

VOP Manual

Wolfgang Thaller

March 24, 2002

Contents

1	License	3
2	Contact Information	4
3	Introduction	5
3.1	What is VOP?	5
3.2	What does VOP stand for?	5
4	Using VOP	6
4.1	Installing (MacOS X)	6
4.2	Installing (Windows)	6
4.3	Installing (Linux)	7
4.4	Command Line Options	7
4.4.1	--include-path (-I)	7
4.4.2	--include-file (-i)	7
4.4.3	--tab-size (-t)	7
4.4.4	--clock (-c)	8
4.4.5	--width (-w)	8
4.4.6	--height (-h)	8
4.4.7	--wireframe (-W)	8
4.4.8	--one-frame-only (-1)	8
4.4.9	--far-clipping-plane (-f)	8
4.4.10	--near-clipping-plane (-n)	8
4.4.11	--output-file (-o)	8
4.5	Interactive Use	8
5	VOP Reference	10
5.1	Extensions	10
5.2	Include Files	10
5.3	Objects	10
5.4	Pigment	11
5.5	Texture	11
5.6	Ignored Blocks	11
5.7	Camera	12

<i>CONTENTS</i>	2
5.8 Functions	12
5.9 Spline Objects	13
5.10 #-Directives	13
5.11 Other differences and omissions	13
6 Release History	14

Chapter 1

License

VOP is freeware. You can use the program for free. Images you create with the help of VOP are your property.

There is absolutely no guarantee that VOP will do what you want it to do. I take no responsibility whatsoever for any damage that may arise directly or indirectly from the use of the program.

Redistribution of unchanged, complete versions is allowed, as long as it is clearly stated that the program is available free of charge from the internet.

The source code is included for documentation purposes, for porting VOP to other platforms, and for educational purposes. For all other use of the source code, please ask for permission.

Please note that the above terms are open for discussion, especially if you want to contribute code to VOP.

VOP is ©2002 Wolfgang Thaller. All rights reserved.

Chapter 2

Contact Information

VOP is being developed by Wolfgang Thaller. You can reach me via e-mail at:

wolfgang.thaller@gmx.net

VOP's home on the Web is at:

<http://www.uni-graz.at/imawww/thaller/wolfgang/vop-intro.html>

Chapter 3

Introduction

3.1 What is VOP?

VOP reads POV-Ray¹ scene description files (*.pov) and displays them using OpenGL. The focus is on speed instead of image quality. By using completely different techniques to create low-quality images quickly, VOP can give you a quick "vision" of what you will get when you use POV-Ray to render your scene. For example, VOP doesn't even attempt to render nice surface textures: all objects look like they are made of uniformly colored plastic.

VOP is not created, maintained or endorsed in any way by the POV-Ray Team. It does not share any source code with POV-Ray. The only thing the two programs have in common is that VOP tries to read POV-Ray's file format. Do not bother the POV-Ray Team with questions related to VOP.

VOP is a work in progress. It is far from complete, but I think it can already be useful in some cases.

3.2 What does VOP stand for?

Well, originally, it stood for POV, backwards: it gives you a Vision of a .POV file, a "Vision Of Persistence". The name sounded better when I started to write the program (I had to decide about the name for the directory on my hard disk), and now I'm stuck with it.

¹ The Persistence of Vision Raytracer (POV-Ray) is an excellent freeware raytracer available at www.povray.org

Chapter 4

Using VOP

4.1 Installing (MacOS X)

As the MacOS port is very new, it is still a bit dirty. It has no proper Mac-style user interface, you'll have to mess with the UNIX command line — read on for instructions, or wait until I have added a proper user interface. I apologize for the inconvenience.

VOP is intended to be run from the command line. However, for technical reasons¹, it is packaged as if it was a double-clickable application. If you double-click it, nothing will happen.

Open a Terminal window instead. Tell VOP where to find its include files (replace ... by the appropriate UNIX-style pathname):

```
setenv VOPINCLUDES /.../include
```

The actual program is inside the `VOP.app` application package, at `VOP.app/Contents/MacOS/VOP`. To start VOP, type

```
/.../VOP.app/Contents/MacOS/VOP /.../myscene.pov
```

Note that, for some reason, you have to specify the absolute path name for your pov file.

4.2 Installing (Windows)

You may put VOP wherever you like. You might want to add the directory where you put it to your `PATH`. VOP will also need to know the location of its include directory. If you don't want to specify an `--include-path` option (see below) everytime you use VOP, you can set the environment variable `VOPINCLUDES`. On Windows 95/98/ME, you can add

¹ If it isn't inside the `VOP.app` application package, it doesn't get its own menu bar and dock icon when it opens a window.

```
SET VOPINCLUDES=C:\...\include
```

to your `autoexec.bat`. On Windows NT/2000 (and probably XP), there is a dialog box for setting environment variables in the "System" Control Panel.

You can also use the above command everytime you open a command line window in which you are going to use VOP.

4.3 Installing (Linux)

Put the binary wherever you like, preferably in your path (for example in `/usr/local/bin`). Set the environment variable `VOPINCLUDES` to point to VOP's include directory:

```
export VOPINCLUDES=/.../include
```

The binary distribution of VOP was compiled on RedHat 7.1. It needs a working version of OpenGL (tested with slightly outdated nVIDIA XFree 4 drivers, and there's no reason why it shouldn't work with Mesa²) and GLU version 1.2 or later (GLU is usually included with OpenGL). VOP also requires `libgmp` version 3 and `libncurses` version 5.

Enjoy!

4.4 Command Line Options

All command line options have both a long form and a short form. If the options require a parameter, it is specified as in `--tab-size=8` or `-t 8`. Long options may be abbreviated as long as it is still clear which option is meant (e.g. `--tab=8`).

