

VRayLight examples

[Home](#) [VRayLight](#)[Example 1: Single-sided vs Double-sided lights](#)[Example 2: Size of lights, shadows and intensity](#)[Example 3: Real-world lights have inverse-square falloff](#)[Example 4: Skylight, self-illuminated panels and VRayLights](#)[Example 5: IBL \(Image-Based Lighting\)](#)[Example 6: IBL and Caustics](#)

Example 1: Single-sided vs Double-sided lights

This example demonstrates the difference between a single-sided and a double-sided planar area light:



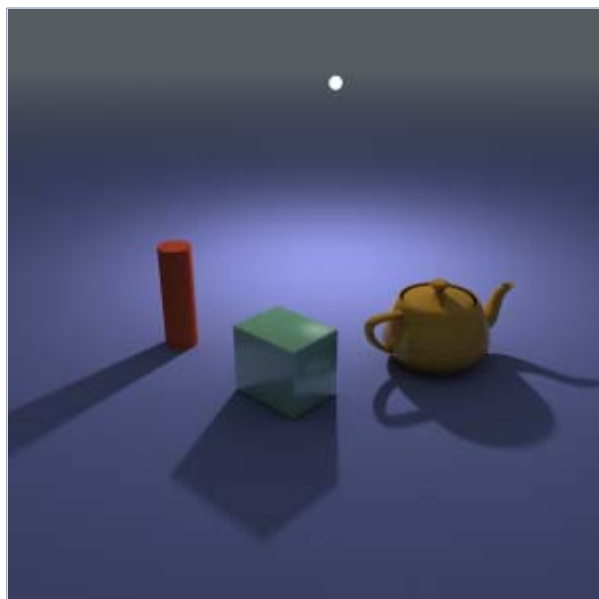
Double-sided is *off*



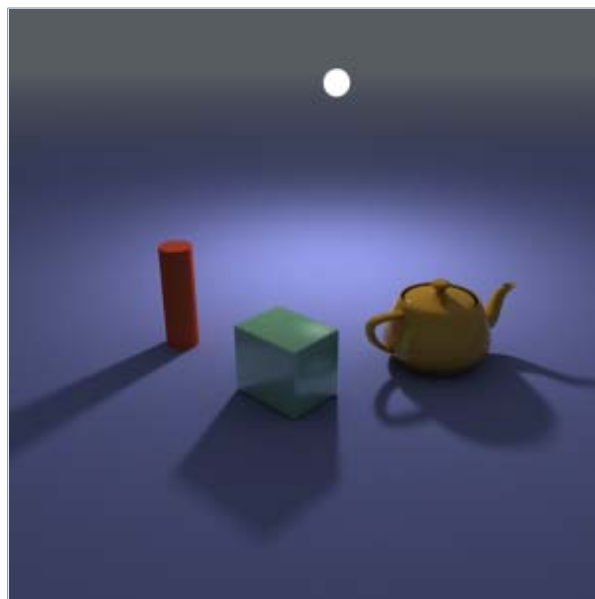
Double-sided is *on*

Example 2: Size of lights, shadows and intensity

The following images show how the size of a light affects shadows. Bigger lights produce blurrier shadows, while smaller lights produce sharper shadows:



U size = 3.0; Normalize intensity is *on*

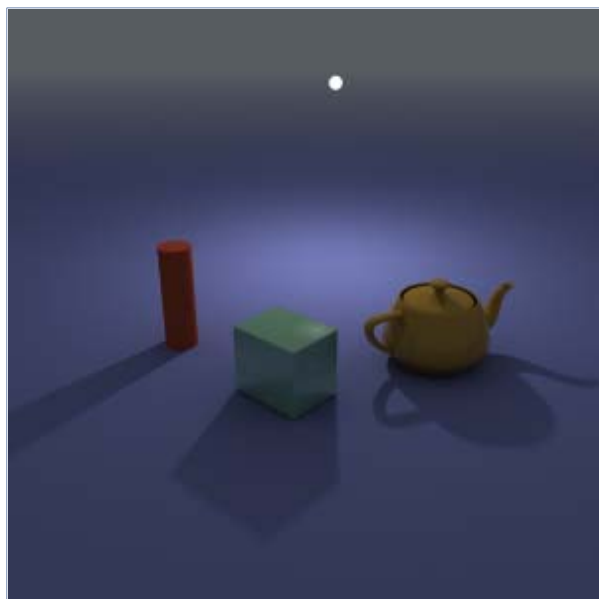


U size = 6.0; Normalize intensity is *on*

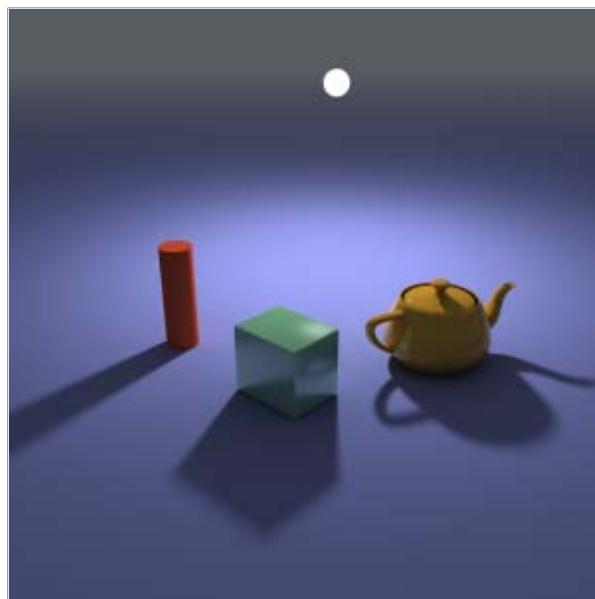


U size

In the above images, the light appears to be of constant intensity. This is because the **Normalize intensity** option was turned **on**. This is what the three images look like with **Normalize intensity off** (the default). Larger luminaires have larger surfaces, which means that they emit more light. Note that the light multiplier had to be adjusted compared to the previous three images in order to produce similar light intensity; however, it is the same multiplier for all three images.



U size = 3.0; Normalize intensity is *off*
(default)



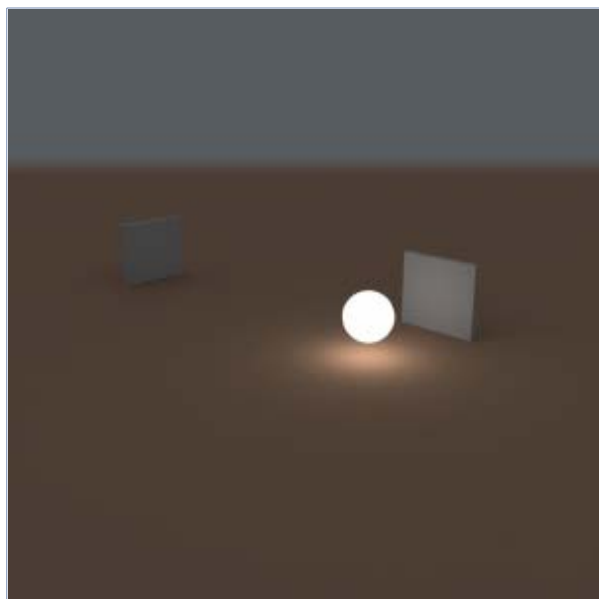
U size = 6.0; Normalize intensity is *off*
(default)



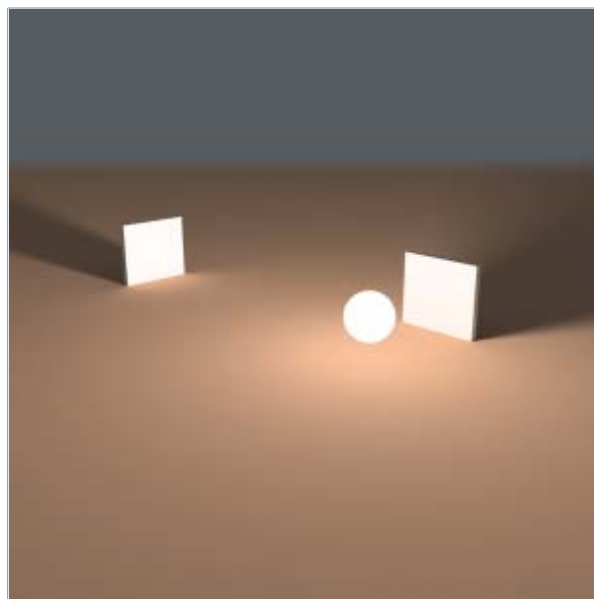
U size

Example 3: Real-world lights have inverse square falloff

The following images demonstrate the **No decay** parameter. In the real world the light sources attenuate with the inverse square of the distance from the light to the shaded surface. However you can disable light decay to achieve behavior similar to the standard 3ds Max lights. The settings for the light source the same for both images with the exception of the **Decay** parameter:



No decay is *off* (default)



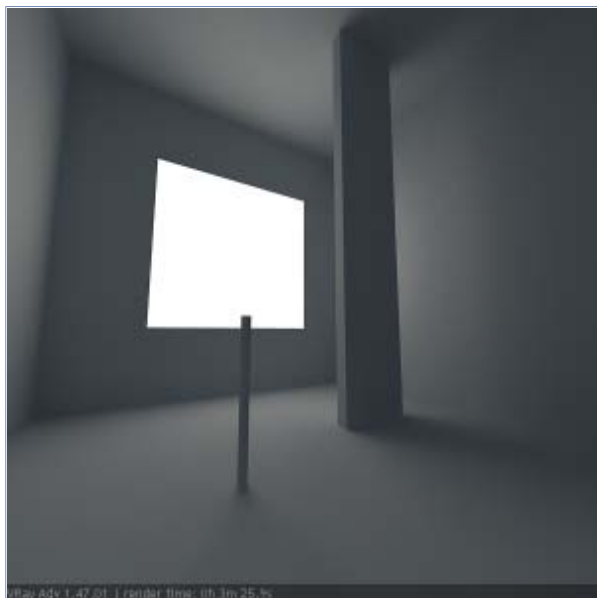
No decay is *on*

Example 4: Skylight, self-illuminated panels and VRayLights

Here is an example of a simple room where the light comes from the environment. The scene was rendered in several different ways:

- with just environment skylight with irradiance map as a primary engine;
- with a self-illuminated panel at the window, again with the irradiance map as a primary engine;
- with a VRayLight at the window and the irradiance map as a primary GI engine;
- with just environment skylight but with brute force GI as a primary engine.

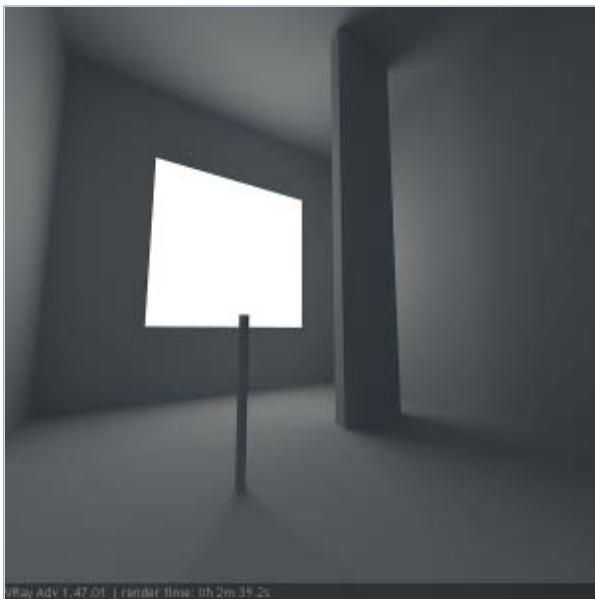
In all cases, the Light cache was used as a secondary GI engine. The environment, the self-illuminated panel, and the VRayLight all have the exact same color and multiplier.



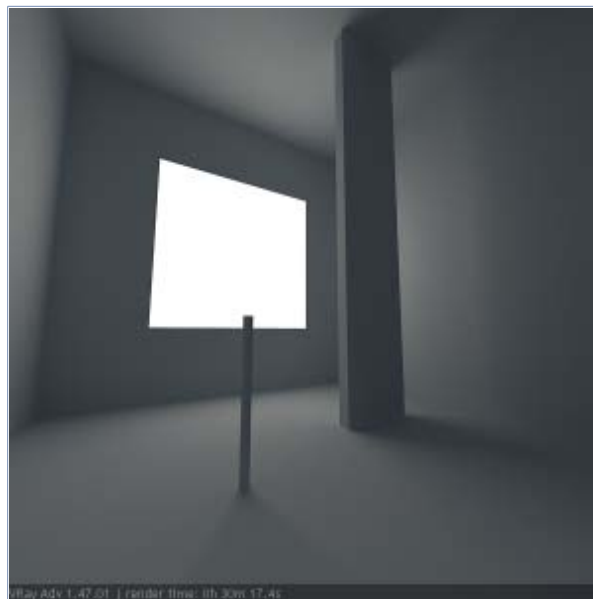
Environment light (skylight) only



Self-illuminated panel at the window



VRayLight at the window



Environment light (skylight) only and brute force GI

As you can see, all methods produce the same light distribution, but there is a difference between render times and quality.

In the first two cases, we rely on the irradiance map to capture the lighting coming from the window. The result is very similar, as well as the rendering times. Since the irradiance map is blurry method, the shadows come out a little blurred. Although we can reduce the blurring by using higher irradiance map settings, this will cost us additional render time.

In the third case, since we use a VRayLight, the shadows come out very sharp and nice, and the rendering time is reduced. This is because the irradiance map was calculated much faster - in the previous two cases, it had to trace a lot of rays to sample the window accurately.

In the fourth case, we used brute force GI instead of the irradiance map. This produces sharp shadows too, since the brute force GI is a non-blurry GI method. However, render time has increased quite a lot.

In this example, using a VRayLight produces the best result in the shortest time. However, if you need to have many lights, this method can become quite slow, since every single light needs to be sampled.

Example 5: IBL (Image-Based Lighting)

IBL (image-based lighting) is a new feature, provided by V-Ray renderer.

The V-Ray dome light has been extended to support arbitrary texture maps that determine the amount of light coming from each direction on the virtual dome hemisphere. V-Ray then uses importance sampling to trace more rays in the directions where most of the light is coming from. This ensures speed and quality that were never before possible with pure gathering GI methods.

