LIGHT BAKING WITH IRAY

Martin-Karl Lefrançois

May 2017

NVIDIA

LIGHT BAKING

What is it?

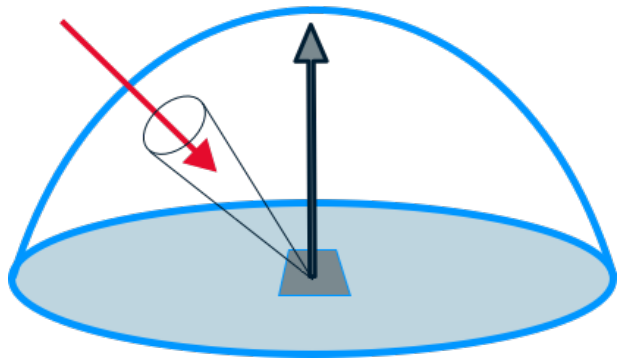
- Computing and storing the information of complex lighting to be used in a real time environment.
- Freezing lights information and storing the data that paint how the light rays bounce around static geometry.
- Caching pre-calculated lighting information in textures or per-vertex or some other form to recreate the lighting in real time.

IRRADIANCE

What is Irradiance

- The act of irradiating; emission of rays of light.
- That which irradiates or is irradiated.
- The radiant power received by unit area of surface (physics)

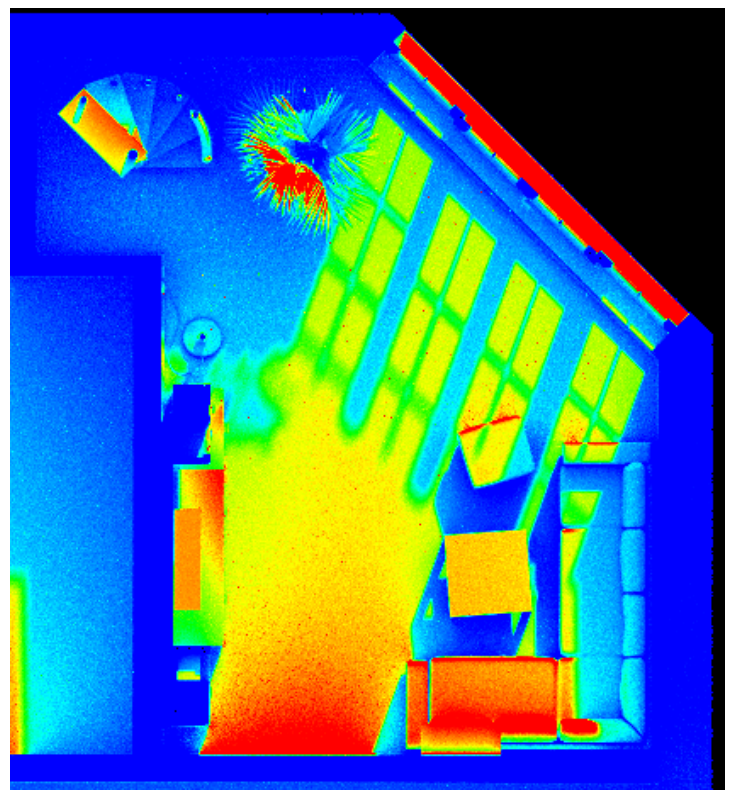
Source: Wiktionary



IRRADIANCE

Intensity of Light

Irradiance is a measurement of solar power and is defined as the rate at which solar energy falls onto a surface.



"irradiance" and "intensity of light" are the same

TOOLS FOR MEASURING IRRADIANCE

Measuring light intensity

Light meter / Exposure Meter

- An instrument that measures the intensity of the light reflected from or falling on a subject
- Calculates the optimum exposure depending on the film speed.

LIGHT METER / LUX METER

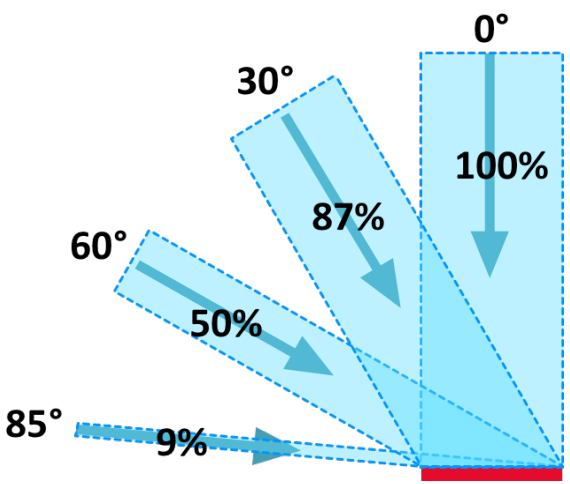
Measuring Brightness

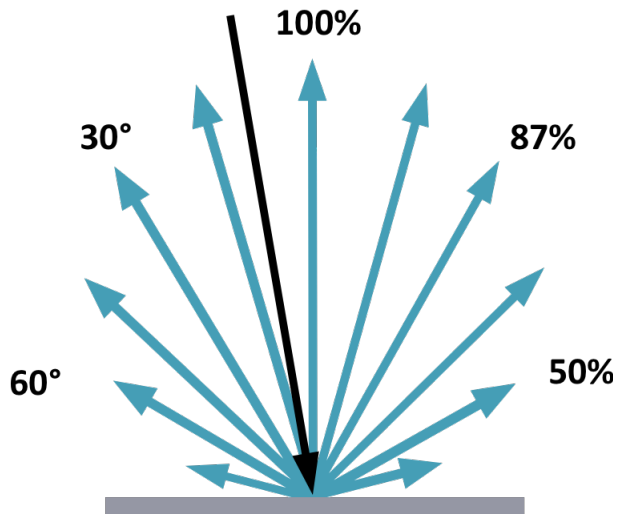
- Lux meter for measuring illuminances.
- It is equal to one lumen per square meter.

DIFFUSE LIGHTING

Lambertian reflectance

Lambert's cosine law in its reversed form (Lambertian reflection) implies that the apparent brightness of a Lambertian surface is proportional to the cosine of the angle between the surface normal and the direction of the incident light.





