

Writing a Light Filter

Light filters are arbitrary shaders that can modify, or filter, the incoming illumination from a given light. Light filters are just regular shaders attached to a light's "filters" parameter. Compared to an old-style RenderMan light shader, the only thing that an Arnold light filter cannot do is modify the light-emitting geometry and its associated sampling.

Arnold currently ships with four built-in light filter shaders, which are explained in detail in the [MtoA docs](#). Here is an example using the built-in gobo shader:

```
image
{
  name myimage
  filename polkadots.jpg
}

gobo
{
  name mygobo
  slidemap mytexture
  rotate 30
  scale_s 2
  scale_t 2
}

spot_light
{
  name myspot
  ...
  filters mygobo
  # you can also attach a stack of filters:
  # filters 3 1 NODE gobo1 gobo2 gobo3
}
```

But you can also write your own light filter. Basically you can use some shaderglobals inputs (such as the light direction `sg->ld`, the distance `sg->ldist`, or, for a spot light, `sg->u/v`) and overwrite the unoccluded light intensity `sg->liu` in-place. Below is a simple example that modulates the light intensity with a Perlin noise procedural texture:

```
#include <ai.h>
#include <strings.h>
AI_SHADER_NODE_EXPORT_METHODS(SimpleLightFilterMethods);
node_parameters { }
node_initialize { }
node_update { }
node_finish { }

shader_evaluate
{
  AtPoint2 p = {sg->u, sg->v};
  sg->liu *= (AiPerlin2(p * 50) + 0.5f);
}

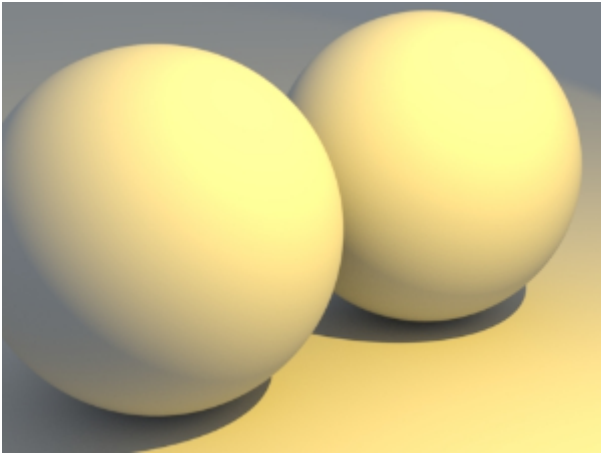
node_loader
{
  if (i > 0)
    return false;
  node->methods = SimpleLightFilterMethods;
  node->output_type = AI_TYPE_RGB;
  node->name = "simple_light_filter";
  node->node_type = AI_NODE_SHADER;
  strcpy(node->version, AI_VERSION);
  return true;
}
```

simple light filter.ass

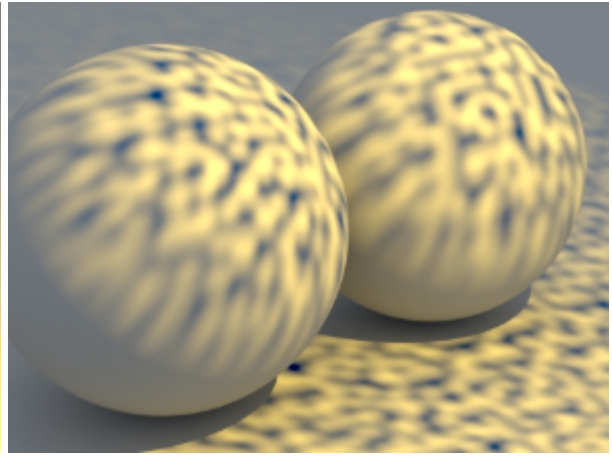
```
options
{
  AA_samples 5
  GI_diffuse_depth 2
  GI_diffuse_samples 3
  background mysky
}
persp_camera
{
  name mycamera
  position 5 3.0 5
  look_at 0 0.8 1
  up 0 1 0
  fov 30
}
sky
{
  name mysky
  color 0.8 0.9 1.0
  intensity 0.5
}
lambert
{
  name mylambert
  Kd 0.5
}
plane
{
  name myplane
  normal 0 1 0
  shader mylambert
}
sphere
{
  name mysphere1
  center 0 1 0
  radius 1
  shader mylambert
}
sphere
{
  name mysphere2
  center 0 1 2
  radius 1
  shader mylambert
}
spot_light
{
  name mylight
  look_at 0 1 0
  position 4 5 1
  intensity 3.141
  color 1.0 0.7 0.1
  cone_angle 65
  penumbra_angle 2
  exposure 6
  filters my_light_filter
}

simple_light_filter
{
  name my_light_filter
}
```

The above example produces the following image:



Without simple light filter



With simple light filter