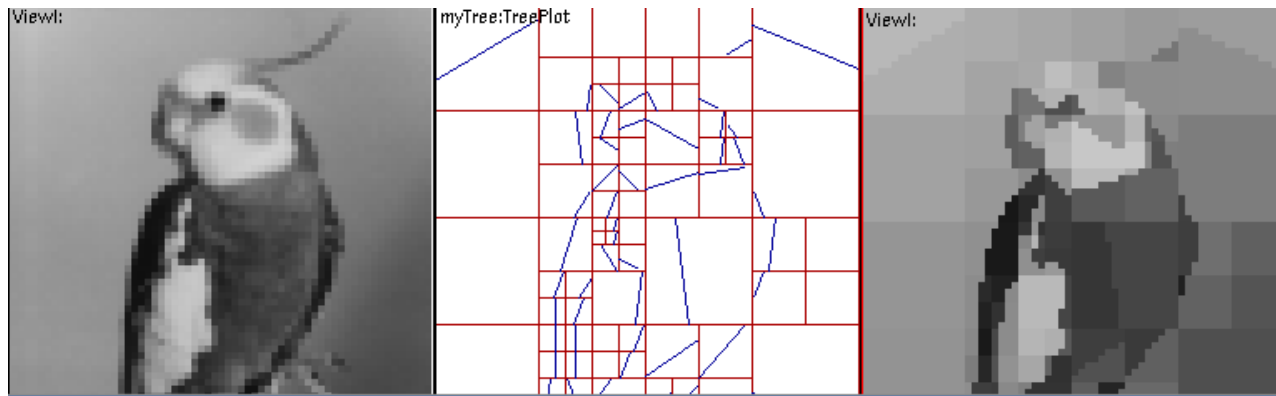
# **MODELE UCZONE NATURA**

---

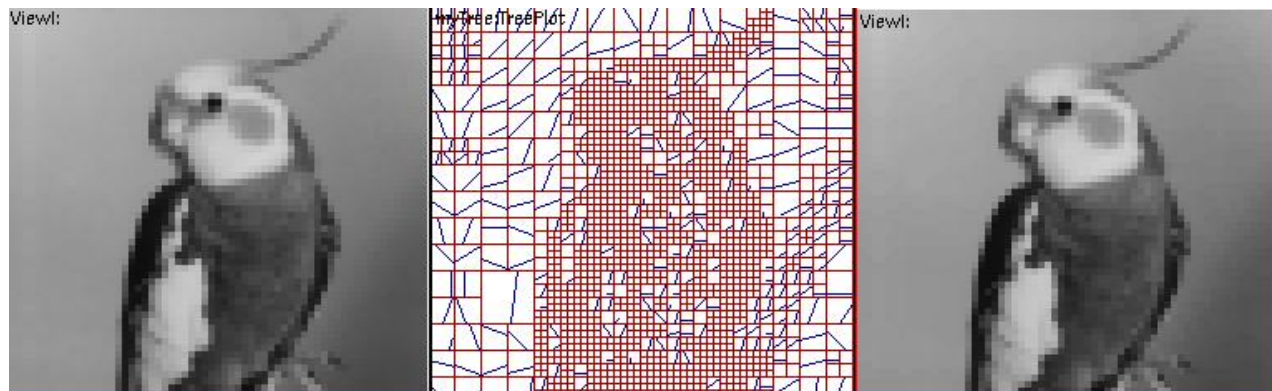
# Baza geometryczna (wedgelets)

Aproksymacja obrazów zbiorem 'kawałków-klinów' o stałej wartości funkcji jasności określonych w dziedzinie obrazu (podział dziedziny rozłączny i zupełny)

$$I = \bigcup_{w \in W} w \quad \text{i } w - \text{const}$$

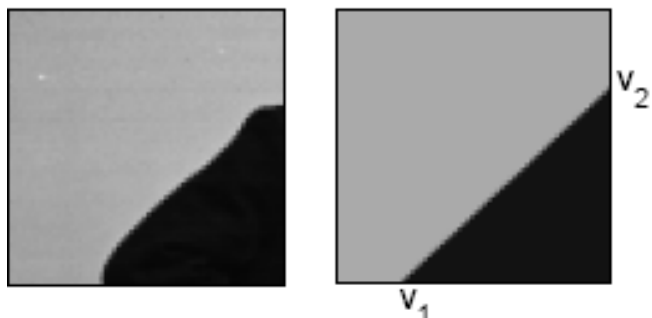
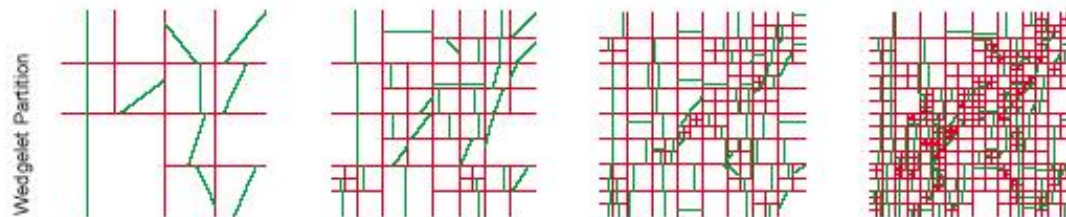
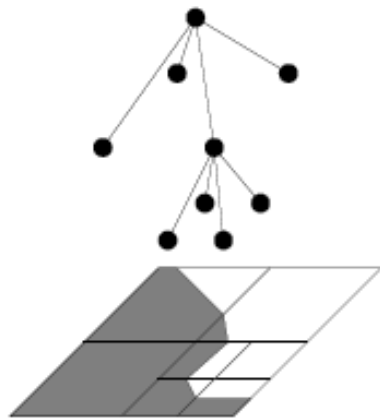


