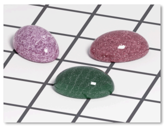
Rendering with Layered Materials

Andrea Weidlich and Alexander Wilkie

Outline

- Introduction
- Layered Surfaces in Computer Graphics
- Combining Individual BRDFs into Layered Models
- Classifying Materials - Using Layered BRDFs to Describe Object Appearance
- Modelling with Layered Surfaces



Introduction

- Efficient and intuitive appearance modelling not entirely solved yet
- Higher degree of control over object appearance is desirable
- Should still look convincing

Alexander Wilkie

LAYERED SURFACES IN COMPUTER GRAPHICS

Where do we get our data from?

- Two approaches exist
  - Explicit storage of tabulated measurements or simulation results
  - Approximation through analytical functions
    - Empirical models
    - Physically based models

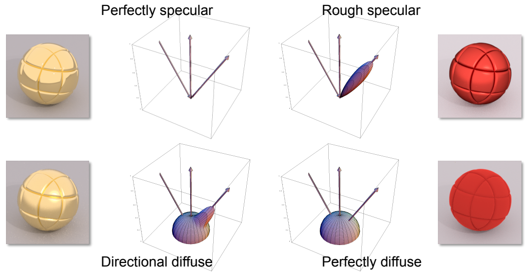
Requirements for Analytical BRDFs

- Reciprocity
  - Sampling directions can be interchanged
  - Due to Helmholtz reciprocity principle – a fundamental law of physics
- Energy conservation
- Fast evaluation
- Expressivity
- (Easy stochastic sampling for MC rendering)

Analytical BRDFs

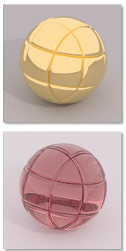
- Empirical models
  - Lambert, Phong, Blinn, Lafortune, Ward
  - Superposition of different components
- Physically based models
  - Torrance-Sparrow, Cook-Torrance, Kajiya, He-Sillion-Torrance-Greenberg (HTSG)
  - Physical material constants needed

Reflection Types



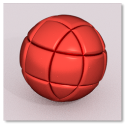
Traditional Reflectance Models – Perfectly Specular

- Do not exist in reality
- Only one outgoing directions
- Incoming angle equals outgoing angle
- Often used to simulate smooth glass / metallic surfaces
- For realistic materials: Frensel coefficients



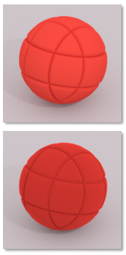
Traditional Reflectance Models – Rough Specular

- Reflect light not only in the ideal direction
- „Highlight“
- Some of the light is reflected slightly off from the ideal specular angle.
- E.g. Phong: Size of the highlight can be changed with exponent



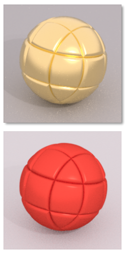
Traditional Reflectance Models – Perfectly Diffuse

- Reflect the incoming light equally in all directions over the hemisphere
- Viewing direction independent.
- E.g.
  - Lambert
  - Oren-Nayar
  - …



Traditional Reflectance Models – Directional Diffuse

- Combination of a rough specular reflector and an ideal diffuse reflector
- Eg.
  - Cook-Torrance
  - Ward
  - He
  - ...

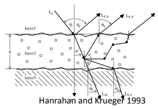


Layered Surface Models

- Many objects consist of layers (e.g. paint, patinas, …)
- No always obvious (e.g. skin)
- Layered surface models offer great potential for creating very convincing renderings
- Sometimes very complex

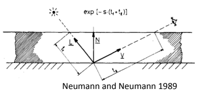
Early Layered Models

- Hanrahan and Krueger (1993)
  - Subsurface scattering in layered surfaces
- Kubelka-Munk Theory (Haase and Meyer 1994)
  - Pigments
- Dorsey and Hanrahan (1996)
  - Metallic Patinas



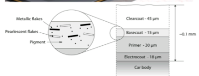
Lacquer

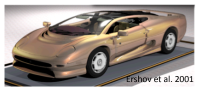
- Neumann and Neumann (1989)
  - Only perfectly specular lacquered objects
- Keleman and Szirmay-Karlos (2001)
  - Simplified Cook Torrance
  - Ignores absorption
- Both use perfectly diffuse base



Pearlescent and Metallic Paint

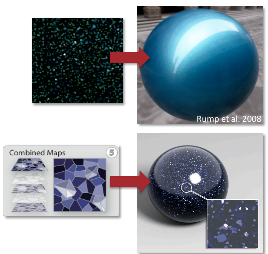
- Ershov et al. (2001, 2004) and Durikovic et al. (2007, 2002)
- Statistical model
- Substrate: Lambert reflector
- Flakes are modelled with a distribution
- Top: clear coat (Fresnel reflectance)
- Interference effect