Cosine Law: $E_\theta = E \bullet \cos(\theta)$

LIGHTS

Diffuse lighting

```
for (i = 0; i < n_lights; ++i)
   result += lightContribution(L[i], N) * diffuse_color;
```

```
Color lightContribution(light& L, vec3 normal) {
  return L.color * dot(L.direction, normal); // -shadow, -AO, …
}
```

LIGHTS

Total contributions

```
Color all_lights(0,0,0)
for (i = 0; i < n_lights; ++i)
   all_lights += lightContribution(L[i], N);
result += all_lights * diffuse_color
```

$$\sum_i Lc_i(Ld_i \bullet N)$$

LIGHTS

Various sources of energy

Environment

- High Dynamic Range Image (HDRi)
- Sun & Sky

Implicit Lights

- Point, spot, ..
- Area: sphere, rectangle, ..

Emissive Objects

- Geometry emitting some energy

IRAY PHOTOREAL

Implementing Light Baking

Iray Photoreal

- Path tracer
- Physically based

MDL (Material Definition Language)

- Define the properties of the material
- Absorption, reflectivity
- Emissivity

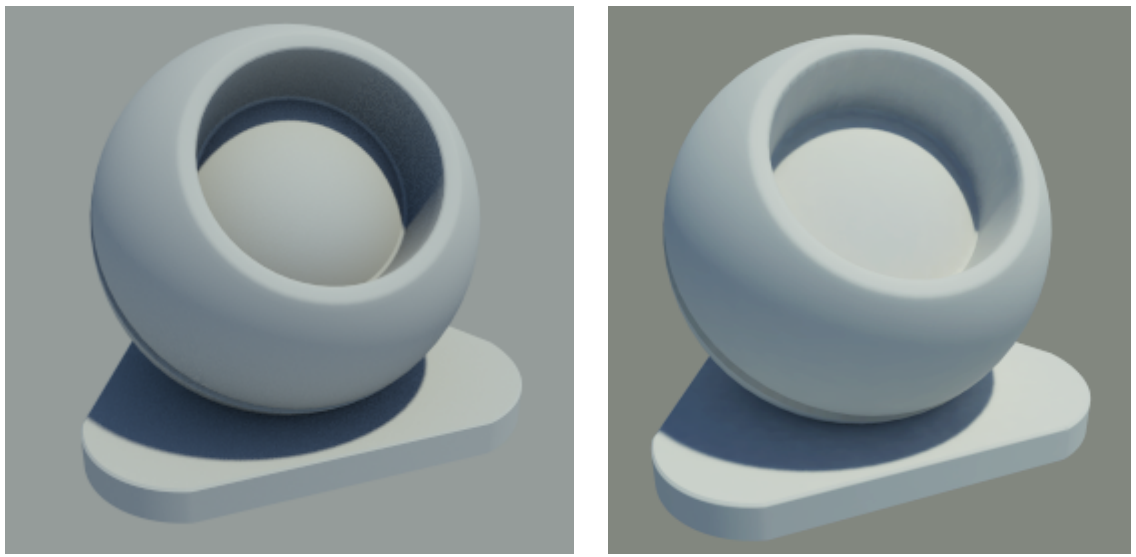
NVIDIA®
Iray®

LIGHT BAKING

Adding all incoming light to a light map (texture)

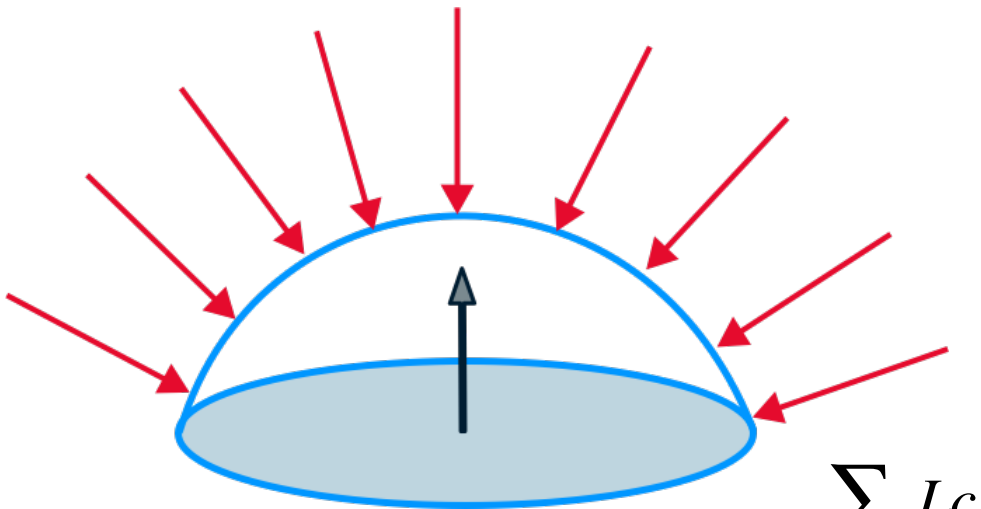
Gathering lighting effect from ray tracer to realtime

- Emission from the Sun & Sky and other HDRi
- Lights and area lights
- Bouncing light (Global illumination)
- Shadows and ambient occlusion
- LPE compatible
- Using MDL



