

Oświetlenie obiektów 3D

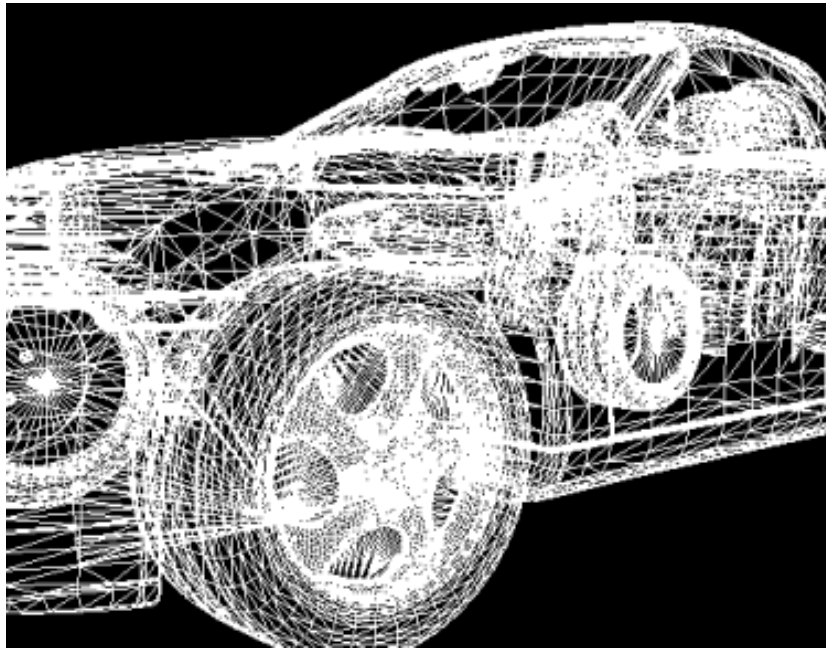
Opracowanie:

dr inż. Grzegorz Szwoch

Politechnika Gdańska

Katedra Systemów Multimedialnych

Rasteryzacja



„Spłaszczony” po
rzutowaniu obraz siatek
wielokątowych obiektów



Barwa każdego piksela
ustalona na podstawie
barwy zapisanej w modelu
oraz wpływu padającego
światła

Rasteryzacja

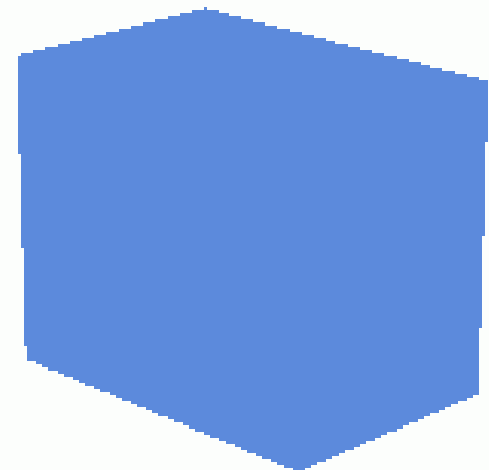
Mamy teraz płaski obraz 2D złożony z trójkątów, ale dla każdego *werteksa* mamy informacje dotyczące sceny 3D:

- położenie (x, y, z)
- barwa (dla różnych typów światła)
- współrzędne wektora normalnego

W procesie **rasteryzacji** barwa każdego piksela jest ustalana na podstawie interpolacji danych z wierzchołków oraz uwzględnienia wpływu światła padającego na scenę 3D.

Światło otoczenia (*ambient*)

- Bez kierunkowe, rozproszone światło.
- Oświetla wszystkie elementy sceny w identyczny sposób.
- Nie tworzy cieni, nie różnicuje oświetlenia.
- Daje w rezultacie płaski obraz, pozbawiony wrażenia trójwymiarowości.
- Przybliżenie np. światła w bardzo pochmurny dzień.



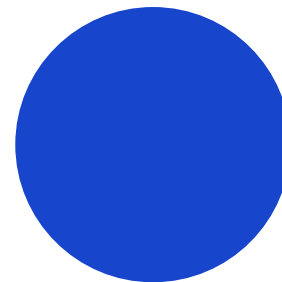
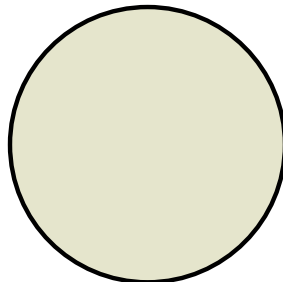
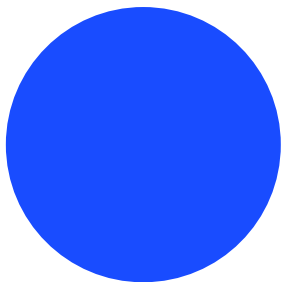
Barwa i natężenie światła

- Wektor opisuje natężenie oraz barwę światła.
- Opis składowych barwnych RGB
 - czerwona, zielona, niebieska (r, g, b).
- Liczby z zakresu od 0 do 1.
- Przykłady:
 - $(1, 1, 1)$: biała
 - $(0, 0, 0)$: czarna
 - $(1, 0, 0)$: czerwona
 - $(1, 0.5, 0)$: pomarańczowa

Barwa i natężenie światła

Przykład obliczeń:

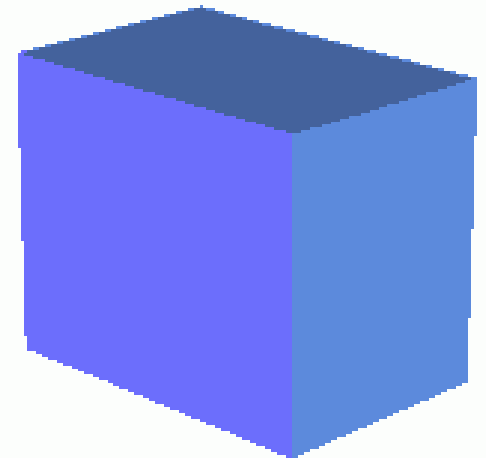
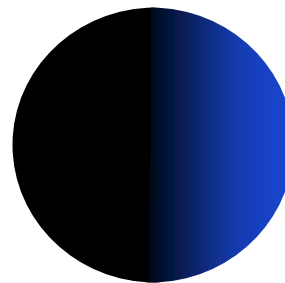
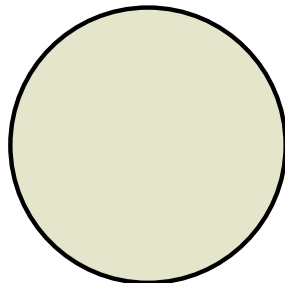
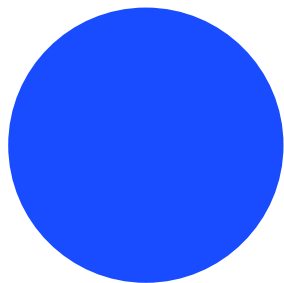
- barwa obiektu zapisana w wierzchołku:
(0.1, 0.3, 1.0)
- barwa światła otoczenia: (0.9, 0.9, 0.8)
- barwa wynikowa:
 $(0.1, 0.3, 1.0) \cdot (0.9, 0.9, 0.8)$
 $= (0.09, 0.27, 0.8)$



Światło kierunkowe

Światło kierunkowe (*directional, diffuse*)