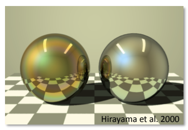
Explicitly Modelled Flakes

- Bigger flakes can be modelled explicitly
- E.g. BTFs (Rump et al. 2008) or Voronoi textures (Weidlich and Wilkie 2008)



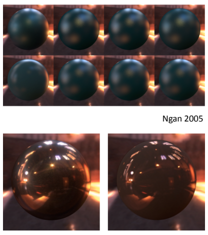
Mulitlayer Films

- Interference effects
- Icart and Arques (1999, 2000)
- Hirayama et al. (2000, 2001)
  - Smooth and rough surfaces
- Granier and Heidrich (2003)
  - RGB Model
  - Includes interference



Comparison

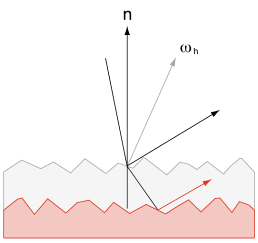
- Layered surfaces can simulate "problematic" materials
  - Interference, SSS, Dispersion, Multilayer Finish
- Do we still need simpler BRDFs?
- Yes!
- Many surfaces can be reasonable simulated with simple Cook-Torrance (Ngan 2005)



Andrea Weidlich

COMBINING INDIVIDUAL BRDFS INTO LAYERED MODELS

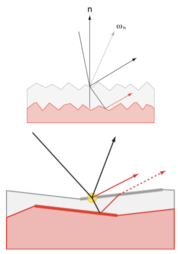
Multi-Layer Reflectance Models



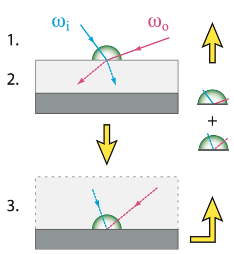
- Individual surfaces are used as components of more sophisticated BRDFs
- Computing of entire BRDF would involve sub-surface scattering
- Simplifications needed

Simplification of the Problem

- Three simplifications
  - Micro-facet is large compared to the thickness of the layer
  - Ray leaves through the same micro-facet that it entered
  - Rays meet at a single point on the next interface
- Still physically plausible



Overview of the Model - BRDF Evaluation



- BRDF of the topmost level is evaluated
- Remaining part of the energy enters the material
- Rays meet at a single point on the next layer; process is repeated
- Light is partly absorbed and attenuated on returning

Mathematical Formulation

$$f_r = f_{r_1}(\theta_i, \theta_r) + T_{12} \cdot f_{r_2}(\theta_{i'}, \theta_{r'}) \cdot a \cdot t$$

attenuated on return

absorbed

next layer

BRDF of topmost level is evaluated

remaining energy

- BRDF of the topmost level is evaluated
- Remaining part of the energy enters the material
- Rays meet at a single point on the next layer; process is repeated
- Light is partly absorbed and attenuated on returning

Evaluation of Topmost Level

- Always dielectric Torrance-Sparrow microfacet surface
- Light that hits an interface in the layer stack is partly reflected, and partly refracted
  - Amount of energy determined through Fresnel terms
  - Appropriate sampling direction is generated for the reflective component
- Done like on a traditional dielectric TS surface
- Remaining part attenuated by Fresnel transmission coefficient T_{12} for air/material

Eurographics 25

Absorption

- Refracted part is assumed to enter the material
  - A part will be absorbed by the varnish material
  - Rest interacts with the next surface in the stack
- Absorption defined by Beer's law

$$I = I_0 e^{-\alpha l}$$

- Length of the path determined by thickness of layer and incident and outgoing angle

$$l = d \cdot \left(\frac{1}{\cos\theta_{i'}} + \frac{1}{\cos\theta_{r'}}\right)$$

Eurographics 26

Example: Increasing Absorption

Eurographics A. Weidlich and A. Wilkie 27

Total Internal Reflections

- All light reflected from lower layers is possibly subjected to total internal reflection
- Exact computation would require explicit simulation
- Approximation: Compensated by "scattering term"
  - Energy that is blocked by geometric factor G enters material
  - Multiplied with Fresnel transmission coefficient for material/air

$$t = (1 - G) + T_{21} \cdot G$$

Eurographics 28

BDRFs in MC Image Synthesis

- Apart from perfectly diffuse surfaces and perfect mirrors, reflection properties are basically only tractable through MC rendering
- Local AND global illumination model needed
  - Path propagation - **global model**
  - BRDF evaluation - **local model**
- Sampling needed
- Simplifications necessary for RTR or production rendering

Eurographics A. Weidlich and A. Wilkie 29

MC Algorithms: Global Evaluation

- Cast outgoing ray for a given incoming direction
- Sampling
  - Each layer is sampled individually
  - Top according to microfacet distribution of topmost layer
  - Incoming ray is refracted
  - Appropriate sampling direction is generated for the reflective component
- Only one ray is followed

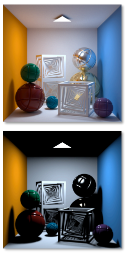
Eurographics A. Weidlich and A. Wilkie 30