LIGHT PROBES

Sampling the Hemisphere



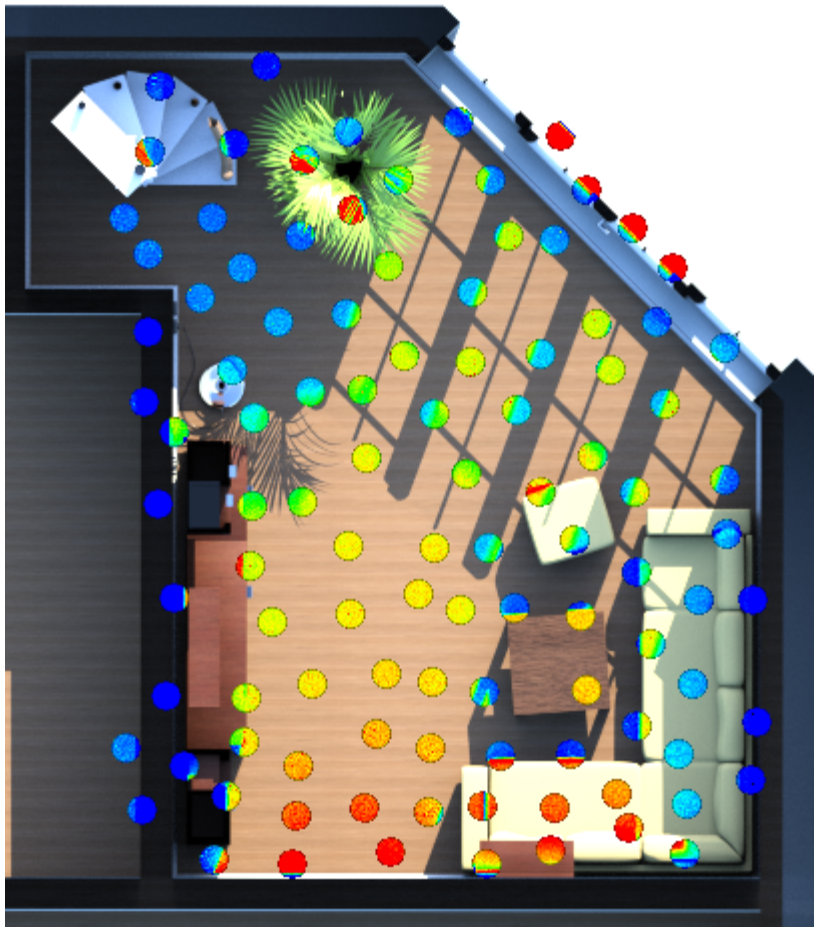
$$\sum_i Lc_i (Ld_i \bullet N)$$

Sampling the hemisphere for all light contribution defined by a point and a normal

LIGHT BAKING

Irradiance Probes (capturing elements)

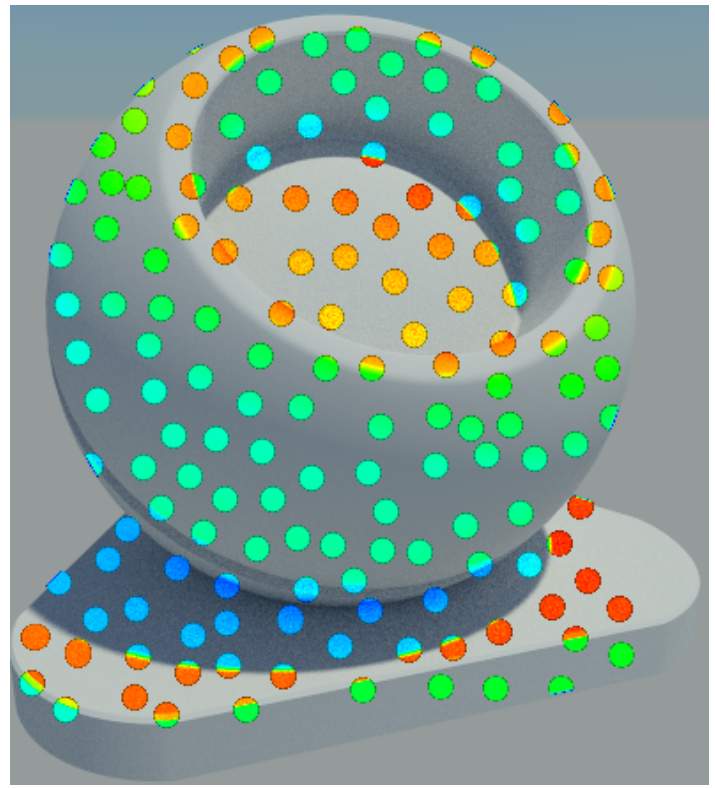
- Irradiance probes are used to render the irradiance at certain locations in the scene.
- Orientation of the camera is irrelevant



LIGHT BAKING

Applying on Object

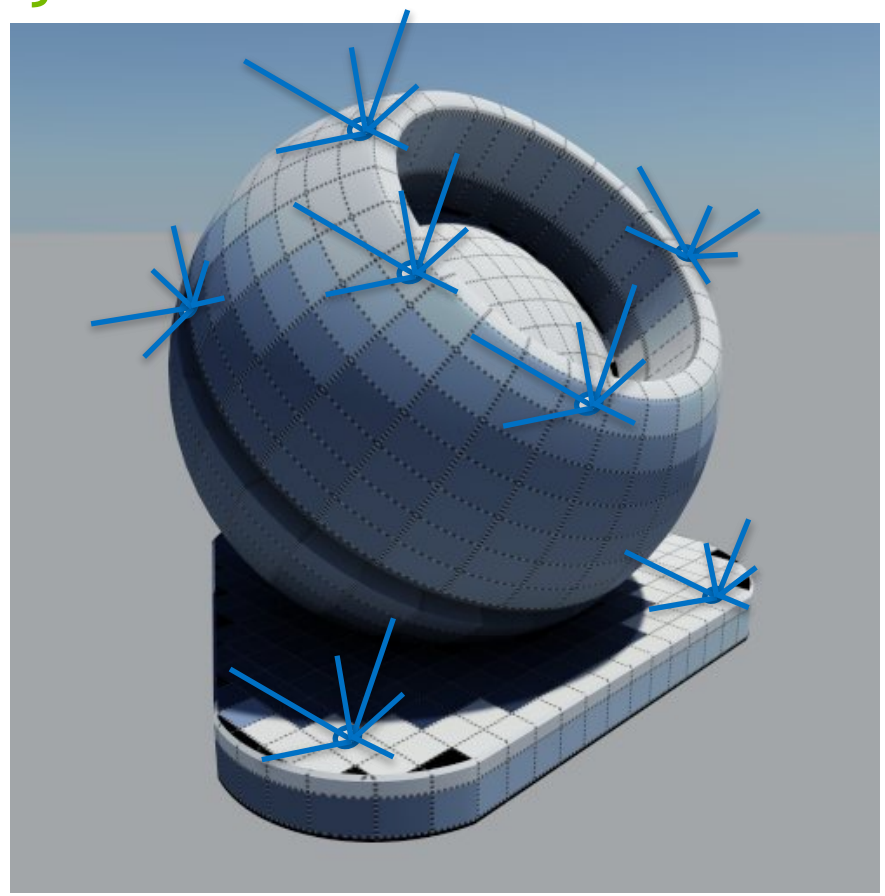
Use light probes and place them on the surface of the object



