



Politechnika Łódzka

Instytut Informatyki

Instrukcja do laboratorium

Fotorealistyczna Grafika Komputerowa

Laboratorium IV

Źródło światła, materiał, model Phong

Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej

dr inż. Piotr Napieralski

mgr inż. Krzysztof Guzek

Łódź, 28.02.2012

Instytut Informatyki

90-924 Łódź, ul. Wólczańska 215, budynek B9

tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: napieral@ics.p.lodz.pl



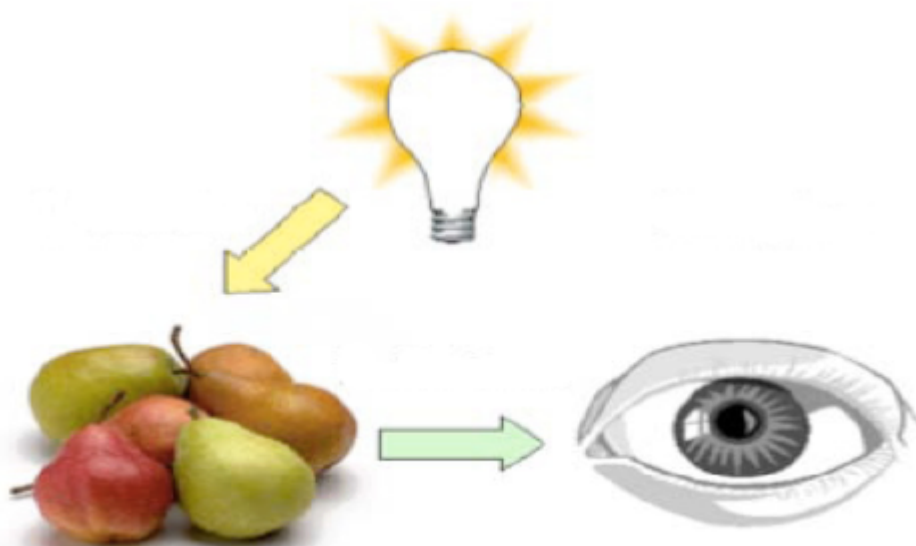
1. Wstęp

*„Wtedy Bóg rzekł: «Niechaj się stanie światłość!»
I stała się światłość. Bóg widząc, że światłość jest
dobra, oddzielił ją od ciemności. I nazwał Bóg
światłość dniem, a ciemność nazwał nocą¹.”*

Gdy otworzymy oczy pierwszą rzeczą jaką widzimy jest światło. To ono odpowiedzialne jest za odbieranie świata przez nasze oczy. Spróbujemy zastanowić się czym jest światło i jaki wpływ ma na postrzeganie otaczającego nas świata.

1.1 Recepcja światła

Najogólniej ujmując światło to fala elektromagnetyczna, przemieszczająca się w przestrzeni z prędkością 299792458 m/s. Światło jest energią w najczystszej postaci. Kolejną istotną właściwością światła jest jego dwoistość, czyli może być traktowane jako fala jak i cząsteczka. Z naszego punktu widzenia, mieszkańców ziemi najwięcej światła mamy podczas dnia, by podczas nocy jego ilość znacząco się zmniejszyła. Światło docierające do nas podczas nocy pochodzi w większości przez odbicie od księżyca i nieba. Możemy przyjąć, że główne źródło energii świetlnej na ziemi pochodzi od słońca, które emituje światło we wszystkich kierunkach niczym duże źródło punktowe. Kierunek rozchodzenia się promieni świetlnych pokrywa się z kierunkiem rozchodzenia się energii, która wywołuje reakcję w odbiorniku (np. oku).

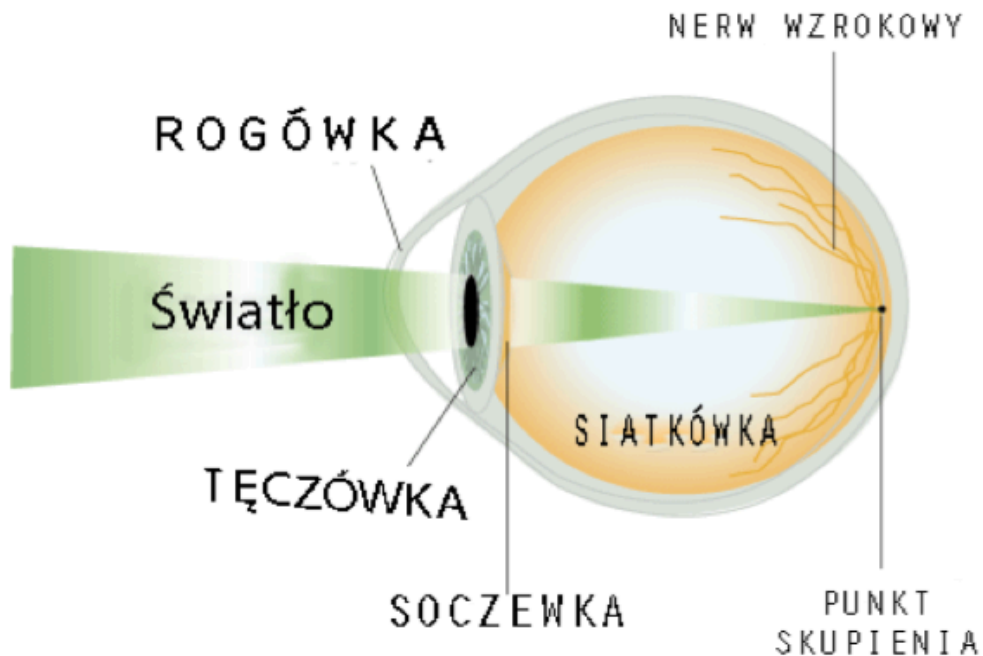


Rys.1 Powstawanie obrazu w oku.

Promień świetlny odbija się od obiektu i dociera do obserwatora

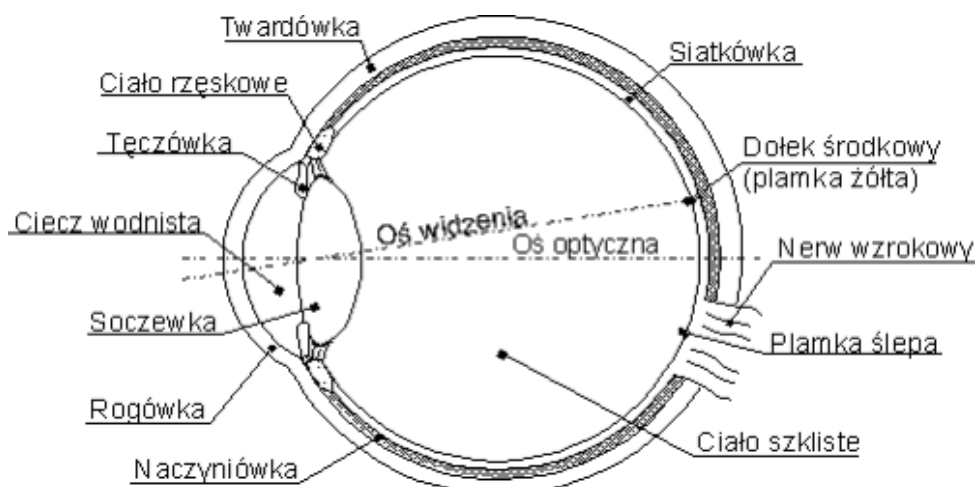
¹ STARY TESTAMENT Księga Rodzaju

Dowolny układ optyczny dokonuje nie tylko przekształceń geometrycznych (przedmiot-obraz), ale również przekształceń energetycznych. Światło widzialne wywołuje więc wrażenia barwne. Prawidłowe oko człowieka zbudowane jest w ten sposób, że na siatkówce otrzymywany jest ostry obraz obserwowanego przedmiotu. Jest to możliwe dzięki takiej budowie oka, która zapewnia skupianie wszystkich promieni świetlnych wpadających do oka w jego ognisku.