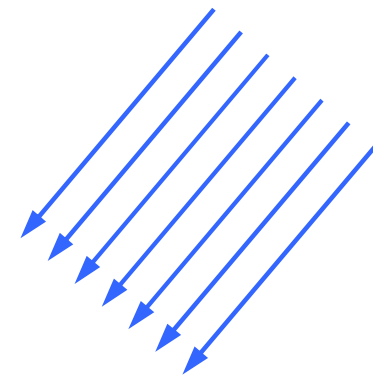
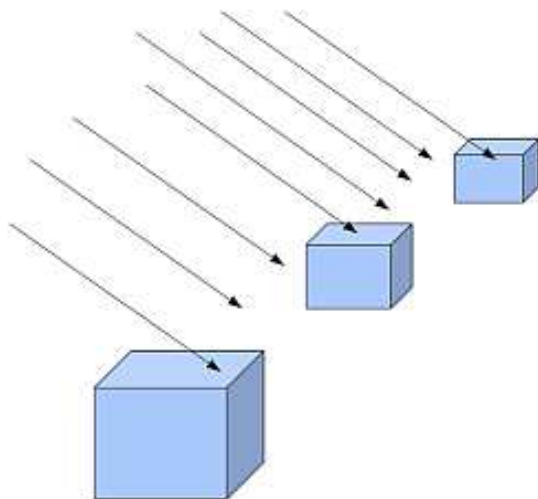
- pada z określonego kierunku
- w dowolnym punkcie sceny można wyznaczyć wektor kierunku światła
- wymaga obliczenia światła w wielu punktach
- różnicuje oświetlenie sceny
- tworzy cienie
- daje wrażenie trójwymiarowości



Źródło kierunkowe

Kierunkowe źródło światła (*directional light*):

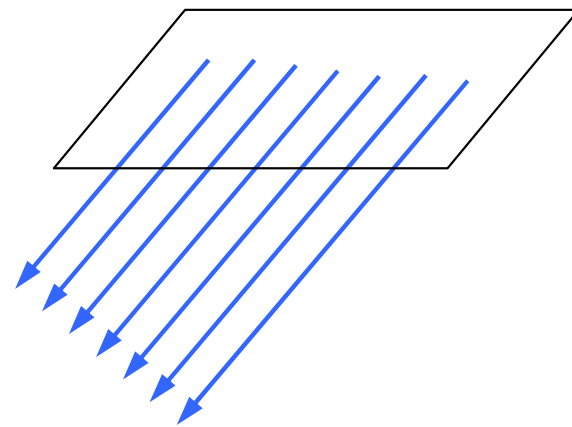
- promienie padające równoległe z określonego kierunku
- opis: kierunek (wektor) i barwa/natężenie
- stałe natężenie
- przykład: światło w słoneczny dzień



Źródło obszarowe

Obszarowe źródło światła (*area light*):

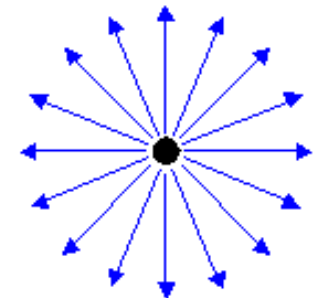
- tak jak źródło kierunkowe,
- ale obszar promieniowania jest ograniczony do pewnego obszaru.
- Przykład: okno w pomieszczeniu, ekran TV, neon, itp.



Źródło punktowe

Punktowe źródło światła (*point light*)

- rozchodzi się z punktu, we wszystkich kierunkach jednakowo
- opis: pozycja i barwa/natężenie
- natężenie maleje wraz z odległością od źródła
- wymaga obliczenia kierunku padania światła dla każdego elementu świata 3D
- nie ma odniesienia w rzeczywistym świecie



Źródło punktowe

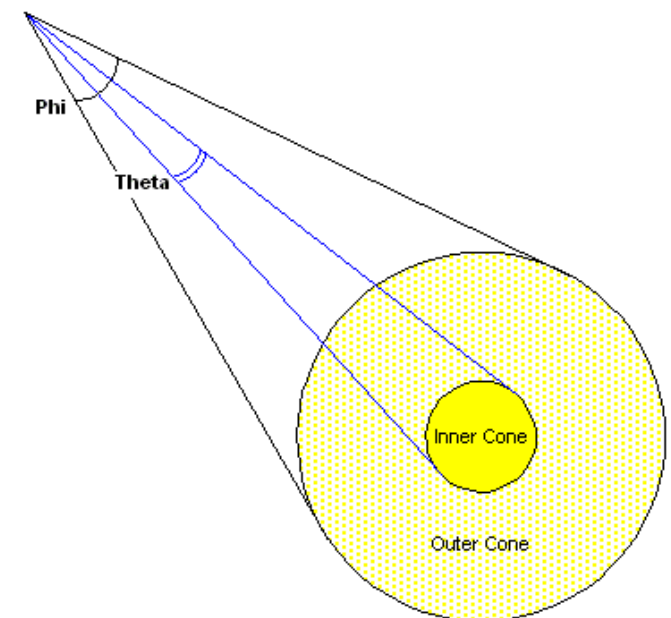
- Natężenie światła punktowego zależy od odległości od źródła.
- Natężenie maleje z kwadratem odległości od źródła.
- Stosuje się wiele modeli tłumienia.
- Przykładowy model (L_0 – natężenie w punkcie źródła, r – odległość od źródła):

$$L = \frac{L_0}{a_0 + a_1 r + a_2 r^2}$$

Reflektor

Reflektor (*spot light*):

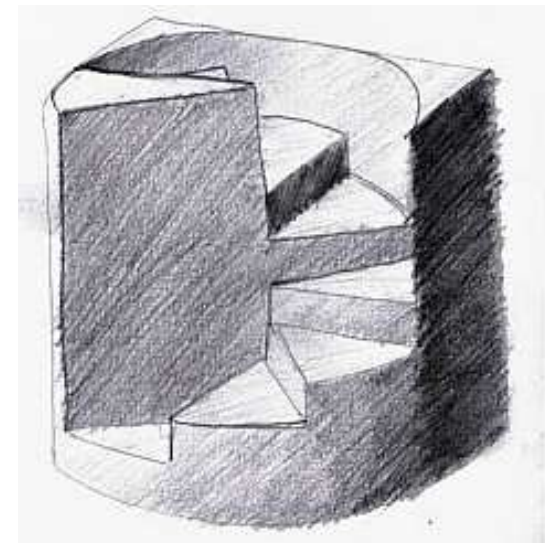
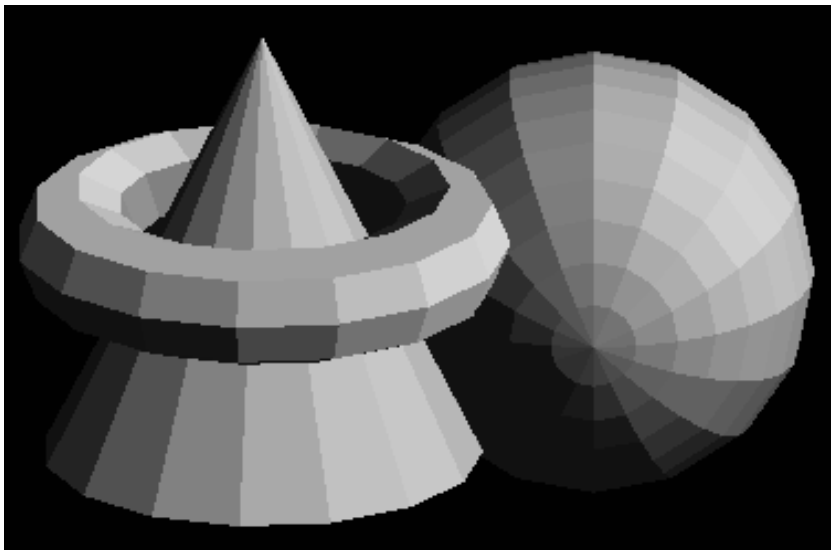
- źródło punktowe, z ograniczonym zakresem promieniowania
- opisane dodatkowo przez dwa stożki:
 - w środku wewnętrznego stożka
 - promieniowanie z pełną intensywnością
 - pomiędzy stożkami
 - stopniowe tłumienie
 - na zewnątrz stożków
 - brak światła
- przykład: latarnia uliczna, reflektor na scenie



