

Introduzione alla Generazione di Immagini Fotorealistiche

*Corso di Dottorato in Matematica e Informatica
Università degli Studi della Basilicata*

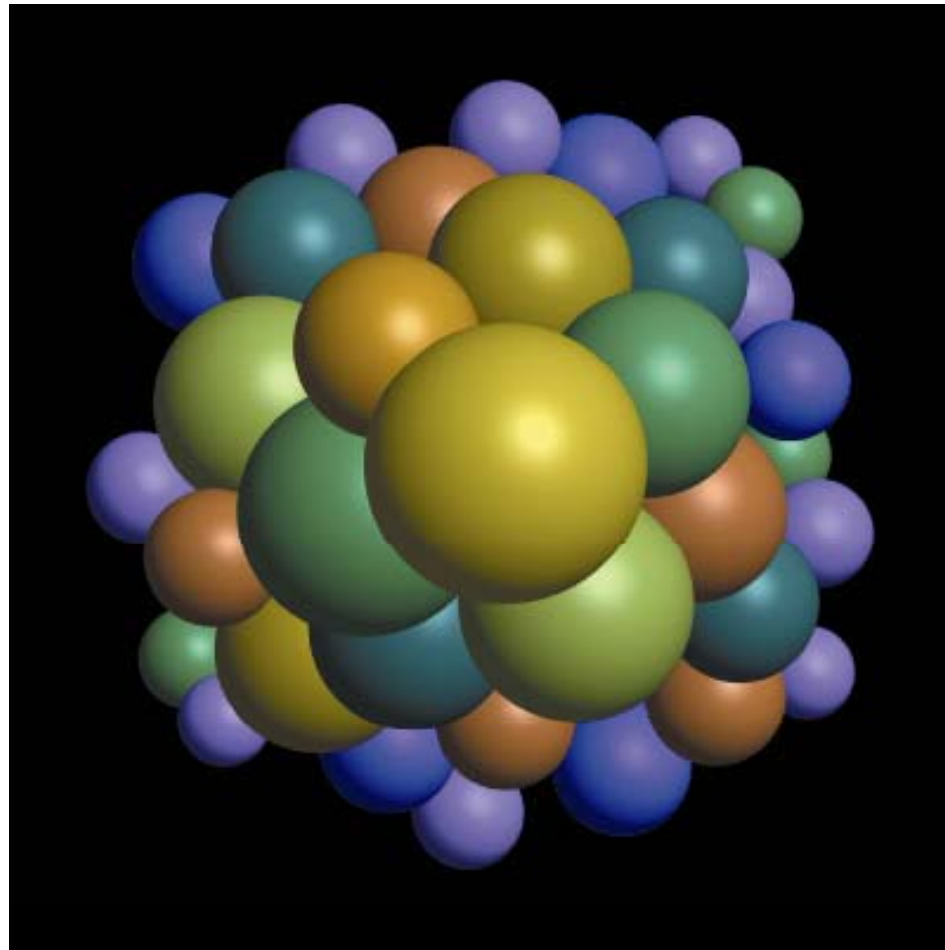
Dott. Ugo Erra

10° Lezione – Riflessione speculare

Sommario

- Riflessione speculare
 - Il modello completo di Phong
 - Definizione del materiale con specularità
-

Cosa vogliamo ottenere?

