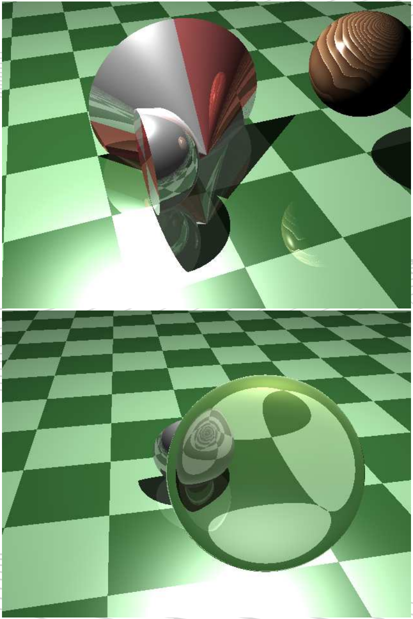
Ray tracing and procedural modelling

Computer Graphics (DT3025)

Martin Magnusson
November 21, 2016



Last time

- Rasterisation vs ray tracing
- The rendering equation (reflectance-only version)
- Ray–object intersections
    - Explicit ray equation and implicit surface equations
- A taxonomy of ray tracing methods

Today

- More about ray tracing
  - Refraction
  - “Distribution ray tracing”
  - Ray-tracing optimisations
- Sub-surface scattering
- Procedural modelling

Links to reading material in the book are given in footnotes.

Approximating the rendering equation

Hughes et al. 2013, Chp 31.

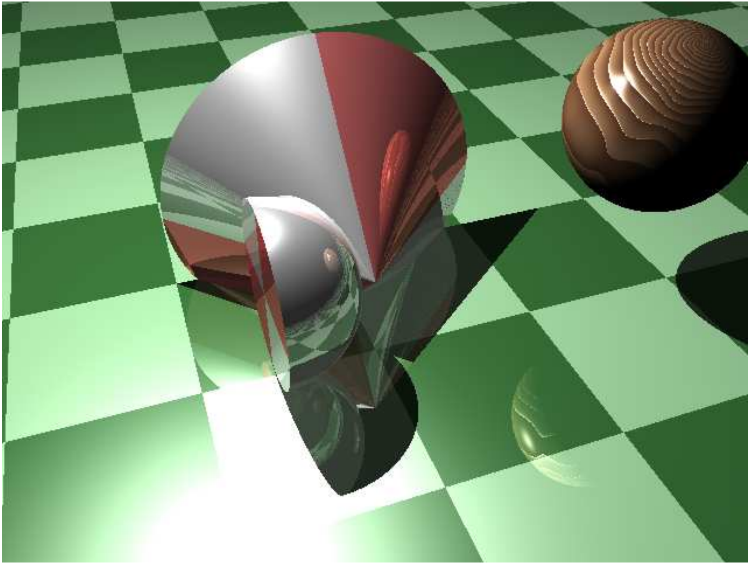
Ray tracing outline (1/2)

```
render_image {
  for (all pixels [x][y])  {
    ray R = getPixelRay( x, y, origin );
    colour C = raytrace( R, 1.0 );
    image[x][y] = C;
  }
}
```

Ray tracing outline (2/2)

```
colour raytrace( ray R, double contribution ) {
  point P = raycast( R, Scene ); // get closest intersection
  colour C = emitted light from P inwards R;
  for (all light sources L) {
    // cast shadow feeler to check light source line-of-sight
    if (L is visible from P) {
      C += light from L scattered at P inwards R; // L*BRDF*cos
    }
  }
  double c = P.mirror_amount;
  if (c > 0) {  // P is partly specularly (impulse) reflective
    dir = reflection direction; // or refraction...
    ray R_new( P, dir ); // new ray from P out towards dir
    C = (1 - c) * C;
    C += raytrace( R, contribution * c );
  }
  return C;
}
```

