

POVLAB 4.0

A fast 3D graphic modeller for POV-Ray

by Denis Olivier

All rights reserved, national and international

Copyright © 1994-1997.

For more information please, contact :

ChromaGraphics

Denis Olivier

58, rue Robert Ballion

33700 Mérignac

France

Fax 1 : (33) 56 47 29 54 (always ready)

Fax 2 : (33) 46 05 67 43 (always ready)

Phone : (33) 56 47 29 54

Internet : dolivier@cyberstation.fr

Compuserve : >INTERNET:dolivier@cyberstation.fr

Home page : <http://www.povlab.org> (POVLAB support site and news).

Gallery page : <http://www.cyberstation.fr/~dolivier/author.html> (raytraced pictures).

Persistence of Vision Raytracer 3.0 (tm).

POV-Ray and Persistence of Vision Raytracer are trademarks of the POV-Ray Development Team.

Please, see **WHATSNEW.TXT** for what's new in this version !

Read **README.TXT** before running POVLAB !

1. TABLE OF CONTENTS

1.	TABLE OF CONTENTS.....	2
2.	GENERAL INFORMATION.....	6
3.	RUNNING POVLAB.....	7
3.1.	BEFORE RUNNING POVLAB.....	7
3.2.	DIRECTORIES	7
3.2.1.	<i>Image</i>	7
3.2.2.	<i>Hfield</i>	7
3.2.3.	<i>Maps</i>	7
3.2.4.	<i>Pattern</i>	7
3.2.5.	<i>Plugins</i>	7
3.2.6.	<i>Povlab</i>	7
3.2.7.	<i>Raw</i>	7
3.2.8.	<i>Scene</i>	7
3.2.9.	<i>System</i>	7
3.2.10.	<i>Texture</i>	7
3.2.11.	<i>Truetype</i>	7
3.3.	HARDWARE REQUIREMENTS	8
3.3.1.	<i>Config.sys</i>	8
3.3.2.	<i>Autoexec.bat</i>	8
3.3.3.	<i>Environment for dos</i>	8
3.4.	USING POVLAB AND WINDOWS 95.....	9
3.5.	RUNNING POVLAB	9
3.6.	POV-RAY AND POVLAB.....	9
3.7.	WINPOV AND POVLAB.....	9
3.8.	MEMORY WITH POVLAB.....	9
3.9.	TROUBLE-SHOOTING COMMON PROBLEMS	10
3.9.1.	<i>Using a log file</i>	11
4.	INTERFACE.....	12
4.1.	AXES.....	12
4.2.	BOTTOM TOOLBAR BUTTONS	12
4.2.1.	<i>General information</i>	12
4.2.2.	<i>Zooming out button (-)</i>	12
4.2.3.	<i>Zooming in button (+)</i>	12
4.2.4.	<i>Recentering button</i>	12
4.2.5.	<i>Zooming in a selected area</i>	12
4.2.6.	<i>Translating scene in viewport</i>	12
4.2.7.	<i>Full screen button (TOGGLE)</i>	12
4.2.8.	<i>Refresh the viewport (blue arrow)</i>	12
4.2.9.	<i>Saving session button (floppy)</i>	12
4.2.10.	<i>Fast rendering button (eye)</i>	13
4.2.11.	<i>Translate object</i>	13
4.2.12.	<i>Rotate object</i>	13
4.2.13.	<i>Scale object</i>	13
4.2.14.	<i>Modifying object button (?)</i>	13

4.3.	BOTTOM PARAMETER BAR	13
4.3.1.	General information	13
4.3.2.	Objects in POVLAB	13
5.	TOP MENU	15
5.1.	GENERAL	15
5.1.1.	Environment	15
5.1.2.	Memory	15
5.1.3.	Microsoft mouse	15
5.1.4.	Configure mouse	15
5.1.5.	Run image viewer	15
5.1.6.	Run scene viewer	15
5.1.7.	MS-DOS	15
5.1.8.	Quit	15
5.2.	FILE	15
5.2.1.	New	16
5.2.2.	Rename	16
5.2.3.	Load	16
5.2.4.	Save	16
5.2.5.	Save when exit	16
5.2.6.	-> .POV	16
5.2.7.	Set output tab	16
5.3.	DISPLAY	16
5.3.1.	Redraw viewport	16
5.3.2.	Redraw all viewports	16
5.3.3.	Display axes	16
5.3.4.	Display grid	16
5.3.5.	Normal display	16
5.3.6.	Fast display	16
5.3.7.	Cubic display	16
5.3.8.	Mesh precision	17
5.3.9.	Image control	17
5.4.	INTERFACE	17
5.4.1.	General information	17
5.5.	TOOLS	17
5.5.1.	General information	17
5.5.2.	Copy by translation	17
5.5.3.	Copy by rotation	17
5.5.4.	Extruder	17
5.5.5.	Object align	17
5.5.6.	Make 3D fonts	17
5.5.7.	Load Plugins	17
5.5.8.	Create a B-Spline paths	20
5.5.9.	Edit B-Spline	20
5.5.10.	Move vertex on B-Spline	20
5.5.11.	Delete vertex on B-Spline	20
5.5.12.	Add vertex on B-Spline	20
5.5.13.	Edit vertex on B-Spline	21
5.5.14.	Smooth vertices	21
5.6.	OBJECTS	21
5.6.1.	General information	21

5.6.2.	<i>Raw objects</i>	21
5.6.3.	<i>Height-field</i>	21
6.	TOOL BAR BUTTON ON LEFT/RIGHT	22
6.1.	CAMERA BUTTON.....	22
6.1.1.	<i>General information</i>	22
6.2.	LIGHT BUTTON	22
6.2.1.	<i>General information</i>	22
6.3.	MATERIAL BUTTON.....	22
6.3.1.	<i>General information</i>	22
6.3.2.	<i>Texture files</i>	22
6.3.3.	<i>Graphic tiles</i>	23
6.3.4.	<i>Texture button</i>	23
6.3.5.	<i>Acquire button</i>	23
6.3.6.	<i>Assign button</i>	23
6.3.7.	<i>Rotate button</i>	23
6.3.8.	<i>Finish button</i>	24
6.3.9.	<i>Mapping</i>	24
6.3.10.	<i>Setup</i>	24
6.4.	SELECTION BUTTON	24
6.4.1.	<i>General information</i>	24
6.4.2.	<i>Inverse</i>	24
6.4.3.	<i>Area</i>	24
6.4.4.	<i>Color</i>	24
6.4.5.	<i>By names</i>	24
6.4.6.	<i>Parameters</i>	25
6.5.	RENDER BUTTON	25
6.5.1.	<i>General information</i>	25
6.5.2.	<i>Final</i>	25
6.5.3.	<i>Region</i>	25
6.5.4.	<i>Background</i>	25
6.5.5.	<i>Palette</i>	25
6.5.6.	<i>Video</i>	25
6.5.7.	<i>View last</i>	25
6.5.8.	<i>Setup</i>	25
6.6.	MODIFY BUTTON	25
6.6.1.	<i>Rotate</i>	25
6.6.2.	<i>2D scale</i>	26
6.6.3.	<i>Move</i>	26
6.6.4.	<i>Copy</i>	26
6.6.5.	<i>Delete</i>	26
6.6.6.	<i>Center XYZ</i>	26
6.6.7.	<i>Reinit.</i>	26
6.6.8.	<i>Manual</i>	26
6.6.9.	<i>Color</i>	26
6.6.10.	<i>Smooth</i>	26
6.7.	DISPLAY BUTTON.....	26
6.7.1.	<i>General information</i>	26
6.7.2.	<i>Object</i>	26
6.7.3.	<i>None</i>	27
6.7.4.	<i>All</i>	27