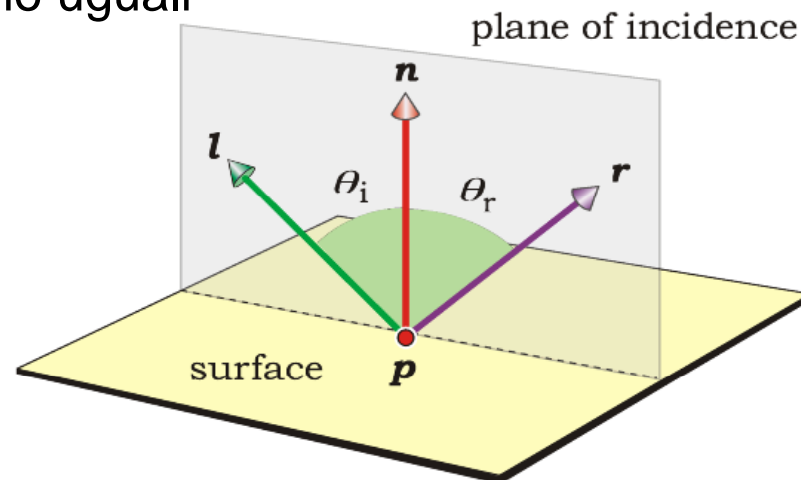


Introduzione

- La riflessione speculare permette ad un materiale di riflettere parte della luce
 - La riflessione avviene principalmente nella direzione del raggio riflesso
 - Gli oggetti che riflettono al luce in questo modo presentano un zona di massima luce visibile in una particolare direzione
 - La riflessione speculare insieme alla riflessione ambientale e diffusa rappresenta la terza componente del modelli di illuminazione diretta di Phong
-

Il piano di incidenza

- Consideriamo un punto p su di una superficie ed i seguenti vettori
 - l direzione del raggio incidente nel punto p
 - n normale della superficie nel punto p
 - r direzione del raggio riflesso nel punto p
- I vettori r ed l insieme alla normale n appartengono al *piano di incidenza*
 - In base alla *legge della riflessione* l'angolo incidente θ_i e l'angolo di riflessione θ_r sono uguali

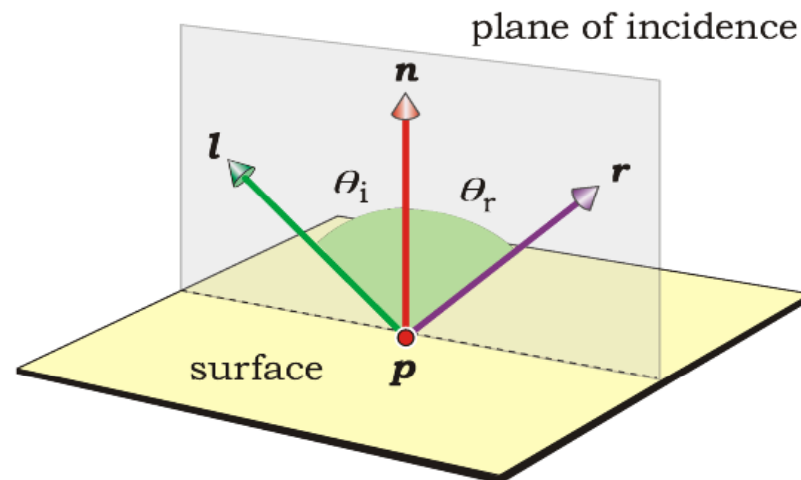


Calcolo del vettore r

- Per ottenere il vettore r basta notare che i vettori l, r ed n sono coplanari e di conseguenza r è una combinazione lineare di l ed n