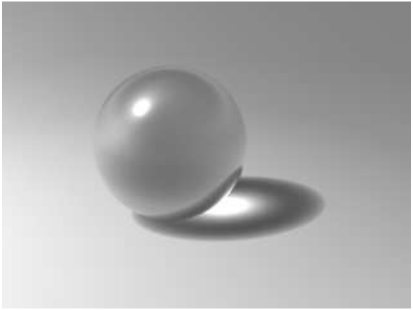
Multi-bounce specular reflections



Transparent shadows (shadow rays)

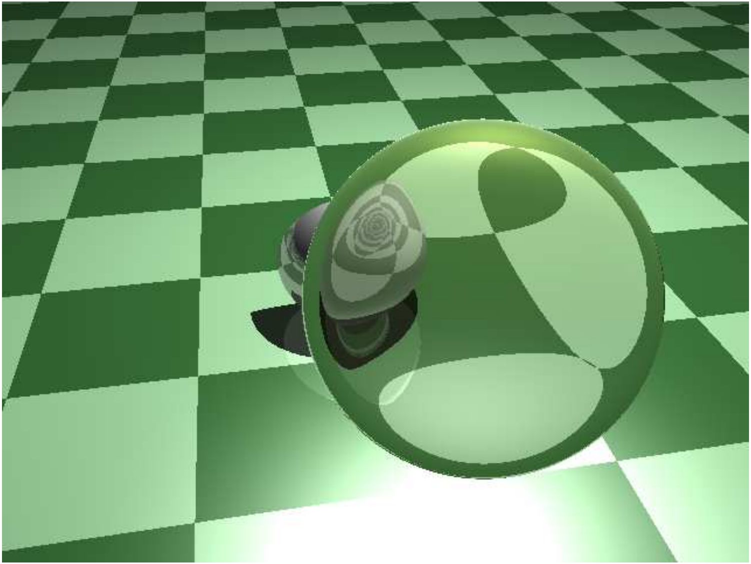

blender.org

Simple transparent shadow


happy-digital.com

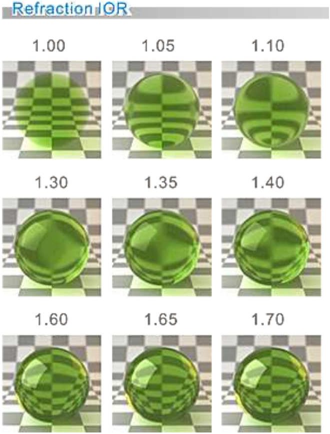
Caustics

Transparent objects (ordinary rays)



Bending of waves

- Light waves have different speed in different materials.
- Fastest in vacuum, slower in water, glass, etc.
- Index of refraction (IOR): speed ratio compared to vacuum.
  - Air: 1.0003
  - Water: 1.33
  - Glass: ≈ 1.5–1.9



vray.com

See Hughes et al. 2013, Chp 26.4.3.

Water wave refraction

- Why do waves often arrive straight towards the beach?
  - Water waves slow down in shallow water.
  - (Waves get taller and denser, but keep temporal frequency.)
  - Slowing wave down (light, water) doesn't affect *temporal frequency*,
  - so the wavelength (spatial frequency) must change.

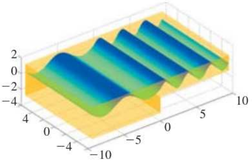


Figure 26.12: Waves traveling to the right bunch up as they slow down.

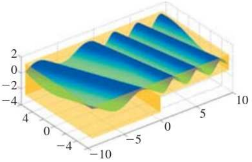
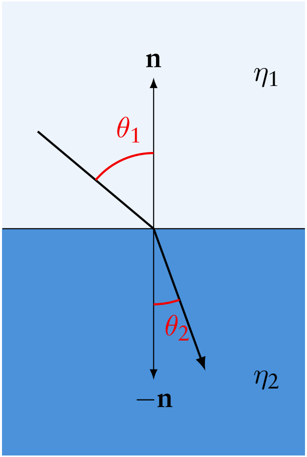


Figure 26.13: Off-axis waves change direction "at an interface."

