



Ray Tracer cz.2

Michał Chwesiuk

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie
Wydział Informatyki

9 Maja 2017



Plan zajęć laboratoryjnych

Równanie
oświetlenia
Obliczanie
wektorów
normalnych

- Podłączanie bibliotek
Zapis danych do pliku graficznego
- Generowanie promienia pierwotnego
Import sceny z pliku
- Algorytm ray tracingu
Obliczanie przecięć z kulą
- Obliczanie przecięć z trójkątem
Korekcja Gamma
- Równanie oświetlenia
Obliczanie wektorów normalnych
Wyznaczanie cieni
- Rekurencyjne śledzenie promieni odbitych
- Antyaliasing



Model Blinna-Phong'a

Kolor obiektu

$$i_{out} = i_{amb} + \sum_{k=1}^N (i_{dif}^k + i_{spe}^k)$$

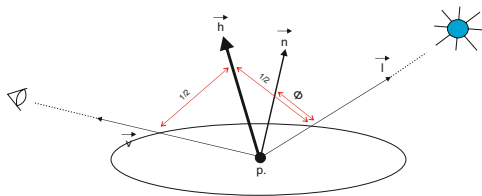
$$i_{amb} = m_{amb} \cdot s_{amb}$$

$$i_{dif} = m_{dif} \cdot s_{dif} * (|\vec{l}| \cdot |\vec{n}|)$$

$$i_{spe} = m_{spe} \cdot s_{spe} * (|\vec{n}| \cdot |\vec{h}|)^{m_{shi}}$$

$$\vec{h} = \frac{|\vec{l}| + |\vec{v}|}{2}$$

$$p = ray_{pos} + t \cdot ray_{dir}$$





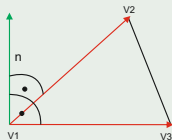
Wektory normalne

Wyznaczanie
cieni

Rekurencyjne
śledzenie
promieni
odbitych

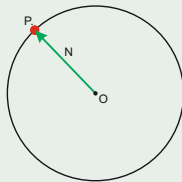
Trójkąt

$$\vec{n} = |(\vec{V}_2 - \vec{V}_1) \times (\vec{V}_3 - \vec{V}_1)|$$



Kula

$$\vec{n} = |\vec{P} - \vec{O}|$$





Wyznaczanie cieni

Wyznaczanie
cieni

Rekurencyjne
śledzenie
promieni
odbitych

Funkcja *trace_ray*

```
function COLOR trace_ray(RAY, &COLOR, &ENERGY) {  
    INTERSECTION = find_intersection(RAY, true)  
    if(INTERSECTION == null) return COLOR  
    generate SHADOW-Ray(INTERSECTION POINT)  
    INTERSECTION = find_intersection(SHADOW-Ray, false)  
    if(INTERSECTION == null) {  
        compute new_color(solve light equation)  
        COLOR = COLOR + ENERGY * new_color    }  
    decrease ENERGY  
    if(ENERGY < small value) return COLOR  
    generate SECONDARY-Ray  
    trace_ray(SECONDARY-Ray, COLOR, ENERGY)  
    return COLOR }
```



Rekurencyjne śledzenie promieni odbitych

Wyznaczanie
cieni

Rekurencyjne
śledzenie
promieni
odbitych

Funkcja *trace_ray*

```
function COLOR trace_ray(RAY, &COLOR, &ENERGY) {  
    INTERSECTION = find_intersection(RAY, true)  
    if(INTERSECTION == null) return COLOR  
    generate SHADOW_RAY(INTERSECTION POINT)  
    INTERSECTION = find_intersection(SHADOW_RAY, false)  
    if(INTERSECTION == null) {  
        compute new_color(solve light equation)  
        COLOR = COLOR + ENERGY * new_color    }  
    decrease ENERGY  
    if(ENERGY < small value) return COLOR  
    generate SECONDARY_RAY  
    trace_ray(SECONDARY_RAY, COLOR, ENERGY)  
    return COLOR }
```



Promień odbity

Promień odbity

$$\begin{aligned} ray_{pos} &= \vec{P} \\ ray_{dir} &= \vec{v}' \\ \vec{v}' &= \vec{v} - (2 \cdot \vec{v} \cdot \vec{n}) \cdot \vec{n} \end{aligned}$$