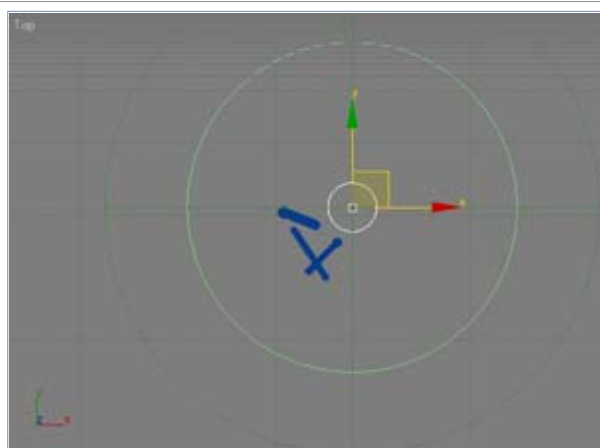
- Easy setup for image-based lighting (IBL);
- Importance sampling of the dome texture for fast and clean results;
- Support for photon mapping for fast and accurate caustic effects.

In the following example we will show how this works with a HDR



image.

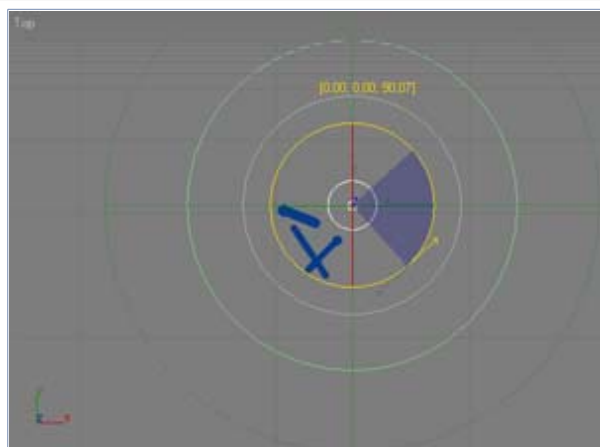
Emit radius: 150.0cm



*Initial position of the dome-light.
Z axis is perpendicular to the ground
plane.*



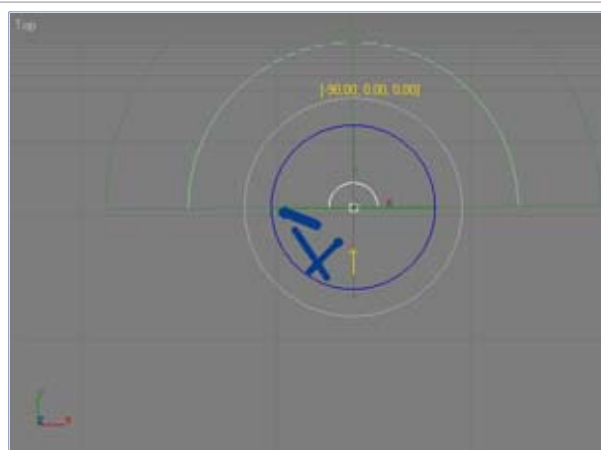
Rendered image



Position 1
*This time the dome is rotated along
its Z axis to 90 degrees. Notice this
has no effect, as the dome is using
the same part of the HDRI.*



Rendered image

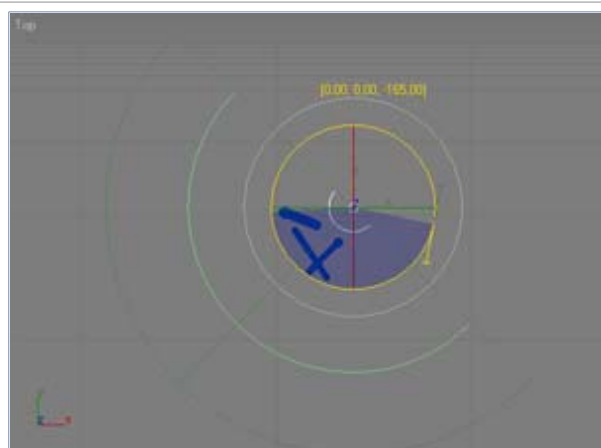


Position 2

*The dome is rotated along its **X** axis to 90 degrees. Now the dome uses another part of the HDRI.*