MC Algorithms: Local Evaluation

- Compute the entire BRDF for arbitrary given input AND output directions
- Necessary for e.g. bidirectional path tracing, but also in shader language
- PDF needed to weight multiple samples
  - PDF of each component is evaluated
  - PDFs of the individual BRDFs are weighted and added
  - Weight depends on reflection properties

Shader-based Language

- Calculates the appearance of an object in a scene under a set of light sources
- Only interested in single point
  - Local illumination model needed
  - Global illumination is ignored
  - No rays have to be traced, no sampling needed



Example: RenderMan SL

- C-based language
- Can be used in any RenderMan-compilant renderer (e.g. 3Delight, Pixie, …)
- Five different shaders, surface shader describe appearance of a surface
- Result defined by
  - Colours of the surface and the light source(s)
  - Position / orientation of the surface relative to the light
  - Roughness of the surface

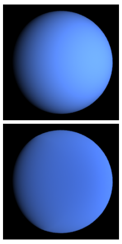
Implementing the Model in RSL

- Two components
  - Oren-Nayar surface
  - Torrance-Sparrow surface
- Can be found in many renderers under different names
- Use as many existing components as possible, e.g. refract(), reflect(), …

Component 1: Oren-Nayar

- Microfacet-based diffuse surface
- Perfectly diffuse for sigma = 0
- More and more retro-reflective for increasing sigma

$$f_r(\omega_i,\omega_o) = \frac{\rho}{\pi}(A + B\max(0,\cos(\phi_i-\phi_o))\sin\alpha\tan\beta)$$
$$A = 1 - \frac{\sigma^2}{2(\sigma^2+0.33)}$$
$$B = \frac{0.45\sigma^2}{\sigma^2+0.09}$$
$$\alpha = \max(\theta_i,\theta_o)$$
$$\beta = \min(\theta_i,\theta_o)$$



sigma = 1.0

Oren-Nayar Shader Code (shader from Larry Gritz)

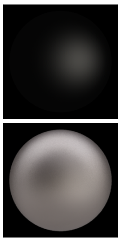
```
Eye = -normalize (I);
theta_r = acos (Eye . Nf);
sigma2 = sigma*sigma;
LN = normalize(L);
cos_theta_i = LN . Nf;
cos_phi_diff = normalize( Eye-Nf*(Eye.Nf) ) . normalize( LN - Nf*(LN.Nf) );
theta_i = acos ( cos_theta_i );
alpha = max ( theta_i, theta_r );
beta = min ( theta_i, theta_r );
C1 = 1 - 0.5 * sigma2 / ( sigma2 + 0.33 );
C2 = 0.45 * sigma2 / ( sigma2 + 0.09 );

if (cos_phi_diff >= 0) C2 *= sin(alpha );
else C2 *= (sin(alpha) - pow(2*beta/PI,3));

C3 = 0.125 * sigma2 / ( sigma2 + 0.09) * pow (( 4*alpha*beta ) / ( PI*PI ), 2 );
L1 = Cs * (cos_theta_i * (C1 + cos_phi_diff * C2 * tan(beta)
        + ( 1 - abs(cos_phi_diff) ) * C3 * tan( ( alpha + beta ) / 2 ) ));
L2 = ( Cs * Cs ) * (0.17 * cos_theta_i * sigma2 / (sigma2+0.13) * (1 - cos_phi_diff * (4 * beta * beta )/ (PI * PI)));
return (L1 + L2) * Cl;
```

Component 2: Torrance-Sparrow

- Small facets (microfacets) that are perfect mirrors
- Appearance given by
  - Distribution function (Blinn)
  - Fresnel term for each microfacet (refractive index)
  - Geometric attenuation

$$f_r = \frac{FDG}{4 \cdot (N \cdot L)(N \cdot V)}$$

- Limited to metals and transparent dielectrics



beta = 20.0

Torrance-Sparrow Shader Code

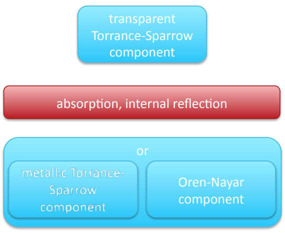
```
color torranceSparrowReflector(
        normal Nf; vector V; vector Ln;  vector
H; color eta, kappa; float m;)
{
   extern vector I;
   color cook = 0, idotn;

   float D = blinnD(Nf, H, m);
   float G = geom(Nf, H, Ln, V);
   color F;
   F = complexFresnelReflector(V, H, eta,
        kappa);

   cook += D*G*F;

   float vdotn = V.Nf;
   cook /= ( 4.0 * vdotn);
   return cook;
}
```

```
float blinnD( normal Nn; vector H; float exponent; )
{
   float ndoth = Nn.H;
   return  (exponent + 2.0 ) * 0.1591549 *
            pow( max(0.0, abs(ndoth) ), exponent );
}
```

```
float geom( normal Nn; vector H; vector L; vector V )
{
   float ndoth = Nn.H;
   float ndotv = Nn.V;
   float ndotl = Nn.L;
   float vdoth = V.H;

   float masking = 2 * ndoth * ndotv/vdoth;
   float shadowing = 2 * ndoth * ndotl/vdoth;
   return min( 1, min(masking, shadowing) );
}
```