*zgrubnie*



*dokładniej*

# Modelowanie z Wedgelets

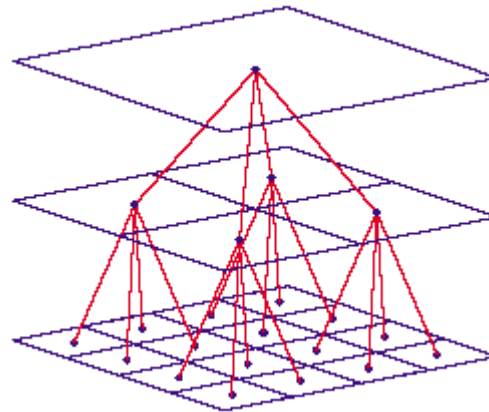


*D.L. Donoho, X. Huo, 'Beamlets and Multiscale Image Analysis'*

aproxymacja za pomocą liniowych nieciągłości (klin) - błąd przybliżenia regulowany skalą

# Dekompozycja na kwadraty

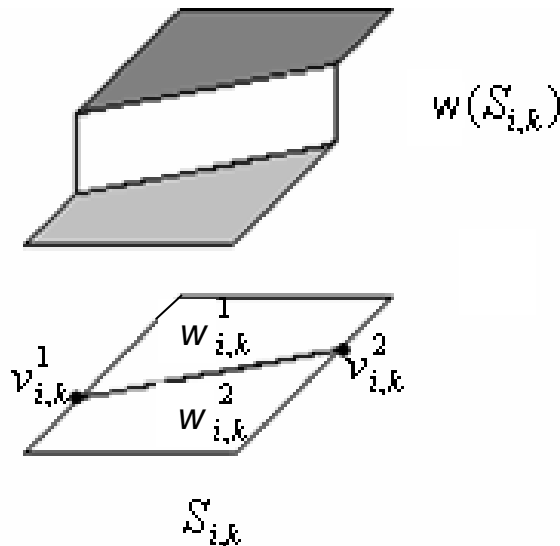
- segmentacja drzewa czwórkowego (hierarchiczna struktura dekompozycji):  
podział obrazu  $I([0,1]^2)$  na bloki  $S$  (kwadraty  $2^k$ ),  $k=0, \dots, K$



piramida dekompozycji

# Wedgelets (falki W)

Wedgelet (falka W) jest to kawałkami stała funkcja określona na kwadracie diadycznym (długość boku jest równa  $2^k$ ) z nieciągłością liniową.



Definicja falki W dla  $S_{i,k}$ :  
 $(v_{i,k}^1, v_{i,k}^2), w_{i,k}^1, w_{i,k}^2$

Kwadrat  $S$  podzielony na dwa regiony jest dziedziną falki. Orientację liniowej nieciągłości określa para punktów  $v$ , a wartości funkcji są stałe w obu regionach

Falka W jest zdegenerowana jeśli funkcja jest stała ( $w(S_{i,k}) = \text{const}$  - bez nieciągłości)

# Kwadraty i kliny - porównanie

---

oryginał



kwadraty



Aproksymacja liniowa  
funkcji jasności (platelets)

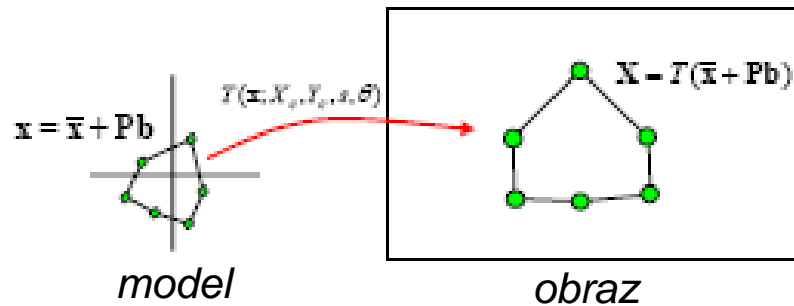
---

kliny

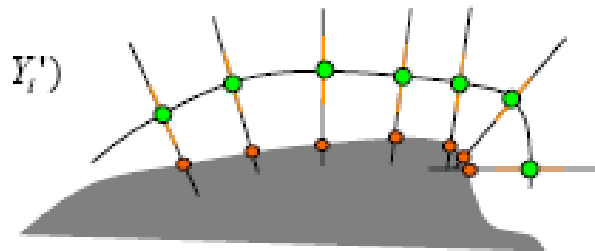
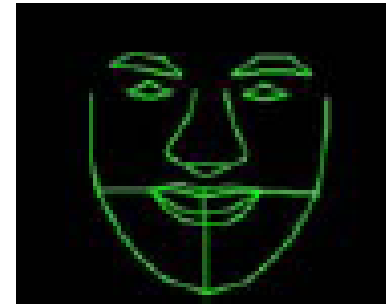