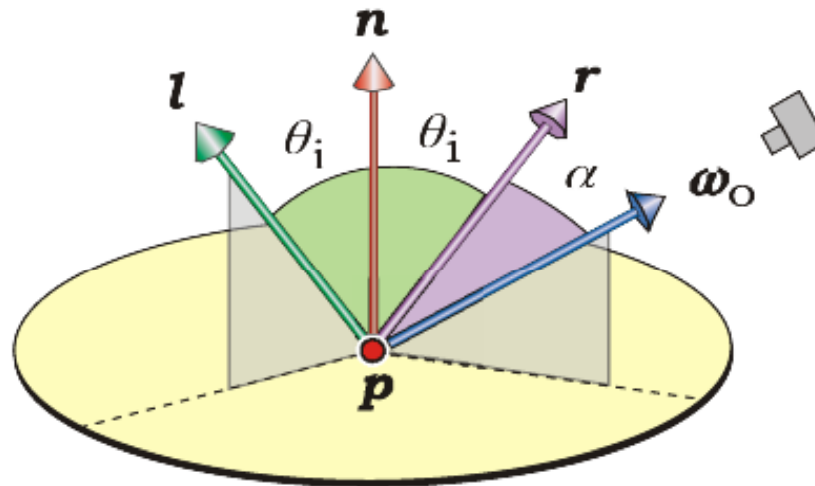
$$r = -l + 2(n \cdot l)n$$

- Il vettore r sarà per costruzione unitario



Il modello di riflessione speculare

- Supponendo che un ipotetico osservatore si trovi nella direzione ω_o
- La riflessione speculare ha un picco lungo il vettore r e decresce in all'aumentare dell'angolo α tra r e ω_o



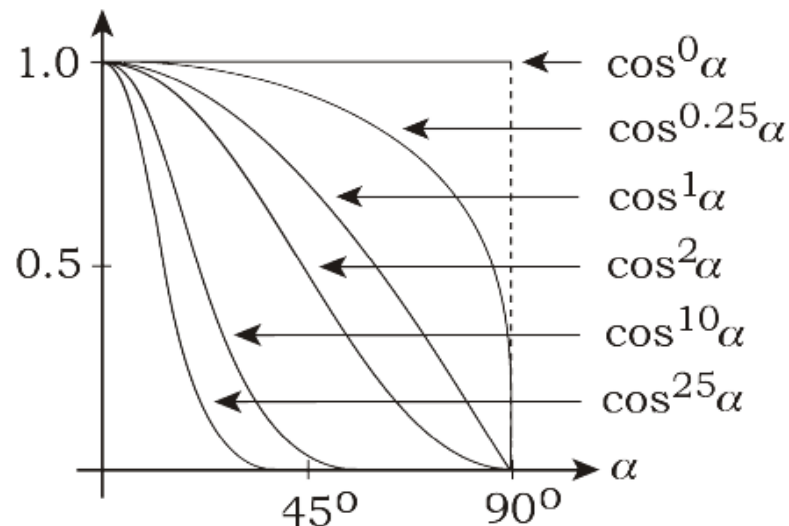
Il modello di Phong

- Il lavoro di Phong modella la riflessione speculare in funzione del valore α con l'espressione

$$(\cos \alpha)^e - (\mathbf{r} \cdot \boldsymbol{\omega}_o)^e$$

dove e è l'esponente di Phong ed è maggiore o uguale a 0

- Intuitivamente il valore di e permette di modellare la dimensione del bagliore percepito dall'osservatore

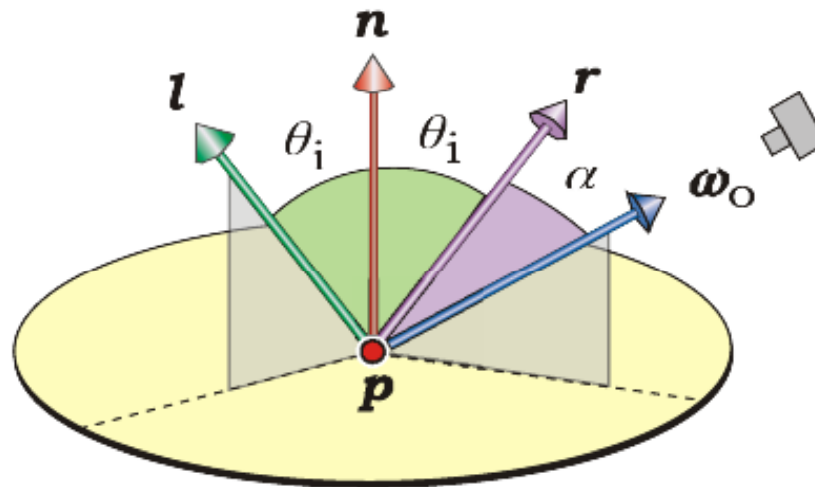


Il modello di Phong

- Utilizzando l'espressione di Phong, la radianza riflessa da una singola sorgente luminosa è