6.7.5.	<i>Inverse</i>	27
6.7.6.	<i>Normal, quick and cubic</i>	27
6.7.7.	<i>Freeze</i>	27
6.7.8.	<i>Ignore</i>	27
6.8.	CSG BUTTON	27
6.8.1.	<i>General information</i>	27
6.8.2.	<i>How does it work ?</i>	27
6.8.3.	<i>Very important !</i>	27
6.8.4.	<i>Difference, union, intersection, merge buttons</i>	28
6.8.5.	<i>Remove CSG</i>	28
6.8.6.	<i>Show, hide</i>	28
7.	KEYBOARD ASSIGNMENTS	29
8.	MANAGING TEXTURES	30
8.1.	GENERALITIES	30
8.2.	COMMAND LINE	30
8.3.	WHEN POV-RAY ISN'T IN THE PATH.....	30
8.4.	HOW IT WORKS	31
8.5.	NAME THE TEXTURE THUMBNAILS.....	31
8.6.	ASSIGN NAMES AND THUMBNAILS TOGETHER.....	31
8.7.	OPTIONS USED IN MAKE THEM FOR POV-RAY AND TGA VIF	31
9.	FONT LIST	31
9.1.	POVLAB CONTAINS ALSO TRUE TYPE FONTS	31
9.2.	COPYRIGHT	32
9.3.	AUTHOR	32
10.	CREDITS	33
11.	CONTACT AND FEEDBACK	35
11.1.	GENERAL INFORMATION FOR CONTACT OR FEEDBACK	35
11.2.	E-MAIL.....	35
11.3.	POSTAL MAIL	35
11.4.	IMAGE AND SCENE FILES	35
11.5.	WHERE TO FIND OUT MORE ABOUT POVLAB AND UTILITIES.....	35
12.	UPDATES AND HOW TO REGISTER	37
12.1.	GENERAL INFORMATION.....	37
12.2.	LIMITATIONS IN THE SHAREWARE VERSION	37
12.3.	WHY REGISTER.....	37
12.4.	HOW TO REGISTER ?.....	37
12.5.	BANK TRANSFER	37
12.6.	CHECK.....	38
12.7.	CASH.....	38
12.8.	VERY LOW-COST UPDATES	38
12.9.	LOW-COST UPDATES	38
12.10.	FULL « PRO » VERSION	39
12.11.	ORDER FORM.....	39

2. GENERAL INFORMATION

POVLAB is a graphic modeller for POV-Ray. Here in Europe, many people use POVLAB because they can make scenes very quickly with its intuitive interface. While everyone can't be a Mike Miller, this software includes everything necessary to start creating and rendering scenes quickly. When I receive a particularly good scene from a user, I include it with this package. I work on implementing new functions for future updates every day. This software works VERY WELL!

This documentation was created with Word for Windows 6.0, using the Arial, Windings, and Times New Roman fonts.

Try it out, give it to your friends, and please register.

If you think that this program is good, tell me. Send me textures and scenes you've created with it. I work on it everyday. You can find it on Compuserve (go graphdev), and on the Internet (povray.org). Share your ideas for features you'd like to see in future versions with me, and we can discuss them.

I'm looking for correspondents (sysops, BBS for raytracing, Internet, Compuserve) from all over the world to help distribute Povlab. If you are interested, please let me know who you are!

Have fun working with POVLAB, and remember, suggestions, criticisms, and bug reports are always welcome.

3. RUNNING POVLAB

3.1. Before Running POVLAB

There are no ready-made thumbnail textures included with POVLAB because of the added size these would take up on the distribution disks. To create the textures, use the utility MAKETHEM.EXE (included with POVLAB) after you've read the .TXT file in the Texture directory.

3.2. Directories

3.2.1. Image

Sample image files.

3.2.2. Hfield

Includes several TGA and GIF files with which you can create height-field objects.

3.2.3. Maps

Store GIF, IFF, and TGA files here to use as maps files (bump, map, opacity).

3.2.4. Pattern

Patterns for a nice, cool interface.

3.2.5. Plugins

Examples of how to manage and create plugins. You will find some functioning .EXEs, too.

3.2.6. Povlab

Default Root directory for POVLAB.
Documentation, loader, etc.

3.2.7. Raw

.RAW objects.

3.2.8. Scene

.SCN files (scenes) for Povlab.

3.2.9. System

System files, overlays, etc.

3.2.10. Texture

Include files, texture thumbnail tiles, and libraries.

3.2.11. Truetype

Truetype font demo files created by Guy Roger.

3.3. Hardware Requirements

Requires at least a 386 with 4 Mb of memory, a VGA card, and a mouse. For adequate performance, you will need to have at least a 486 with 8 Mb of memory. Of course, an FPU (floating-point unit) is required.
30 Mb free space on your hard disk is also recommended.

