IBL Light Sources

Image based light creates from a high dynamic range image virtual light sources which the raytracer can see as it can see a single radial or the sun. How the lights are distributed is investigated.

Introduction

Image based light is a means to light a virtual 3D scene with natural light. To do so, a photograph is needed that has everything around, above and below it covered: a fully spherical panorama. Additionally, not only the environment and the colours must be captured, also the light.

Obviously, the sun in the sky is brighter than a street lamp and such a one is brighter than the pavement or a meadow in the shadow of a tree. The difference in light is so big that no apparatus can catch it fully. Besides, neither can the eye, it needs some time to adapt to the darkness of a room stepped into from the sunny outside. The brightness range the eye can handle is limited — but this limit is greatly extended by the eye's ability to adapt.

High dynamic range images can be created by adaption as well. Several pictures are taken from the same location with different exposure times. The picture taken with the longest exposure time has the shadowy parts nicely lit but the blue sky with the clouds is completely white. The shortest exposure has the bright white clouds and the sun nicely lit but the ground remains black, even where the sun shines on it. By combining a series of such exposures, the full — or at least a large part — of the brightness levels can be recorded.

Such a huge range from dark to bright cannot be displayed on any medium, whether on paper or on a computer screen. Neither could the eye look at such a picture. The dynamic range must be reduced by tone-mapping. We do not discuss this extensive topic here.

Rather, we concentrate on image base lighting; that is what such a high dynamic range spherical panorama is for.

Converting the image to light sources

Many 3D programs can use high dynamic range images (HDRI) for image based lighting (IBL), Bryce is among them since version 6.0. The light intensities contained in the image must be converted so that the program can use them. There are several algorithms that can do this. Bryce 6.0 used the Monte Carlo algorithm like some other 3D programs do. Unfortunately, the implementation in Bryce 6.0 was flawed.

Median-cut algorithm

From Bryce 6.1 hence, the median-cut algorithm is used to create the light sources. In a nutshell, the image is divided into two parts with different sizes according to the brightness in these two parts. Then, each part is again divided in the same manner and so forth until the limit set is reached. In Bryce, this limit is set by *Quality* and each quality step is double than the previous one because all existing rectangles have to be divided, doubling the amount of rectangles and hence point light sources. The more light sources, the better the brightness and colour information in the picture are represented.

Each rectangle gets one point light source like a radial in Bryce. Each light source has the same brightness (diffuse) and moving the *HDRI Effect* slider controls all light sources at the same time. Bright parts of the picture have many small rectangles with a light source close together so that more light shines from this position. The shadow under the tree is assigned a large rectangle with one light source and hence less light shines from there. The colour for each light source is taken from the average colour in the rectangle in which centre it sits.

Finding the lights in Bryce

When the HDRI is loaded from the IBL tab, Bryce creates the lights. What is shown is a linearly tone-mapped representation of the HDRI. There is no direct means to make the point lights visible. The lights created from the image work like any radial light in Bryce, i.e. it shines in all directions but the light source is never visible. There is no wire frame representation for the IB lights.

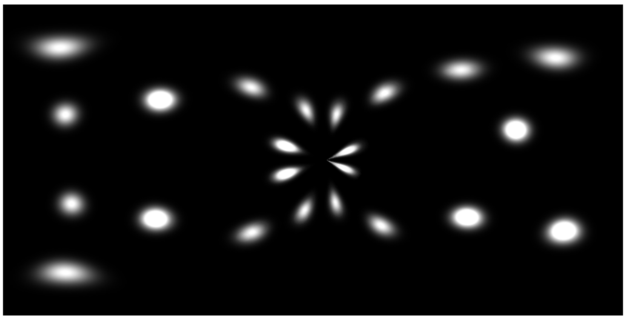
However, a point light creates a specular highlight on an object lit by it. The colour of the *Specular Halo* for the material on the object controls the size of this bright spot. The specular highlight is smallest if the *Specular Halo* colour is fully black.