LIGHT BAKING

Sampling the object

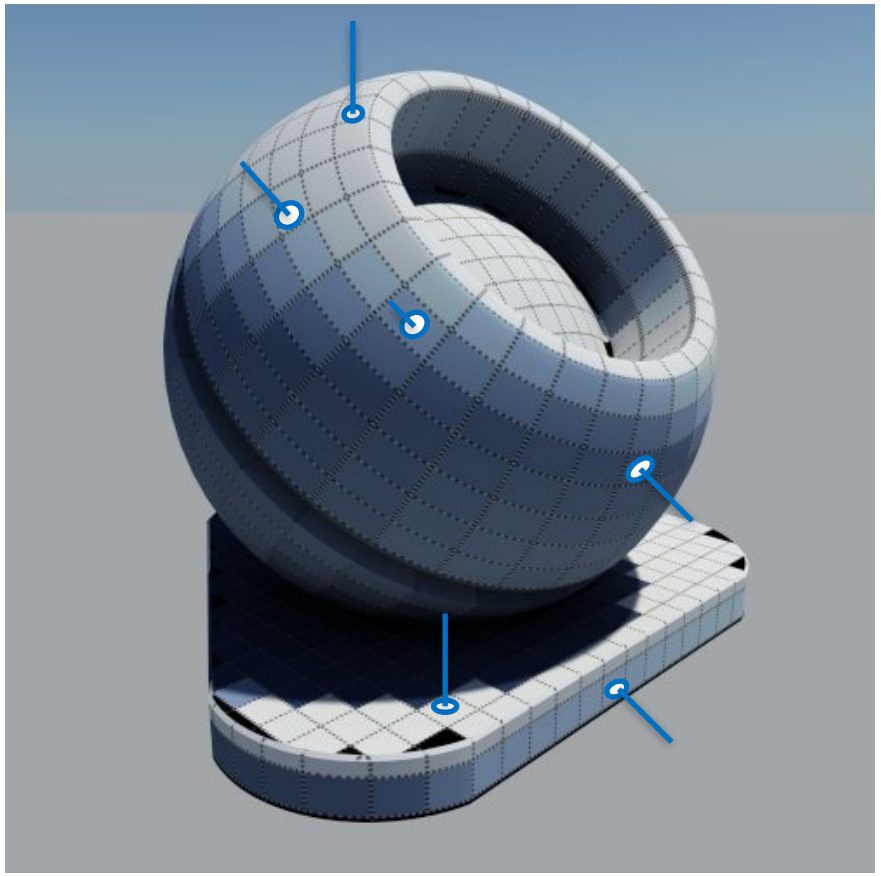
Sampling the object in 3D space for each texel.



LIGHT BAKING

Irradiance Probes

For each texel, add a light probe



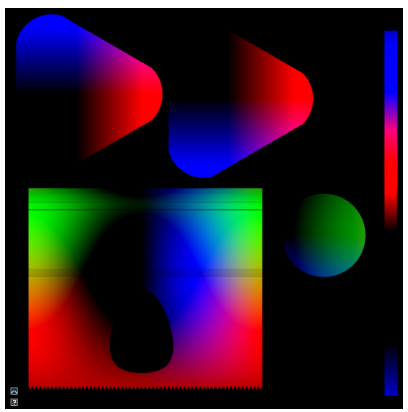
LIGHT BAKING

Finding 3D position of each texel

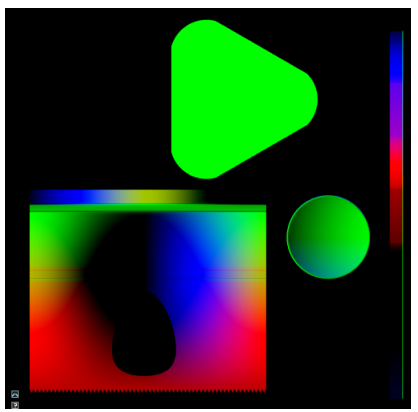
Rasterizing all triangles in the UV domain

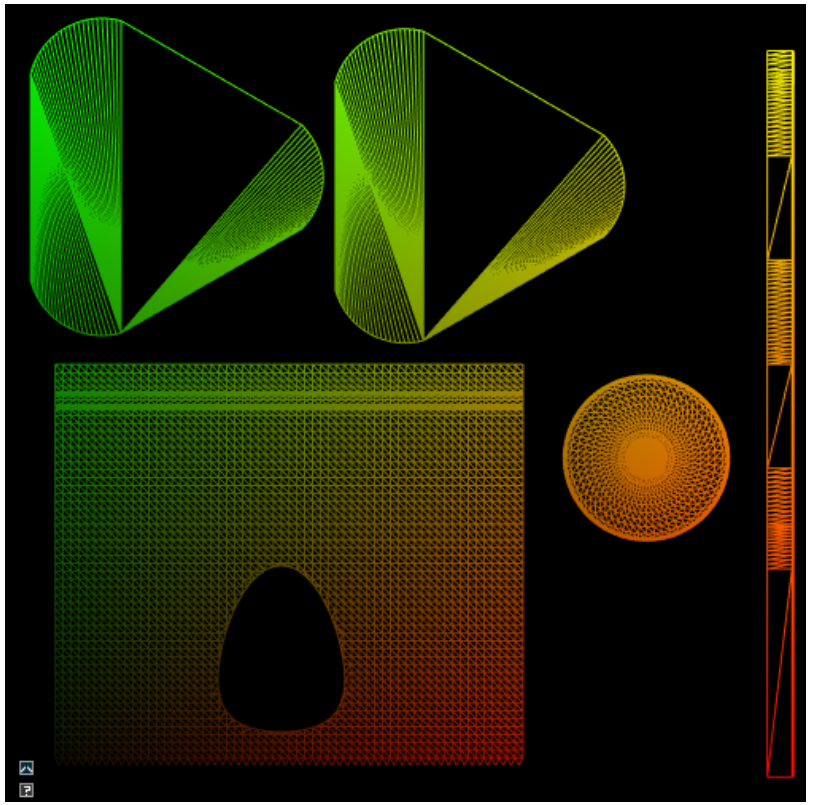
- Output position and normal

Position



Normal





UV BAKING

GLSL for UV baking

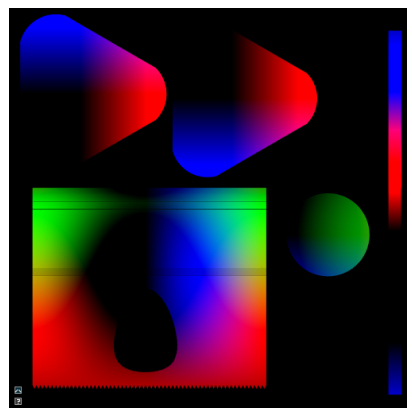
Vertex:

```
gl_Position = vec4(texCoord, 1) * 2.0 - 1.0;
```

