GRAFIKA KOMPUTEROWA

Wykład 6: grafika trójwymiarowa (rendering)

Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Materiały opracowane w ramach zadania 15 „Modyfikacja międzywydziałowych studiów I stopnia na kierunku Inżynieria Biomedyczna”
projektu „NERW PW. Nauka - Edukacja - Rozwój - Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

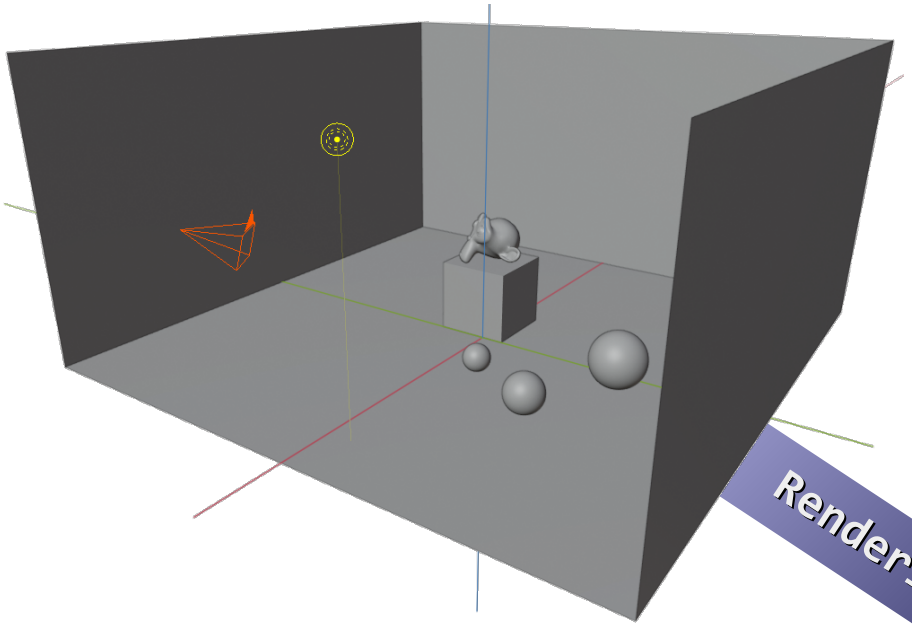
Fundusze Europejskie
Wiedza Edukacja Rozwój

Politechnika Warszawska

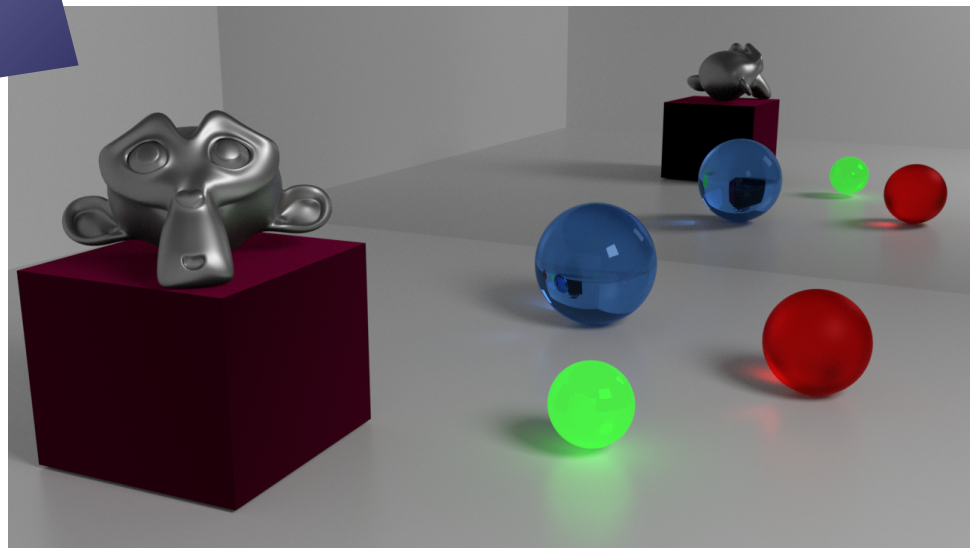
Unia Europejska
Europejski Fundusz Społeczny

Rendering 3D

Celem renderingu jest wygenerowanie dwuwymiarowego obrazu na podstawie trójwymiarowych modeli obiektów. Proces tworzenia wizualizacji musi uwzględniać zarówno źródła światła zdefiniowane w obrębie sceny, jak i pozycję obserwatora.

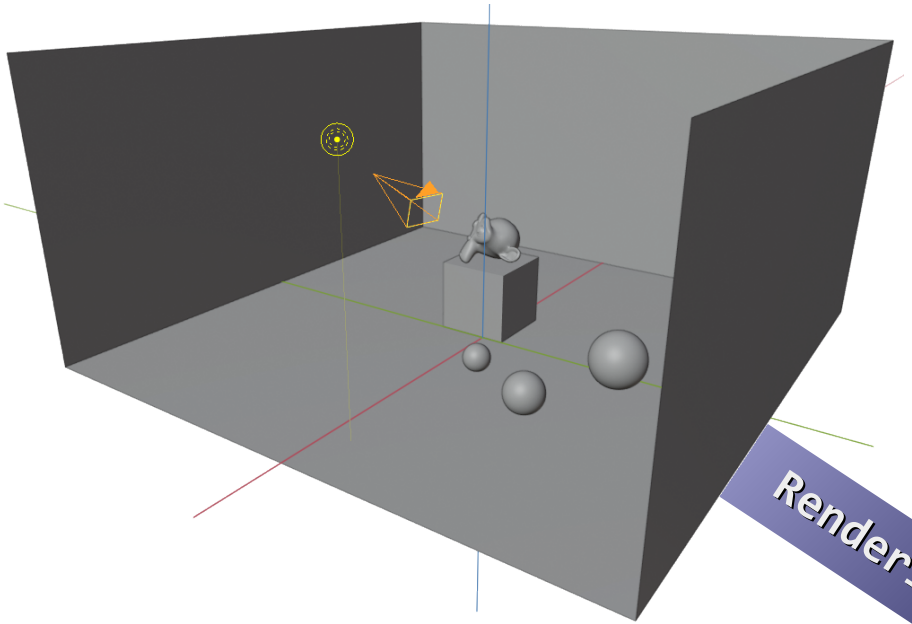


Rendering

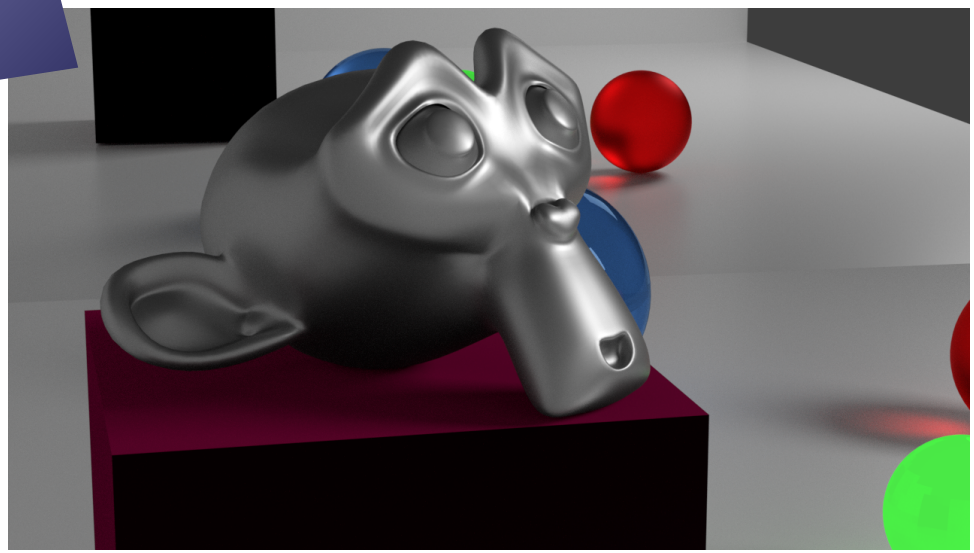


Rendering 3D

Celem renderingu jest wygenerowanie dwuwymiarowego obrazu na podstawie trójwymiarowych modeli obiektów. Proces tworzenia wizualizacji musi uwzględniać zarówno źródła światła zdefiniowane w obrębie sceny, jak i pozycję obserwatora.

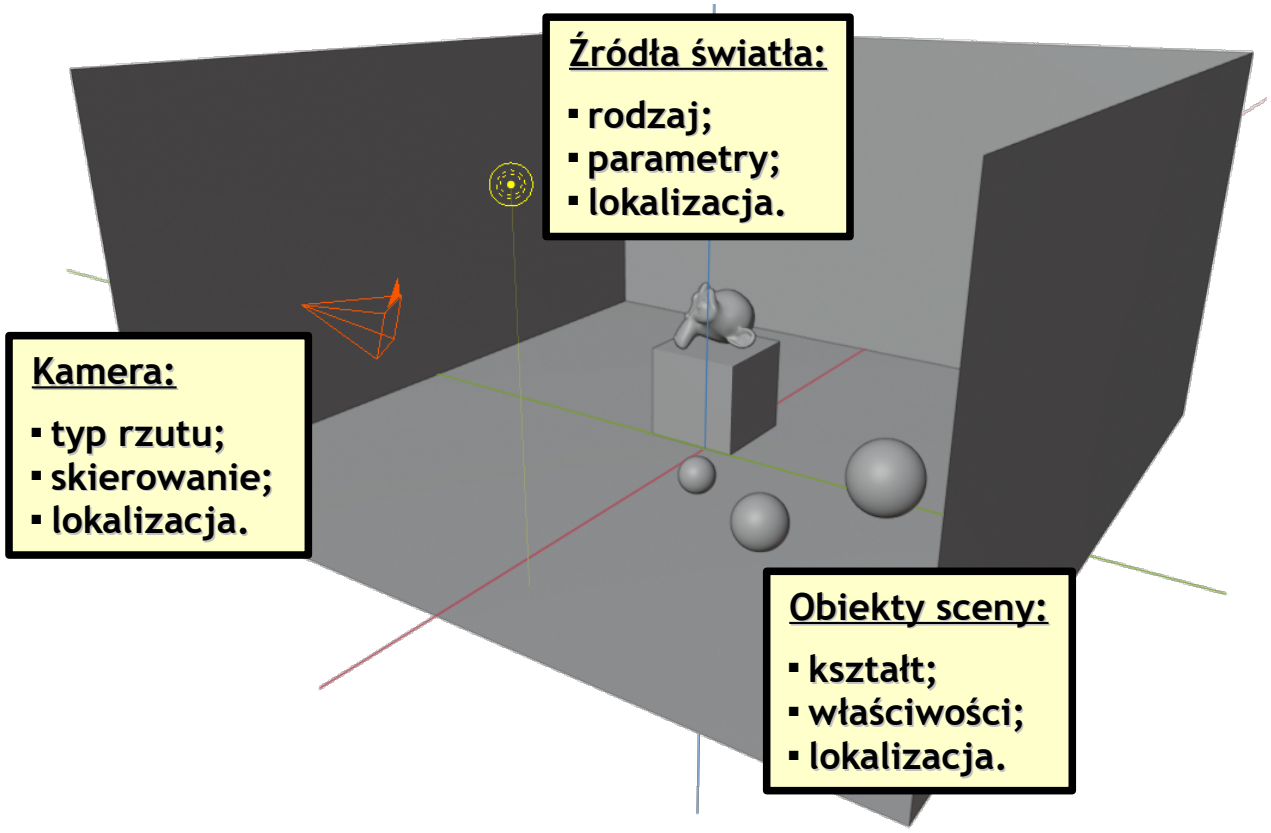


Rendering



Danymi wejściowymi dla etapu renderingu są:

- modele opisujące geometrię i parametry powierzchni obiektów;
- pozycje i wymiary obiektów określone w układzie współrzędnych sceny;
- definicje źródeł światła: ich rozmieszczenie, skierowanie, rodzaj, barwa, …;
- definicja kamery: pozycja, orientacja, sposób rzutowania.

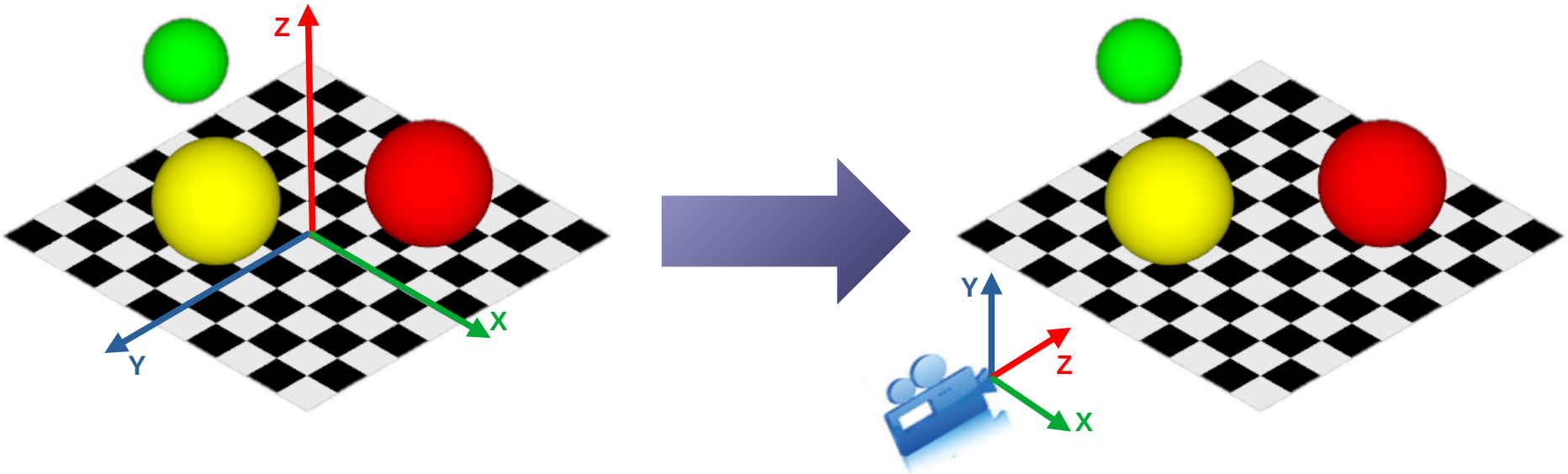


Dane wyjściowe mają postać bitmapowego obrazu o zadanych rozmiarach.

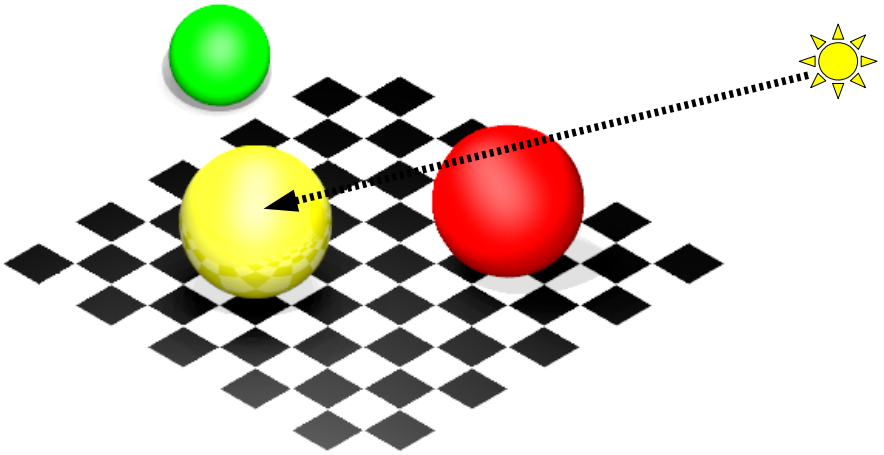
Rendering 3D

Rendering wymaga rozwiązania szeregu problemów, z których najważniejsze to:

- przeniesienie obiektów sceny do układu współrzędnych związanego z kamerą;

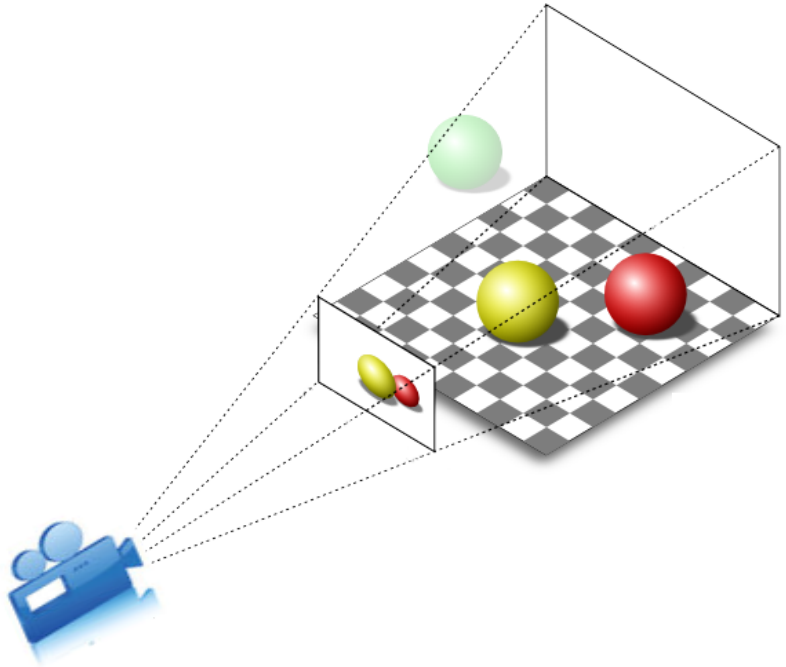


- określenie wpływu źródeł światła na kolory powierzchni obiektów;

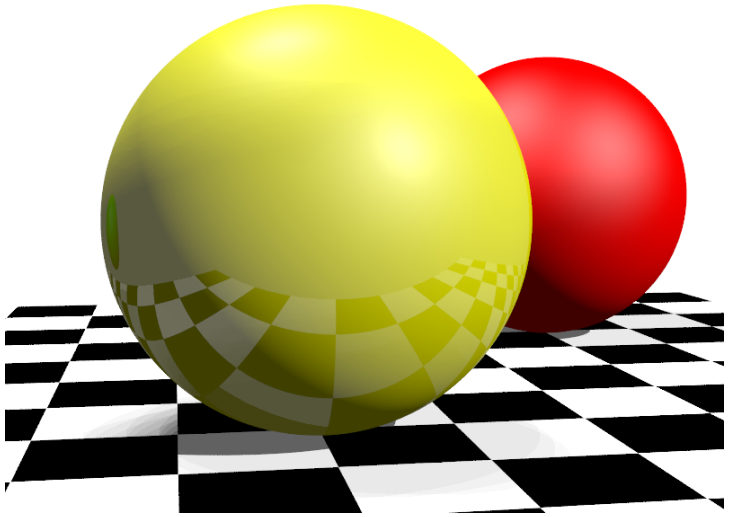


Rendering 3D

- rzutowanie trójwymiarowych obiektów na płaszczyznę;

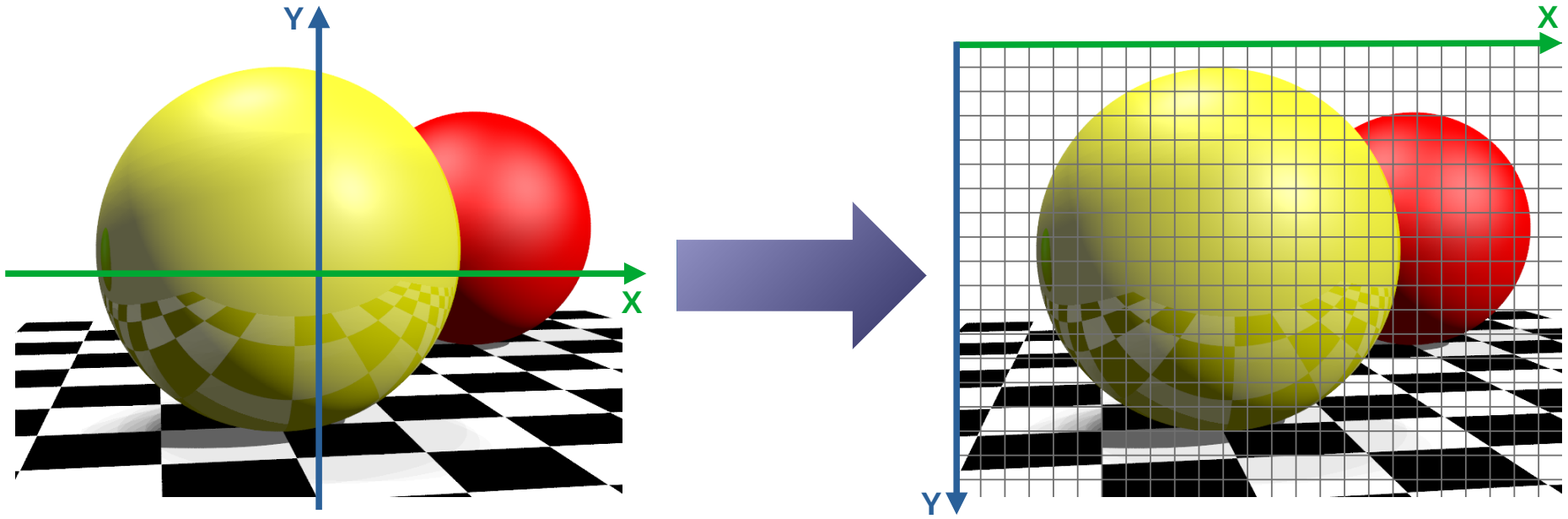


- eliminacji obiektów i powierzchni niewidocznych;

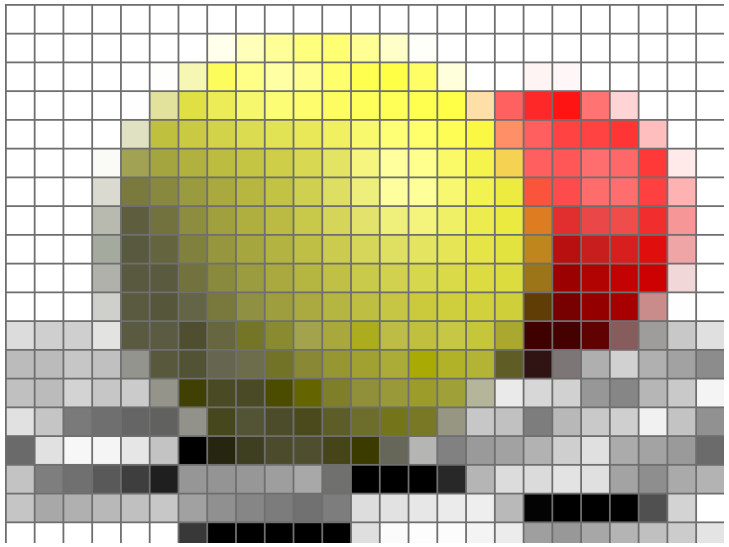


Grafika 3D: rendering

- przeniesienie rzutów obiektów do układu współrzędnych związanego z obrazem;



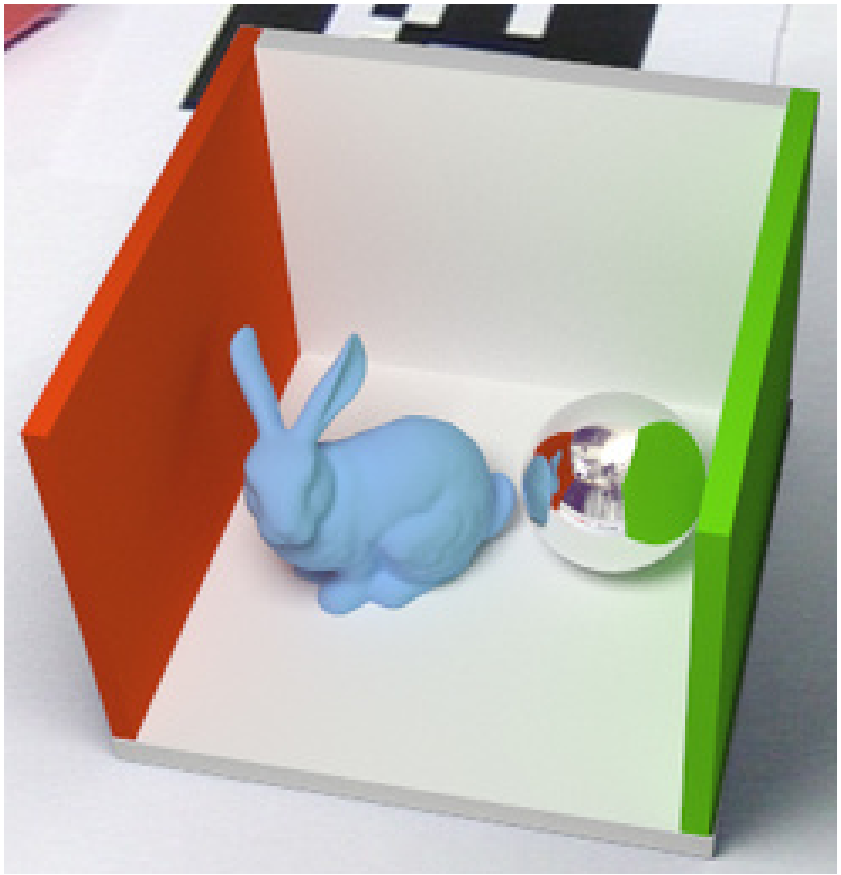
- rasteryzacja rzutów obiektów w celu generacji obrazu wyjściowego.



