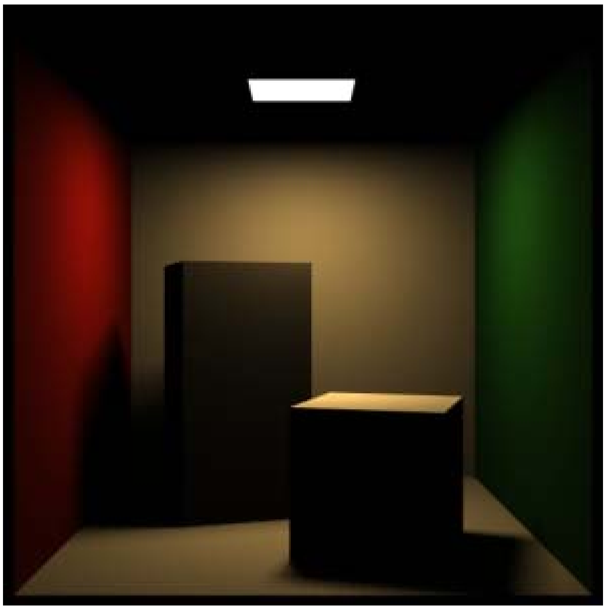
link: http://www.ypoart.com/tutorials/photon/index.php

Introduction to Photon Mapping

Ray tracing is a fast rendering technique developed several years ago which have been used by talented artists to produce several stunning pictures.

Compared to scan-line renderer, ray tracing is a huge improvement because it can render objects image reflections in mirror-like surfaces.



Cornell box rendered with ray tracing

One drawback of ray tracing, however, is that it cannot render indirect illumination. Indirect illumination comes from light reflected from other objects around. Therefore, any part of an object which is not directly lit by a light source or which is obscured by another object is rendered black.

To help compensate for the missing indirect illumination, ambient illumination value was added to ray tracing. Ambient lighting, however, is the same everywhere in a scene or on each models which tends to produce flat shadings. Clever shadows computing algorithm, such as the one found in A:M when you set the darkness property of your lights to anything less than 100%, can fake indirect illumination even better.

Better faking technique of indirect illumination have been developed by artists who understood the interaction between light and surfaces. Among those techniques is the extensive use of secondary lights, even negative lights, placed at critical locations within the scene in order to simulate indirect lighting. This technique eventually developed into the skylight technique which is essentially an array of light designed to simulate indirect light coming from a sky (see skylights). Skylights, however, cannot be used in an indoor scene setting and still cannot fake inter-reflection from other objects.

Today, there exist several rendering techniques which can compute indirect lighting. However, because those techniques take the indirect illumination into account, they are all much slower than ray-tracing alone. Imagine a scene where there is not only a few lights but thousands even millions of lights. Anyone who have used a skylight at least once will immediately have a feeling of the huge time that such a scene would take to render. This is, in essence, what a typical global illumination renderer tries to achieve.

Cornell box rendered with Photon Mapping

But don't get discouraged yet. Among those techniques, Photon Mapping is one of the most efficient algorithms which can render illumination effects like indirect illumination and caustics.

And A:M uses an implementation of the Photon Mapping technique for those effects. With the current implementation, the Photon Mapping may be used to render indirect illumination or caustics but not both at the same time although it is possible to produce an image with both at the same time by compositing two renders: one with the indirect illumination setting and the other with the caustics.

Warning! long render time ahead!

This said, I guess I will never warn often enough, even though it is one of the most efficient algorithm, it is nevertheless much slower than ray tracing. So if you are trying to cut render time by every means, then photon mapping is probably not for you.

Photon Mapping and the other renderers

Ray tracing

Ray tracing works by shooting rays from the eye through each pixels and determine the color of the first surface hit by the ray by computing how it is illuminated by each lights in the scene. With ray-tracing, an object receives light only where its surface sees the light directly and is non obstructed from the light source by other objects. Because ray tracing starts from the eye and not from the lights, it can't compute ***light*** reflection. It can only compute ***image*** reflection. There have been several adaptations of the ray-tracing rendering engine to compute indirect illumination coming from the other objects in the scene, but those ray-tracing engine are typically excessively slow and produce excessive noise in the final render.

Radiosity

Radiosity includes the first algorithms that where developed to compute direct and indirect illumination. Those algorithm worked by subdividing the whole scene in smaller patches until each patch had a uniform illumination. The illumination was then computed between each two patches in the scene starting from the lights.

There are two main shortcomings to the radiosity type of algorithms. First they can only compute direct and indirect illumination on rough surfaces. Radiosity cannot render glossy, reflective or transparent refractive surfaces nor caustics caused by those surfaces. Second, it works only on objects which can easily be subdivided into smaller and smaller patches thus excluding any procedurally defined objects. And, of course, storing the information about lighting information between evert two patches in the scene required huge memory requirements as well as long processing time.

Global Illumination

Global Illumination, or GI for short, refers to the class of algorithms which can potentially render all known illumination phenomena, e.g. direct and indirect illumination falling on and coming from rough, dull, glossy, reflective, transparent and translucent surfaces and can also produce caustics.

All GI algorithms are based on Monte-Carlo estimations. That is they use large number of random samples in order to compute an estimate of a solution which is as close as possible to reality. And this is the main disadvantage of most of the GI algorithms. Because it uses random samples, it produces noisy renders unless a huge number of samples is used. So it is not uncommon to have GI renders compute billions of samples and illumination interactions for a render thus resulting in days, even weeks of render time.

Biased GI

GI takes so much time because it tries to compute an exact solution through the use of thousand of samples per pixels, each of those samples can bounce several tens of times in the scene. Exact solutions are called unbiased estimates. They may be interesting for research purposes but not very interesting for entertainment purposes where render time is of primary concern.

Biased GI algorithms have been developed which does not try to compute an exact solution but rather an approximate or good enough solution. Biased GI algorithms still use Monte-Carlo sampling but with much less samples and fills in the rest of the estimate through statistical techniques. Those algorithms usually work in two passes where the first pass computes a rough estimate of the global illumination and the second pass uses a step called final gathering where the illumination at each pixel is estimated by sampling the previously computed rough global illumination. The rough global illumination estimate works because it is never viewed directly as is the case for unbiased GI.

Photon Mapping Fundamentals

Photon Mapping is currently one of the fastest algorithms available for simulating Global Illumination. It is a Monte-Carlo based technique, which computes a huge part of the global illumination solution by a rough but acceptable statistical estimation instead of the usual attempt to compute the most exact solution as possible. Another advantage of the Photon Mapping technique is that it uses the already available ray-tracing rendering engine although not exclusively for ray-tracing.

The Photon Mapping technique was developed by Henrik Wann Jensen and the first published papers discussing the technique appeared in 1995. For anyone interested in better understanding the technique, Jensen have published the book "Realistic Image Synthesis Using Photon Mapping".

Photon Mapping is a two-pass algorithm, which takes a mixed approach, and adds two very clever twists.

Shoot and store photons

In the first pass, photons are shot from the light into the scene. They are bounced around interacting with all the types of surfaces they encounter. The first twist is that instead of redoing those same computations over and over, a few thousand of time for each pixels, the photons are stored only once in a special data structure called a photon map for later reuse. The second twist is that instead of trying to completely fill the whole scene with billions of photons, a few thousands to a million photons are sparsely stored and the rest is statistically estimated from the density of the stored photons.

The Photon Map is not an image file as its name may imply but a hierarchical 3D spatial search acceleration structure called a kd-tree.

After all the photons have been stored in the map, a statistical estimate of the illumination at each photon is computed.

Gather illumination

In the second pass, direct illumination is computed just like regular Ray-Tracing but the indirect illumination, which comes from the walls and other objects around, is computed from querying the stored photons in the photon map. When rendering indirect illumination, the two passes use a Monte-Carlo estimation although, in some very particular cases, the second step could dispensed from the Monte-Carlo estimation. When rendering caustics, the second step renders the photon map directly and thus don't use the Monte-Carlo estimation.

When rendering indirect illumination, the second pass uses a Monte-Carlo estimation technique known as the final gathering because, at each pixel, several rays are shot from the first hit point into the environment. At each secondary hit, the photon map is queried in order to gather the illumination coming from the objects around in the environment.

The final gathering technique is not exclusive to photon mapping though. It is used in the majority of the multi-pass GI algorithms.

Photon mapping parameters.

Because the Photon Mapping technique uses two Monte-Carlo estimations, it is inherently much slower than the ray-tracing alone. But how much slower depends on the setting of the different parameters controlling the two passes of the algorithm, mainly parameters which controls the number of samples to use in the Monte-Carlo estimate. For the first pass, this means the number of photons in the scene and for the second pass, it means the number of Monte-Carlo samplings.

In addition, the statistical estimate of the illumination at each photon is performed with a density estimation. Here, the number of photon samples is the most important parameter that will affect the render time.

Roughly, the lower those parameters, the fastest the render and vice-versa. However, too low values will produce too rough estimates and thus too many artefacts in the render.