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = k_s (\mathbf{r} \cdot \boldsymbol{\omega}_o)^e l_s c_l$$

dove k_s è un coefficiente di riflessione speculare



Una BRDF per superfici glossy

- Possiamo definire una BRDF per superfici speculari glossy come

$$f_{r,s}(\mathbf{l}, \boldsymbol{\omega}_o) = k_s (\mathbf{r} \cdot \boldsymbol{\omega}_o)^e$$

- Questa definizione rispetta anche la proprietà di reciprocità della BRDF, ovvero

$$f_{r,s}(\mathbf{l}, \boldsymbol{\omega}_o) = f_{r,s}(\boldsymbol{\omega}_o, \mathbf{l})$$

- La radianza riflessa sarà quindi

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = k_s (\mathbf{r} \cdot \boldsymbol{\omega}_o)^e l_s c_l (\mathbf{n} \cdot \mathbf{l})$$

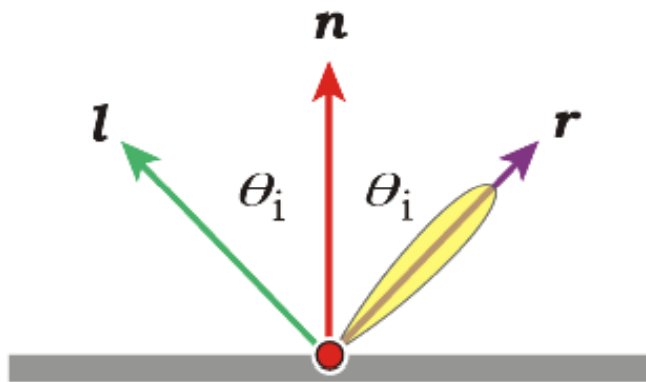
Ambient+Diffuse+Specular

- Utilizzando la linearità della BRDF possiamo ricavare una espressione della radianza riflessa che tenga conto della componente ambientale, diffusa e riflessa

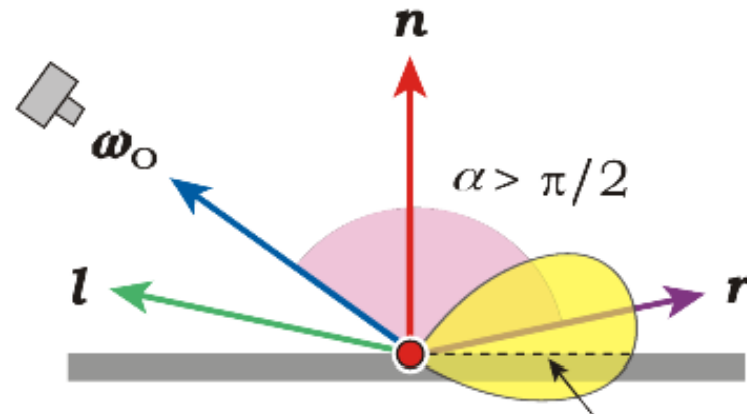
$$L_o(p, \omega_o) = k_a \mathbf{c}_d * (l_s \mathbf{c}_l) + \sum_{j=1}^n (k_d \mathbf{c}_d / \pi) (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j) + \sum_{j=1}^n k_s (r_j \cdot \omega_o)^e (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j)$$

- $k_a \in [0,1]$ coefficiente di riflessione ambientale del materiale
 - $k_d \in [0,1]$ coefficiente di riflessione diffusa del materiale
 - $k_s \in [0,1]$ coefficiente di riflessione speculare del materiale
 - \mathbf{c}_d è il colore diffuso del materiale
 - \mathbf{c}_l è il colore della luce
 - $l_s \in [0, \infty)$ è un fattore di scala della luce
-