# Modele obiektowe, czyli pasowanie wzorców (dane treningowe)



1) wpisanie modelu w obraz



2) dopasowanie

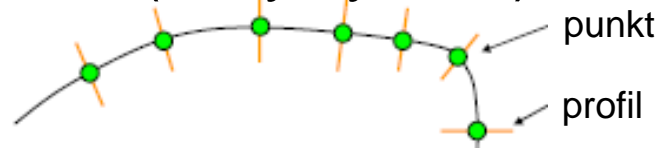


# Metoda aktywnych konturów

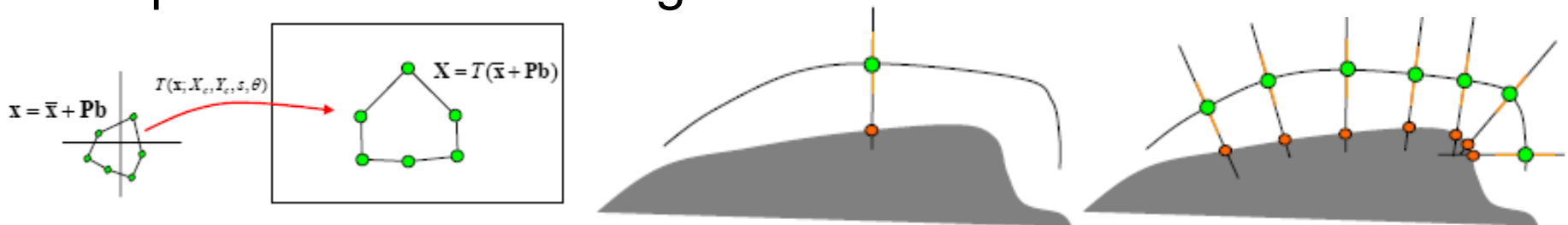
- Skalowalny wzorzec z danych treningowych (różne parametry modelu) - elastyczny model opisuje **typowy kształt** krawędzi obiektu pasowany do obrazu



- Model punktu i profilu (statystyczne)



- Dopasowanie do realnego kształtu obiektu



*globalne przekształcenie wstępne  
(często manualne)*

---

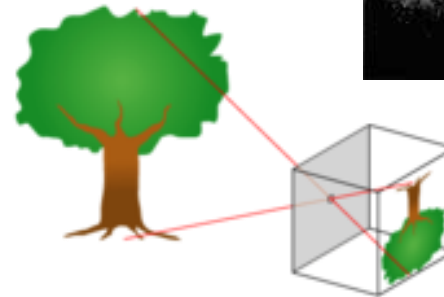
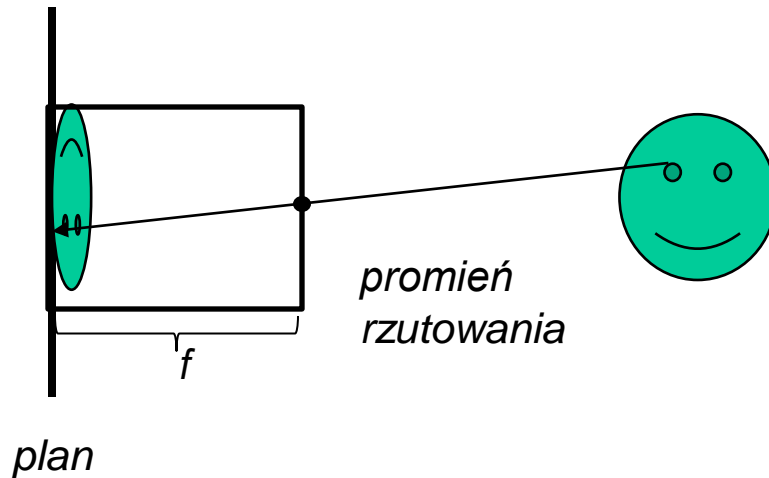
# **MODELE KAMERY**

---

# Modele kamery - otworkowa

## Camera obscura (prototyp aparatu fot.)

- poczernionego wewnątrz pudełka (dla zredukowania odbić światła)
- na jednej ścianie niewielki otwór (pełniący rolę obiektywu), na drugiej matowa szyba lub klisza
- cechy obrazu: rozmycie, nieograniczona głębia ostrości, miękkość, łagodne kontrasty, brak dystorsji, pastelowa kolorystyka



<http://www.pinhole.art.pl/>



Średnica otworka mniej więcej 0,3-1mm ( $d = 1,9\sqrt{f\lambda}$ )