Cieniowanie

Cieniowanie (*shading*):

- uwzględnienie wpływu światła kierunkowego na obiekty w świecie 3D
- „zacielenowanie” słabiej oświetlonych obszarów
- uwydatnia trójwymiarowość obiektów
- nie ma związku z rysowaniem cieni obiektów



Wektory normalne

Wektor normalny (*normal*) jest to wektor:

- prostopadły do powierzchni,
- o długości 1.

Wektor normalny trójkąta (*triangle normal*):

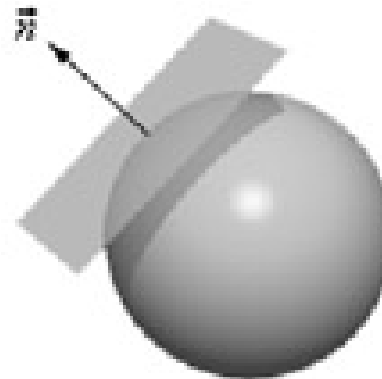
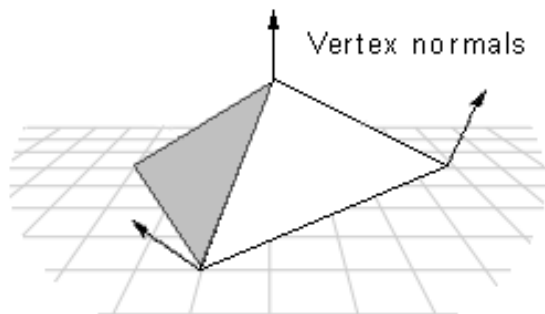
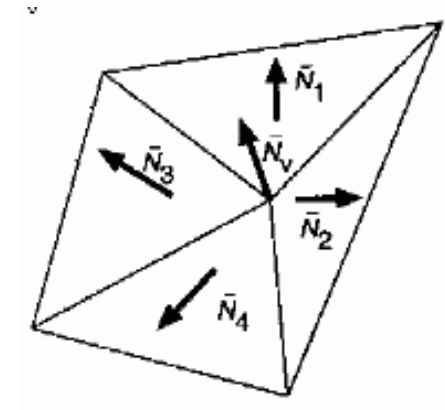
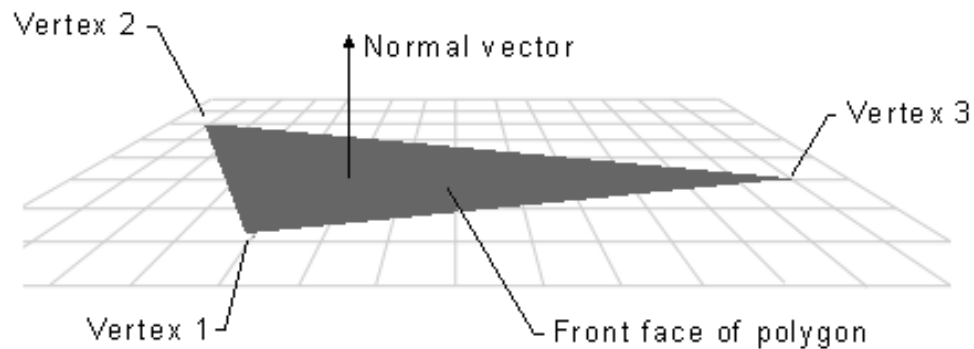
- prostopadły do trójkąta,
- „w górę” od przedniej ściany trójkąta.

Wektor normalny wierzchołka (*vertex normal*):

- uśrednione normalne wszystkich trójkątów, które łączy dany wierzchołek.

Wektory normalne

Wektory normalne trójkąta i wierzchołka



Wektor normalny trójkąta

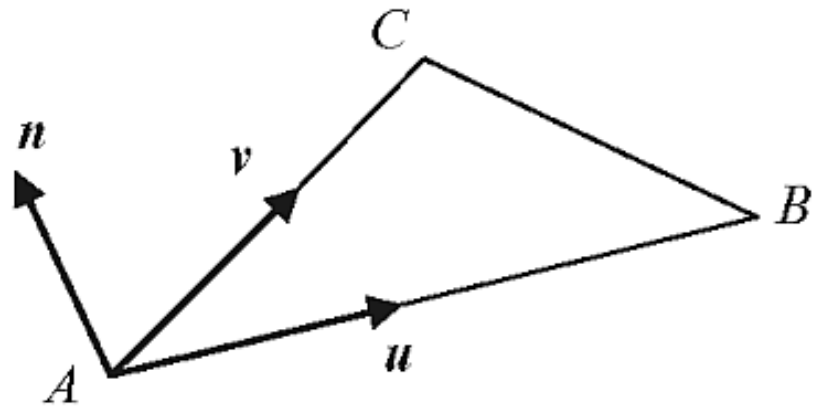
Obliczenie za pomocą **iloczynu wektorowego**.
Kolejność wektorów ma znaczenie!

