Wyznaczanie powierzchni widocznych

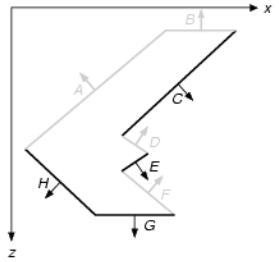
- Dla danego zbioru obiektów i specyfikacji parametrów widzenia należy określić, które linie albo powierzchnie obiektów są widoczne, tak żeby było możliwe wyświetlenie tylko linii albo powierzchni widocznych
- Algorytm z precyzją obrazową:

```
for (każdy piksel obrazu) {
  wyznacz obiekt najbliższy obserwatora, który jest napotkany przez
    promień rzutowania przechodzący przez piksel;
  narysuj piksel o odpowiedniej barwie;}
```

- Algorytm z precyzją obiektową

```
for (każdy obiekt) {
  wyznacz te części obiektu, których rzut nie jest zasłonięty przez inne
    części tego lub innych obiektów;
  narysuj te części z wykorzystaniem odpowiedniej barwy;}
```

Wybieranie tylnych ścian

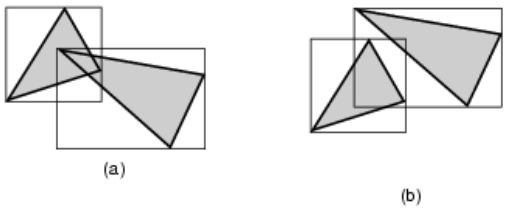


Założenie: wszystkie wielokąty zostały tak zdefiniowane, że ich wektory normalne są skierowane na zewnątrz wielościanu

Jeżeli kąt pomiędzy wektorem normalnym wielokąta a wektorem skierowanym do obserwatora przekracza 90 stopni, to wielokąt nie jest widoczny

Badanie kąta może odbyć się przy użyciu iloczynu skalarnego wektorów (iloczyn dodatni -> wielokąt widoczny)

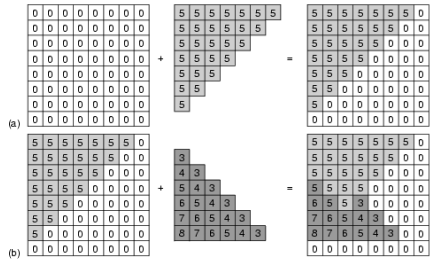
Prostokąty ograniczające



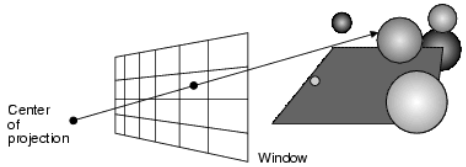
Jeżeli spełniony jest jeden z poniższych warunków, to wielokąty są rozłączne:
$x_{Amax}<x_{Bmin}$, $x_{Bmax}<x_{Amin}$, $y_{Amax}<y_{Bmin}$, $y_{Bmax}<x_{Amin}$

Algorytm z-bufora

```
for (y=0; y<YMAX; y++) {
  for (x=0; x<XMAX; x++) {
    WritePixel(x, y, BACKGROUND_VALUE);
    WriteZ(x, y, 0);
  }
}
for (każdy wielokąt) {
  for (każdy piksel w rzucie wielokąta) {
    pz = wartość z wielokąta dla piksela (x,y);
    if (pz>=ReadZ(x, y)) {
      WriteZ(x, y, pz);
      WritePixel(x, y, barwa wielokąta dla piksela (x,y));
    }
  }
 }
}
```



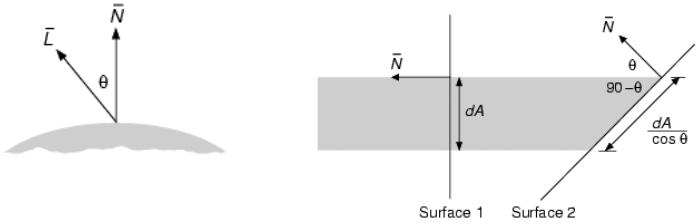
Metoda śledzenia promieni




```
for (każdy przeglądany wiersz obrazu) {
  for (każdy piksel w przeglądanym wierszu) {
     wyznaczenie promienia ze środka rzutowania przez piksel;
     for (każdy obiekt w scenie) {
       if (obiekt jest przecinany i jest najbliższy z dotychczas rozważanych)
          rejestrowanie przecięcia i nazwy obiektu;
     }
     ustawienie barwy piksela zgodnie z barwą najbliższego przeciętego obiektu;
  }
}
```