4.4.1 `--include-path (-I)`

Specify an additional directory to search for includes.

4.4.2 `--include-file (-i)`

Tell VOP that the specified file will be included. This is used only as a hint to improve parsing speed, it has no other effects and is not required.

4.4.3 `--tab-size (-t)`

Change the tab size that is used for outputting error messages. The default setting is 4 spaces.

²Mesa is a 3-D graphics library with an API which is very similar to that of OpenGL — see www.mesa3d.org

4.4.4 `--clock (-c)`

Specify a different clock value to start with.

4.4.5 `--width (-w)`

Specify the initial width of the window, in pixels. Default is 640.

4.4.6 `--height (-h)`

Specify the initial height of the window, in pixels. Default is 480.

4.4.7 `--wireframe (-W)`

Render the entire scene as a wireframe model without hidden line removal.

4.4.8 `--one-frame-only (-1)`

Exit immediately after rendering the first frame. Only useful for profiling purposes.

4.4.9 `--far-clipping-plane (-f)`

Used to specify the position of the "far clipping plane". Objects which are farther away from the camera than the specified distance will not be visible. Default is 1000.

If your scene is "big", you will have to increase this value in order to see all objects. Note that increasing this value decreases the rendering accuracy — if you set it to a billion, objects that are just a few units big will probably not render correctly.

4.4.10 `--near-clipping-plane (-n)`

Used to specify the position of the "near clipping plane". Objects which are closer to the camera than the specified distance will not be visible. Default is 0.1.

4.4.11 `--output-file (-o)`

Convert the scene to an Alias WaveFront .OBJ file. Unfortunately, CSG and textures are not yet supported.

4.5 Interactive Use

<RETURN> Reload changed source files.

<BACKSPACE>³ Restart animation from `clock == 0`.

<SPACE> Print frames per second.

<ESC> Exit the program.

If you right-click on the window, a menu appears that allows you to choose a camera⁴. The "default camera" is the camera that POV-Ray would use. In order to have additional entries in the menu, you have to **#declare** the cameras in your `.pov` file.

³Apple refers to this key as "DELETE"

⁴ On MacOS, Apple decided to put this menu in the menu bar and name it "Tools". The name doesn't fit at all, but I don't know how to change it (without taking the time to write a proper Mac-native user interface for MacOS X). Right-clicking doesn't work, even if you have a multi-button mouse.

Chapter 5

VOP Reference

VOP is intended to read POV-Ray scene description files unchanged, but for now it only understands a part of the POV-Ray scene description language. On the other hand, it supports a few extensions that would probably make no sense for POV-Ray.

By definition, if VOP and POV-Ray interpret some input differently, it's VOP's fault. For details on how the things listed below are supposed to work, see the POV-Ray documentation. If the behaviour differs, drop me an e-mail so that I can either fix it or add a note here.

5.1 Extensions

- The symbol `__VOP__` is defined to one.
Use this with `#ifdef` to make your scene suitable for both VOP and POV-Ray.
- ... some small things left out here ...

5.2 Include Files

VOP currently has a very incomplete set of include files. I don't want to violate POV-Ray's copyright conditions by copying and modifying the original POV-Ray include files.

5.3 Objects

VOP currently supports the following object types:

- Sphere
- Box

- Cylinder
- Cone
- Torus
- Prism
 - Linear sweep and linear splines only.
- Mesh
 - VOP's implementation of `mesh` is slow and I think the meshes are not shaded correctly at this time. Avoid them.
- Union
- CSG
 - Slow.
 - Merge
 - Intersection
 - Difference
- Planes
 - OpenGL doesn't understand the concept of infinite objects, so for now, a square of 100x100 units is used instead. When used for CSG, a plane is replaced by a box that is five times bigger than all other objects in the CSG taken together.

5.4 Pigment

Only simple solid color pigments are currently supported. If you specify `quick_color`, it has to be at the end of the pigment block. If VOP finds anything else in the body of a pigment, it generates a warning message and ignores the pigment.

The filter and transmit components of the color are ignored.

5.5 Texture

No texture patterns or other fancy things are supported. Use `#ifndef(_VOP_)` to prevent VOP from reporting an error.

5.6 Ignored Blocks

The contents of `normal`, `finish` and `interior` blocks is simply ignored. Light sources are ignored as well.

5.7 Camera

Only normal perspective cameras are supported.

5.8 Functions

- abs
- acos
- acosh
- asin
- asinh
- atanh
- cos
- cosh
- degrees
- exp
- ln
- log
- radians
- rand
- seed
- sin
- sinh
- sqrt
- tan
- tanh
- vdot
- vlength
- Splines Objects (see below)
- No user defined functions.

5.9 Spline Objects

- `linear_spline`
- `cubic_spline`
- `quadratic_spline` — slightly incorrect at this time

5.10 #-Directives

- `#include`
- `#declare`
- `#local`
- `#macro`
- `#if`, `#else`
- `#ifdef`
- `#ifndef`
- `#while`
- `#switch`
 - `#case`
 - `#else`
 - `#break`
 - `#range` is not yet supported.

5.11 Other differences and omissions

- Vector declares *always* require a semicolon at the end.
- Returning values via macro parameters is not supported.
- String declares and string expressions are not supported.

Chapter 6

Release History

Release 240302 (24th of March 2002)

- MacOS X binary now available
- Bugs fixed in CSG.
If you find a case where it doesn't work, tell me.
- More float functions supported.
- Splines (linear and cubic)
- Prism Object (linear spline & linear sweep only)
- quick_color pigments
- VOP can now be compiled using GHC 5.03 (thanks to Sven Panne).
- Improved handling of macros, declares and while loops.
- Some support for planes.

Release 280202 (28th of February 2002)

Initial Release