


POV-Ray

The **Persistence of Vision Ray Tracer**, most commonly acronymed as **POV-Ray**, is a [cross-platform ray-tracing](#) program that generates images from a [text-based scene description](#). It was originally based on DKBTrace, written by David Kirk Buck and Aaron A. Collins for [Amiga](#) computers. There are also influences from the earlier [Polyray](#)^[6] raytracer because of contributions from its author, Alexander Enzmann. POV-Ray is [free and open-source software](#), with the [source code](#) available under the [AGPLv3](#).

POV-Ray



Original author(s)	David Kirk Buck, Aaron A. Collins, Alexander Enzmann
Developer(s)	The POV-Team
Initial release	July 29, 1991 ^{[1][2]}
Stable release	3.7.0 (November 9, 2013 ^[3]) [±] (https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/POV-Ray&action=edit)
Preview release	v3.8.0-alpha.10064738 (February 19, 2019 ^[4]) [±] (https://en.wikipedia.org/w/index.php?title=Template:Latest_preview_software_release/POV-Ray&action=edit)
Repository	github.com/POV-Ray/povray (https://github.com/POV-Ray/povray) 
Written in	C++
Operating system	Cross-platform
Type	Ray tracer
License	AGPLv3 ^[5]
Website	www.povray.org (http://www.povray.org)

History



A vase on a pedestal rendered with DKBTrace 2.12

Sometime in the 1980s, David Kirk Buck downloaded the [source code](#) for a [Unix](#) ray tracer to his [Amiga](#). He experimented with it for a while and eventually decided to write his own ray tracer named DKBTrace after his initials. He posted it to the "You Can Call Me Ray" [bulletin board system](#) (BBS) in Chicago, thinking others might be interested in it. In 1987, Aaron A. Collins downloaded DKBTrace and began working on an [x86](#) port of it. He and David Buck collaborated to add several more features.

When the program proved to be more popular than anticipated, they could not keep up with demand for more features. Thus, in July 1991, David turned over the project to a team of programmers working in the "GraphDev" [forum](#) on [CompuServe](#). At the same time, David felt that it was inappropriate to use his initials on a program he no longer maintained. The name "STAR-Light" (Software Taskforce on Animation and Rendering) was initially used, but eventually the name became "PV-Ray", and then ultimately "POV-Ray" (Persistence of Vision Ray Tracer),^[2] a name inspired by [Dalí's](#) painting, *The Persistence of Memory*.^{[7][8]}

Features of the application, and a summary of its history, are discussed in a February 2008 interview with David Kirk Buck and Chris Cason on episode 24 of [FLOSS Weekly](#).^[8]

Features



Glass scene rendered by POV-Ray demonstrating [radiosity](#), [photon mapping](#), [focal blur](#), and other photorealistic capabilities. (Image

POV-Ray has matured substantially since it was created. Recent versions of the software include the following features:

- a [Turing-complete](#) scene description language (SDL) that supports macros and loops^[9]
- a library of ready-made scenes, textures, and objects
- support for a number of [geometric primitives](#) and [constructive solid geometry](#)
- several kinds of [light sources](#)
- atmospheric effects such as [fog](#) and *media* ([smoke](#), [clouds](#))
- [reflections](#), [refractions](#), and light [caustics](#) using [photon mapping](#)
- surface patterns such as [wrinkles](#), bumps, and [ripples](#), for use in [procedural textures](#) and [bump mapping](#)
- [radiosity](#)
- support for [textures](#) and rendered output in many image formats, including [TGA](#), [PNG](#), and [JPEG](#), among others
- extensive [user documentation](#)

One of POV-Ray's main attractions is its large collection of third-party-made assets and tools. A large number of tools, textures, models, scenes, and tutorials can be found on the web. It is also a useful reference for those wanting to learn how [ray tracing](#) and related 3D geometry and computer graphics algorithms work.

Current version

The current official version of POV-Ray is 3.7. This version introduces:

- support for [symmetric multiprocessing](#) (SMP), to allow the renderer to take advantage of [multiple processors](#)
- support for [high-dynamic-range imaging](#) (HDRI), including the [OpenEXR](#) and [radiance](#) file formats
- improved [bounding](#) using [BSP](#) trees

Some of the main introduced features of the previous release (3.6) are:

- extending [UV mapping](#) to more primitives
- adding 16- and 32-bit integer data to a density file

- improving [64-bit compatibility](#)

In July 2006, [Intel Corporation](#) started using the [beta version](#) of 3.7 to demonstrate their new [dual-core Conroe processor](#) due to the efficiency of the SMP (symmetric multiprocessing) implementation.