Hughes et al. 2013

Snell's law

$$\frac{\eta_1}{\eta_2} = \frac{\sin\theta_1}{\sin\theta_2}$$



Avoiding sin

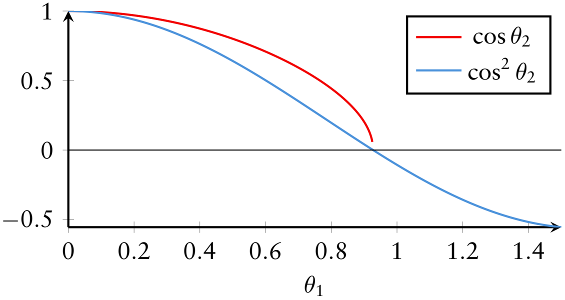
- It is more convenient to use cos in the computations (because we get that from $\mathbf{n} \cdot \boldsymbol{\omega}$).
- Using trigonometric identity $\sin^2 \alpha + \cos^2 \alpha = 1$,
- we can write Snell's law as

$$\cos \theta_2 = \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - (\cos \theta_1)^2)}.$$

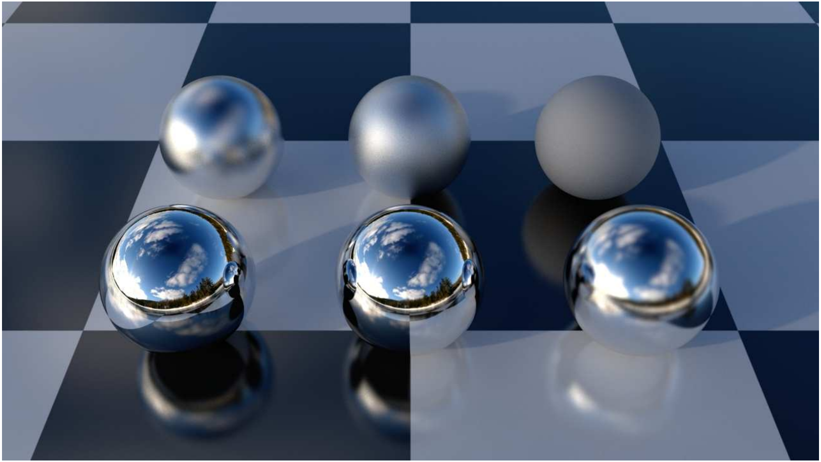
Critical angle

$$\cos\theta_2 = \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - (\cos\theta_1)^2)}.$$

- What if the stuff inside the sqrt is negative?
- Total internal reflection: no light gets transmitted.

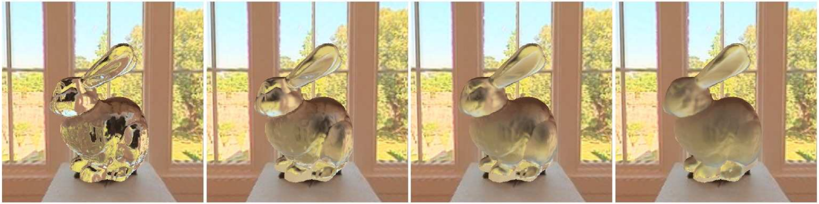


Distribution ray tracing: for non-impulse rays



elementalray.wordpress.com

Blurry translucency



Rousiers et al. 2011

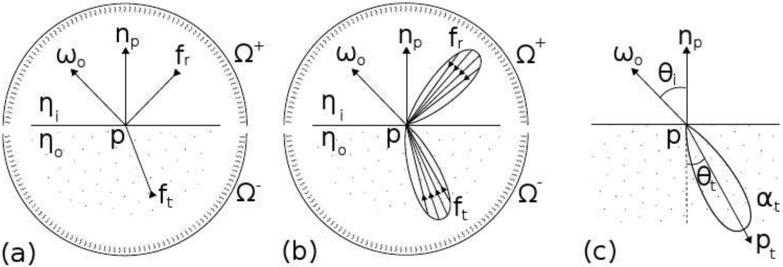
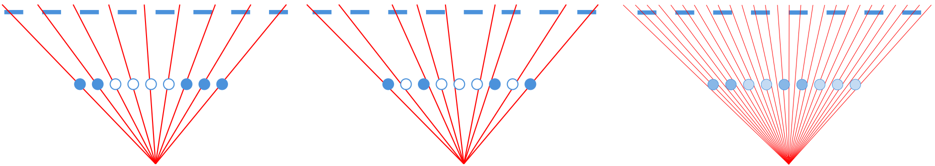