Rendering 3D: oświetlenie lokalne wobec globalnego

Algorytmy stosowane podczas renderingu mogą znacząco różnić się, w zależności od tego, w jakim celu i w jakich warunkach generowane są obrazy.

Podczas tworzenia obrazów statycznych lub wstępnie przygotowywanych animacji (klatek filmu) można użyć kosztownych obliczeniowo algorytmów, rozwiązujących problem oświetlenia globalnego, czyli takiego opisu rozchodzenia się światła, który uwzględnia wzajemne oddziaływania pomiędzy wszystkimi obiektami sceny.

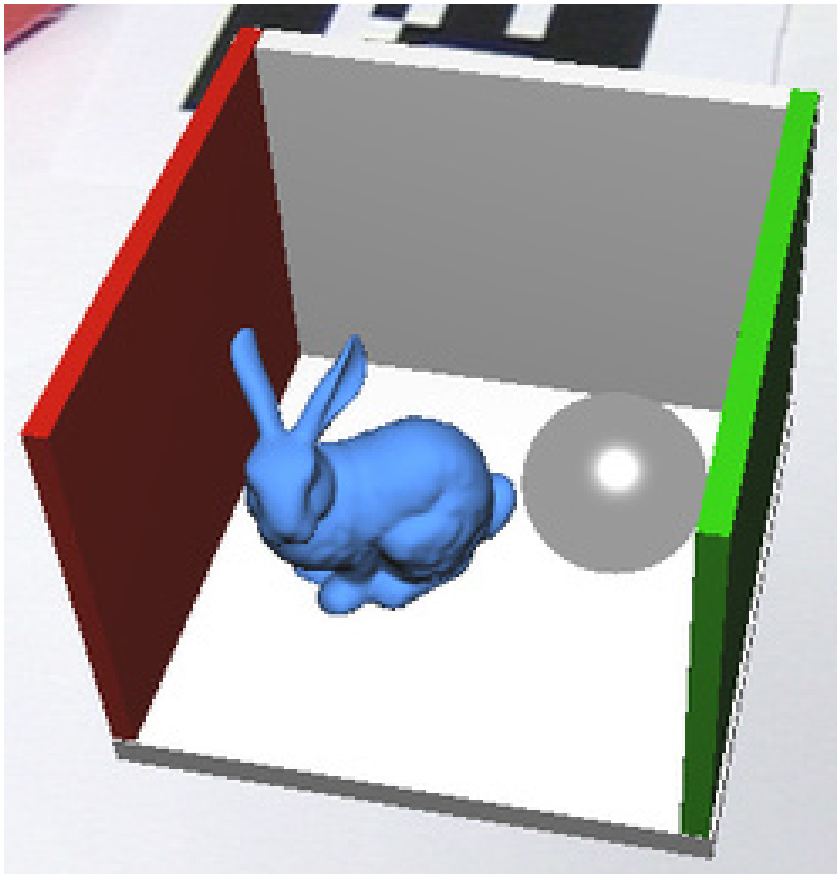


[http://www.peterkan.com/phdthesis.php]

Rendering 3D: oświetlenie lokalne wobec globalnego

Algorytmy stosowane podczas renderingu mogą znacząco różnić się, w zależności od tego, w jakim celu i w jakich warunkach generowane są obrazy.

Tworzenie animacji na bieżąco (np. w grach) wymaga zachowania ścisłego reżimu czasowego, co prowadzi do konieczności uproszczenia obliczeń. Wówczas rendering często uwzględnia jedynie bezpośredni wpływ źródeł światła na lokalne właściwości poszczególnych obiektów.



[http://www.peterkan.com/phdthesis.php]

GRAFIKA KOMPUTEROWA

Wykład 6: grafika trójwymiarowa (rendering – technika śledzenia promieni)

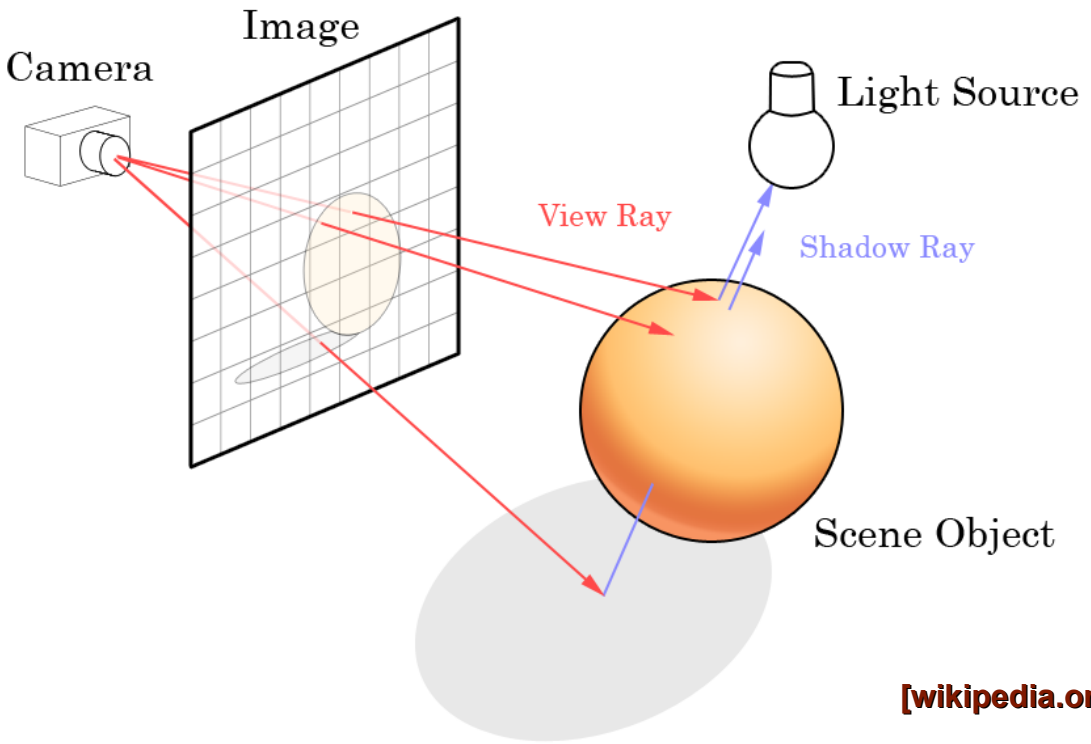
Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Rendering 3D: metoda śledzenia promieni

Jednym z najczęściej stosowanych sposobów rozwiązywania problemu globalnego oświetlenia jest **technika śledzenia promieni (*ray tracing*)**, która pozwala uzyskać wysokiej jakości obrazy o dużym stopniu realizmu.

Główna idea metody polega na prowadzeniu przez kolejne piksele obrazu promieni biegnących od punktu obserwacji w kierunku sceny. Kolor danego piksela jest wyznaczany na podstawie symulacji interakcji przechodzącego przezeń promienia świetlnego z trójwymiarowymi modelami obiektów.

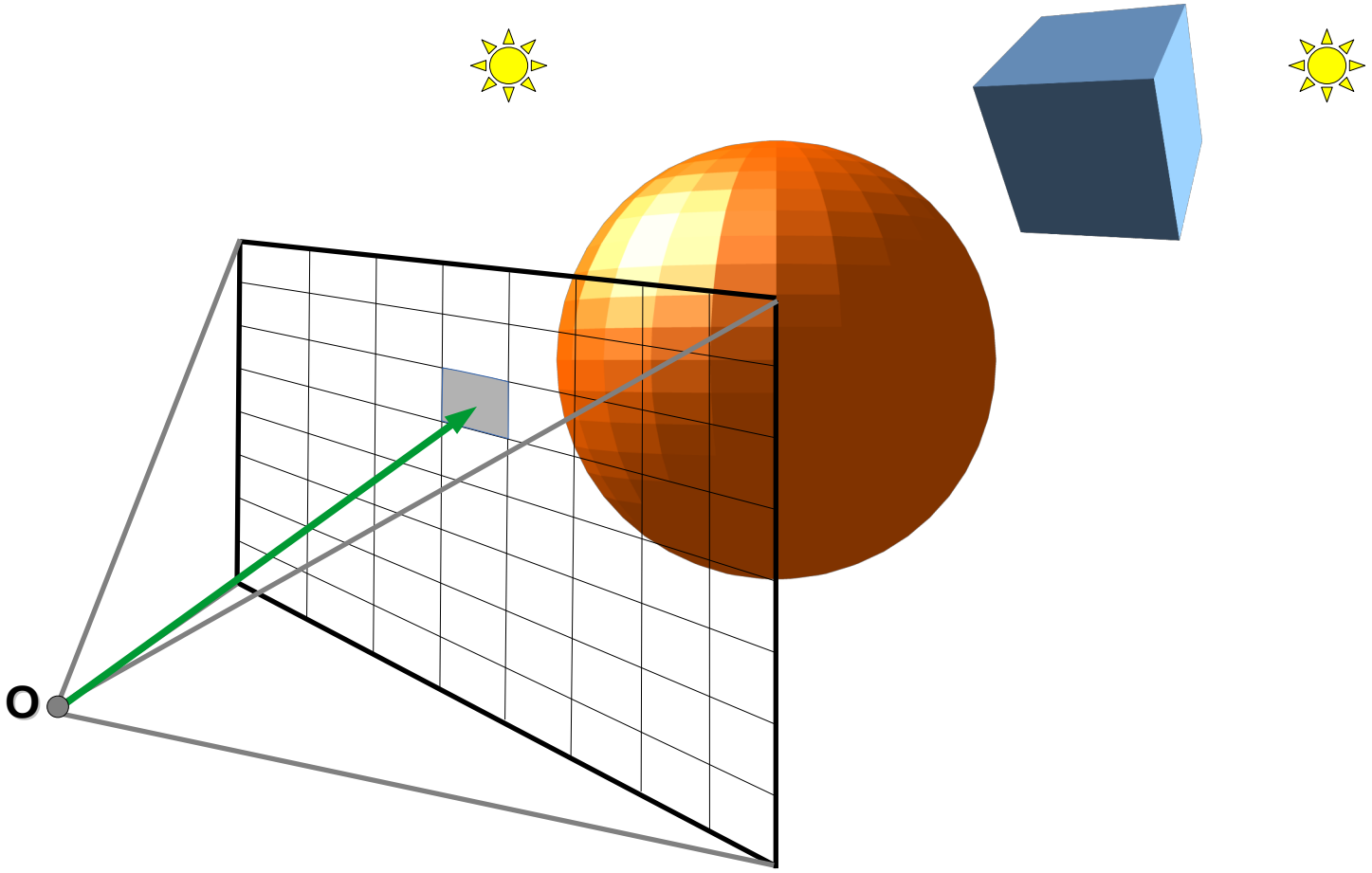


[wikipedia.org]

Rendering 3D: metoda śledzenia promieni

W najprostszej odmianie metody uwzględniane są tylko promienie pierwotne, które biegną od punktu obserwacji, do pierwszego napotkanego obiektu sceny. Algorytm wyznaczania barwy pojedynczego piksela składa się z następujących kroków:

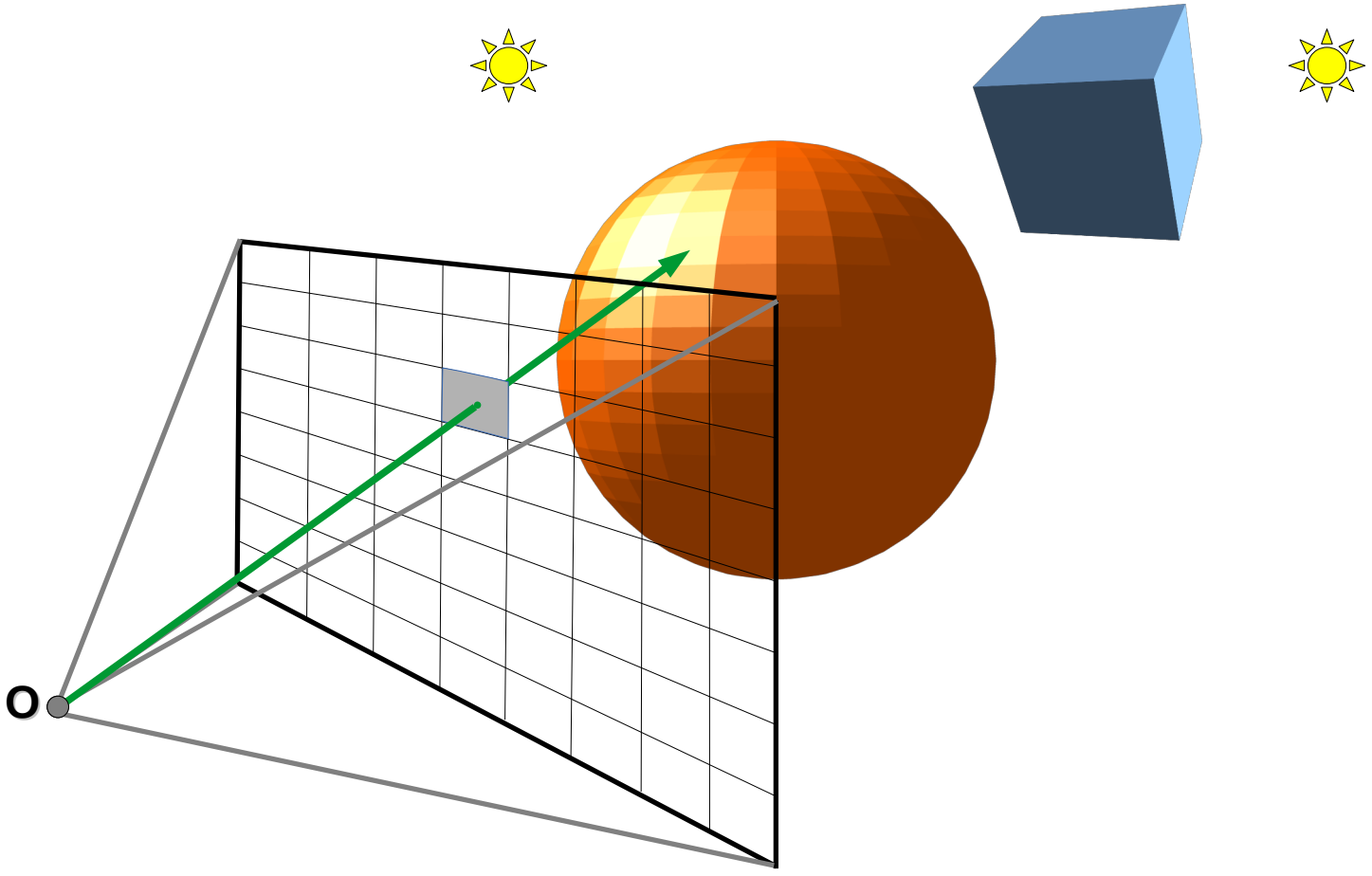
1. Z punktu, w którym znajduje się kamera wyprowadzany jest promień pierwotny, przecinający rzutnię w miejscu wyznaczonym przez współrzędne środka piksela.



Rendering 3D: metoda śledzenia promieni

W najprostszej odmianie metody uwzględniane są tylko promienie pierwotne, które biegną od punktu obserwacji, do pierwszego napotkanego obiektu sceny. Algorytm wyznaczania barwy pojedynczego piksela składa się z następujących kroków:

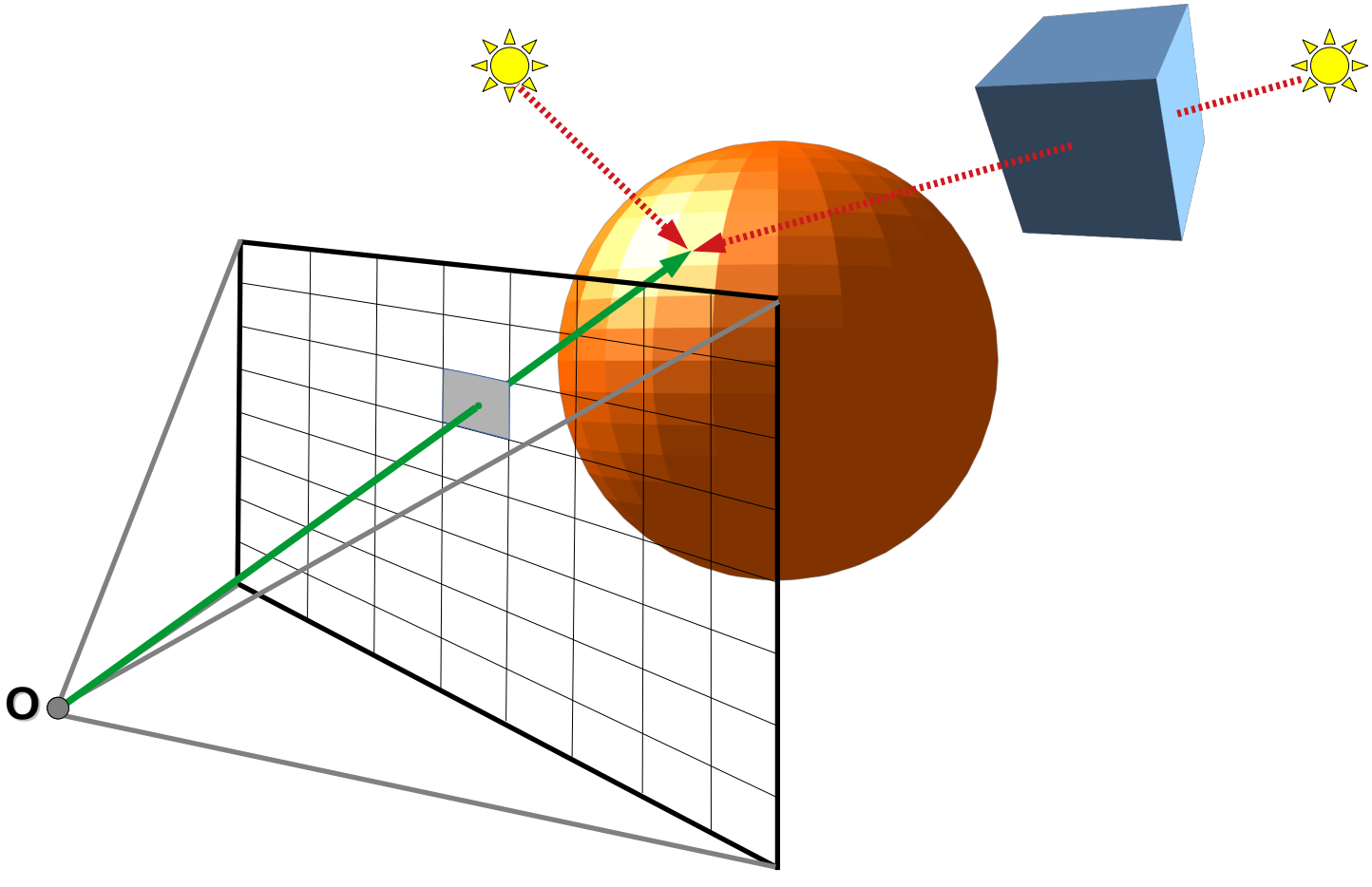
2. Znajdowany jest pierwszy punkt przecięcia promienia pierwotnego z najbliższym obiektem występującym w obrębie sceny.



Rendering 3D: metoda śledzenia promieni

W najprostszej odmianie metody uwzględniane są tylko promienie pierwotne, które biegną od punktu obserwacji, do pierwszego napotkanego obiektu sceny. Algorytm wyznaczania barwy pojedynczego piksela składa się z następujących kroków:

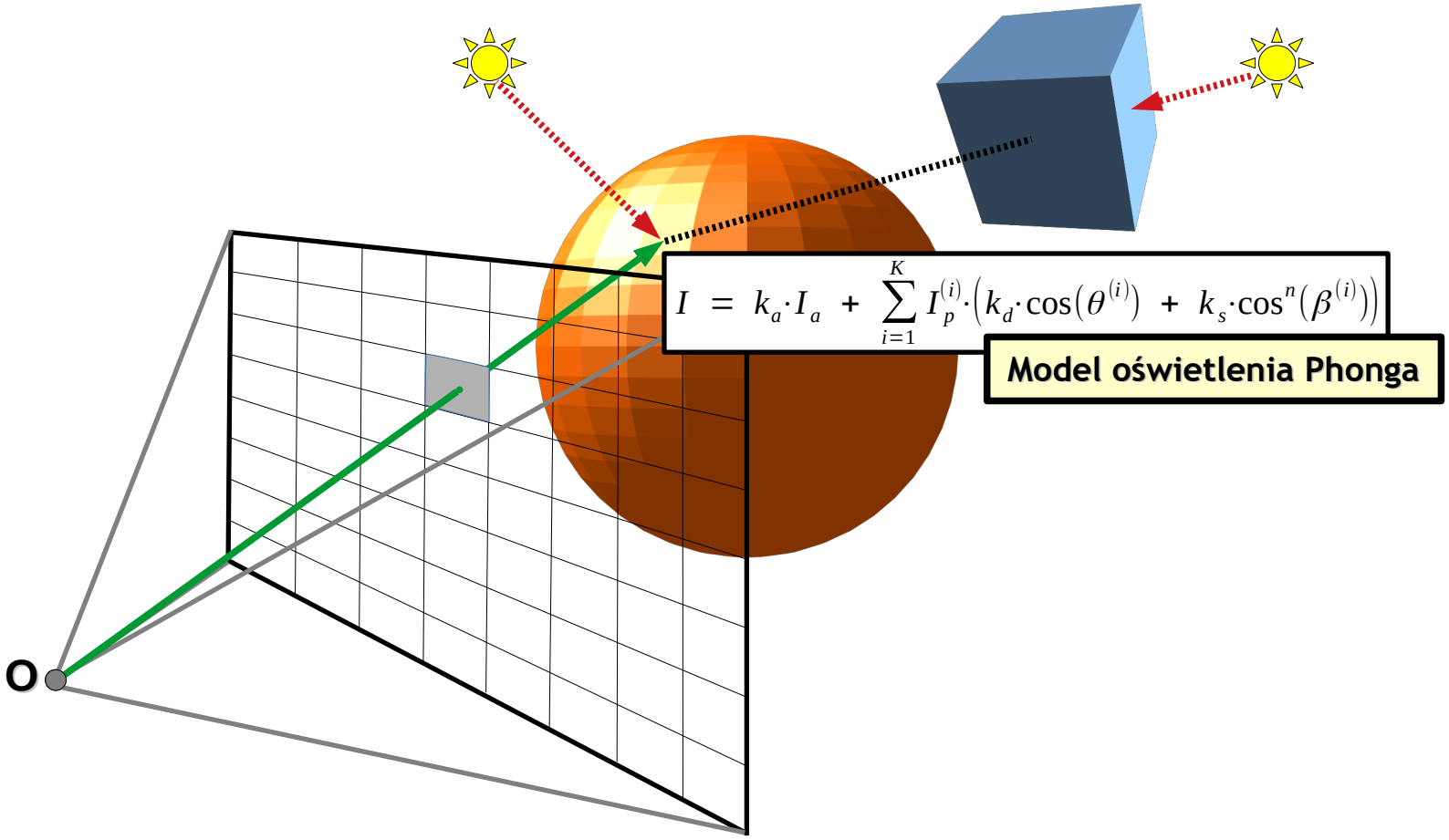
3. Testowana jest widoczność źródeł światła zdefiniowanych w scenie. W tym celu wyznaczane są promienie cienia, biegnące od źródeł, do punktu przecięcia.