Lokalny model oświetlenia (1/4)

Odbicie rozproszone światła (dyfuzyjne, lambertowskie)



$$I = I_p k_d \cos\theta = I_p k_d \left(\overline{N} \cdot \overline{L}\right)$$

Z uwzględnieniem światła otoczenia:

$$I = I_a k_a + I_p k_d \left(\overline{N} \cdot \overline{L}\right)$$

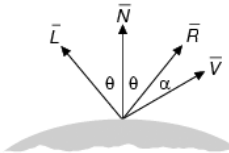
Model oświetlenia (2/4)

Tłumienie źródła światła

$$I = I_a k_a + f_{att} I_p k_d \left(\overline{N} \cdot \overline{L}\right)$$

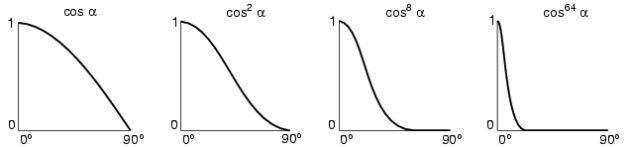
$$f_{att} = \frac{1}{d_L^2}$$

Odbicie lustrzane źródła światła (Phonga)



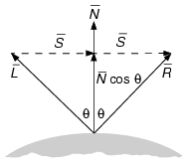
$$I = I_a k_a + f_{att} I_p k_d \left(\overline{N} \cdot \overline{L} + k_s \cos^n \alpha\right) =$$
$$= I_a k_a + f_{att} I_p k_d \left(\overline{N} \cdot \overline{L} + k_s \left(\overline{R} \cdot \overline{V}\right)^n\right)$$

Model oświetlenia (3/4)



Różne wartości $\cos^n\alpha$ w modelu Phonga

Obliczanie wektora odbicia



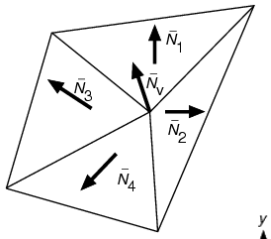
$$\overline{R} = \overline{N}\cos\theta + \overline{S}$$
$$\left|\overline{S}\right| = \sin\theta$$
$$\overline{S} = \overline{N}\cos\theta - \overline{L}$$
$$\overline{R} = 2\overline{N}\cos\theta - \overline{L}$$
$$\overline{R} = 2\overline{N}\left(\overline{N} \cdot \overline{L}\right) - \overline{L}$$

Model oświetlenia (4/4)

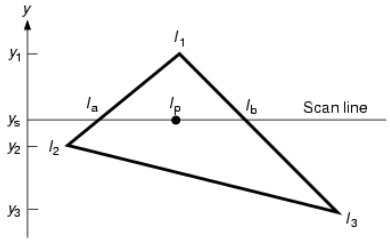
Wiele źródeł światła

$$I = I_a k_a + \sum_{i=1}^{m} f_{att_i} I_{p_i} k_d \left(\overline{N} \cdot \overline{L_i} + k_s \left(\overline{R_i} \cdot \overline{V} \right)^n \right)$$

Cieniowanie Gourauda



1. Interpolacja wektorów normalnych w wierzchołkach
2. Wyznaczenie oświetlenia w wierzchołkach
3. Interpolacja oświetlenia wewnątrz wielokąta

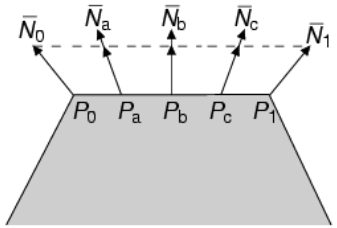


$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 - (I_1 - I_3)\frac{y_1 - y_s}{y_1 - y_3}$$