The specular output from an HDRI is notoriously a bit low. If the HDRI is not used as background, *Intensity* that controls the brightness of the background can be set to any amount desired; and if *Apply to light source* is enabled, *Specularity* gets boosted like diffuse (*HDRI Effect*) gets amplified.

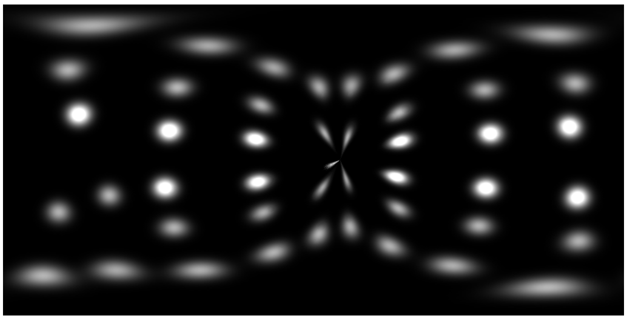
Counting the lights

The *Quality* setting represents the number of rectangles and hence point light sources used. The lowest number is *16*, the next double, *32*, then *64* up to *4096*. This does not seem to be correct. At *Quality 16*, we can count 23 lights, at *Quality 32* we find 46 lights, at *64* I counted 71 and at *Quality 128* only 116 but this last count may be wrong. The specular highlights are rather large and if two lights are close together, only one can be discerned.

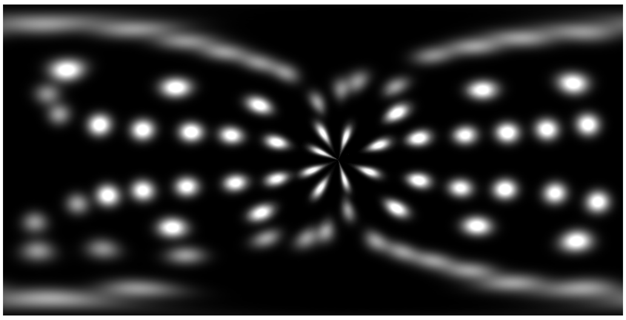
A sphere was rendered like a mirror ball but the sphere was not set as a mirror for the environment but as a mirror for the specular blobs. The rendered sphere was transformed from the "mirror ball" to the spherical projection. The back side of a mirror ball has very few information and I moved that part to the centre. The lights are visible, though distorted. The HDRI used was a uniformly white sphere.