Rendering 3D: metoda śledzenia promieni

W najprostszej odmianie metody uwzględniane są tylko **promienie pierwotne**, które biegną od punktu obserwacji, do pierwszego napotkanego obiektu sceny. Algorytm wyznaczania barwy pojedynczego piksela składa się z następujących kroków:

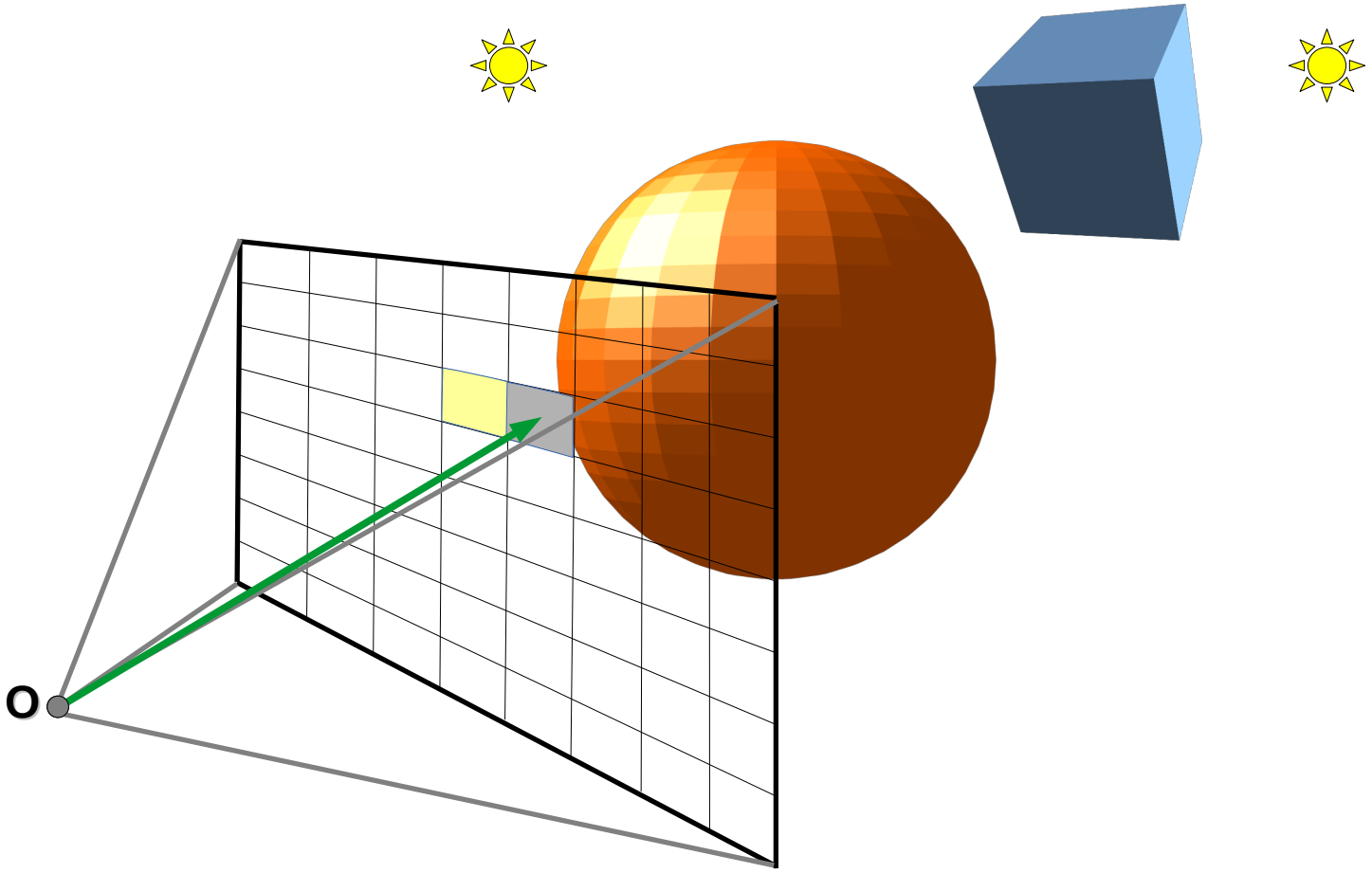
4. Przy użyciu **modelu oświetlenia** (np. modelu Phonga) określany jest kolor punktu. Bierze się przy tym pod uwagę właściwości obiektu i parametry źródeł światła. Niewidocznym źródłom przypisuje się kolor czarny, co pozwala tworzyć cienie.



Rendering 3D: metoda śledzenia promieni

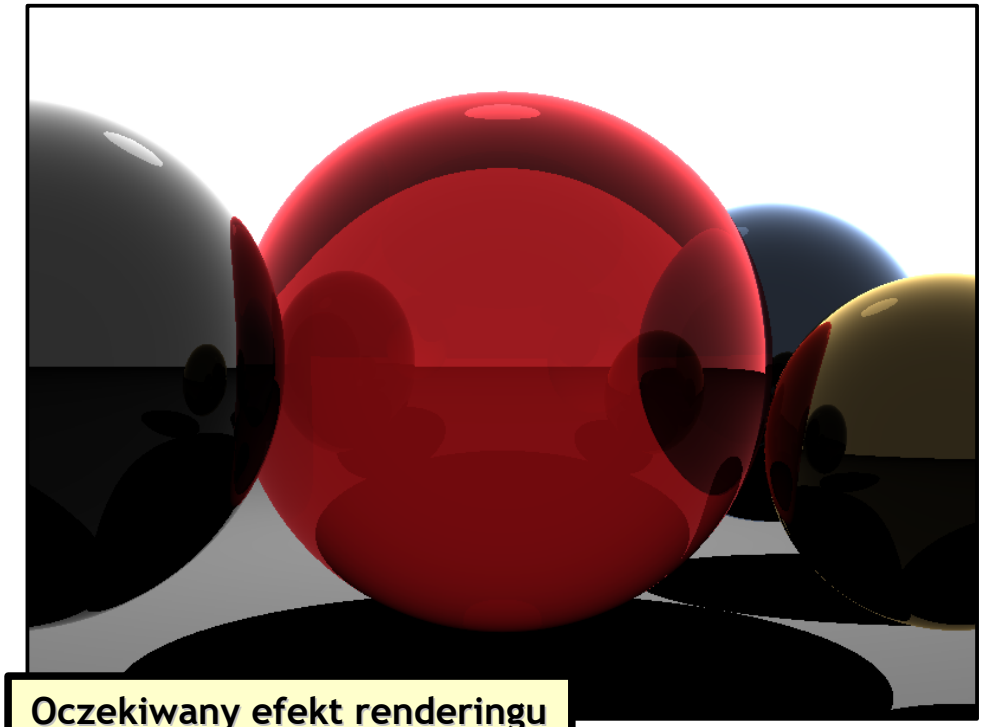
W najprostszej odmianie metody uwzględniane są tylko promienie pierwotne, które biegną od punktu obserwacji, do pierwszego napotkanego obiektu sceny. Algorytm wyznaczania barwy pojedynczego piksela składa się z następujących kroków:

5. Wyznaczony kolor staje się kolorem piksela, przez który prowadzony był promień, a cały proces jest powtarzany dla kolejnych pikseli.

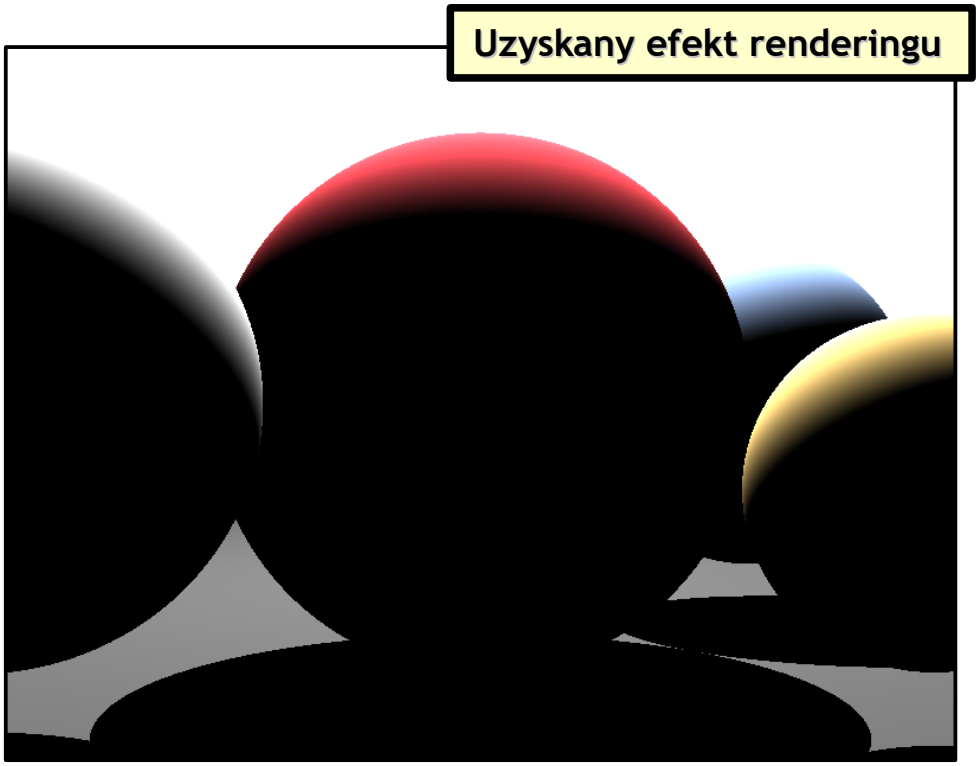


Rendering 3D: metoda śledzenia promieni

Przedstawiona powyżej podstawowa wersja algorytmu ma poważne ograniczenie: uwzględnia tylko oświetlenie padające bezpośrednio na obiekty przez co nie pozwala prawidłowo wizualizować powierzchni refleksyjnych i przezroczystych.



Oczekiwany efekt renderingu

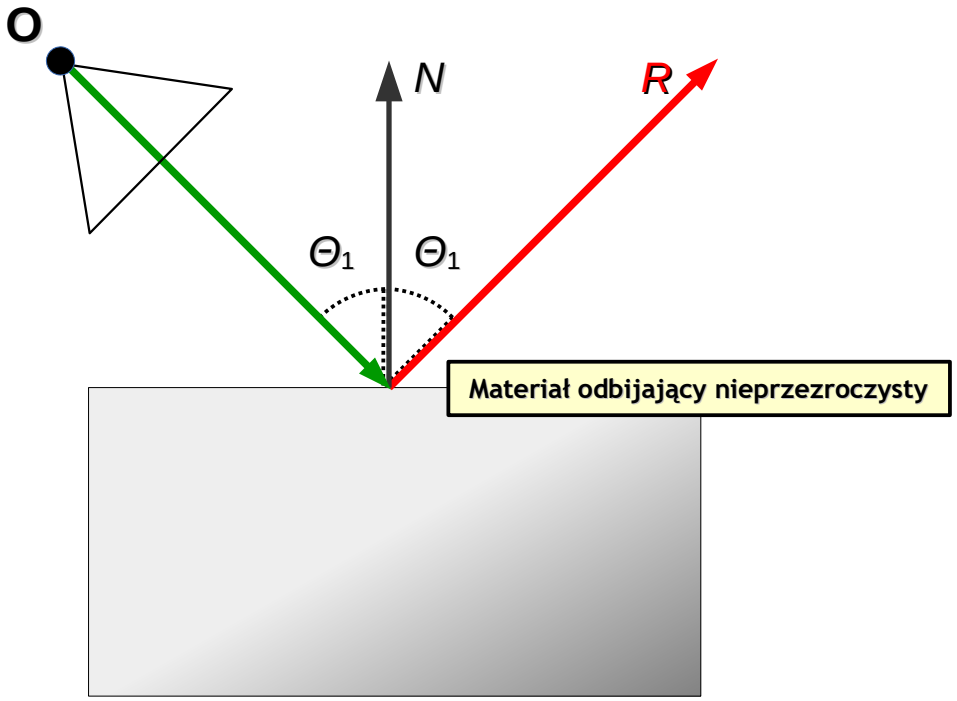


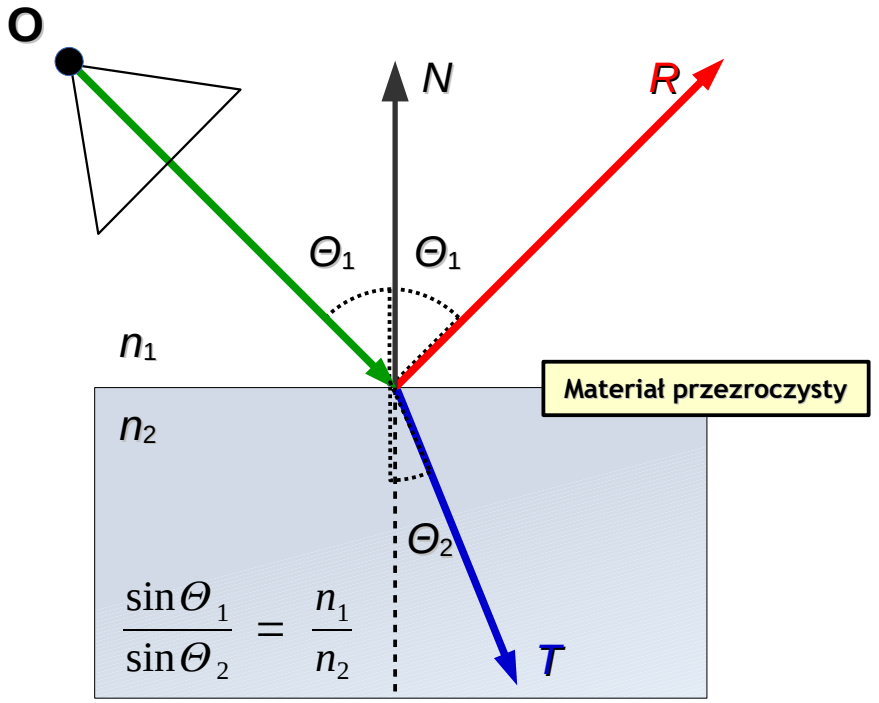
Uzyskany efekt renderingu

Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Dlatego w praktyce stosuje się **wersję rekurencyjną algorytmu śledzenia promieni**, która z punktu przecięcia promienia pierwotnego z powierzchnią złamującą lub odbijającą światło wysyła **promienie wtórne**.

Dla powierzchni refleksyjnych generowany jest promień **odbity** (*R*), a dla materiałów przezroczystych dodatkowo promień **załamany** (*T*). Ich kierunki wyznaczane są wobec wektora normalnego (*N*) zgodnie z prawami całkowitego odbicia i refrakcji.



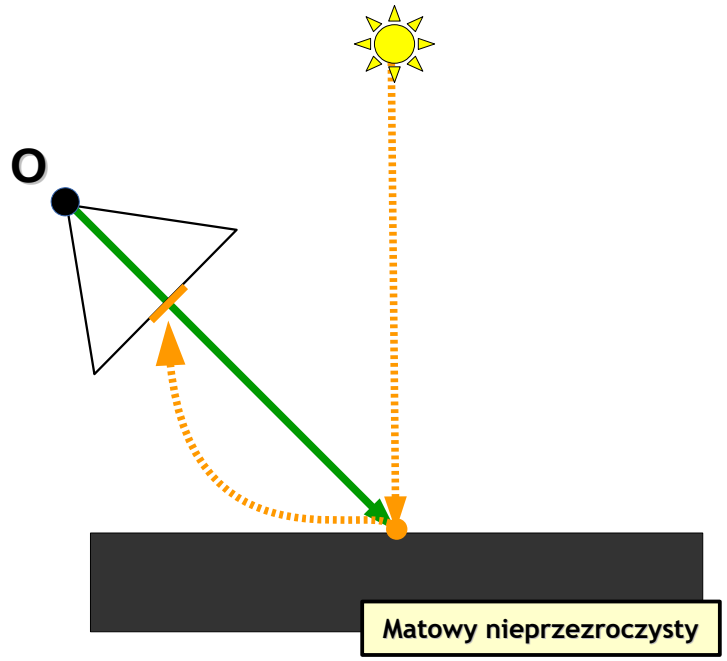


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

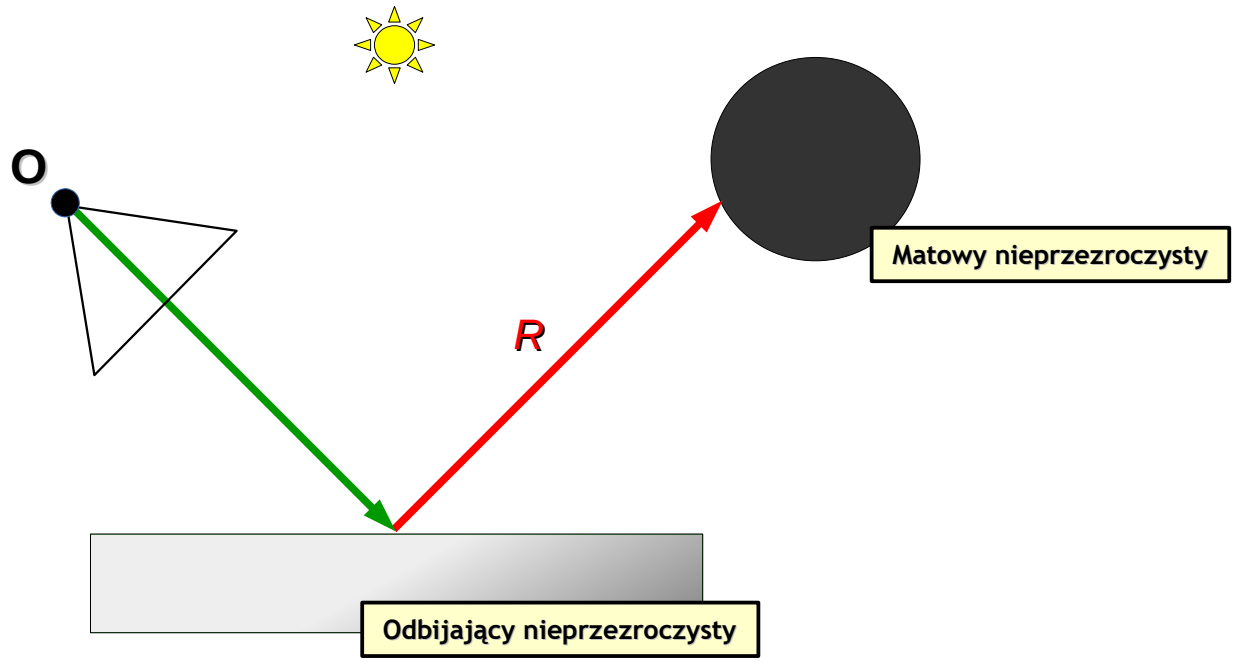


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). **Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne**.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

