

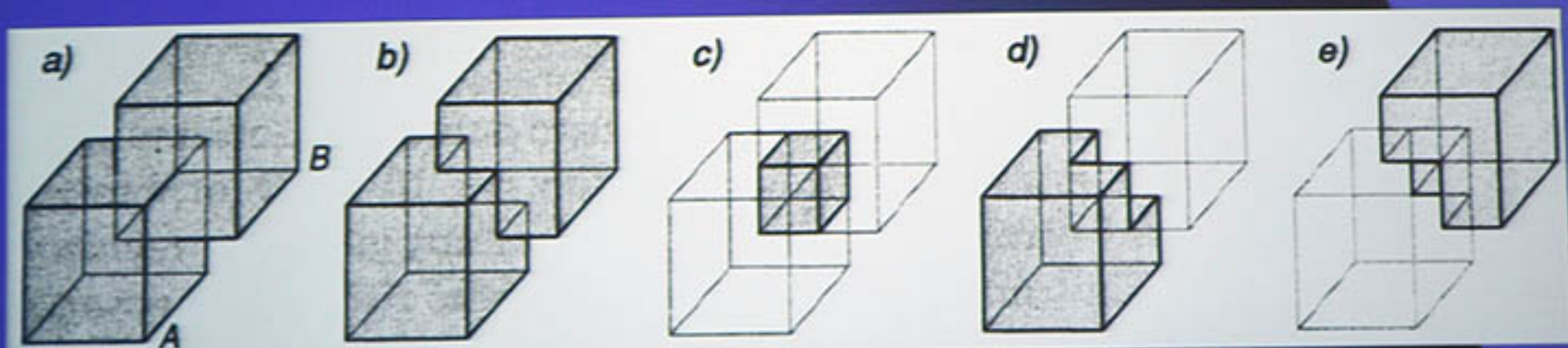
Bryły – operacje boolowskie

Zbiór operacji boolowskich

- Suma
- Różnica
- Przecięcie

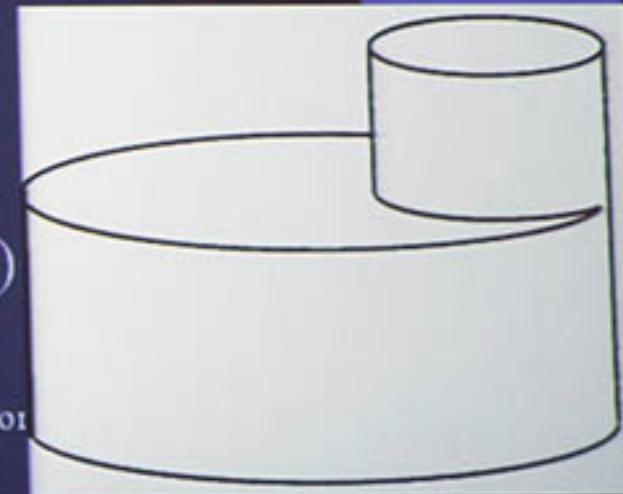
W wyniku tych operacji mogą powstać bryły, odcinki punkty.

Regularyzowane operatory boolowskie – wykonanie operacji na bryłach daje bryłę.



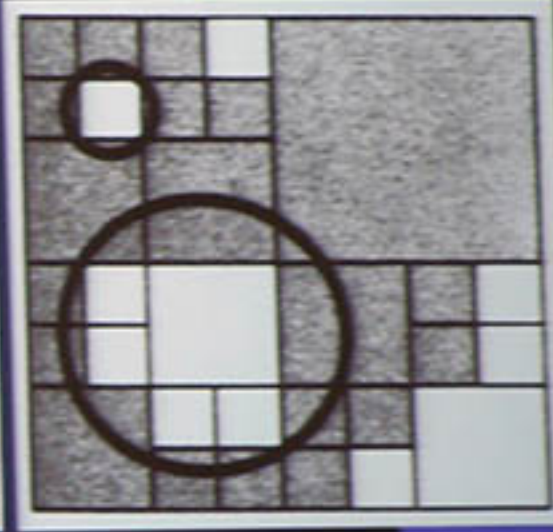
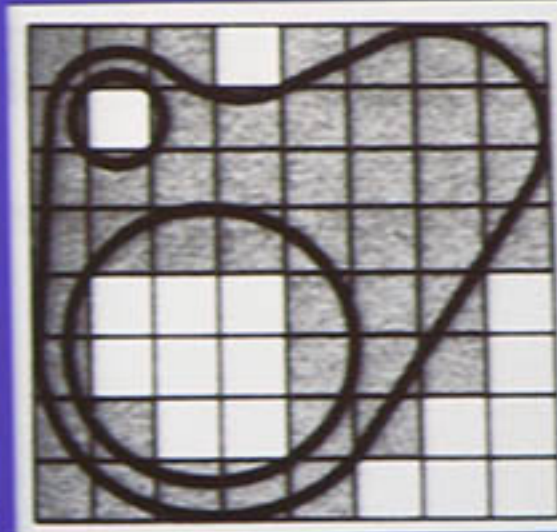
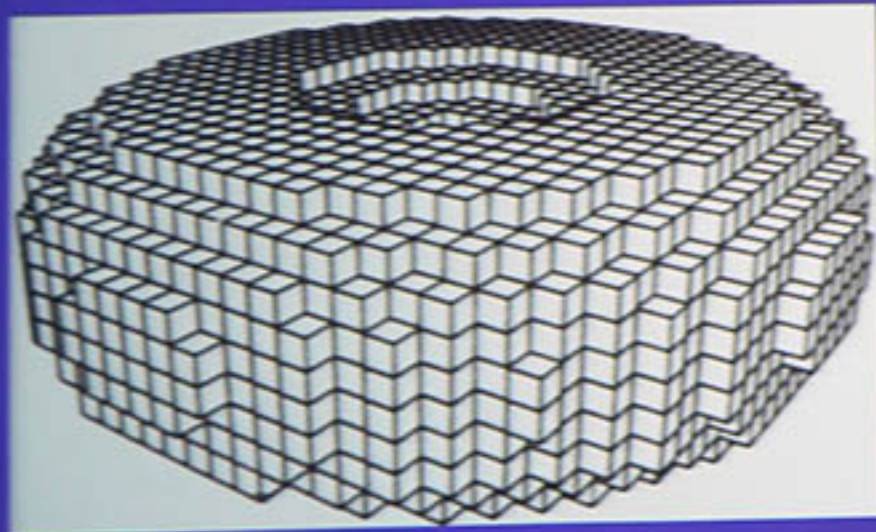
Sposoby reprezentacji brył