The Monte-Carlo Estimation Technique

Here about this Monte-Carlo buzz word. Ideally, an exact evaluation of the global illumination value at a given pixel in a given scene would require the evaluation of several integrals. But because the integrals that we find in a typical 3D scene are not analytically evaluatable, we have to resort to an approximate evaluation technique, which is called Monte-Carlo integration.

Please sit on the chair...

The name Monte-Carlo comes from the fact that it is really driven by chance. Suppose you have two large dices one numbered from 1 to 360 gives you an azimuth value and the other numbered from 1 to 180 gives you an elevation value. You throw the two dices and then record the environment as seen through a pin-hole in the direction suggested by the dices.

If you repeated this often enough, you will eventually have a good idea of what the whole environment looks like. Obviously, repeating this process only a few times will not give you a good idea of the environment. Especially, if each time you repeated it, the pin-hole oriented toward the sky, then you could very well conclude wrongly that the whole environment is blue.

So it is best to test the scene at many many sample points in order to acquire a statistically plausible estimate of the integral. Obviously, the more samples you take, the higher probability you have that the estimate reflects the true value. Exactly the same idea applies to any Monte-Carlo based estimation technique including photon mapping. The more samples you take, the more exact the estimate.

How Monte-Carlo is used

The main difference between the Photon Mapping and the other Monte-Carlo GI rendering algorithms, since they are both Monte-Carlo based, is in the way they estimate the global illumination.

Traditional Monte-Carlo GI typically launches thousands of rays through each pixels. Each of those rays bounces around in the scene, accumulating the interactions between the rays and the surfaces as they go, until it reaches a light.

There is also another class of Monte-Carlo GI algorithm which does the exact opposite which is shooting photons from the lights, bouncing around in the scene until it reach the camera through each pixels a few thousand times. As can be expected, those algorithms takes just as long to render as one another. The main advantage here is that since the photons are shot from the lights, those algorithms can compute caustics while the previous class of algorithm can't.

Yet another class of Monte-Carlo GI algorithm is called multi-pass. They try to use the best from both the previous classes of GI algorithms. Photon Mapping is a two-pass algorithm, which takes this mixed approach, and adds two very clever twists.

The impact of Monte-Carlo on render time

Just for comparison, if you set Multi-Pass ON in A:M with only one pass, you get the time it takes to render the scene with only one sample per pixel and roughly 1 bounce per ray. This results in 300 thousand light-surface interaction computations for a 640x480 image. Shooting one thousand rays through each pixels would take one thousand time longer.

Now suppose that you turn soft shadows ON by setting the number of rays to 30 which is, BTW, also a Monte-Carlo estimation technique used to estimate soft shadows. You now have 30 samples in the environment per pixels and thus, you already have 9 million light surface interaction to compute which takes 30 times longer to render.

But wait! there's more. In a typical ray-tracing algorithm, rays are only reflected off reflective surfaces while in global illumination algorithms, rays are reflected off every surfaces. Not only those surface where some reflectivity is set. This is precisely what produces the global illumination effect. So you can easily imagine that a typical Monte-Carlo scene render can take several hours to several days to render even several weeks for some architectural simulations.

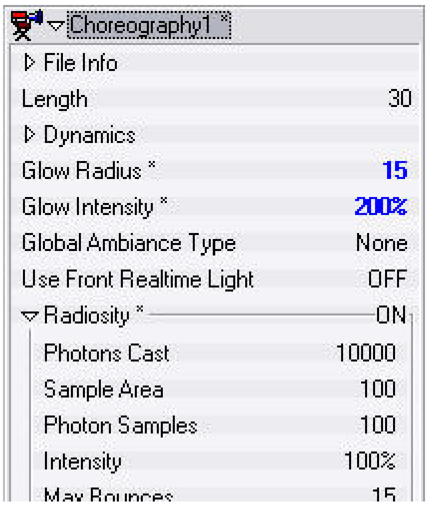
Assuming a 640x480 image with 1000 samples per pixels and an average of 20 bounces per samples, this results un 6 billion light-surface interaction computations. Roughly taking 20 thousand more time to render than our ray-traced image. At this rate, the same image which hypothetically took 1 minute to render with ray-tracing could take 13 days with a pure Monte-Carlo GI renderer

Although it is possible to reduce the number of samples per pixels, doing so in pure Monte-Carlo renderers only produces noisy renders. The lower the number of rays per pixels, the noisier the render. Fortunately, Photon Mapping is much more clever than that.

Photon Mapping Parameters (or properties)

Choreography Properties

Photons Cast



This attribute determine the number of photons that gets stored in the photon map. The actual number of photons shot from the light sources will usually be lower than that. One single photon may be stored several times. Actually almost every time it interacts with a surface. The number of surfaces a photon may interact with is roughly determined by the Max Bounces attributes.

The number of photons stored have an impact on the render time. Even more so when Precompute Irradiance is not set. Photons are shot before the actual renders starts and is not resolution dependent. However, it impacts the Final Gathering step somewhat because the larger the photon map, the longer it takes to find the proper photon in it.

Sample Area

Sample Area and Photon Samples are related attributes. After photons have been stored in the photon Map, it is necessary to do an estimate of the density in photons in order to compute the irradiance at each photon position. This is performed by accumulating all photon energy in the neighborhood of each photon. This neighborhood is a disk with a radius determined by the Sample Area.

Sample Area attribute is specified in 100th of a centimeter. So a sampling area of 100 means a radius of one centimeter. The smallest specifiable area is 0.01 which means a radius of 1 micrometer. Small sampling area may be necessary when rendering a scene with very tight caustics.

Photon Samples

Photon Samples and Sample Area are related attributes. When computing the irradiance estimate, Photon Samples determines the maximum number of photons that are accumulated in the neighborhood of each photons. However, in area where the photons are very sparse, we don't want the irradiance estimate to grow so large that it would include photons which are too far away. This would falsify the estimate. Thus, the neighborhood will never go beyond the Sample Area. On the other hand, in areas where the photons are very dense, the neighborhood may grow much smaller than the Sample Area.

Basically, the irradiance estimate will stop whenever it reaches one of the two limits defined by Sample Area or Photon Samples.

Photon Samples have a major impact on the time it takes to precompute irradiance immediately after storing the photons or to compute irradiance on the fly when rendering caustics.

Intensity

Intensity may be used to increase or decrease the intensity of the caustic effects. It may also be used to increase or decrease the indirect illumination effect but it is less usefull in this respect.

Usually, the default setting of 100% is well balanced with the lights in a scene. However, the illumination computed by the photons is directly dependent on the light Intensity and Fall-Off parameters. If the light Fall-Off goes beyond the points of interest in a scene, the photons energy will be overestimated. It will then be necessary to compensate by lowering the choreography Intensity attribute. For further information about that, see the Fall-Off attribute description below.

Max Bounces

Determines the maximum number of bounces off surfaces that a single photon is allowed to do before being extinguished.

This have an impact on the quality of the indirect illumination result. Real photons can bounce almost ad infinitum. In GI, we have to stop somewhere. Clearly, it would not work to have, lets say, 1 photon bouncing 1 000 000 times. The default value of 20 is fine for almost all situations. But you could get darker shadowed areas by lowering the Max Bounces.

Another use of a low Max Bounces is if you want to use the photon map directly for indirect illumination by setting Final Gathering OFF. For some particular scenes, where indirect illumination is small compared to direct illumination, this may give acceptable result and reduce the render time significantly. In this case, a Max Bounces of 2 or 3 will give more uniform photons in the map.

When rendering caustics. Max Bounces determine the maximum number of time a photon will bounce off reflective surfaces or the maximum number of refractive surface it can pass through before it is aborted. When rendering caustics, photons are stored at the first non-reflective or non-refractive surface it encounters and it is not bounced any further afterward.

This attribute have no real impact on the render time.

Caustics

This is a switch which sets the Photon Mapping to render caustics instead of rendering indirect illumination. It basically turns OFF both Final Gathering and Precompute Irradiance and turns ON a filter for the photon irradiance calculation.

When Caustics is turned ON, photons are stored in the photon map only if they have bounced off a reflective surface or passed through a refractive surface. This ensures that the photons are used in the most effective way for producing caustics. Care should be taken to focus light as tightly as possible on the caustics producing objects though since any photons shot from the lights which do not reach a caustics producing surface will be lost and the time taken to cast it will also be lost.

Final Gathering

This attribute and Monte-Carlo Samples have the most impact on the render time. This turns ON or OFF the final gathering step of the Photon Mapping algorithm. The final gathering consist of accumulating multiple samples of the environment for each pixel in order to compute an estimate of the global illumination received by the pixel. The number of samples is determined by the Monte-Carlo Samples attribute.

Turning Final Gathering OFF means that the renderer will use the illumination directly from the photons stored in the photon map. In some scenes where the illumination comes mostly from lights and where indirect illumination is only used to add depth to the image, it is possible to get good results without Final Gathering. However, even with 1 million photons, the illumination as stored on the photon map is only a rough approximation of the actual true illumination of the scene. This will invariably produce false illumination in corners or edges of objects or on objects with a finely detailed surface.