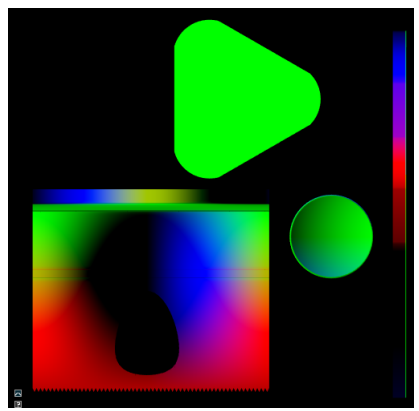
Fragment:

```
fColor1 = vec4(varWorldPos, 1.0);
fColor2 = vec4(varNormal, 1.0);
```

Position



Normal



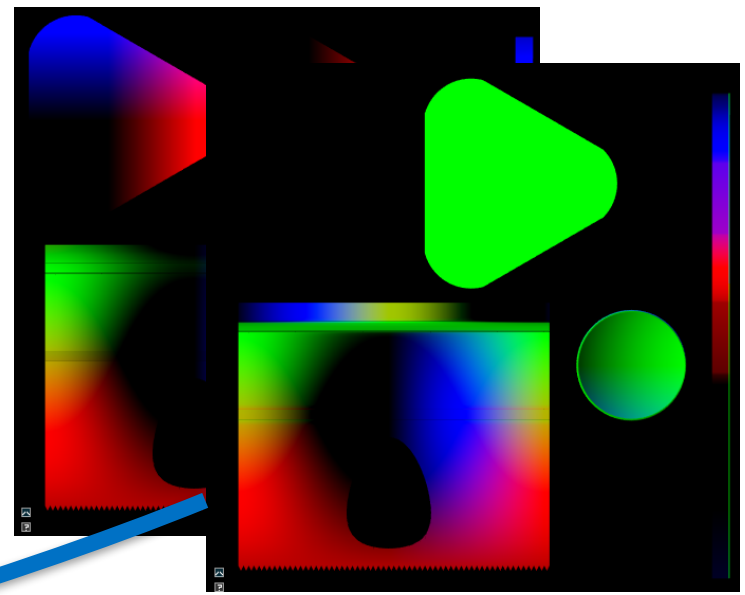
LIGHT BAKING

Convert to Irradiance Probes

Extract all points and normals

Construct a list of irradiance probes

Render

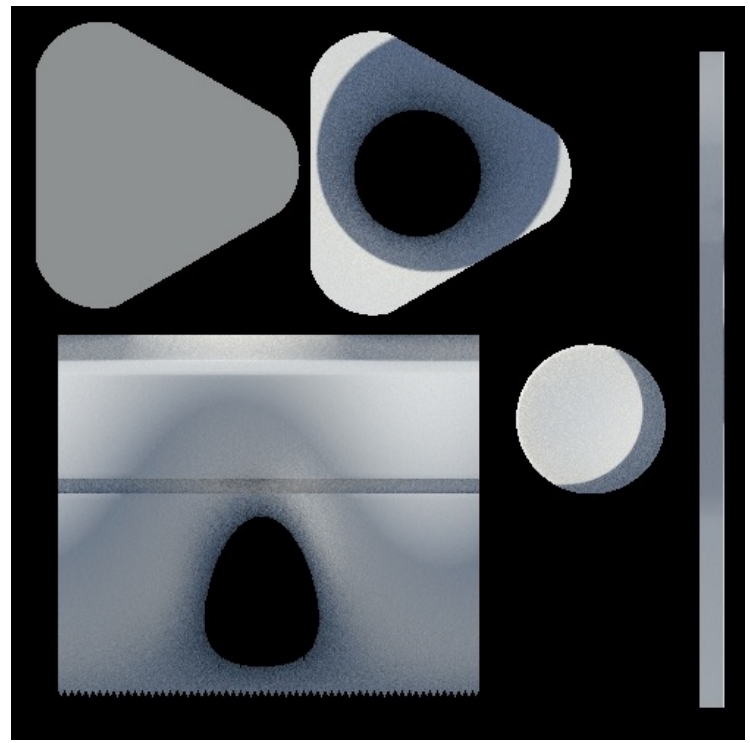


Lp = {{p0,n0}, {p1,n1}, {p2,n2},…}

LIGHT BAKING

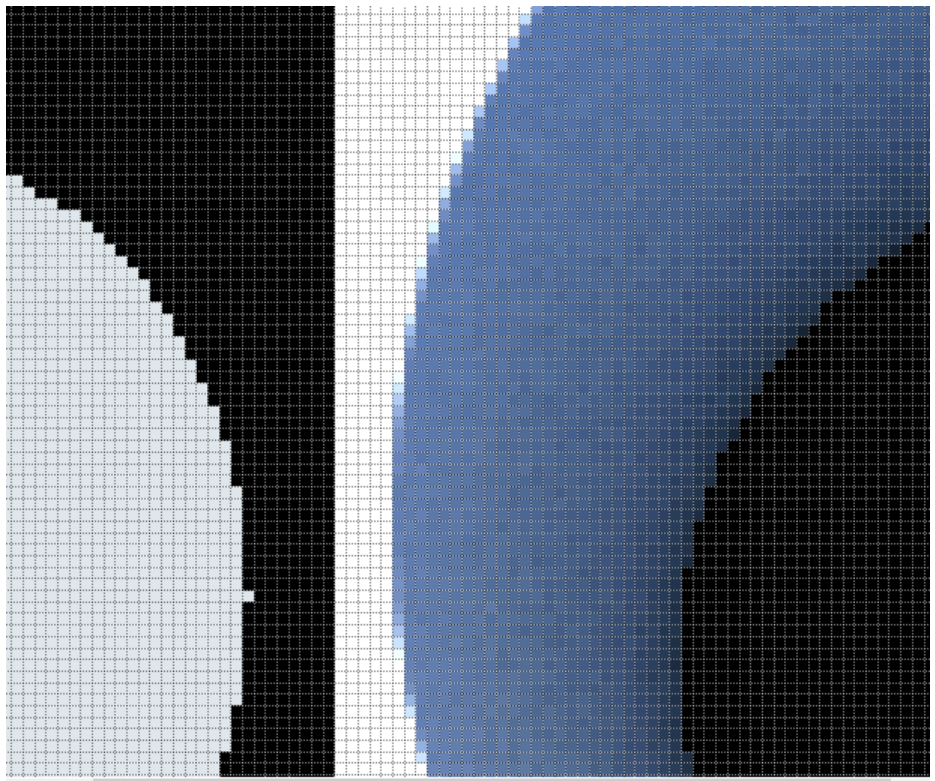
Render

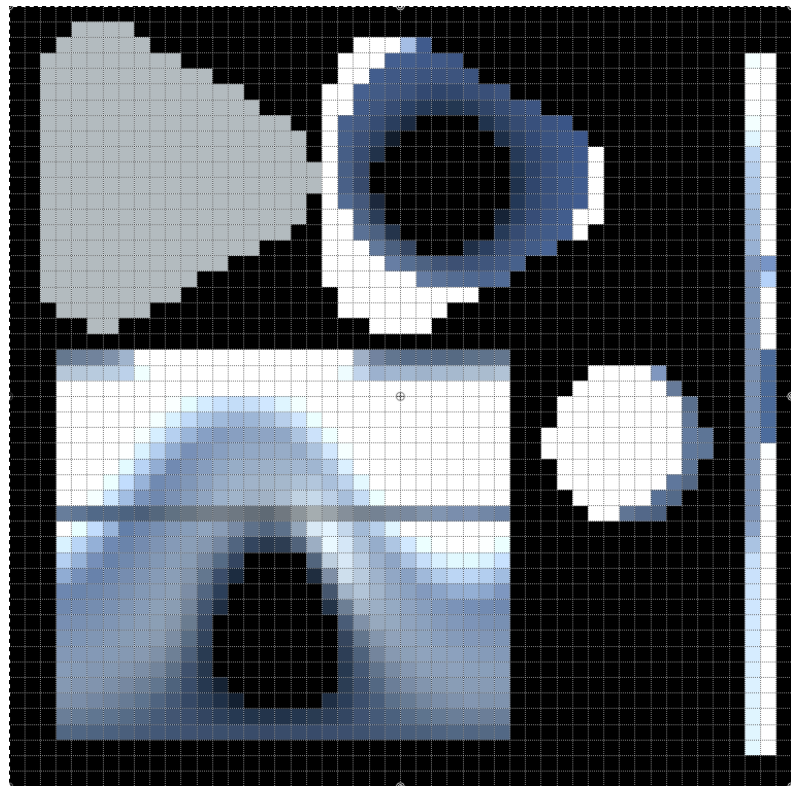
- Render using ‘iray'
- No tonemapper
- RGB floating point to receive the irradiance values
- Use values to put in an image at the earlier extracted positions