- 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



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



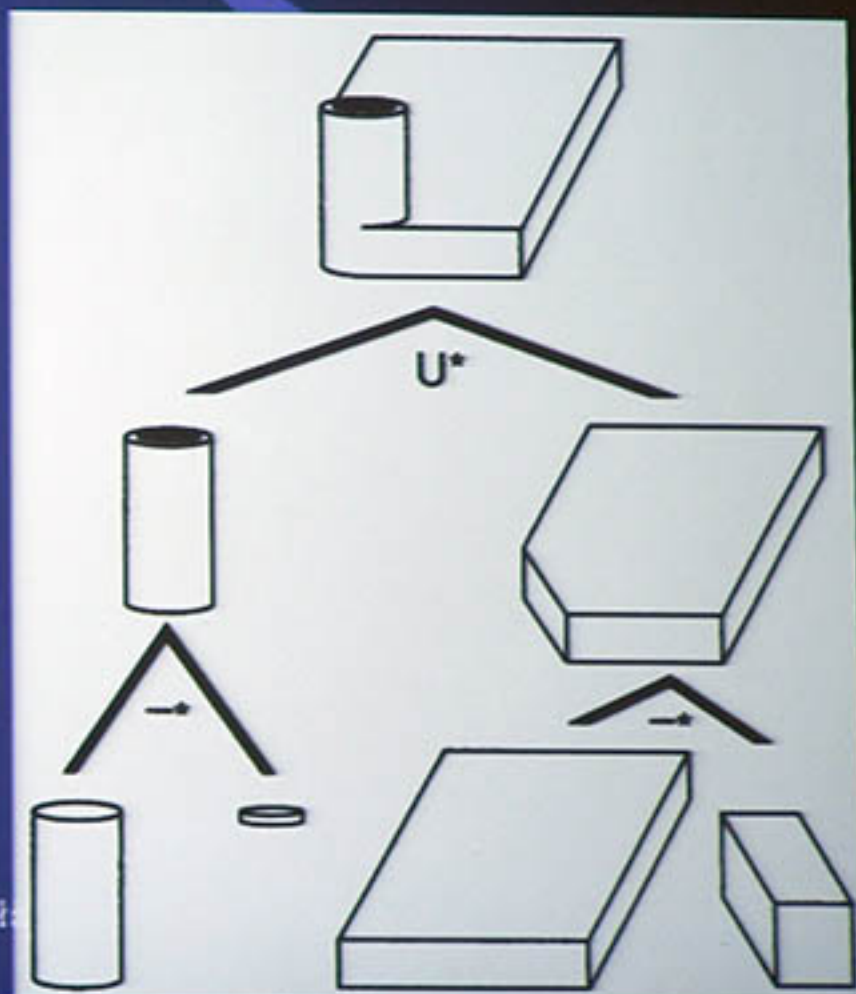
Sposoby reprezentacji brył (c.d.)

■ 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

Eliminacja powierzchni niewidocznych

Wyznaczanie powierzchni widocznych

Które powierzchnie, krawędzie i punkty są widoczne ze środka rzutowania (albo wzdłuż kierunku rzutowania)?