SAMPLE boot files. (These files are examples only. If you are not CERTAIN of the meaning of each line in each file, get someone capable to help with modifying these files !):

3.3.1. Config.sys

```
DEVICE=C:\DOS\HIMEM.SYS /TESTMEM:OFF
BUFFERS=5,0
FILES=30
dos=UMB
LASTDRIVE=J
FCBS=16,0
dos=HIGH
DEVICE=C:\DOS\MOUSE.SYS
country=33,,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
shell=C:\DOS\COMMAND.COM C:\DOS /e:3072 /p
```

3.3.2. Autoexec.bat

```
@ECHO OFF
PATH=C:\GDS;C:\UTIL;C:\DOS;C:\BAT
VERIFY On
PROMPT $P$G
set BLASTER=A220 D1 I5 T1
set TEMP=C:\TEMP
set TMP=C:\TEMP
LH C:\DOS\SMARTDRV 2048 2048 /L (depending on your memory)
LH C:\DOS\DOSKEY
LH C:\DOS\KEYB FR,,C:\DOS\KEYBOARD.SYS
LH C:\DOS\MSCDEX.EXE /D:SONY_000 /L:E /M:10
```

3.3.3. Environment for dos

Adding this line to your config.sys file will improve POVLAB's performance
shell=C:\DOS\COMMAND.COM C:\DOS /e:3072 /p

If you want to run POVLAB from a DOS prompt, add its path to the environment variable PATH in your AUTOEXEC.BAT file.

Add a small (512kb) SMARTDRV cache if you have less than 8 Mb RAM.

Set the DOS option VERIFY to ON. This is not just for POVLAB—it will be better for your disks in general (just type VERIFY on the command line to see if it is ON or OFF).

Please, add C:\POVLAB to your PATH variable, in AUTOEXEC.BAT

3.4. Using POVLAB and Windows 95

With the following exceptions, there are no known problems running POVLAB under Windows 95. First, you can run it from a shell from the DOS command line, but only in a full-screen mode (because of the graphic interface). Second, on some graphic cards, Windows 95 will close the POVLAB session when you are swapping from one application to another, so you should be careful and test it before working with it a lot.

3.5. Running POVLAB

Type POVLAB on the command line to run POVLAB and create a new scene called NONAME.SCEN.

or

type POVLAB [path\]scene_name[.scn] to run POVLAB and load a previously created scene.

If the specified scene does not exist, POVLAB will create a new, blank scene with that name.

3.6. POV-Ray and POVLAB

POVLAB works with either POV-Ray 2.2 or FTP 2.2. If you want to include .INC files, such as COLORS.INC and TEXTURES.INC, add the environment variable POVRAYOPT to your AUTOEXEC.BAT, or add the full path in the .INC library file.

Example :

If the files you want to include are in the path C:\POVRAY\INCLUDE, add the following line to your AUTOEXEC.BAT :

```
set POVRAYOPT=-lc:\povray\include
```

There are many compiled versions of POV-Ray available. The WATCOM version, called IBMWAT386, is preferred because it uses the same DOS extenders for both POV-Ray and POVLAB. There are sometimes conflicts in accessing memory when a different VMM has been used.

3.7. WINPOV and POVLAB

The Windows version of POV-Ray by Chris Cason works well with POVLAB. POVLAB will automatically look for PVENGINE.EXE in the specified directory and write a POVLAB.INI file for it (POVLAB.DEF file for POV-Ray dos version). In this version, there's a pause when re-entering POVLAB after a rendering because some graphic cards have a problem automatically restoring POVLAB's graphic interface.

3.8. Memory with POVLAB

POVLAB is compiled with the Watcom 10.0a C/C++ 32-bit compiler, with options to make it run better on a Pentium, but it works normally on a 386. It uses a DOS-extender (VMM, or Virtual Memory Manager) called DOS4GW.EXE, and can manage virtual memory (up to 32 Mb).

When you run POVLAB, it automatically configures VMM with 16 Mb of virtual memory. There are two ways to change this setting :

- Type POVLAB [scene[.scn]] /novm

This deactivates the VMM, and just uses the physical memory you have (RAM).

- edit the file MODELLER.MEM, and modify the parameters to match your requirements. Please refer to the DOS4GW.DOC file.

Please refer to the OBJECTS section, where I explain how memory and object allocations work.

3.9. Trouble-shooting Common Problems

If POVLAB doesn't run properly, please check your computer for the following:

- POVLAB works best with 8 Mb of memory or more, so make sure you have enough RAM (sorry, I am doing my best to save memory space...).

- Make another config.sys and comment out SHARE.EXE and EMM386.EXE. Sometimes there are conflicts with those TSRs.

- Be sure not to run Windows 3.x or 95, or for Windows 95, boot for a full DOS session (POVLAB can run under Windows 3.x, but I can't support any problems you encounter).

- POVLAB needs some additional disk space for the disk extender. You need approximately 5 Mb more than the size specified in MODELLER.MEM in the section « virtualsize ».

- If you have less than 8 Mb of physical memory, don't add values that are too high to maxmim and maxmem in MODELLER.MEM. The best values are 512 and 2048.

- Be sure you have your VESA tsr loaded (version 1.2), or use a universal VESA driver, such as UNIVESA or UNIVBE. There are some problems with Matrox graphics cards. If someone could help me...

- If you have an ATI MACH 64 graphic card, there are problems with 640x480x256 mode. Edit the file SYSTEM\MODELLER.CFG and try changing the following parameters :

GraphicMode=640 to GraphicMode=800 (if your card supports 800x600)

or

PaletteInterface=256 to PaletteInterface=16

This configures POVLAB in standard VGA graphics mode.

- If you have problems with the mouse, make sure the driver is not too old. Sometimes problems arise in 640x480x256 mode with the Microsoft mouse driver v8.20. Try to find version 9.x. Logitech mouse drivers work normally. Sometimes, putting your mouse driver as the last driver in your CONFIG.SYS or AUTOEXEC.BAT will fix the problem.

- Inhibit virtual memory :

Run POVLAB with the parameter /NOVM as the last parameter on the command line, but be sure that you have enough memory— 8 Mb MINIMUM.

3.9.1. Using a log file

If you have other problems running POVLAB, send me the log file created by POVLAB. To create a log file, change DebugMode= to equal 1 in the file MODELLER.CFG. Restart POVLAB and play with your scene until you encounter the problem, then quit. The POVLAB.LOG file will be created in the POVLAB root directory.

To get more information about your system, please use a diagnostic program that can output a text file describing your system configuration, like Microsoft Diagnostic (MDS.EXE). Generate a set of system report files and compress them with Pkzip, as follows:

```
pkzip -exa plab_log.zip povlab.log report.txt
```

Then send them to me via e-mail or postal mail.

With these file I will have more information about your computer, and can more easily solve your problems.

4. INTERFACE

POVLAB uses the mouse's left button to validate, and the right button to escape. You can also use ENTER to validate and ESC to escape. To access menus, move the mouse to the top of the screen, and then click on the menu you want.

4.1. Axes

In POVLAB, the positive Y axis points up, the positive X axis points to the right, and the positive Z axis points INTO the screen (the same as POVRay).

4.2. Bottom Toolbar Buttons

4.2.1. General information

These buttons allow you to center, zoom, and move in the scene without altering the objects.