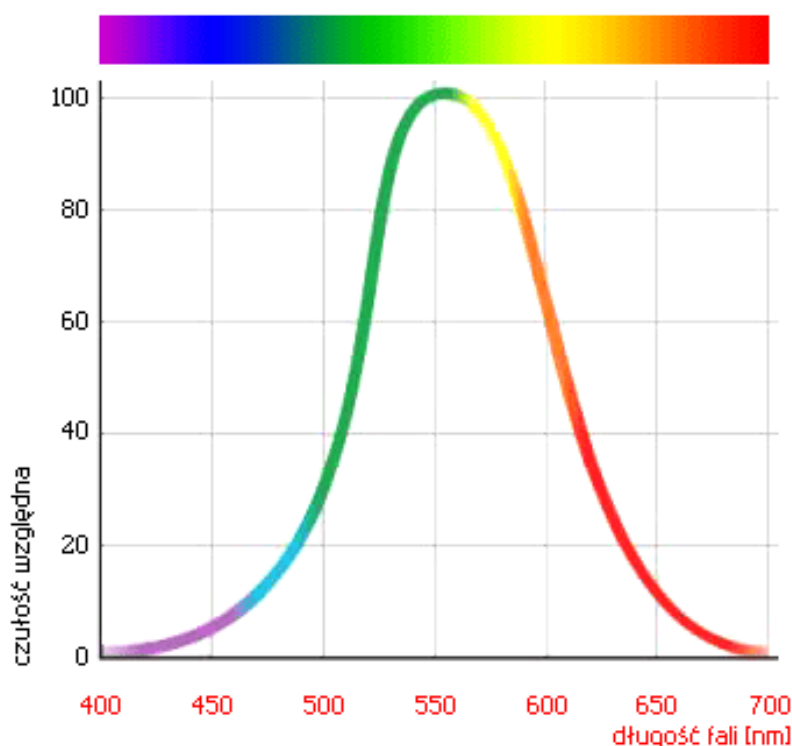
Rys.2 Sposób powstawania obrazu w oku.

Światło wpadające do oka biegnie przez rogówkę, komorę przednią oka, soczewkę i ciało szkliste, by zakończyć swą podróż na siatkówce wywołując wrażenie wzrokowe przekazywane do mózgu za pośrednictwem nerwów łączących się w nerw wzrokowy.



Rys.3 Budowa oka.

Rogówka, wraz z cieczą wodnistą, soczewka i ciałem szklistym, stanowią układ skupiający promienie świetlne tak, by na siatkówce pojawiał się ostry obraz obserwowanego przedmiotu i dawał jak najostrzejsze wrażenie wzrokowe. Dlatego też soczewka ma możliwość zmiany swojego kształtu, a co za tym idzie mocy optycznej. Pozwala to na ogniskowanie na siatkówce przedmiotów znajdujących się w różnych odległościach od oka. Zdolność tę nazywamy akomodacją. Ostre widzenie uzyskiwane jest wtedy, gdy ognisko obrazowe pokrywa się z siatkówką. Oko odbiera tylko część promieniowania na padającego. Związane jest to z własnościami fizyko-chemicznymi rogówki, czopków i pręcików. Odbieramy zatem tylko światło, które mieści się w zakresie tzw. okna optycznego. Okno optyczne to przedział długości fali elektromagnetycznej (światła) od ok. 400nm (co odpowiada światłu o barwie fioletowej) do ok. 700nm (co odpowiada światłu o barwie czerwonej). Powyżej długości 700nm znajduje się niewidoczna dla człowieka podczerwień, a poniżej 400nm, również niewidoczny, ultrafiolet. Do fal elektromagnetycznych zaliczamy także niewidoczne dla człowieka promienie gamma, promienie X i inne.



Rys.4 Czułość ludzkiego oka na fale elektromagnetyczne

Obraz przedmiotu na siatkówce jest odwrócony "do góry nogami", co wynika z fizycznej budowy oka (soczewka odwraca obraz). W pierwszych dniach życia mózg człowieka uczy się widzieć prawidłowy obraz obracając go by w późniejszym życiu robić to automatycznie. Oznacza to, że niemowlę widzi świat "postawiony na głowie" i dopiero po pewnym czasie zaczyna widzieć normalnie.

2. Natężenie światła

Dzięki natężeniu światła będziemy mogli obliczyć kolor obiektów naszego wirtualnego świata. Natężenie składa się z trzech składowych (R (czerwony) G (zielony) B (niebieski)) powinno być dodatnie lub 0. W przypadku wartości ujemnej, zamieniamy jego wartość na 0. Składowe będą przyjmować wartości od 0 do 1 i określać będą kolejne składowe koloru. Prócz samej definicji potrzebne będą metody pozwalające na dokonywanie pewnych operacji między kolejnymi natężeniami jak i skalarom i natężeniami. Podstawowe metody mające znaleźć się w naszej klasie to dodawanie natężeń oraz dzielenie przez skalar. Natężenie światła składać się będzie z trzech składowych koloru (liczby rzeczywiste)

- Konstruktor domyślny tworzy natężenie 0,0,0
- Konstruktor tworzący natężenie przez podanie wszystkich składowych koloru.
- Konstruktor kopiujący
- Metoda zwracająca wartości natężenia dla składowej czerwonej. Wartość pochodzi z przedziału [0,1]. Jeśli wartość przekracza 1, zwrócona zostanie wartość 1.
- Metoda zwracająca wartości natężenia dla składowej zielonej. Wartość pochodzi z przedziału [0,1]. Jeśli wartość przekracza 1, zwrócona zostanie wartość 1.
- Metoda zwracająca wartości natężenia dla składowej niebieskiej. Wartość pochodzi z przedziału [0,1]. Jeśli wartość przekracza 1, zwrócona zostanie wartość 1.
- Metoda która dodaje natężenia do natężenia bieżącego. Natężenie dodawane jest przez wartość każdej składowej koloru (R,G,B).

```
public void add(double r, double g, double b)
{
    red += (r >= 0.0f) ? r : 0.0f;
    green += (g >= 0.0f) ? g : 0.0f;
    blue += (b >= 0.0f) ? b : 0.0f;
}
```

- Metoda dodawania podanego natężenia do bieżącego natężenia.
- Metoda dzielenia bieżącego natężenia przez skalar. Natężenie będzie null jeśli skalar jest 0.

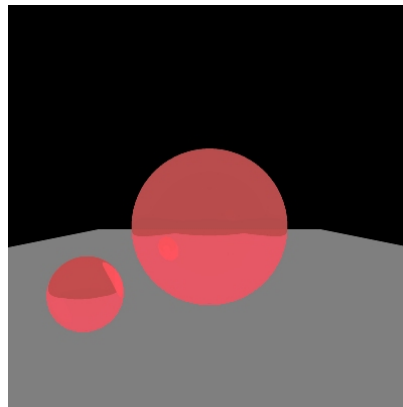
```
if (v != 0.0f)
{
    red /= v;
    green /= v;
    blue /= v;
}
else
{
    red = green = blue = 0.0f;
}
```

3. Źródło światła

Źródło światła tworzymy poprzez umieszczenie go w układzie xyz ,oraz przez podanie jego koloru (natężenia). Wygląd końcowego obrazu będzie zależał nie tylko od położenia źródła, ale też od jego rodzaju. Wektor natężenia ($I = (I_1, I_2, I_3)$) przychodzącego promienia obliczony jest na podstawie rodzaju źródła światła i koloru ($L = (L_1, L_2, L_3)$). Poniżej zostaną opisane podstawowe rodzaje źródeł światła.

3.1 Światło ambient

Nawet jeśli obiekt w scenie nie jest oświetlony bezpośrednio powinien być mimo wszystko widoczny. Możliwe jest to w przypadku metod globalnego oświetlenia gdy bierzemy pod uwagę pośrednie odbicia światła od innych obiektów. Kiedy mamy model oświetlenia bez globalnego oświetlenia najprostszą metodą oświetlenia obiektów nie oświetlonych bezpośrednio jest wprowadzenie źródła światła ambient. Obiekt oświetlony tego rodzaju źródłem będzie posiadał kolor i będzie się wydawał płaski (w każdym punkcie będzie miał ten sam kolor)



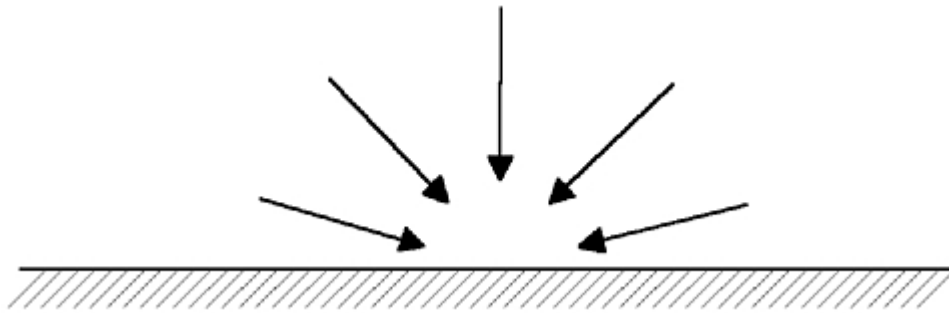
Rys.5 Sfery oświetlone tylko ambient

Światło tego rodzaju nie ma rozmiaru ni położenia. Natężenie takiego oświetlenia jest stałe w każdym punkcie sceny i może mieć zdefiniowany kolor. Tego rodzaju światło niezależny od pozycji obserwatora, źródła i obiektów w scenie W polskojęzycznej literaturze można się spotkać z określeniem światła otaczającego. W fotografii i sztuce nazywane jest także światłem dostępnym lub istniejącym. [6in]. Definicja tego rodzaju oświetlenia będzie następująca:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \quad \text{Równanie 1 światło ambient}$$

Natężenie odbitego światła (I) jest równe kolorowi źródła światła. Tego rodzaju oświetlenie nadchodzi ze wszystkich kierunków tak samo z identycznym natężeniem. Poniższa ilustracja przedstawia światło typu ambientne, nadchodzi ze wszystkich kierunków bez

definicji położenia źródła. Natężenie nadchodzącego światła (I) odpowiada kolorowi światła (L).