- Sposób usuwania a urządzenie graficzne
 - Monitor (usuwanie narysowanych elementów, wielokrotne rysowanie)
 - Urządzenia rysujące

Algorytmy usuwania powierzchni niewidocznych ²

- Podział algorytmów usuwania
 - Dyskretne (pikselowe)
 - Analityczne

 - *Algorytmy z precyzją pikselową*
 - *Algorytm z precyzją obiektową*

 - *Algorytmy przestrzeni danych*
 - *Algorytmy przestrzeni obrazu*

Algorytm z precyzją pikselową

Dla każdego piksela obrazu

- {
- Wyznacz obiekt najbliższy obserwatora, który jest napotykanym przez promień rzutowania przechodzący przez piksel
- Narysuj piksel o odpowiedniej barwie
- }

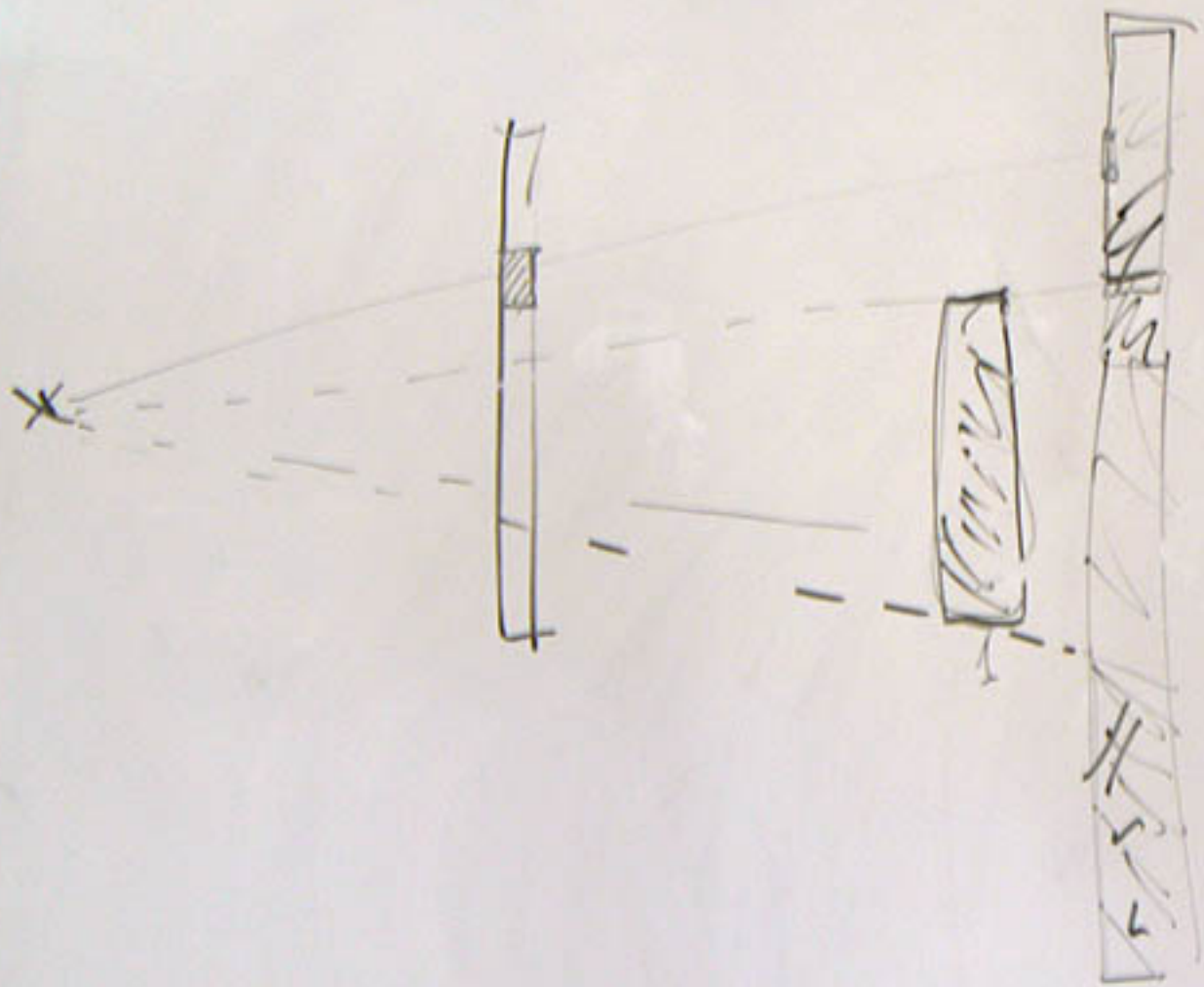