Primitives



Rendering of the [Venn diagram](#) of four spheres created with [constructive solid geometry](#), or CSG. The source is [on the description page](#) of the image.



Some colored [dice](#) rendered in POV-Ray. [CSG](#), [refraction](#) and [focal blur](#) are demonstrated.

POV-Ray, in addition to standard [3D geometric](#) shapes like [tori](#), [spheres](#), and [heightfields](#), supports mathematically defined [primitives](#) such as the [isosurface](#) (a finite approximation of an arbitrary function), the [polynomial](#) primitive (an [infinite](#) object defined by a [15th order or lower polynomial](#)), the [julia fractal](#) (a 3-dimensional slice of a [4-dimensional](#) fractal), the [superquadratic ellipsoid](#) (an intermediate between a sphere and a cube), and the [parametric](#) primitive (using equations that represent its surface, rather than its interior).

POV-Ray internally represents objects using their mathematical definitions; all POV-Ray primitive objects can be described by [mathematical functions](#). This is different from many computer

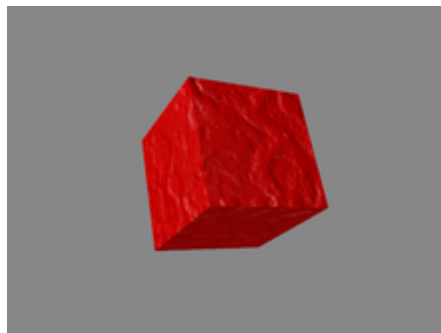
programs that include 3D models, which typically use [triangle meshes](#) to compose all the objects in a scene.

This fact provides POV-Ray with several advantages and disadvantages over other rendering and modeling systems; POV-Ray primitives are more accurate than their polygonal counterparts: objects that can be described in terms of spheres, planar surfaces, cylinders, tori, and the like, are perfectly smooth and mathematically accurate in POV-Ray renderings, whereas polygonal *artifacts* may be visible in mesh-based modeling software. POV-Ray primitives are also simpler to define than most of their polygonal counterparts, e.g., in POV-Ray, a [sphere](#) is described simply by its center and radius; in a mesh-based environment, a sphere must be described by a multitude of small connected polygons (usually [quads](#) or [triangles](#)).

On the other hand, script-based primitive modeling is not always a practical method to create certain objects, such as realistic characters or complex man-made artifacts like cars. Those objects can and should be created first in mesh-based modeling applications such as [Wings 3D](#) and [Blender](#), and then they can be converted to POV-Ray's own mesh format.

Examples of the scene description language

The following is an example of the scene description language used by POV-Ray to describe a scene to render. It demonstrates the use of a background colour, camera, lights, a simple box shape having a surface normal and finish, and the transforming effects of rotation.



POV-Ray image output based on the script

```
#version 3.6;  
// Includes a separate file defining a number of common colours  
#include "colors.inc"  
global_settings { assumed_gamma 1.0 }  
  
// Sets a background colour for the image (dark grey)
```

```

background { color rgb <0.25, 0.25, 0.25> }

// Places a camera
// direction: Sets, among other things, the field of view of the
camera
// right: Sets the aspect ratio of the image
// look_at: Tells the camera where to look
camera { location <0.0, 0.5, -4.0>
          direction 1.5*z
          right      x*image_width/image_height
          look_at    <0.0, 0.0, 0.0> }

// Places a light source
// color: Sets the color of the light source (white)
// translate: Moves the light source to a desired location
light_source { <0, 0, 0>
               color rgb <1, 1, 1>
               translate <-5, 5, -5> }

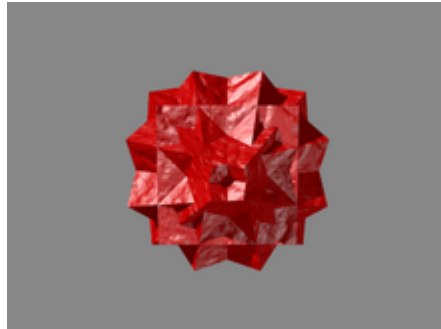
// Places another light source
// color: Sets the color of the light source (dark grey)
// translate: Moves the light source to a desired location
light_source { <0, 0, 0>
               color rgb <0.25, 0.25, 0.25>
               translate <6, -6, -6> }

// Sets a box
// pigment: Sets a color for the box ("Red" as defined in
"colors.inc")
// finish: Sets how the surface of the box reflects light
// normal: Sets a bumpiness for the box using the "agate" in-built
model
// rotate: Rotates the box
box { <-0.5, -0.5, -0.5>,
      <0.5, 0.5, 0.5>
      texture { pigment { color Red }
                finish { specular 0.6 }
                normal { agate 0.25 scale 1/2 }
      }

