

The Wayback Machine - <https://web.archive.org/web/20220307082443/https://en.wikipedia.org/wiki/S...>

WIKIPEDIA

# Software rendering

---

**Software rendering** is the process of generating an image from a model by means of computer software. In the context of computer graphics rendering, software rendering refers to a rendering process that is not dependent upon graphics hardware ASICs, such as a graphics card. The rendering takes place entirely in the CPU. Rendering everything with the (general-purpose) CPU has the main advantage that it is not restricted to the (limited) capabilities of graphics hardware, but the disadvantage is that more semiconductors are needed to obtain the same speed.

Rendering is used in architecture, simulators, video games, movies and television visual effects and design visualization. Rendering is the last step in an animation process, and gives the final appearance to the models and animation with visual effects such as shading, texture-mapping, shadows, reflections and motion blur.<sup>[1]</sup> Rendering can be split into two main categories: real-time rendering (also known as online rendering), and pre-rendering (also called offline rendering). Real-time rendering is used to interactively render a scene, like in 3D computer games, and generally each frame must be rendered in a few milliseconds. Offline rendering is used to create realistic images and movies, where each frame can take hours or days to complete, or for debugging of complex graphics code by programmers.



Software renderer running on a device without a GPU.

## Contents

---

### **Real-time software rendering**

Software fallback

### **Pre-rendering**

### **See also**

### **References**

## Real-time software rendering

---

For real-time rendering the focus is on performance. The earliest texture mapped real-time software renderers for PCs used many tricks to create the illusion of 3D geometry (true 3D was limited to flat or Gouraud-shaded polygons employed mainly in flight simulators.) *Ultima Underworld*, for example, allowed a limited form of looking up and down, slanted floors, and rooms over rooms, but resorted to sprites for all detailed objects. The technology used in these games is currently categorized as 2.5D.

One of the first games architecturally similar to modern 3D titles, allowing full 6DoF, was *Descent*, which featured 3D models entirely made from bitmap textured triangular polygons. Voxel-based graphics also gained popularity for fast and relatively detailed terrain rendering, as in *Delta Force*, but popular fixed-function hardware eventually made its use impossible. *Quake* features an efficient software renderer by Michael Abrash and John Carmack. With its popularity, *Quake* and other polygonal 3D games of that

time helped the sales of graphics cards, and more games started using hardware APIs like DirectX and OpenGL. Though software rendering fell off as a primary rendering technology, many games well into the 2000s still had a software renderer as a fallback, *Unreal* and *Unreal Tournament* for instance, feature software renderers able to produce enjoyable quality and performance on CPUs of that period. One of the last AAA games without a hardware renderer was *Outcast*, which featured advanced voxel technology but also texture filtering and bump mapping as found on graphics hardware.

In the video game console and arcade game markets, the evolution of 3D was more abrupt, as they had always relied heavily on single-purpose chipsets. 16 bit consoles gained RISC accelerator cartridges in games such as *StarFox* and *Virtua Racing* which implemented software rendering through tailored instruction sets. The Jaguar and 3DO were the first consoles to ship with 3D hardware, but it wasn't until the PlayStation that such features came to be used in most games.

Games for children and casual gamers (who use outdated systems or systems primarily meant for office applications) during the late 1990s to early 2000s typically used a software renderer as a fallback. For example, *Toy Story 2: Buzz Lightyear to the Rescue* has a choice of selecting either hardware or software rendering before playing the game, while others like *Half-Life* default to software mode and can be adjusted to use OpenGL or DirectX in the Options menu. Some 3D modeling software also features software renderers for visualization. And finally the emulation and verification of hardware also requires a software renderer. An example of the latter is the Direct3D reference rasterizer.

But even for high-end graphics, the 'art' of software rendering hasn't completely died out. While early graphics cards were much faster than software renderers and originally had better quality and more features, it restricted the developer to 'fixed-function' pixel processing. Quickly there came a need for diversification of the looks of games. Software rendering has no restrictions because an arbitrary program is executed. So graphics cards reintroduced this programmability, by executing small programs per vertex and per pixel/fragment, also known as shaders. Shader languages, such as High Level Shader Language (HLSL) for DirectX or the OpenGL Shading Language (GLSL), are C-like programming languages for shaders and start to show some resemblance with (arbitrary function) software rendering.