Algorytm z precyzją obiektową

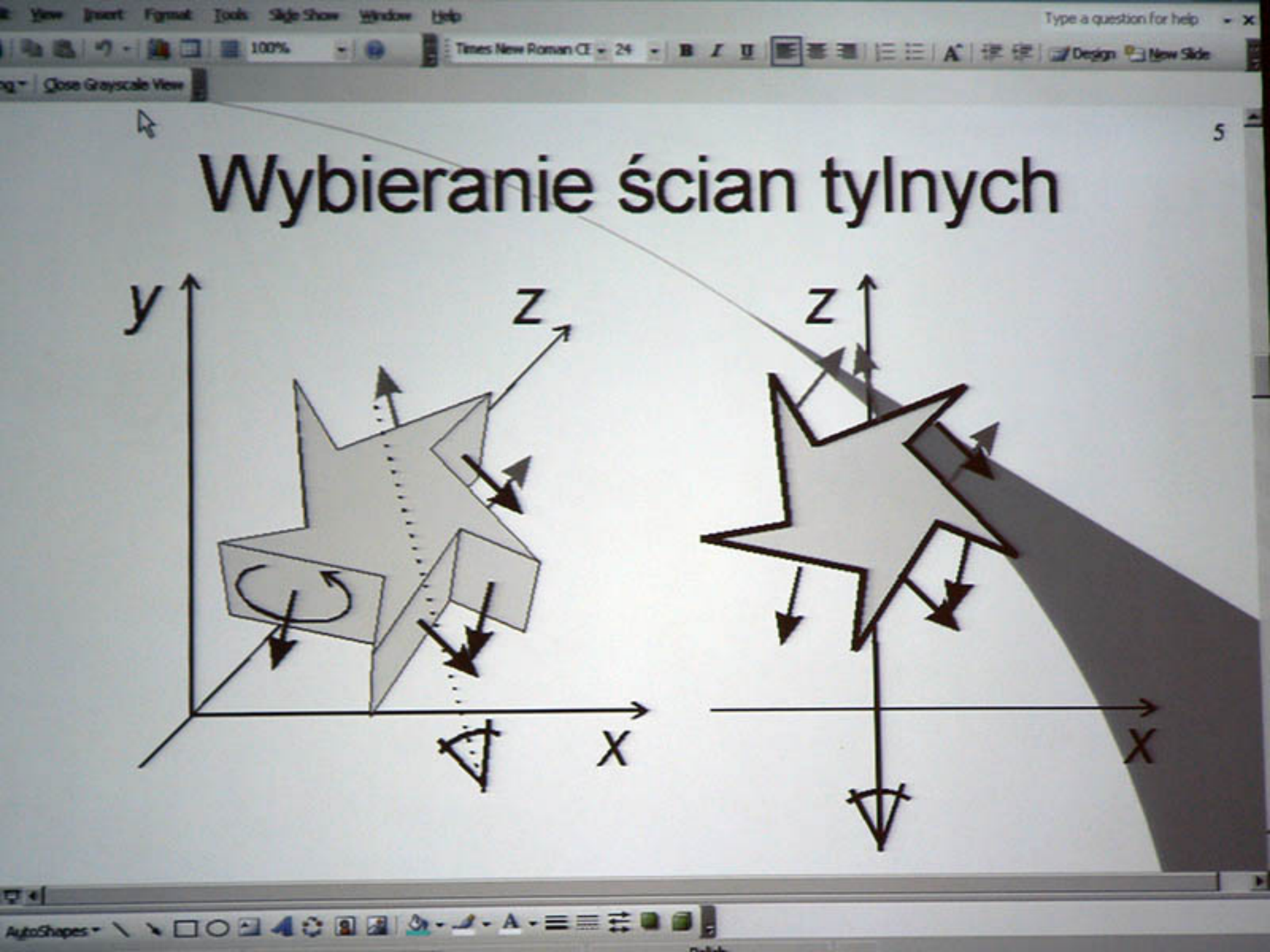
Dla każdego obiektu

{

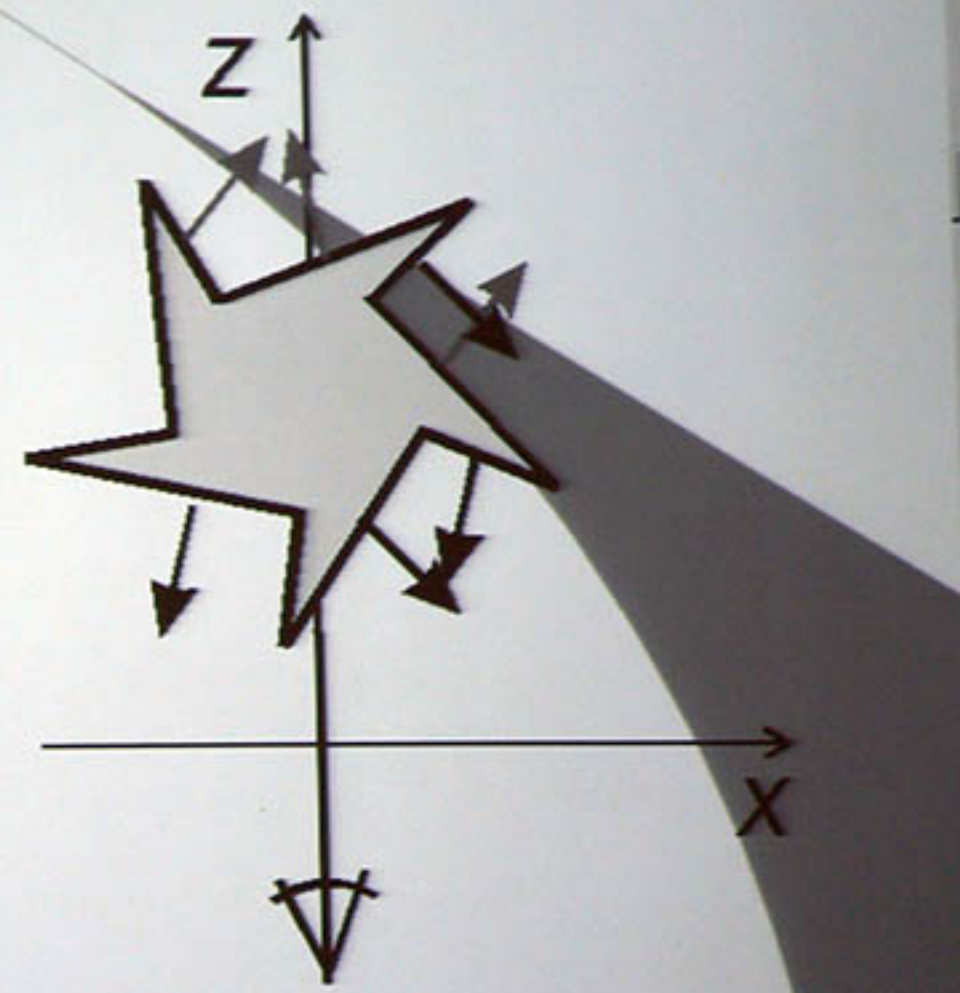
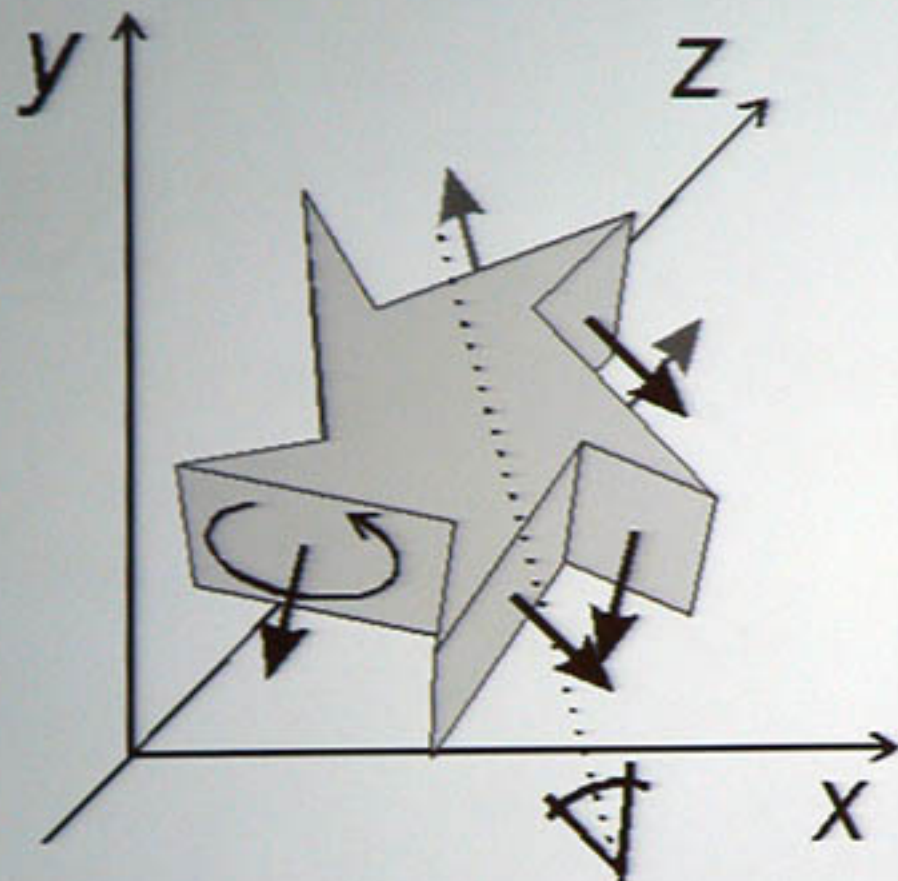
- wyznacz te części obiektu, których rzut nie jest zasłonięty przez inne części tego lub innych obiektów;
- narysuj tę część obiektu