Combined Shaders

- Combine both components
- Include absorption and internal reflection
- Physically *plausible*

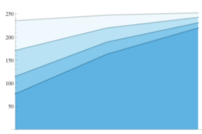


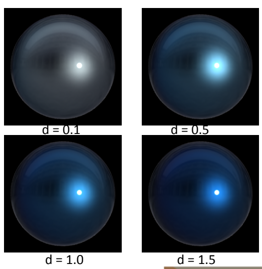
Absorption

- Input
  - Surface normal *N*
  - Viewing direction *V*
  - Light direction *L*
  - Thickness *d*
  - Absorption *alpha*
- Inverse colour!

```
extern vector I;
normal Nf, Nn;
float idotn;
Nn = normalize(N);
Nf = faceforward(Nn, I);
float r = 0, b = 0, g = 0;

vector Ln = normalize(L);
float ldotn = Ln.Nf;
float vdotn = V.Nf;
vdotn = clamp(vdotn, 0.001, 1.0);
ldotn = clamp(ldotn, 0.001, 1.0);

r = exp( -alpha[0] * d * (1.0/ vdotn + 1.0/ldotn) );
g = exp( -alpha[1] * d * (1.0/ vdotn + 1.0/ldotn) );
b = exp( -alpha[2] * d * (1.0/ vdotn + 1.0/ldotn) );

return (r,g,b);
```


Increasing Absorption





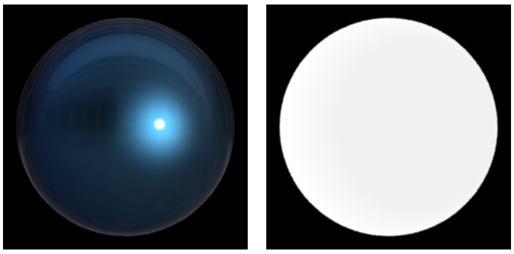
- Change in hue, saturation
- Alpha = [0.8 0.3 0.1]

Internal Reflection

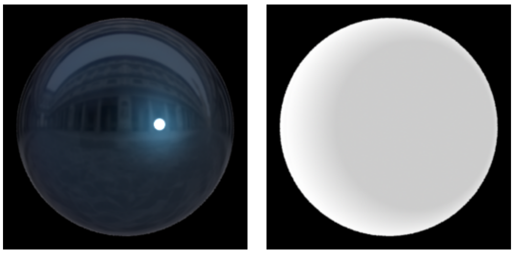
- Input
  - Surface normal *N*
  - Viewing direction *V*
  - Light direction *L*
  - Half vector *H*
  - IOR *eta*

```
float tir(
            normal N; vector V; vector Ln; vector H;
            float eta;
            )
{
   extern vector I;
   extern point P;
   normal Nf, Nn;

   Nn = normalize(N);
   Nf = faceforward(Nn, I);

   float G = geom(Nf, H, Ln, V);
   float F, Kt;
   fresnel(V, H, 1.0/eta, F, Kt);

   return (1-G) + (1-F) * G;
}
```

With and Without Internal Reflection



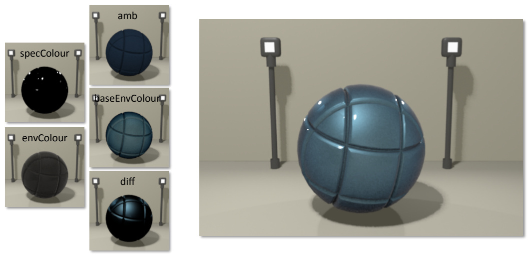
With and Without Internal Reflection



Final Shader

```
if ( H.V > 0.0 )
{
   specColor +=  Cl * torranceSparrowReflector(Nf, V, Ln, H, eta, kappa, beta);

   vector refractLn = refract(Ln, -H, eta_avg);
   vector refractV = refract(-V, -H, 1.0/eta_avg);
   color a = absorption(Nf, refractV, refractLn, alpha, d);
   float t = tir(Nf, refractV, refractLn, H, eta_avg);
   fresnel(V, H, 1 / eta_avg, fr, ft);
   envColor = envColor  + (env * fr);

   if ( base != "diffuse" )
   {
      spec_base = Cl * torranceSparrowReflector(Nf, V, Ln, H, eta_base, kappa_base, beta_base);
      diff += spec_base * ft * a * t;
    }
    else
      diff += ( (Cl * orenNayarReflector( Nf, I, L, sigma, P, Cs ) * basecolor)) * ft * a * t;
}
```