# Kamery otworkowe



[http://www.kwant.info/projekty/lab\\_cyfrowy/](http://www.kwant.info/projekty/lab_cyfrowy/)

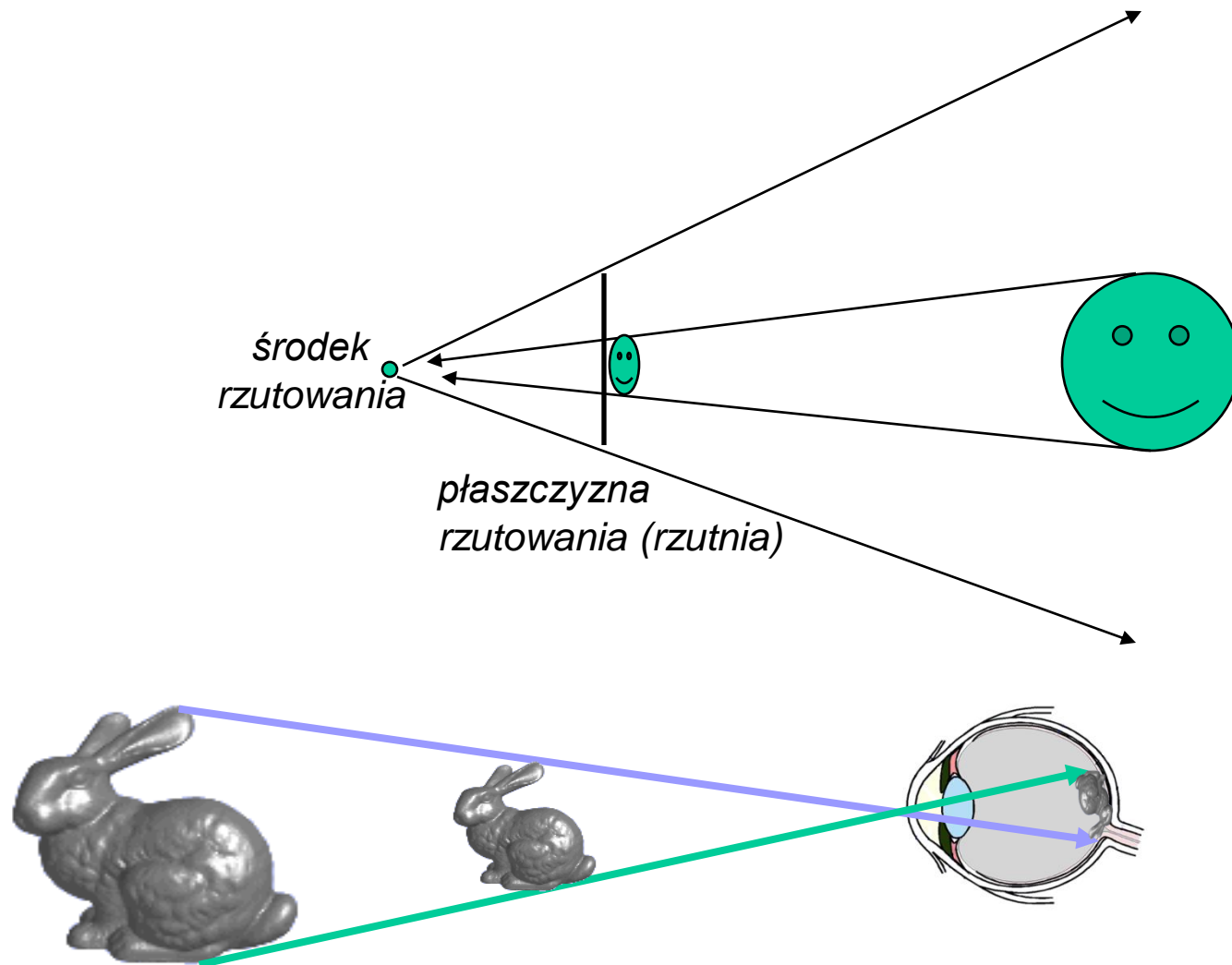


<http://pedagog.umcs.lublin.pl/~wafra/otworek.html>



<http://sulisz.redblog.dziennikwschodni.pl/2009/03/13/kamery-otworkowe-vermeer-powstaja-w-polsce/>

# Kamera w grafice komputerowej



---

**EFEKTY 3W**

---

# Efekty przestrzenne na obrazie

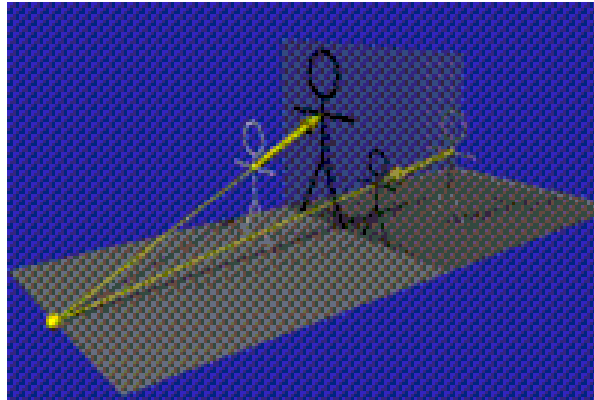
---

- Perspektywa
    - Dalsze obiekty są mniejsze
    - Linie równoległe wydają się zbieżne
    - Głębina ostrości
  - Oświetlenie
    - zróżnicowanie oświetlenia powierzchni
    - cienie
  - Stereoskopia (dwoje oczu)
  - Ruch
  - Współdziałanie zmysłów
  - ...
-