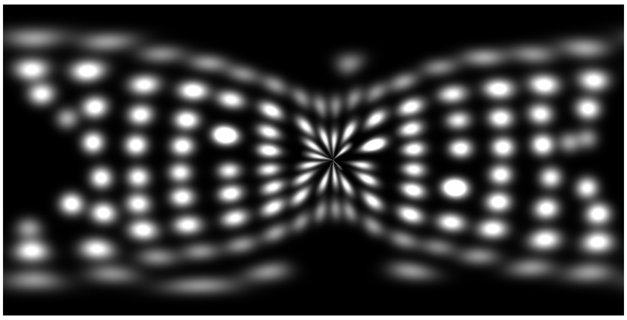
Quality 16, 23 lights



Quality 32, 46 lights

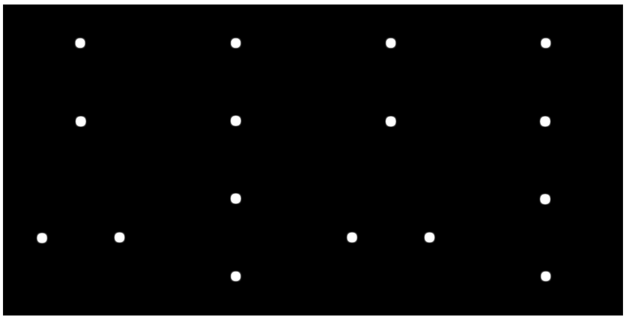


Quality 64, 71 lights

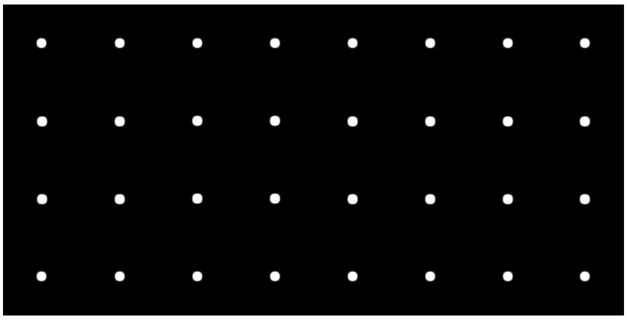


Quality 128, 116 lights

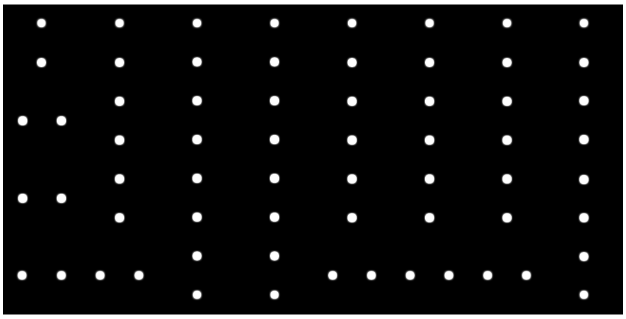
The distribution of the lights is not as we would expect, it is too irregular. Below, the lights were generated by an external median-cut algorithm from the same HDRI.



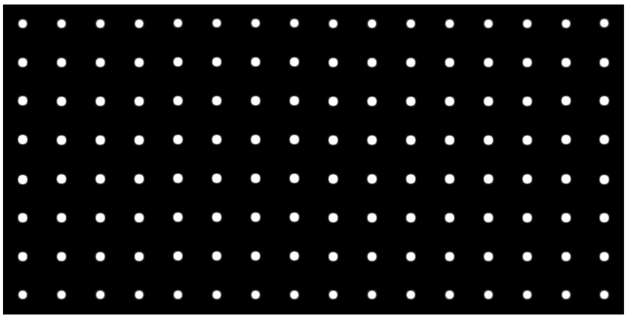
16 lights



32 lights

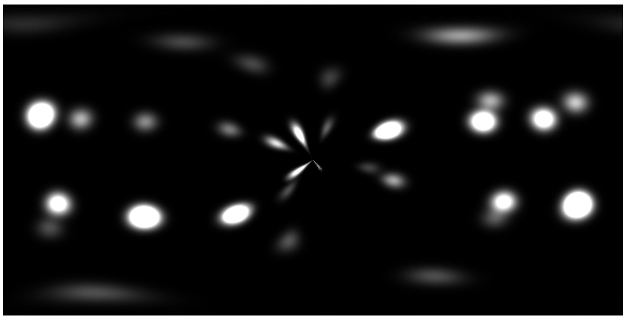


64 lights

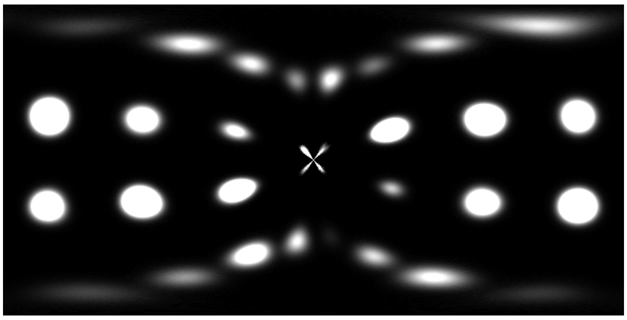


128 lights

The distribution of 16 and 64 lights is a bit unexpected but the number is correct. These lights can also be used in Bryce as an HDRI as well.