Turning Final Gathering ON will make sure that each pixel gets its fair share of illumination estimate. Thus no corners, edges or fine surface details will be left in the dark.

Monte-Carlo Samples

This determines the number of samples of the environment at each pixel in the rendered image. A value of 150 normally gives good results but it is very dependent on the scene lights setup. Values lower than 50 gives strong illumination aliasing and artifacts.

This attribute have the most impact on the render time since it is resolution dependent and imposes, for each pixel, an illumination computation which is directly proportional to the number of samples.

Normally, a Monte-Carlo estimate would require much more samples than 150. However, the algorithm uses the same Quasi-Monte-Carlo sampling pattern at each pixels thus reducing significantly the noise in the render. This, however at the expense of a very biased render.

Jittering

In scenes illuminated mostly by indirect illumination such as a room where the illumination comes through a door from another room for example or scenes with very sharp illumination contrasts in the environment, because the Final Gathering uses the exact same sampling pattern from pixel to pixel, it will produce geometric patterns in the illumination.

Jittering allows to add some randomization to the sampling pattern. 100% Jittering will basically produce sampling patterns similar to a pure Monte-Carlo sampler thus producing noise in the illumination calculation. By specifying a lower percentage, it is possible to find an acceptable tradeoff between the noise and the geometric illumination patterns.

Unfortunately, scenes illuminated mostly by indirect illumination are difficult scenes to render for any Monte-Carlo based GI algorithms. It typically produces a lot of noise in the image unless several thousand samples are used per pixel. By keeping a regular sampling pattern and adding some jittering to it, some sort of control on the type of noise is relegated to the end user.

Precompute Irradiance

This attribute should always be ON especially when using Final Gathering as it dramatically reduces the render time.

If Final Gathering is turned OFF, then also turning Precompute Irradiance OFF might accelerate the render. The break even point is around the number of photons vs the number of pixels in the render. Roughly, if the number of pixels to render (taking into account the multi-passes or the anti-aliasing computations) is lower than the number of photons, then not precomputing irradiance might be advantageous. This is an unlikely situation though because assuming 1 million photons and a 640x480 pixels images with Anti-Aliasing, it is very likely that the total number of actual pixels computations will be higher that 1 million because of the adaptive pixel sub-sampling produced by anti-aliasing.

Another reason to turn Precompute Irradiance OFF when Final Gathering is also OFF is because it will produce different illumination patterns. This might be interesting for someone who wants to develop a personal rendering style.

Calculate Radiosity

This is not a property per se but can be selected in the choreography contextual menu (right-click menu). Selecting that option will calculate a photon map.

This option does the same calculation as when a final render is launched so it is not necessary to select this option before launching a final render to file. This option allows to test radiosity with progressive render. After changing something in the scene and before trying a quick or a progressive render, select this option to recalculate the photon map.

Note: Contrary to the old A:M implementation of radiosity where this option did produce an actual map of the illumination in a Targa file, this option will not do that for Photon Mapping. Even though there is the word Map in Photon Mapping, this map is a 3D spatial partitioning data structure and is not storable in an image file, nor usable as an image.

Light Properties

In radiosity renderers, light is treated slightly differently than for ray-tracing. Because Photon Mapping uses and combines both types of calculations.

Ray tracing shoots rays from the eye into the scene and thus it is possible to play tricks on how light intensity is computed. When a ray reaches the light, it knows exactly what the scene looked like from its point of view, because it just traveled through it. Photon Shooting, on the other hand, shoot photons from the light into the scene and when the photon leaves the light, it have no knowledge of the scene characteristics that it will encounter, It is thus not possible to play tricks with light intensity.

Thus, it is important to set the light parameters in such a way that it is compatible between the ray tracing engine and the photon shooting engine. Otherwise, the scene illumination will be unbalanced and it might then be difficult to try to compensate with the other radiosity parameters.

Another very important consideration which is indirectly related to the lights, is the closure of the scene. Photons are emitted from the lights into the scene and are stored (or used) only when they hit an object surface. This means that any photon that does not hit a surface is a lost photon and the time it takes to compute whether or not it hits something is also lost. This can potentially have a very bad effect on the rendering time. As a consequence, a good rule to follow is to always close the scene. For interior scene, it means that the interior should have all the walls, a full floor and a full ceiling. For exterior scenes, set your scene inside a hemisphere and on a floor which extends to the edge of the hemisphere. If you need a black void over your scene, then set your hemisphere to black and its radiance to 0%. By doing this, you ensure that all photons are stored instead of being lost in the void.

Type

Currently, the three lights types are supported with radiosity although with different level of compatibility.

Bulb

Photon shooting is fully compatible with bulb light parameters.

Bulb lights emits photons in all directions. Thus it may very well shoot photons in directions which are not important to the scene. Because of this, in many situations, it is better to use a klieg light instead of a bulb light.

Klieg

For klieg light, there is currently some differences in the way the photons are emitted and how intensity is computed. For one thing, the Width Softness is not directly supported although a klieg light with a width greater than 0 will automatically produce a softened illumination border. Normally, because photons are not used on the first surface they encounter, this should not cause a problem.

Also, in order to help simulate a flat luminaire, when the width of a Klieg light is set to 180° and the Width Softness is set to 100%, the photons are distributed following a cosine law which, in this particular case, is equivalent to a width softness of 100%.

Sun

There are two differences. Photons are emitted only from inside the width of the sun light while for ray tracing, a sun illuminates the whole scene no matter the width of the sun. In Ray-Tracing, the width of the sun is used to help compute soft shadows but photons are emitted strictly in parallel with the direction of the sun thus technically not producing soft shadows although this should not be a problem because photons are not used on the first surface they encounter and are not used to compute direct shadows but only indirect shadows.

The other difference is that because all photons are emitted following a parallel path, there is no photon power falloff. The photon power is strictly determined from the intensity and the fall-off parameters as will be explained below.

Width

Photons are emitted from the whole surface of the light. For a bulb light, photons are emitted from the surface of the sphere determined by the width attribute. For a klieg light and sun light, photons are emitted from the surface of the disk determined by the width attribute.

Width Softness

Width Softness applies to Klieg lights. When shooting photons, the width falloff is not taken into account. Rather, a natural softness emerges from the width of the light.

When the width of a Klieg light is set to 180° and the Width Softness is set to 100%, the photons are distributed following a cosine law.

Fall-Off

Fall-Off is the most important attribute to set right. Photon energy is balanced with the intensity of the light at the specified fall-off distance and further away.

With Ray-Tracing, light intensity is pretty much constant between the light source and the fall-off distance. The specified light Intensity is achieved at the fall-off distance and starts to fall-off as distance increases further away from the light. The intensity decrease follows the inverse square law meaning that intensity decrease to the square as the distance increases. Or said from another point of view: light intensity increases as it approaches the light but stops increasing(1) when it reaches the fall-off distance. With ray-tracing, it is possible to play tricks with light intensity because light intensity is computed in an analytic way.

When emitting photons however, this is a different story. Photons don't loose their energy as they travel. Instead, the total energy that reaches a surface is derived from the photon density on that surface. The inverse square law is a direct consequence of the dispersion of photon as they leave a point source. Thus, it is not possible to play the same tricks with the photon energy.

A consequence of all this is that in order to have a balanced photon illumination with the ray-traced illumination, the fall-off distance of the lights needs to be set carefully. The common habit of enclosing the scene inside the fall-off distance is a no-no with photon mapping. This is often used when trying to compensate for the absence of indirect illumination but with the additional indirect illumination that is brought by Photon Mapping, this will only produce wildly over-illuminated scenes.

As a rule of thumb, set your fall-off distances is such a way that it does not reach any of the objects in the scene and then adjust the intensity of the lights to get the illumination you are looking for. You can trust photon mapping to add much more illumination to your scene.

(1) This is a simplification. Actually the intensity is not quite constant but very nearly so.

Color

Light color contributes to the intensity. Darker light colors will provide less energy at the same intensity than brighter colors.

Intensity

When using Photon Mapping, the relation between Intensity and the Fall-Off distance is extremely important. Refer to Fall-Off for further explanation.

The specified Intensity is the light intensity that objects get within and up to the light fall-off distance. From the fall-off distance and further away, light intensity decreases. With Ray-Tracing, light intensity will practically never increase beyond the specified Intensity alue even if the distance is extremely near the light source. This is not the case with Photon Mapping.

Attenuation

In real word physics, light attenuation strictly follows the inverse square law. Photon Mapping strictly follows that law too. Because of that, you should always leave the light attenuation to 100%.

Surface Properties

Surface attributes are considered the same way whether they come from the whole object, a group or a material.

Diffuse Color

Darker color will absorb more photons. This means less photons bounces. Black color will absorb all photons thus photons hitting a black surface will not bounce back in the environment.

The diffuse color is also used to filter the energy of the photons be it for radiosity to produce color bleeding or for caustics produced by reflection or refraction.