This is great for working in the 3D universe, and for making better scene compositions.

4.2.2. Zooming out button (-)

Zooms out the viewport. Use ALT to zoom out all viewports.

4.2.3. Zooming in button (+)

Opposite of the zoom-out function.

4.2.4. Recentering button

Recenters the scene based on all objects (except for ignored objects) in the current viewport— use ALT for all viewports. Cameras and light sources are considered objects in this case. If you don't want to use them in this function, hide them.

4.2.5. Zooming in a selected area

Zoom in on an area in the selected viewport that you have selected with the mouse..

4.2.6. Translating scene in viewport

Translate the scene to another area in the selected viewport.

4.2.7. Full screen button (TOGGLE)

With this function, you can obtain a full-screen display of the current viewport. It's very useful for doing accurate work. When you are finished with full-screen mode, click on this button again to display all four viewports.

4.2.8. Refresh the viewport (blue arrow)

Refresh the viewport— holding ALT displays all the viewports.

4.2.9. Saving session button (floppy)

Save the scene.

4.2.10. Fast rendering button (eye)

You can use this feature to run a render session quickly. If you want specific parameters for rendering, prepare them in the « parameters » dialog box, and then render the scene with this button as many times as you want.

4.2.11. Translate object

Allows you to enter translation parameters for an object directly from the keyboard. There are two options. The first (set) places the object at this new point. The second (add) adds these values to the position of the currently selected object(s).

4.2.12. Rotate object

Works the same as for translate, but rotates the object instead of translating it.

4.2.13. Scale object

Works the same as for translate, but scales the object instead of translating it.

4.2.14. Modifying object button (?)

Click on this button and then on an object—a dialog box will be displayed allowing you to enter coordinates for the selected object. You can modify other parameters, like the object's name, or for some objects (e.g., blobs), specific parameters.

4.3. Bottom Parameter Bar

4.3.1. General information

This bar is located at the bottom of the screen, and shows you states of the following variables, as shown below :

00000 00000 00 00 00 CL00 C00 0000Mb 640x480 FILE.SCN 00 [TEXTURE]

In order :

Number of objects in the scene.
Number of selected objects.
Number of omnilights in the scene.
Number of spotlights in the scene.
Number of arealights in the scene.
Number of cylindrical lights in the scene.
Number of cameras in the scene.
Free memory (physical + virtual).
Screen resolution in pixels.
Current scene name.
Number of raw objects in the scene.
Current texture name.

4.3.2. Objects in POVLAB

POVLAB supports the following object primitives, as described bellow :

Sphere

Cube (box)

Cylinder	Pyramid
Tube *	Prism
Ring *	X-Plane
Torus *	Y-Plane
Cone	Z-Plane
Truncated cone *	Disk *
Raw objects	Blob (spheres and cylinders)
Hemisphere (1/2 sphere)	Heightfield
Quarter of torus (1/4 torus) *	Extrusion
Quarter of tube (1/4 torus) *	B-Spline paths *

All objects are scaled to a 2-unit cube when they are created.

* These objects are constructed in the viewport, and are not scaled to a 2-unit cube.

5. TOP MENU

Where necessary, I explain some of the important details about a given function. I have intentionally omitted explanations for some functions where it is obvious how to use them.

If you find that something needs to be better explained, please let me know so that I can explain it better in future versions of the documentation.

5.1. General

5.1.1. Environment

Specify :

Path to POV-Ray.

Path for the output .POV files created by POVLAB.

Path for the output pictures generated by a renderer.

Path to POVLAB.TEX. You need to modify this path if you have not installed POVLAB in the root of the C: drive.

Path for your own graphics viewer (POVLAB leaves 2 Kb in memory when in use).

Check box to use (or not) WINPOV (only for Windows 95 users, not tested on other operating systems).

5.1.2. Memory

See parameters in the file DOS4GW.DOC, or in the MEMORY section.

5.1.3. Microsoft mouse

If your mouse driver is too old, you may experience problems. This feature deactivates some recent functions that your driver may not support.

5.1.4. Configure mouse

Add a function to the right mouse button.

5.1.5. Run image viewer

Run your preferred image viewer.

5.1.6. Run scene viewer

Run a text viewer to see the POV script generated by POVLAB. The best utility to use is

README.COM from Borland, a little utility that can read text files of any size.

5.1.7. MS-DOS

Shell to DOS.

5.1.8. Quit

Exit to DOS.

5.2. File

5.2.1. New

Delete all objects, cameras, and lights, but don't change the name of the scene.

5.2.2. Rename

Rename the scene if you want to save it under another filename.

5.2.3. Load

Select a new scene and load it into memory.

5.2.4. Save

Save the current scene to disk.

5.2.5. Save when exit

Save the current scene when exiting POVLAB.

5.2.6. -> .POV

Generate the script for POV-Ray without rendering the scene.

5.2.7. Set output tab

Select the number of spaces you want when indenting lines of code in .POV scripts created by POVLAB.

5.3. Display

5.3.1. Redraw viewport

Redraw the current viewport.

5.3.2. Redraw all viewports

Redraw all viewports.

5.3.3. Display axes

Display/hide the axes in the viewports. (TOGGLE)

5.3.4. Display grid

Display/hide the grid in the viewports. (TOGGLE)
You can't snap to the grid when it is hidden.

5.3.5. Normal display

Display objects normally (complex wireframe).

5.3.6. Fast display

Display objects quickly, with just one or two segments per face.

5.3.7. Cubic display

Display only the bounding boxes of objects.

5.3.8. Mesh precision

Can display more or fewer faces for tori and height-fields. Important: the greater the number of faces, the more time and memory is required to generate the display.

5.3.9. Image control

You can choose between three types of display dithering when reviewing the last rendered image; none (with beautiful Match bands !), ordered (fast, and good for previewing), and random noise (if you have a fast processor). The better the quality, the more time it takes to render....

5.4. Interface

5.4.1. General information

Parameters for the graphic environment. Nothing to explain; this is very simple to use.

5.5. Tools

5.5.1. General information

Power tools to generate objects. The extruder generates a triangle as a polygonal shape (like raw objects).

5.5.2. Copy by translation

5.5.3. Copy by rotation

5.5.4. Extruder

5.5.5. Object align

5.5.6. Make 3D fonts

Some fonts don't work with the parser, so please be careful of strange Truetype fonts. It is best to use the ones distributed with Windows (from Microsoft), because they have good font metrics.

POVLAB is only able to accept words or individual letters, spaces are illegal characters. To construct a sentence or a phrase you will need to enter each word separately and place them separately.

5.5.7. Load Plugins

This new feature is very powerful, especially for creating complex shapes.