Median-cut 32, Quality 32; 32 lights



Median-cut 32, Quality 4096; 32 lights

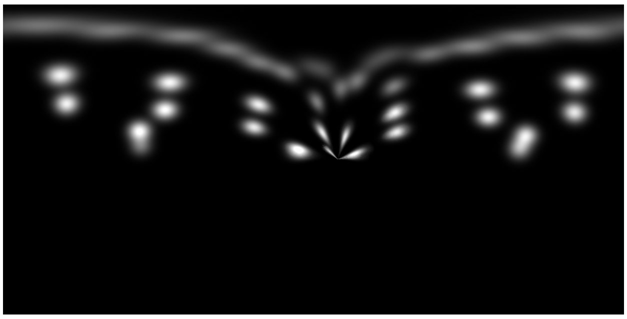
Comparing the 32 lights generated by an external median-cut algorithm used in Bryce as HDRI and the lights, we discover they are quite different at first glance. If we consider the errors introduced by transforming a mirror ball into the spherical projection makes the difference smaller. Particularly so if we compare the render with the maximum quality of 4096 we notice that there are always four lights vertically and eight rows and we can consider the result accurate.

When using quality 32 the lights generated in Bryce are only approximately at the same location as in the original. For the light (diffuse), this will hardly make a difference since the light spreads out from the radials created from the image.

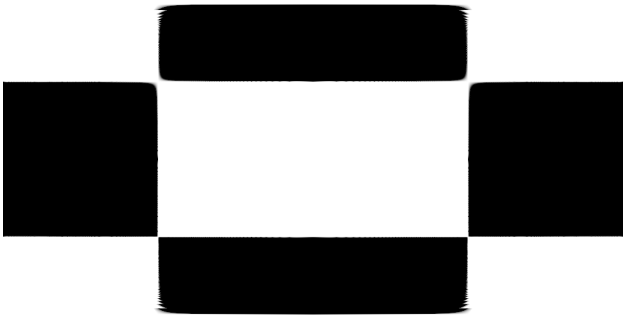
Distribution of non-uniform HDRI

The above is all a bit academic; though a uniform white HDRI gives very nice omnidirectional light to illuminate an object. By the way, we have not — and will not — discuss the area lights created from an HDRI for true ambience (TA) use.

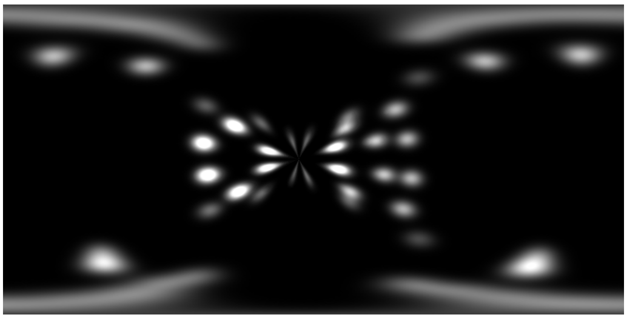
Hemisphere HDRI



Hemisphere Quality 64



Checker HDRI



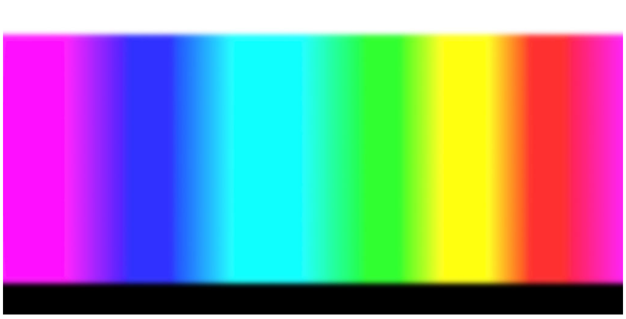
Checker Quality 64

Two black and white HDRIs, above the upper half from zenith to horizon is lit, the ground is black. The lights shown are distributed accordingly. The light distribution for the checkerboard also seems quite right. On the hemisphere, I count 37 lights, which is about half of the 71 lights counted for a fully white HDRI rendered with 64 lights.