Transparency

Transparency defines the percentage of photons which are bounced back in the environment vs the percentage of photons which are filtered through the surface. A 10% transparent surface will bounce back 90% of the non absorbed photons and let through 10% of the non absorbed photons.

Non-absorbed photons are dependent on the darkness of the color attribute and the radiance attribute.

Index of Refraction

Index of Refraction defines a transparent surface which will bend the photon path thus potentially producing Caustics. Potentially because if the Caustics attribute is turned OFF, then there will not be enough photons casted on the refractive objects to actually produce visible caustics.

When producing caustics, a large number of photons must reach the refractive or transparent surface. Thus, when setting up a scene for producing caustics, it is necessary to light the caustics producing surfaces as tight as possible.

Reflectivity

Reflectivity defines a reflective surface which will reflect photon thus potentially producing Caustics. Potentially because if the Caustics attribute is turned OFF, then there will not be enough photons casted on the reflective objects to actually produce visible caustics.

When a photon hits a surface, Reflectivity is in competition with Transparency and transparency have priority. If a surface is both transparent and reflective, the transparency takes precedence. For instance, if a surface was to be defined as 100% transparent and 100% reflective, then 100% photons would pass through the surface thus leaving no more photons to be reflected.

The priority of transparency over reflectivity is even more important to consider when rendering scenes with the "Caustics" attribute turned ON. In this case, if a surface is both transparent and reflective, then all photons will be refracted and none will be reflected. This is because caustics require a very even distribution of photons and thus cannot be split between refraction and reflection. Transparent object which produces caustics are generally convex and will converge photons, thus producing nice refracted caustics while they will generally diverge reflected photons, thus producing no observable reflected caustics anyway and this is why refractive caustics have precedence over reflective caustics.

Reflecting photons is different than bouncing photons. When reflecting photons, the direction the photon bounces back is very precise just like in a mirror while when a photon is bounced back by a diffuse surface, the direction the photon bounces back is at random.

When producing caustics, a large number of photons must reach the reflective surface. Thus, when setting up a scene for producing caustics, it is necessary to light the caustics producing surfaces as tight as possible.

Radiance

Radiance is the equivalent of reflectance in the radiosity literature. Reflectance is not the same thing as reflectivity. Reflectance specify the amount of photons that are bounced back in the environment while relfectivity describes how images of the environment are reflected in the surface.

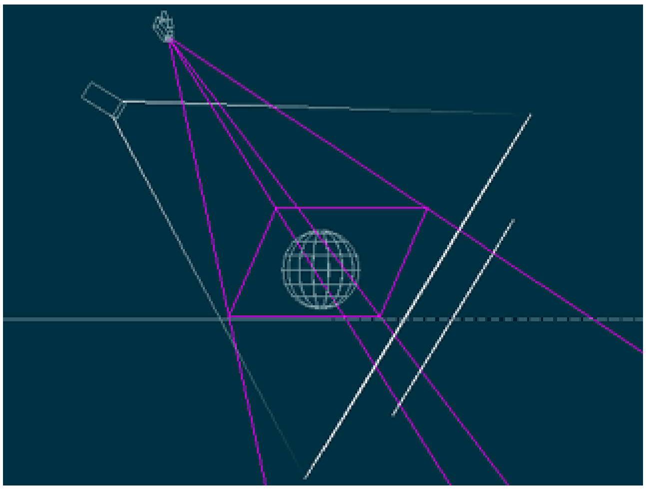
The default value of 100% for radiance will generally give good results. However, in the real world, even a canonical Munsen white piece of paper only have an average reflectance of about 92%. A radiance of 100% on a white surface would mean that the photons are not absorbed and should theoretically bounce ad infinitum which is not realistic nor feasible computationally. So a good rule of thumb for radiance value in normal radiosity situations is to simply set it at around 90% to 95% and let the color attribute take care of the true reflectance.

The Radiance attribute let you control the re-emissivity of the surfaces. For instance, you could enclose your scene inside a hemisphere but set the hemisphere radiance to 0% to make sure that it will not contribute illumination. Or if you want to render caustics, you can set all surfaces radiance to 0% except for the reflective or refractive surfaces which are expected to transmit photons.

Photon Mapping for Refractive Caustics

Caustics provide interesting lighting effects. They are the quickest to setup in a scene and they are also the quickest to render. There are two types of caustics. One is called refractive caustics when the caustics are produced by photons passing through transparent and refractive material and the other is called reflective caustics when the caustics are produced by photons being reflected off a reflective material.

While setting up a surface attribute that can exhibit one form of caustics is not difficult, setting up a surface that will properly exhibit the two form simultaneously can be a little trickier and requires more thinking and knowledge of surface properties.



Download the initial project file my clicking here

First I will review how to setup a scene for refractive caustics.

Setting up a scene for rendering refractive caustics is very simple. All you need is a transparent and refractive object that will bend the photon paths, a light that will emit photons toward the refractive object and a surface which will receive and display the photons.

In the example scene, the sphere acts as a refractive object, a klieg light oriented toward the sphere will emit photons and a simple ground plane will receive the photons.

The first thing to do when rendering caustics is to set the "Caustics" attribute in Choreography Properties to ON. But don't turn it ON immediately because there are other settings that we want to test using raytracing only.

Light setup

The second thing to do is to setup the light properties. Finding the correct set of radiosity attributes in the choreography properties can take some time and we want to have an initial setup that will render the quickest as possible and give us the most useful information on our progress.

Type

When rendering caustics, using a klieg light will ensure that the photon shooting step will be kept to a reasonable time. It is possible to use a bulb light. However, bulb lights will shoot photons in all direction but only those photons which reach the refractive surface will be used resulting in a lot of unused photons and thus a lot of lost time shooting photons for nothing.

Shadows

Make sure that the light shadow property is set to raytraced and only 1 ray is cast. This may be changed later after finding the proper radiosity attributes.

Width

Set the light width to 0. This will not speedup render time but it will greatly help find the optimal radiosity attributes. This may also be changed later.

Fall-Off

Set the light fall-off to the minimum distance that will give a proper lighting. Ideally, for radiosity, light fall-off should be set in such a way that it does not actually reach any object in the scene and then its intensity should increased accordingly. At least, its fall-off distance should never exceed the point where the caustics are projected on the receiving surface. Increasing fall-off beyond this point wil only produce overbright caustics and it will be necessary to compensate with photon intensity afterward which may not be even possible in situation where the fall-off extends way too far. In the example scene, here, the fall-off extends to about the center of the caustics projected on the ground. This will produce overbright caustics but it is OK for demonstration purposes. (See the description of the light fall-off)

Refractive Object Setup

There are some necessary attributes to setup for refractive setup. They are:

Transparency

The object needs to be transparent. 98% transparency will do.

Index of Refraction

The object needs an Index of Refraction. This is important otherwise the photons will not bend at all and thus will not produce caustics. An IOR of 1.1 will do.

The other following attributes are not a requirement for refractive caustics setup but will make a more believable glass sphere. They are:

Reflectivity

A glass sphere would also be reflective. A reflectivity of 40% will do. A reflective surface will also produce reflective caustics. The consequence of having both reflective and refractive attribute on a surface is that the photons will be split 50/50 between refracted photons and reflected photons thus potentially producing more grainy caustics. But this can be taken care of by increasing the number of photons in the scene. For the purpose of realism, the total of transparency plus reflectivity should not be higher than 100%. Although the photon mapper will render such a surface, physically, this would mean that more photons leave the surface than it receive which is clearly not realistic.

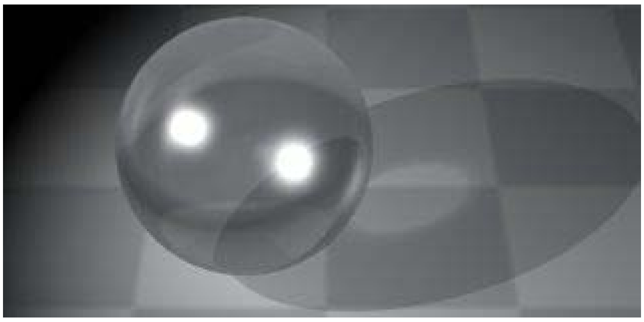
Specular Size and Intensity

A glass sphere would not quite look like a glass sphere without specular highlight. Set the Specular size to 50% and the intensity to 100%.

Reflectivity Fall-Off

This is a nice trick I learned from examining Fabrice Favé original Sphere scene. By adding a Reflectivity fall-off, it gives a nice diaphanous transparency to the sphere. A value of 350 will do.

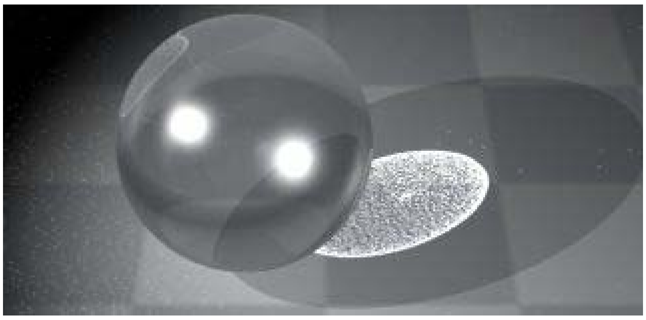
Choreography Setup



The initial scene render should look something like that.

