"***HDRI For Artists***" Siggraph 2008 – mental ray HDRI FAQ
By
Håkan "Zap" Andersson, mental images
version 4.1: April 25, 2008

General mental ray questions

What is/are the primary reasons I should use mental ray over my 3D applications default renderer?

While traditionally, computer graphics has lived in a simple universe where light went from 0 (black) to 1 (white), entire mental ray lighting pipeline is unclamped floating point, i.e. it supports colors outside the traditional "black" to "white" range and therefore deals very well natively with high dynamic range data.

Furthermore, mental ray can natively and in a physically correct manner illuminate a scene or an object solely from the HDRI data. This is done by using the global illumination techniques "Final Gathering" or "Irradiance Particles", or specific IBL (Image Based Lighting) shaders designed for the purpose.

What is Global Illumination and why is it important?

"Global Illumination" is a catch-all term for any illumination technique that simulates light that is not coming directly from a light source, i.e. that has bounced a couple of times in the scene. This light is generally known as "indirect light".

What are photons? Why should I care?

Photons are one of the global illumination techniques in mental ray. Photons start at the light and propagate through the scene with the help of special "photon shaders", and gets stored at surfaces. For historical reasons, the photons are sometimes named "global illumination" in some documentation or application UI's. One should be aware that this is a bit of misnomer.

While Photons *can* be used together with HDRI, they are not the primary tool, because one would have to use a special photon emission shader to emit photons in a pattern that represent the HDRI lighting, which is difficult to do efficiently and with any quality.

What is Final Gather? Why should I care?

Final Gathering (henceforth known as FG) is a different global illumination technique that starts from the shaded point and works its way *backwards* in the scene by sampling a quasi-random set of rays above the shaded point. FG can very efficiently be *combined* with Photons, wherein FG only handles the “last bounce” of light.

FG is also responsible for sampling the *environment*, where is where you would normally put your HDRI map. Hence, FG is generally the primary technique to light an object based on HDRI.

What are Irradiance Particles and why should I care?

Irradiance Particles (henceforth known as IP) are a new global illumination technique in mental ray. Being new, this technique is not integrated (as of the typing of this FAQ) in any shipping product.
From the standpoint of HDRI, IP’s are very similar to FG, but they have one advantage, and this is that they perform an *importance sampling* of the environment, i.e. they will send more samples in the direction of the bright parts of the environment map.

In a nutshell, how do I use HDRI with mental ray?

The basic process is very simple, but varies a little bit depending if you are using a standalone mental ray, or the mental ray built into Autodesk 3ds Max, Autodesk Maya or Softimage XSI.

The basic steps are as follows:

- Add your HDRI as the global mental ray *scene environment* (also known as the *camera environment*, since the attribute is actually connected to the mental ray *camera*)
- Enable Final Gathering (or Irradiance Particles)
- Make sure the environment is “seen” by FG (or IP)

The last step is application specific, and will be covered in detail below.

How does FG really work, and how do I tune it?

FG tries to figure out how much indirect light is arriving at a point by sending a set of quasi-random rays into the surrounding (which may be other objects, or just the environment map), and averaging (integrating) the result over the hemisphere above the shaded point. This is known as a Quasi-Monte-Carlo integration. The result of such an integration inherently contains some noise, but less noise when more samples (rays) are used.

Since calculating FG for every (sub) pixel rendered would be prohibitively expensive and noisy, FG is instead created in a pre-pass, in which set of “FG points” are created in the scene. At render time, information is interpolated between these points to smooth out the result. (You can see these FG points as green dots in the image if you turn on the FG diagnostics mode).

There are three main parameters guiding the FG algorithm: Density, Accuracy (= number of rays), and Interpolation.

The **Density** sets how close together the FG points are in the image. The higher the number, the finer detail can be resolved in the final image, but slower. Lower numbers tend to be very fast, but lack detail.

The **Interpolation** is how many FG points are weighted to generate the final (smoothed) result. Lower means more detail, showing more of the individual FG points, which may be noisy. Higher means less detail, blending more FG points together, to give a smoother result.