If you don't want to create a Plugin, you don't need to read the rest of this section. To use Plugins, click on the .EXE for the Plugin you want, fill out the parameter box, and wait for the process to be completed.

Let's see how it works.

While POVLAB is running, you can shell to DOS, run another program, and then come back to the modeller. Plugins are used a little differently. You can call a Plugin (an external .EXE file) from the POVLAB interface, ask it to send some graphic data to POVLAB's interface, and the Plugin creates a dialog box for all the parameters you need.

POVLAB displays the dialog box and lets you enter what you want, just as if you were in POVLAB (inline help, buttons, radios, cases,). Actually, you are still in POVLAB, but the dialog box has been created by the external Plugin. Ok so far?...

Finally, click on the « ok » button and POVLAB will spawn the external Plugin. Now the Plugin is controlling the machine, while POVLAB waits for it to return the result. After the external process has finished, POVLAB checks the directory to see if a special scene has been created. If it has, it loads this file, just like a merge. (CAUTION: this object will be inserted into the existing scene!)

Now, let's look at it again in more detail. First, you need to create a program with whichever compiler you want. For example, Borland C/C++ 3.1 or Watcom 16 C/C++. You can work in another language, such as Pascal, Fortran or BASIC, but you need to respect the following conditions:

This program will not switch the processor to real mode (like PowerBASIC 2.10f, which I use). Try not to call expanded/extended memory. It must be compatible with the next procedure call. As an example, I've included the source code for a small program written in Borland C that generates a 3D grid. Check it before running your own !

The calling procedure, step by step :

First, POVLAB runs the program with the argument « **/ASK** ». This asks the external process to produce a file containing a set of commands describing parameters needed for the Plugin to work. POVLAB will read this and create a dialogue box for the user to fill in within POVLAB.

TITLE : text_with_underscore_for_spaces

The main title for the program, which is displayed in the dialog box.
There's one title, so it's not available in the command list.

COPYRIGHT : text_with_underscore_for_spaces

Copyright for the process. Show it in the dialog box too.
There's one copyright, so it's not available in the command list.

WINDOW : X Y

Width and height for the window. The window is displayed at the center of the four viewports. With X equal to 200, your window will extend from centerX-200 to centerX+200, making it 400 pixels wide. It works the same way for Y.
There's one window, so it's not available in the command list.

Attention : remember to design it for a **640x480 interface**.

TEXTZONE : X Y Label_text N Help_text

Display a text zone where you can enter some parameters.

X is for the left-right position, Y for the top-bottom one. They refer to the top-left corner of the dialog box. Label_text identifies what you need to enter. N for the initial parameter and Help_text for the inline help.

With this command you just enter numbers, such as floats or integers.

RADIO : X Y Label_text Bool Help_text Group

Tells POVLAB to begin a radio case sequence. The first three parameters are all the same as **TEXTZONE**. Bool tells if the radio is checked or not. Help_text is like **TEXTZONE** and Group tells us which group of radios we are in. When POVLAB recognizes the keyword « RADIO », it first sees if there's already a group created with the number Group. If not, it creates it; if yes, it appends the new entry to the sequence. You can create eight groups of any entries you want but there won't be more entries than the maximum number of commands allowed. Bool initializes the entry to 1 (checked) or 0. In this type of command, you can only have one entry checked, because there is only one choice per group at one time. Checking two entries doesn't do anything. In this case, use the CASE keyword.

CASE : X Y Label_text Bool Help_text

The same as for the command parameters. With this command, you can either check the case or not; one case for one choice.

MESSAGE : Message_text

Send a message to the bottom line parameter of the interface.

You can send just one message per data file. If you send more than one, the first message will be erased. It's not available in the command list.

END :

Show it's the end of the script. It's not available in the command list.

Okay, that's done. No more commands for now !

All the spaces in the text for labels and help must be replaced by an underscore. Very important! The file you will generate needs to have the .PLG extension, and needs to have the same name as the executable.

Now, POVLAB will send all the commands that you have already described. They will be sent in the order they arrived in the data file. The following example is from GRID3D.C, with entries modified a little bit for clarity:

```
fprintf(Fichier,"TITLE: GRID_3D_1.0\n");
fprintf(Fichier,"COPYRIGHT:                                Copyright_Denis_Olivier_1995_-
_All_rights_reserved.\n");
fprintf(Fichier,"WINDOW: 410 180\n");
fprintf(Fichier,"TEXTZONE:                                83           60           Width_of_Array           5
Width_of_the_object's_array\n");
fprintf(Fichier,"TEXTZONE:                                83           80           Height_of_Array           3
Height_of_the_object's_array\n");
fprintf(Fichier,"TEXTZONE:                                83          100           Depth_of_Array           7
Depth_of_the_object's_array\n");
fprintf(Fichier,"TEXTZONE: 240   60 Nb_Width 2 Nb_of_objects_in_Width\n");
fprintf(Fichier,"TEXTZONE: 240   80 Nb_Height 3 Nb_of_objects_in_Height\n");
fprintf(Fichier,"TEXTZONE: 240  100 Nb_Depth 4 Nb_of_objects_in_Depth\n");
```

```

fprintf(Fichier,"TEXTZONE:      83  130  Object_size  0.2  Scale_for_object_(X-Y-
Z)\n");
fprintf(Fichier,"RADIO: 345   60  Cube 1 Generate_cubes 1\n");
fprintf(Fichier,"RADIO: 345   80  Sphere 0 Generate_spheres 1\n");
fprintf(Fichier,"RADIO: 345  100  Cylinder 0 Generate_cylinders 1\n");
fprintf(Fichier,"CASE: 240 125 Pause 1 Pause_after_computed\n");
fprintf(Fichier,"MESSAGE: The_array_begin_in_<0,0,0>\n");
fprintf(Fichier,"END:\n");

```

In this example, the parameters will be returned by POVLAB in order, unless the initial entries are changed:

5 3 7 2 3 4 0.2 1 1

This corresponds to 7 text zones, 1 return of 3 radio buttons and 1 case. The float is formatted to send 5 decimal values by POVLAB.

Now, you just need to read the command line parameters and compute the shape.

The last thing to do is to generate a scene file. Like the .PLG file, it needs to have the .INC extension and the same name as the executable. You need to know the description for the .SCN file. This is described in detail in another chapter. After you have generated your include file and quit the process, POVLAB will load the .INC file, and remove all the temp files you have created (.PLG and .INC).

To make it easier, I've added a small library to the Plugins directory that sends the output of your calculations directly to a file, without knowing what it does. It all files with the PLUGINS name (a .H and a .C file). Check all the samples done with this library and the other ones made by several users.