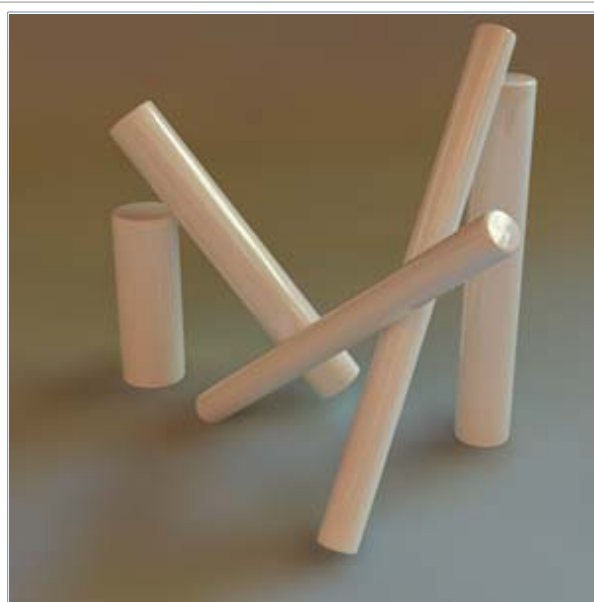


Rendered image

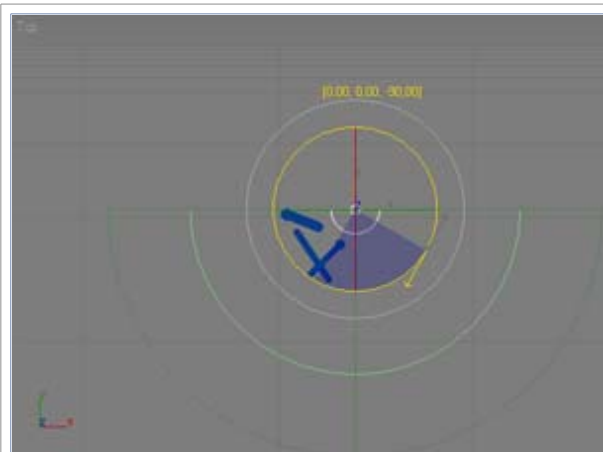


Position 3

*The dome is rotated along its **X** axis to -165 degrees. Notice how the shadows have changed, due to the softer light in the HDRI.*

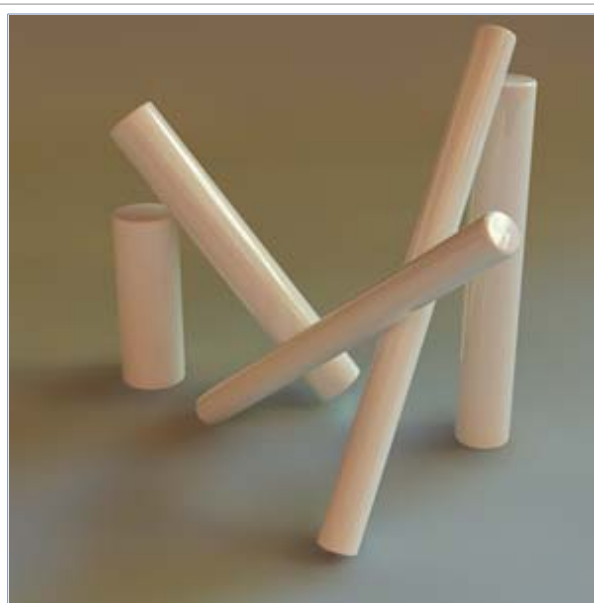


Rendered image



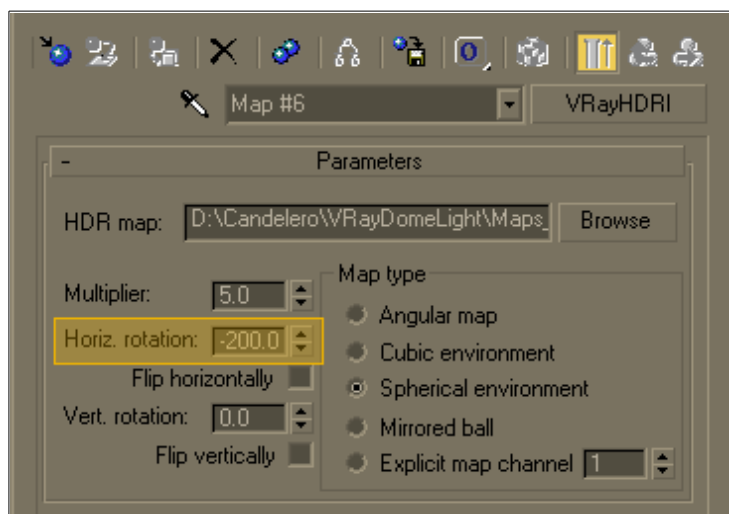
Position 4

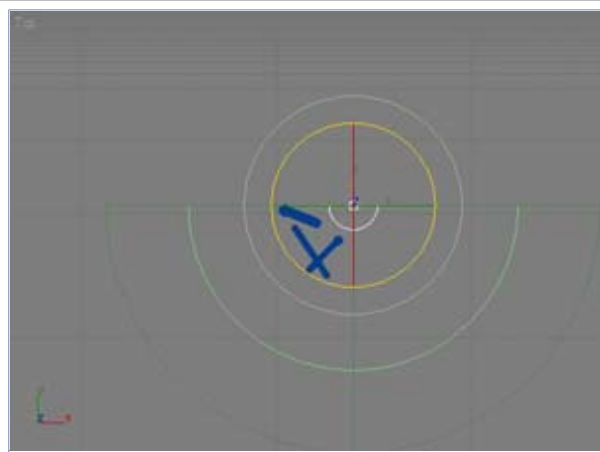
*Another 90 degrees on the **X** axis from the previous position. A slightly sharper shadow has appeared.*



Rendered image

Now we are going to show how the HDRI parameters also can affect the rendered image. We will rotate the HDRI from the material editor.