The **Accuracy** (aka “number of rays”) is how many rays are sent out to sample the scene to create the data in one FG point.

- A low-frequency (slowly changing contrast, low detail) environment need fewer FG rays (lower **Accuracy**), and can hence render faster. This can be as low as 50 rays.
- A high-frequency (fast changing high contrast, high detail) need more FG rays (higher **Accuracy**) and can hence render slower. This can be towards the thousands of rays for exceptional cases.

This is due to the fact that if the environment contains many small very bright areas, it is unlikely for rays to hit them, and your result will be noisy if you do not send many (thousands of) rays into it, which may be very slow. This means that, a *smooth* (i.e. slowly changing with low-frequency contrast) environment yields much less noise and needs far fewer rays than a busy (quickly changing high frequency contrast) environment.

So, some quick rules of thumb:

- Want it faster? Decrease the accuracy or the density
- Want it smoother? Increase the accuracy or the interpolation
- Want it more detailed? Increase the density or decrease interpolation

You quickly realize this is the traditional “faster / cheaper / better – pick two” triangle. Luckily, there is a trick to get the “best of all three” with a bit of black magic known as “ambient occlusion”.

By using a "decent" Accuracy (number of rays) (100-250), a fairly low density (0.1-0.5), and a fairly high interpolation (30-50), you get a result that is representative of the indirect lighting, but is far too smooth (i.e. lacking detail).

By combining this with a short-range ambient occlusion render (or use the ambient occlusion built into the mia_material(_x) shader), you can get these details back in a visually pleasing (albeit not strictly "accurate") way.

What is the Final Gather Filter and when do I use it?

If your HDR is high contrast, you can get a very noisy FG result if you do not use the FG filter at all (None/Off/0). Unless your HDR is very smooth, you most likely need to set the filter to at least 1 ("Standard"). One rarely needs to go higher than 2.

Is there a way to avoid the FG interpolation completely?

Yes. This is known as the "force" mode. For this, Density and Interpolation ceases to have any meaning, since no pre-pass and no interpolation is done at all, only the Accuracy (number of rays) matter.

Should I pre-blur my HDR's to avoid having to use an excessive number of FG rays?

When using the HDR for lighting through FG, and if you are after efficiency (i.e. fast renders) and plan to do some animation (i.e. want to avoid frame-to-frame flickering) – then yes.

Since the FG rays tries to sample an entire hemisphere, the "theoretically ideal result" is the perfect blurring (convolution) of all light in that hemisphere.

But you can get nearly identical *apparent lighting* from an environment map that has been pre-blurred into a low-frequency environment map and sampled with *very few* FG rays as sampling the original high-frequency environment map with *very many FG rays!*

You can use programs like ***SmartIBL*** or ***HDRShop*** to do what is called a "diffuse convolution" function (for efficiency, this is generally done on a vastly scaled down copy of the full HDR). The result of doing this is much better than trying something like a Gaussian blur in Photoshop, because it will wrap around the edges correctly, as well as across the "poles" of the spherical map.

How can I use a high detail HDR for reflections, AND get efficient lighting using FG?

This is where you would use two <u>different</u> HDR's – one for *lighting* (which is visible to FG rays) and one for *reflections* (that are visible to any other rays such as reflections, refractions, etc.)

A low resolution HDR used for lighting is prepared as in the previous question (i.e. a diffuse convolution from SmartIBL or HDRShop) and the full resolution HDR is used for reflections.

Some applications already have the "split" between environments used for lighting and environment used for reflections/refractions built into the user interface. This is discussed below.

In mental ray itself, you can use the mip_rayswitch shader from the production library. Put mip_rayswitch in the camera environment, and put your low-rez "pre-blurred" HDR map in mip_rayswitch's "finalgather" input, and your regular high-rez in all the other inputs.

What are some great mental ray shaders to use with HDRI?