No more for now !

5.5.8. Create a B-Spline paths

This interesting feature lets you create smoothed objects along a path. For example, you can use the draw function to draw a worm-like shape. Each key vertex contains a description of a sphere, with a radius. Smooth connections are made between these points, connected by tangent cones. The more the curve is subdivided, the smoother it will be..

5.5.9. Edit B-Spline

You can edit some of the B-spline parameters, such as the polynomial degree and the number of subdivisions between each control point. Play with the degree value to change the tension and vertices fitting.

5.5.10. Move vertex on B-Spline

Move a vertex in 3D, to create some incredible shapes.

5.5.11. Delete vertex on B-Spline

Delete a vertex in 3D.

5.5.12. Add vertex on B-Spline

Add a vertex between two others.

5.5.13. Edit vertex on B-Spline

Edit the diameter of the sphere in one key vertex.

5.5.14. Smooth vertices

Smooth vertices between the two end vertices. In fact, the radius of the spheres are used to approximate each other between the first and the last point.

5.6. Objects

5.6.1. General information

Use this menu to create objects. It is very simple to use. The last object created will be shown after the object's name. You can have up to 20,000 objects simultaneously in POVLAB. I think it's a lot, but you can still get an « out of memory » error, if there's no memory left. In this case, try adding more virtual memory.

You should also know that each object isn't initialized when POVLAB first runs. So, for 20,000 objects, POVLAB needs 20,000x4 bytes (40 Kb) to prepare the array. But each object you allocate afterward will take more memory (when you are running POVLAB, a line « Allocate *n* bytes per object » shows you how much memory each object requires). Here, it really allocates *n* bytes so, if you get a memory error, find out how much memory all the objects you have in your scene need by calculating *n* bytes multiplied by the number of objects.

By default, POVLAB pre-allocates 16 Mb of virtual memory. With 20,000 objects allocated, you will use up 9 Mb of this memory. Keep in mind that you need to allocate space for triangles (raw objects, heightfields) too. If you change the amount of virtual memory, be careful !

5.6.2. Raw objects

The RAW format supported by POVLAB is :

```
OBJECT_NAME  
X1 Y1 Z1 X2 Y2 Z2 X3 Y3 Z3  
...
```

With X_n, Y_n, Z_n for each vertex facet (triangle). Refer to the examples in the RAW directory.

5.6.3. Height-field

First, load a GIF or Targa (TGA) file. You can adjust the mesh tension if you want, but do this before you load an image file. It will take too much time to read the image file again and recompute the heightfield shape before it is displayed.

You will find the water_level keyword in the « ? » button, and then « parameters ».

Be careful! POVLAB doesn't support the GIF89a file format. By default, Fractint creates its GIF files in the GIF89a format. Please see Fractint's documentation about how to change this.

6. TOOL BAR BUTTON ON LEFT/RIGHT

6.1. Camera Button

6.1.1. General information

There's nothing very important to say here, except that you can manage up to 10 cameras per scene with the *registered version* of POVLAB. The camera you select with the select tools will become the current one.

For some features, such as focal blur one, please refer to the POV-Ray documentation for a more complete explanation.

6.2. Light Button

6.2.1. General information

There are three types of lights in POVLAB; spotlight, omnilight, and arealight.

Spotlight and omnilight are easy to understand, but further explanation of how an arealight works may be helpful to you.

The implementation may seem strange, but it works well. The area in arealight (in POVLAB, not POV-Ray) corresponds to a *radius*. When generating a scene for POV-Ray, the arealight will be $R=\text{radius}$: $\langle -R, -R, -R \rangle \langle R, R, R \rangle$. The parameter 'soft' equals the number of lights you want between $-R$ and $+R$.

For advanced features, such as atmosphere, please refer to the POV-Ray documentation for a better explanation of their use.

6.3. Material Button

6.3.1. General information

Managing textures is very simple. You need at least two files; one for the texture's name (i.e., POVLAB.TEX), while the other is a simple POV-Ray .INC file (i.e., POVLAB.INC) where you declare textures as you usually do with POV-Ray.

You can manage up to 300 textures per library. The current texture is displayed on the bottom-right of the screen.

6.3.2. Texture files

When you download POVLAB for the first time, you must have the following two files: POVLAB.TEX and POVLAB.INC. The first line of POVLAB.TEX contains the path to POVLAB.INC (e.g., C:\POVLAB\TEXTURE\POVLAB.INC). If you haven't installed POVLAB on drive C:, you must change this line to match your installation drive. The rest of the file contains entries for each texture, each with two arguments. The first is the name

of the texture, as described in the file POVLAB.INC. The second argument is the path to access the tiles so that the texture can be shown in the modeller.

Example :

```
PATH_TO_ACCESS_INCLUDE_FILE
texture1      path_to_texture_1      // comments...
texture2      path_to_texture_2      // comments...
texture3      path_to_texture_3      // comments...
... ..
```

6.3.3. Graphic tiles

The files for graphic tiles are in GIF87a format, so you can manage your own library very easily in POVLAB. Just render all the textures you want in 100x100x256 colors, and then convert them with an image processor, like Image Alchemy (preferred), Piclab, PhotoStyler, etc. You will need to use the palette file PALETTE.PAL to convert between TGA and GIF formats, or all colors will be off when you see the textures in the modeller.

If you are using Image Alchemy:

```
alchemy -fPALETTE.PAL -g TILE.TGA [TILE.GIF]
```

If you are using Piclab, enter the following parameters on the command line :

```
tload TILE.TGA
pread PALETTE.PAL
set dither on
makepal
map
tsave TILE.GIF
```

You need to delete the first two lines in PALETTE.PAL to use this file with Piclab.

Then, just add the textures' names and the path to access them in POVLAB.TEX, and the texture description in POVLAB.INC.

6.3.4. Texture button

Shows the textures in POVLAB.TEX. Just click on the texture you want and assign it to your object or selection. You can't see graphic tiles in 16 color mode.

6.3.5. Acquire button

Acquire a texture from an object, and make it current.

6.3.6. Assign button

Assign the current texture to an object or selection (more than one object).

6.3.7. Rotate button

Rotate the texture on the object(s) from -180° to 180°.

6.3.8. Finish button

Allow the user to specify finish specifications (such as phong, refraction, reflection, etc.). This can affect a single object, or the current selection of objects.

6.3.9. Mapping

Choose a mapping in .GIF, .TGA or .PNG format.

The mapping will be placed in the X/Y dimension first, the ratio will be the same as for the picture.

6.3.10. Setup

This is the setup for the mapping texture.