Note that the A:M renderer tries to simulate a caustics even when not using photon mapping

But now is the time to add some real caustics.



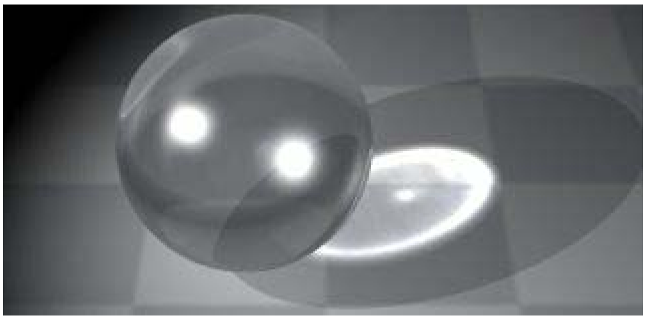
Simply turning the Radiosity ON and then Caustics ON on the choreography properties will give this render.

In this case, the default setting of 10k photons, sampling area 100 and 100 photon samples produces grainy caustics.

Grainy caustics generally means that the sampling area is too small. But a number of photon samples too small will also produce this result. However, 100 photon samples is generally a good start so it is best to first focus on the sampling area.

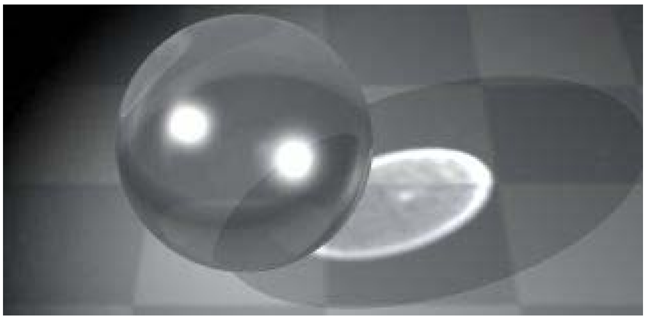
How much too small is not easy to guess. I use a procedure called dichotomic search to help find an almost optimal set of attributes. The idea of dichotomic search is to try out large variations of one attribute and then reduce by half the variation inside the span which offers the best probability of nice render.

For instance, I know that a sampling area of 100 is too small. I will then try a much larger sampling area. I will try 1000.



This will produce this render. It can be noticed that now, the caustics is much softer but also a little too muddled on the edges.

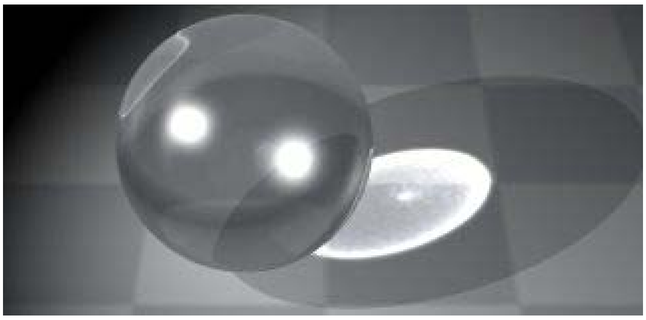
Following the dichotomic search rule, I would test a value about midpoint between 100 and 1000. So I will next test a sampling area of 500.



Which produces this render.

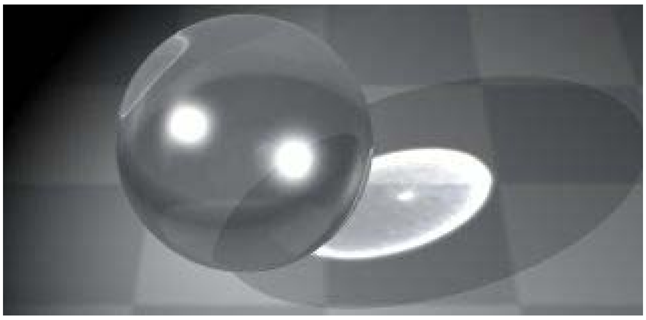
I find the caustics still a little bit muddled on the edges but grain starts to appear so still tweaking the the sampling area will not get me the quality I want.

So mMy next best bet is to increase the number of photons.



Increasing the number of photons to 100k should be a good start for the dichotomic search. It renders like this.

It seems overkill so by the dichotomic rule, I will reduce the number of photons to about half between 10k and 100k.



50k photons looks good. Nice caustics edge definition and no annoying grain.

I could probably keep on searching for a more optimal attribute set but for the purpose of this tutorial, I will let it like that.

Sometime, finding the optimal caustics set of attributes can look like a black art. It is not provided that a very systematic method is followed. This systematic method is the dichotomic search. It is the fastest method available for this type of search. The next section will cover a little bit on that method.

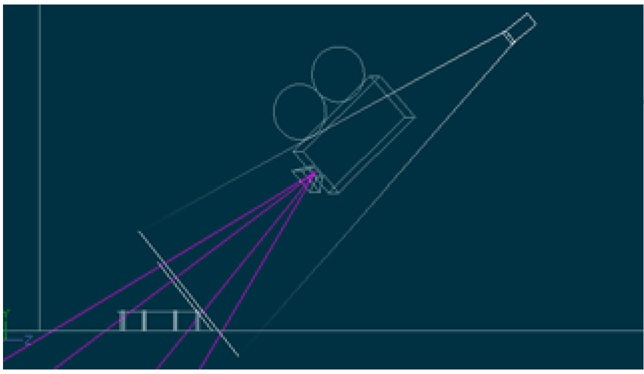
Photon Mapping for Reflective Caustics

Setting a scene for reflective caustics requires a reflective object which will reflect photons back in the scene, a light which will emit photons and a surface which will receive the photons.

In the example scene, the reflective object is a ring, the light is a klieg light and the receptive object is a simple floor.

Rings are often used in caustics test scenes because they exhibit a nice characteristic cardioids caustics.

Here again, we shall start by setting the scene without regards to the caustics themselves. So the Radiosity



Download the initial project file my clicking here

attribute should be turned OFF for now.

Light setup

Type

Here again, the light is a klieg light for the same reason of trying to shoot as little unused photons as possible.

Shadows

Set the light shadow property to raytraced and only 1 ray cast. This may be changed later after finding the proper radiosity attributes.

Width

Set the light width to 0. This will not speedup render time but it will greatly help find the optimal radiosity attributes. This may also be changed later.

Fall-Off

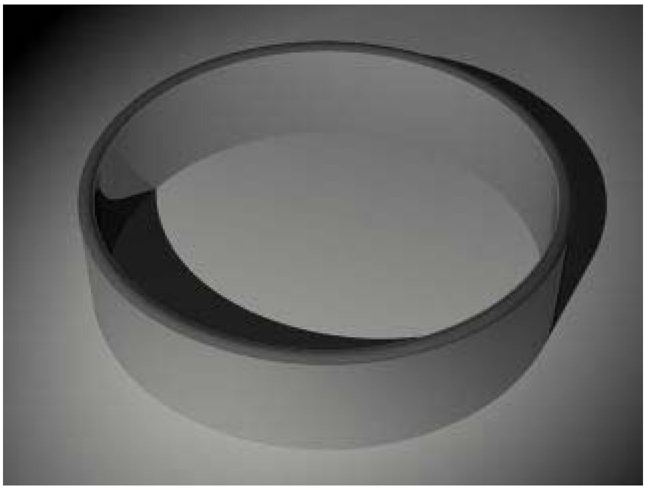
Set the light fall-off to the minimum distance that will give a proper lighting. In the example scene, here, the fall-off is set so it just touches the ring object. This will produce caustics intensity which are well balanced with the illumination of the scene. (See the description of the light fall-off)

Reflective Object Setup

Reflectivity

The only attribute that needs to be setup here is the reflectivity. A reflectivity of 100% would look unnatural. In the example, the reflectivity is set to 50%.

Choreography Setup



Rendering the scene would produce the basic scene similar to this one.

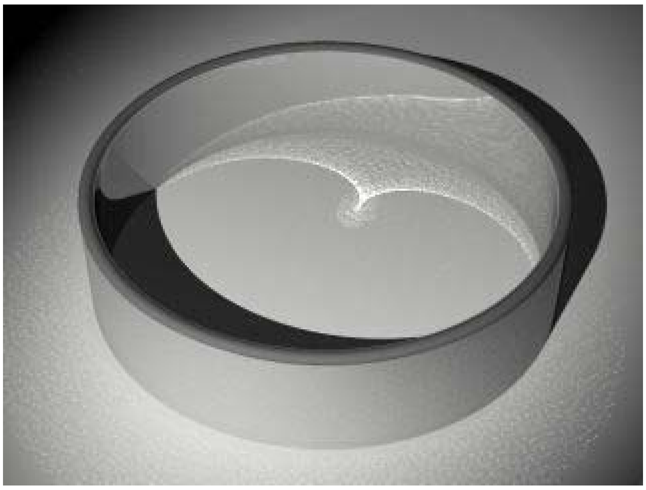
The caustics sharpness is dependent on the radiosity attributes set but also on the object size. Using the default radiosity attributes set on objects of different size will produce different quality of caustics. So you shall get similar results if you use the supplied scene but you may get different results with your own scene.