There are several useful shaders from various shader libraries:

From the "Production" shader library:

- mip_rayswitch: Allows splitting between different HDR's for lighting and reflections
- mip_rayswitch_environment: Allows showing a different background image to the surrounding environment
- mip_matteshadow: Allows integration of CG objects by e.g. casting shadows onto the background imagery etc.

From the "Architectural" shader library:

- mia_material(_x) : General, physically correct high performance material shader
- mia_envblur: Utility shader to perform efficient glossy reflections of environments

More details in the following questions.

How can I avoid having my HDR come out blurry in the background?

By not using it as the *background*, only as the *environment*. You can do this most easily by using the mip_rayswitch_environment. This shader has two inputs, one, called “background”, is used only for rays that come directly from the camera and goes straight (without being bent by reflections or refractions), and one, called “environment”, that is used for every other ray.

When applied to the scene environment, the practical result becomes that the input named “background” will show what you see “straight ahead” directly through the virtual camera lens, whereas the input named “environment” is what you see in reflections, refractions, and what will light the scene.

The same technique is used when you’d want a background at all in your final image – i.e. want it to be suitable for compositing. You still use mip_rayswitch_environment, but you set the “background” input to transparent black (0 0 0 0).

How can I have my CG objects cast shadows onto my HDR / background?

This is what the mip_matteshadow shader does. Explaining all the details of this shader is beyond the scope of this FAQ, but its main purpose is to introduce shadows, reflections and bounce light between CG objects and a background environment that is represented as a photograph (which could be HDR).

The basic principle is that you create “stand in objects” for the real world objects present in the background plate, like a ground plane, and whatever else is necessary. These objects use the mip_matteshadow shader as their surface shader, and have the background image mapped (often using mip_cameramap to do proper projective mapping) to it’s “background” input. You then add whatever new “synthetic” objects you may want to introduce into the scene.

Since the stand-in objects are supposed to “represent” objects that already exists in a photographic background plate, and such a background plate already contains any inter-object reflections, inter-object shadows, and inter-object bounce-light transfer, it is important for mip_matteshadow to handle this correctly; while the new synthetic objects should cast shadows onto the stand-in objects, and a stand-in object should cast a shadow onto the synthetic object, the stand-in objects should not cast shadows on each other (since these already exist in the background plate) … same goes for reflections etc. And mip_matteshadow all handles this correctly, and, for the most part, automatically.

Can I render out layers even if I use HDR for my lighting?

This depends on the application, but the general answer is “yes”. One would have to take into account that what for most other lights would be considered “specular highlights” will from an HDR be considered “reflections”, and what most other lights would consider “diffuse” in the HDR environment case is considered “indirect illumination”.

Can I use an HDR and also use a normal directional light? (or other standard 3D lights?)

Yes, and as a matter of fact this is encouraged. FG cannot resolve strong directional components in the HDR map (IP can, to an extent) and hence a HDR map of the sunlit sky will not generate a high quality rendering with a strong solar shadow.

It is much more efficient to remove the super-bright sun-disk from the HDR (at least from the HDR used for *lighting with FG*) and use a traditional CG light in place of the sunlight.

In general, lighting that covers a small area in the HDR (i.e. subtends a small solid angle on the environment map) are more efficiently handled with “real” light sources.

Naturally, some tools (like the Maya IBL) exist to convert the HDR into “real lights”.

Do I have to raytrace to use HDRI?

It is not required to use ray traced *reflections* per se, but for HDR lighting to even begin to look realistic, some form of occlusion probing (generally done through the Ambient Occlusion shader, which does a form of raytracing) is necessary.

Besides, almost all meaningful rendering effects require some form of raytracing, and mental ray is a *very fast raytracer*, so it’s not like back in 1980 when you had to run for the hills every time someone said the word “raytracing”….

Can I just use the hdr for lighting and not for reflections?
Can I just use the hdr for reflections and light it myself?

Yes and Yes. Using a technique like e.g. mip_rayswitch, this can easily be accomplished.

Also, by only lighting "traditionally" and not having FG turned on at all, the scene will not be lit by the HDR at all, but it will still reflect it.