Note that the mapping will be represented with a cube when you modify it (with scale, rotation or translate), and the mapped face will have a little line in the upper left corner to indicate the origin of the picture (upper left too).

6.4. Selection Button

6.4.1. General information

Selection groups objects so that you can perform operations on them.

If 'Select' is checked, then any changes made to one of the objects selected is passed onto all other objects.

If you have a group of objects selected without the 'Select' case being checked, the operation only affect a single object.

6.4.2. Inverse

Reverses the 'Select' option. (TOGGLE)

6.4.3. Area

Select all objects that are completely within the bounding area defined by the mouse.

6.4.4. Color

Change the Selection's color, whether single or multiple objects are selected.

6.4.5. By names

Select objects by their name. You can select them with the mouse, with the spacebar, or the « all » button . A mark appears next to the selected name. You can click again to deselect it. In the text area « name », you can specify directly the names matching the object, with or without wildcards « ? » and « * ». The syntax is like a DOS command.

Example :

If you want to select all the objects beginning with « Sphere », then enter the argument « Sphere* »

6.4.6. Parameters

You can modify some parameters for all the selected objects. This is especially useful in cases where it would take a long time to change the parameters individually for each object.

6.5. Render Button

6.5.1. General information

For more detailed information about specific parameters, please refer to the documentation given with POV-Ray.

Just before rendering, POVLAB asks you whether you wish to save the scene. If the menu item « Save when exit » is checked, then this is automatic.

There are four quick modes for rendering an image. Preview, quick, optimum, and region. They generate, respectively, a 128x96, 320x240, 640x480, or a user-defined size (in pixels), as a non-anti-aliased TGA image file.

6.5.2. Final

This is where you specify parameters for rendering an image. All of these parameters are described in the official POV-Ray documentation.

6.5.3. Region

Allows you to render a selected region of the camera's viewport.

6.5.4. Background

Defines the background color for the raytraced picture. You can specify one of up to 16M colors (in 24-bit mode).

6.5.5. Palette

Allows you to change the palette option for POV-Ray (option +d?x, where x is the palette option).

6.5.6. Video

Forces POV-Ray to render in a specific video card or graphic mode.

6.5.7. View last

View the last rendered image. If the size of the image is larger than the display setting, then it will be scaled. You can view it with one of three dithering types. See « dithering type » for more information.

6.5.8. Setup

Specify special keywords for the global settings in a POV-Ray render.

6.6. Modify Button

6.6.1. Rotate

Rotate the object or selection between -180 and +180 degrees.

6.6.2. 2D scale

Scale the object or selection along one or two axes (TAB to change direction, ENTER to change axis).

6.6.3. Move

Translate the object or selection from its current position to another (TAB to change direction).

6.6.4. Copy

Works like the move function, but duplicates the object or selection (left mouse button).

6.6.5. Delete

Delete an object or selection.

6.6.6. Center XYZ

Place the object at $X=0$, $Y=0$, and $Z=0$.

6.6.7. Reinit.

Restore the object's default parameters, except the translate vector for object and texture.

6.6.8. Manual

Allows you to enter parameters for the three vectors, scale, translate, and rotate, as well as for the object and textures.

6.6.9. Color

Change the object's color. This is very useful when you want to group several objects by color to differentiate them from other objects.

6.6.10. Smooth

Works with raw objects only. Allows you to alternate the appearance of facets between smoothed and flat triangles (i.e., a diamond cannot be smoothed), or just for a special effect.

6.7. Display Button

6.7.1. General information

This allows you to display or hide some objects or selections. This provides a good way to reduce the number of objects visible in the viewports. You can render just the objects that are shown. Hidden objects cannot be manipulated, selected, or deleted.

6.7.2. Object

Hide an object. The object remains in the scene, but you can't see it. Remember, hidden objects won't be generated by POVLAB in the output file, or rendered through the display buttons, so don't forget to unhide them before running a final render.

6.7.3. None

Hide all objects in the current scene.

6.7.4. All

Show all objects hidden by the **None** command.

6.7.5. Inverse

Hide all objects you can see, and show currently hidden objects. (TOGGLE)

6.7.6. Normal, quick and cubic

Show objects with different appearance settings. Normal shows all triangles, quick only displays one or two segments per face, cubic is just a bounding box around the object (object is not visible within the box). In this way, you can display complex objects more quickly (e.g., raw objects).

6.7.7. Freeze

Freeze an object. You can see it, but cannot perform operations on it. Great for making a background for placing other objects. To use the feature, select Freeze and then click on an object. To unfreeze, repeat.

6.7.8. Ignore

You can ignore an object when re-centering viewports. When you create very large objects, like sea or clouds, it's very useful to be able to ignore them, since the focus of the scene is usually very small in comparison.

6.8. CSG Button

6.8.1. General information

POVLAB handles four Constructive Solid Geometry (CSG) boolean operations; union, intersection, difference, and merge.

6.8.2. How does it work ?

It is very simple. You click on the boolean operation you want, then on the root object on which all further objects will operate, then on the object that will perform the CSG operation (operator) on the root object of the CSG.

Example: we have the root object (OBJECTS1), the type of operation you need, and the operator object (OBJECTS2).

All operators (OBJECTS2) appear as dashed lines.

6.8.3. Very important !

Root objects (OBJECTS1) can have as many CSG operators (OBJECTS2) as you want associated with them, but you can't perform CSG operations on an operator (OBJECTS2).

6.8.4. Difference, union, intersection, merge buttons

Performs boolean operations on objects.

6.8.5. Remove CSG

Delete a boolean operation (does not delete the object).

6.8.6. Show, hide

Show or hide a boolean operator in the scene (works with selection too).

7. KEYBOARD ASSIGNMENTS

A : Redraw all viewports
C : Re-center current viewport
R : Redraw the current viewport
O : Show object's dialog box
T : Show texture dialog box
1/2/3 : Change view front/left/top in full screen

+ : Zoom into current viewport
- : Zoom out current viewport
Space : Use selection
Up/down : Move scene to up/down
Left/right : Move scene to left/right

Alt-A : Qelect all objects
Alt-B : Box display
Alt-C : Hide/show camera(s)
Alt-F : Fast display
Alt-H : Show this help screen
Alt-L : Hide/show light(s)
Alt-N : Normal display
Alt-R : View last image
Alt-S : Save the scene on disk
Alt-U : Deselect all objects
Alt-X : Quit POVLAB

Alt-F1 : Align objects
Alt-F4 : Quit POVLAB

F1 : Shell to MS-DOS
F2 : Run picture viewer

8. MANAGING TEXTURES

8.1. Generalities

This file describes how to manage textures with the new utilities. For more information, please read the documentation POVLAB.DOC.