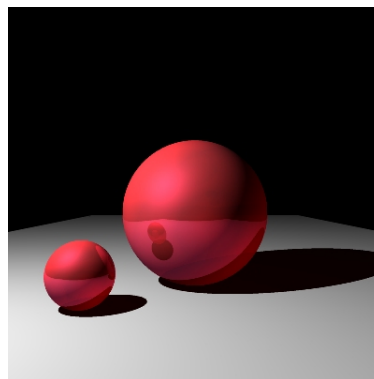
Rys.6 Źródło światła ambient

3.2 Światło kierunkowe

W przypadku źródła kierunkowego, wszystkie promienie są równoległe a natężenie światła nie zależy od odległości. Równanie będzie wyglądało podobnie jak w przypadku oświetlenia ambient:

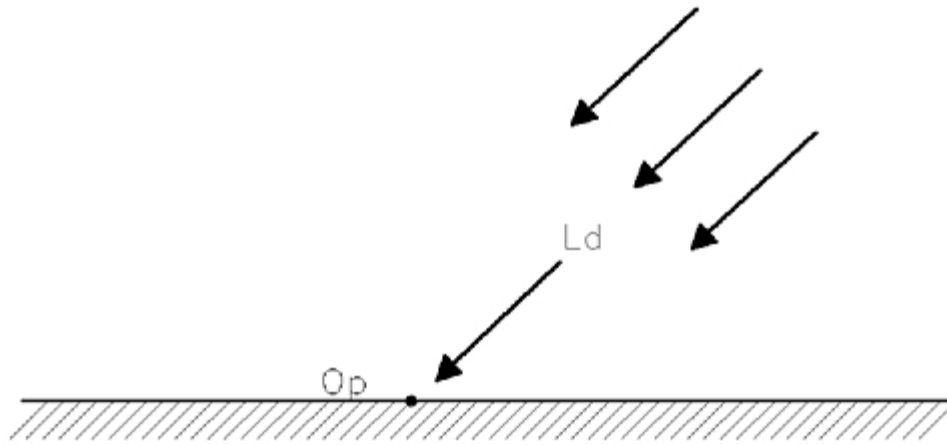
$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \quad \text{Równanie 2 Kierunkowe źródło światła}$$

W przyrodzie tego rodzaju światło występuje w przypadku oświetlenia słonecznego. Światło jest na tyle daleko że nie jest brana pod uwagę odległość źródła a jedynie kierunek z którego nadchodzi.



Rys.7 Sfery oświetlone kierunkowym źródłem światła

Poniższa ilustracja pokazuje w jaki sposób kierunkowe źródło światła oświetla powierzchnię w scenie. Tego rodzaju źródło wysyła równoległe promienie, w przedstawionym przykładzie promienie nadchodzą z prawej strony i docierają do płaszczyzny. Kierunek źródła oznaczona jest jako L_d , punkt oświetlany oznaczony jest jako O_p .



Rys.8 Definicja kierunkowego źródła światła

Natężenie wektora \mathbf{l} światła docierającego do punktu na płaszczyźnie może zostać zaimplementowane w następujący sposób:

```
class DirectionalLight : public Light{
public:
    Point3D direction;

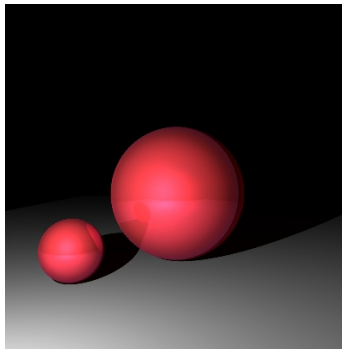
    int read(FILE* fp);
    void write(FILE* fp=stdout);

    Point3D getDiffuse(Point3D cameraPosition, IntersectionInfo
iInfo);
    Point3D getSpecular(Point3D cameraPosition, IntersectionInfo
iInfo);

    int      isInShadow(IntersectionInfo iInfo, Shape* shape);
};
```

3.3 Światło punktowe

Światło punktowe, jak sama nazwa wskazuje, emitowane jest z określonego punktu i rozchodzi się we wszystkich kierunkach z jednakowym natężeniem. Dla tego typu światła, poza kolorem, intensywnością i pozycją, musimy zdefiniować zasięg działania światła i współczynniki, określające sposób zanikania światła, wraz z oddalaniem się od źródła. Jeżeli tego nie zrobimy, to światło w całym swoim zakresie będzie posiadało maksymalną wartość, a oświetlone nim obiekty będą posiadały tylko powierzchnie białe, widoczne z pozycji światła, lub zupełnie ciemne. Światła punktowe są znacznie bardziej kosztowne pod względem obliczeniowym od światła kierunkowych, czy otaczających, gdyż w trakcie obliczeń natężenie światła musi być obliczone dla wszystkich wierzchołków danego obiektu. Tego typu światła znakomicie poprawiają realizm rysowanych scen i mogą odwzorowywać obiekty takie jak choćby żarówka.



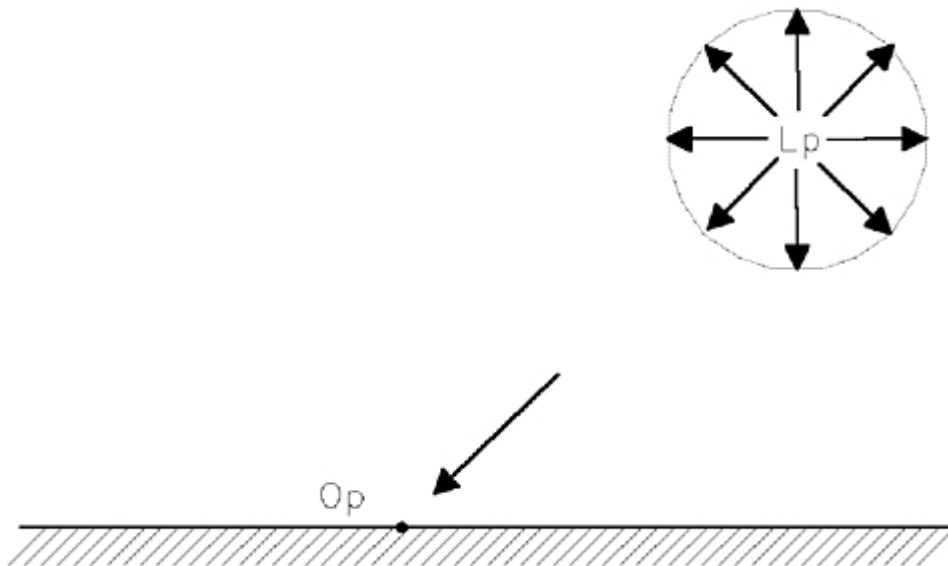
Rys.9 Sfery oświetlone punktowym źródłem światła

W przypadku punkowego źródła, możemy wprowadzić zanikanie natężenia w zależności od odległości punktu od położenia źródła. Oświetlenie obliczane jest wtedy na podstawie współczynników zanikania, oznaczonych w poniższym wzorze jako (a_1 oraz a_2). Dodatkowo należy wprowadzić odległość oświetlanego punktu (O_p) od położenia źródła (L_p).