What are the most important things I need to know in order to set up my mental ray scene and use and hdr to light with?

When rendering with HDR's involved (actually, any time you render with physical intensities outside the traditional 0-1 range) your scene will probably end up Very Bright – i.e. your render is a HDR in and of itself.

And to view a HDR on a LDR computer monitor, you must use some form of tone mapping (and proper gamma correction) to the final image so you can see it (otherwise it becomes an overexposed wash of white).

You can either choose to do render-time tone mapping through e.g. the lens shader mia_exposure_photographic, or, alternatively, you can render to a HDR image format (such as EXR) and do the tone mapping in post.

In my quality presets, should I use "Preview Final Gather" as my default setting for using HDRs with mental ray?

When you render FG, mental ray will calculate the FG in a pre-pass. There is an option to "see" this. It looks like a grungy low-rez render, but it allows you to quickly judge if your image will turn out okay or not. It is a good idea to use this!

There is a different, older "preview" method for FG, which one should NOT use. It skips the prepass, and renders the buckets individually. This will yield tile artifacts between the buckets, and is really only useful as a quick draft-test render mode. This method is largely superseded by the aforementioned mode.

Which multi-pixel filtering method is best for mr?

Avoid the "box" filter except for tests. For video, the "Gaussian" is often best. For stills, either of the sharpening filters ("Mitchell" or "Lanczos") tends to be preferred.

What does final gather accuracy do and mean in simple terms?

It is the number of rays shot into the hemisphere to try to resolve the indirect lighting hitting a point.

Can I use the Physical Sun and Sky instead of using and HDR? Why or why not?

Yes, the Physical Sky is a mathematical simulation of the day sky. It can be used instead of, or in combination with a HDR. Since the Physical Sky is inherently smooth (no high frequency content or discontinuities) it is very suitable for lighting via FG. One can still (with the help of e.g. mip_rayswitch) use a HDR for reflections.

Mental ray shaders – What is the DGS shader? Why and when should I use it?

It is a legacy shader which you should not really use these days. I suggest looking at the mia_material instead, which does all DGS does and more, much more.

Mr shaders – what are the mia material shaders and when should I use mia_material_x instead of mia_material?

The mia_material is a generic hard-surface material shader that is physically correct and high performance. You should generally use this for pretty much anything that isn't soft or mushy.

The mia_material and mia_material_x runs the same shader code, it's just slightly different interfaces where the former only exists for compatibility reasons. In general, you would use mia_material_x.

When I use surfaces with glossy reflections with my HDR's I get noisy renderings, what can I do?

For this exact reason, the shader mia_envblur (from the architectural library) exists. It's a shader that wraps around your existing environment shader, and can do efficient mip-mapped blurring of it on demand. By coupling this with mia_material(_x) (from the same library), the shaders can talk to each other in such a way that the glossiness of an environment lookup is communicated to the environment map, and it returns properly pre-filtered values. NOTE: The mia_material(_x) should have it's "single_env_sample" option turned on for this to work.

What does the mip_mirrorball shader do?

It is a simple one-step unwrapper. Instead of taking a photo of a mirror ball and unwrapping it in an external application, and taking the pain to align it properly in 3d space such that all reflections point in the right way, and trying to hide any artifacts on the back of the ball… the mip_mirrorball converts this all to a simple one-step process.

Simply shoot a background photo, and then simply shoot the mirror ball photo (in HDR, preferably, but even an LDR shot can get you quite far), and simply crop the image to the edges of the sphere. When this cropped image is used in mip_mirrorball, the shader automatically unwraps it for the given camera angle; i.e. it assumes that the correct angle to use is the one of the rendering camera (this is why your background shot should be taken from the same angle). So the shader is only useful as long as the CG camera position does not change (at least not much).

How do I get ambient occlusion to work with my HDR? I want to have both when I render my images.

You can use the built in ambient occlusion of the mia_material(_x) for a "detail enhancement" effect. You can also render it separately.