$$\vec{n} = \vec{u} \times \vec{v}$$

$$n_x = u_y v_z - u_z v_y$$

$$n_y = u_z v_x - u_x v_z$$

$$n_z = u_x v_y - u_y v_x$$

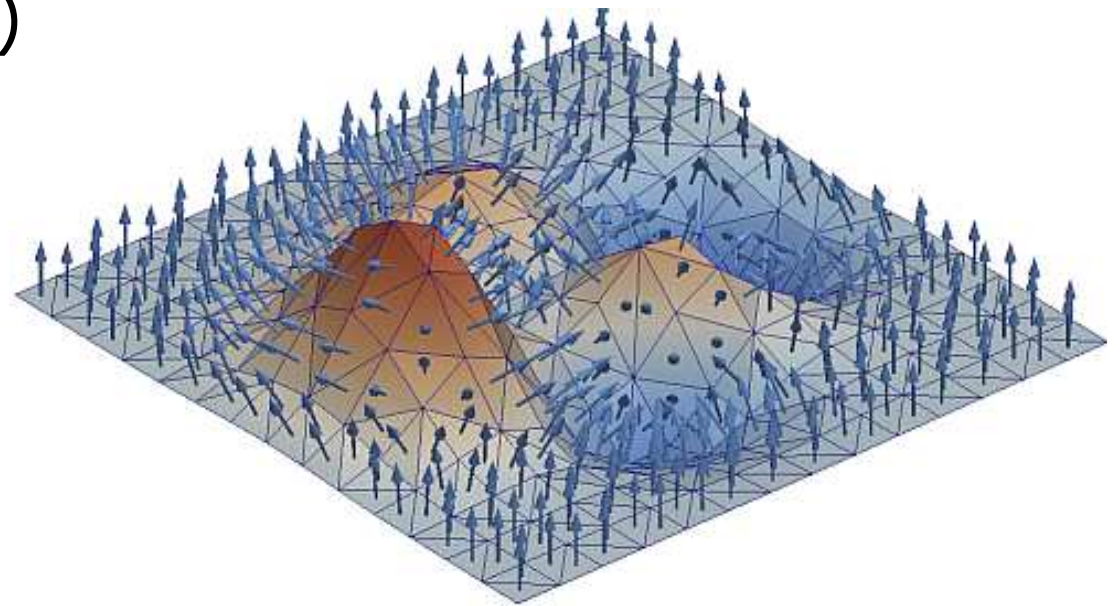


$$\vec{u} = \frac{B - A}{\|B - A\|}$$

$$\vec{v} = \frac{C - A}{\|C - A\|}$$

Wektory normalne

- Wektory normalne opisują orientację powierzchni obiektu.
- Są niezbędne do przeprowadzenia operacji cieniowania.
- Niemal zawsze są obliczane na etapie modelowania i **zapisywane w siatce obiektu**.
- Muszą być poddawane operacjom obrotu (ale nie skalowania!) wraz z obiektem.



Wektor światła

Wektor światła – jednostkowy wektor od badanego punktu (np. wierzchołka) do źródła światła.

Dla źródła kierunkowego:

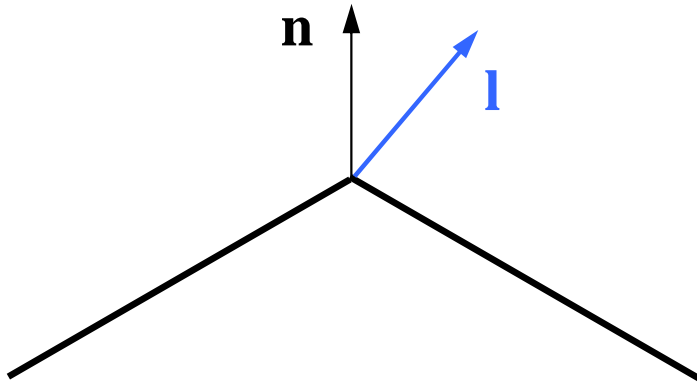
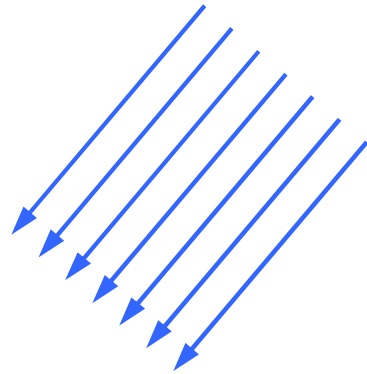
- odwrócenie kierunku światła,
- normalizacja

Dla źródła punktowego:

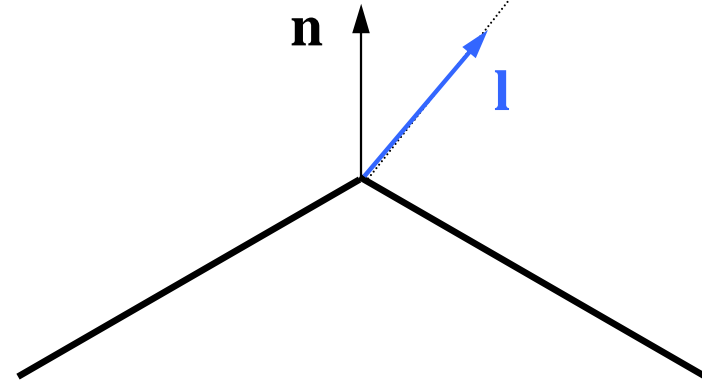
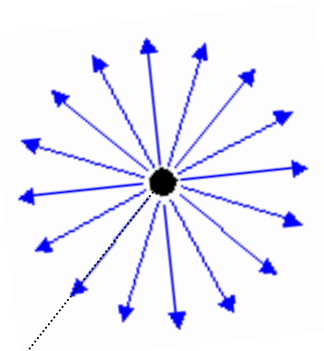
- wyznaczenie wektora do punktu światła,
- normalizacja

Wektor światła

światło
kierunkowe



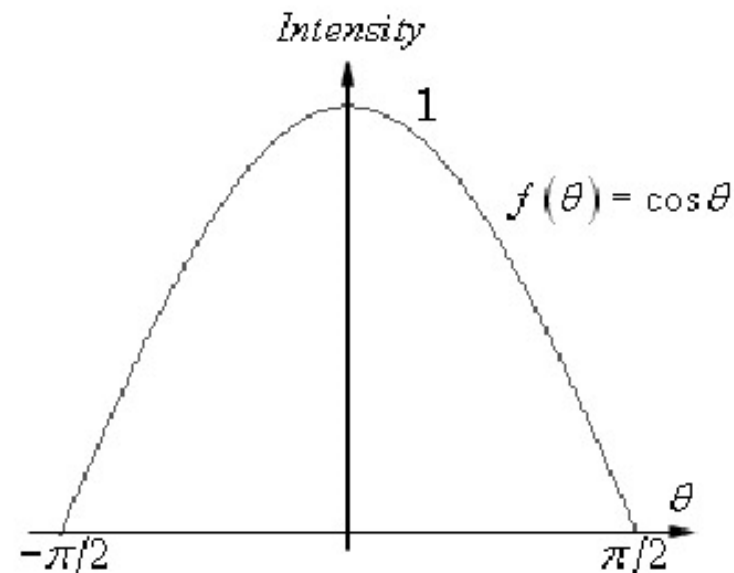
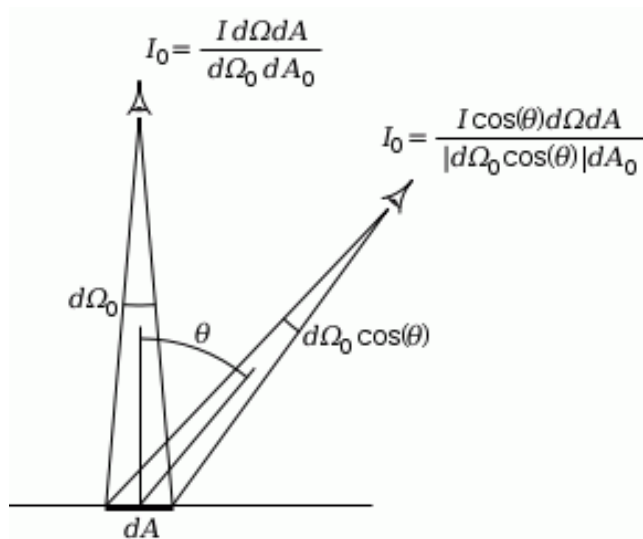
światło
punktowe



Model oświetlenia Lamberta

Model Lamberta jest najprostszym i najczęściej stosowanym modelem oświetlenia.

Natężenie oświetlenia powierzchni jest wprost proporcjonalne do cosinusa kąta pomiędzy wektorem kierunku światła a wektorem normalnym powierzchni.



Model oświetlenia Lamberta

Jak obliczyć cosinus kąta?