Esempi

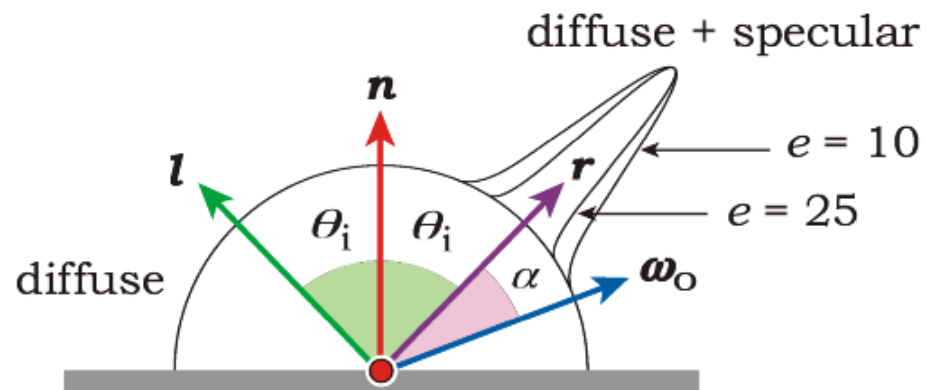


$e=20$

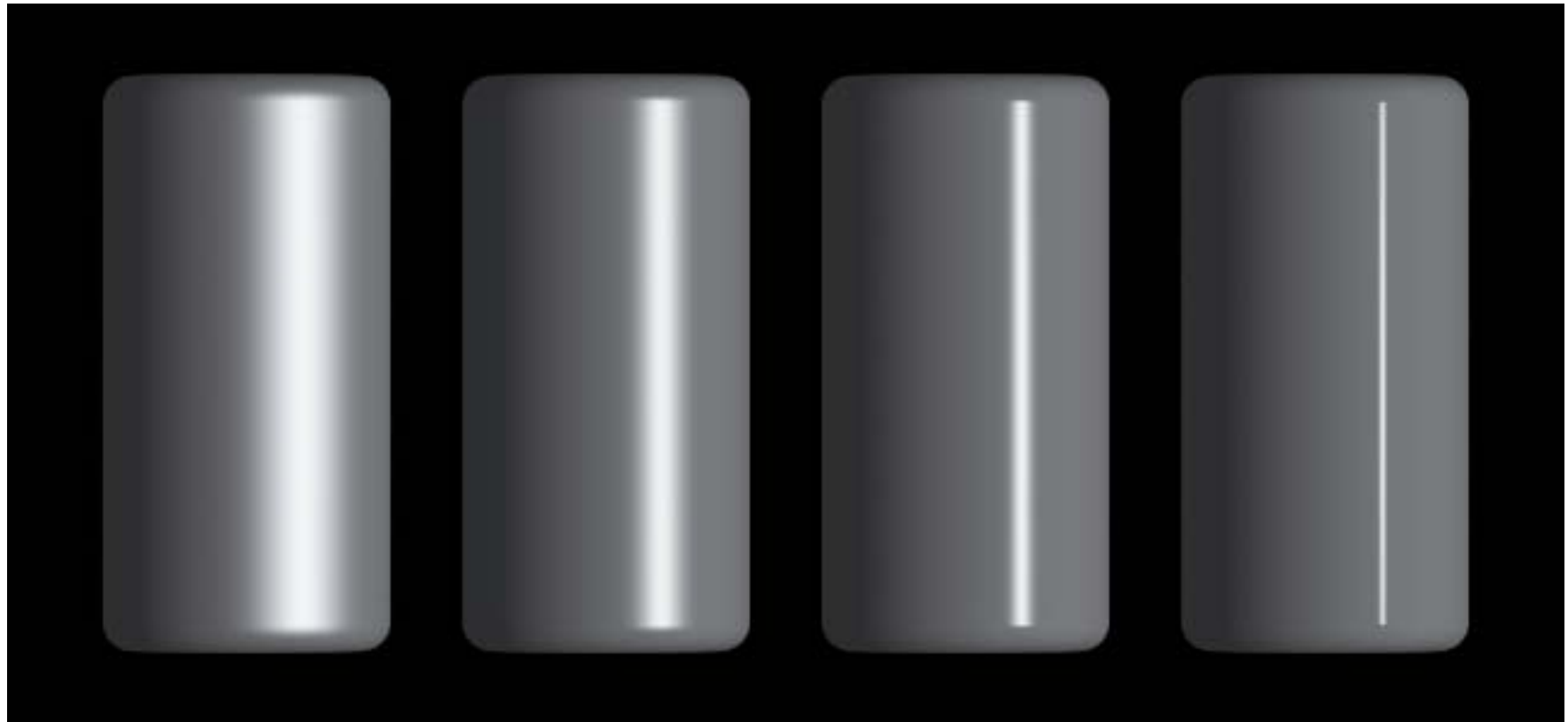


lobe penetrates surface

$e=2$



Esempi



$$k_a = 0.25, k_d = 0.6, k_s = 0.2$$
$$e = 5, 20, 100, 1000$$

Colore speculare

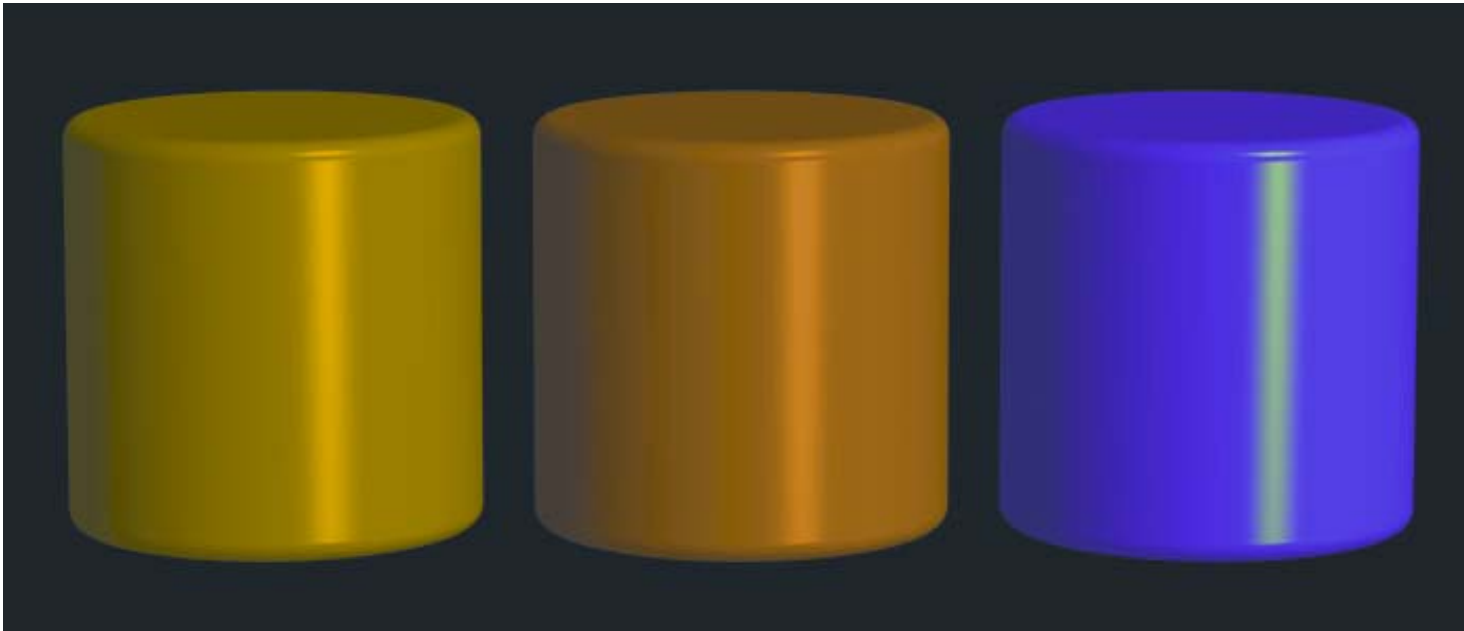
- In natura la specularità si manifesta come un colore ottenuto miscelando il colore diffuso del materiale ed il colore della luce

$$L_o(p, \omega_o) = k_a \mathbf{c}_d * (l_s \mathbf{c}_l) + \sum_{j=1}^n (k_d \mathbf{c}_d / \pi) (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j) + \sum_{j=1}^n k_s (r_j \cdot \omega_o)^e (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j)$$