}





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

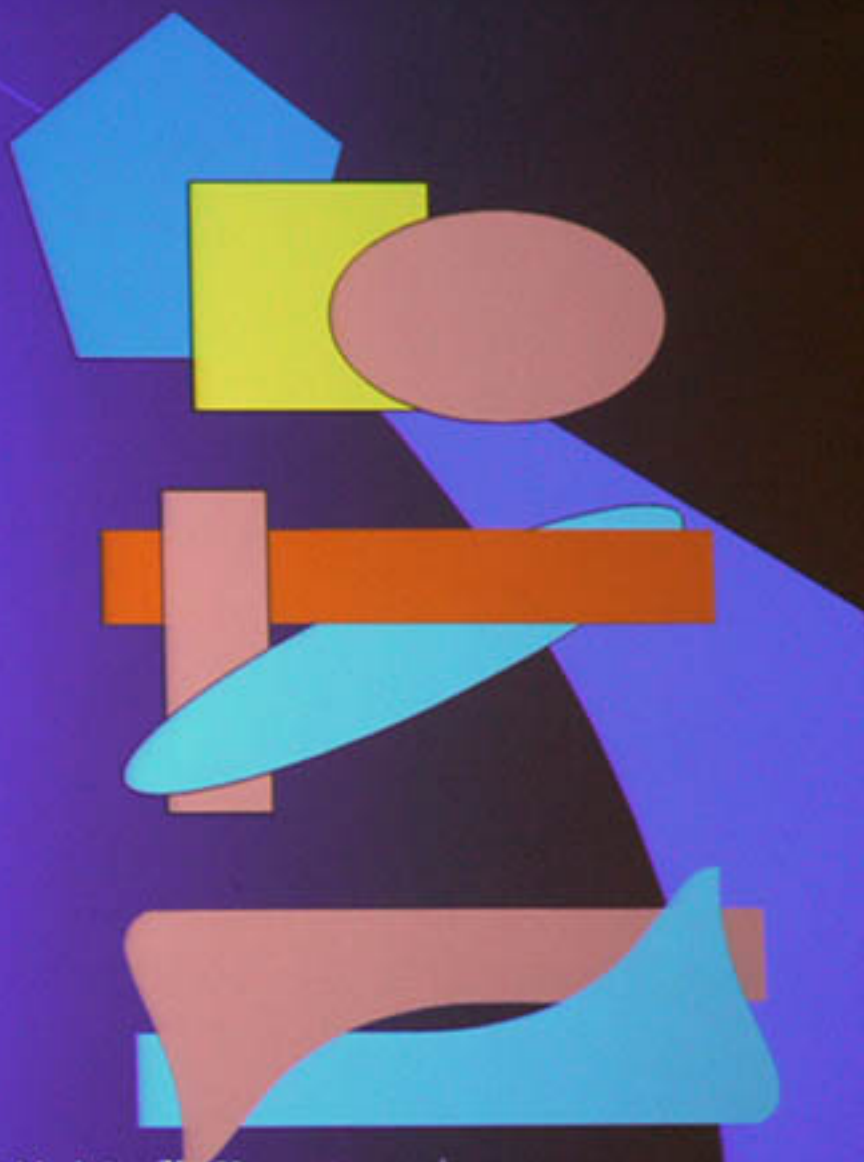
Algorytm Malarski (sortowanie ścian)

Problemy związane z zasłanianiem 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.