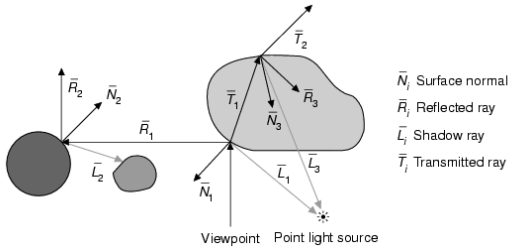
$$I_p = I_b - (I_b - I_a)\frac{x_b - x_p}{x_b - x_a}$$

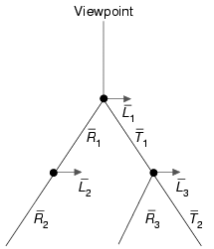
Cieniowanie Phonga



1. Interpolacja wektorów normalnych w wierzchołkach
2. Interpolacja wektorów normalnych wewnątrz wielokąta
3. Wyznaczenie oświetlenia wewnątrz wielokąta

Rekursywna metoda śledzenia promieni





Globalny model oświetlenia (model Whitteda)

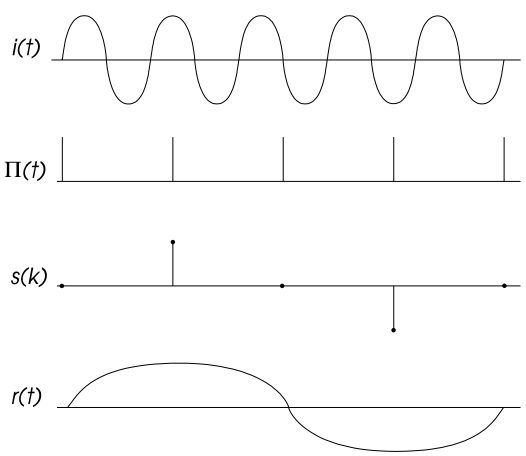
$$I = I_a k_a + \sum_{i=1}^{m} f_{att_i} I_{p_i} k_d \left(\overline{N} \cdot \overline{L_i} + k_s \left(\overline{R_i} \cdot \overline{V} \right)^n \right) + k_s I_r + k_t I_t$$

Rekursywna metoda śledzenia promieni
Pseudokod

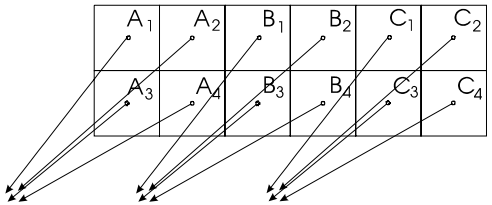
```
begin
for każdy piksel P do {
zbuduj promień R wychodzący z punktu widzenia obserwatora
        i przechodzący przez piksel P
Intensity = trace ( R )
wypełnij piksel P kolorem i intensywnością 'Intensity'
}
end

procedure trace ( Ray )
  begin
    for każdy obiekt E sceny do{
      znajdź punkt przecięcia promienia Ray z obiektem E
        (jeżeli taki istnieje)
      }
    if promień Ray nie przecina żadnego obiektu then {
      return (kolor i intensywność tła)
            }
      else {
        Określ najbliższy punkt przecięcia I
        return (Model_oświetlenia(I, trace(promień_lustrzany),
                trace(promień_załamany))
            }
  end
```

Antialiasing (1/2)

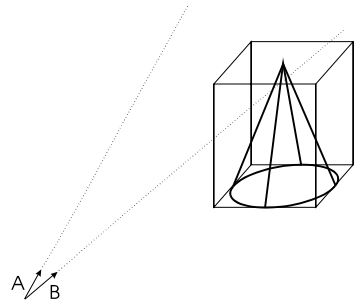


Antialiasing (2/2)
Supersampling



$\frac{1}{4}\sum_{i=1}^{4} A_i$	$\frac{1}{4}\sum_{i=1}^{4} B_i$	$\frac{1}{4}\sum_{i=1}^{4} C_i$