Figure 2: *BSDF and notation. (a) A specular BRDF f_r and a specular BTDF f_t, (b) A glossy BRDF f_r and a rough BTDF f_t,*

Supersampling (anti-aliasing)

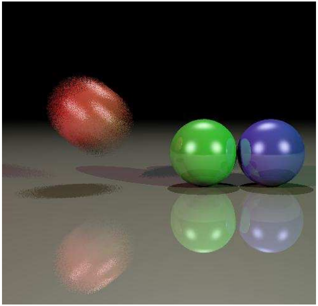
- Casting a single ray through the centre of each pixel leads to aliasing.
- Solution 1: randomise the direction within the pixel's area.
- Solution 2: cast many rays, and compute their average.

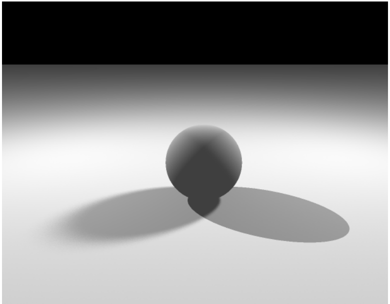


Other kinds of “distribution ray tracing”

Same principle can be applied for other phenomena too:

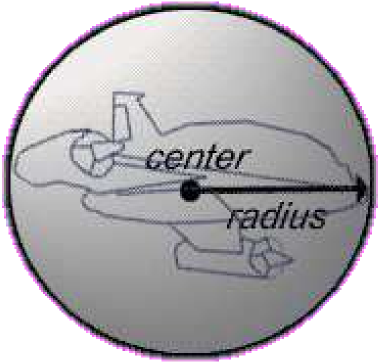
- Move objects, sample rays over time window: *motion blur*.
- Sample over lens area (not pinhole camera): *depth of field*.
- Sample several nearby point light sources: *area light*.





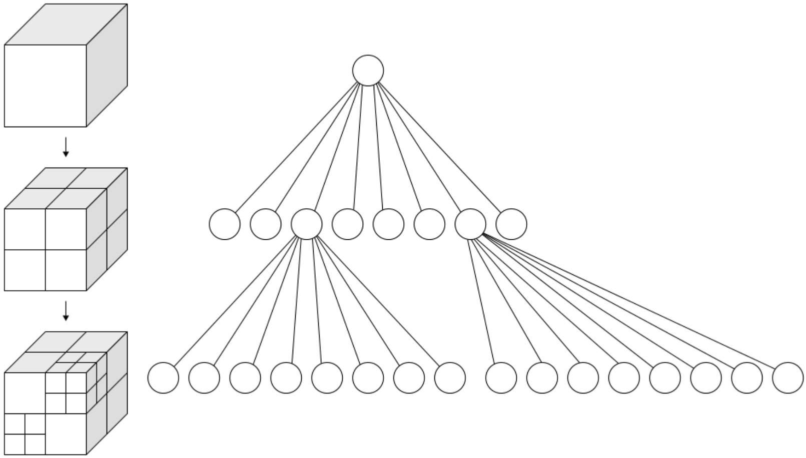
Optimisations

Bounding volumes




```
mesh{
  "complicated_object.3ds"
  bounded_by{
    sphere{
      <0,0,0>, 10
    }
  }
}
```

Octrees



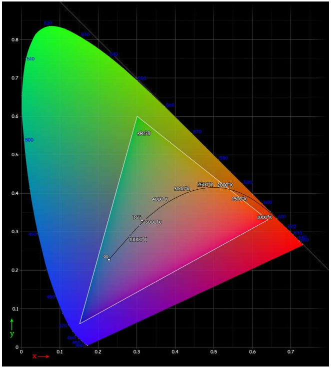
Hughes et al. 2013, Chp 37.6

Light transport is a function of frequency

- More general form of the reflectance equation:

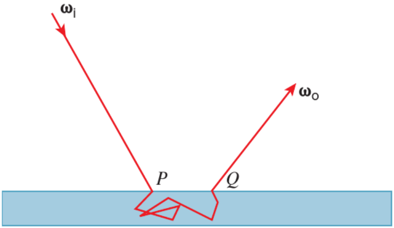
$$L^{\mathrm{ref}}(P, \omega, \lambda) = \int_{\omega_\mathrm{i} \in \mathbb{S}^2_+(P)} L(P, -\omega_\mathrm{i}, \lambda) f_\mathrm{r}(P, \omega_\mathrm{i}, \omega_\mathrm{o}, \lambda)(\omega_\mathrm{i} \cdot \mathbf{n}_P)\, \mathrm{d}\omega_\mathrm{i}$$

- λ is *frequency* (related to *wavelength* in reference medium).
- Approximate: compute separate equations for R, G, B colours, just as before.
- (Although the R, G, B colours in common colour models do not correspond to a particular light frequency.)

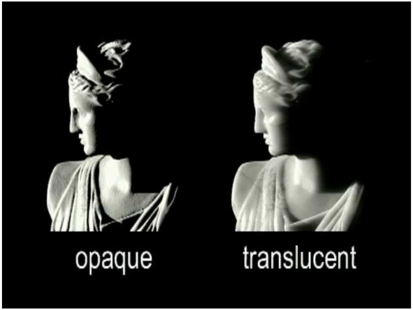


Spigget, via Wikipedia

BRDF vs BSSRDF



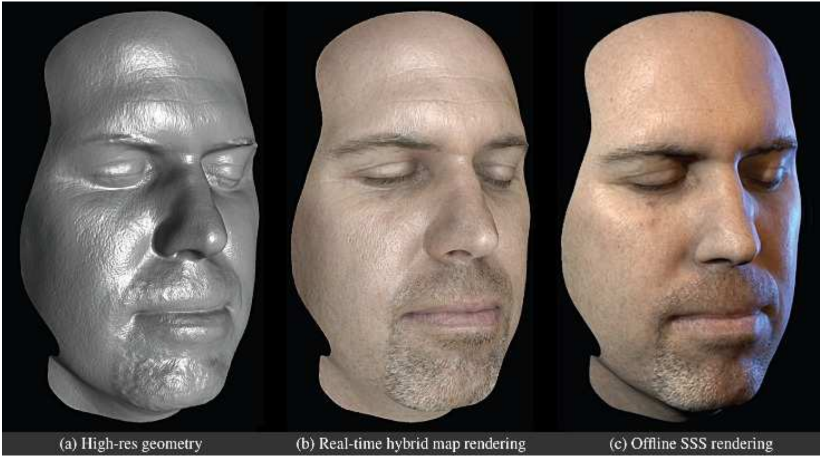
Hughes et al. 2013



Jensen et al. 2001

neilblevins.com

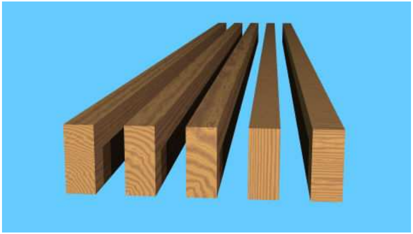
Acquiring BSDFs (out of scope)