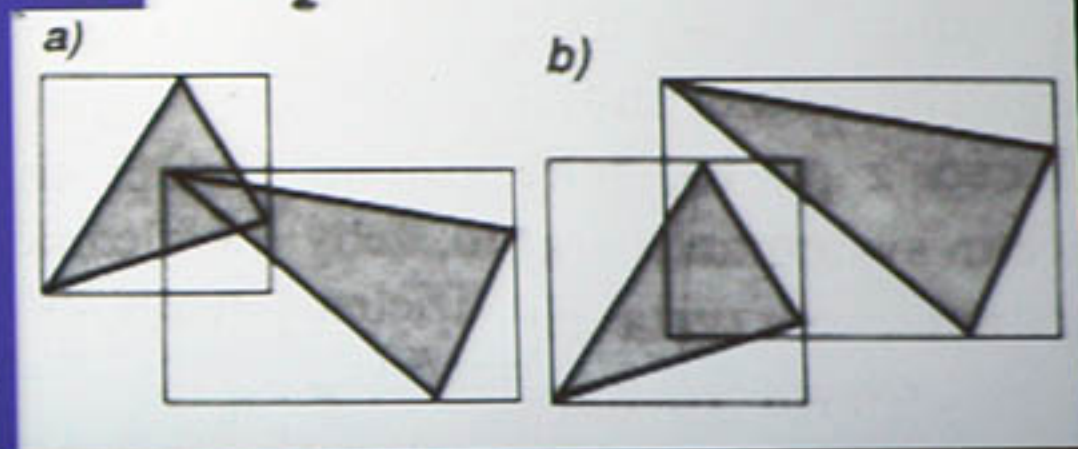
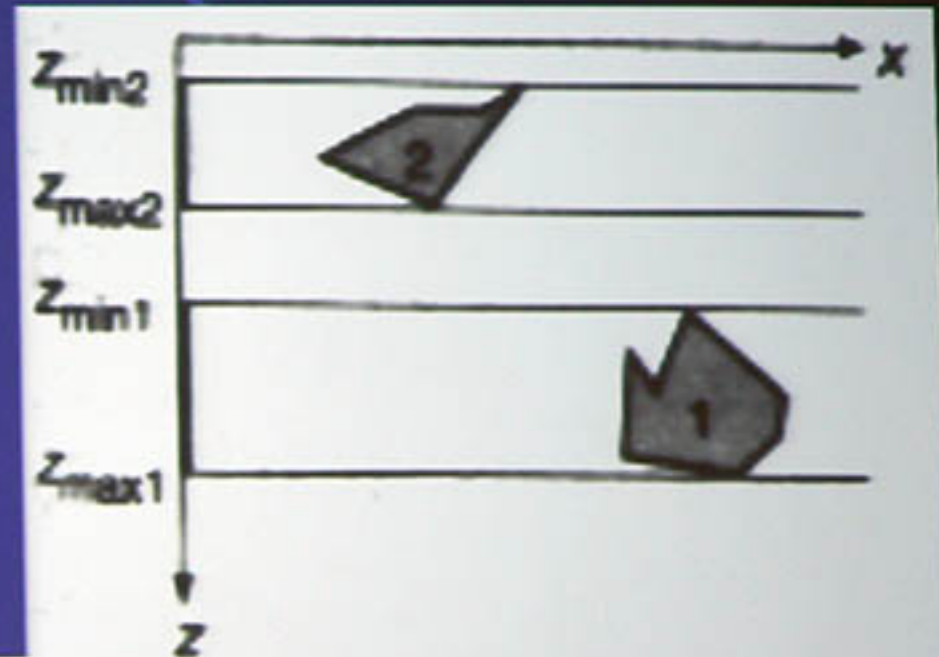
Relacje zasłaniania ścian

- Zasłanianie częściowe
 - A zasłania B,
 - B zasłania C,
 - Ale A nie zasłania C
- Wzajemne zasłanianie
 - A zasłania B,
 - B zasłania C,
 - C zasłania A
- Wzajemne zasłanianie dwóch ścian
 - A zasłania B,
 - B zasłania A



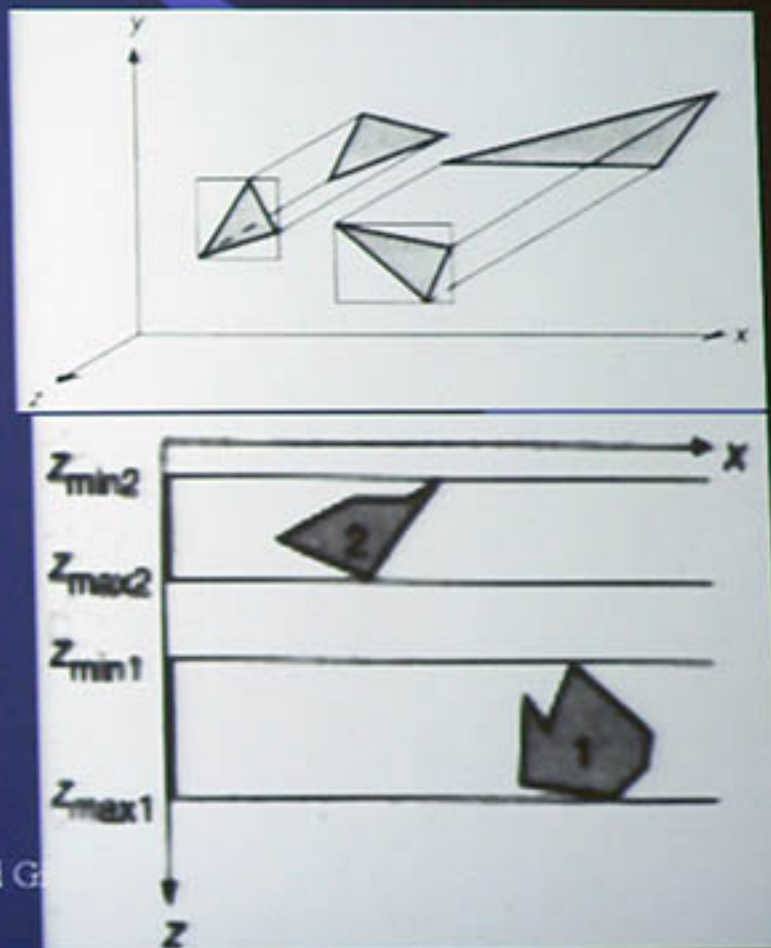
Efektywne obliczanie zasłaniania

- Obliczanie w 2D czy 3D
- Ograniczenia obiektów
 - x, y, z - ograniczenia
 - xy -ograniczenia prostokąty ograniczające
 - xyz - ograniczenia (bryły ograniczające)