# Rzutowanie

- Rzut perspektywiczny (opis kamery)

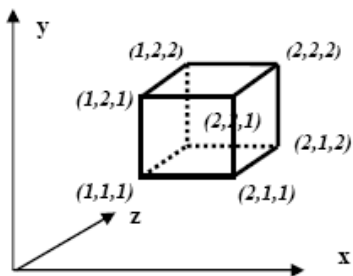


- Rzut równoległy (środek rzutowania  $\rightarrow \infty$ ): ortogonalny lub ukośny (proste rzutowanie prostopadłe lub nie do rzutni)

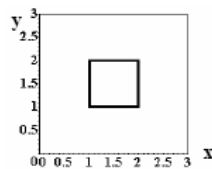


# Rzuty równoległe

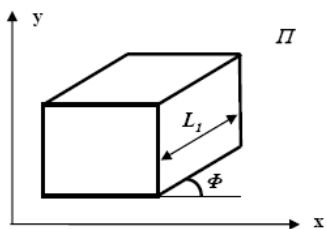
Obiekt rzutowany



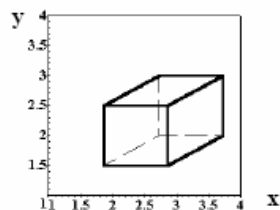
rzut ortogonalny



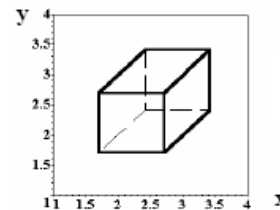
rzuty ukośne



*kawaleryjski*



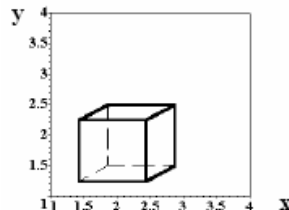
$\Phi = 30^\circ$



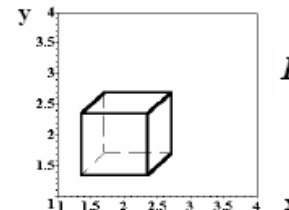
$\Phi = 45^\circ$

$$L_1 = 1 / \operatorname{tg} \alpha = 1, \alpha = 45^\circ$$

*gabinetowy*

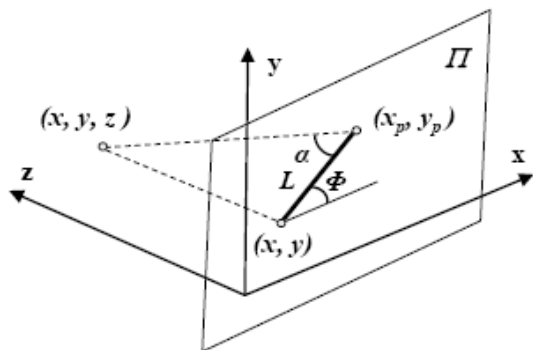


$\Phi = 30^\circ$



$\Phi = 45^\circ$

$$L_1 = 1 / \operatorname{tg} \alpha = 1 / 2, \alpha \approx 63^\circ$$

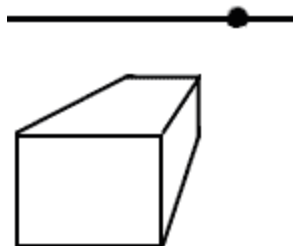
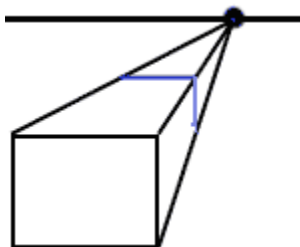


$\Phi$  - lewo-prawo

$\alpha$  - góra-dół

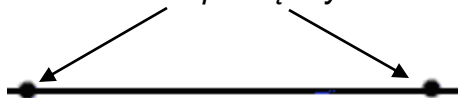
# Rzut perspektywiczny

- **jednopunktowy**

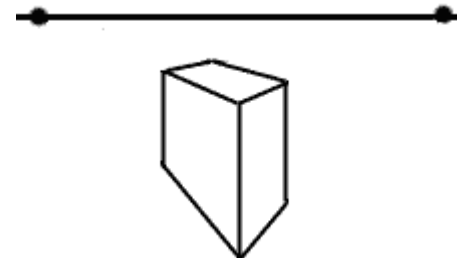
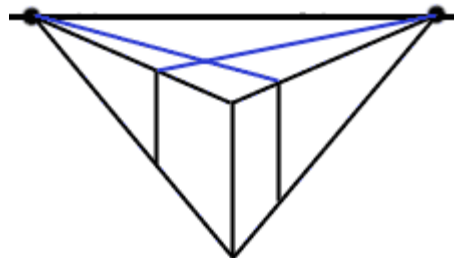