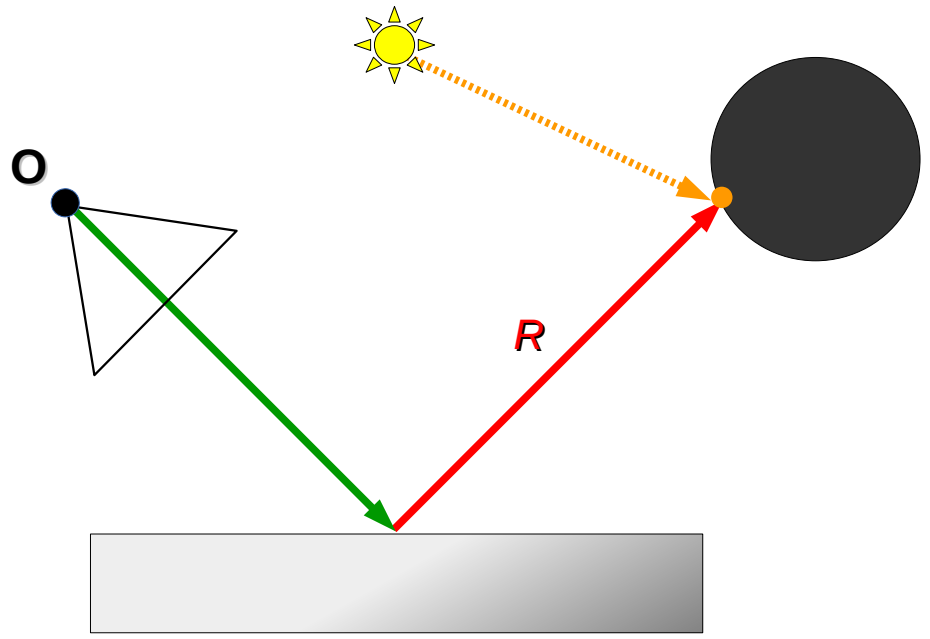


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

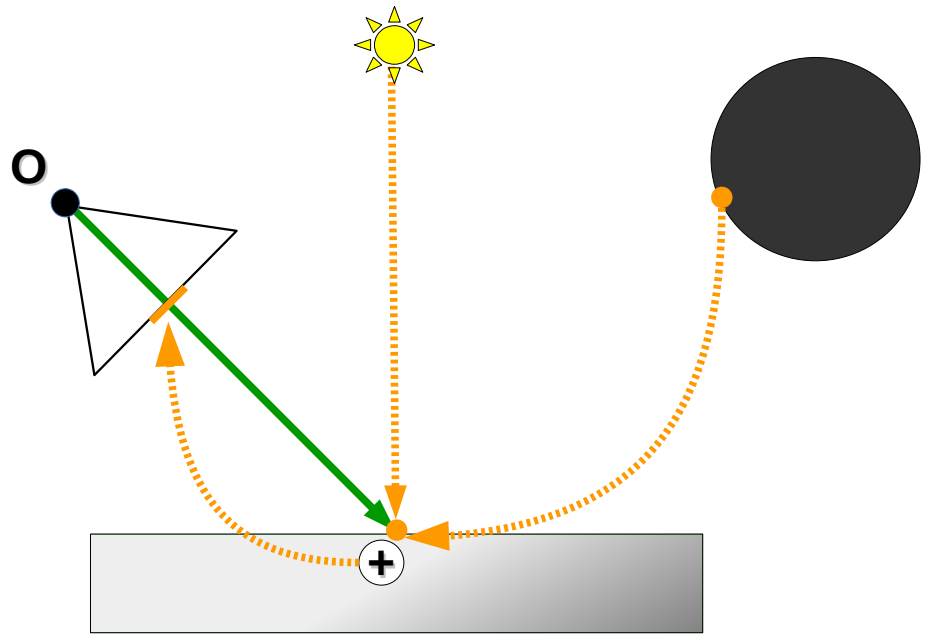


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

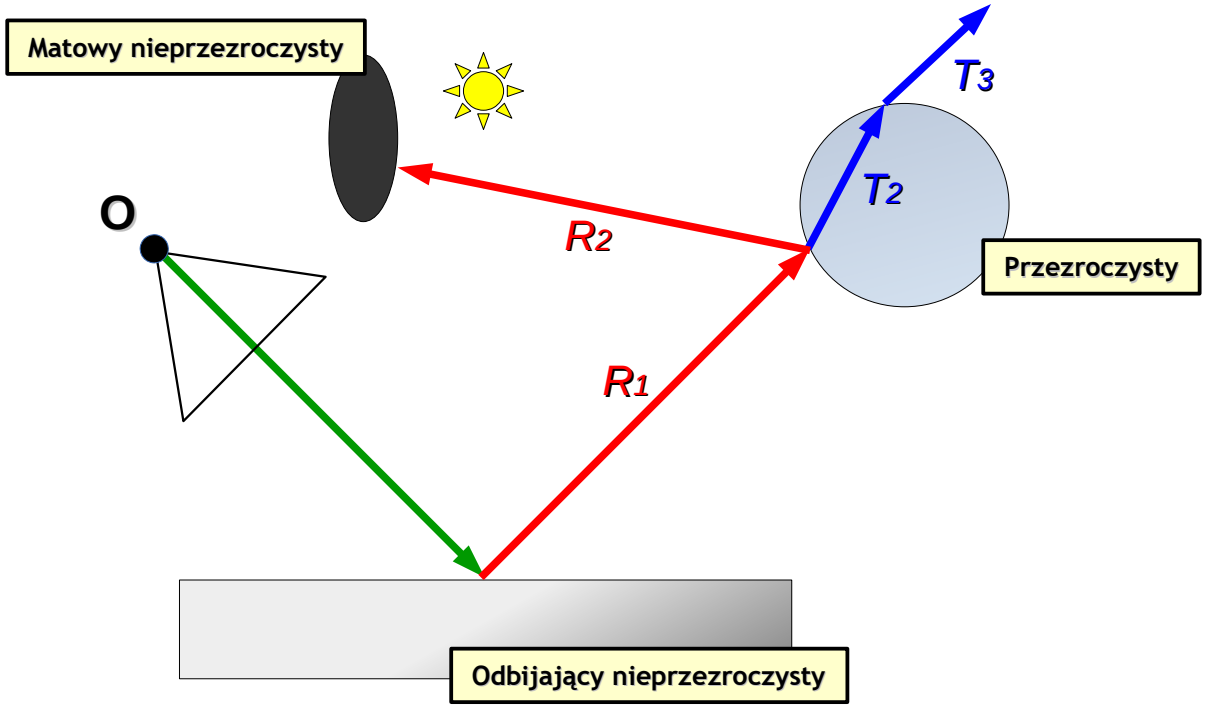


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

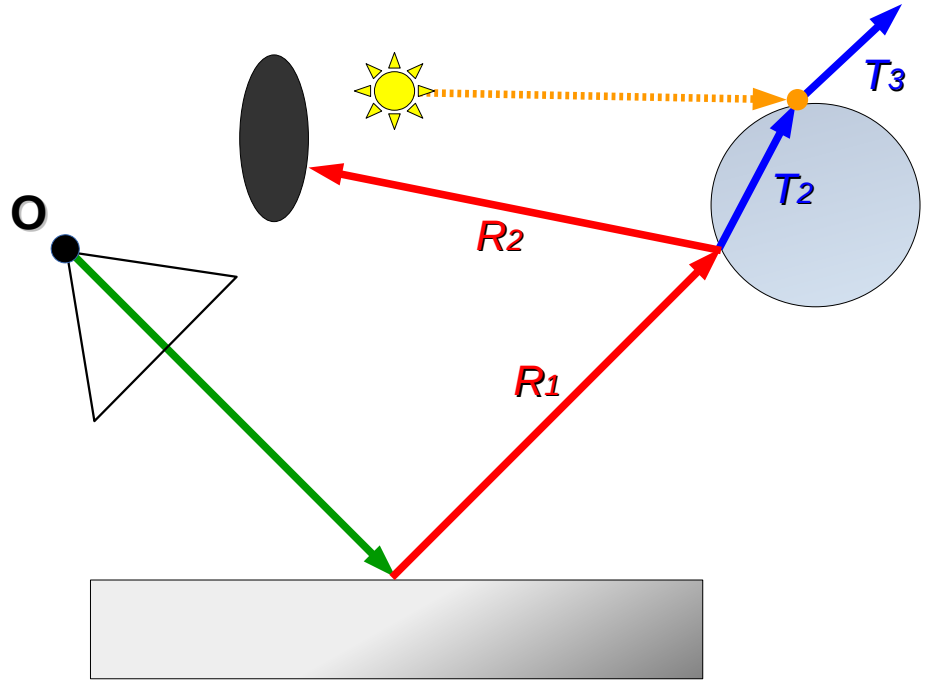


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

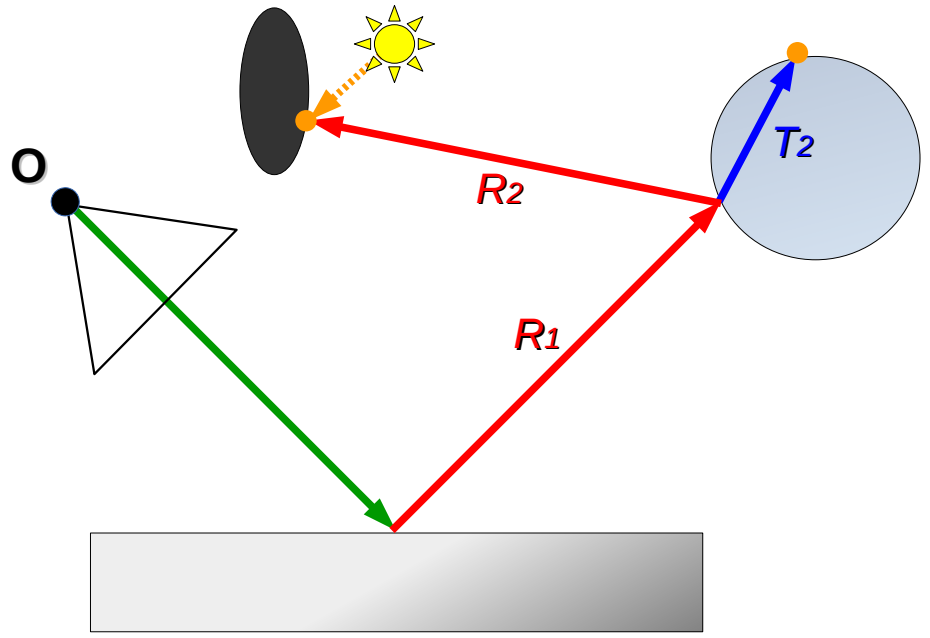


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

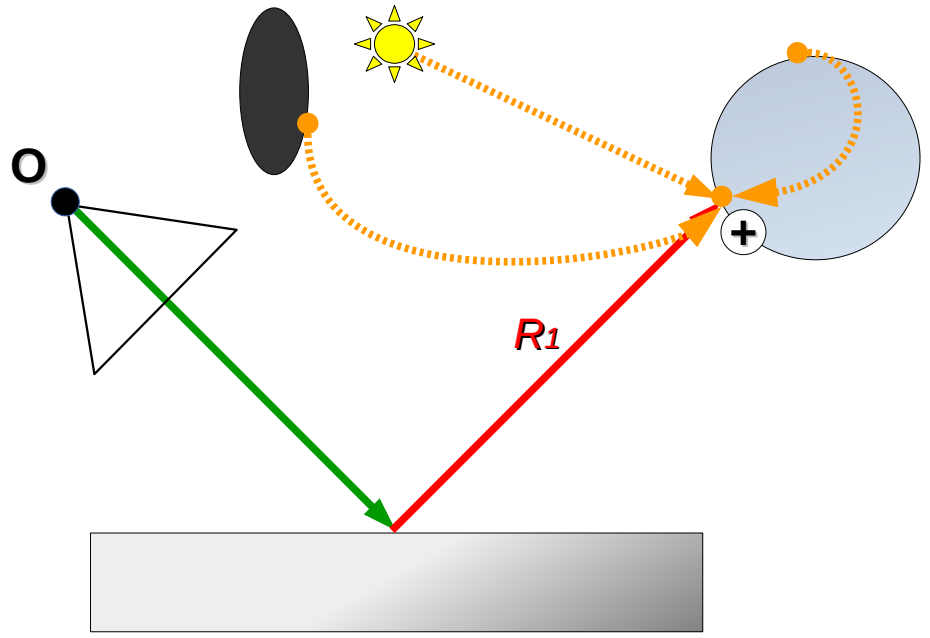


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.

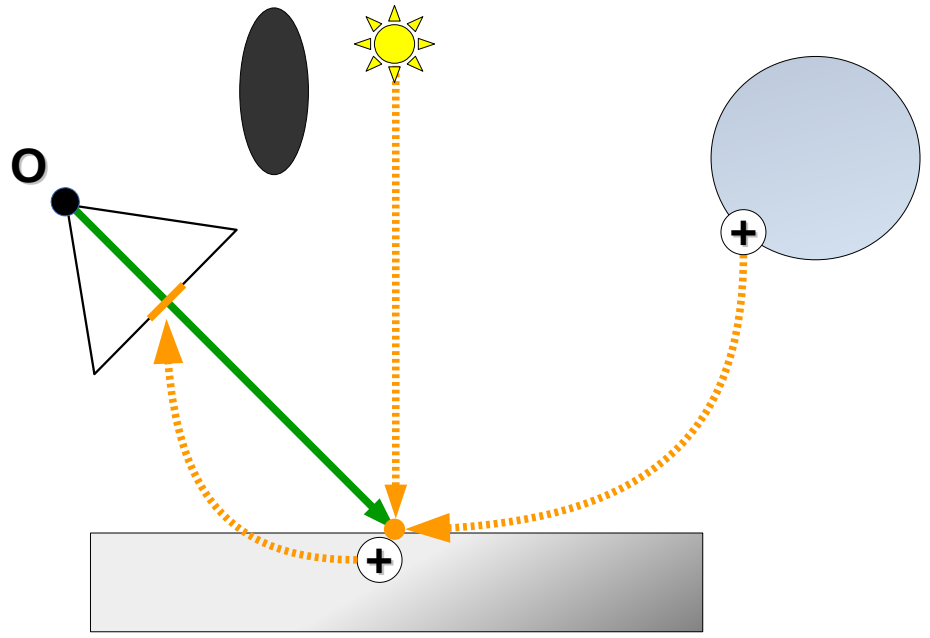


Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Jeśli promień pierwotny napotka obiekt matowy, to w punkcie przecięcia obliczana jest wartość modelu oświetlenia (tak samo, jak w podstawowym algorytmie). Jeżeli jednak obiekt jest przezroczysty lub odbijający, to generowane są promienie wtórne.

W punkcie przecięcia promienia wtórnego z obiektem sceny wyznaczana jest wartość modelu oświetlenia, wpływająca na kolor punktu, z którego ten promień wyszedł.

Przecięcie przez promień wtórny kolejnego obiektu odbijającego lub przezroczystego spowoduje emisję nowych promieni wtórnych.



Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Do działania algorytmu wymagane jest wprowadzenie do modelu oświetlenia członów rekurencyjnych związanych z promieniami odbitymi i ulegającymi załamaniu:

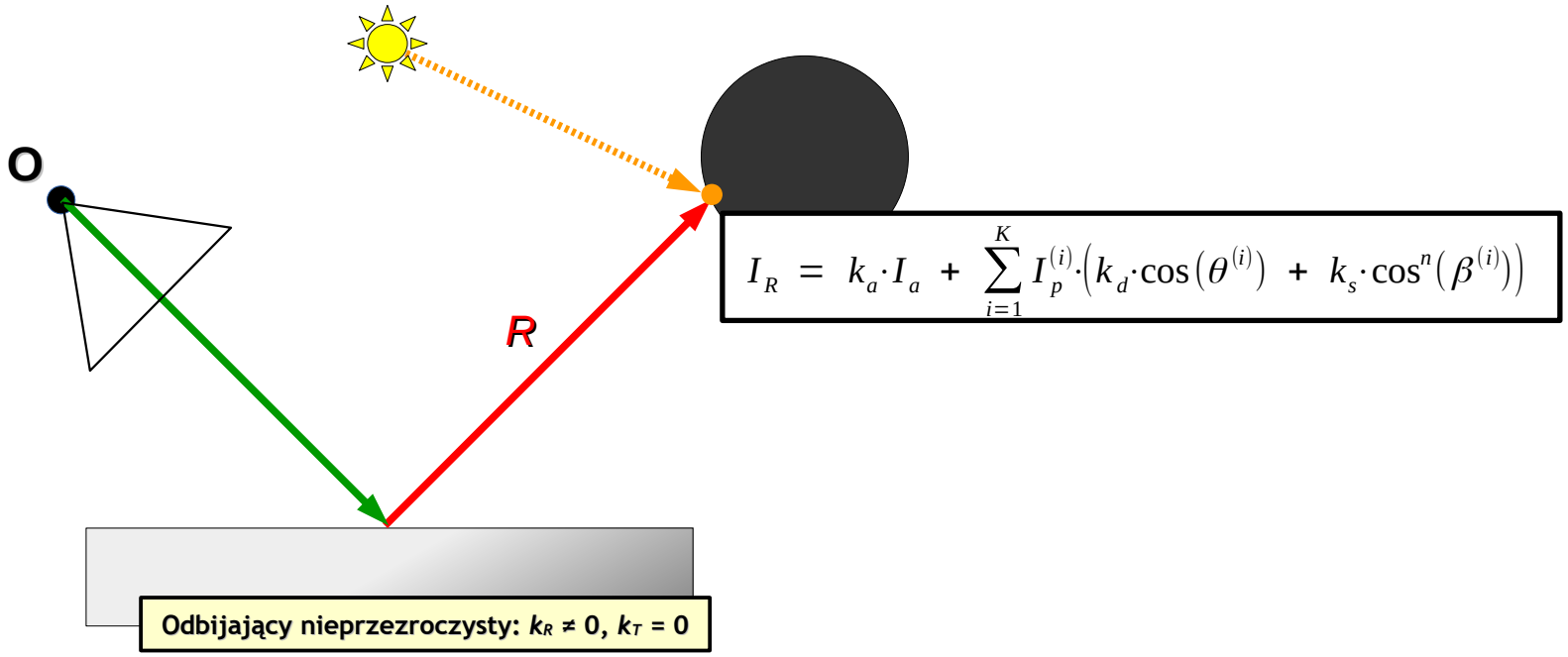
$$I = \underbrace{k_a \cdot I_a + \sum_{i=1}^{K} I_p^{(i)} \cdot \left(k_d \cdot \cos\left(\theta^{(i)}\right) + k_s \cdot \cos^n\left(\beta^{(i)}\right) \right)}_{\text{„Zwykły” model Phonga}} + \underbrace{k_R \cdot I_R}_{\text{Wpływ odbicia}} + \underbrace{k_T \cdot I_T}_{\text{Wpływ załamania}}$$

Współczynniki k_R i k_T opisują względny udział odbicia i refrakcji dla danego materiału. W najprostszym podejściu można im przypisać stałe wartości, ale w rzeczywistości zależą od kąta padania oraz polaryzacji światła (wykład 5, opis modelu refrakcji).

Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Do działania algorytmu wymagane jest wprowadzenie do modelu oświetlenia członów rekurencyjnych związanych z promieniami odbitymi i ulegającymi załamaniu:

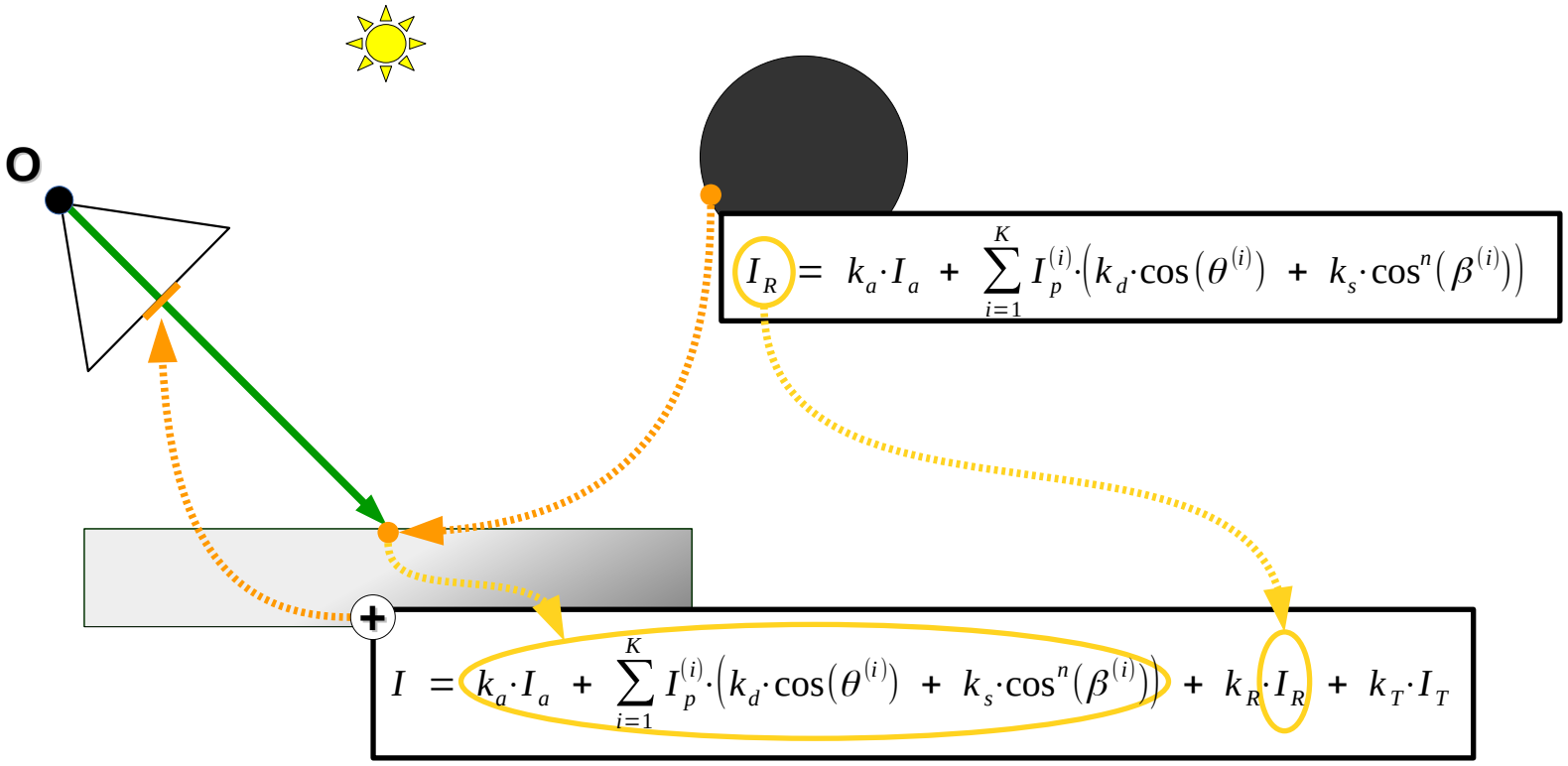
$$I = \underbrace{k_a \cdot I_a + \sum_{i=1}^{K} I_p^{(i)} \cdot \left(k_d \cdot \cos\left(\theta^{(i)}\right) + k_s \cdot \cos^n\left(\beta^{(i)}\right) \right)}_{\text{„Zwykły” model Phonga}} + \underbrace{k_R \cdot I_R}_{\text{Wpływ odbicia}} + \underbrace{k_T \cdot I_T}_{\text{Wpływ załamania}}$$



Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

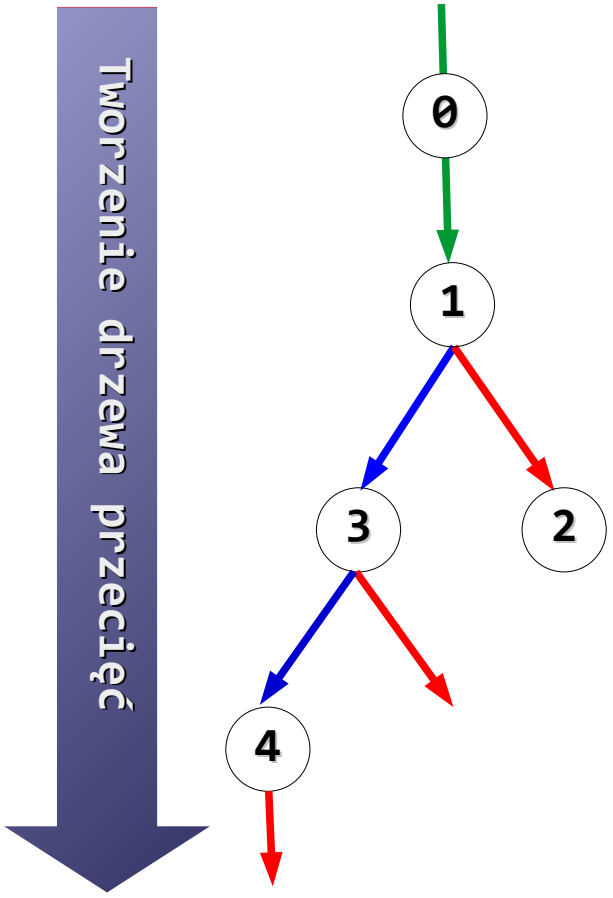
Do działania algorytmu wymagane jest wprowadzenie do modelu oświetlenia członów rekurencyjnych związanych z promieniami odbitymi i ulegającymi załamaniu:

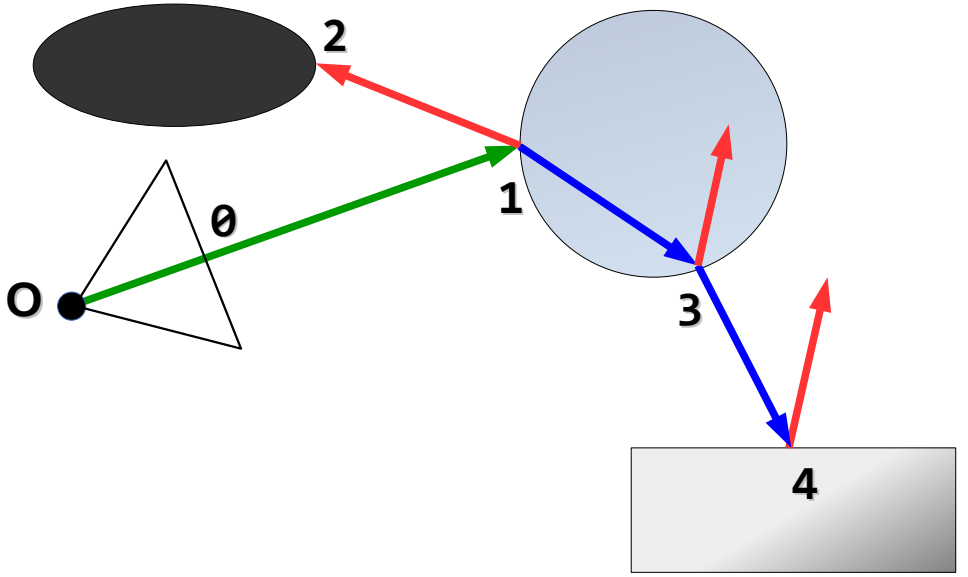
$$I = \underbrace{k_a \cdot I_a + \sum_{i=1}^{K} I_p^{(i)} \cdot \left(k_d \cdot \cos\left(\theta^{(i)}\right) + k_s \cdot \cos^n\left(\beta^{(i)}\right) \right)}_{\text{„Zwykły” model Phonga}} + \underbrace{k_R \cdot I_R}_{\text{Wpływ odbicia}} + \underbrace{k_T \cdot I_T}_{\text{Wpływ załamania}}$$



Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

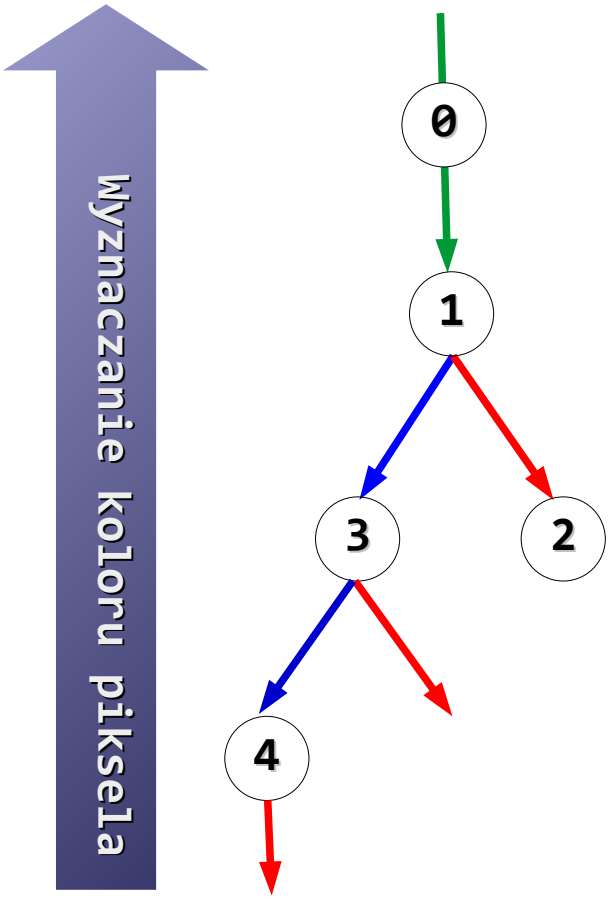
Bieg promieni można opisać w postaci drzewa przecięć, o węzłach reprezentujących zjawiska zachodzące na powierzchni obiektów. Jego budowanie zaczyna się od promienia pierwotnego, a kończy gdy promienie wtórne nie przecinają się już z kolejnymi obiektami lub po osiągnięciu zadanej z góry maksymalnej liczby przecięć.