LIGHT BAKING

Different resolutions





LIGHT BAKING

Result on Object

Without and with light map texture

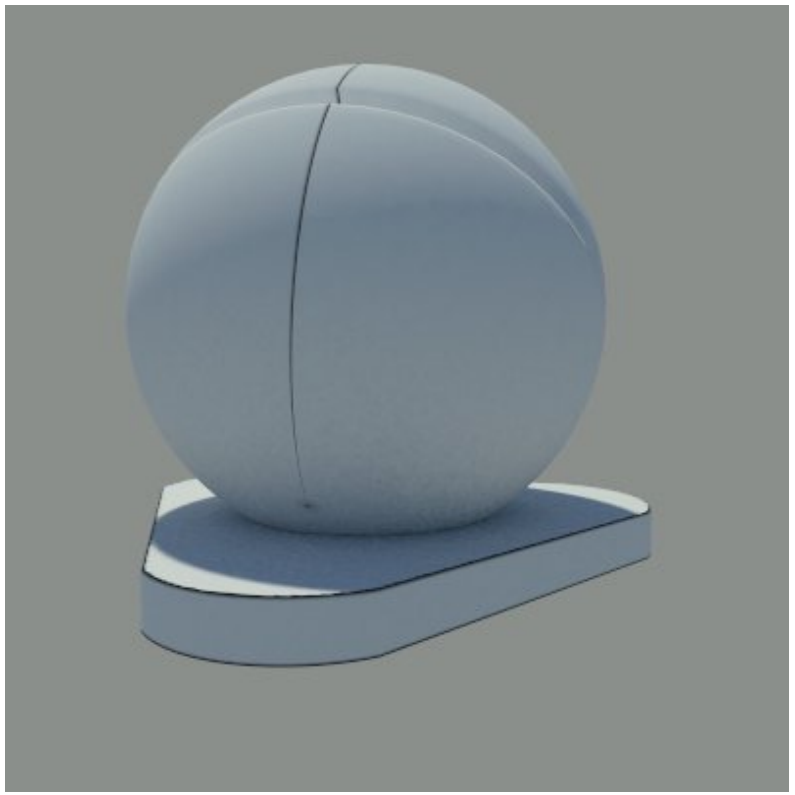
LIGHTING

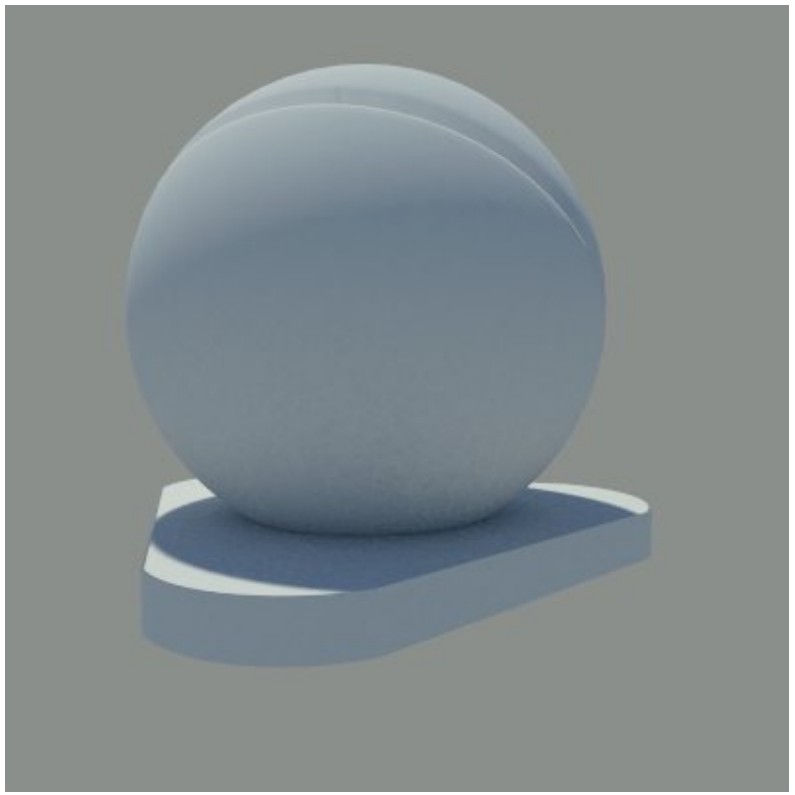
GLSL Fragment for lighting

```
vec4 cc = color;
if (has_texture)
{
  // Ajdusting the texture to linear space
  cc = pow(texture(tex, varTexCoord0.xy), vec4(2.2));
}
fragcolor = cc * texture(lbk_tex, varTexCoord1.xy);
```