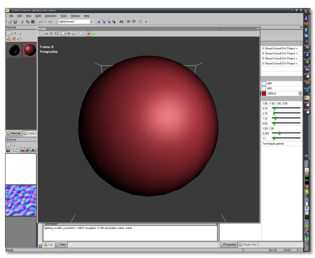

Result



Real-time Rendering

- Real-time implementation possible
- Simplifications needed
  - No “real” metallic reflections
  - Fresnel coefficients can be approximated and precomputed
  - Stored in a texture

Realtime Screenshots



Performance

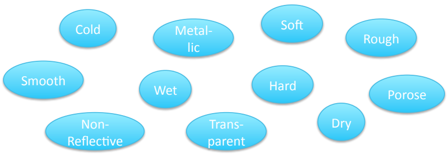
- Performance depends on
  - Number of layers
  - BRDF of the individual layers
- Overall performance overhead only a few percent
- Realtime possible with approximations

Andrea Weidlich

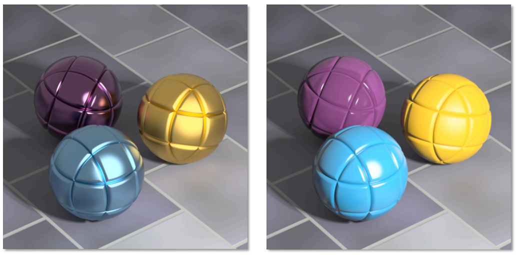
CLASSIFYING MATERIALS – USING LAYERED BRDFS TO DESCRIBE OBJECT APPEARANCE

Material Classification

- What gives a material its characteristic appearance?
- Easier if based on physical properties (e.g. reflectance)
- Simplifies modelling



Metallic vs. Non-Metallic

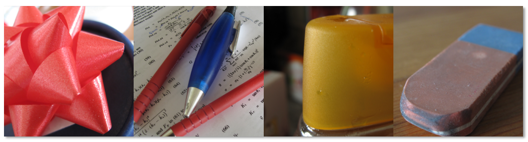


From Smooth To Rough – Metallic

smooth → rough

- Less reflective, environment less visible
- Highlight becomes bigger, blurrier, less prominent

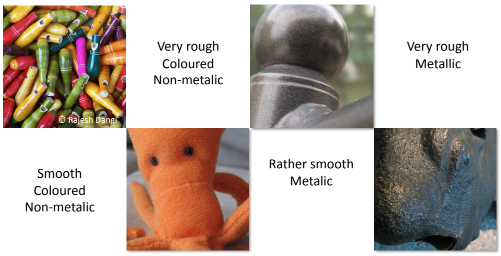
From Smooth To Rough – Non-Metallic



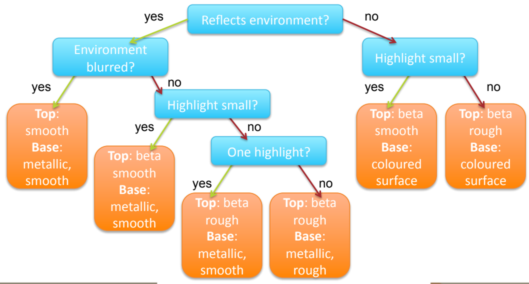
smooth → rough

- Coloured
- Highlight becomes bigger, blurrier, less prominent

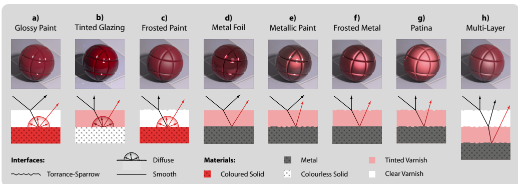
Examples



Classification According to Reflectance



Final Classification



Alexander Wilkie

MODELLING WITH LAYERED SURFACES

Traditional Materials

- Wide range of different materials
- Change of the layering and parameters of the individual surfaces
- Top surface always transparent Torrance-Sparrow surface
- Arbitrary numbers of layers possible
- Normally only two or three

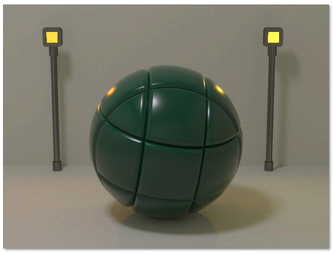
Type a) and b) – Glossy Paint

- Smooth top surface
- Roughness between 0.01 and 3 to 4 degrees
- Bottom surface diffuse (Lambert, Oren-Nayar, …)
- Index of refraction (IOR) between 1.35 and 1.7
  - Acrylic paint 1.4 –1.5
  - Plastic 1.46 – 1.55
  - Ceramic glazing ~ 1.6

Example Acrylic Lacquer

Torrance Sparrow
m = 3.5°
IOR = 1.5
d = 0.0
a = (0.0,0.0,0.0)