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \frac{1}{a_1 + a_2 |O_p - L_p|}$$

Równanie 3 punktowe źródło światła

Sposób oświetlania punku O_p przez punktowe źródło światła, znajdującego się w punkcie L_p ilustruje poniższy rysunek:



Rys.10 Punktowe źródło światła

Definicja punktowego źródła światła może wyglądać w następujący sposób:

```
class PointLight : public Light{
public:
    Point3D location;
    Flt constAtten;
    Flt linearAtten;
    Flt quadAtten;

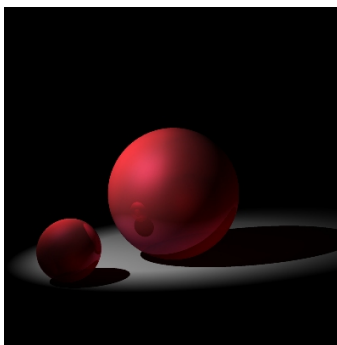
    int read(FILE* fp);
    void write(FILE* fp=stdout);

    Point3D getDiffuse(Point3D cameraPosition,IntersectionInfo
iInfo);
    Point3D getSpecular(Point3D cameraPosition,IntersectionInfo
iInfo);

    int      isInShadow(IntersctionInfo iInfo,Shape* shape);
};
```

3.4 Światło reflektorowe

Dla tego typu światła, poza kolorem, intensywnością, pozycją oraz kierunkiem w przestrzeni, musimy określić kąt, między krawędziami tworzącymi stożek oświetlenia. Istnieje też możliwość zdefiniowania dwóch stożków – wewnętrzny i zewnętrzny. Musimy wtedy wprowadzić parametr, określający sposób zanikania światła między granicą stożka wewnętrznego, a stożka zewnętrznego. Światło rzucające przez tego typu reflektor składa się z dwóch stożków: wewnętrzny jest jaśniejszy, emitujący właściwe światło i zewnętrzny, zwykle ciemniejszy, określający obszar, w którym światło zanika.



Rys.11 Sfery oświetlone reflektorem.

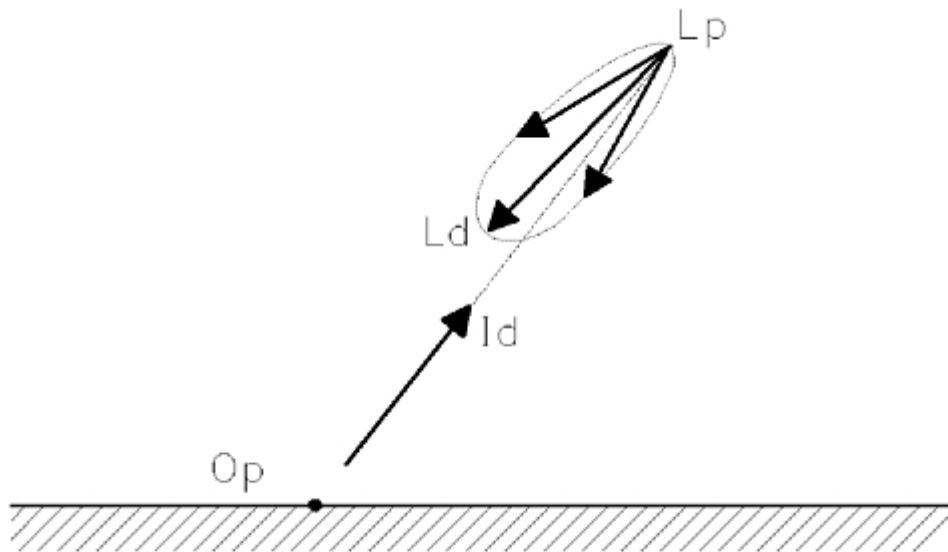
Definicja tego typu źródła światła podobna będzie do światła punktowego. Wprowadzony zostanie parametr zanikania (a_1 and a_2), parametr odpowiedzialny za koncentrację natężenia światła (c_a), odległość między (O_p) a źródłem (L_p), kierunek świecenia źródła (L_d), istotny i charakterystyczny dla tego rodzaju źródła parametr to kąt rozwarcia (θ) rozchodzących się promieni światła. Jest to kąt pomiędzy wektorem kierunkowym źródła

światła a krawędzią stożka światła, definiuje promień stożka światła. Dla oświetlanego punktu równanie będzie następujące:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \frac{\left| \frac{L_d \cdot (O_p - L_p)}{|O_p - L_p|} \right|^{C_e}}{a_1 + a_2 |O_p - L_p|}$$

Równanie 4 Reflektorowe źródło światła

Poniższy rysunek ilustruje oświetlanie punktu O_p przez reflektorowe źródło światła.



Rys.12 Reflektorowe źródło światła

Definicja reflektorowego źródła światła może wyglądać w następujący sposób:

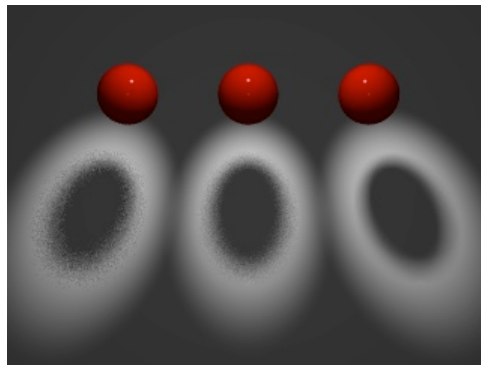
```
class SpotLight : public Light{
public:
    Point3D location;
    Point3D direction;

    Flt constAtten;
    Flt linearAtten;
    Flt quadAtten;
    Flt cutOffAngle;
    Flt dropOffRate;
    int read(FILE* fp);
    void write(FILE* fp=stdout);

    Point3D getDiffuse(Point3D cameraPosition,IntersectionInfo
iInfo);
    Point3D getSpecular(Point3D cameraPosition,IntersectionInfo
iInfo);
    int isInShadow(IntesectionInfo iInfo,Shape* shape);
};
```

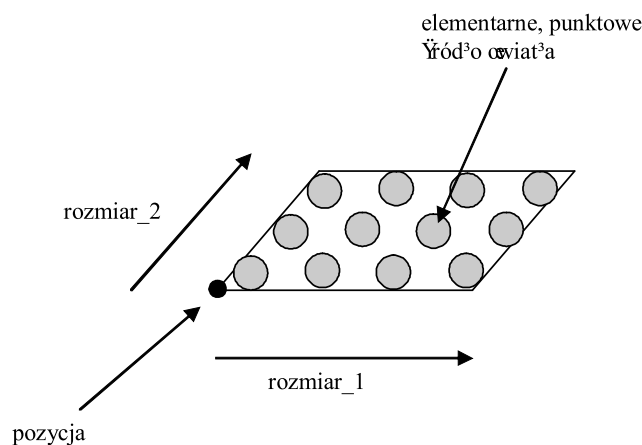
3.5 Światło powierzchniowe

Powierzchniowe źródło światła posiada kształt wielokąta, świecącego całą swoją powierzchnią. Światło wysyłane jest we wszystkich kierunkach. Powierzchniowe źródło światła pozwala na uzyskanie tzw. miękkich (rozmytych) cieni za obiektami. Można je zdefiniować za pomocą prostokątnej matrycy punktowych źródeł światła. Użytkownik definiuje położenie prostokątnej matrycy w przestrzeni, jej rozmiar oraz rozdzielczość - liczbę elementarnych, punktowych źródeł światła.



Rys.13 Sfery oświetlone powierzchniowym źródłem światła.

Zwróćmy uwagę na to, że rozmiar powierzchniowego źródła światła definiowany jest przez wektory - w szczególnym przypadku takie źródło światła może nie być prostokątem.



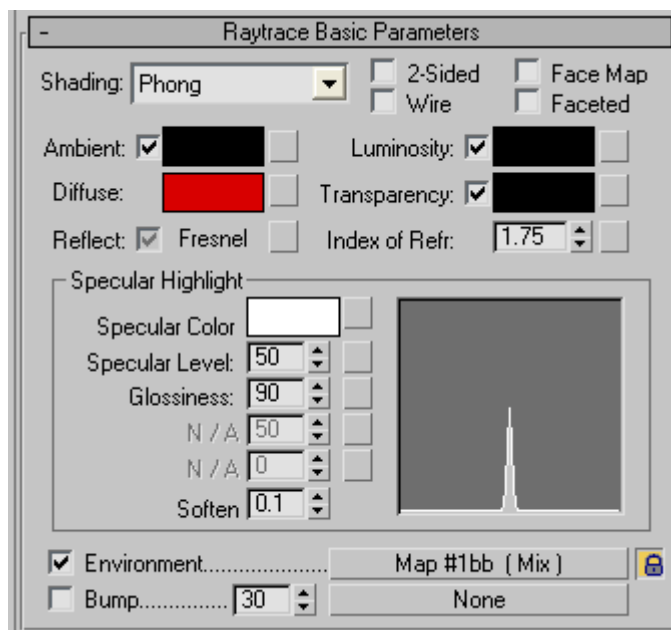
Rys.14 Powierzchniowe źródło światła

Rozdzielczość powierzchniowego źródła światła określa się zwykle eksperymentalnie - zbyt duża liczba elementarnych źródeł światła znacząco wydłuża czas obliczeń. Przeciętne powierzchniowe źródło światła składa się z kilkunastu elementarnych źródeł punktowych.