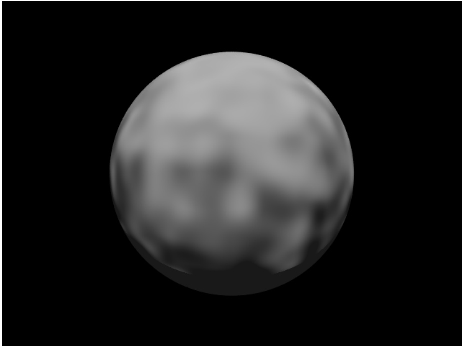
Ma et al. 2007

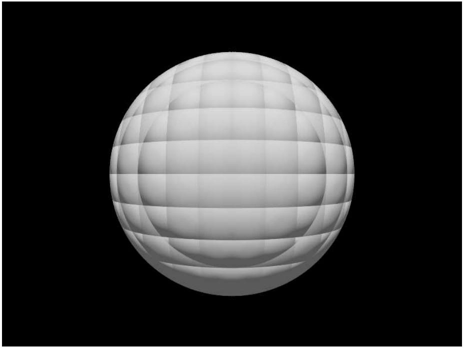
Procedural textures

Procedural wood and stone example

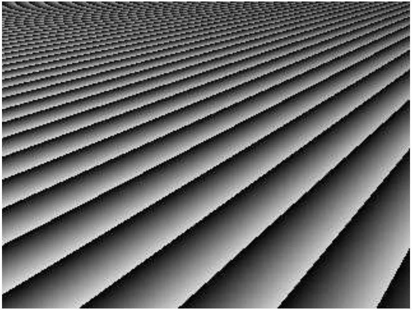


Procedural normals

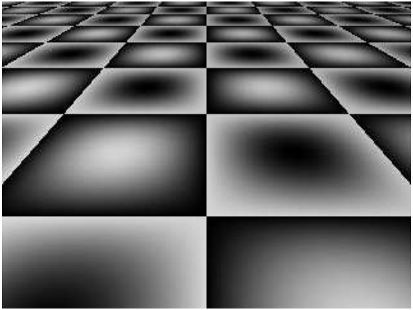




Example functions

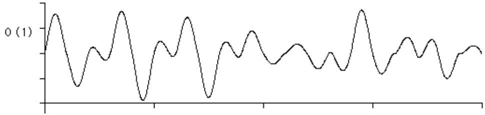


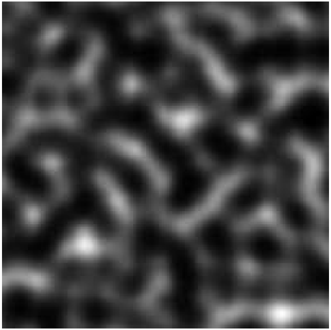
$\mathrm{fmod}(x + z, 1.0)$



$\mathrm{fmod}(\sin(x) \cdot \cos(y), 1.0)$

Perlin noise

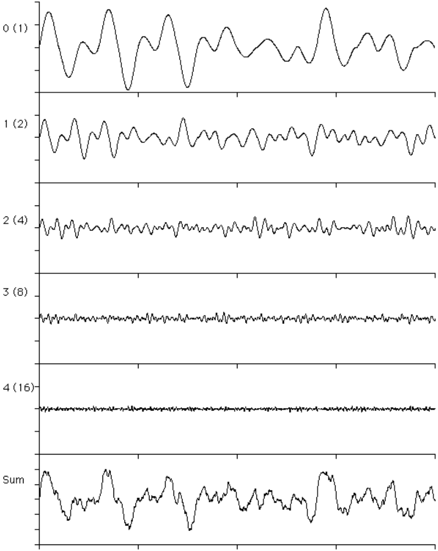


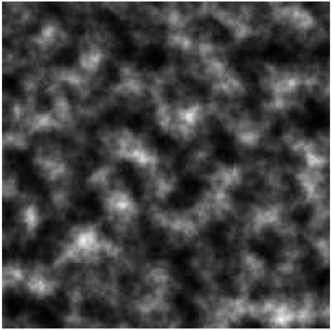


Paul Bourke

See Hughes et al. 2013, Chp 20.8.2; and also Perlin's home page if you like.