```

```
    }  
    rotate <45,46,47> }
```

The following script fragment shows the use of variable declaration, assignment, comparison and the while loop construct:



POV-Ray image output based on the script

```
#declare the_angle = 0;  
  
#while (the_angle < 360)  
    box {      <-0.5, -0.5, -0.5>  
              <0.5, 0.5, 0.5>  
              texture { pigment { color Red }  
                        finish { specular 0.6 }  
                        normal { agate 0.25 scale 1/2 } }  
              rotate the_angle }  
    #declare the_angle = the_angle + 45;  
#end
```

Modeling

The POV-Ray program itself does not include a [modeling](#) feature; it is essentially a pure renderer with a sophisticated model description language. To accompany this feature set, third parties have developed a large variety of modeling software, some specialized for POV-Ray, others supporting import and export of its data structures, including the free and open-source 3D creation suite [Blender \(https://en.blender.org/index.php/Extensions:2.6/Py/Scripts/Render/POV-Ray\)](https://en.blender.org/index.php/Extensions:2.6/Py/Scripts/Render/POV-Ray) .

A number of additional POV-Ray compatible modelers are linked from [Povray.org: Modelling Programs \(http://www.povray.org/resources/links/3D_Programs/Modelling_Programs/\)](http://www.povray.org/resources/links/3D_Programs/Modelling_Programs/) .

In 2007, POV-Ray acquired the rights to [Moray \(http://www.stmuc.com/moray/medown.html\)](http://www.stmuc.com/moray/medown.html) ^[10] an interactive 3-D modeling program long used with POV-Ray. However, as of December 2016, Moray development is stalled.

Software

Development and maintenance

Official modifications to the POV-Ray source tree are done and/or approved by the POV-Team. Most patch submission and/or bug reporting is done in the POV-Ray newsgroups on the news.povray.org ([nntp://news.povray.org/](http://news.povray.org/)) [news](#) server (with a [Web interface also available \(http://news.povray.org/g/groups/\)](http://news.povray.org/g/groups/)). Since POV-Ray's source is available there are unofficial [forks](#) and patched versions of POV-Ray available from third parties; however, these are not officially supported by the POV-Team.

Official POV-Ray versions currently do not support [shader](#) plug-ins.^[11] Some features, like [radiosity](#) and [splines](#) are still in development and may be subject to syntactical change.

Platform support

POV-Ray 3.6 is distributed in compiled format for [Macintosh](#), [Windows](#) and [Linux](#). Support for [Intel Macs](#) is not available in the [Macintosh](#) version, but since [Mac OS X](#) is a version of [Unix](#) the Linux version can be [compiled](#) on it.

The 3.7 versions with [SMP](#) support are officially supported for Windows and Linux. Unofficial [Macintosh](#) versions for v3.7 can be found.^[12]

POV-Ray can be [ported](#) to any platform which has a compatible [C++](#) compiler.

Licensing

Originally, POV-Ray was distributed under its own *POV-Ray License*. Namely, the POV-Ray 3.6 Distribution License^[13] and the POV-Ray 3.6 Source License,^[14] which permitted free distribution of the program source code and binaries, but restricts commercial distribution and the creation of derivative works other than fully functional versions of POV-Ray.

Although the [source code](#) of older versions is available for modification, due to the above 3.6 and prior license restrictions, it was not [open source](#) or [free software](#) according to the [OSI](#) or the [FSF](#) definition of the term. This was a problem as [source code](#) exchange with the greater [FOSS](#) ecosystem was impossible due to [License incompatibility](#) with [copyleft licenses](#).