LIGHT BAKING ISSUES

Seam and borders

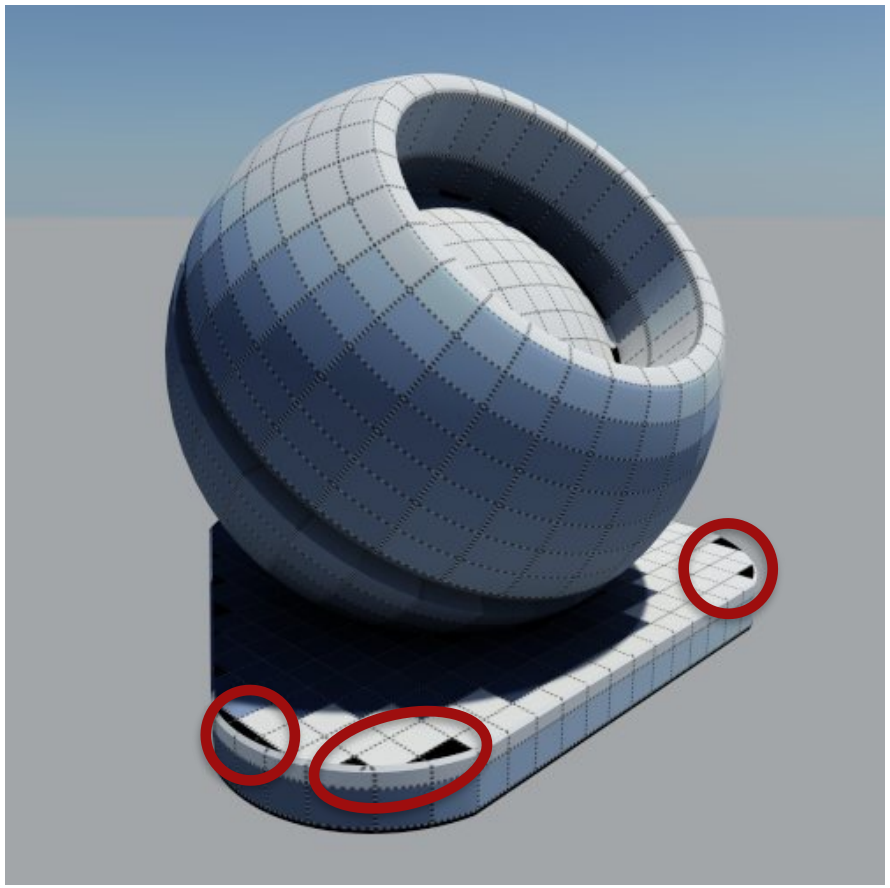




Fixed by enlarging the contour

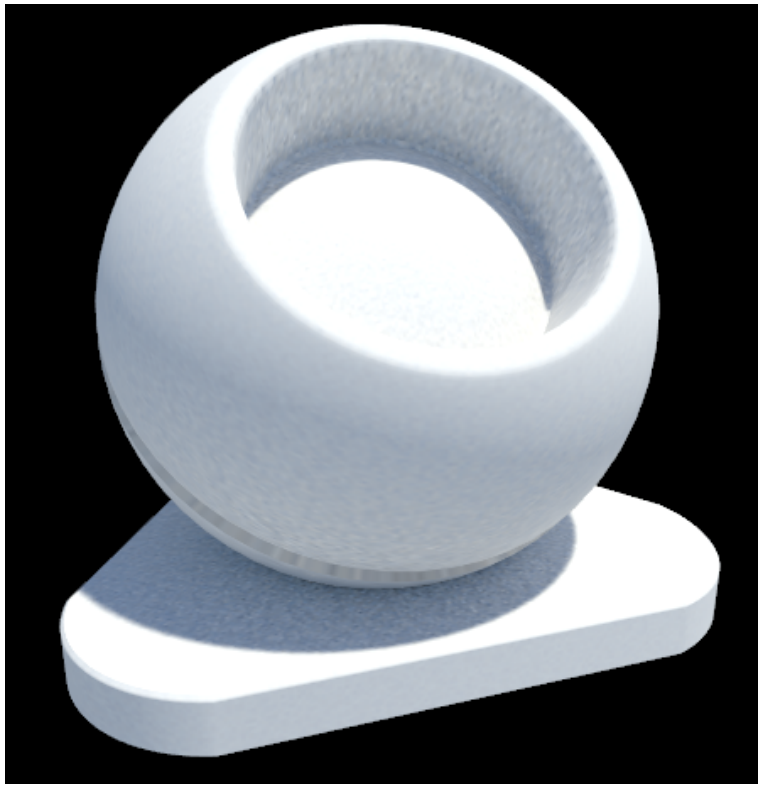
LIGHT BAKING

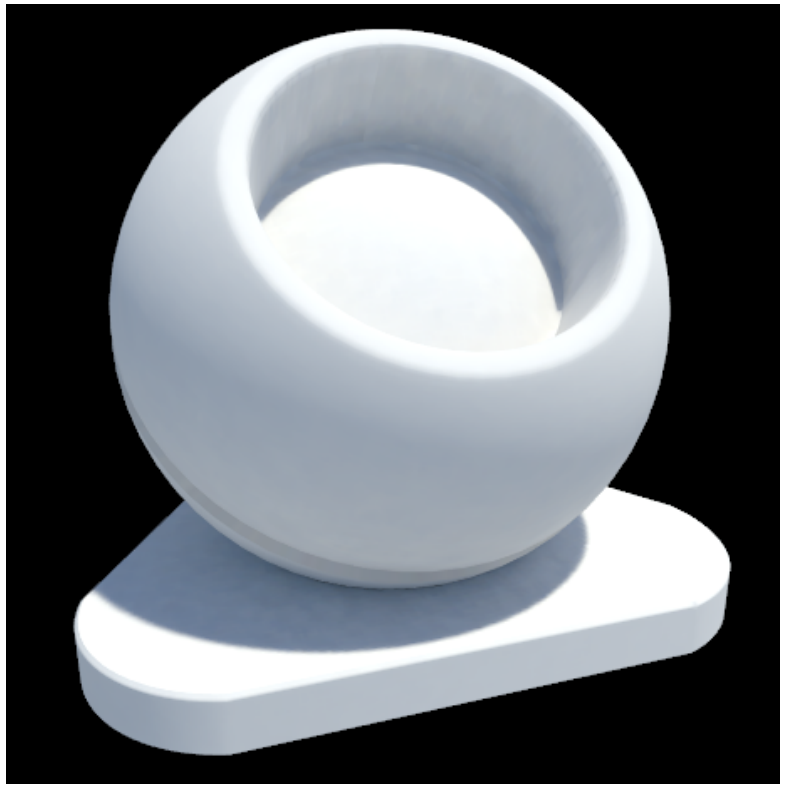
Missing Texels



LIGHT BAKING

Removing some noise with Median Blur





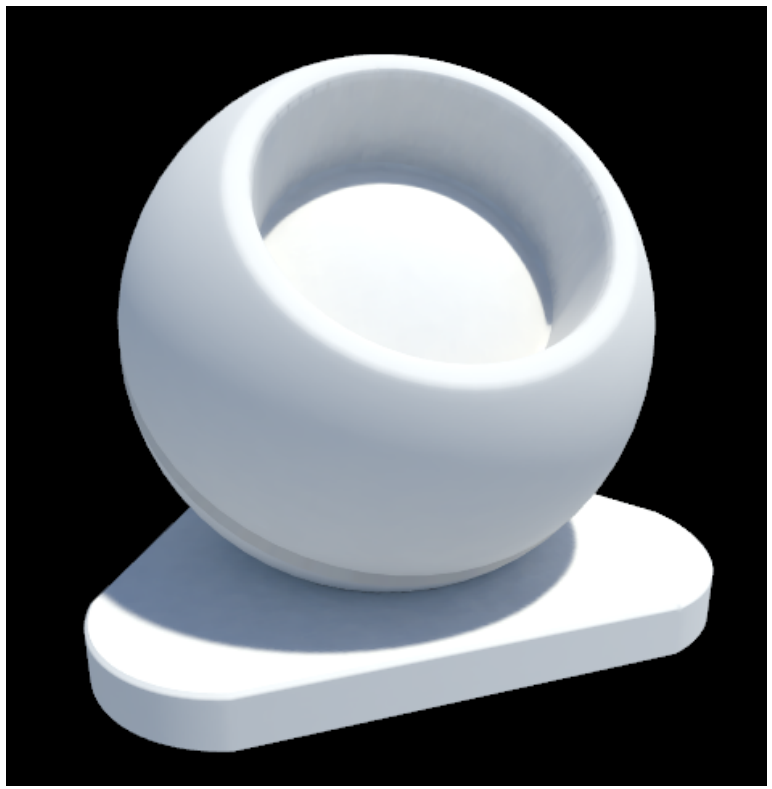
LIGHT BAKING

Speed

512x512, 100 iterations

1,4 seconds

18 million rays/seconds



DEMO

Iray and Live

Iray 2000 iterations

Live @60Hz

EXTRA

LPE

Light Path Expression

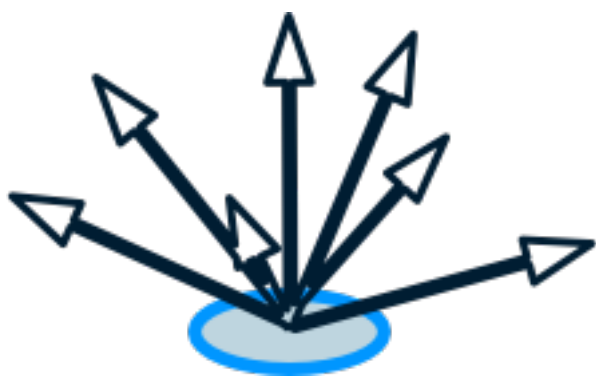
- One light map per group of lights
  - `L.*'light_group1'I` : Path from a light source to the irradiance point, bounce n-times and touch one of the light which is part of group1.
- Avoid objects to be part of the lighting contribution:
  - `L.*[^'obj1']I` : Any path that goes from any light to the irradiance point, but not touching the object obj1.

GLOSSINESS

Angle dependent illumination

Compute the incoming light from dependent direction.

Store the value with the incoming direction and use it for glossy light reflection.



NEXT

Enhancement

Multisampling to cover the entire area of the texel

- A radius to the light probe could be better

QUESTIONS?

THANKS !