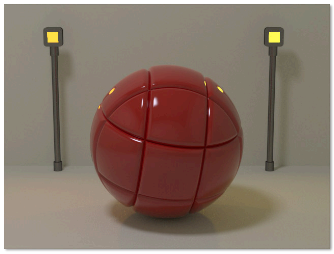
Oren Nayar
Sigma = 0.0
Colour = (0.06,0.3,0.15)



Example Red Lacquer

Torrance Sparrow
m = 1.5°
IOR = 1.5
d = 1.2
a = (0.01,0.99,0.99)

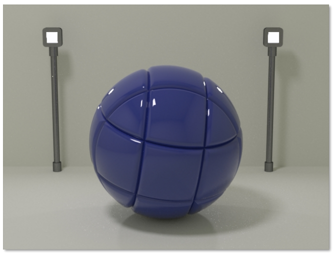
Oren Nayar
Sigma = 0.0
Colour = (1.0,0.03,0.05)



Example Ceramic Glaze

Torrance Sparrow
m = 0.1°
IOR = 1.7
d = 0.0
a = (0.0,0.0,0.0)

Oren Nayar
Sigma = 0.0
Colour = (0.25,0.25,1.0)



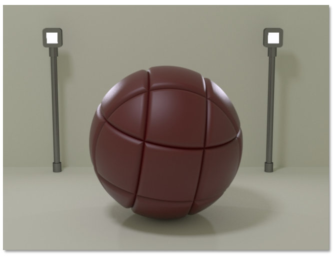
Type c) – Frosted Paint

- Low gloss component
- Rough top surface
- Roughness between 5 and 40 degrees
- Bottom surface diffuse (Lambert, Oren-Nayar, …)

Example Frosted Acrylic Paint

Torrance Sparrow
m = 10 °
IOR = 1.5
d = 0.0
a = (0.0,0.0,0.0)

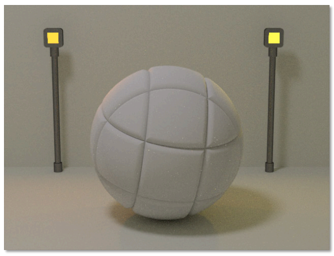
Oren Nayar
Sigma = 0.0
Colour = (0.3,0.02,0.01)



Example Sugar

Torrance Sparrow
m = 35°
IOR = 1.35
d = 0.0
a = (0.0,0.0,0.0)

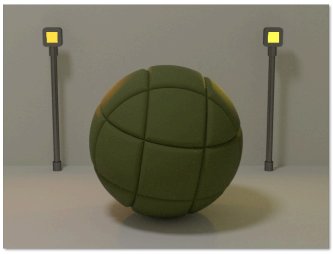
Oren Nayar
Sigma = 0.26
Colour = (1.0,1.0,1.0)



Example Latex

Torrance Sparrow
m = 25°
IOR = 1.3
d = 0.0
a = (0.0,0.0,0.0)

Oren Nayar
Sigma = 0.34
Colour = (0.3,0.4,0.1)



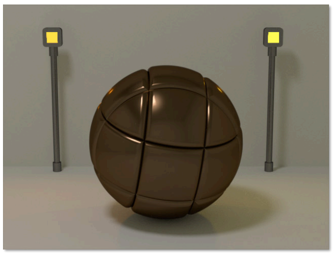
Type d) – Metal Foil

- Almost ideal specular surface
- Top surface smooth (roughness 3 to 4 degrees)
- Base surface smooth metallic (roughness up to 2 degrees)
- Fresnel component has to be computed for R, G and B component
  - Gold: IOR (0.1,0.42,1.56), kappa (3.8, 2.5, 1.9)
  - Silver: IOR (0.14,0.13,0.157), kappa (4.44, 3.25, 2.40)
  - Aluminium: IOR (1.94, 1.0, 0.6), kappa (8.21,6.69,4.86)

Example Brass

Torrance Sparrow
m = 1°
IOR = 2.0
d = 0.5
a = (0.6,1.0,1.0)

Torrance Sparrow
m = 4°
IOR = (0.1, 0.42, 1.56)
kappa = (3.8,2.5,1.9)



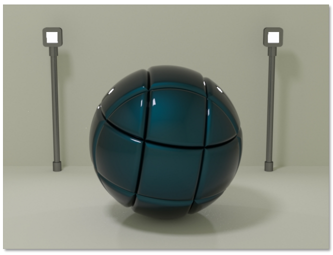
Type e) – Metallic Paint

- Specular surface with diffuse component (directional diffuse)
- Smooth top surface
- Rough metallic base surface
- Colour due to absorption

Example Car Paint

Torrance Sparrow
m = 1°
IOR = 1.5
d = 1.5
a = (0.98,0.4,0.28)

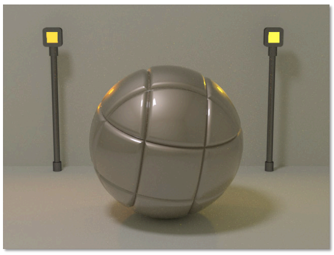
Torrance Sparrow
m = 12°
IOR = (1.94,1.0,0.6)
kappa = (8.21,6.69,4.86)