4. Definicja materiału

„Nadzieja, być może, stanowi ten sam materiał, z którego uczyniona jest nasza dusza.²”

Wspomniana funkcja BRDF definiowała zachowanie świetlnego promienia po interakcji z płaszczyzną. Zachowanie światła będzie zależało od materiału z którego jest zbudowany dany obiekt.



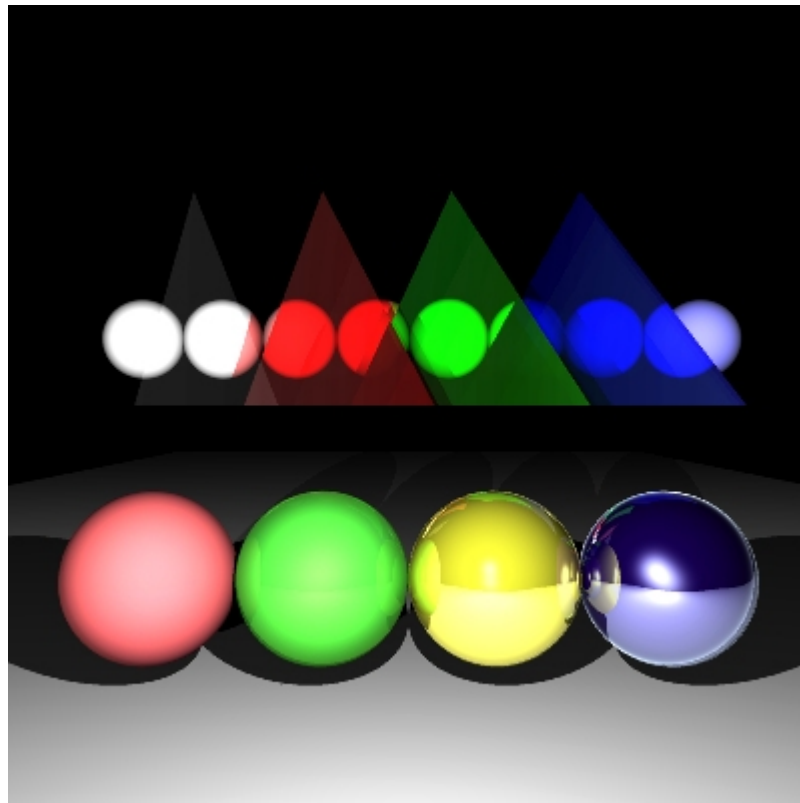
Rys.15 Definiowanie materiału w programie 3dmax

Do każdego obiektu przypisany jest materiał, na który składają się współczynniki określające zachowanie światła po zetknięciu się z jego powierzchnią. Można wyróżnić cztery wspomniane współczynniki:

- K_a –współczynnik odbicia ambient, (niezależny od pozycji światła ani obserwatora)
- K_s –współczynnik odbicia specular (zależny od pozycji światła oraz obserwatora)
- K_t –współczynnik odbicia transmission (niezależny od pozycji światła ani obserwatora)
- K_d –współczynnik odbicia diffuse (niezależny od pozycji obserwatora)
- n –współczynnik gładkości powierzchni

Wszystkie współczynniki zostaną omówione w rozdziale o równaniu oświetlenia. Służą one do procentowego ustalenia wpływu składowych na wynikowe natężenie w badanym punkcie.

² Gabriel Marcel



Rys.16 Przegląd możliwych materiałów

Poniżej znajduje się przykładowa definicja materiału:

```
class Material
{
public:

    Material(const Colour& diffuse_colour, float
specular_amount, float specular_coeff, float reflect_fraction);

    ~Material(){}

    Colour diffuse_colour;

    float specular_amount; //[0, infinity]. 0 = matt
    float specular_coeff; //(0, infinity). for phong lighting
a.k.a shineness.

    float reflect_fraction; //[0,1]. 1 == total reflection, i.e.
a perfect mirror

};
```

Można też zdefiniować materiał za pomocą tablic przyjmujących jako elementy, odpowiednie składowe:

```

class Material
{
    private double[] kAmbient = new double[3];
    private double[] kDiffuse = new double[3];
    private double[] kSpecular = new double[3];
    private double alpha;
    private Texture texture;
    private bool hasTexture;

    public Material()
    {
        for (int i = 0; i < 3; i++)
        {
            kAmbient[i] = 0.3f;
            kDiffuse[i] = 0.5f;
            kSpecular[i] = 0.8f;
        }

        this.alpha = 100;
        this.hasTexture = false;
    }
}

```

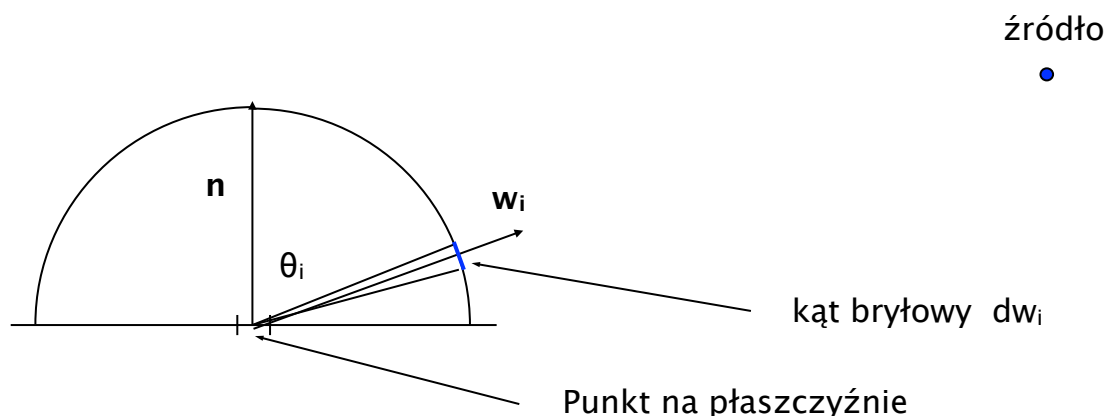
5. Model oświetlenia

5.1 Funkcja BRDF

Założmy że mamy dany kierunek nadejścia światła w_i , oraz kierunek odbicia w_o , które są zdefiniowane dla relatywnie małego elementu powierzchni. BRDF jest zdefiniowana jako stosunek ilości odbitego światła w kierunku w_o , do ilości światła które dotarło do płaszczyzny z kierunku w_i . Ilość energii odbitej zdefiniujemy jako L_o a nadchodzącej E_i . BRDF określimy więc następującym równaniem:

$$BRDF = \frac{L_o}{E_i}$$

Wartość światła z kierunku w_i , jest proporcjonalne do wartości światła przychodzącego z kąta bryłowego. Zakładamy że źródło światła na rysunku ma natężenie L_i . Gdy kąt bryłowy jest odpowiednio mały, możemy rozważać płaski obszar kąta na sferze. Rezultatem będzie obszar oświetlony przez tą samą intensywność światła L_i z każdego miejsca kąta bryłowego. Oznacza to będzie że punkt oświetlony jest przez światło z obszaru $L_i \cdot dw_i$. Wartość ta zależna jest od kąta lecz nie od położenia punktu na płaszczyźnie.