We shall add caustics by turning Radiosity ON and Caustics ON in the Choreography properties panel. As a first test, we may leave the default values for the other radiosity attributes as they are.

The result should look similar to this render.

Not so bad. But it certainly could be improved with much sharper caustics edges. Too muddled caustics edges generally means that the sampling area is too large.

Applying the dichotomic search procedure, I will first try a sampling area 1/10th the current default of 100.



Which will produce a render like this one.

Now the caustics edges are much sharper but the sparse photon area of the caustics is grainy which indicates that the sampling area is now too small.

For the next test, I will try a sampling area about midpoint between 100 and 10. I will test a sampling area of 50.

Which renders like this.

The result is somewhat still grainy. I will have to increase the sampling area some more. However, I also notice that the caustics edges are muddy again and will certainly get muddier as I increase the sampling area. I can take care of the sharpness later but the grain is in direct relation to the sampling area. So I will now try a sampling area about midpoint between 100 and 50. I will try a value of 75.

Which produces this render.

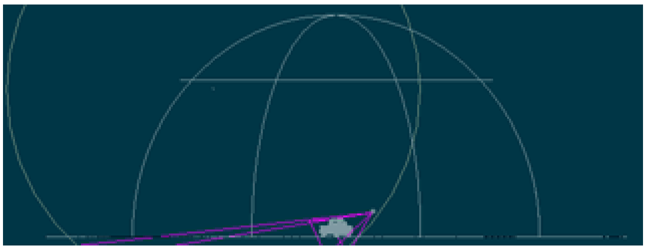
I got rid of the grain in the caustics but at the expense of muddled caustics edges. This reflective ring bounces photons not only inside the ring but also in front of the ring. As a consequence, about half the photons cast on the ring are used for the inside caustics. Also, the ring is a rather large object. So my next best bet is to increase the number of photons in the Scene.

Trying with 100k photons and rendering will produce this.

Now this is caustics the way I like them. 100k photons might be a little expensive though so I could find a more optimal number of photons by using the dichotomic search procedure again on this particular attribute. But for the purpose of this demonstration, it is good enough and I will leave it like this.

Photon Setup for Outdoor Scenes

Photon Mapping can be very effective in replacing a skylight rig with a sun. This is not the only use of Photon Mapping for outdoor scene but it is the most basic setup which you will want to improve in different ways,



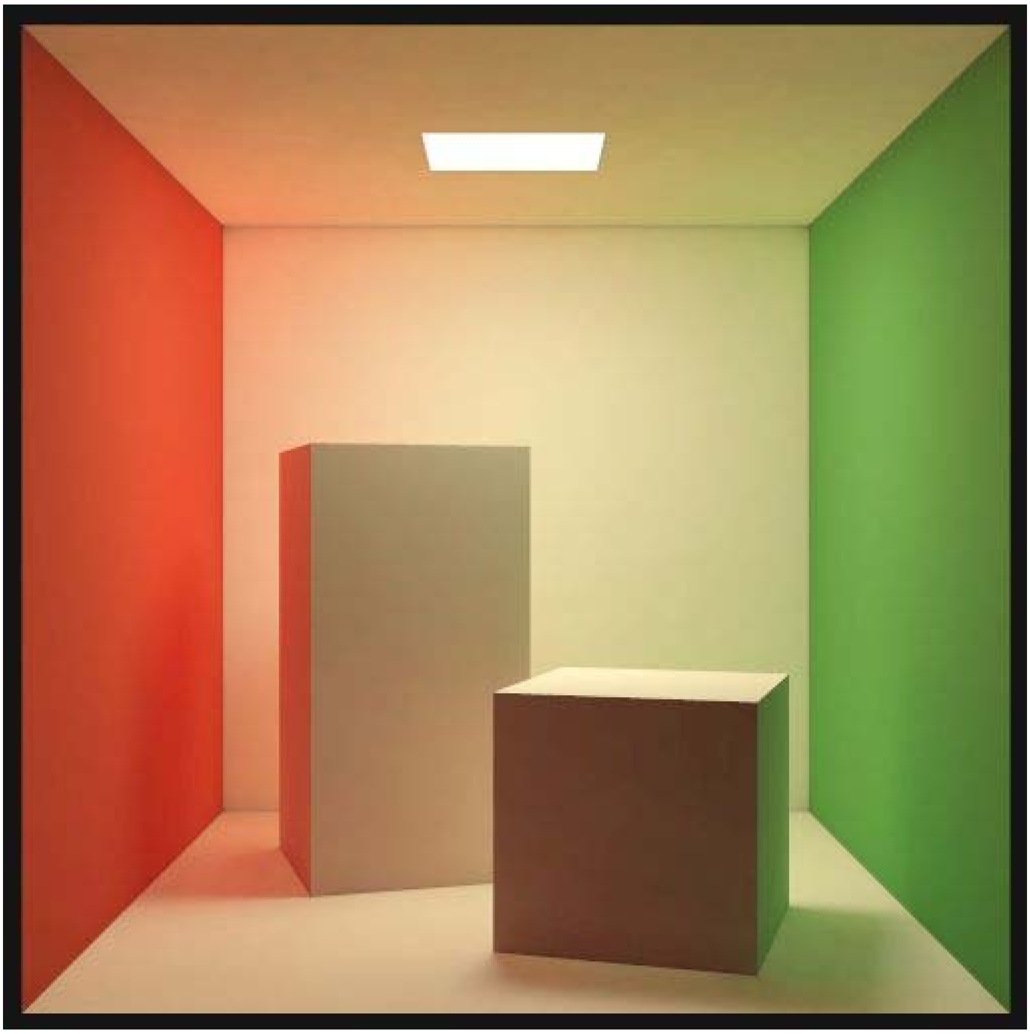
The basic scene setup is simple: place an object on a floor and cover it with a skydome making sure the floor closes the skydome. Then place a bulb light near the dome, turn ON radiosity and render with Photon Mapping.

This is the basic idea. But in order to get a nice indirect illumination, here again, the radiosity attributes need some special attention. The first issue to consider is the small size of the object of interest relative to the size of the whole scene. Because of this, the object itself will receive a small number of photons relative to the rest of the scene. Of course, it would be possible to compensate with a tighter dome but this would place the light too close to the object. The other option is to shoot a large quantity of photons in the scene. Shooting 1M photons is not unreasonable for such a scene.

The second issue to consider is the number of photon bounces. In this case, a large number of bounces would pollute the sky with photons coming from the ground, the main object and the other objects in the scene thus producing an indirect illumination from the sky wich is no more bluish. This is clearly not what we want

The Cornell Box Tutorial - Setup

So here is the Cornell Box in all its glory:



Important note: All renders in this tutorial have been Gamma corrected with a Gamma of 2.2. This is a necessary post correction in order to match photos of a real Cornell Box since photos themselves have a gamma corection of about 2.2.

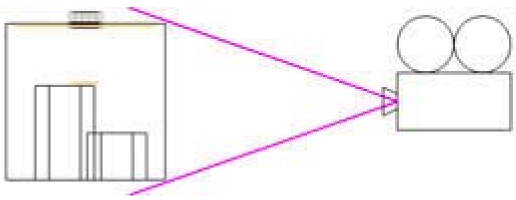
And this is the scene we are going to use for the tutorial.

The Cornell Box is probably the oldest scene used for comparing radiosity rendering engines. The most popular configuration is described at the http://www.graphics.cornell.edu/online/box/data.html.

For this tutorial, I invite you to download the Cornell-Demo-Defaults.prj project file. This is the actual Cornell Box data I modeled in A:M from the published specs for both dimensions and surface properties. The Cornell Box was a real box with real blocs inside. A photograph of the inside of the box was taken and the radiosity simulation renders were compared to the real photograph to see if the algorithm was working right. Thus, the Cornell Box have only 3 walls to allow taking a photo of the inside.

Here are a few things to note about this scene:

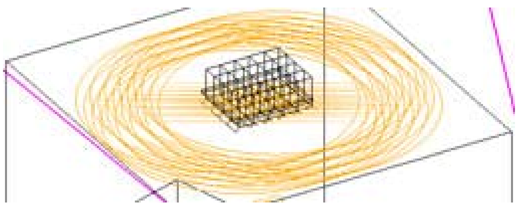
Small size of the scene



First thing to note is this scene is small. Very small compared to a normal room size. The room itself is 55cm wide by 55 cm high by 57 cm deep.

It is important to note that the Cornell Box is small relative to a normal room. Room size have a major influence on the way Photon Mapping properties should be set so it should always be noted carefully and objects in the room as well as the room itself should be modeled to actual scale.