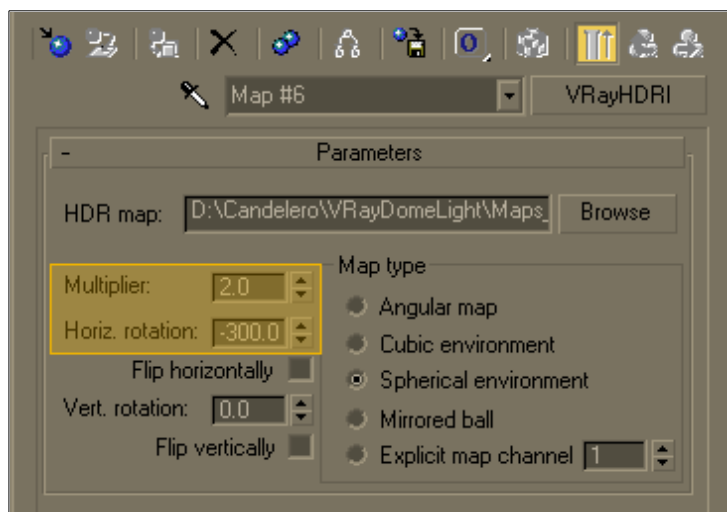
Position 5

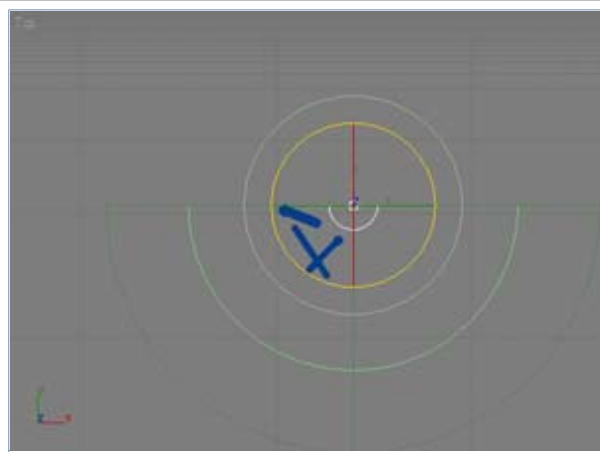
*Rotating the HDRI to **-200** degree. Now the dome obviously uses some of the brightest range from it.*



Rendered image

This time we will reduce the HDRI multiplier and add more rotation.





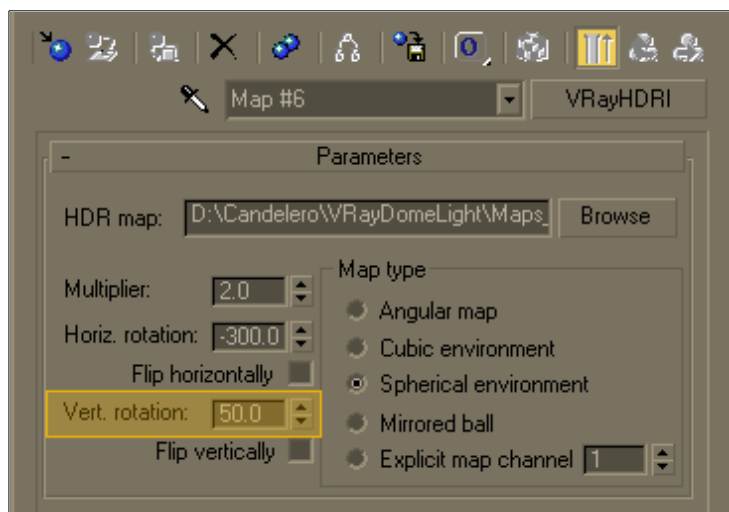
Position 5

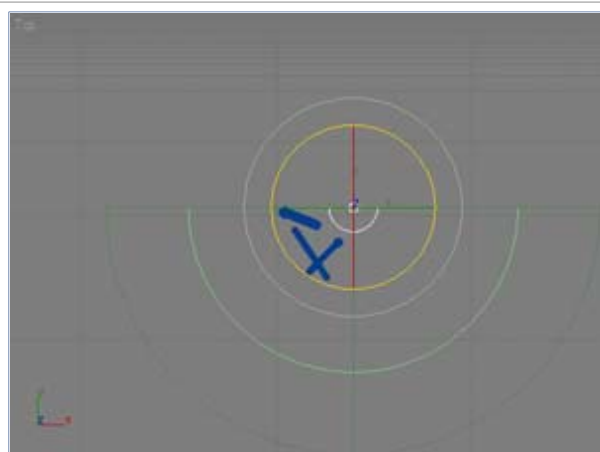
*Rotating the HDRI to **-300** degree. Notice how the multiplier affects the overall scene lighting. We haven't change the dome position.*



Rendered image

We will go further, adding some vertical rotation to the HDRI.