Otoczkowanie



n - ilość wszystkich śledzonych promieni,
nb - ilość promieni przecinających otoczkę obiektu,
Cb - koszt znajdowania przecięcia promienia z otoczką,
Co - koszt znajdowania przecięcia z opisywanym obiektem geometrycznym,

C1 = n * Co

C2 = n * Cb + nb * Co

Warunek efektywności: C2 < C1

n * Cb + nb * Co < n * Co

nb / n < 1 - Cb/Co

Light Caching

```
procedure Model_oświetlenia_z_light_caching
                (I, intensity_lustrzane, intensity_załamane)
begin
intensity = Ia
for każde źródło światła L do {
zbuduj promień R wychodzący z punktu I w kierunku L
if not promień R przecina L.cache then {
  for każdy obiekt E do {
    sprawdź, czy R przecina E
          }
  if R nie przecina żadnego obiektu then {
          intensity = intensity + (N ° L)
          }
    else {
       L.cache = pierwszy przecinany obiekt
          }
    }
}
intensity = intensity + intensity_lustrzane
intensity = intensity + intensity_załamane
return (intensity)
end
```

Znajdowanie przecięcia promienia z płaszczyzną

Promień

$$\vec{P} = \vec{O} + t\vec{D}$$

$$0 \le t < \infty$$

Płaszczyzna

$$\left(\vec{P} - \vec{P'}\right) \cdot \vec{N} = 0$$

$$ax + by + cz + d = 0$$

Rozwiązanie równania:

$$\left(\vec{P} - \vec{P'}\right) \cdot \vec{N} = \left(\vec{O} + t\vec{D} - \vec{P'}\right) \cdot \vec{N} = 0$$

$$t = -\frac{\left(\vec{O} - \vec{P'}\right) \cdot \vec{N}}{\vec{D} \cdot \vec{N}}$$

Znajdowanie przecięcia promienia ze sferą

Promień

$$\vec{P} = \vec{O} + t\vec{D}$$

$$0 \le t < \infty$$

Sfera

$$\left(\vec{P} - \vec{C}\right)^2 - R^2 = 0$$

$$\left(\vec{O} - t\vec{D} - \vec{C}\right)^2 - R^2 = 0$$

Rozwiązanie równania

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$at^2 + bt + c = 0$$

$$a = \vec{D}^2 = 1, b = 2\left(\vec{O} - \vec{C}\right) \cdot \vec{D}, c = \left(\vec{O} - \vec{C}\right)^2 - R^2$$

Modelowanie krzywych – krzywe trzeciego stopnia

$$Q(t) = \left[x(t)\, y(t)\, z(t)\right]^T$$

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

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

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

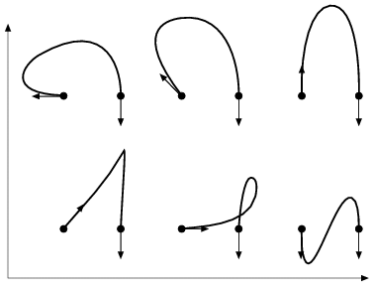
$$0 \le t \le 1$$

Krzywa Hermite'a

Określona przez dwa punkty końcowe (P_1, P_4) i przez dwa wektory styczne w tych punktach końcowych (R_1, R_4)

Macierz geometrii Hermite'a: $G_H = [P_1 P_4 R_1 R_4]$

$$Q(t) = (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4$$



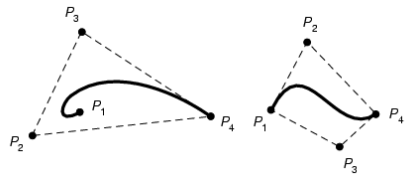
Krzywa Béziera

Określona przez dwa punkty końcowe (P_1, P_4) i przez dwa inne punkty (P_2, P_3), które mają wpływ na wektory styczne w punktach końcowych

Macierz geometrii Béziera: $G_B = [P_1 P_2 P_3 P_4]$

$$R_1 = Q'(0) = 3(P_2 - P_1), \quad R_4 = Q'(1) = 3(P_4 - P_3)$$

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



Powierzchnia Béziera