Rys.17 Oświetlenie punktu na płaszczyźnie przez punktowe źródło światła

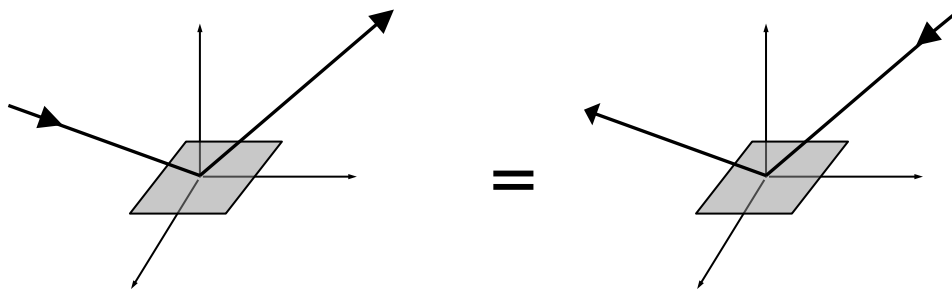
By punkt ten był uwzględnionym, powinniśmy rozłożyć lub rzutować światło na dany punkt. Tego rodzaju operacja podobna jest do odbicia lambertowskiego dla powierzchni matowych, gdzie wartość światła odbitego mnożyliśmy przez $\cos\theta_i = N \cdot w_i$ dostaniemy więc $E_i = L_i \cos\theta_i dw_i$. Końcowe równanie funkcji BRDF przybierze następującą postać:

$$BRDF = \frac{L_o}{L_i \cos\theta_i dw_i}.$$

Wychodząc z tej definicji, możemy zaobserwować dwie interesujące rzeczy. Pierwsza to fakt że BRDF nie jest ograniczone obszarem $[0,1]$. Jednakże natężenia L_o do L_i muszą być w przedziale $[0,1]$. Dzieląc przez cosinus wartości BRDF będą wartościami większymi od 1. Drugie spostrzeżenie to fakt że BRDF nie jest funkcją o wartości skalarnej (bez jednostki) BRDF zawiera dzielenie przez kąt bryłowy wynika z tego że jej jednostką będzie odwrotność stereoradianów (sr^{-1}).

Istnieją dwa rodzaje funkcji BRDF i dwie podstawowe własności tej funkcji. BRDF możemy podzielić funkcje na izotropowe i anizotropowe. Dwie podstawowe własności to zachowanie energii i zasada wzajemności. Izotropowość użyte jest do opisanie BRDF reprezentuje właściwości odbić które się nie zmieniają w zależności od rotacji powierzchni wokół wektora normalnego. Anizotropia natomiast jest zjawiskiem polegającym na zależności właściwości fizycznych od charakterystycznych kierunków materiału. Pojęcie anizotropii optycznej jest najczęściej kojarzone z przechodzeniem światła przez kryształy i zjawiskiem dwójłomności. Spowodowane to jest zależnością współczynnika załamania światła od kierunku rozchodzenia się fali względem głównego przekroju kryształu. Wyróżnia się anizotropię naturalną (wykazuje ją większość kryształów) i anizotropię wymuszoną, spowodowaną takimi czynnikami zewnętrznymi jak działanie pól elektrycznych (zjawisko Kerra) i magnetycznych (zjawisko Cottona-Moutona) lub odkształcenia mechaniczne (ściskanie lub rozciąganie w zadanym kierunku). Z anizotropią mamy także do czynienia w przypadku odbicia promieniowania od powierzchni materiału. Wiele powierzchni, zarówno naturalnych, jak i uzyskanych w wyniku technologicznej obróbki odbija światło w sposób anizotropowy, zależny od kierunku jego padania – w

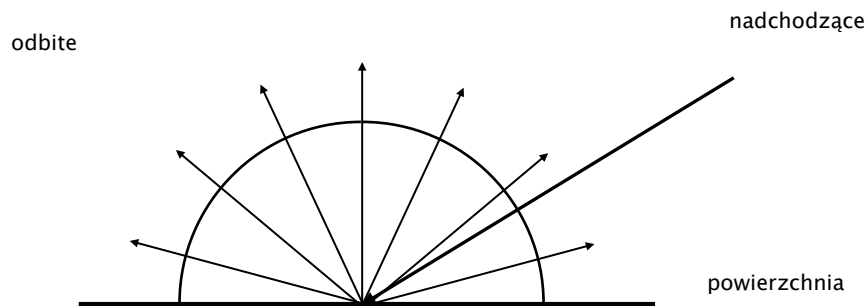
sposób zależny od usytuowania powierzchni względem źródła światła. Dobrym przykładem powierzchni wykazującej naturalne właściwości anizotropowe jest powierzchnia drewna. Odbija ona światło zależnie od kąta między kierunkiem padania a kierunkiem słoików przekroju. Powierzchnia metalu poddana obróbce mechanicznej (np. polerowaniu) będzie odbijała światło zależnie od kąta między padającym promieniem, a kierunkiem obróbki. Funkcja BRDF bazuje na prawach fizyki dlatego też spełnia dwa prawa które są jej podstawowymi własnościami. Pierwsza to wzajemność nazywana też symetrycznością która oznacza, że zamiana obserwatora i źródła światła nie spowoduje zmiany opisu zjawiska. Drugą własnością jest zasada zachowania energii, zgodnie z którą suma całkowitej energii wypromieniowanej na skutek odbicia światła od powierzchni jest nie większa niż energia światła padającego. Poniższy rysunek ilustruje zasadę symetryczności:



Rys.18 Symetryczność funkcji BRDF

Zachowanie energii możemy zilustrować w następujący sposób:

$$BRDF_{\lambda}(\theta_i, \phi_i, \theta_o, \phi_o) = BRDF_{\lambda}(\theta_o, \phi_o, \theta_i, \phi_i).$$



Rys.19 Funkcja BRDF dla powierzchni matowej

Wartość odbitego światła musi być mniejsza bądź równa wartości padającego światła. Suma wszystkich wychodzących kierunków BRDF wymnożonych przez kąt bryłowy musi być mniejsza od 1. Zapisując tą własność za pomocą równania:

$$\sum_{out} BRDF_{\lambda}(\theta_i, \phi_i, \theta_o, \phi_o) \cos \theta_o dw_o \leq 1$$

Rozważając tę własność pod kątem ciągłej półsfery wszystkich odbitych promieni. Suma oznaczona zostanie całką po sferze i równanie przybierze postać:

$$\int_{\Omega} \text{BRDF}_{\lambda}(\theta_i, \phi_i, \theta_o, \phi_o) \cos \theta_o d\omega_o \leq 1.$$

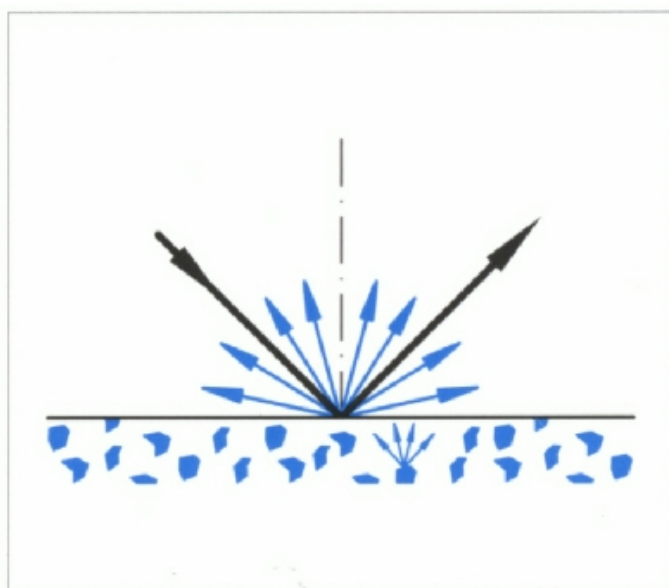
Symbol \int_{Ω} oznacza całkę po wszystkich kierunkach półsfery (proszę nie wpadać w panikę).

5.2 Równanie oświetlenia Lamberta

W przypadku używania oświetlenia stałego w każdym punkcie sceny, powstały obraz jest płaski i nie odwzorowuje trójwymiarowości obiektu. Najprostszym sposobem odwzorowania oświetlenia jest zamodelowanie go dla powierzchni matowych, w tym celu używa się znane z optyki prawo Lamberta. Zostało sformułowane przez Johanna Heinricha Lamberta w 1760 r i określa zależność światłości od kierunku obserwacji:

Jeżeli pewna powierzchnia promieniuje lub rozprasza światło, to światłość I w kierunku tworzącym kąt θ z normalną do powierzchni promieniującej.

W grafice komputerowej odbicie lambertowskie używane jest do symulowania odbicia światła od powierzchni matowych (diffuse). Odbicie takie zakłada że światło będzie rozpraszane we wszystkich kierunkach tak samo, nie będzie więc zależęć od położenia obserwatora.



