

VRay Features

The VRay raytracer is available in two packages, the **Basic Package** and the **Advanced Package**. The **Basic Package** with its balanced set of features and low price is ideal for students and amateur artists. The **Advanced Package** includes several additional capabilities and is aimed at professional artists.

Features included in the Basic Package

- ? True raytraced reflections and refractions
- ? Glossy reflections and refractions
- ? Area shadows (soft shadows). Includes box and sphere emitters
- ? Indirect Illumination (global illumination, global lighting). Different approaches include direct computation (brute force), and irradiance maps.
- ? Motion Blur. Includes Quasi-Monte Carlo sampling approach
- ? Depth-Of-Field camera effect.
- ? Antialiasing. Includes fixed, simple 2-level and adaptive approaches.
- ? Caustics
- ? G-Buffer (RGBA, material/object ID, Z-buffer, velocity etc.)
- ? An optimized material for faster rendering and special effects like translucency

Features included in the Advanced Package

Includes all basic features plus:

- ? G-buffer based antialiasing.
- ? Photon mapping
- ? Reusable irradiance maps (save and load support). Incremental sampling for fly-through animations.
- ? Reusable caustic and global photon maps (save and load support).
- ? Motion blur with analytic sampling
- ? True HDRI support. Includes *.hdr, *.rad image loader with proper texture coordinates handling for both cubic and angular maps. Map your images directly without distortions or cropping.
- ? Native area lights for physically correct illumination.
- ? Native material for more physically accurate and faster materials calculations.
- ? Distributed rendering for utilizing all of your studios computers. Based only on TCP/IP enables one to connect through the Internet as well.
- ? Different camera types: fish-eye, spherical, cylindrical and cubic cameras
- ? Displacement mapping. Includes a fast 2D bitmap algorithm as well as true 3D displacement mapping.

Distributed rendering

Introduction

Distributed rendering is a technique for distributing a single render job within a single frame across many computers in a network. There are different approaches of doing this but the main concept is to reduce the render times by dividing different parts of the rendering pipeline and giving each participant different parts of the job. The most common way to do this is to divide the frame to be rendered into small regions(buckets) and give each machine to render a number of them. Then get the results and combine them into the final image.

VRay organization

VRay supports DR. It divides the frame into regions and spreads them across the participants in the distributed rendering. This is done thoroughly through TCP/IP protocol which is the standard protocol of the Internet and thus the most common protocol that is supported by the hardware. VRay itself does not need additional file or directory sharing (note that you may actually need some file/directory sharing for the bitmaps or other additional files used during rendering). The distribution management is divided into **Render Clients** and **Render Servers**.

Render client is the computer that the user is currently using. It divides the frame into rendering regions and spreads it across the **Render Servers**. It then manages the rendering process by monitoring which servers need information to continue rendering and retrieves the processed results. In the Render Client there is an interface to manage the Render Servers - to include or exclude them from the rendering pipeline - as well as to control their state. When a bucket is rendered the Render Client collects the result, displays it and sends another bucket to the Render Server (if there are more rendering regions to be processed).

Render server is the computer that is part of the so-called render farm - a machine that receives a part of the frame - the bucket - renders it and sends it back to the client. Its state is monitored and controlled by the Render Client.