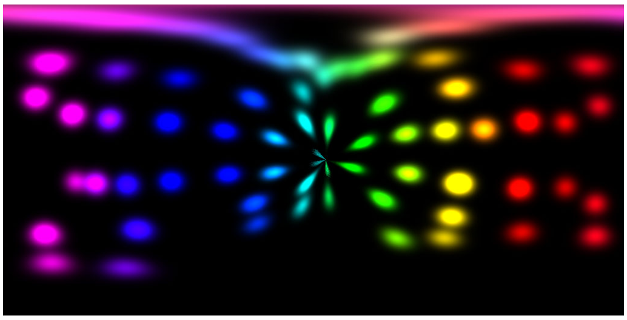
The checkerboard somehow evades counting because of the stretching on the four fully white corners. It could be one light at each corner, but there could be more.

Mirrored

What has not been considered is that the light sources shown as specular highlights on the sphere are actually mirrored. This did not matter up to now because symmetrical HDRIs were used. This is different for the following example of a coloured rainbow HDRI.



Rainbow HDRI (mirrored)



Quality 64, 64 lights

The zenith is fully white, the nadir fully black, the rest is rainbow coloured. I count 64 lights, though there is some uncertainty. We miss a white light on top but the absence of a light at the bottom is as expected. Please take the distortion due to the mirror ball to spherical projection in account.

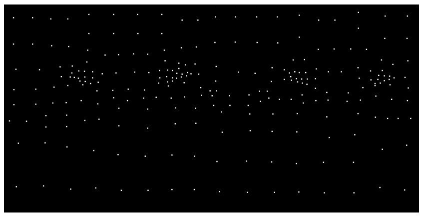
Bryce flips the HDRI horizontally when it loads the HDRI so that the backdrop looks right. The camera is inside the HDRI sphere. If an LDRI of the HDRI is used on a sphere, it looks correct from the outside and if the camera is within the sphere, the LDRI must be mirrored.

Three examples

The method of counting specular dots is unreliable because if several lights are packed close together, we cannot properly count the lights. Here are some typical examples.