- **dwupunktowy**

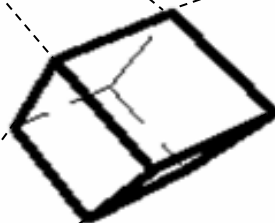
*punkty pozorne związane z układem współrzędnych obiektu*



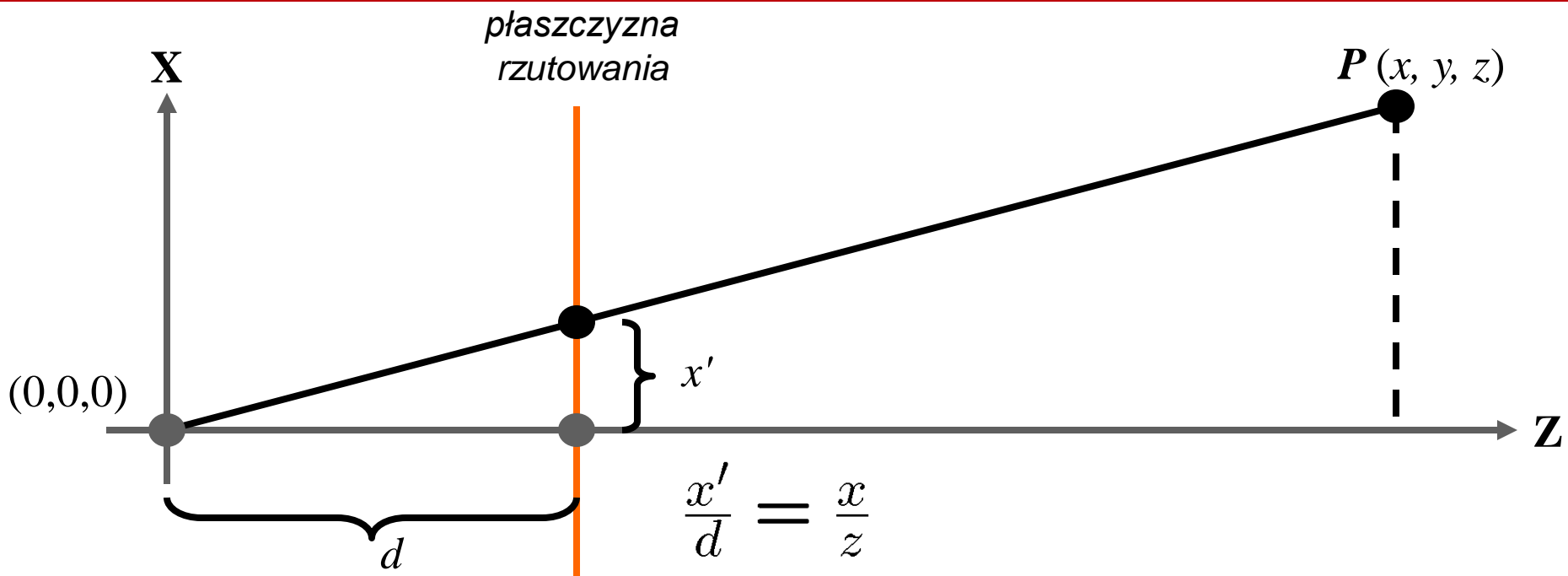
*krawędź centralna*



- **trójpunktowy**



# Obliczanie wymiarów rzutowanych obiektów



analogicznie

$$\frac{y'}{d} = \frac{y}{z}$$

zatem

$$x' = \frac{d \cdot x}{z} = \frac{x}{z/d}, \quad y' = \frac{d \cdot y}{z} = \frac{y}{z/d}, \quad z' = d$$

# Rzutowanie

---

W rachunku macierzowym mamy

- dla rzutu perspektywicznego (współrzędne  $\left( \frac{x}{z/d}, \frac{y}{z/d}, d \right)$  )