Rys.20 Odbicie lambertowskie

Ruch obserwatora nie wpływa na sposób oświetlenia obiektów zbudowanych z materiałów lambertowskich. Równanie Lamberta możemy zapisać:

$$I_D = k_d \times I_L \times |L \cdot N| \cos \alpha$$

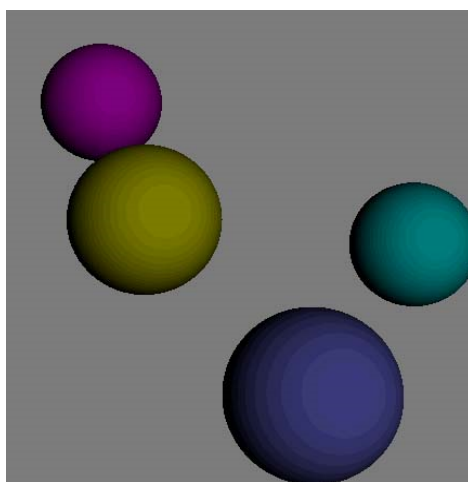
Korzystając z następującej własności:

$$L \cdot N = |L| |N| \cos \alpha$$

Kąt α jest kątem między normalną a wektorem „światlnym”. Rozbłysk będzie obliczany na podstawie iloczynu skalarnego normalnej do powierzchni i znormalizowanego wektora do światła. Wektor do światła otrzymuje się przez stworzenie wektora od oświetlanego punktu do źródła światła. Otrzymany skalar mnożony jest przez współczynnik materiału oraz natężenie światła:

$$I_D = L \cdot N \times k_d \times I_L$$

I_D jest natężeniem światła odbitego od matowej powierzchni (refleksem światła). k_d jest współczynnikiem który odpowiada ilości odbitego światła od takiej powierzchni (kolorowi powierzchni). I_L jest natężeniem nadchodzącego światła. Z własności funkcji cosinus wynika bezpośrednio pewna własność którą można zaobserwować przy oświetlaniu powierzchni przez źródło światła. Jeśli źródło znajduje się bezpośrednio nad obiektem natężenie rozbłysku jest maksymalne, gdyż $\cos = 1$ gdy pozycja będzie zmierzać do kąta 90 stopni, natężenie będzie malało do zera, gdyż $\cos(\pi/2) = 0$.



Rys.21 Sfery wykonane z materiału lambertowskiego

Odbicie lambertowskie jest najprostszym rodzajem odbicia, rysunek ilustruje sfery pocieniowane tą metodą. Poza metodą Lamberta istnieje wiele innych metod cieniowania matowych powierzchni stosowanych w silnikach renderujących, np.: Orena-Nayara, metoda Toon itp.

Przykładowy kod obliczający oświetlenie w badanym punkcie:

```
shade = dot_product( light_vector, normal_vector )
if ( shade < 0 )
shade = 0
point_color = object_color * ( ambient_coefficient +
diffuse_coefficient * shade )
```

5.3 Powierzchnie lustrzane

W przypadku obiektów zbudowanych z materiału matowego, światło rozchodzi się we wszystkich kierunkach tak samo, zostaje rozproszone. Dla obiektów lustrzanych, światło zostanie odbite w konkretnym i określonym kierunku. Jeśli odbite światło trafi bezpośrednio do obserwatora następuje rozbłysk (ang. highlight). Obliczając rozkład oświetlenia dla takiego obiektu musimy brać pod uwagę pozycje obserwatora. Ilość światła docierającego do obserwatora jest proporcjonalna do kąta pomiędzy odbitym światłem a nadchodzącym promieniem (celowo nie sprecyzowałem punktu nadejścia promienia, gdyż może on nadejść od obserwatora w przypadku promienia pierwotnego lub od innego obiektu w przypadku promienia wtórnego). W przypadku tego rodzaju odbicia kąt ten jest taki sam jak kąt pomiędzy nadchodzącym światłem a odbitym promieniem.

Odbicie lustrzane zależy od pozycji obserwatora i będzie się zmieniało w zależności od jego położenia. Należy tu podkreślić że odbicie zwierciadlane/lustrzane nie definiuje powierzchni mającej właściwości odbijania innych przedmiotów jak lustro, bardziej definiuje kierunkowe odbicie światła a co za tym idzie powstanie charakterystycznego rozbłysku w postaci plamy świetlnej. By otrzymać powierzchnie symulującą zachowanie lustra należało by przeprowadzić rekursję promienia w celu znalezienia natężenia od odbijanego obiektu. Podobnie jak powierzchnia matowa, materiał zwierciadlany posiada wiele implementacji, najbardziej rozpowszechnioną jest metoda Phonga. W celu obliczenia natężenia na powierzchni lustrzanej należy obliczyć wektor odbity oraz wektor skierowany do obserwatora:

$$I_{out} = I_{in} (k_s \cos^n \Theta)$$

Podobnie jak miało to miejsce w równaniu Lamberta funkcja trygonometryczna zostanie zastąpiona przez iloczyn skalarny:

$$I_{out} = I_{in} (k_s (R \cdot V)^n)$$

Równanie dla obliczenia kierunku promienia odbitego R ma postać:

$$R = I - 2 \times (N \cdot I) \cdot N$$

R jest wektorem kierunkowym promienia odbitego, I jest kierunkiem nadejścia promienia, oraz N jest normalna w punkcie przecięcia. Mając kierunek odbitego promienia powinniśmy mieć także punkt jego pochodzenia. Jest to punkt przecięcia obiektu z nadchodzącym promieniem. Funkcja rozbłysku \cos^n została wprowadzona przez Phonga po analizie doświadczalnej odbicia światła od zwierciadlanych powierzchni. Kontroluje ona

rozmiar jasnej plamy, która pokazuje się na powierzchniach. Model Phong'a jest modelem empirycznym i jego użycie nie jest planowane dla renderingów fizycznych: jego zadaniem jest wizualne wrażenie zwierciadłaności niektórych obiektów przez dodanie rozbłysków na ich powierzchni.

Implementacja dla powierzchni zwierciadlanej:

```
double r, g, b, cosinus;
I = ray.direction().normalizujVector();
N = s.normal(intersection);
R = I - (N * N.iloczynSkalarny(I) * 2.0f);
ss = ray.direction().normalizujVector().iloczynSkalarny(R);
if (-ss > 0)
{
    specular = Math.Pow(ss, a);
}
else
{
    specular = 0;
}
specular *= specularCoef;
sIntensity = source.Intensity * specular;
```

5.4 Równanie oświetlenia Phong'a

Model oświetlenia Phong'a może być rozważany jako specjalny przypadek BRDF. Przypomnę tutaj równanie oświetlenia Phong'a :

$$I_{out} = I_{in} \left(k_d (L \cdot N) + k_s (R \cdot V)^n \right)$$

gdzie L wektorem w kierunku źródła światła, V w kierunku obserwatora, N jest normalną do powierzchni, a R kierunkiem odbicia. k_d oraz k_s odpowiednio współczynnikami odbicia lustrzanego i rozproszonego współczynnik n rozbłyskiem odbicia lustrzanego. Często model oświetlenia Phong'a zawiera wartość światła ambient, które symuluje światło otoczenia (globalnego). Jako że rozważamy na razie tylko oświetlenie bezpośrednie model oświetlenia Phong'a zapiszemy jako:

$$L_o = L_i \left(k_d (L \cdot N) + k_s (R \cdot V)^n \right)$$

gdzie L_o jest natężeniem odbitym do obserwatora a L_i natężeniem ze źródła światła. Zapisując to równanie z użyciem funkcji BRDF otrzymamy:

$$L_o = L_i \text{Refl}(w_i, w_o)$$

$$L_o = \text{Refl}(w_i, w_o) L_i$$

$$L_o = \frac{\cos\theta_i dw_i}{\cos\theta_i dw_i} \text{Refl}(w_i, w_o) L_i$$

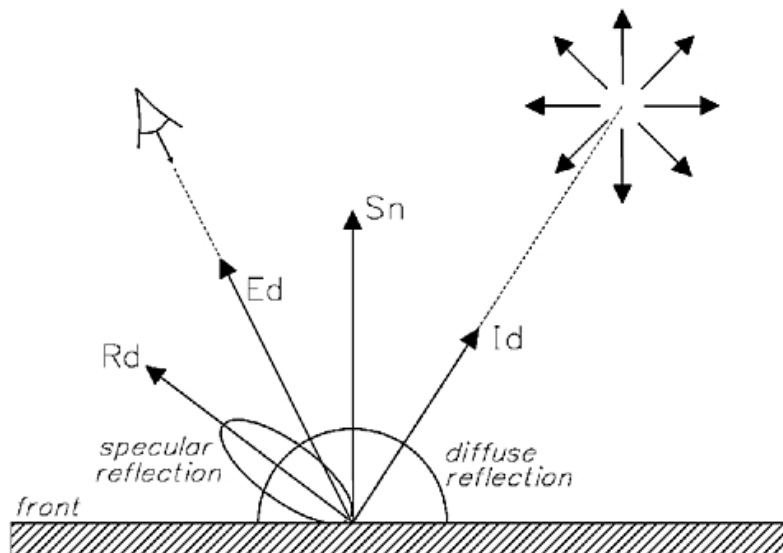
$$L_o = \frac{\text{Refl}(w_i, w_o)}{\cos\theta_i dw_i} L_i \cos\theta_i dw_i$$