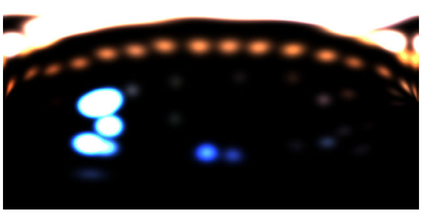
HDRI used

Quality 64

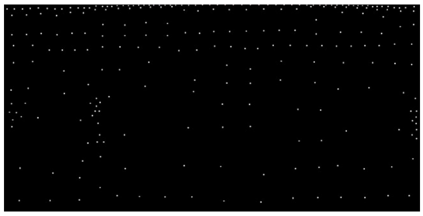


External median-cut 256 lights

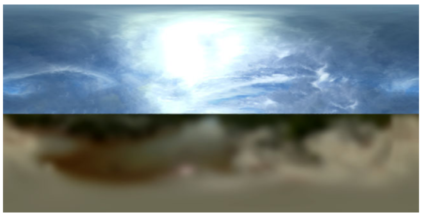
HDRI used



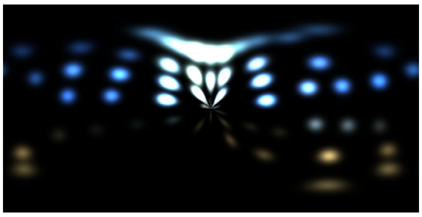
Quality 64



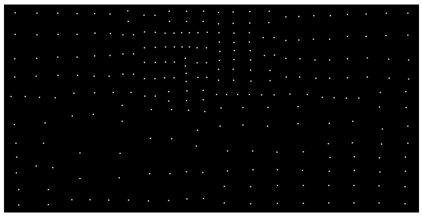
External median-cut 256 lights



HDRI used



Quality 64



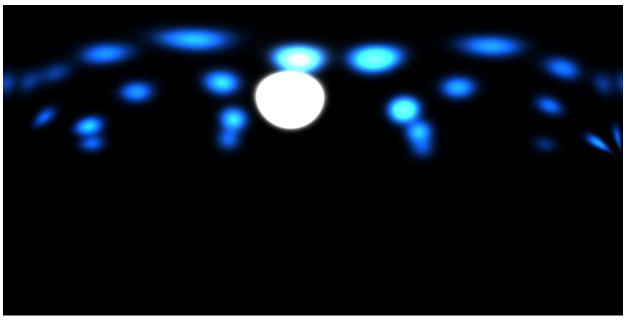
External median-cut 256 lights

The distribution of the point light sources can be nicely observed on the image made with an external median-cut algorithm (zoom to see them better; the images are embedded larger than shown here).

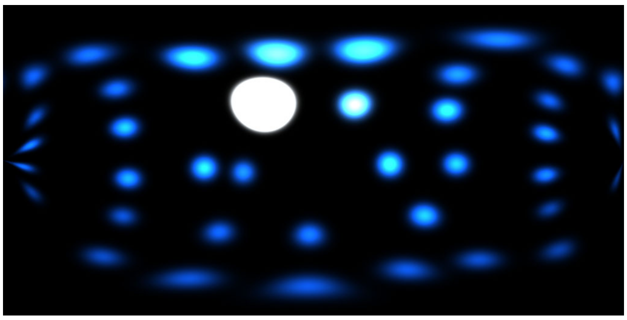
The Bryce renders are mirrored to match the other pictures. For the middle row, *Specularity* was greatly boosted to make also the less bright lights visible. In the lower row, note that sun behind the clouds is behind the sphere where the resolution is lowest. The render with the lights was moved by 180° and mirrored so it matches the picture.

HDRI from Sky

An HDRI can be directly created from the sky. There are two options: the full sphere or the *Sky Dome only*. What is the difference concerning the number of lights? As a base, the *Lazy Afternoon* sky (installed, Daytime, the first) was used but *Ambient* was set to white and *Sky Dome* to black. *Sun/Moon Shadows* set to 100%, no haze, fog or clouds, sun *Diffuse* to 500, the rest unchanged.