$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- dla rzutu ortogonalnego

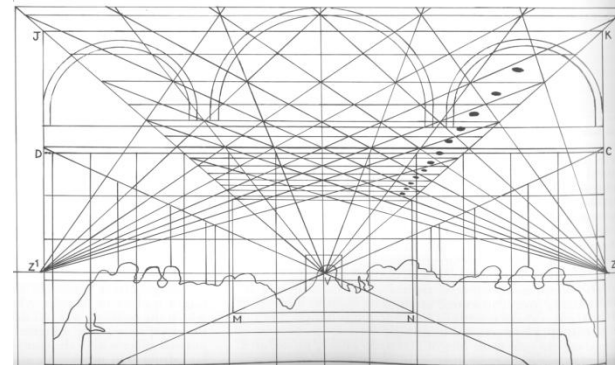
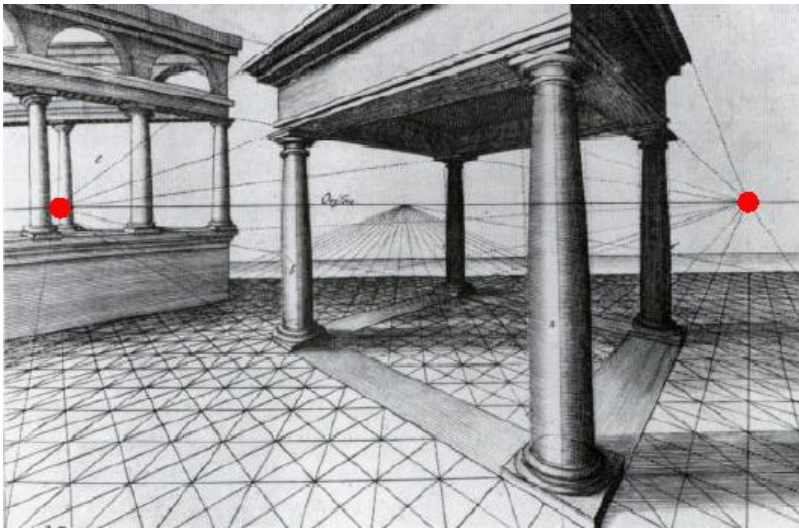
$$\begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

---

# Perspektywa - przykłady



Baptysterium św. Jana



*Ostatnia wieczerza, Leonardo da Vinci, 1495-1498*



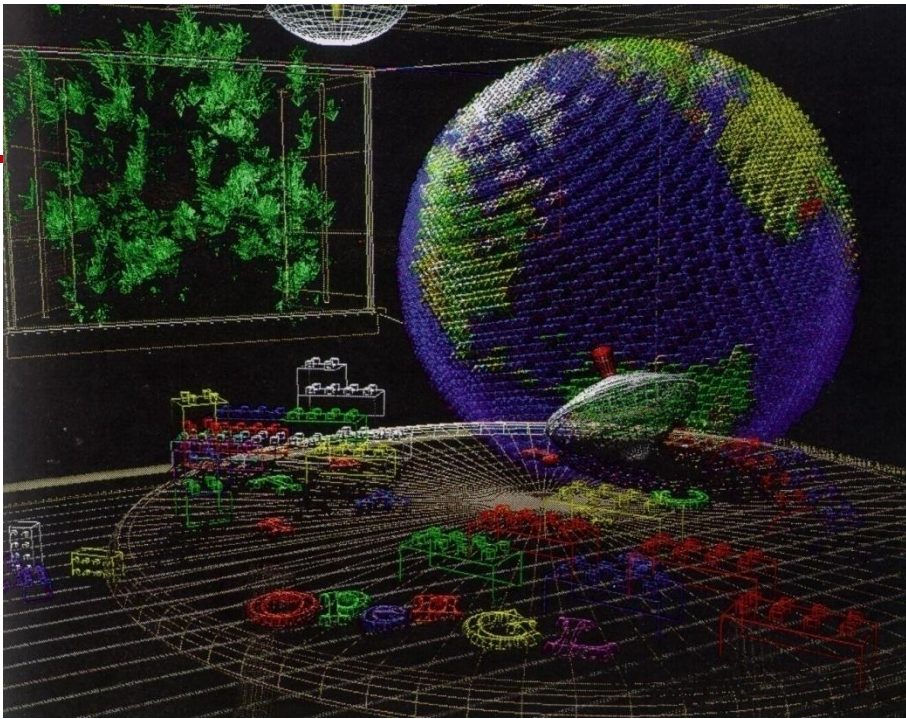
# Przykład odtwarzania

za

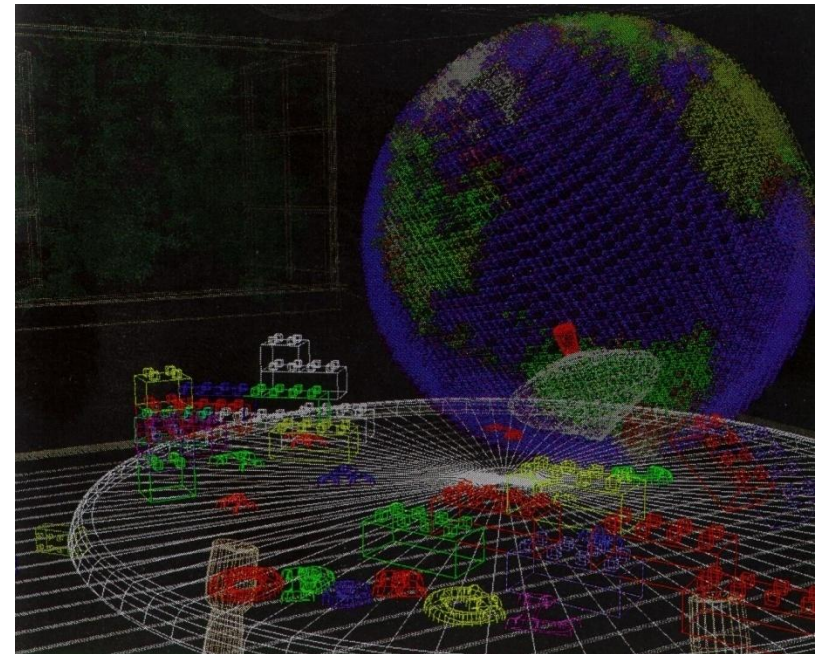
*David Brogan*

[dbrogan@cs.virginia.edu](mailto:dbrogan@cs.virginia.edu)