Since the adoption of graphics hardware as the primary means for real-time rendering, CPU performance has grown steadily as ever. This allowed for new software rendering technologies to emerge. Although largely overshadowed by the performance of hardware rendering, some modern real-time software renderers manage to combine a broad feature set and reasonable performance (for a software renderer), by making use of specialized dynamic compilation and advanced instruction set extensions like SSE. Although nowadays the dominance of hardware rendering over software rendering is undisputed because of unparalleled performance, features, and continuing innovation, some believe that CPUs and GPUs will converge one way or another and the line between software and hardware rendering will fade.<sup>[2]</sup>

## Software fallback

For various reasons such as hardware failure, broken drivers, emulation, quality assurance, software programming, hardware design, and hardware limitations, it is sometimes useful to let the CPU assume some or all functions in a graphics pipeline.

As a result, there are a number of general-purpose software packages capable of replacing or augmenting an existing hardware graphical accelerator, including:

- RAD Game Tools' Pixomatic, sold as middleware intended for static linking inside D3D 7-9 client software.
- SwiftShader, a library sold as middleware intended for bundling with D3D9 & OpenGL ES 2 client software.
- The swrast, softpipe, & LLVMpipe renderers inside Mesa work as a shim at the

system level to emulate an OpenGL 1.4–3.2 hardware device.

- WARP, provided since Windows Vista by Microsoft, which works at the system level to provide fast D3D 9.1 and above emulation. This is in addition to the extremely slow software-based reference rasterizer Microsoft has always provided to developers.
- The Apple software renderer in CGL, provided in Mac OS X by Apple, which works at the system level to provide fast OpenGL 1.1–4.1 emulation.

## Pre-rendering

---

Contrary to real-time rendering, performance is only of second priority with pre-rendering. It is used mainly in the film industry to create high-quality renderings of lifelike scenes. Many special effects in today's movies are entirely or partially created by computer graphics. For example, the character of Gollum in the Peter Jackson *The Lord of the Rings* films is completely computer-generated imagery (CGI). Also for animation movies, CGI is gaining popularity. Most notably Pixar has produced a series of movies such as *Toy Story* and *Finding Nemo*, and the Blender Foundation the world's first open movie *Elephants Dream*.

Because of the need for very high-quality and diversity of effects, offline rendering requires a lot of flexibility. Even though commercial real-time graphics hardware is becoming higher in quality and more programmable by the day, most photorealistic CGI still requires software rendering. Pixar's RenderMan, for example, allows shaders of unlimited length and complexity, demanding a general-purpose processor. Techniques for high realism like raytracing and global illumination are also inherently unsuited for hardware implementation and in most cases are realized purely in software.

## See also

---

- 3D computer graphics
- Z-buffering

## References

---

1. "LIVE Design - Interactive Visualizations | Autodesk" (<https://web.archive.org/web/20220307082443/https://web.archive.org/web/20140221045119/http://usa.autodesk.com/adsk/servlet/item?id=17940930&siteID=123112>). Archived from the original (<https://web.archive.org/web/20220307082443/http://usa.autodesk.com/adsk/servlet/item?id=17940930&siteID=123112>) on February 21, 2014. Retrieved 2016-08-20.
2. Valich, Theo (2012-12-13). "Tim Sweeney, Part 2: "DirectX 10 is the last relevant graphics API" | TG Daily" (<https://web.archive.org/web/20220307082443/https://www.tgdaily.com/business/tim-sweeney-part-2-directx-10-is-the-last-relevant-graphics-api/>). *TG Daily*. Archived (<https://web.archive.org/web/20220307082443/https://web.archive.org/web/20160304145146/http://www.tgdaily.com/business-and-law/features/36410-tim-sweeney-part-2-directx-10-is-the-last-relevant-graphics-api>) from the original on March 4, 2016. Retrieved 2016-11-07.

---

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Software\\_rendering&oldid=1059791079](https://en.wikipedia.org/w/index.php?title=Software_rendering&oldid=1059791079)"

---

**This page was last edited on 11 December 2021, at 17:11 (UTC).**

Text is available under the Creative Commons Attribution-ShareAlike License 3.0; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.