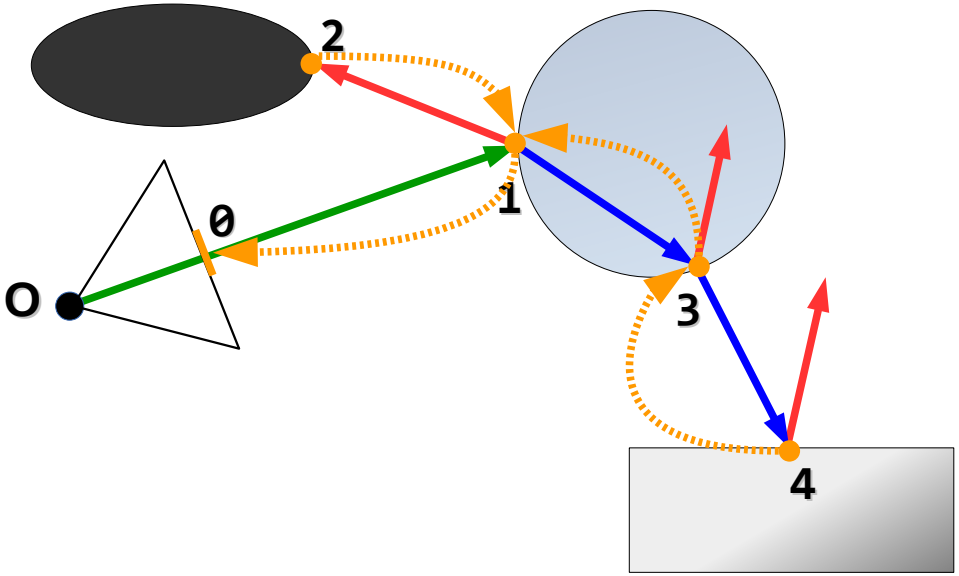




Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

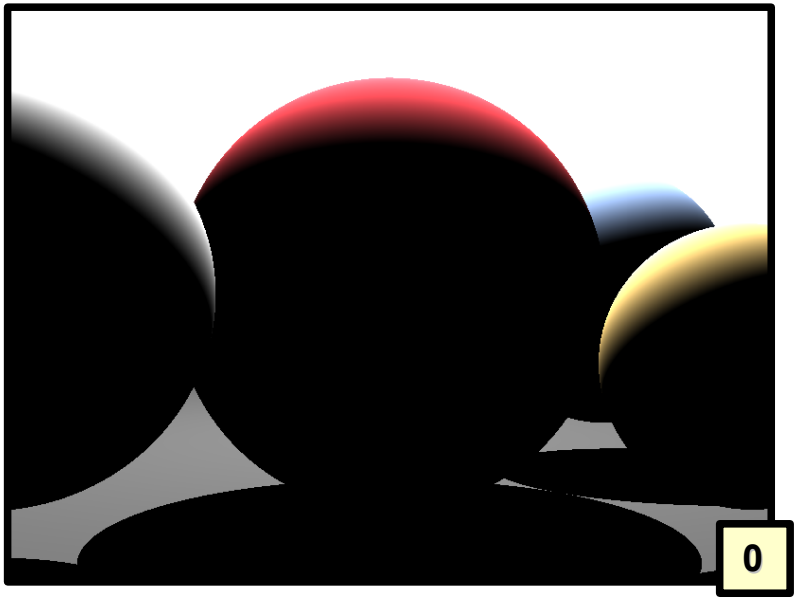
Kolor rozpatrywanego piksela jest wyznaczany przez analizę drzewa, poczynając od liści, a kończąc na korzeniu, przy czym na wartość wyznaczaną z modelu oświetlania w danym węźle wpływają wyniki uzyskane w węzłach potomnych (tzn. uwzględniane są człony rekurencyjne związane z odbiciem i refrakcją).

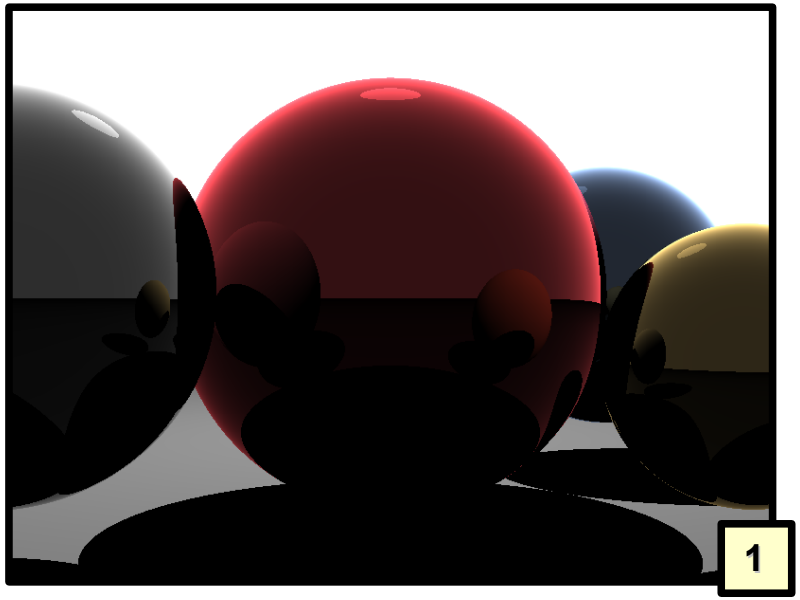


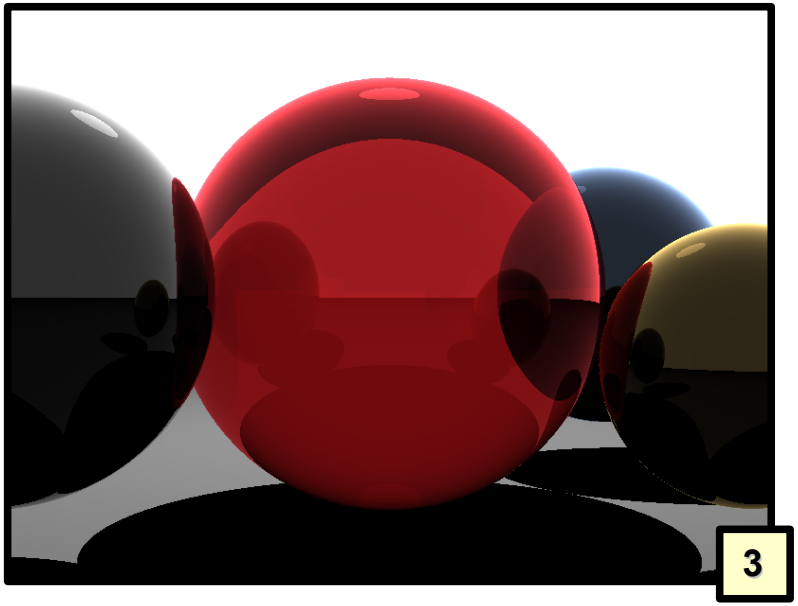


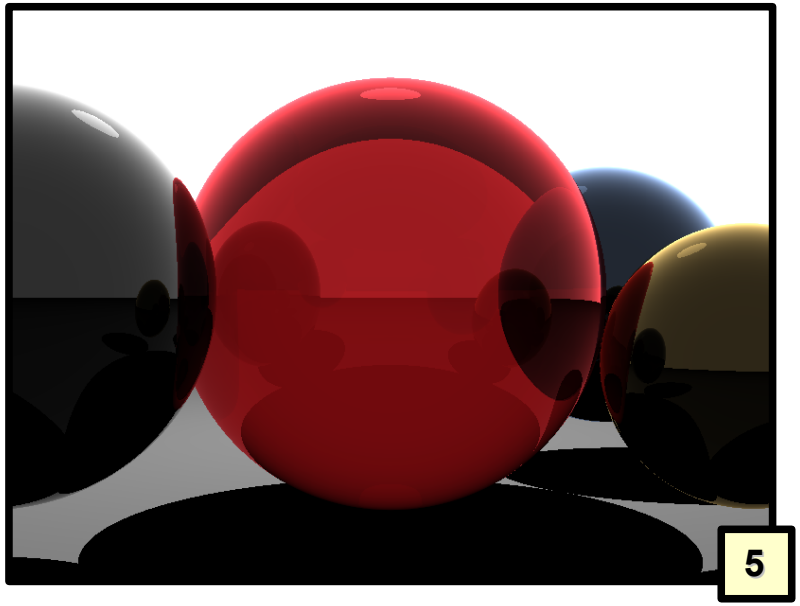
Rendering 3D: metoda śledzenia promieni (wersja rekurencyjna)

Maksymalna liczba możliwych obić lub refrakcji liczonych dla danego promienia pierwotnego (głębokość rekursji) jest parametrem algorytmu, którego wartość musi być kompromisem między stopniem realizmu, a czasem obliczeń.







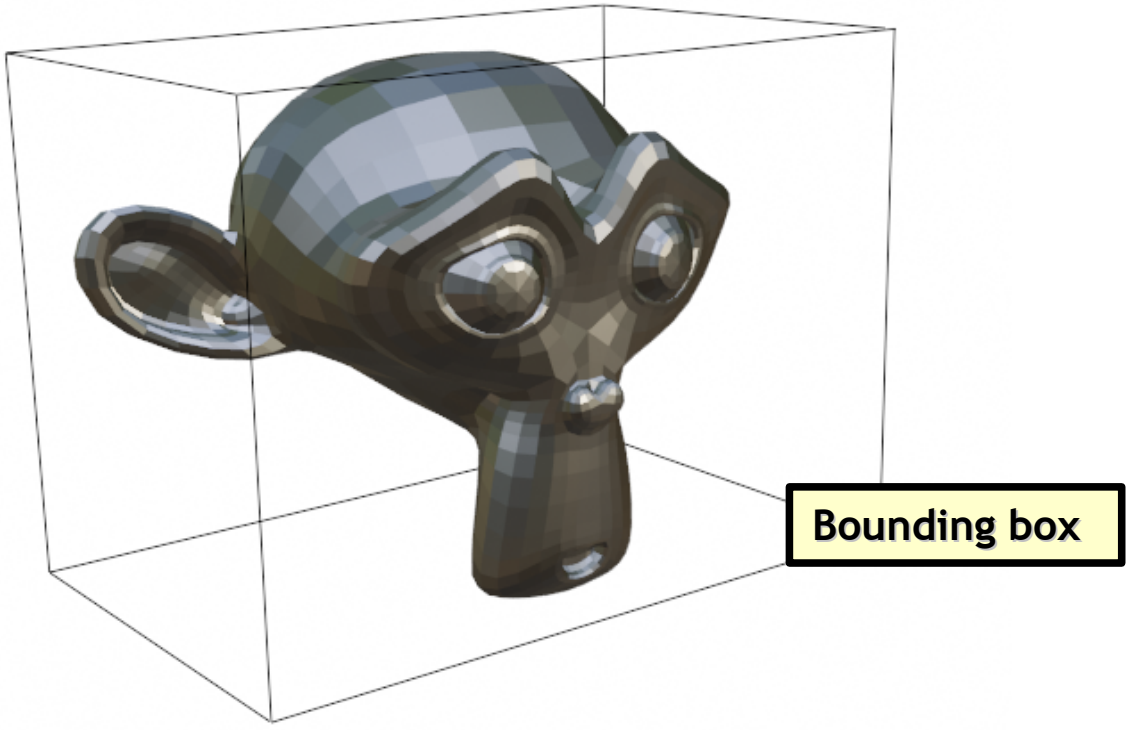


Rendering 3D: metoda śledzenia promieni (modyfikacje)

Metoda śledzenia promieni od chwili powstania podlega nieustannemu rozwojowi i doczekała się wielu modyfikacji wpływających na wydajność i jakość renderingu.

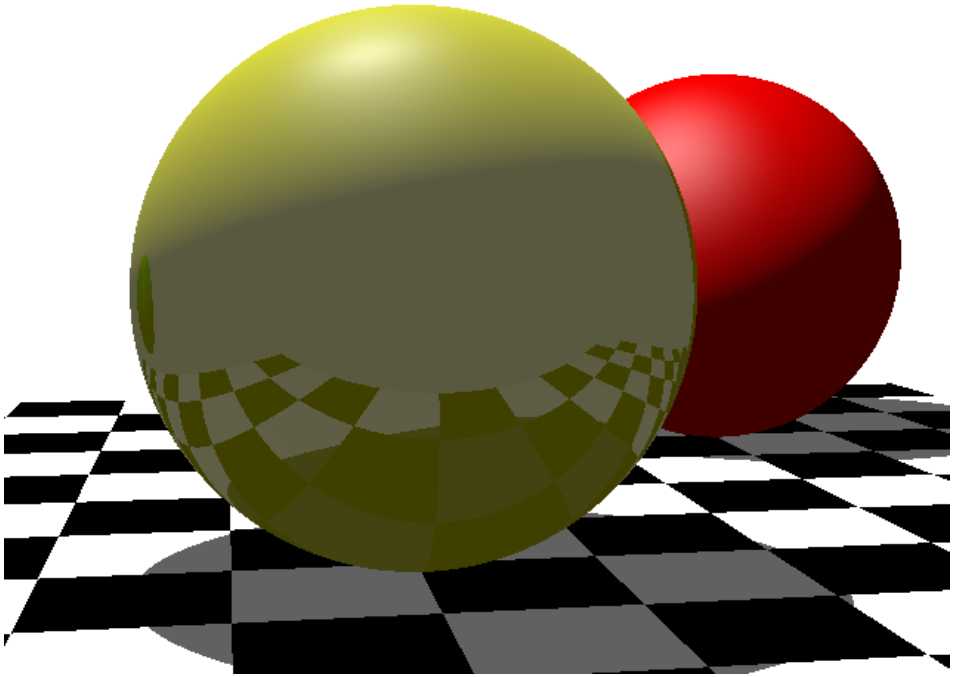
Przyspieszenie renderingu możliwe m. in. dzięki:

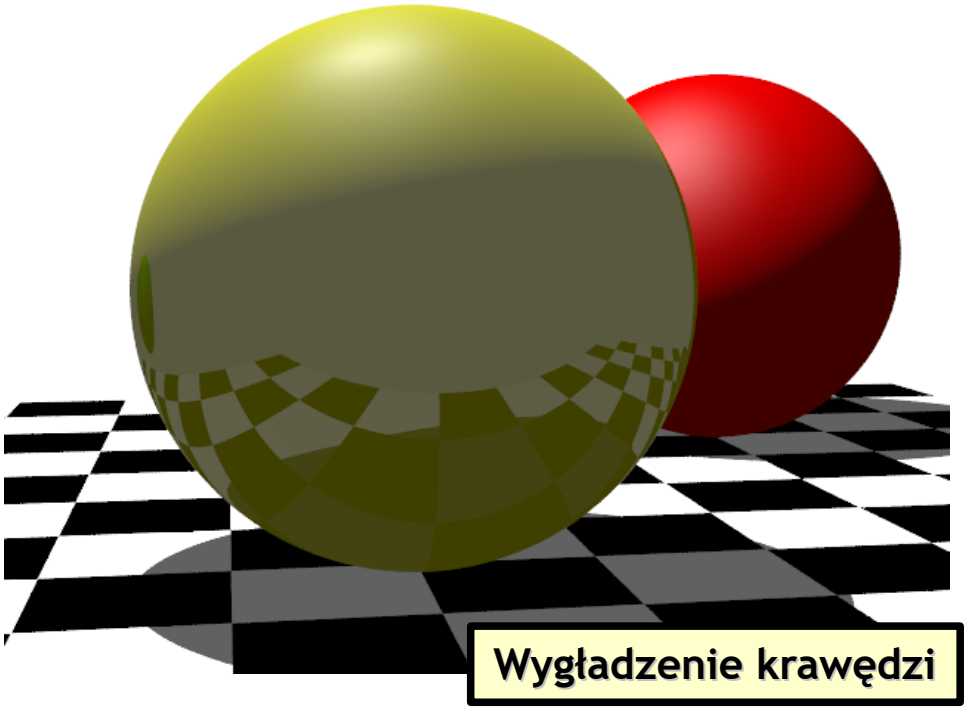
- obliczeniom równoległym – każdy piksel może być przetwarzany niezależnie;
- poprawieniu efektywności wyznaczania przecięć promieni z obiektami dzięki użyciu struktur danych pozwalających na wydajne przeszukiwanie przestrzeni (np. drzew KD, ang. KD-trees) oraz brył brzegowych (ang. *bounding volumes*).



Rendering 3D: metoda śledzenia promieni (modyfikacje)

Poprawę jakości i stopnia realizmu obrazów można uzyskać poprzez zastosowanie **śledzenia rozproszonego (*distribured ray tracing*)** całych grup promieni pierwotnych, wtórnych i/lub cienia, zwykle generowanych w sposób losowy. Podejście to pozwala wygładzić krawędzie obiektów oraz uwzględnić dodatkowe efekty wizualne, w tym: **miękkie cienie**, **głębię ostrości** i **rozmycie wynikające z ruchu**.

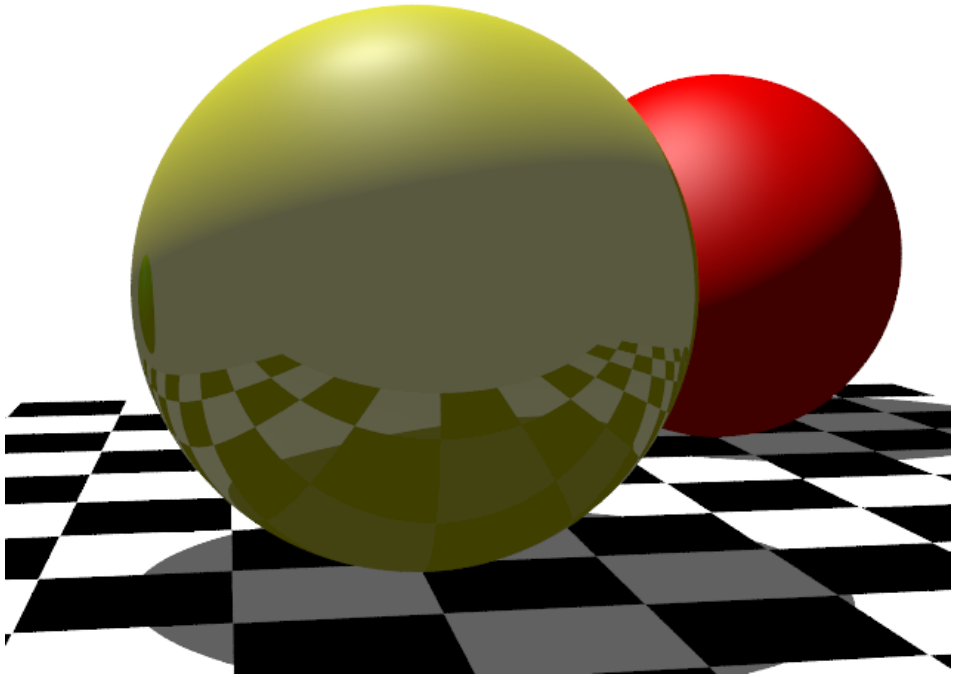


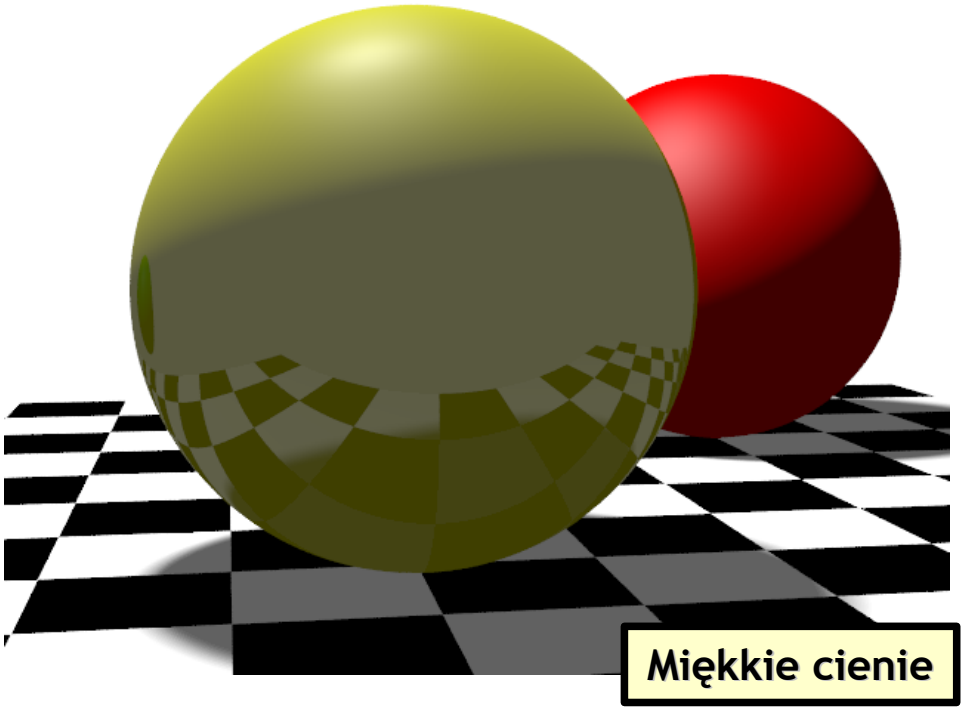


Wygładzenie krawędzi

Rendering 3D: metoda śledzenia promieni (modyfikacje)

Poprawę jakości i stopnia realizmu obrazów można uzyskać poprzez zastosowanie **śledzenia rozproszonego (*distribured ray tracing*)** całych grup promieni pierwotnych, wtórnych i/lub cienia, zwykle generowanych w sposób losowy. Podejście to pozwala wygładzić krawędzie obiektów oraz uwzględnić dodatkowe efekty wizualne, w tym: **miękkie cienie**, **głębię ostrości** i **rozmycie wynikające z ruchu**.

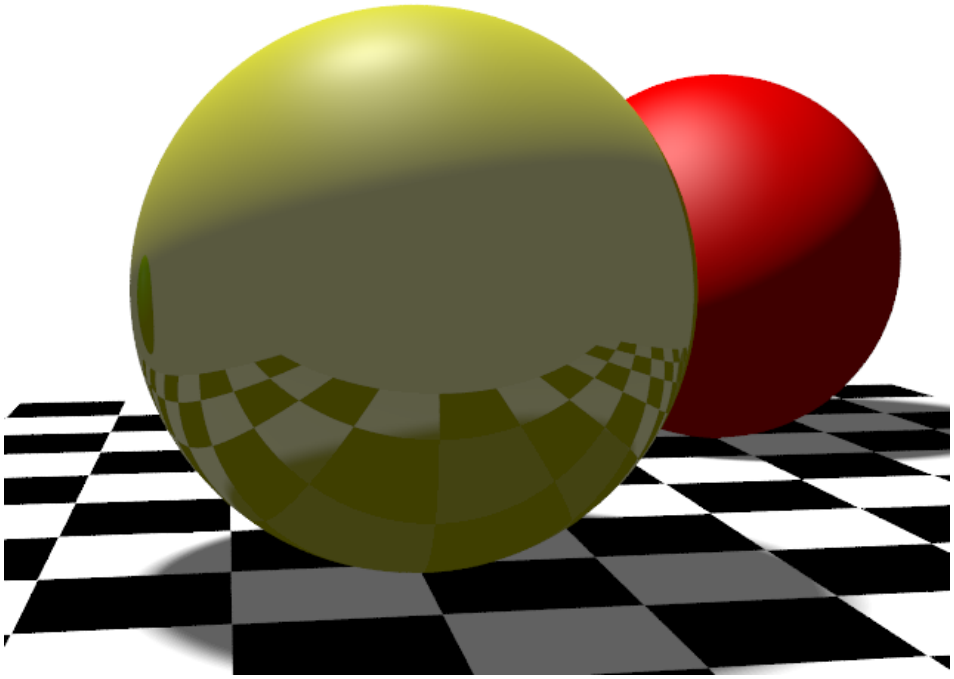


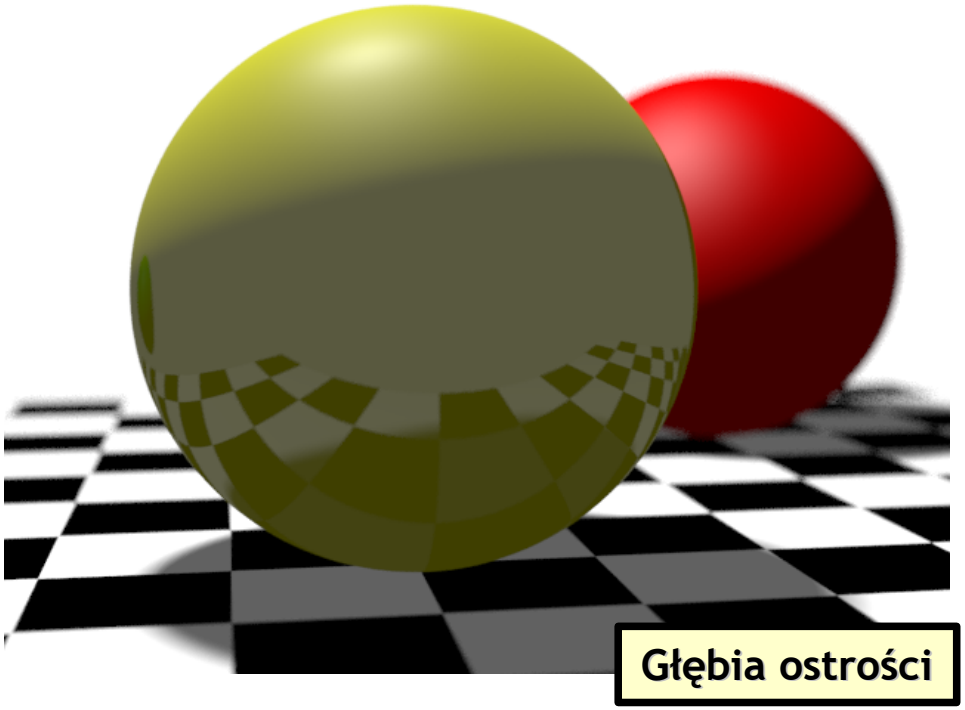


Miękkie cienie

Rendering 3D: metoda śledzenia promieni (modyfikacje)

Poprawę jakości i stopnia realizmu obrazów można uzyskać poprzez zastosowanie **śledzenia rozproszonego (*distribured ray tracing*)** całych grup promieni pierwotnych, wtórnych i/lub cienia, zwykle generowanych w sposób losowy. Podejście to pozwala wygładzić krawędzie obiektów oraz uwzględnić dodatkowe efekty wizualne, w tym: **miękkie cienie**, **głębię ostrości** i **rozmycie wynikające z ruchu**.