One of the reasons that POV-Ray was not originally licensed under the free software [GNU General Public License](#) (GPL), or other open source licenses, is that POV-Ray was developed before the GPL-style licenses became widely used; the developers wrote their own license for the release of POV-Ray, and contributors to the software worked under the assumption their contributions would be licensed under the POV-Ray 3.6 Licenses.

In 2013, with version 3.7, POV-Ray was [relicensed](#) under the [Affero General Public License](#) version 3 (or later).^[15] Thus POV-Ray is since then [free software](#) according to the [FSF](#) definition and also [open source software](#) according to the [Open Source Definition](#).

See also

Wikimedia Commons has media related to [POV-Ray](#).

- [Blender](#) – a free and open-source software program for 3D modeling, animation, and rendering
- [Kerkythea](#) – a freeware ray-tracing program with enhanced [Sketchup](#) compatibility
- [Sunflow](#) – an open-source rendering system for photo-realistic image synthesis, written in Java
- [YafaRay](#) – a popular free open-source ray-tracing engine

References

1. "POV-Ray: Documentation: 1.1.5.3 A Historic 'Version History'" (<http://www.povray.org/documentation/view/3.6.1/10/>) . *povray.org*. Retrieved 2018-12-19.
2. "POV-Ray: News" (<http://www.povray.org/news/index.php#323>) . *povray.org*. Retrieved 2018-12-19.
3. "Download" (<http://www.povray.org/download/>) . POV-Ray. 2013-11-09. Retrieved 2013-11-09.
4. "Releases" (<https://github.com/POV-Ray/povray/releases>) . POV-Ray. 2019-02-19. Retrieved 2019-02-19.
5. "POV-Ray License" (<http://www.povray.org/povlegal.html>) . *povray.org*. Retrieved 2014-05-05.
6. "Polyray v1.7" (<http://paulbourke.net/dataformats/polyray/>) . Alexander R. Enzmann. Retrieved 2016-07-05.
7. "Documentation: 1.1.5 The Early History of" (<http://www.povray.org/documentation/view/3.6.0/7/>) . POV-Ray. Retrieved 2014-05-05.
8. "The TWiT Netcast Network with Leo Laporte" (<http://twit.tv/floss24>) . Twit.tv. 2008-02-07. Retrieved 2014-05-05.
9. [Paul Bourke: Supershape in 3D](http://paulbourke.net/geometry/supershape/) (<http://paulbourke.net/geometry/supershape/>) are examples of POV-Ray images made with very short code

10. "News: Moray Announcement" (<http://www.povray.org/news/moray-announcement.php>) . POV-Ray. 2007-02-01. Retrieved 2014-05-05.
11. for such an implementation, see e.g., <http://www.aetec.ee/fv/vkhomep.nsf/pages/povman2> Archived (<https://web.archive.org/web/20070207122812/http://www.aetec.ee/fv/vkhomep.nsf/pages/povman2>) 2007-02-07 at the [Wayback Machine](#)
12. [povrayunofficial_mac](http://megapov.inetart.net/povrayunofficial_mac/index.html) (http://megapov.inetart.net/povrayunofficial_mac/index.html) on megapov.inetart.net
13. "POV-Ray 3.6 Distribution License" (<http://www.povray.org/distribution-license-3.6.html>) . povray.org. Retrieved 2016-12-12.
14. "POV-Ray 3.6 Source License" (<http://www.povray.org/source-license-3.6.html>) . povray.org. Retrieved 2016-12-12.
15. Cason, Chris (8 November 2013). "Download POV-Ray 3.7.0" (<http://www.povray.org/download/>) . Retrieved 11 November 2013. "Starting with version 3.7, POV-Ray is released under the AGPL3 (or later) license and thus is Free Software according to the FSF definition. [...] Full source code is available, allowing users to build their own versions and for developers to incorporate portions or all of the POV-Ray source into their own software provided it is distributed under a compatible license (for example, the AGPL3 or – at their option – any later version)."

External links

Wikimedia Commons has media related to ***POV-Ray***.

- [Official website](http://www.povray.org) (<http://www.povray.org>)
 - [POV-Ray](https://curlie.org/Computers/Software/Graphics/3D/Animation_and_Design_Tools/POV-Ray) (https://curlie.org/Computers/Software/Graphics/3D/Animation_and_Design_Tools/POV-Ray) at [Curlie](#)
-
-