Multi-frequency Perlin noise

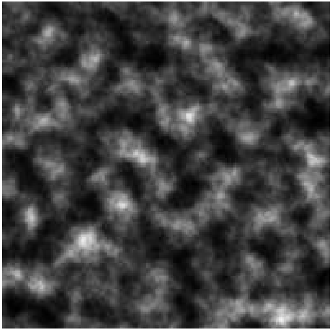


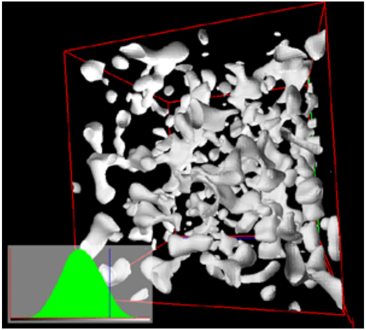


Paul Bourke

Perlin noise in 2D and 3D

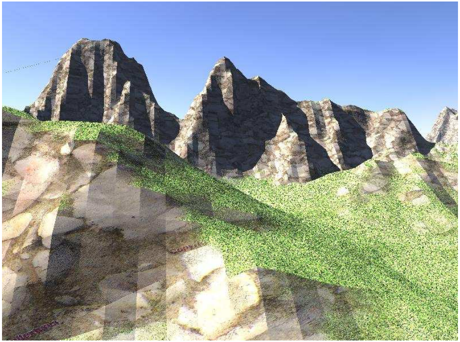
- Analogously in 2D and 3D:





Paul Bourke

Height field vs isosurface landscape



Jaime Vives Piqueres

Example of isosurfaces

(Note: using POV-Ray syntax)

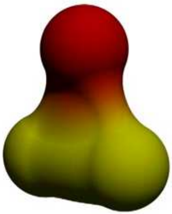
```
#declare F1 = function { x*x + z*z - 0.04 };
#declare F2 = function { min(F1(x,y,z-0.25), F1(x,y,z+0.25)) };
#declare F3 = function
  {  F2(x*cos(5*y)+z*sin(5*y),y,x*-sin(5*y)+z*cos(5*y)) };

isosurface {
  function { F1(x,y,z) }
  contained_by { box { <-3.0, 0.0, -3.0>, <3.0, 2.0, 3.0> }}
  accuracy 0.01
  max_gradient 50.0
}
```

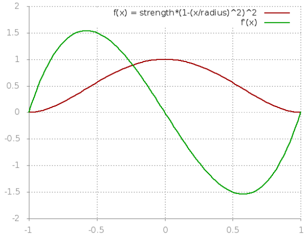
F1 F2 F3

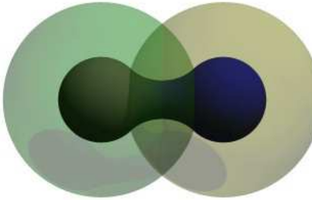
Blobs

- Adding (spherical) density functions.
- "Blobs"/"metaballs".



Benedek Vartok





POV-Ray user manual

Example of blob modelling

Summary

- Ray tracing: secondary rays for reflection and refraction
- Distribution ray tracing: sample more rays for “blurry” phenomena
- Bounding volumes and space partitioning for optimising intersection tests
- Subsurface scattering in the rendering equation
  - Can be done cheaply with precomputed scattering function
- Procedural surfaces and materials
  - Perlin noise

Limitations of ray tracing

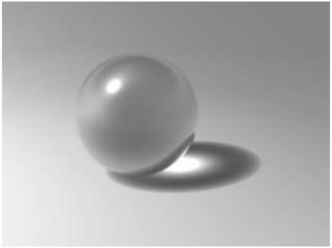
Next lecture: indirect lighting and path tracing

Time and place

- Mon Nov 28, 13.15–15.00
- T-141

Reading material

- Hughes et al.:
  - 29.3
  - 31.10
  - 31.16–17
  - (31.18 — but it's overly detailed)



References

- John F. Hughes et al. (2013). *Computer graphics: principles and practice (3rd ed.)* Boston, MA, USA: Addison-Wesley Professional, p. 1264.
- Henrik Wann Jensen et al. (2001). “A Practical Model for Subsurface Light Transport”. In: *SIGGRAPH’2001*. URL: https://graphics.stanford.edu/papers/bssrdf/.
- Wan-Chun Ma et al. (2007). “Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination”. In: *Eurographics*.
- Charles de Rousiers et al. (2011). “Real-time Rough Refraction”. In: *Symposium on Interactive 3D Graphics and Games*. I3D ’11. San Francisco, California: ACM, pp. 111–118. URL: http://doi.acm.org/10.1145/1944745.1944764.