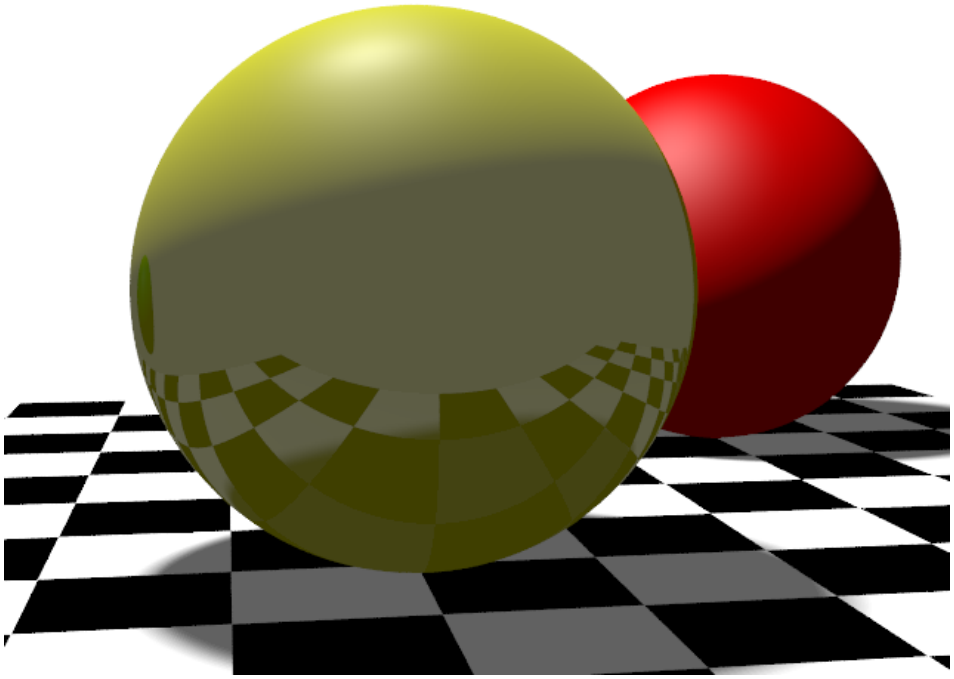


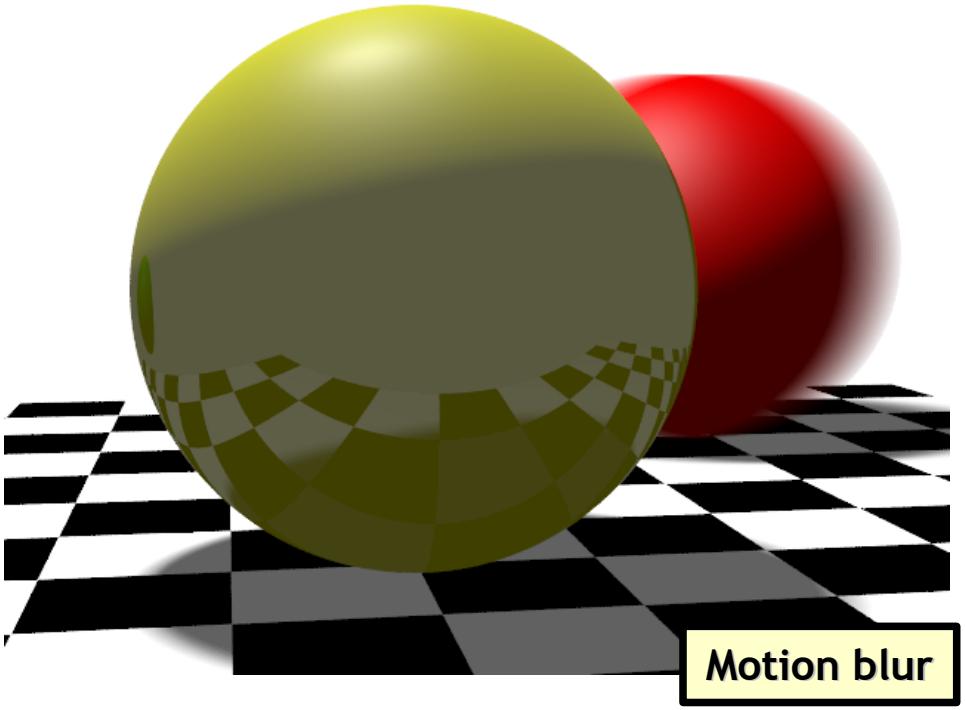


Głębia ostrości

Rendering 3D: metoda śledzenia promieni (modyfikacje)

Poprawę jakości i stopnia realizmu obrazów można uzyskać poprzez zastosowanie **śledzenia rozproszonego (*distribured ray tracing*)** całych grup promieni pierwotnych, wtórnych i/lub cienia, zwykle generowanych w sposób losowy. Podejście to pozwala wygładzić krawędzie obiektów oraz uwzględnić dodatkowe efekty wizualne, w tym: **miękkie cienie**, **głębię ostrości** i **rozmycie wynikające z ruchu**.





Rendering 3D: metoda śledzenia promieni (podsumowanie)

Zalety metody śledzenia promieni:

- **jest prosta koncepcyjnie**: jeden nieskomplikowany algorytm obejmuje wszystkie zagadnienia związane z renderingiem: wyznaczanie kolorów obiektów, rzutowanie, rozstrzyganie widoczności, zmianę układu współrzędnych, rasteryzację;
- **uwzględnia większość zjawisk optycznych** wpływających na realizm obrazów;
- **jest łatwa w implementacji**, również w architekturze równoległej.

Wady metody śledzenia promieni:

- **nie rozwiązuje w pełni problemu oświetlania globalnego**, bo nie uwzględnia różnic w natężeniu światła otoczenia w obrębie sceny, które wynikają z rozproszonych odbić od matowych powierzchni;
- **nie pozwala prawidłowo wizualizować części efektów**, np: rozszczepienia, dyfrakcji i interferencji fal świetlnych;
- **jest bardzo kosztowana obliczeniowo**.

GRAFIKA KOMPUTEROWA

Wykład 6: grafika trójwymiarowa (rendering w „czasie rzeczywistym”)

Tymon Rubel

Zakład Elektroniki Jądrowej i Medycznej
Instytut Radioelektroniki i Technik Multimedialnych PW

Rendering 3D: rendering w czasie rzeczywistym

W przypadku gier oraz innych zastosowań wymagających tworzenia grafiki „w czasie rzeczywistym” (*real-time*) ważnym kryterium doboru technik renderingu okazuje się ograniczenie czasu obliczeń, nawet kosztem realizmu.

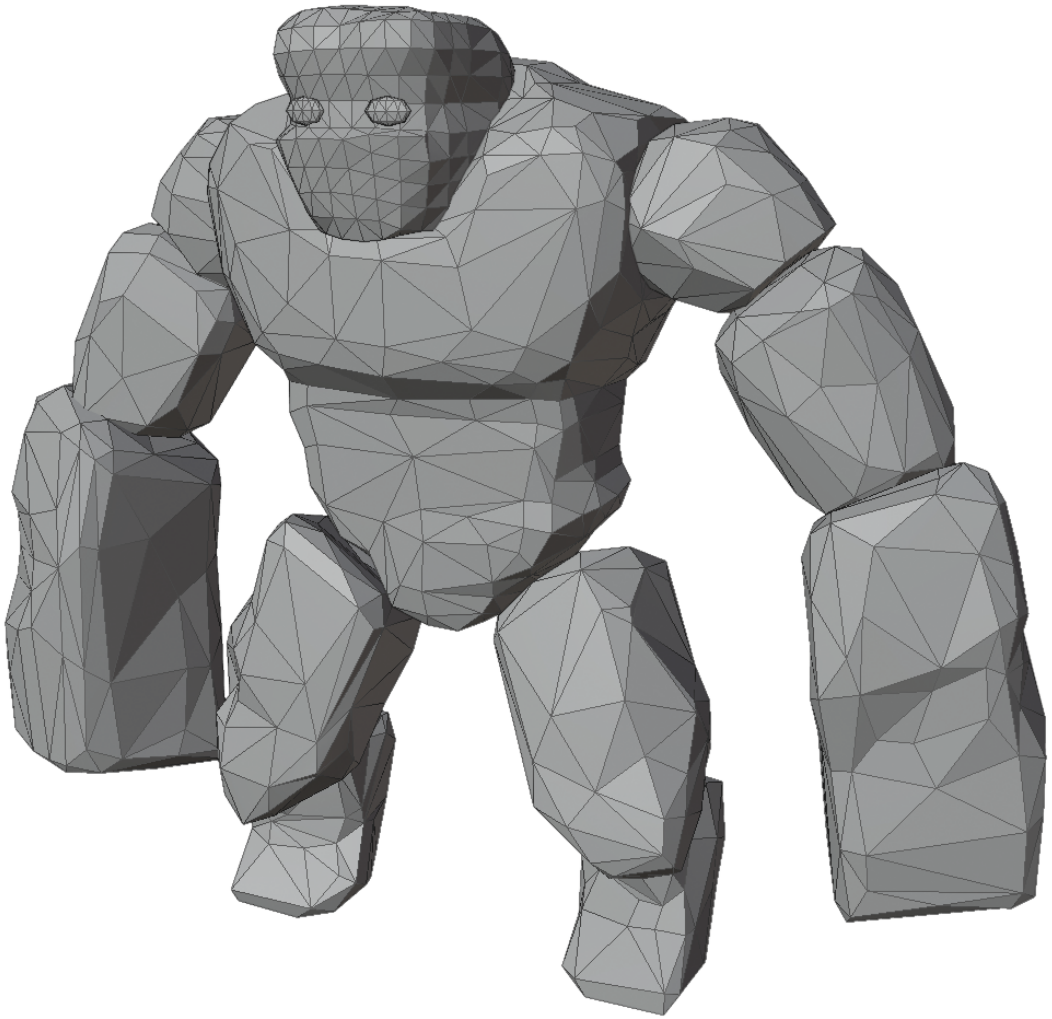
Opisany wcześniej algorytm śledzenia promieni jest zbyt mało efektywny czasowo, aby mógł być powszechnie wykorzystywany do renderingu animacji, których klatki są wyznaczane na bieżąco w trakcie wyświetlania.

Film: jedna klatka generowana dwie godziny

Gra: co najmniej 25 klatek na sekundę

Rendering 3D: rendering w czasie rzeczywistym (modele)

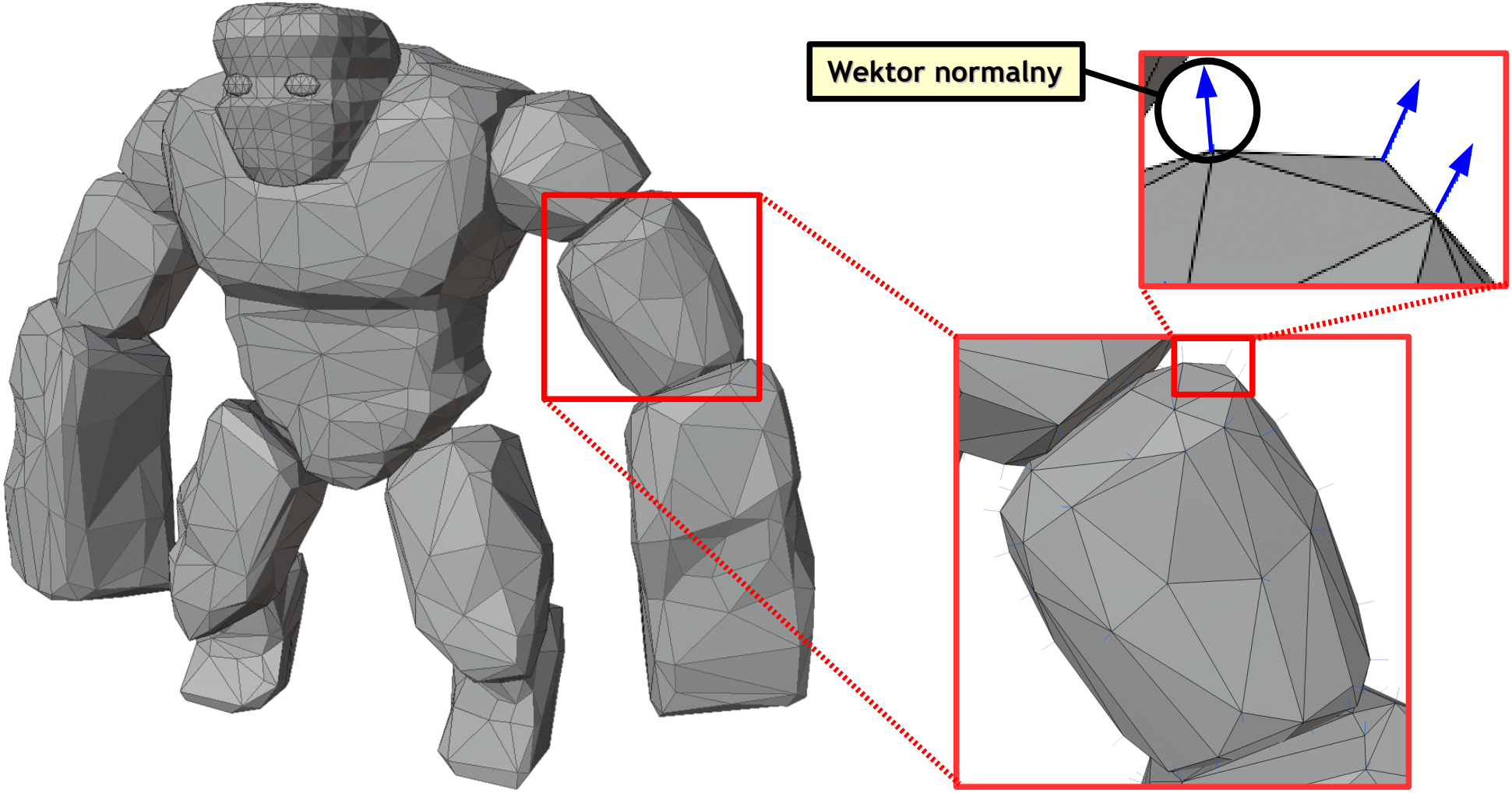
Uproszczenia stosowane podczas generacji grafiki w czasie rzeczywistym dotyczą już etapu modelowania. Do opisu obiektów używa się relatywnie prostych modeli, zwykle w postaci siatki trójkątów (*triangle mesh*).



[free3d.com/3d-model/giant-stone-765391.html]

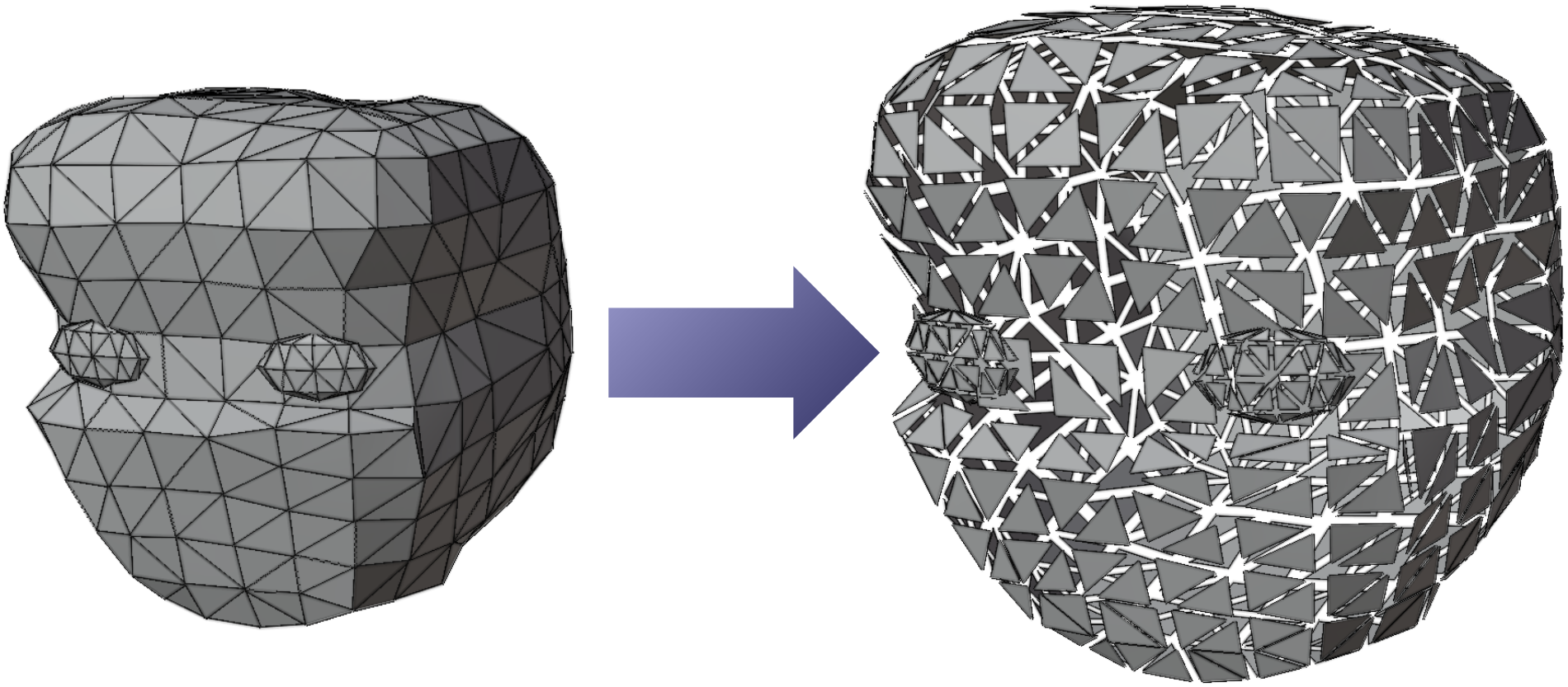
Rendering 3D: rendering w czasie rzeczywistym (modele)

Każdy wierzchołek w siatce opisywany jest przez współrzędne i wektor normalny do powierzchni w danym miejscu. Dodatkowo przechowywane są informacje o kolorze i parametrach optycznych materiału obiektu.



Rendering 3D: rendering w czasie rzeczywistym (potok renderujący)

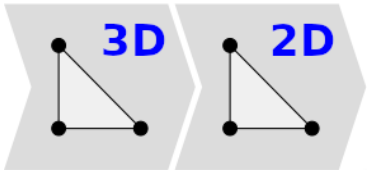
Obraz generowany jest w procedurze rasteryzacji 3D, wykonywanej niezależnie dla wszystkich wielokątów wchodzących w skład modelu.



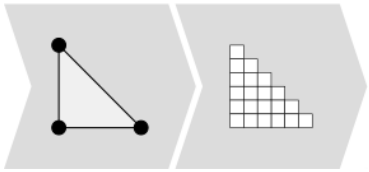
Rendering 3D: rendering w czasie rzeczywistym (potok renderujący)

Procedura rasteryzacji 3D implementowana jest w postaci oddzielnych kroków, które realizują kolejne etapy przetwarzania wielokątów, w tym:

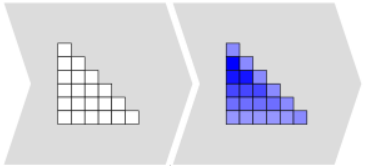
- **transformacje geometryczne i rzutowanie na płaszczyznę;**



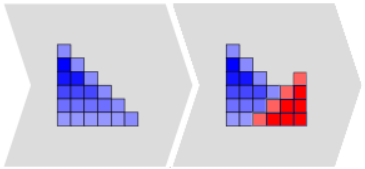
- **rasteryzację 2D i przycinanie** (zamiana wielokątów na zbiory pikseli);



- **cieniowanie** (określanie wpływu źródeł światła na kolory pikseli);

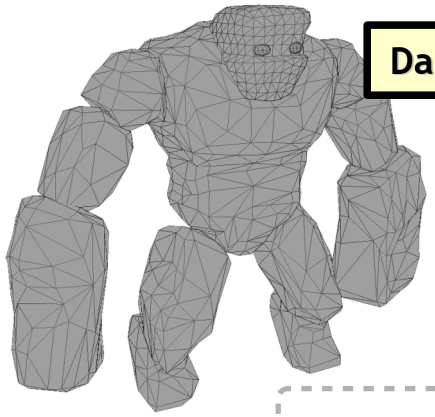


- **testowanie widoczności i przysłaniania obszarów.**

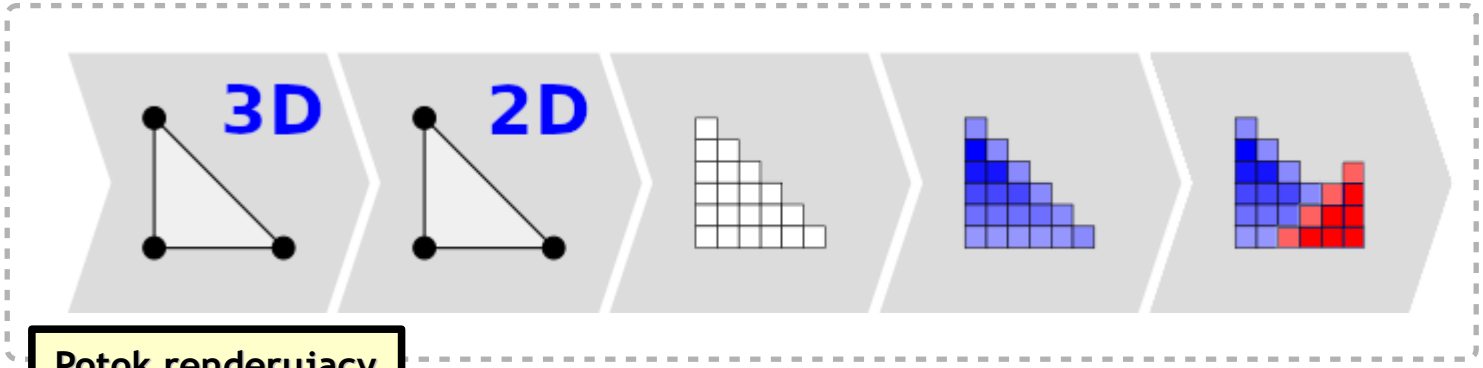


Rendering 3D: rendering w czasie rzeczywistym (potok renderujący)

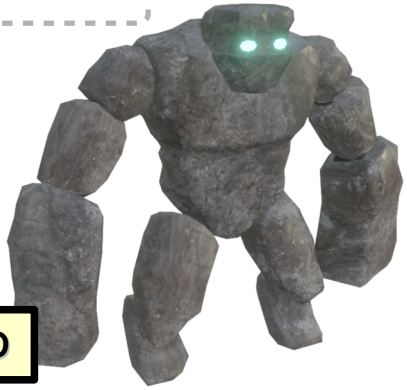
Kolejne etapy przetwarzania są wykonywane w taki sposób, aby wyniki poprzedniego kroku były jednocześnie danymi wejściowymi dla następnego. Cały ten ciąg operacji określany jest mianem **potoku renderującego (*rendering pipeline*)**.



Dane o wierzchołkach siatki



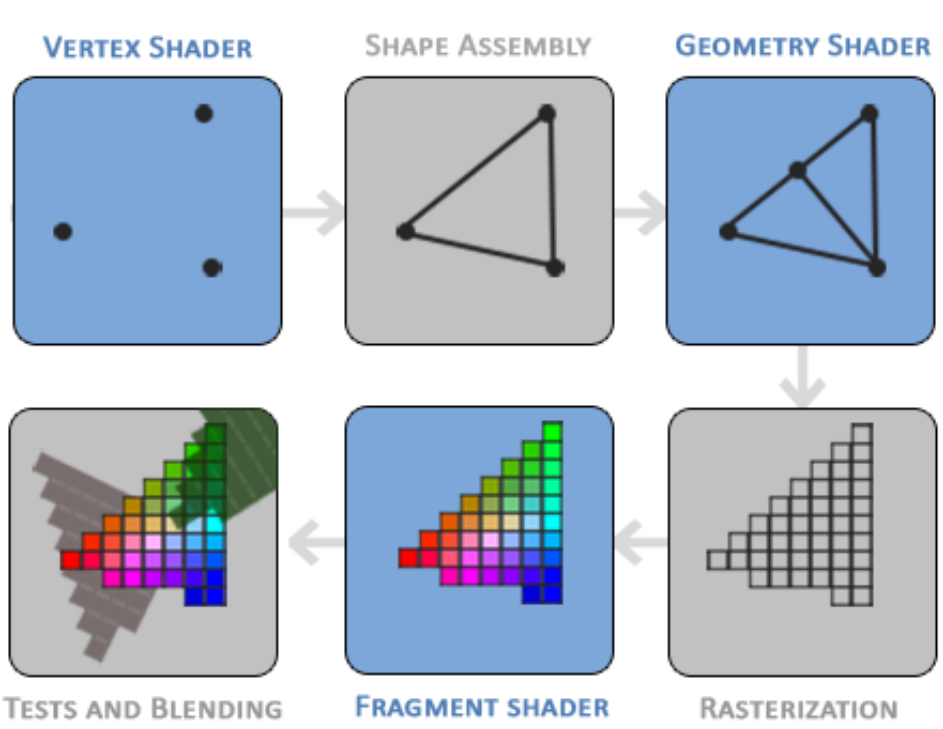
Potok renderujący



Wynik rasteryzacji 3D

Rendering 3D: rendering w czasie rzeczywistym (potok renderujący)

Współczesne procesory graficzne (*GPU*, *Graphics Processing Unit*) wspomagają potok renderujący na poziomie sprzętu, a dostęp do niego odbywa się poprzez odpowiednie interfejsy programistyczne (np. DirectX, OpenGL, Vulkan) i specjalizowane procedury nazywane shaderami.