- \mathbf{n} – wektor normalny punktu
- \mathbf{l} – wektor jednostkowy do źródła światła
- iloczyn skalarny wektorów:

$$\vec{\mathbf{n}} \cdot \vec{\mathbf{l}} = n_x l_x + n_y l_y + n_z l_z$$

$$\vec{\mathbf{n}} \cdot \vec{\mathbf{l}} = \|\vec{\mathbf{n}}\| \cdot \|\vec{\mathbf{l}}\| \cdot \cos(\theta)$$

$$\cos(\theta) = n_x l_x + n_y l_y + n_z l_z$$

$$f(\theta) = \max(\cos(\theta), 0)$$

Model oświetlenia Lamberta

Przykład obliczeń (źródło kierunkowe):

- barwa obiektu zapisana w modelu:
 $(0.1, 0.3, 1.0)$
- barwa światła kierunkowego: $(0.9, 0.9, 0.8)$
- bez uwzględnienia modelu oświetlenia:
 $(0.1, 0.3, 1.0) \cdot (0.9, 0.9, 0.8)$
 $= (0.09, 0.27, 0.8)$
- obliczenie kąta padania:
 $\theta = 30^\circ, \cos(\theta) = 0.866$
- wynikowa barwa punktu:
 $0.866 \cdot (0.09, 0.27, 0.8) = (0.08, 0.23, 0.69)$

Cieniowanie płaskie

Cieniowanie płaskie (*flat shading*).

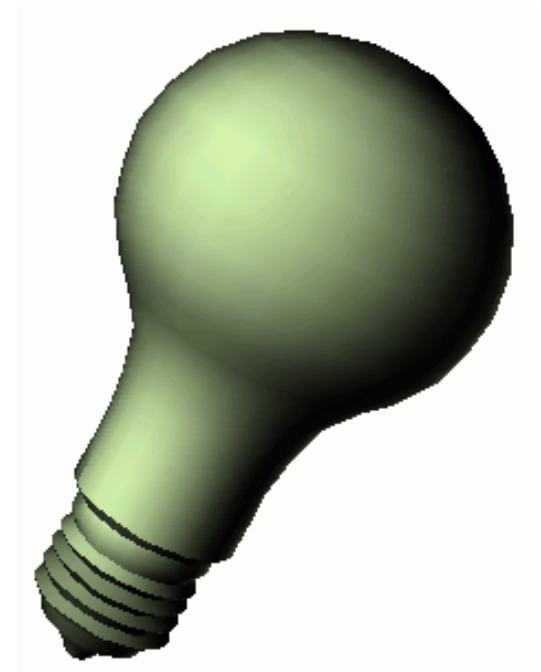
- Model oświetlenia jest stosowany jeden raz do każdego wielokąta siatki.
- Tworzy efekt w postaci barwnych wielościanów.
- Jest szybkie – z tego względu bywa stosowane do „wstępnych renderów” jako szybki podgląd efektu.



Cieniowanie Gourauda

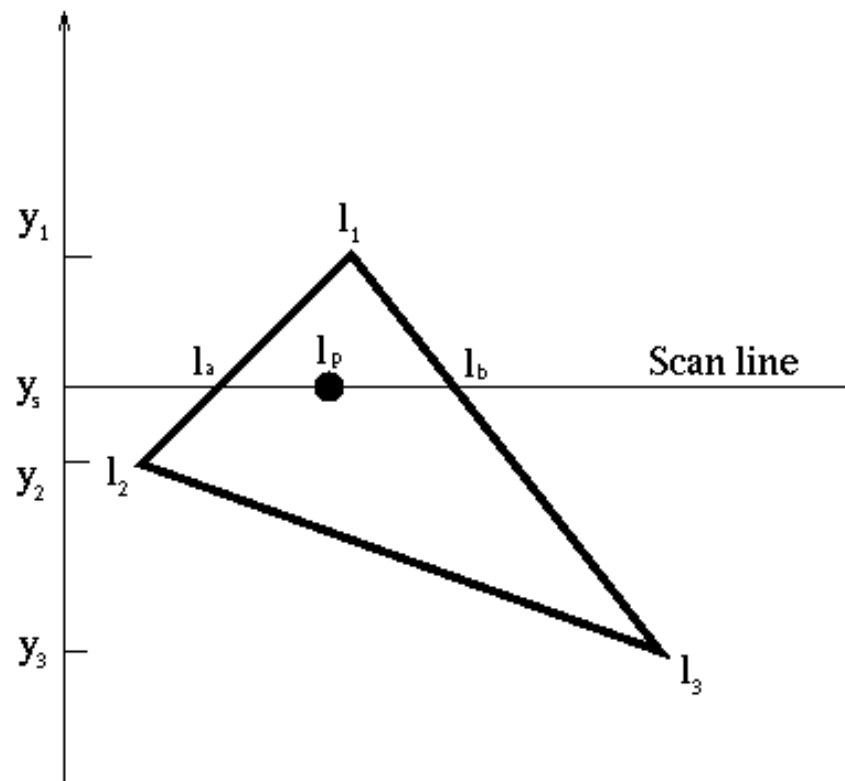
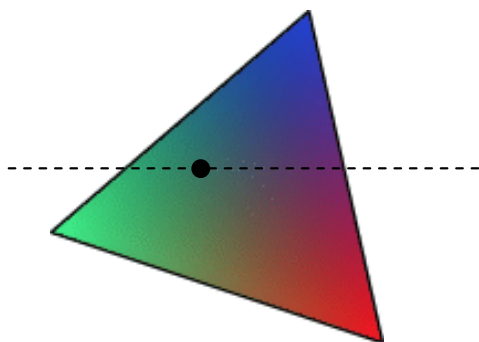
Cieniowanie Gourauda:

- Zastosowanie modelu oświetlenia do każdego wierzchołka siatki obiektu.
- **Interpolacja** obliczonej **barwy**:
 - wzdłuż krawędzi trójkąta,
 - wzdłuż linii poziomej.
- Wygładza przejścia pomiędzy barwami.
- Inna nazwa:
oświetlenie wierzchołkowe
(*vertex lighting*)



Cieniowanie Gourauda

- Programista oblicza barwę w wierzchołkach: l_1, l_2, l_3
- Rasteryzer dokonuje interpolacji wzdłuż krawędzi trójkąta: obliczenie l_a, l_b
- Następnie interpoluje wzdłuż linii skanowania (*scanline*): obliczenie l_p



Materiał

Materiał (*material*) opisuje charakter powierzchni obiektu – zdolność do odbijania światła:

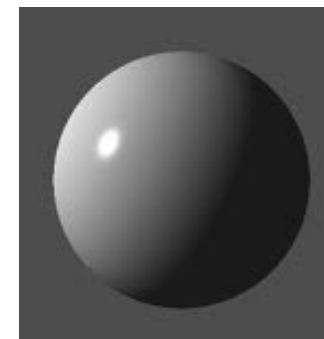
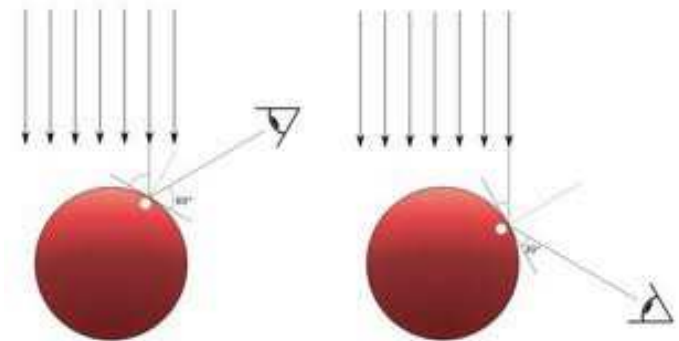
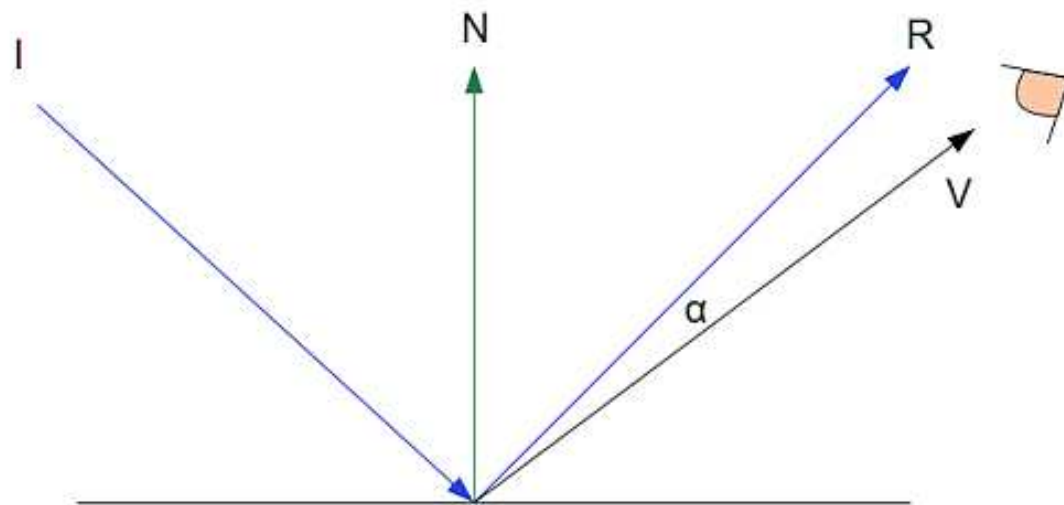
- powierzchnia **błyszcząca** – odbija światło, powstają odblaski (*specular*), przykład: kula bilardowa
- powierzchnia **matowa** – pochłania światło, nie powstają odblaski, przykład: gumowa piłka

Materiał jest definiowany za pomocą **barwy odblasku** (*specular light*), zapisanej w modelu:

- bliska białej $(1,1,1)$ – pow. błyszcząca,
- bliska czarnej $(0,0,0)$ – pow. matowa

Odblaski (specular)

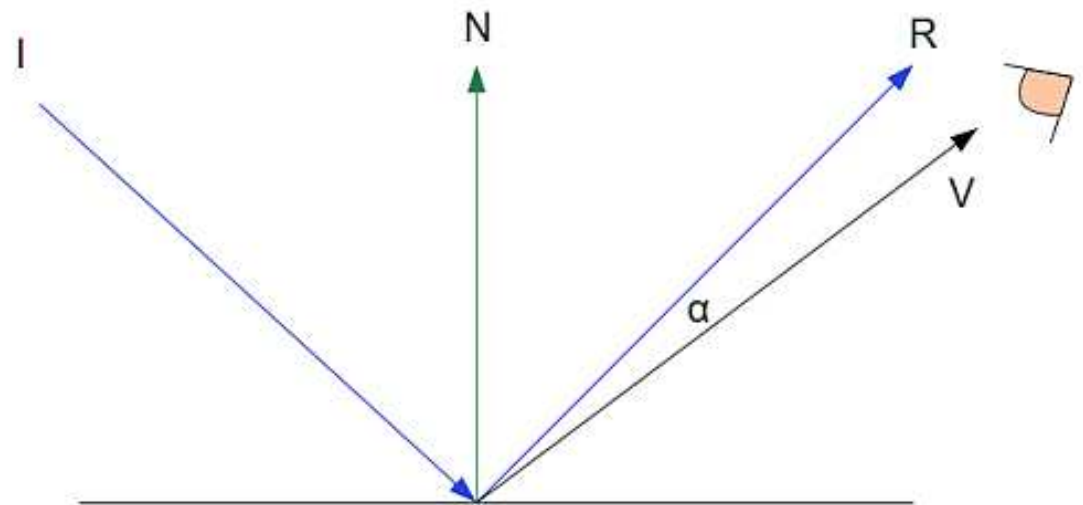
Odblask powstaje gdy światło kierunkowe odbija się od powierzchni obiektu i trafia w pobliże kamery.



Odblaski (specular)

Obliczenie odbłasku:

- obliczenie wektora odbitego światła R (kąt odbicia = kąt padania)
- obliczenie wektora V od punktu do kamery
- obliczenie cosinusa kąta α (iloczyn skalarny)
- podniesienie wyniku do potęgi p
 - określa twardość (skupienie) odbłasku
- przemnożenie przez barwę odbłasku



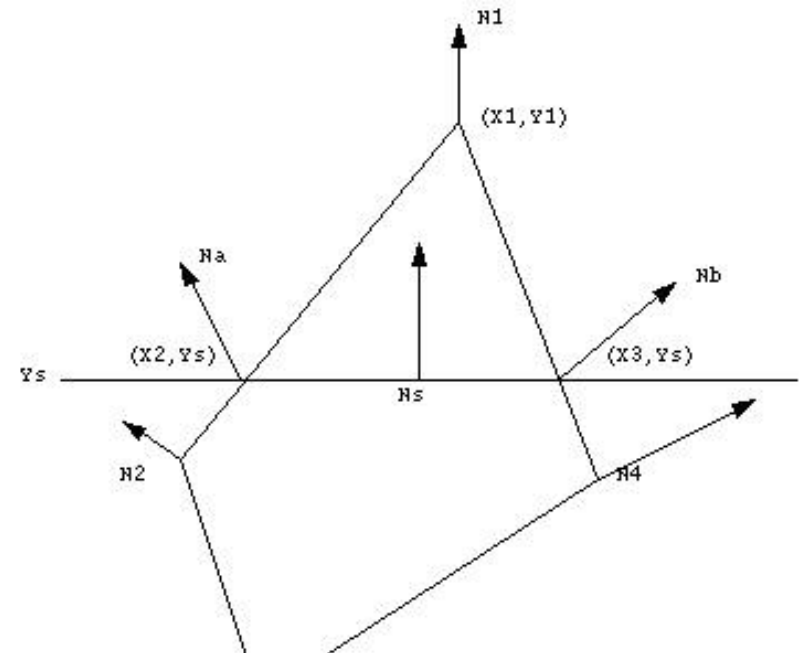
Odblaski (specular)

- Metoda cieniowania Gourouda nie nadaje się do dokładnego odwzorowania odbłasków.
- Gdy promień pada na wierzchołek – powstaje odbłask.
- Gdy promień pada pomiędzy wierzchołki - zostaje „rozmyty”.
- Zniekształcenia są wyraźnie widoczne gdy obiekt się obraca.
- Potrzebna jest metoda cieniowania, która stosuje model oświetlenia do każdego fragmentu niezależnie.