Example Gold Metallic Paint

Torrance Sparrow
m = 1°
IOR = 1.85
d = 0.2
a = (0.35,0.35,0.9)

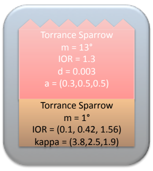
Torrance Sparrow
m = 13°
IOR = (0.14,0.13,0.16)
kappa = (4.44, 3.25, 2.4)

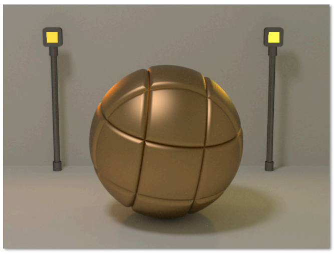


Type f) – Frosted Metal

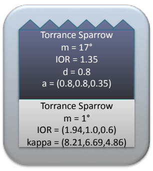
- Top surface rough
- Base surface smooth metallic
- Base surface “becomes” rough
- Top and bottom surface have nearly same effective roughness
- Highlights have approximately the same size

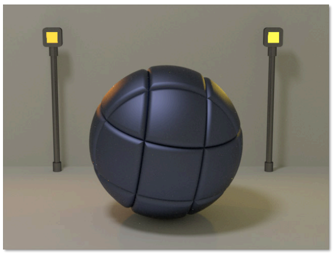
Example Gold Paint





Example Blue Metallic

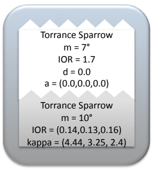


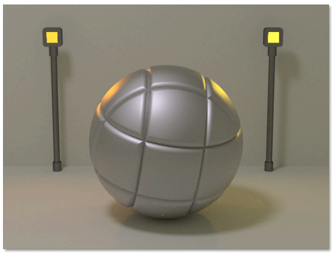


Type g) – Patina

- Diffuse surface
- Top surface transparent diffuse
- Bottom surface rough metallic
- Hardly any highlight is visible
- Often low IOR (1.3 to 1.5)

Example Silver Paint





Type h) – Multi-Layer

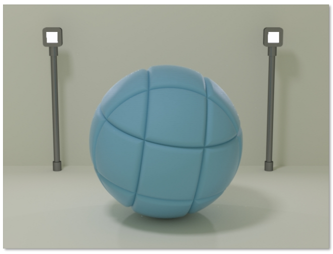
- Sometimes two layers not enough
- Can simulate complex materials with e.g. fanned-out highlights

Example House Paint

Torrance Sparrow
m = 30°
IOR = 1.3
d = 0.0
a = (0.0,0.0,0.0)

Torrance Sparrow
m = 7°
IOR = 1.3
d = 0.0
a = (0.0,0.0,0.0)

Oren Nayar
Sigma = 0.34
Colour = (0.5,0.9,1.0)

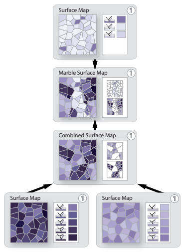


Heterogeneous Surfaces

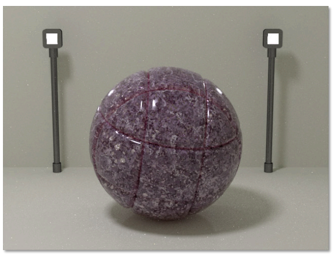
- Simulate natural structures with procedural textures
- Elements have different base roughness and layer thickness
- Procedural textures are used to simulate patterns
- Typical examples: sparkling effects (e.g. lacquer, gemstones)

Example Sunstone

Example: Lavender Lepidolite Close up



Example Leptdolite

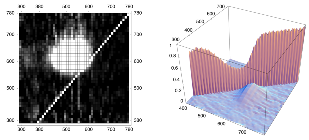


Various Heterogeneous Materials

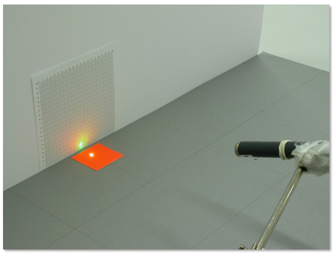
Fluorescent Surfaces

Fluorescence

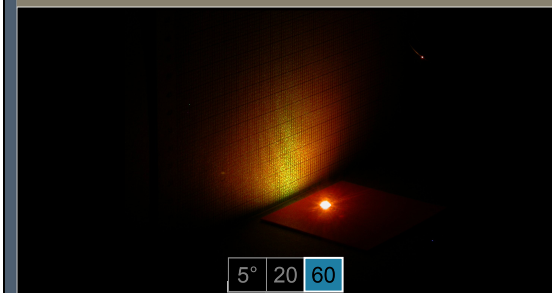
- Re-radiation of incident energy at different wavelengths
- Extends reflection spectra to matrices
- Common effect, but hard to measure - bispectral photometers needed



