Wikipedia

Photon mapping

In computer graphics, **photon mapping** is a two-pass global illumination algorithm developed by Henrik Wann Jensen that approximately solves the rendering equation. Rays from the light source and rays from the camera are traced independently until some termination criterion is met, then they are connected in a second step to produce a radiance value. It is used to realistically simulate the interaction of light with different objects. Specifically, it is capable of simulating the refraction of light through a transparent substance such as glass or water, diffuse interreflection between illuminated objects, the subsurface scattering of light in translucent materials, and some of the effects caused by particulate matter such as smoke or water vapor. It can also be extended to more accurate simulations of light such as spectral rendering.

Unlike path tracing, bidirectional path tracing, volumetric path tracing and Metropolis light transport, photon mapping is a "biased" rendering algorithm, which means that averaging many renders using this method does not converge to a correct solution to the rendering equation. However, since it is a consistent method, any desired accuracy can be achieved by increasing the number of photons.

Contents

Effects

Caustics

Light refracted or reflected causes patterns called caustics, usually visible as concentrated patches of light on nearby surfaces. For example, as light rays pass through a wine glass sitting on a table, they are refracted and patterns of light are visible on the table. Photon mapping can trace the paths of individual photons to model where these concentrated patches of light will appear

Diffuse interreflection

Diffuse interreflection is apparent when light from one diffuse object is reflected onto another. Photon mapping is particularly adept at handling this effect because the algorithm reflects photons from one surface to another based on that surface's bidirectional reflectance distribution function (BRDF), and thus light from one object striking another is a natural result of the method. Diffuse interreflection was first modeled using radiosity solutions. Photon mapping differs though in that it separates the light transport from the nature of the geometry in the scene. Color bleed is an example of diffuse interreflection.

Subsurface scattering

A model of a wine glass ray traced with photon mapping to show caustics.

Subsurface scattering is the effect evident when light enters a material and is scattered before being absorbed or reflected in a different direction. Subsurface scattering can accurately be modeled using photon mapping. This was the original way Jensen implemented it; however, the method becomes slow for highly scattering materials, and bidirectional surface scattering reflectance distribution functions (BSSRDFs) are more efficient in these situations.

Usage

Construction of the photon map (1st pass)

With photon mapping, light packets called *photons* are sent out into the scene from the light sources. Whenever a photon intersects with a surface, the intersection point and incoming direction are stored in a cache called the *photon map*. Typically, two photon maps are created for a scene: one especially for caustics and a global one for other light. After intersecting the surface, a probability for either reflecting, absorbing, or transmitting/refracting is given by the material. A Monte Carlo method called *Russian roulette* is used to choose one of these actions. If the photon is absorbed, no new direction is given, and tracing for that photon ends. If the photon reflects, the surface's bidirectional reflectance distribution function is used to determine the ratio of reflected radiance. Finally, if the photon is transmitting, a function for its direction is given depending upon the nature of the transmission.

Once the photon map is constructed (or during construction), it is typically arranged in a manner that is optimal for the k-nearest neighbor algorithm, as photon look-up time depends on the spatial distribution of the photons. Jensen advocates the usage of kd-trees. The photon map is then stored on disk or in memory for later usage.

Rendering (2nd pass)

In this step of the algorithm, the photon map created in the first pass is used to estimate the radiance of every pixel of the output image. For each pixel, the scene is ray traced until the closest surface of intersection is found.

At this point, the rendering equation is used to calculate the surface radiance leaving the point of intersection in the direction of the ray that struck it. To facilitate efficiency, the equation is decomposed into four separate factors: direct illumination, specular reflection, caustics, and soft indirect illumination.

For an accurate estimate of direct illumination, a ray is traced from the point of intersection to each light source. As long as a ray does not intersect another object, the light source is used to calculate the direct illumination. For an approximate estimate of indirect illumination, the photon map is used to calculate the radiance contribution.

Specular reflection can be, in most cases, calculated using ray tracing procedures (as it handles reflections well).

The contribution to the surface radiance from caustics is calculated using the caustics photon map directly. The number of photons in this map must be sufficiently large, as the map is the only source for caustics information in the scene.

For soft indirect illumination, radiance is calculated using the photon map directly. This contribution, however, does not need to be as accurate as the caustics contribution and thus uses the global photon map.

Calculating radiance using the photon map

In order to calculate surface radiance at an intersection point, one of the cached photon maps is used. The steps are:

1. Gather the N nearest photons using the nearest neighbor search function on the photon map.
2. Let S be the sphere that contains these N photons.

3. For each photon, divide the amount of flux (real photons) that the photon represents by the area of S and multiply by the BRDF applied to that photon.
4. The sum of those results for each photon represents total surface radiance returned by the surface intersection in the direction of the ray that struck it.

Optimizations

- To avoid emitting unneeded photons, the initial direction of the outgoing photons is often constrained. Instead of simply sending out photons in random directions, they are sent in the direction of a known object that is a desired photon manipulator to either focus or diffuse the light. There are many other refinements that can be made to the algorithm: for example, choosing the number of photons to send, and where and in what pattern to send them. It would seem that emitting more photons in a specific direction would cause a higher density of photons to be stored in the photon map around the position where the photons hit, and thus measuring this density would give an inaccurate value for irradiance. This is true; however, the algorithm used to compute radiance does *not* depend on irradiance estimates.
- For soft indirect illumination, if the surface is Lambertian, then a technique known as irradiance caching may be used to interpolate values from previous calculations.
- To avoid unnecessary collision testing in direct illumination, shadow photons can be used. During the photon mapping process, when a photon strikes a surface, in addition to the usual operations performed, a shadow photon is emitted in the same direction the original photon came from that goes all the way through the object. The next object it collides with causes a shadow photon to be stored in the photon map. Then during the direct illumination calculation, instead of sending out a ray from the surface to the light that tests collisions with objects, the photon ma is queried for shadow photons. If none are present, then the object has a clear line of sight to the light source and additional calculations can be avoided.
- To optimize image quality, particularly of caustics, Jensen recommends use of a cone filter. Essentially, the filter gives weight to photons' contributions to radiance depending on how far they are from ray-surface intersections. Thi can produce sharper images.
- Image space photon mapping achieves real-time performance by computing the first and last scattering using a GPU rasterizer.

Variations

- Although photon mapping was designed to work primarily with ray tracers, it can also be extended for use with scanline renderers.

External links

- Global Illumination using Photon Maps
- *Realistic Image Synthesis Using Photon Mapping* ISBN 1-56881-147-0
- Photon mapping introduction from Worcester Polytechnic Institute
- *Bias in Rendering*
- Siggraph Paper

Retrieved from "https://en.wikipedia.org/w/index.php?title=Photon_mapping&oldid=832032825"