- Introducendo un termine *colore speculare* \mathbf{c}_s possiamo parametrizzare il colore speculare

$$L_o(p, \omega_o) = k_a \mathbf{c}_d * (l_s \mathbf{c}_l) + \sum_{j=1}^n (k_d \mathbf{c}_d / \pi) (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j) + \sum_{j=1}^n k_s (r_j \cdot \omega_o)^e \mathbf{c}_s (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j)$$

Colore speculare



La sottoclasse GlossySpecular

```
class GlossySpecular: public BRDF {
public:

    GlossySpecular(void);

    virtual RGBColor f(const ShadeRec& sr, const Vector3D& wo, const Vector3D& wi) const;

    virtual RGBColor sample_f(const ShadeRec& sr, const Vector3D& wo, Vector3D& wi, float&
pdf) const;

    virtual RGBColor rho(const ShadeRec& sr, const Vector3D& wo) const;

    void set_ks(const float ks);
    void set_exp(const float exp);
    void set_cs(const RGBColor& c);

    void set_normal(const Normal& n);

private:
    float          ks;
    RGBColor       cs;          // specular color
    float          exp;        // specular exponent
    Sampler*       sampler;    // for use in sample_f
};
```

La sottoclasse GlossySpecular

```
// no sampling here: just use the Phong formula
// this is used for direct illumination only

RGBColor GlossySpecular::f(const ShadeRec& sr, const Vector3D& wo, const Vector3D&
wi) const {
    RGBColor      L;
    float         ndotwi = sr.normal * wi;
    Vector3D      r(-wi + 2.0 * sr.normal * ndotwi); // mirror reflection direction

    float         rdotwo = r * wo;

    if (rdotwo > 0.0)
        L = ks * cs * pow(rdotwo, exp);

    return (L);
}
```

$$f_{r,s}(\mathbf{l}, \boldsymbol{\omega}_o) = k_s (\mathbf{r} \cdot \boldsymbol{\omega}_o)^e c_s$$

La classe Phong

```
class Phong: public Material {  
  
    public:  
  
        Phong(void);  
  
        virtual RGBColor shade(ShadeRec& s);  
  
    protected:  
  
        Lambertian* ambient_brdf;  
        Lambertian* diffuse_brdf;  
        GlossySpecular* specular_brdf;  
};
```

Il metodo Phong::shade

```
RGBColor Phong::shade(ShadeRec& sr) {  
  
    Vector3D wo = -sr.ray.d;  
    RGBColor L = ambient_brdf->rho(sr, wo) * sr.w.ambient_ptr->L(sr);  
    int num_lights = sr.w.lights.size();  
  
    for (int j = 0; j < num_lights; j++) {  
        Vector3D wi = sr.w.lights[j]->get_direction(sr);  
        float ndotwi = sr.normal * wi;  
  
        if (ndotwi > 0.0)  
            L += (diffuse_brdf->f(sr, wo, wi) +  
                specular_brdf->f(sr, wo, wi)) * sr.w.lights[j]->L(sr) * ndotwi;  
    }  
  
    return (L);  
}
```

$$L_o(p, \omega_o) = k_a \mathbf{c}_d * (l_s \mathbf{c}_l) + \sum_{j=1}^n \left((k_d \mathbf{c}_d / \pi) + k_s (r_j \cdot \omega_o)^e \right) (l_{s,j} \mathbf{c}_{l,j}) (n \cdot l_j)$$

Shading

- La componente diffuse e speculare differiscono dalla indipendenza e dalla dipendenza dall'osservatore
- Lo shading è definito anche come *viewer-independent* e *viewer-dependent*