Light arrangement



Second thing to note is the light arrangement.

The cornell Box is illuminated by a single flat and rectangular luminaire with a white diffuser. This type of luminaire have a very specific lighting characteristics. To simulate the luminaire, I arranged 25 klieg lights in a 4x5 array. This is actually one klieg light instanciated 25 times in the choreography.

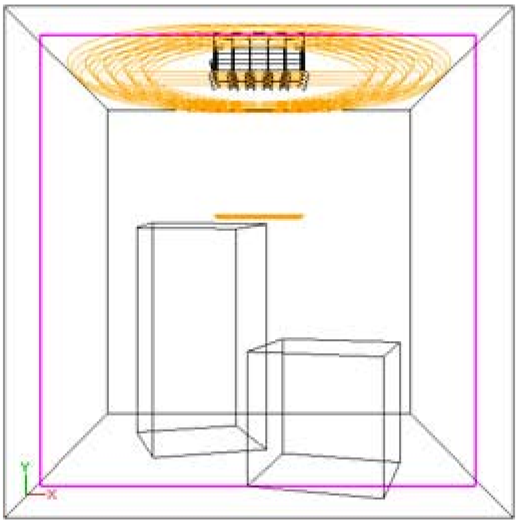
I decided on a 4x5 array because I wanted to stay as close as possible to the original rectangle of the luminaire. I could have decided to use one large klieg light instead. But even though the representation of a klieg light face is a square the beam it project is clearly round. The shadow characteristics of a rectangular light fixture is sufficiently different from that of a round fixture that I felt I could use several lights instead of just one. Beside, in order to produce the same nice soft shadows from one large klieg light, I would need to multiply its number of ray cast by the number of lights in the array. In the end, that would not save on render time either. So the choice of using a light array instead of one single large light is a design choice. Not a technical choice. It is entirely based on the type of shadow quality I was looking for.

Light properties

Third thing to note is how the klieg light is setup. To simulate the way a flat diffuse light illuminates a scene, the klieg's Cone Angle is set to 180° and the Width Softness is set to 100%. The 100% Width Softness ensures that illumination falls-off at a cosine rate according to the angle from the light surface. This is how light from a flat surface behaves in reality. That is light emission reduces according to the cosine of the angle between the luminaire plane and the direction it is emited.

When setting a scene for Photon Mapping, or for any radiosity or Global Illumination, it is important to take care of all aspects of the scene and models in order to get realistic rendering. Failure to properly replicate surface and light characteristics will almost certainly produce non realistic renders. Adding radiosity to a crap scene will only produce crap radiosity renders. In other words, using Photon Mapping or radiosity requires more technical knowledge about light and surface characteristics and their behaviors than using traditional renderers.

Falloff distance



Fourth thing to note is the klieg's Fall-Off distance.

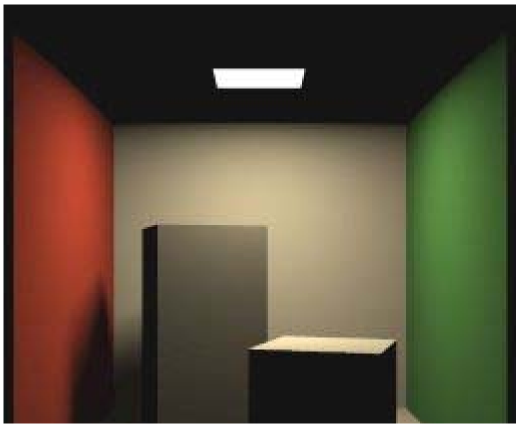
The Fall-Off distance should be set is such a way that it doesn't touch any object in the scene. In this scene, you can see that the Fall-Off stops short of touching the walls and the tall bloc. This is especially important when the light intensity is small to ensure that all objects will be directly illuminated in a natural way.

If the fall-off distance is too long, the parts of the objects that falls inside the falloff distance will receive constant illumination. That is, no illumination gradient will be visible. This is clearly not something we want if wee wish to realistically render the scene.

The Fall-Off distance could be shorter but ideally it just needs to not touch any objects in a scene. If the falloff distance if too short, then light intensity needs to be cranked up very high. Sometime it is not really set a falloff distance that won't touch any objects in the scene. When this is not possible, then there are other rules to take care of.

In radiosity scene, all lights must cast shadows. Ray-traced shadows with 100% darkness, 16 Rays Cast and "Distribute Among Passes" set to ON in the shadows properties. Multi Rays Cast and distributing rays among passes is not only to get softer shadows, it is also to ensure that the shadows are actually black. When a light is setup to cast only one ray, the renderer does its best to simulate indirect lighting which results in non-black shadows. This is useful for normal Ray traced renders but not needed, nor desired for radiosity renders. Light shadow properties sould always be set to raytracing with more than one ray and shadow darkness set to 100% when lighting a scene for radiosity. Z-buffer shadows don't produce realistic shadows and should be avoided for radiosity renders.

Low light intensity



Fifth thing to note is the low light intensity.

The render to the right is a Raytrace render of the Cornell Box for comparison purpose.

For Raytrace render, you would probably want to have higher light intensity. But for Radiosity render, because the surfaces will also contribute light in the scene, light intensity are lower. In fact, light intensity can only really be adjusted by trial and error in Radiosity render mode. But there are ways to do quick preview radiosity renders.

Also, because radiosity renders must be gamma corrected with a gamma of 2.2, the "Preview Gamma" in the Tools-Options-Render pannel should also be set to 2.2 otherwise, it will be very difficult to find the proper light settings by trial and error because the final render will be very different from the preview renders.

On the choreography properties pannel, turn the Radiosity property to ON and render with the default radiosity settings which are: Photon Cast : 10000, Sample Area : 100, Photon Samples : 100, Intensity : 100%, Max Bounces : 15, Caustics : OFF, Final Gathering : ON, Samples : 100, Jittering : 0%, Precompute Irradiance : ON.

Calculating the Sample Area

First Photon Mapping render

The result of rendering the Cornell Box scene with the previous settings should look like the picture displayed to the right.

Before continuing on to the radiosity properties settings, I'd like to multi-passes and render time.

Rendering this scene with Antialiasing (no multipass) or multipass with 1 pass or multipass with 9 passes, require basicaly the same render time and gives basically the same result. This is one important concept . On my computer, the render times were almost the same for all types of render:

10m35s for antialiasing render
06m33s for one pass render
10m55s for 9 passes render

You may have expected the 9 passes render to take approximately 9 times longer than the one pass render. But for Photon Mapping, this is not the case. For the Photon Mapping rendering engine, the most time and CPU consuming step is the Final Gathering step. And the time it takes is directly influenced by the number of Final Gathering samples. However, the Final Gathering samples are distributed among the passes. So when rendering only one pass, the final gathering need to sample all the 100 samples per pixel while when rendering 9 passes, each pass samples only 100/9 = 11 samples per pass.

So because of that, when you use Photon Mapping, you don't have to worry about using multipass. Multipass will not have a huge impact on Photon Mapping render times.

Getting the radiosity render right

The radiosity default settings are rarely appropriate for a given scene. A noisy render like that is indicative that there either is not enough photons in the scene or the Sample Area is too small. Since this scene is small compared to a normal room size, the noise is not so bad. But for a normal room size, the noise can look really bad.

As it turns out, for this scene, increasing the number of photons to 1 million would be an appropriate solution. But it would considerably increase the render time too. For such a simple scene, it is really not necessary to go to such high number of photons. We could actually decrease the number of photons and get a very acceptable radiosity render provided the Sample Area is adjusted appropriately. It is always advisable to try increasing the sample area first.

There are basicaly two ways to find the proper sample area. A visual procedure and a mathematical estimate. Finding the correct Sample Area is not trivial. It can be found by trial and error but there is a more systematic way to proceed which I will review later. But first, there is also a mathematical formula to help guess a good starting point. Instead of explaining this mathematical formula, I have concocted an Excel spreadsheet which you may download by clicking on the picture below.

Room width (cm) :	55		
Room depth (cm) :	57		
Room height (cm) :	55	Total room area (cm²) :	18590
Estimated additional surfaces (cm²) :	443	Total area to light (cm²) :	19033
Number of photons :	10000	Photons per cm² :	0,525403
Sampling area :	100	**Suggested sampling area :**	**750**
Photon samples :	100	Samples cover (cm²) :	190,33

For those without access to Excel, I've added a detailed mathematical explanation at the end of this tutorial page so you can do the math with a calculator.

So enter the room size which is 55x57x55cm and an estimate of additional object surfaces in the room. In the Cornell Box case, the estimate is simple since we know the tall and short blocs sizes which are 34x17x17cm and 17x17x17cm respectively. And since the room itself is missing the front wall, a 55x55cm area is subtracted which gives 443cm additional.

Simply enter the suggested Sample Area value in the property box and rerender at 9 passes. You should get a result similar to the render to the right.

That was easy.