Fluorescent Reflection Experiment

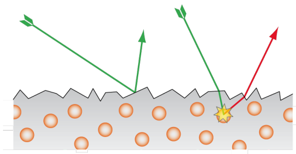


Fluorescent Sample @ 5°, 20°, 60°



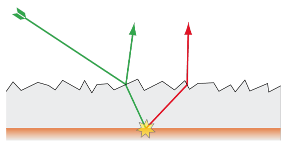
Bi-Coloured Reflection Pattern

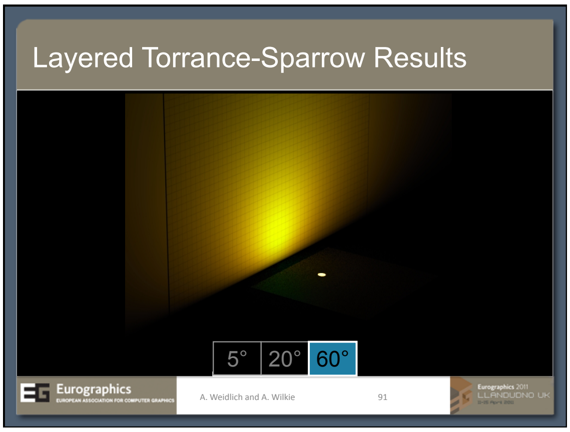
- Rays which are reflected by the substrate retain their colour
- Rays which interact with the colorant molecules undergo wavelength shift

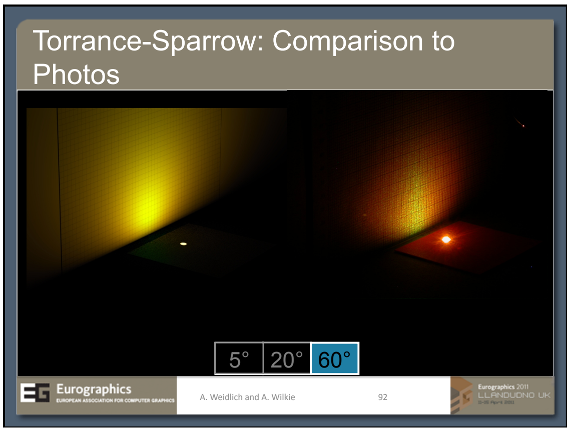


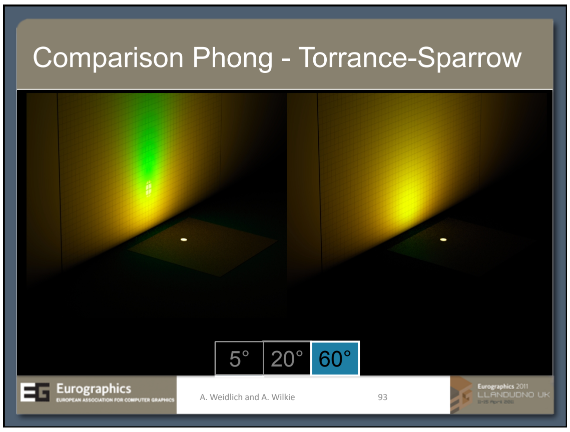
Layered Torrance-Sparrow Model

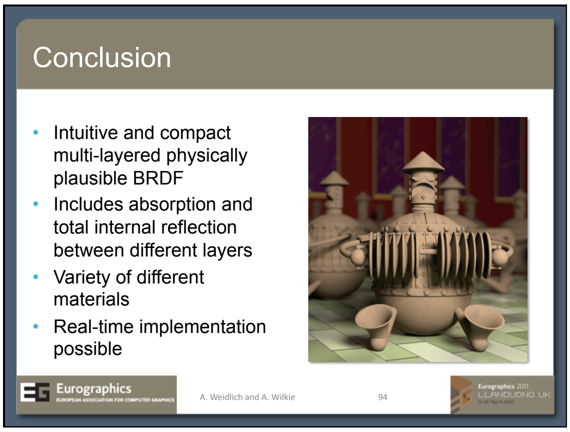
- Special case of surface type c) – rough dielectric layer over Lambertian fluorescent surface
- Re-radiation matrix instead of simple colour
- No attenuation in the substrate




Layered Torrance-Sparrow Results
5° 20° 60°
Eurographics
A. Weidlich and A. Wilkie
91


Torrance-Sparrow: Comparison to Photos
5° 20° 60°
Eurographics
A. Weidlich and A. Wilkie
92


Comparison Phong - Torrance-Sparrow
5° 20° 60°
Eurographics
A. Weidlich and A. Wilkie
93


Conclusion
• Intuitive and compact multi-layered physically plausible BRDF
• Includes absorption and total internal reflection between different layers
• Variety of different materials
• Real-time implementation possible
Eurographics
A. Weidlich and A. Wilkie
94