Cieniowanie Phong

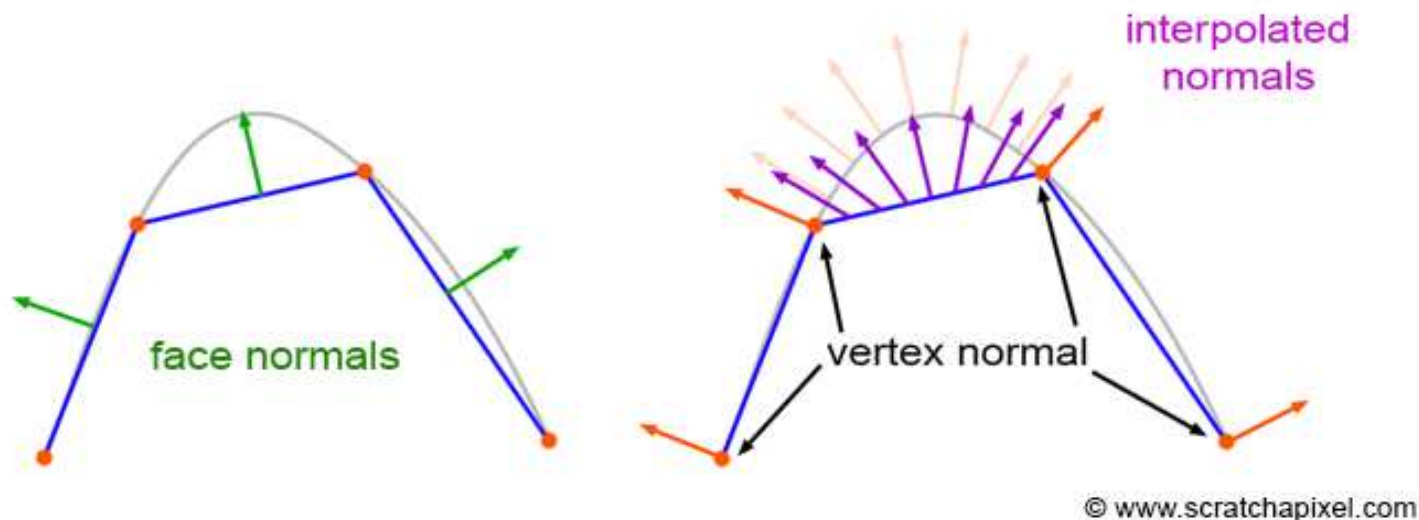
Metoda cieniowania Phong:

- **interpolowany** jest **wektor normalny**,
- model oświetlenia jest stosowany do każdego fragmentu, z użyciem interpolowanego wektora normalnego
- obliczona barwa jest zapisywana w pikselu
- znacznie więcej obliczeń
- dokładność dla odbłasków
- inna nazwa: oświetlenie fragmentowe (*fragment lighting*)

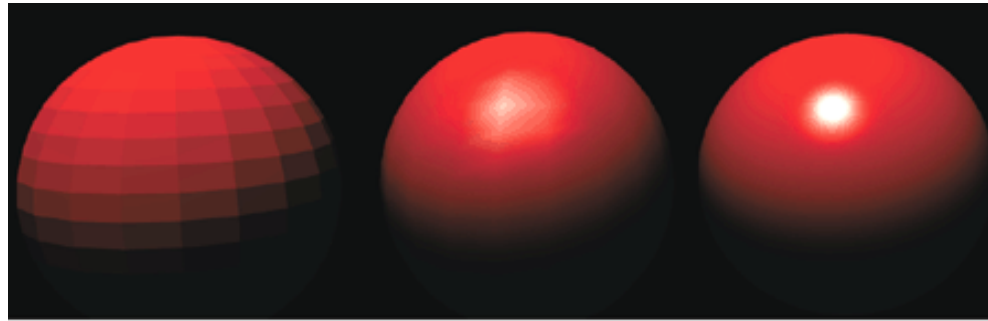


Cieniowanie z interpolacją

- Cieniowanie płaskie uwidacznia siatkę wielokątową obiektu (wielościan).
- Metody Gourouda i Phong'a stosują interpolację, która „wygładza powierzchnię”.
- Dzięki temu uzyskujemy wrażenie płynnego przejścia barwy.



Cieniowanie z interpolacją



Flat

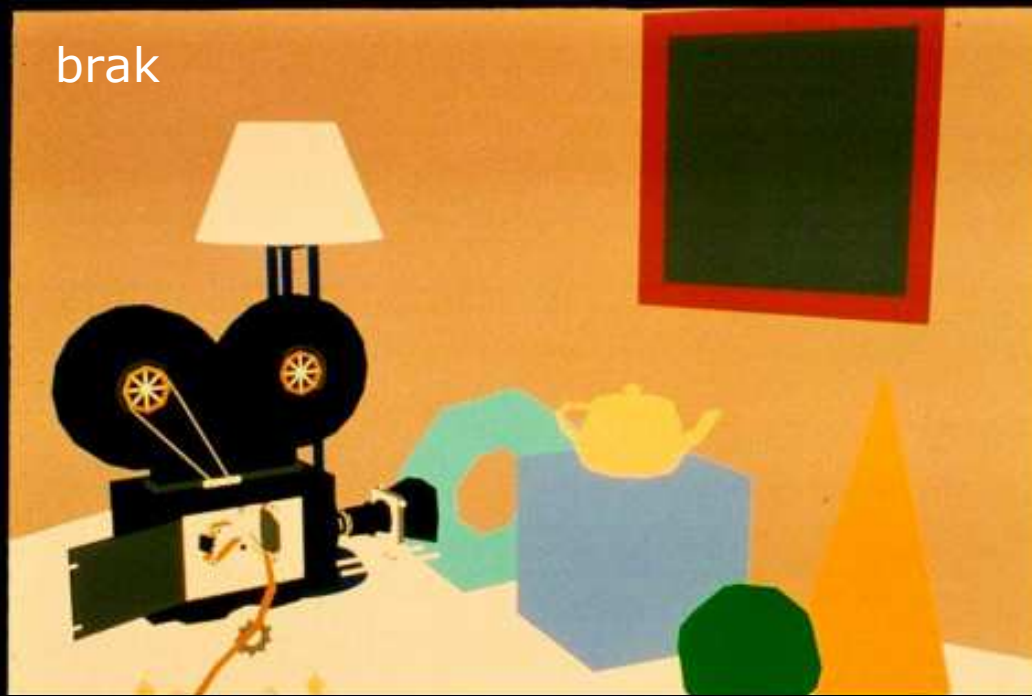
Gouraud

Phong

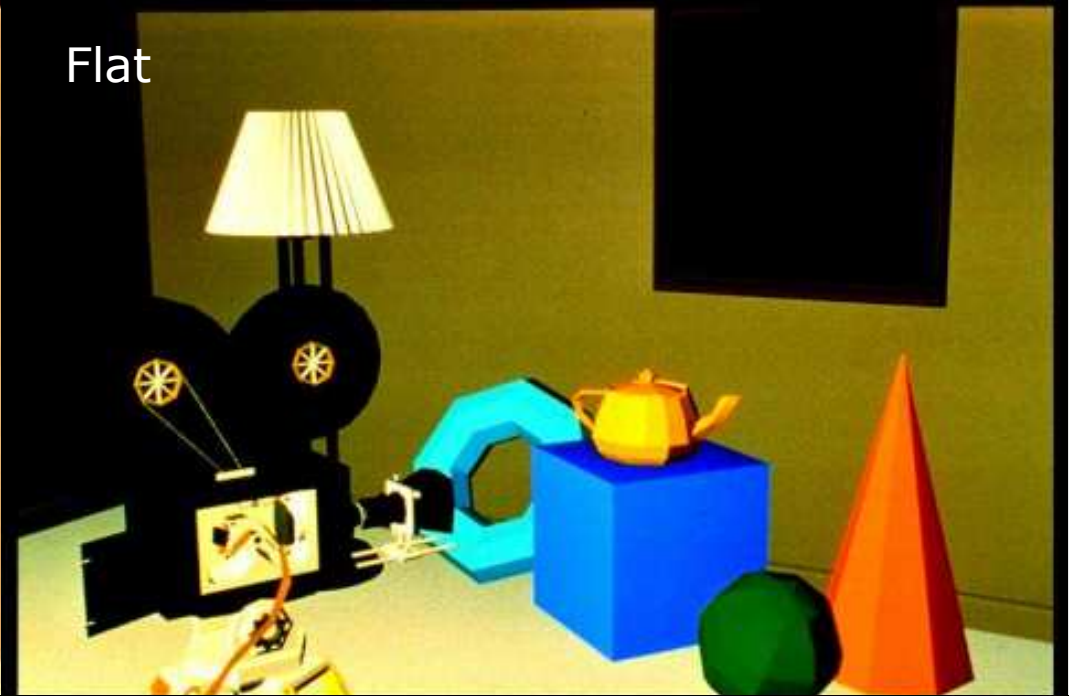


Porównanie metod cieniowania

brak



Flat



Gouraud



Phong



Model ADS

Model oświetlenia ADS uwzględnia trzy rodzaje światła, stosowane równocześnie:

- A – światło otoczenia (*ambient*)
- D – światło kierunkowe (*diffuse, directional*)
- S – odbłaski świetlne (*specular*).

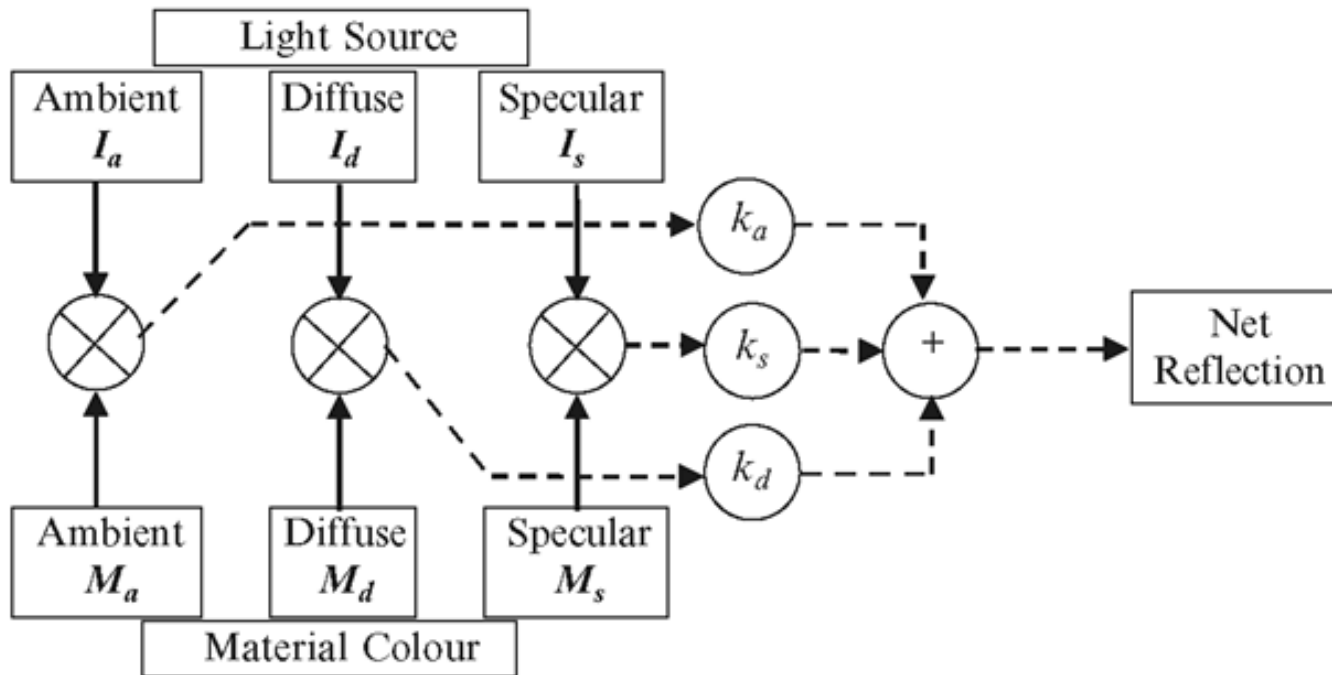
Każdy typ oświetlenia jest definiowany osobno:

- dla źródła światła (światło emitowane),
- dla wierzchołka siatki (światło odbite).

Wynikowa barwa piksela: suma oświetlenia

$A + D + S$

Model ADS



Model ADS

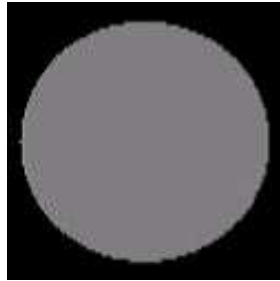
- Nie musimy podawać każdej z 6 barw. Jeżeli chcemy pominąć wpływ jednego ze światła, podajemy wartość „biała” (1,1,1).
- Typowo podajemy barwy:
 - źródła światła otoczenia (globalnie)
 - źródła światła kierunkowego (dla każdego źródła)
 - barwy odbłasku modelu (dla każdego wierzchołka modelu).
- Barwa światła kierunkowego dla modelu jest zwykle próbkowana z tekstury.

Model ADS

Dlaczego używamy **światła otoczenia** (*ambient*)?

- Rzeczywista scena jest oświetlona światłem **odbitym** od powierzchni.
- Model ADS w rasteryzacji czasu rzeczywistego uwzględnia **tylko światło bezpośrednie** – pomija odbicia.
- Obszary sceny nie oświetlone światłem bezpośrednim pozostałyby czarne.
- Aby doświetlić scenę, dodajemy światło otoczenia.

Przykład obliczania modelu ADS



Barwa
padającego
światła

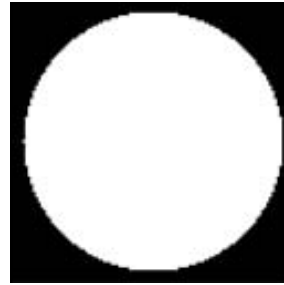
ambient



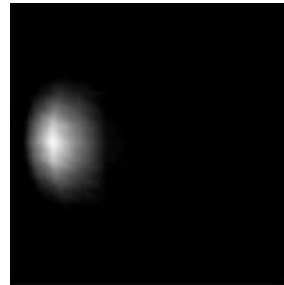
diffuse



specular



Barwa
zdefiniowana
w wierzchołku



Wynik dla
danego
światła



Wynik łączny

Podsumowanie - obliczenie cieniowania

- Barwy światła odbitego – zapisane w modelach.
- Typ, barwy i pozycje/kierunki źródeł światła – zdefiniowane w opisie sceny.
- Wybrany model cieniowania – np. Phong.
- Dla każdego ze źródeł światła kierunkowego:
 - obliczyć kąt padania światła,
 - obliczyć barwę światła kierunkowego,
 - obliczyć odbłask.
- Dodać światło otoczenia.
- Suma barw wszystkich światel A , D , S jest przenoszona na piksel obrazu.

Podsumowanie - opis wierzchołka w modelu

Jeden wierzchołek obiektu:

- współrzędne wierzchołka (x, y, z)
- współrzędne wektora normalnego $[x, y, z]$
- barwa odbitego światła otoczenia A
(zwykle biała: $1,1,1$)
- barwa odbitego światła kierunkowego D
(zwykle biała: $1,1,1$)
- materiał – barwa odblasku S