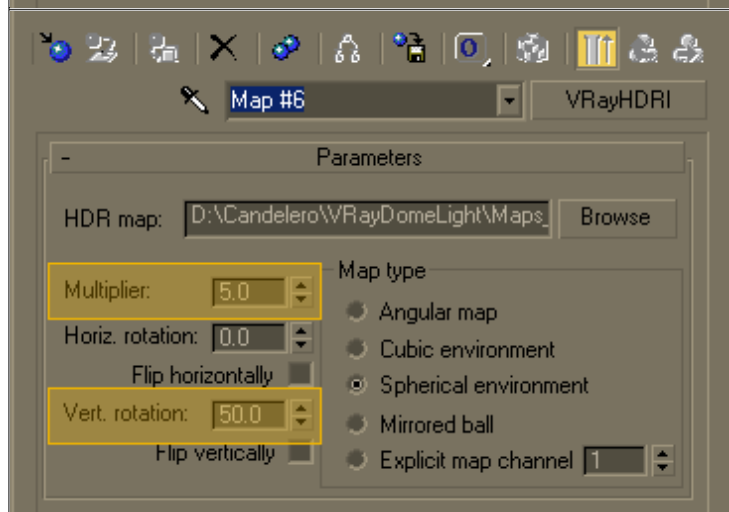
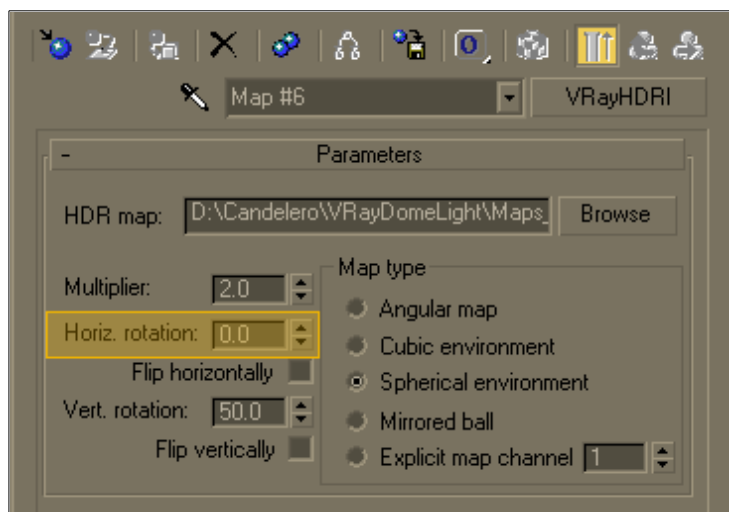
Position 5

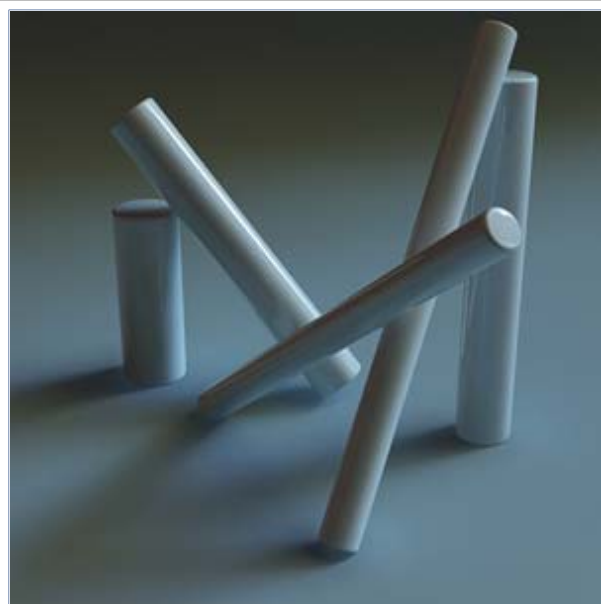
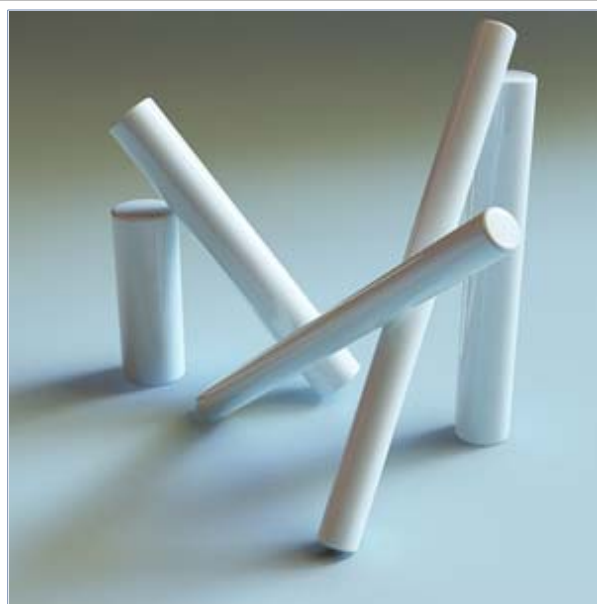
Adding **50** degrees vertical rotation to the HDRI. See how light and shadows have changed a lot. The highlight position is different now.