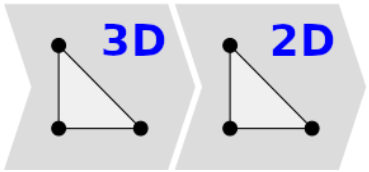


[learnopengl.com]

Rendering 3D: rendering w czasie rzeczywistym (potok renderujący)

Procedura rasteryzacji 3D implementowana jest w postaci oddzielnych kroków, które realizują kolejne etapy przetwarzania wielokątów, w tym:

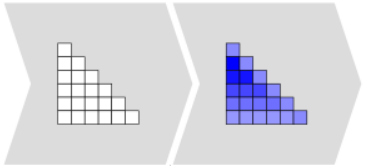
- transformacje geometryczne i rzutowanie na płaszczyznę;



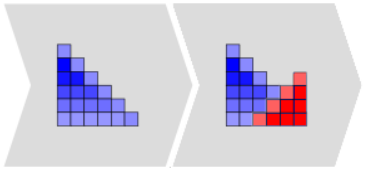
- rasteryzację 2D i

UWAGA: transformacje geometryczne, rzutowanie i rasteryzacja realizowane są w sposób już wcześniej omówiony (wykłady 3 i 5), dlatego dalej opisano jedynie nowe zagadnienia: cieniowanie oraz eliminację powierzchni niewidocznych.

- cieniowanie (określanie wpływu źródeł światła na kolor pikseli);

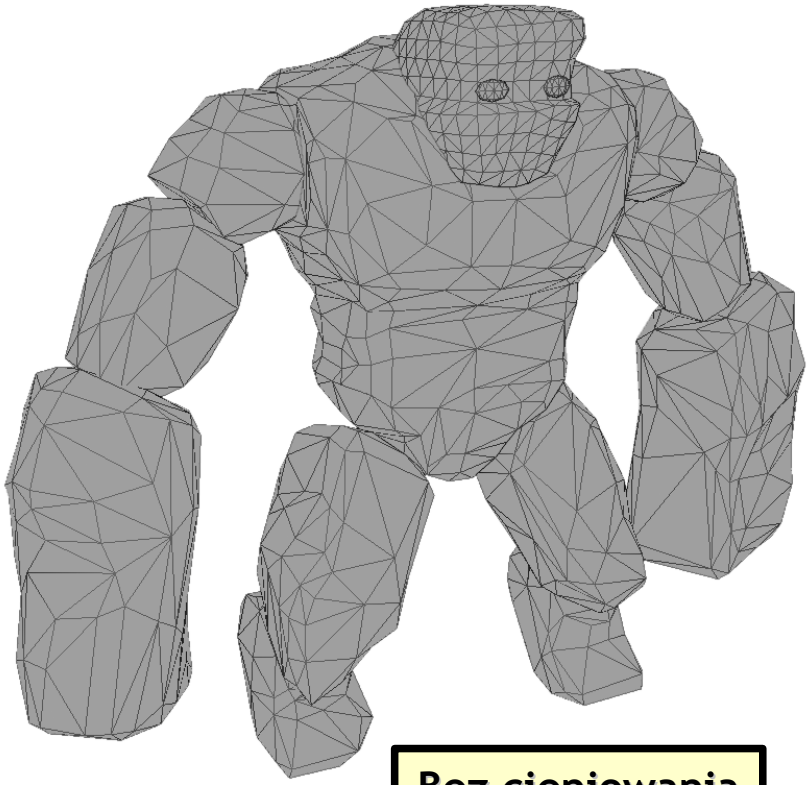


- testowanie widoczności i przysłaniania obszarów.

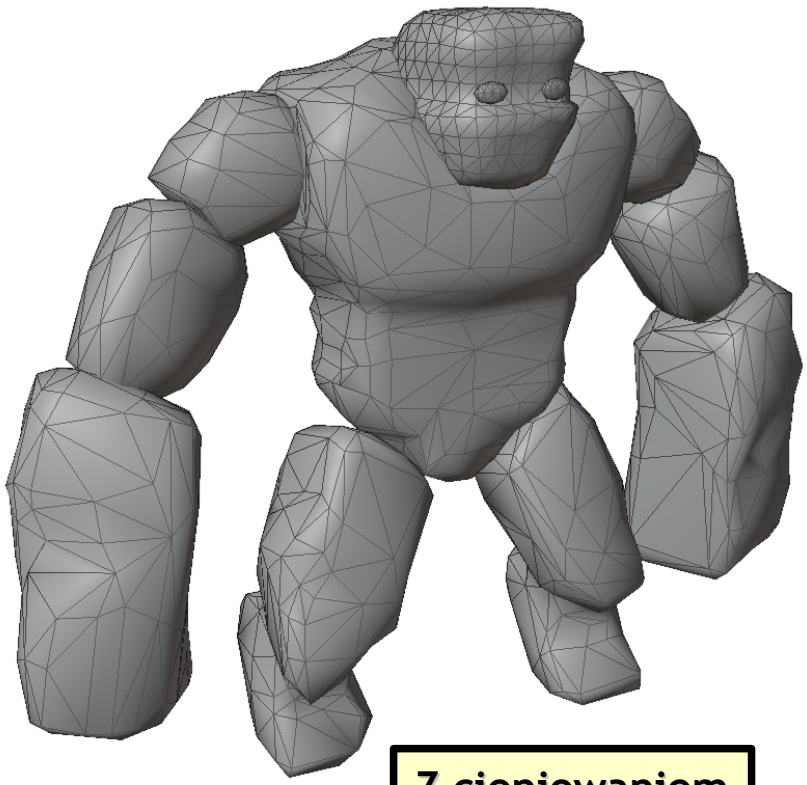


Rendering 3D: rendering w czasie rzeczywistym (cieniowanie)

Cieniowanie (*shading*) polega na wyznaczaniu kolorów pikseli wchodzących w skład zrzutowanych na płaszczyznę wielokątów z uwzględnieniem zdefiniowanych w scenie źródeł światła i zadanego modelu oświetlenia (np. modelu Phonga).



Bez cieniowania

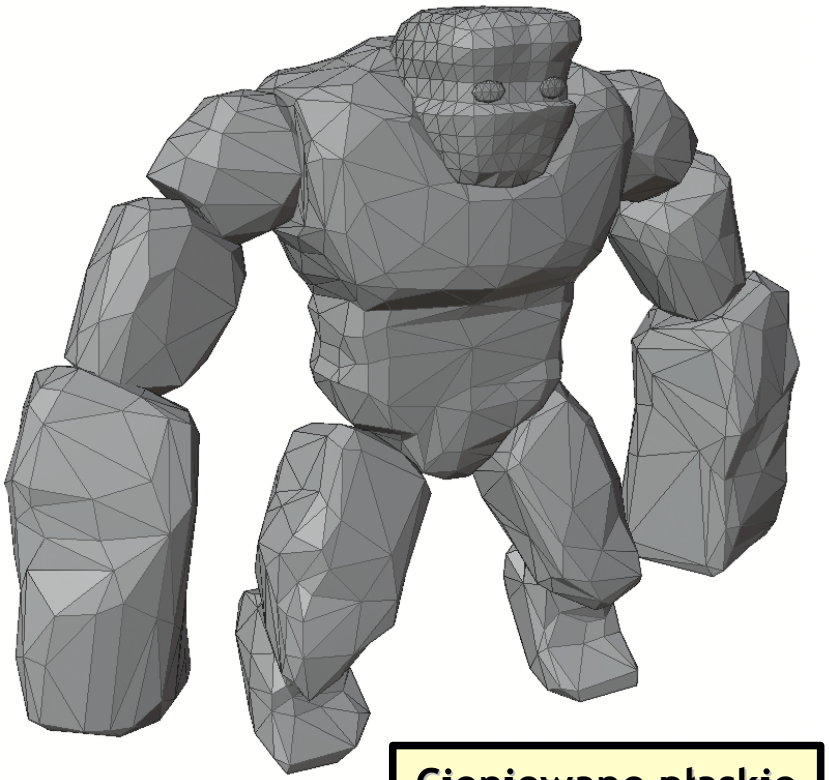


Z cieniowaniem

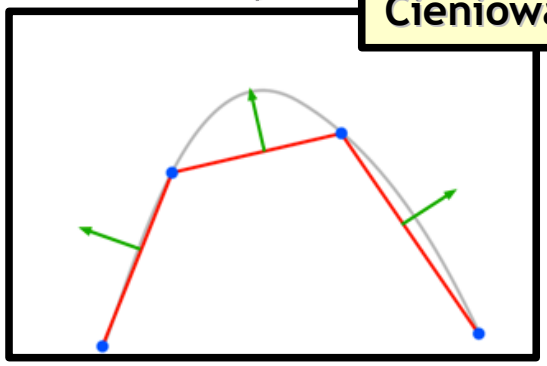
Cieniowanie w rasteryzacji 3D korzysta jedynie z lokalnych właściwości wierzchołków wielokąta (kolory materiały, współrzędne, wektory normalne) i informacji o źródłach światła. Pomimo tego pozwala znacząco poprawić fotorealizm, przy jednoczesnym zachowaniu możliwości użycia modeli złożonych z niewielkiej liczby wielokątów.

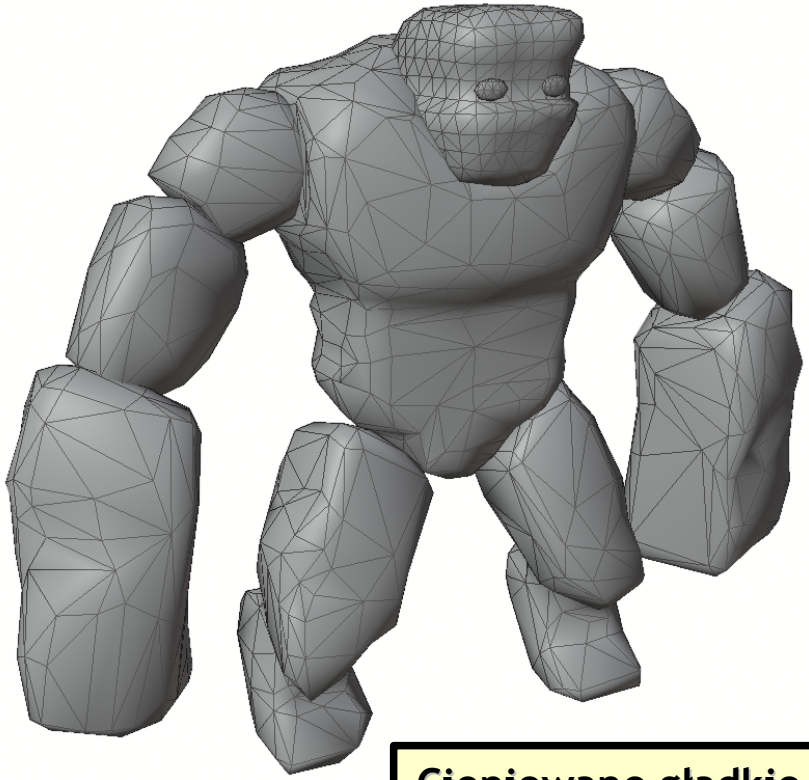
Rendering 3D: rendering w czasie rzeczywistym (cieniowanie)

W zależności algorytmu kolor pikseli może być jednolity w całym obszarze wielokąta (**cieniowanie płaskie**) lub zmieniać się, symulując interakcję światła z powierzchnią, której elementem jest dany wielokąt (**cieniowanie gładkie**, w tym: **Gourauda** i **Phoga**).

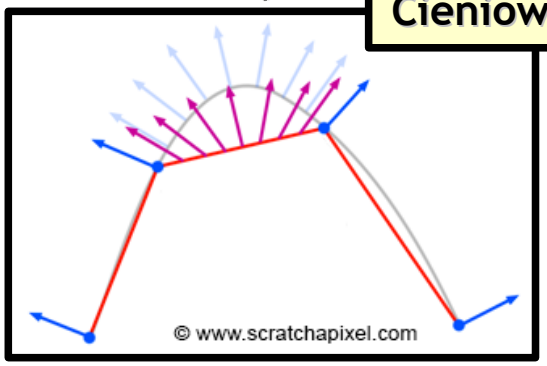


Cieniowane płaskie



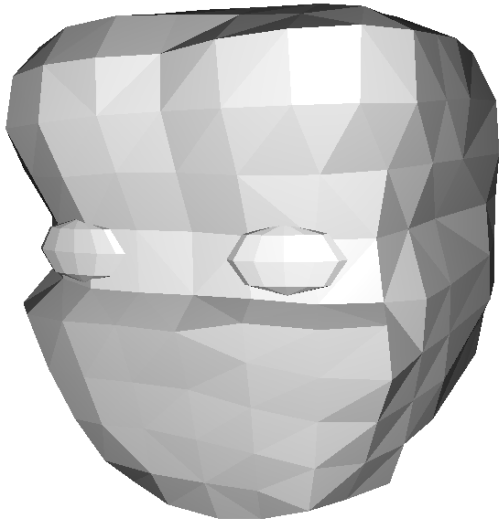


Cieniowane gładkie



Rendering 3D: rendering w czasie rzeczywistym (cieniowanie płaskie)

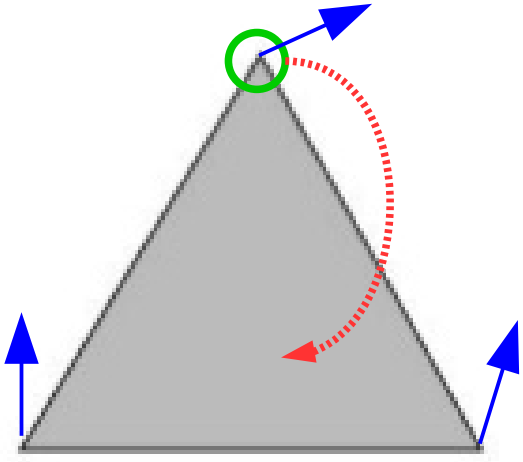
W **cieniowaniu płaskim** każdy wielokąt wypełniany jest jednym kolorem. Tym samym zakłada się, że w obszarze pojedynczego wielokąta kąt padania światła jest stały.



Kolor jakim zostanie wypełniony wielokąt określany jest na podstawie pojedynczego wierzchołka. Korzysta się przy tym ze związanego z nim wektora normalnego oraz wybranego modelu oświetlenia.

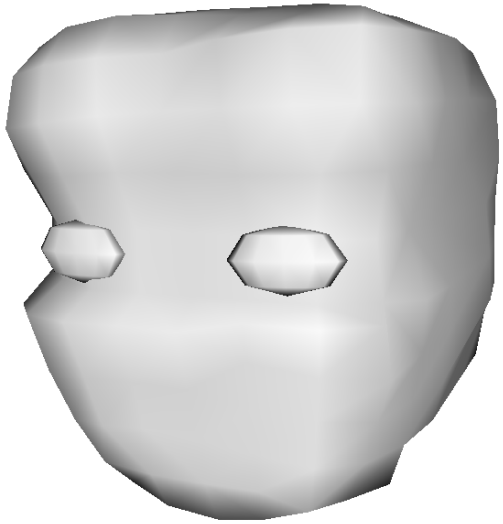
- obliczenie koloru z modelu oświetlenia, np. modelu Phonga:

$$I = k_a \cdot I_a + \sum_{i=1}^{K} I_p^{(i)} \cdot \left(k_d \cdot \cos\left(\theta^{(i)}\right) + k_s \cdot \cos^n\left(\beta^{(i)}\right) \right)$$

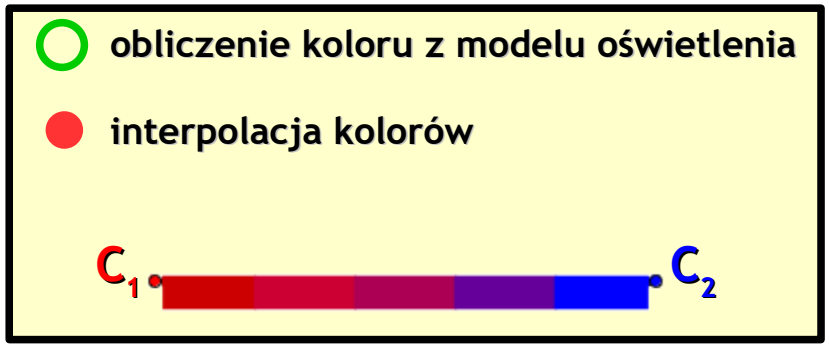


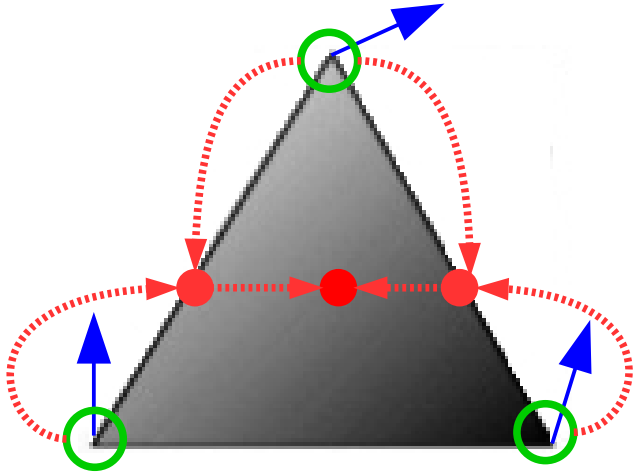
Rendering 3D: rendering w czasie rzeczywistym (cieniowanie Gourauda)

Cieniowanie Gourauda polega na przypisywaniu pikselom kolorów obliczonych przez interpolację liniową kolorów wyznaczonych dla wierzchołków wielokąta.



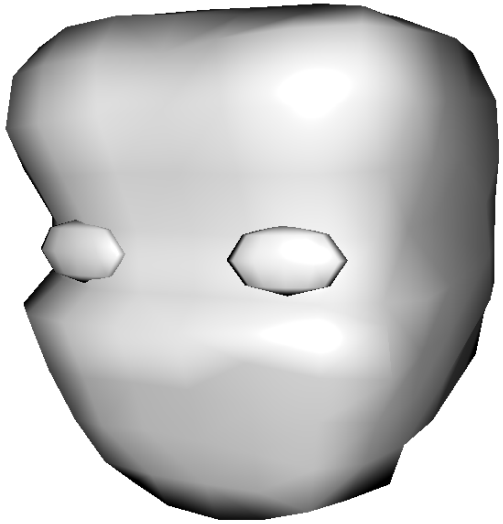
W pierwszej kolejności określane są kolory wszystkich wierzchołków wielokąta (przy użyciu modelu oświetlenia). Następnie wyznaczane są kolory pikseli krawędzi przez interpolację liniową kolorów par wierzchołków, a w końcu - kolory pikseli kolejnych wierszy wnętrza wielokąta, poprzez interpolację liniową barw krawędzi.



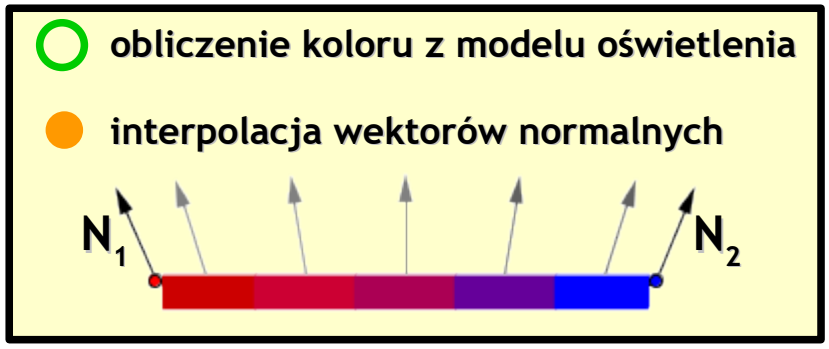


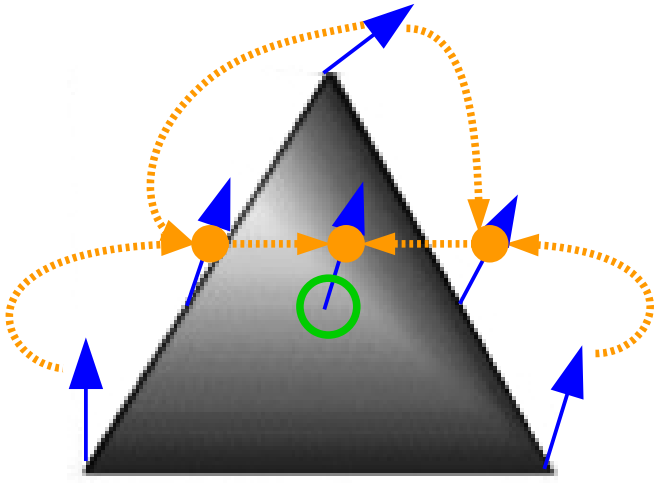
Rendering 3D: rendering w czasie rzeczywistym (cieniowanie Phonga)

W **cieniowaniu Phonga** (nie mylić z modelem oświetlenia tego samego autora) kolor każdego z pikseli wielokąta jest liczony niezależnie na podstawie modelu oświetlenia.



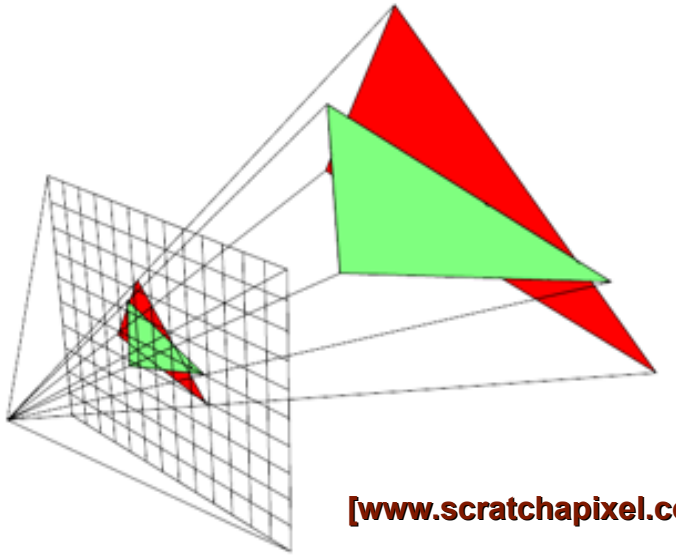
Również w tym przypadku wykorzystywana jest interpolacja liniowa, ale dotyczy ona wektorów normalnych. Znając wektory normalne dla wierzchołków można obliczyć interpolowane wektory każdego piksela wewnątrz wielokąta (analogicznie jak to ma miejsce dla barw w cieniowaniu Gourauda), a następnie wyznaczyć jego kolor.





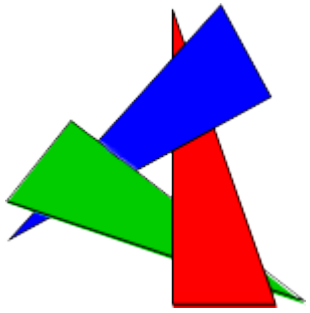
Rendering 3D: rendering w czasie rzeczywistym (problem widoczności)

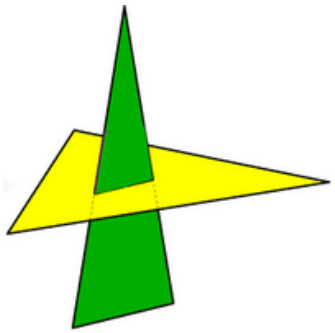
Po operacji rzutowania wszystkie wielokąty wchodzące w skład modeli obiektów są dane w postaci dwuwymiarowych figur na płaszczyźnie. Jednak w rzeczywistości pozycje ich wierzchołków charakteryzują się również pewną głębokością (niezerową wartością współrzędnej Z), co musi znaleźć odzwierciedlenie w obrazie.