Efektywne obliczanie zasłaniania

- Jeśli na rzutni prostokąty ograniczające się nie przecinają to żadna z dwóch brył nie zasłania drugiej
- *Testowanie minmax*
Jeśli wzdłuż osi z (kierunku patrzenia) wszystkie wierzchołki pierwszej bryły leżą przed wszystkimi wierzchołkami 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 (pkt 1.)
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

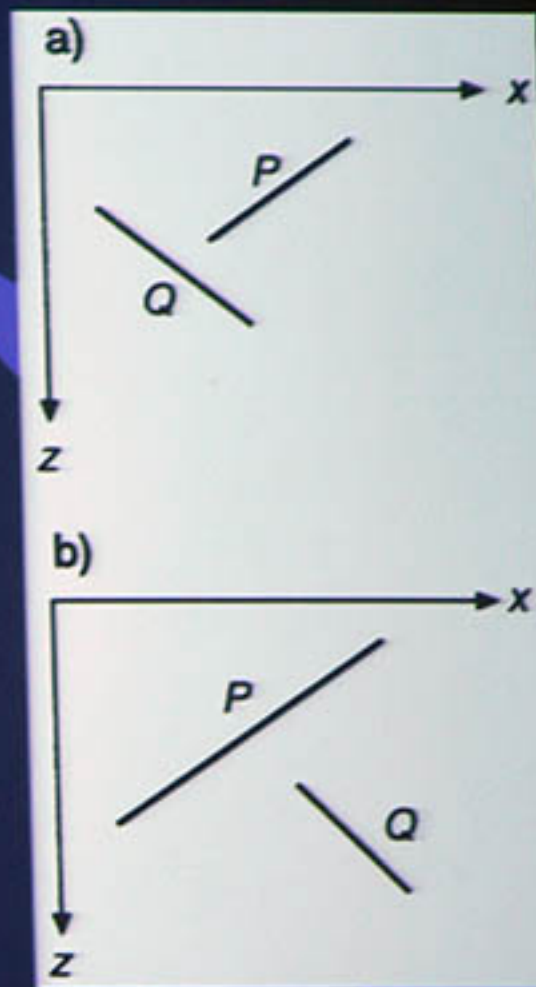
Algorytm sortowania ścian (c.d.)

C) czy ścian P leży całkowicie po niewidocznej dla obserwatora stronie ściany Q

D) czy Q jest całkowicie po tej stronie P co obserwator

E) Czy są rozłączne rzuty ścian na płaszczyznę xy

4. Jeśli żaden z testów A-E nie jest spełniony to zamieniamy ściany P i Q i badamy ponownie C i D

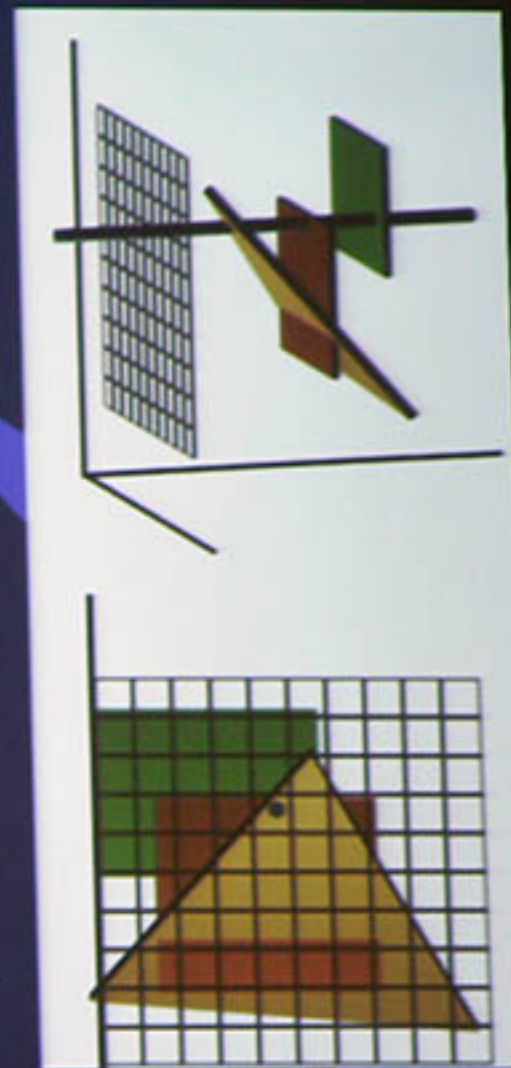


Algorytm Z-Bufora (bufora głębokości)

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) $Z_{\text{buf}}[x, y] := 0$;
- Dla każdego wielokąta {
 - Dla każdego piksela rzutu wielokąta {
 - $z :=$ wartość współrzędnej z wielokąta dla piksela (x, y)
 - if $(z \geq Z_{\text{buf}}[x, y])$ {
 - $Z_{\text{buf}}[x, y] = z$; WritePiksel(x, y , kolor wielokąta)

Zalety algorytmu

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