If one do not want to use FG at all, a similar effect can be obtained by using a diffuse convoluted copy of the HDR environment, looked up by the surface normal, in the "bright" input of the ambient occlusion shader. This yields a very similar "global light effect" to what FG would have yielded in the same scene, but without any actual bounce-light occurring, just occlusion.

Application Specific Questions

How do I do HDR lighting in Autodesk 3ds Max; and what is the 3ds Max "Skylight"?

To use any HDR lighting in 3ds Max you must have a "Skylight" in the scene. If you do not have a "Skylight" in the scene, the environment will not light the scene at all.

The "Skylight" has two options:

- Use Scene Environment
- Sky Color (with an additional "Map" slot)

If you only have one HDR and want to light the scene with it as well as see it in reflections, you will simply add your HDR as the global "Environment" map (in the "Environment" dialog). You must then add a "Skylight" and set it to "Use Scene Environment", enable FG, and render.

If you also have a different low-rez HDR for lighting, you have two options:

- Keep the "Skylight" in "Use Scene Environment" mode, but use the mip_rayswitch shader (as described in a previous question) in the scene environment instead.

OR

- Set the "Skylight" to "Sky Color" mode, and apply the low-rez HDR in the "Map" slot directly in the Skylight itself.

The two methods are for all practical purposes functionally equivalent, it's just a matter of if the switching happens inside the max built in environment shading functionality, or inside mip_rayswitch. The end result is the same.

To easier apply the techniques from the rest of this FAQ, though, the 1st method is preferred, because it more mimics that "standard mental ray behavior".

To create a 3ds Max "Skylight":

- From the "Create" panel choose "Lights", choose among the "Standard" lights (not "Photometric") and pick the "Skylight"
- Insert into the scene
- Go to the "Modify" panel
- Apply "Use Scene Environment" or "Sky Color" as applicable.

How do I use HDR lighting in Autodesk Maya? What is the Maya "IBL node"?

When using HDR lighting in Maya, there are a couple of different ways to do it:

- Putting your environment (using an appropriate lookup node, for example mib_lookup_spherical or any of the Maya environment lookup nodes) as the mental ray camera environment, and render with FG

OR

- Creating a Maya "IBL" node

The former method strictly uses FG and "standard mental ray behavior". The second method uses a Maya-specific shader for Image Based Lighting, the "IBL Node", which in itself has a whole bunch of additional features and options.

To add an environment to the camera:

- Select the camera
- On its AE template, open 'mental ray' (<u>not</u> "Environment"!)
- In "Environment Shader", add an appropriate lookup shader for the type of HDR you have (e.g. mib_lookup_spherical for a spherical map, etc.)

To add an IBL node:

- Open the "Render Settings" dialog
- In the dialog, open up the "Environment" section
- Beside "Image Based Lighting" click "Create"
- On the "Common" tab, turn OFF "Enable Default Light"

Once the IBL node has been created, here are some tips:

- Feel free to scale the IBL sphere up as needed to more easily view your 3D objects... there is no actually geometry, so this will not affect the results of the rendering.
- To adjust the overall brightness of the scene/HDR, you can adjust the "Color Gain" to control the amount of light given by the HDR image.

The IBL also has the "Emit Light" button. When this is on, FG is not used to create the lighting, but a set of "pseudo-lights" that are automatically set up based on the intensity of the HDR. This yields more defined shadows, but generally takes longer to render.

How do I use HDR lighting in Softimage XSI?

When using HDR lighting in XSI one generally uses the built in “Environment” node:

- From “Render” menu choose “Edit Current Pass”
- In the dialog choose the “Pass Shaders” Tab
- In the “Environment” section click “Add”
- Add the “Environment” shader

In closing….

What are some good websites where I can find more mental ray information?

I can suggest

- http://mentalraytips.blogspot.com, my “*mental ray tips*” blog
- http://www.MyMentalRay.com, a user community site
- http://www.LAmrUG.org, Los Angeles mental ray Users Group
- www.mentalimages.com