Rendered image

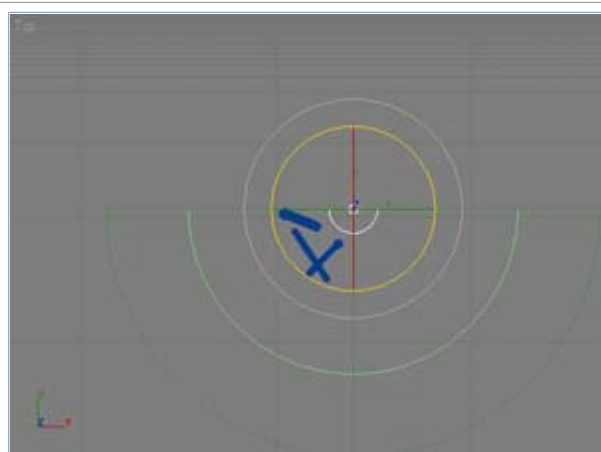
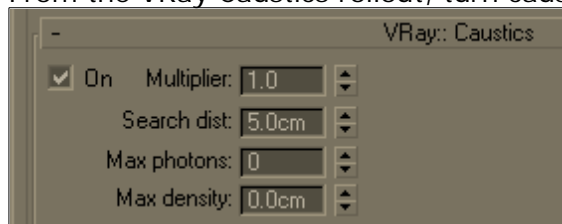
Reset the horizontal rotation to **0.0**. Multipliers: **2.0** and **5.0**



*Rendered image**Rendered image*

Example 6: IBL and Caustics

From the V-Ray Caustics rollout, turn caustics on. Assign a glass material on the cylinders.

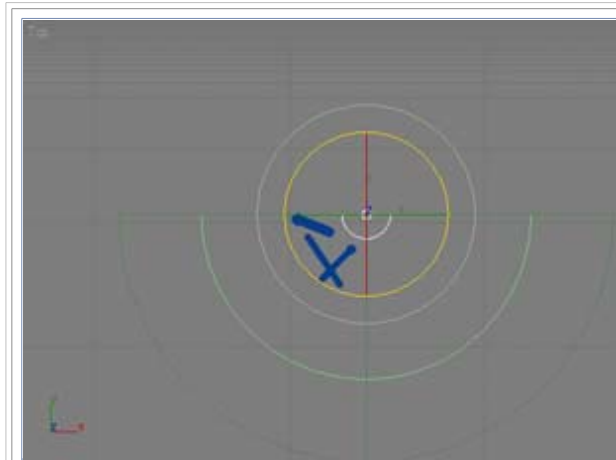


Position 5

We are using the last setup of the HDRI from the Example 5. The caustic effect is almost not visible.

*Rendered image*

We will raise the multiplier to **3.0** and set the search distance to **1.0** unit.



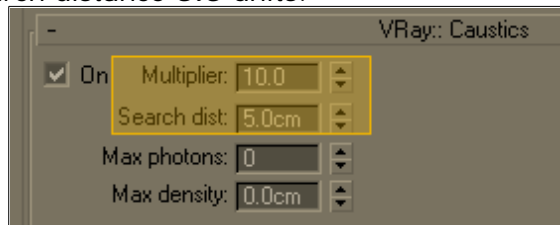
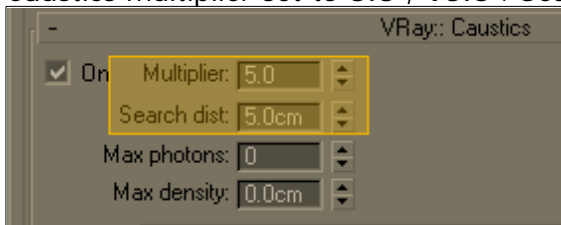
Position 5

Now the effect is visible, though the search distance appear to be too small.



Rendered image

Caustics multiplier set to **5.0** , **10.0** . Search distance **5.0** units.



Rendered image

Rendered image

Other caustics examples.



Rendered image



Rendered image