[www.scratchapixel.com]

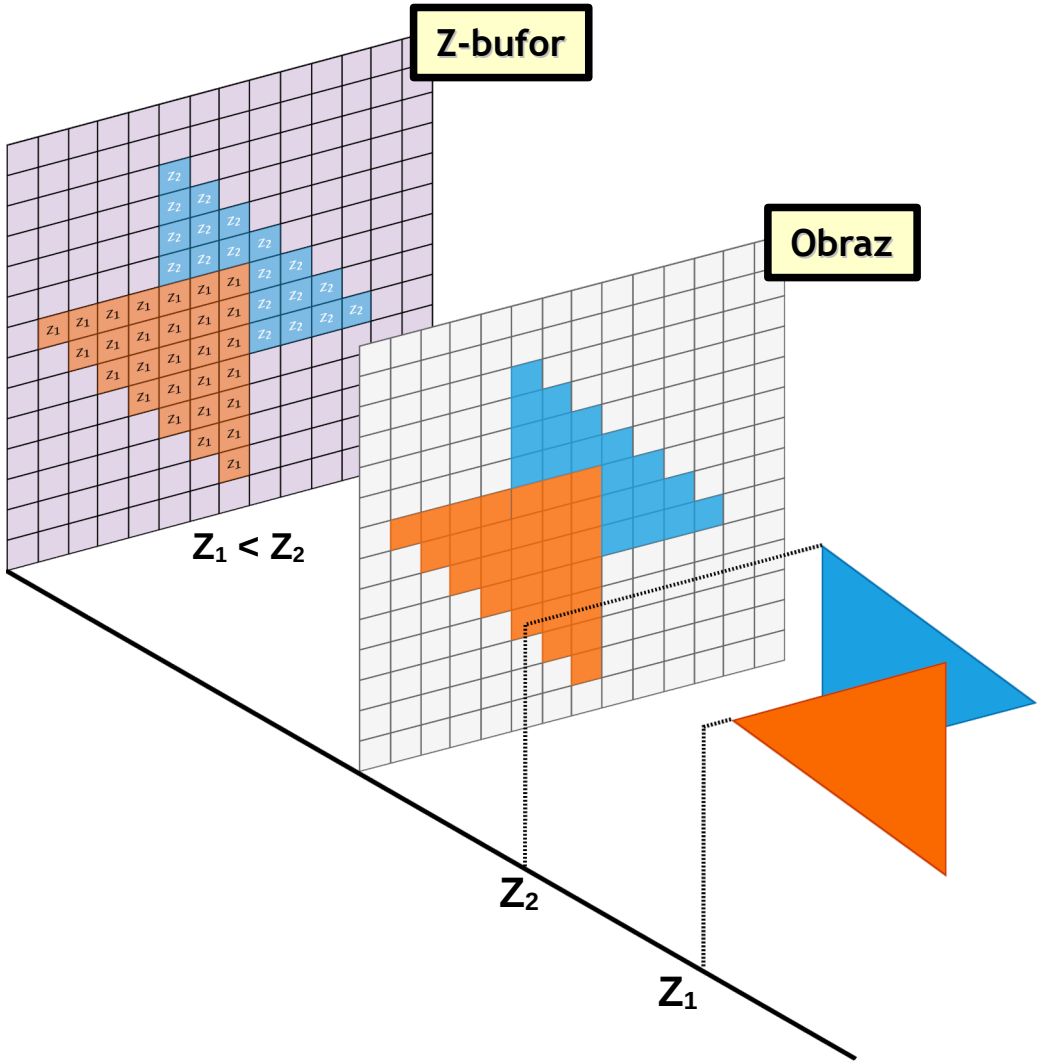
Na pierwszy rzut oka problem rozstrzygania widoczności wydaje się trywialny, ale istnieje cały szereg sytuacji, w których jego rozwiązanie wcale nie jest oczywiste.





Rendering 3D: rendering w czasie rzeczywistym (problem widoczności)

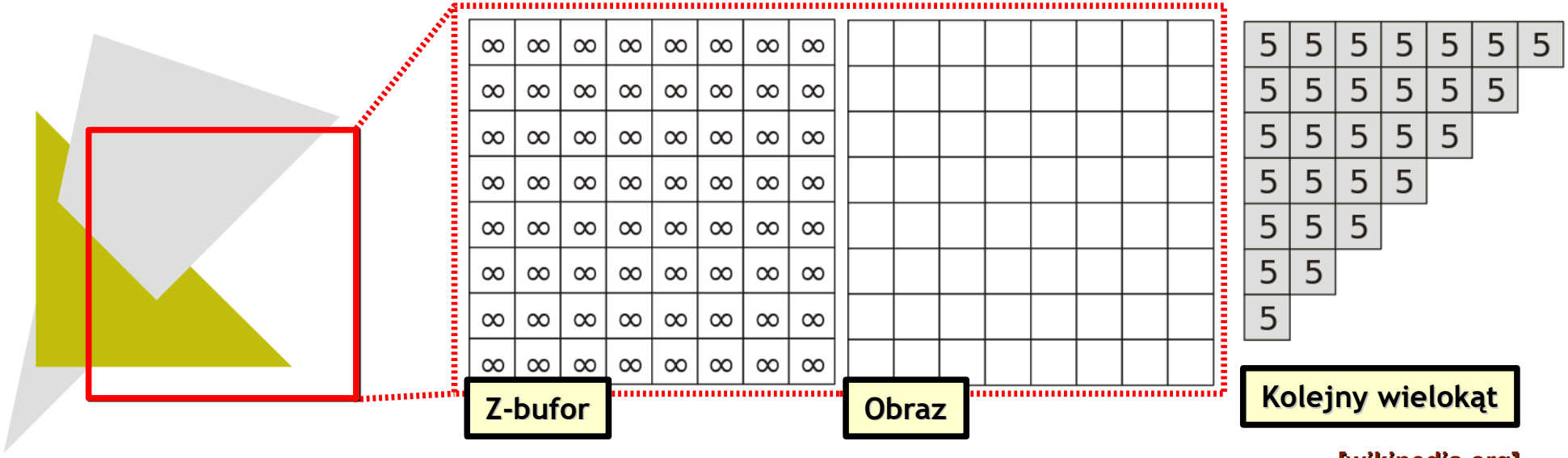
Przykładem popularnego – a jednocześnie bardzo prostego koncepcyjnie – podejścia do rozstrzygania widoczności jest użycie **bufora głębokości (Z-bufora)**, będącego dwuwymiarową tablicą liczb całkowitych (zwykle 24- lub 32-bitowych) o rozmiarze równym rozmiarowi obrazu. Każdy element bufora reprezentuje jeden piksel obrazu i służy do zapamiętania odpowiadającej mu głębokości (wartości współrzędnej *Z*).



Rendering 3D: rendering w czasie rzeczywistym (problem widoczności)

Algorytm tworzenia obrazu przy użyciu bufora głębokości wygląda następująco:

1. Wypełnić piksele obrazu barwą tła i nadać elementom Z-bufora wartość ∞.
2. Wypełniać rzuty kolejnych wielokątów na płaszczyznę, przy czym dla każdego piksela o współrzędnych (x_p, y_p) i głębokości z_p należy uprzednio sprawdzić wartość znajdującą się w odpowiadającym mu elemencie bufora:
   - jeżeli z_p jest mniejsze od zawartości bufora (czyli poprzednio rysowany wielokąt znajdował się dalej od obecnego), to piksel jest wypełniany kolorem wynikającym z algorytmu cieniowania, a jego głębokość jest zapamiętywana w buforze;
   - w przeciwnym razie piksel nie jest wypełniany.

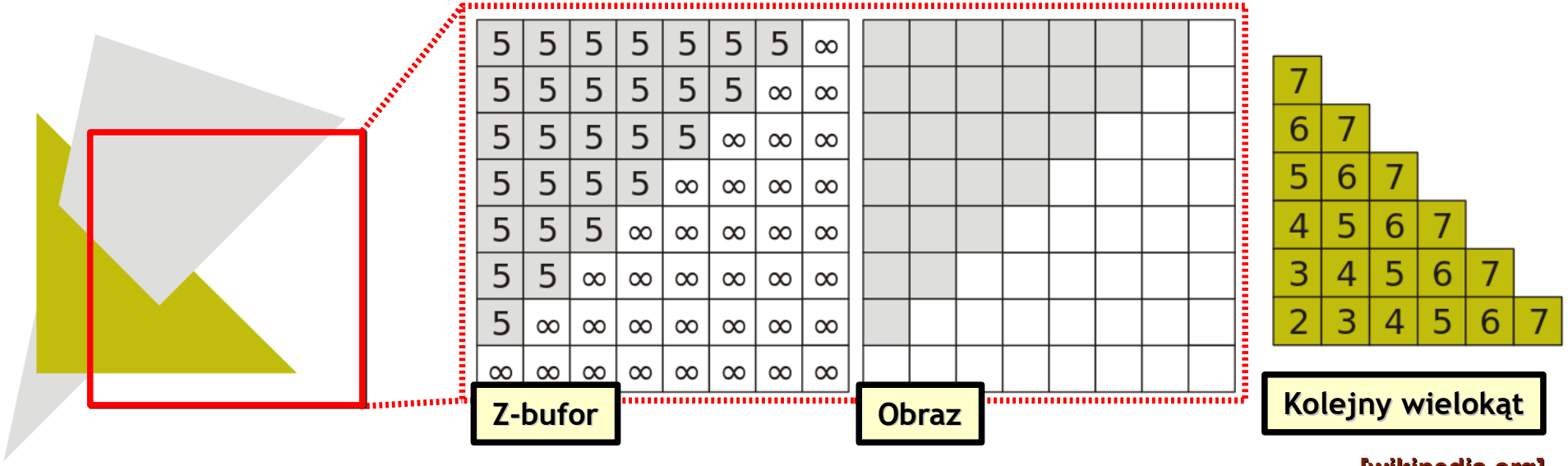


[wikipedia.org]

Rendering 3D: rendering w czasie rzeczywistym (problem widoczności)

Algorytm tworzenia obrazu przy użyciu bufora głębokości wygląda następująco:

1. Wypełnić piksele obrazu barwą tła i nadać elementom Z-bufora wartość ∞.
2. Wypełniać rzuty kolejnych wielokątów na płaszczyznę, przy czym dla każdego piksela o współrzędnych (x_p, y_p) i głębokości z_p należy uprzednio sprawdzić wartość znajdującą się w odpowiadającym mu elemencie bufora:
   - jeżeli z_p jest mniejsze od zawartości bufora (czyli poprzednio rysowany wielokąt znajdował się dalej od obecnego), to piksel jest wypełniany kolorem wynikającym z algorytmu cieniowania, a jego głębokość jest zapamiętywana w buforze;
   - w przeciwnym razie piksel nie jest wypełniany.

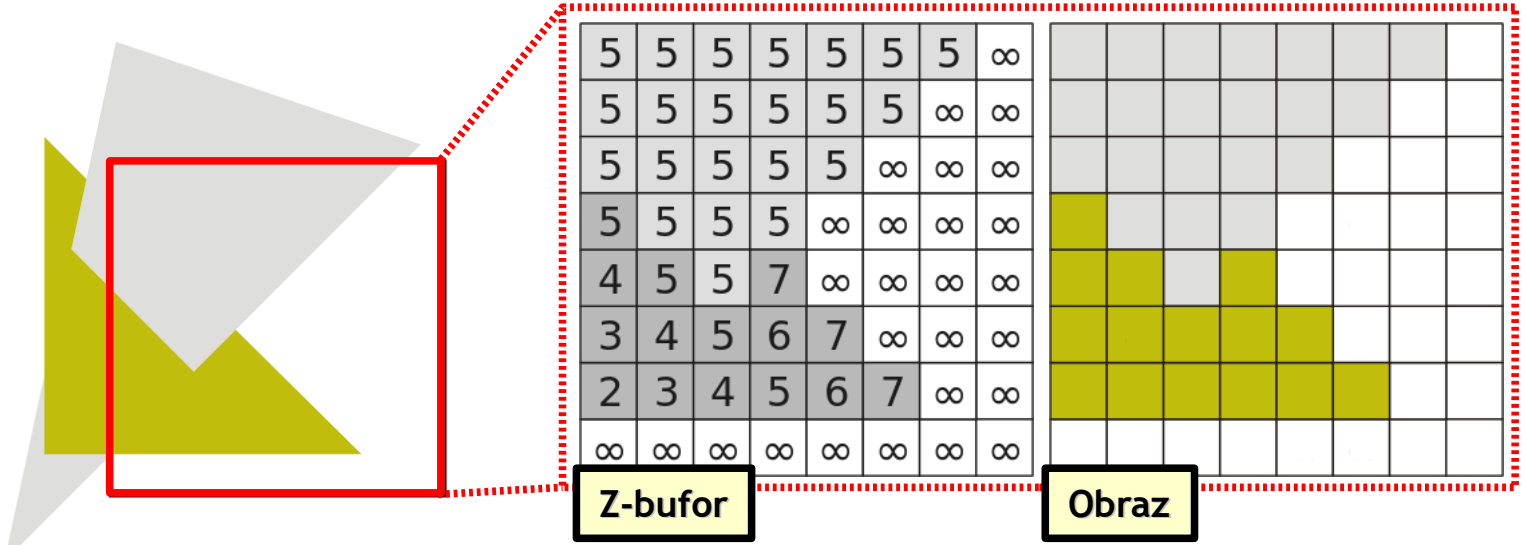


[wikipedia.org]

Rendering 3D: rendering w czasie rzeczywistym (problem widoczności)

Algorytm tworzenia obrazu przy użyciu bufora głębokości wygląda następująco:

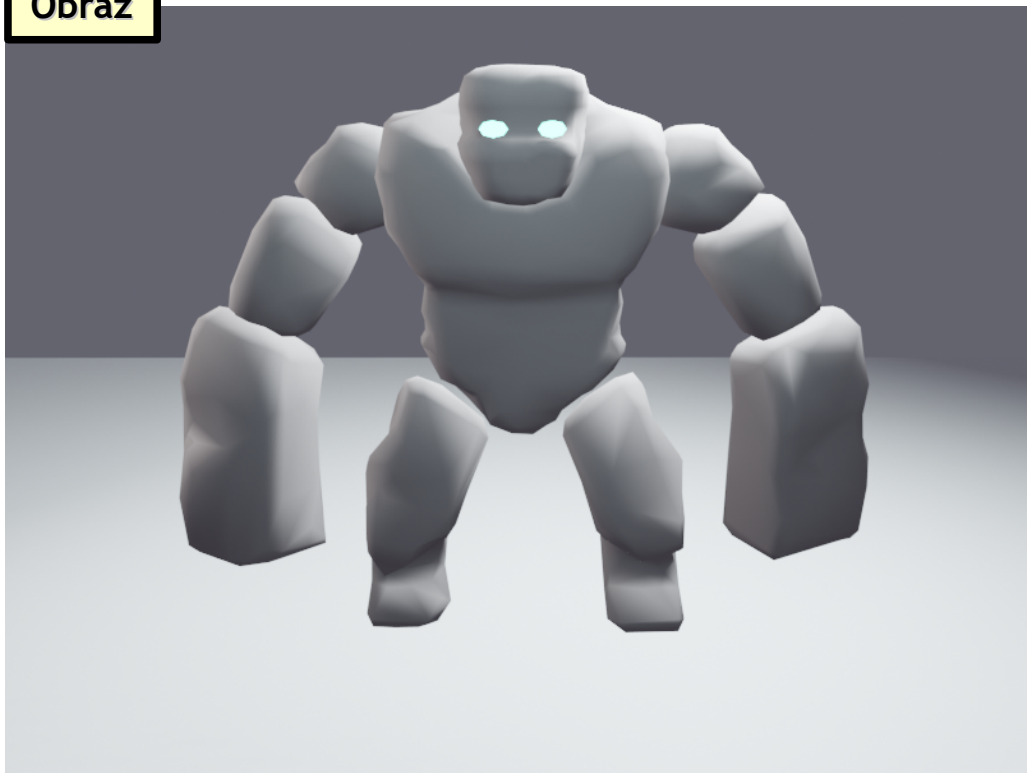
1. Wypełnić piksele obrazu barwą tła i nadać elementom Z-bufora wartość ∞.
2. Wypełniać rzuty kolejnych wielokątów na płaszczyznę, przy czym dla każdego piksela o współrzędnych (x_p, y_p) i głębokości z_p należy uprzednio sprawdzić wartość znajdującą się w odpowiadającym mu elemencie bufora:
   - jeżeli z_p jest mniejsze od zawartości bufora (czyli poprzednio rysowany wielokąt znajdował się dalej od obecnego), to piksel jest wypełniany kolorem wynikającym z algorytmu cieniowania, a jego głębokość jest zapamiętywana w buforze;
   - w przeciwnym razie piksel nie jest wypełniany.



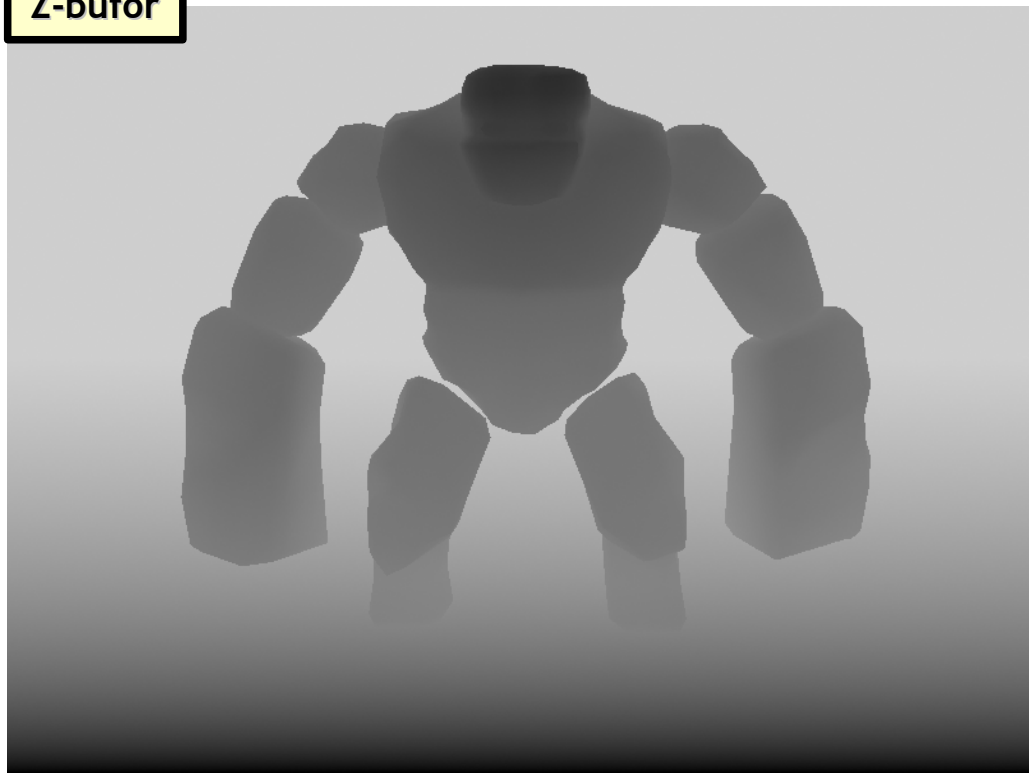
[wikipedia.org]

Rendering 3D: rendering w czasie rzeczywistym (problem widoczności)

Obraz

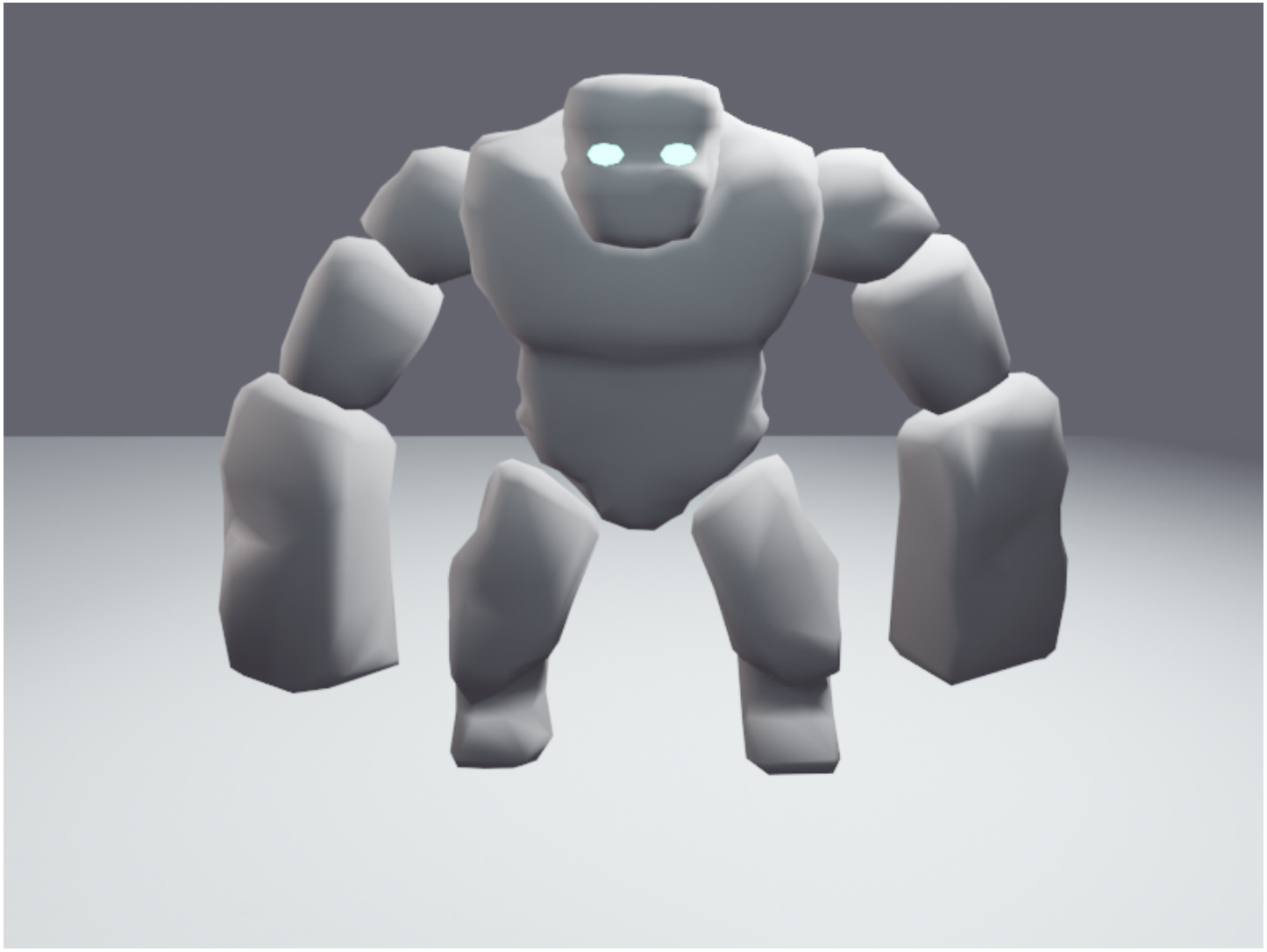


Z-bufor



Rendering 3D: rendering w czasie rzeczywistym (realizm)

Inaczej niż w przypadku techniki śledzenia promieni, **rasteryzacja rozwiązuje tylko problem oświetlania lokalnego**, bo bierze pod uwagę jedynie światło pochodzące bezpośrednio od źródeł, nie uwzględniając jego interakcji z innymi obiektami sceny. W efekcie renderowane obrazy cechują się relatywnie niskim stopniem realizmu.



Rendering 3D: rendering w czasie rzeczywistym (realizm)

Możliwe jest zmodyfikowanie podstawowej wersji algorytmu przez wprowadzenie dodatkowych kroków związanych z teksturowaniem i mapowaniem wypukłości oraz symulujących efekty optyczne związane z wpływem otoczenia na obiekty (np. cienie, odbicie i złamanie, emisję światła).

Rendering 3D: rendering w czasie rzeczywistym (realizm)

Możliwe jest zmodyfikowanie podstawowej wersji algorytmu przez wprowadzenie dodatkowych kroków związanych z teksturowaniem i mapowaniem wypukłości oraz symulujących efekty optyczne związane z wpływem otoczenia na obiekty (np. cienie, odbicie i złamanie, emisję światła).

Rendering 3D: rendering w czasie rzeczywistym (realizm)

[nvidia.com]

DZIĘKUJĘ ZA UWAGĘ

Materiały opracowane w ramach zadania 15 „Modyfikacja międzywydziałowych studiów I stopnia na kierunku Inżynieria Biomedyczna”
projektu „NERW PW. Nauka - Edukacja - Rozwój - Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Fundusze Europejskie
Wiedza Edukacja Rozwój

Politechnika Warszawska

Unia Europejska
Europejski Fundusz Społeczny