Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Français
Português
Русский
中文
Edit links

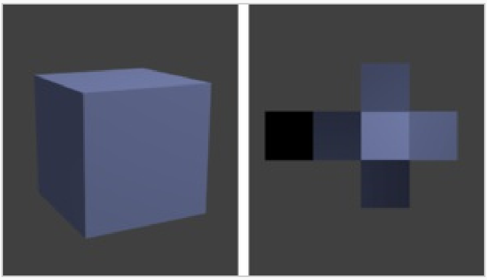
Lightmap

From Wikipedia, the free encyclopedia

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(January 2010)*

This article **possibly contains original research**. Please improve it by verifying the claims made and adding inline citations. Statements consisting only of original research should be removed. *(January 2010)*

A **lightmap** is a data structure which contains the brightness of surfaces in 3d graphics applications such as video games. Lightmaps are pre-computed, and normally used for static objects only. They are particularly suited to urban and indoor environments with large planar surfaces.



Cube with a simple texture lightmap (shown on the right) applied.

Contents [hide]

History [edit]

Quake was the first computer game to use lightmaps to augment rendering. Before lightmaps were invented, realtime applications relied purely on Gouraud shading to interpolate vertex lighting for surfaces. This only allowed low frequency lighting information, and could create clipping artifacts close to the camera without perspective-correct interpolation. Discontinuity Meshing was sometimes used especially with radiosity solutions to adaptively improve the resolution of vertex lighting information, however the additional cost in primitive setup for realtime rasterization was generally prohibitive. John Carmack's Quake software rasterizer used surface caching to apply lighting calculations in texture space once when polygons initially appear within the viewing frustum (effectively creating temporary 'lit' versions of the currently visible textures as the viewer negotiated the scene).

As consumer 3d graphics hardware capable of multitexturing became more popular, engines began to combine light-maps in realtime as a secondary "multiply-blend" texture layer.

Limitations [edit]

Lightmaps are scaled using lumels (lumination elements). The smaller the lumels, and therefore the more of them present in the image, the higher the resolution (and visual quality), at the price of game speed and performance. For example, a lightmap scale of 4 lumels per world unit would give a lower quality than a scale of 16 lumels per world unit. It's important for level designers to make a compromise between performance and quality. If a high lumel per world unit scale is set too frequently then the game may use up too much system resources and affect the performance negatively. Lumel size can also be limited by the amount of disk storage space or download time (for the map) or texture memory (for the 3D engine itself) available, although some games attempt to pack multiple lightmaps together to help circumvent these limitations.

Lightmap resolution and scale are two different things. The resolution is the area, in pixels, available for storing one or more surface's lightmaps. The number of individual surfaces that can fit on a lightmap is determined by the scale. Lower scale values mean higher quality and more space taken on a lightmap. Higher scale values mean lower quality and less space taken. A surface can have a lightmap that has the same area, so a 1:1 ratio, or smaller, so the lightmap is stretched to fit.

Lightmaps in games are usually colored texture maps, or per vertex colors. They are usually flat, without information about the light's direction, whilst some game engines use multiple lightmaps to provide approximate directional information to combine with normal-maps. Lightmaps may also store separate precalculated components of lighting information for semi-dynamic lighting with shaders, such as ambient-occlusion & sunlight shadowing.

Creation [edit]

When creating lightmaps, any lighting model may be used, because the lighting is entirely precomputed and real-time performance is not always a necessity. A variety of techniques including ambient occlusion, direct lighting with sampled shadow edges, and full radiosity bounce light solutions are typically used. Modern 3d packages include specific plugins for applying light-map UV-coordinates, atlas-ing multiple surfaces into single texture sheets, and rendering the maps themselves. Alternatively game engine pipelines may include custom lightmap creation tools. An additional consideration is the use of compressed DXT textures which are subject to blocking artifacts - individual surfaces must not collide on 4x4 texel chunks for best results.

In all cases, soft shadows for static geometry are possible if simple occlusion tests (such as basic ray-tracing) are used to determine which lumels are visible to the light. However, the actual softness of the shadows is determined by how the engine interpolates the lumel data across a surface, and can result in a pixelated look if the lumels are too large. See texture filtering.

Lightmaps can also be calculated in real-time[1] for good quality colored lighting effects that are not prone to the defects of Gouraud shading, although shadow creation must still be done using another method such as stencil shadow volumes or shadow mapping, as real-time ray-tracing is still too slow to perform on modern hardware in most 3D engines.

Photon mapping can be used to calculate global illumination for lightmaps.

See also [edit]

- Discontinuity Meshing
- Environment map
- SSAO
- Shader
- Texture mapping

References [edit]

1. ^ Dynamic Lightmaps in OpenGL. Retrieved Jul. 07, 2014.

Categories: Video game design