*Introduction to computer graphics*



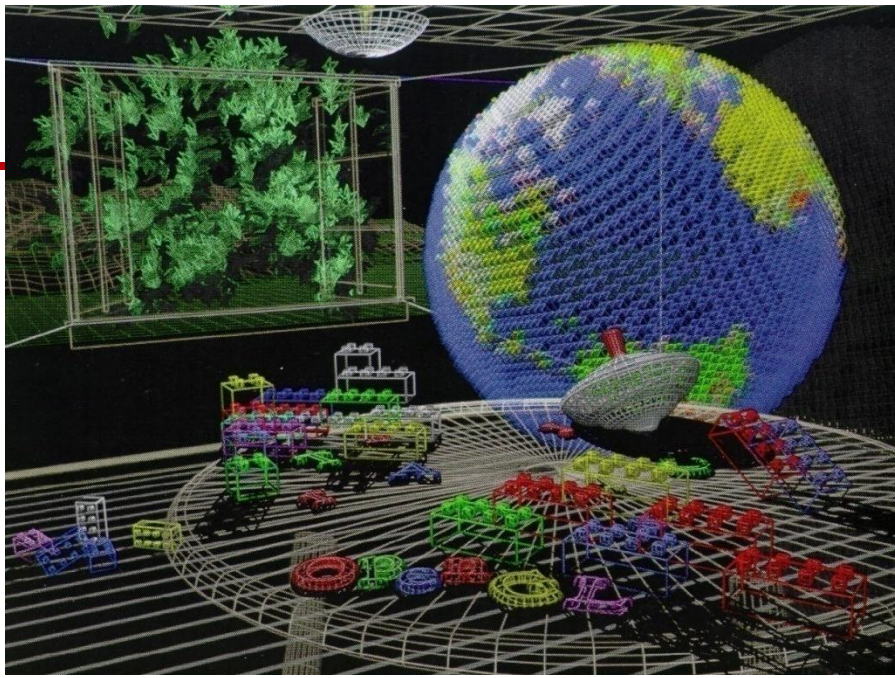
widok siatki – zrzutowany model sceny



bardziej dopasowany model siatki  
(głębka – rozmywanie elementów  
bardziej odległych)



# proste cieniowanie



siatka antyaliasingowa



cieniowanie  
(równomierne)  
wielokątów



# urealnienie



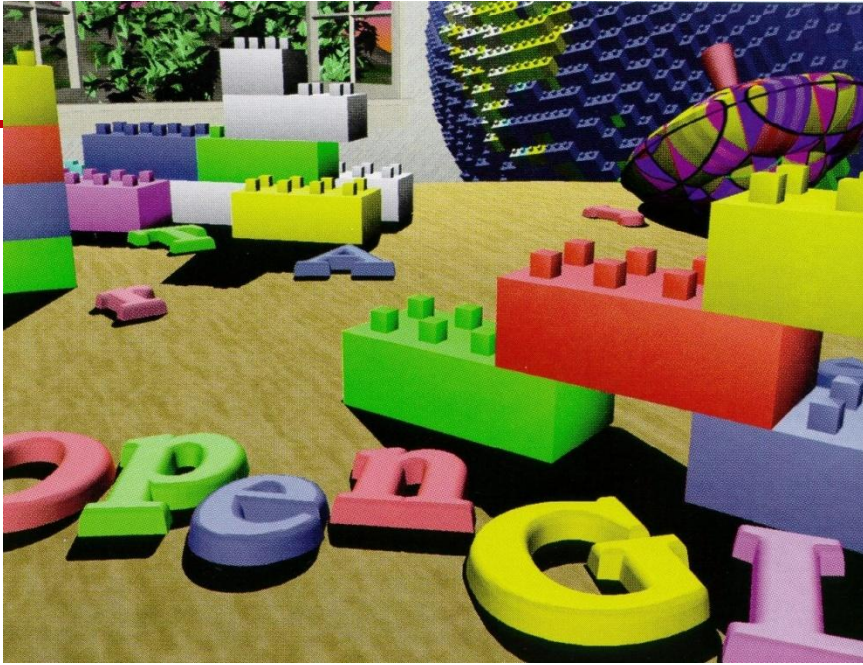
zróżnicowane cieniowanie  
wielokątów



tekstury i cienie



# dodatkowe efekty



przybliżenie



zamglonie