Fortunately, for the Cornell Box scene, it is easy to estimate the total surface area. In most scenes, however, the estimated additional surface cannot be computed as easily and must be guessed. It is easier when the size of the room have been entrered and a relative additional surface area can then be guessed from that. For some other scenes, with a lot of nooks and cranies, the size of the room can only be used as a first estimate and finding the correct sample area will require further trial and error from that starting point.

As a rule of thumb, it is safer to have larger sample area than smaller ones. So it is better to guess larger additional surface areas than smaller areas. The larger the sample area from the optimal, the more the light bleeding approaches a simple ambiance setting. So ultimately, one could reduce the number of photons and increase the sample area to a point where each surface emits a constant ambiant component. For scene with several small surfaces, however, this may not be so easy to achieve.

Now that you have this Excel utility, you may want to try with lower and higher number of Photons. See how low you can go and still have acceptabe radiosity renders by setting the appropriate sample area. And see how the number of photons in a scene affects the render time.

Computing the Sample Area with a calculator

The idea is to estimate a sampling area that will most probably leave no holes in the photon coverage. The information we need for that are : the total surface area to cover with photons and the area that is covered by each photon irradiance estimate.

The most significant measures to help estimate the surface area to cover with photon is the size of the room. Here, the room was 55x57x55 cm. So the total surface area of the room is the sum of each wall areas. There are 4 walls of 55x57 and one wall of 55x55 (the front wall is missing) so we have (55x57x4) + (55x55) = 15565cm². Then we need to add the estimated area of all the objects in the room. In most situation, it is not practical to actually compute the surface area of each objects the way we did it for the Cornell Box. A rough guess will do just fine and an overestimated guess is better than an underestimated guess. So let's say we guess 5000 additional cm² and let's round the total estimated surface area to 21000cm².

The next measure we need is the coverage of each photon irradiance estimate. We shoot 10000 photon on 21000cm² surface. That means that we will get approximately 1 photon per 2.1 cm². And since we use 100 photon samples for each irradiance estimate, each photon irradiance estimate will cover an average of 210 cm².

With this estimated photon irradiance coverage, we can estimate the sample size in this way : The radius of a circle covering a 210 cm² area is SquareRoot(210/Pi) which gives 8.18 cm. And since the sample area is given in 1/100th of cm, we get get a sample area of 818.

Recap

Given a room where W is the width, H is the height and D is the depth, Scene Area = (W*H*2) + (W*D*2) + (H*D*2) + Additional object areas.

Irradiance Coverage = Scene Area * Photon Samples / Number of Photons.

Estimated Sample Area = SquareRoot(Irradiance Coverage * Pi) * 100.

Steffen Gross calculate surface plugin

Steffen Gross have programmed a plugin that will calculate a given scene total surface area. Here is a short description of its usage:

- Plugin is only for A:M V11 and V11.1 (PC of course)
- Start the plugin from the chor , the chor MUST be in the Frontview (Numpad-2), otherwise you get false results for the total surface.
- You can change all inputfields, but hit"Recalculate" after changing them to get the correct sample area.
- The changed values are stored in the corresponding properties in the chor after hitting OK , Cancel return the plugin without any changes.
- Use it at your own risk ....

Visually Finding Photon Properties

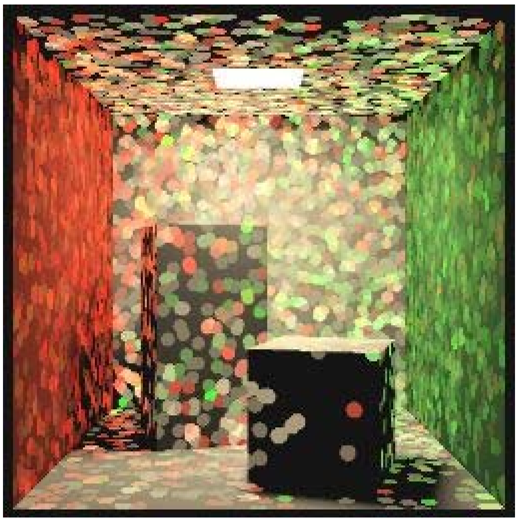
The alternative way to determine optimal Photon Mapping properties is to directly visualize the Photon Map. To demonstrate that, we will use the Original Cornell Box project.

First reset all Photon Mapping properties to their default values. Number of photons : 10,000, Sample Area : 100, Photon Samples : 100, Max Bounces : 15.

Then, this is the most important step, set Final Gathering to OFF.

In the render to file pannel, set multipass to ON but set the number of passes to 1. This is an important step When you visualize the photon map directly, you want only one pass in order to clearly see how the photons cover the surfaces. If you used more than one pass, because the light is sampled at different positions at each pass, it would be difficult to visually figure the actual area covered by each photon.

A render should produce something like this:



This is the polka dot syndrome. When the sample area is too small, you can clearly see the polka dots. The isolated polka dot size is exaclty the size of the sampling area and eack polka dot is actually one photon.

The photons overlap but they are still individually visible. Ultimately, you could truely see each individual photons if you rendered large enough and with a sample area small enough. Try setting the sample area to 20 and render for an example.

The idea of the visual procedure is to increase the Sample Area untill the direct visualization of the Photon Map gives almost acceptable renders.

By acceptable, I mean that the photons, in the scene, overlap one another and blur themselves to a point where they are no more individually noticeable and their illumination is nice and soft.

For instance increasing the Sample Area to 250 gives the render to the right which is still not quite acceptable.

The polka dots are still clearly visible on the short block front face but this particular face is not relevent in this scene since it faces the void and will always receive very little photons whatever the settings we may try. A look anywhere else, and in particular at the walls and the other blocks surfaces shows very eneven variation of irradiance.

Ideally, we want to set the sample area in such a way that the resulting photon merge will be much more smoother than what we have here and should look almost like an acceptable radiosity render. That is the irradiance on the walls and on the objects have a nice smooth gradient. Some noticeable blotches are acceptable (and unavoidable) but there should be no noticeable hard edges in the irradiance, except, in this case, on the small block front face.

Usually, a radiosity scene should be set inside a closed environment. The Cornell Box scene is an exception because, in this case, we want to replicate the illumination of a real box which had an opening instad of the front wall. Any opening in the scene that ends up into a void will unnecessarily generate photons that will hit nothing. Any photon that hits nothing will be replaced by a new photon so this increases the render time in a very inefficient way.

Increasing the Sample Area to 750 as computed by the Excel utility gives the render to the right, which could almost be usable as is.

When you reach a point where increasing the Sample Area does not improve the smoothness of the photon map anymore, then you have reached the limit of what the Photon Samples can give you. If you want to improve a little bit further, you will have to increase the Photon Samples property and then the Sample Area.

If you did the examples in this page, you noticed that direct visualization of the Photon map is much faster than with Final Gathering. Because of that, it is tempting to try to find direct visualization properties that would produce perfect illumination. However, it will not be possible to get a perfect illumination in that way. The direrect photon map visualization will always produce dark corners and edges on the illuminated objects. This is intrinsicaly related to the way irradiance is computed. So be already advised. The direct visualization of the photon map is suitable for finding appropriate properties by trial and error but it cannot be used as a replacement for Final Gathering.

Adjusting Photon Samples

For instance, increasing the Photon Samples to 500 and then the Sample Area to 1700, you would get the render to the right which, I personally think is pushing it a little too far as most of the irradiance gradients are completely lost. The previous setting left no large holes and kept the gradients very visible. There is a good compromise to be found somewhere.

In general, increasing the photon samples will produce less saturated illuminations. But the effect is really scene dependent. Changing Photon Samples in a highly detailed scence with lots of small objects or objects with lots of small modeled details may make a difference. But there is no formal way of determining its optimal value, unlike sample area.

If changing Photon Samples in a particular scene does not make any difference, then your scene does not have the complexity where it is necessary to increase the photon samples. In this case, it is always recommended to leave it to the smallest value as possible since it can have a big impact on render time.

I still haven't found a user-friendly explanation for this property but here is one: After shooting photons in a scene, we have a very noisy distribution of photons everywhere coming from everywhere. In order to make sense out of that, we need to compute a statistical irradiance values by averaging a certain number of neighboring photons around each individual photons in the scene. That is what "precomputing irradiance" does. The maximum number of neighbor photons that may considered for this statistical evaluation is determined by the Photon Samples property.

Visual procedure

- Turn Final Gathering OFF
- Select multipass but set only one pass.

- Change the sample area incrementally to get the best photon distribution based at how dense the photons look on the surfaces.
- If the density looks too washed out or flat, then lower the Sample Area
- If the density looks too separated or dotty and muddled or full of variation, then raise the Sample Area.

Conclusion

The interesting aspect with this method of searching for the optimal properties is that you have a visual way to judge the irradiance distribution. It is a good practice to use both methods. That is finding good estimates with the Excel utility and improving the estimate with the visual method.

Different ways to render a Cornell Box

What follows are renders that went wrong during the development of the A:M Photon Mapping rendering engine. You will not be able to reproduce those renders.