Sky Dome only, Quality 32, 26 lights

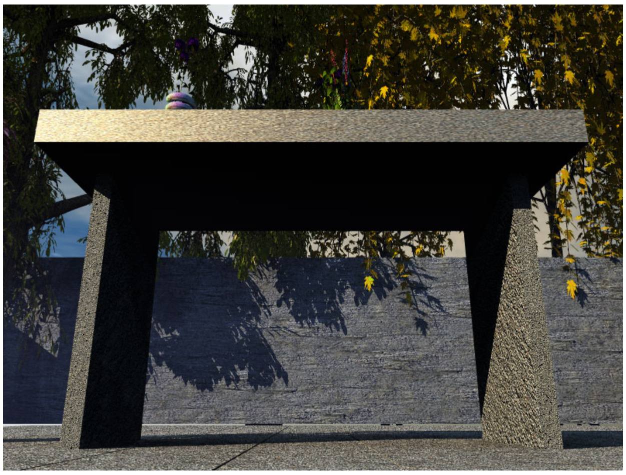


Full sky, Quality 32, 39 lights

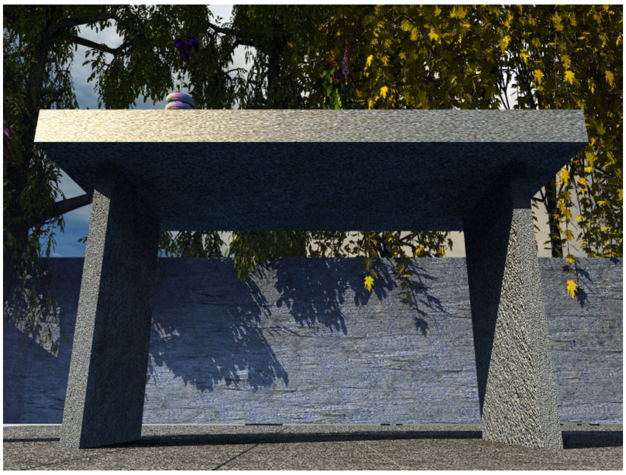
The idea to use the *Sky Dome only* is to have double as much lights above the horizon than if the full sky is used. The comparison above shows that this is a myth and I apologise for being guilty of having spread this myth.

On second thought, however, this is not surprising. The median-cut algorithm works on the full HDRI and if only the sky dome is used, all lights below the horizon are removed. Those redundant lights cannot just be distributed above the horizon because there were never rectangles defined for them.

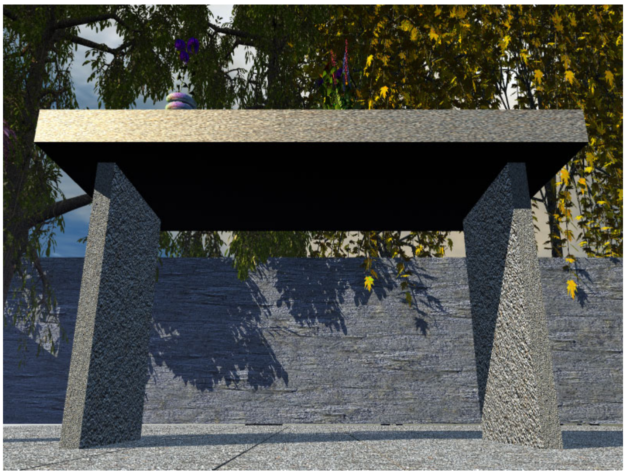
There is no harm using the sky dome only, but neither is there any advantage. If the ground which divides the parts above and below the horizon is made either (partly) transparent or set to no shadow casting (as in the examples below), the IB light can shine up from below and emulate the light reflected from the ground to the underside of an object.



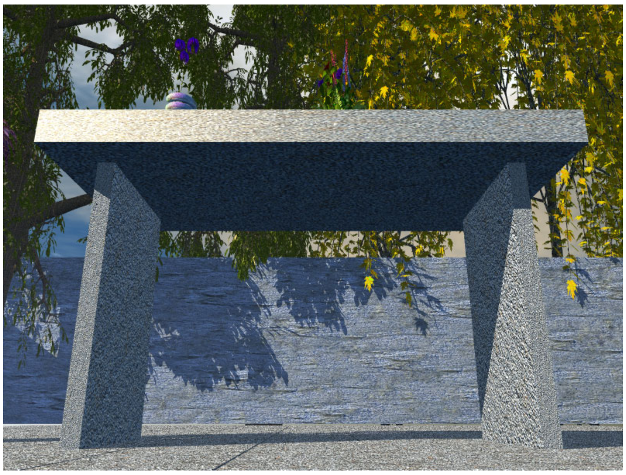
Quality 64, Sky Dome only, 100%



Quality 64, full sky, 184%



Quality 64, Sky Dome only, no shadows 11%



Quality 64, full sky, no shadows, 12%

The difference of using the sky dome only and the full sky is obvious. Even if IBL shadows are disabled, the underside of the table remains black if only the sky dome is used. The percentages are the render times. The *Saturation* was kept quite high at 85% for all four renders to make the effect of the ambient light from the HDRI more obvious. If the IBL shadows are disabled, *HDRI Effect* should be reduced. This was not done on purpose for the examples above.

December 2014/horo