Zauważmy że równanie to przypomina końcowe równanie oświetlenia BRDF. Model oświetlenia Phong może być rozważany jako specjalny przypadek BRDF, wykorzystywany przy materiałach z właściwościami odbijania aproksymowanymi przez:

$$\text{BRDF}(\theta_i, \phi_i, \theta_o, \phi_o) = \frac{\text{Refl}(w_i, w_o)}{\cos\theta_i dw_i}$$

$$\text{BRDF}(\theta_i, \phi_i, \theta_o, \phi_o) = \frac{k_d (\mathbf{w}_i \cdot \mathbf{N}) + k_s (\mathbf{R} \cdot \mathbf{w}_o)^n}{\cos\theta_i dw_i}$$

Fizycy zauważyli, że model Phong w ogólnym przypadku pasuje do wyników badań empirycznych. Pojawia się jednak kilka odstępstw od tej reguły. Największą różnicę widać w kształcie funkcji rozbłysku, której postać nie zmienia się wraz ze zmianą kąta padania światła. Okazuje się, że w rzeczywistości, im bardziej rośnie kat padania tym zdolność odbijania zwierciadlanego powierzchni rośnie. Inna zaobserwowana różnica jest fakt, że kierunek maksymalnego rozbłysku nie zawsze jest zgodny z wektorem zwierciadlanego odbicia. W literaturze można wskazać wiele prac porównujących, przede wszystkim pod względem obliczeniowym, różne podejścia do opisu zjawiska odbicia. Znane funkcje BRDF można podzielić na dwie grupy : zależności opracowane eksperymentalnie i zależności mające podłoże fizyczne. Pierwszą grupę stanowią zależności, których opis matematyczny został eksperymentalnie dobrany do oczekiwanych (lub zmierzonych) efektów. Nie mają one żadnego uzasadnienia teoretycznego, ale są dobrą aproksymacją rzeczywistych zjawisk.



Drugą grupę stanowią opracowania, które powstały na podstawie odpowiedniej teorii fizycznej opisującej gładkość (chropowatość) powierzchni. Obie jednak grupy są pewnym przybliżeniem rzeczywistości. Natomiast ze względów praktycznych nie zawsze jest celowe korzystanie z funkcji BRDF powstałych na podstawie pomiarów rzeczywistej powierzchni, gdyż jest to bardzo kosztowne obliczeniowo. Nawet w takich przypadkach stosuje się pewne aproksymacje. Warto więc zastanowić się nad wyborem odpowiedniej funkcji BRDF. Szczególnie jest to istotne w sytuacji prowadzenia obliczeń z wykorzystaniem gotowych pakietów numerycznych gdzie wybór dostępnych parametrów może być ograniczony.

Prosta implementacja równania Phong'a:

```
for (int i = 0; i < img.Width; i++)
{
    for (int j = 0; j < img.Height; j++)
    {
        srodekX = -1.8f + (i + 0.5f) * widthPixel;
        srodekY = 1.0f - (j + 0.5f) * heightPixel;
        Ray ray = new Ray(new Point(0, 0, 1), new Point(srodekX,
srodekY, 0));

        Point intersetion = s.intersect(ray);
        if (intersetion != null)
        {
            //Vector n = new Vector(s.getCenter(), intersetion);
            //Vector l = new Vector(intersetion, source.getPosition());
            double r, g, b, cosinus;
            I = ray.direction().normalizujVector();
            N = s.normal(intersetion);
            R = I - (N * N.iloczynSkalarny(I) * 2.0f);

            ss = ray.direction().normalizujVector().iloczynSkalarny(R);
            if (-ss > 0)
            {
                specular = Math.Pow(ss, a);
            }
            else
            {
                specular = 0;
            }
            specular *= specularCoef;
            sIntensity = source.Intensity * specular;
            // diffuse
            cosinus = ray.direction().normalizujVector
().iloczynSkalarny(s.normal(intersetion));
            r = -source.Intensity.getRed() * k * cosinus;
            g = -source.Intensity.getGreen() * k * cosinus;
            b = -source.Intensity.getBlue() * k * cosinus;
```

```

        Intensity diffuseIntensity = new Intensity(255, g,
b) + aIntensity;
        img.setPixel(i, j, sIntensity + diffuseIntensity);
    }
    else img.setPixel(i, j, aIntensity);
}
}

```

6. Zadanie

Do tworzonej aplikacji należy dodać klasę odpowiedzialną za materiał (rozszerzyć definicję prymitywu o referencję do materiału).

Zdefiniować klasę abstrakcyjną źródła światła i jej konkretyzację w postaci źródła punktowego i powierzchniowego (dla chętnych).

Zmodyfikować algorytm raytracingu (w momencie wykrycia przecięcia z najbliższym obiektem, zamiast ustawiać domyślny kolor piksela, należy poprowadzić z tego punktu promień w kierunku źródła światła i sprawdzić czy przecina inne obiekty - w zależności od wyniku obliczany jest kolor piksela na podstawie modelu Phong).

7. Bibliografia

- [1] **Ashdown Ian, P. Eng., LC, FIES.** Photometry and Radiometry A Tour Guide for Computer Graphics Enthusiasts. : John Wiley & Sons in 1994
- [2] **Woźniak Władysław Artur.** Radiometria i Fotometria. : Instytut Fizyki Politechniki Wrocławskiej 2004
- [3] **Wynn Chris.** An Introduction to BRDF-Based Lighting : NVIDIA Corporation 2000
- [4] **Rusinkiewicz Szymon.** A Survey of BRDF Representation for Computer Graphics : CS348c, Winter 1997
- [5] **Glassner, Andrew S.** Space Subdivision for Fast Ray Tracing : IEEE Computer Graphics & Applications, March 1988, volume 8, number 2, pp. 60–70
- [6] **Marlon John.** Focus On Photon Mapping : Premier Press 2003 ISBN 1-1-59200-008-8
- [7] **Shirley Peter, R. Keith Morley.** Realistic Ray Tracing: Second Edition : AK Peters; 2nd edition (July 2003) ISBN-13: 978-1568811987

- [8] **Whitted Turner.** An improved illumination model for shaded display : Communications of the ACM archive. Volume 23 , Issue 6 (June 1980)
- [9] **Appel Arthur.** The notion of quantitative invisibility and the machine rendering of solids : Proceedings of ACM National Conference 1967
- [10] **Sunday Dan.** Intersections of Rays, Segments, Planes and Triangles in 3D : softSurfer 2006
- [11] **Maciej Falski** Przegląd modeli oświetlenia w grafice komputerowej: Praca magisterska Uniwersytet Wrocławski Wydział Matematyki i Informatyki Instytut Informatyki 2004
- [12] **Christophe Schlick.** An inexpensive BRDF model for physically-based rendering : Computer Graphics Forum 1994

- [1in] **Dr inż. Władysław Artur Woźniak.** Strona informacyjna Instytut Fizyki . : <http://www.if.pwr.wroc.pl/~wozniak/>
- [2in] **The Australian National University Faculty of Engineering and Information Technology (FEIT)** . Global Illumination Models Physically Based Illumination, Ray Tracing and Radiosity : <http://escience.anu.edu.au/lecture/cg/GlobalIllumination/printNotes.en.html>
- [3in] **Wikipedia.** Wolna encyklopedia internetowa . : <http://pl.wikipedia.org/>
- [4in] **Softsurfer.** List of Algorithm Titles in the softSurfer Archive. . : http://softsurfer.com/algorithm_archive.htm
- [5in] **Henrik Wann Jensen.** Strona informacyjna Henrika Wann Jensena. . : <http://www.gk.dtu.dk/~hwj>
- [6in] **Jiajun Zhu** CS 645 Computer Graphics <http://www.cs.virginia.edu/~jz8p/>
- [7in] **Webster's New Millennium™ Dictionary of English**, Preview Edition (v 0.9.7)

Copyright © 2003–2009 Dictionary.com, LLC <http://dictionary.reference.com>