The textures are described by 2 files, POVLAB.TEX and POVLAB.INC. The first contains library assignments for both the texture's name and the thumbnail's name. POVLAB.INC is a POV-Ray filelike TEXTURES.INC.

I've written 2 utilities to help manage the textures library. MAKETHEM.EXE is the program that actually manages the library. TGAVGIF.EXE is a program to convert TGA files to 8 bits GIF files, with good color reduction and dithering. You can use it for other images besides texture thumbnails. It needs DOS4GW.EXE in the path to run.

Here is how to render each thumbnails for unallocated textures names. Go to the POVLAB\TEXTURE path and run MAKETHEM.EXE

You will see :

MAKE THEM ALL release 1.0, (C) Copyright ChromaGraphics, 1994-1996.
Texture library maker and manager for POVLAB.
All rights reserved, (R) Denis Olivier - Feb 29 1996.

Syntax : makethem [option(s)]

-Ppath : specify path for POV-Ray (default current)
-lpath : specify file library.TEX (default POVLAB.TEX)
-D : display while rendering (+d0)
-F : enable Faster Than POV-Ray speedup options (2.2 only)
-A : use antialiasing rendering option (+a0.3)
-? : syntax and help screen

Example : makethem -pc:\povray -ic:\povlab\povlab.tex

-P : to tell the program where POV-Ray is(just the path)
-l : choose one texture library for POVLAB.
-D : display the rendering image while computing
-F : only for POV-Ray 2.2 users
-A : use antialiasing for better display

8.2. Command line

If POV-Ray is in the path and you want to add new thumbnails to your POVLAB.TEX, just enter "makethem" on the dos prompt.

8.3. When POV-Ray isn't in the path

type "makethem -pc:\povray".
Here POV-Ray dir is C:\POVRAY

8.4. How it works

First, MAKETHEM will read the POVLAB.TEX file (or the one you've specified) and check for unallocated texture names.
Then it prepares a .POV file, sends it to POV-Ray, and converts it into .GIF format with TGAVGIF.EXE.
If you stop the rendering before the TGA file is complete no .GIF file will be created.

8.5. Name the texture thumbnails

MAKETHEM will automatically assign new names to textures thumbnails.
The prefix is TEX, followed by 4 digits (e.g., 0000), and then .TGA or .GIF.
You can have up to 9999 files.
MAKETHEM will check when it finds an unallocated texture name whether TEX0000.TGA exists, and if it does will continue to check for TEX0001.TGA, TEX0002.TGA, etc. until it finds an unallocated name.

8.6. Assign names and thumbnails together

When MAKETHEM has finished, it generates the file MAKETHEM.LOG, which has a similar structure to POVLAB.TEX, with names and thumbnails associated.
You can edit the file POVLAB.TEX, add the match the new thumbnails file to their respective names, and then remove the .TGA files.

8.7. Options used in MAKETHEM for POV-Ray and TGAVGIF

POV-Ray : -iTEXTURE.POV -oTEX?????.TGA +ft +w100 +h100 +v +mb2 +a +x
TGAVGIF : TEX????? /f /i

Be carefull : TGAVGIF needs the DOS4GW.EXE extender to work properly.
So put the directory where the dos extender can be found in the path (DOS, for example)

9. FONT LIST

9.1. POVLAB contains also TrueType fonts

- Balcco Bold
- Candle Bold
- Candle Bold Italique
- Cargo Bold

- Dekko
- Inca
- Slead Medium
- Wire Book.

The shareware versions of BALCCO BOLD, CANDLE BOLD, CARGO MEDIUM, SLEED MEDIUM and WIRE BOOK are without accentuated chars. The complete versions can be found at the adress at the end of this text.

9.2. Copyright

BALCCO BOLD, CANDLE BOLD, CARGO MEDIUM, DEKKO, INCA, SLEED MEDIUM and WIRE BOOK are copyrighted by Guy ROGER. When you acquire the original disk version of these fonts, you can only use them on one computer at one time. You can only use them in personal, commercial or public documents, only made on your computer.

All rights are retained by the author and are protected under French laws. Fonts should be use as any copyrighted documents (as books). You can't copy, distribute, sell or give these fonts without the explicit consent of the author.

All diffusions, free of charge anyway, can be done after you have contacted the author.
French law of 11 march 1957 for books and artistic productions.

9.3. Author

Guy ROGER
Multimedia Designer
3 Jardin Montaigne
37300 JOUE LES TOURS
FRANCE
(33) 47672071

10. CREDITS

Klaus Lepschi

A fabulous guy who believe in me and produce a wonderfull illustrated tutorial (you will find it on the web) and some very interestings programs and utilities around POVLAB and 3d.

Nicolas Kelemen

DP-Tools (ASP vendor) very good programs, especially « L'Atelier d'Aton ».+++ Has everything you need to make computer-generated pictures (cd-rom).

Juan R. Migoya

Correction and update of MAKETHEM. Support in mailing list, FAQ support.

Alain lioret

Distribution, publicity.

Dan Farmer

The first American to speak with me about my proram and to discuss POV-Ray and POVLAB with me. Thanks also for his support on Compuserve.

Chris Cason

Sysop of povray.org and author of « Raytrace ! the official POV-Ray cd-rom).

Adam Shiffman

Sysop of The Graphic Alternative (TGA bbs and home page).

The POV-Team and D.K.Buck for DKB-Trace.

TOTO and their music (help for my spirit when I'm coding software).

As Steve Lukather says, « I want to take this opportunity to recognize my friends around the world who have given me incredible support and guidance » :

Angus Chrisitan (documentation).

Bart Wakkee & Ralph Gortzen (Sea dos viewer).

Cess J. Van Der Mark (pictures, debug, plugins).

David Anjo (debug).

Edward Stodghill (debug and documentation).

Eric Weber (beta tests, pictures, textures).

Guy Roger (truetype font samples).

Herve Martin (beta tests and scenes).

Jay Sprenkle (debug).

Jean Arnaud (truetype font files support).

Karim Bentroudi (beta tests and scene).

Lee Bryars (debug, pictures, textures, support on Compuserve).

Martin Conlon (debug, pictures, OS/2 support).

Mike Miller (stone and others textures).

Paul Cleveland (beta, debug).

Rene Schwietzke (debug).

Richard DuPertuis (debug).

Robert J.Hodkinson (plugins, pictures, documentation and debug)

Robert Seidel (plugins and debug).
Sebastien Guelt (beta tests and scene).
Stephane Marty (C programming and Watcom support).
Stephane Trolley (beta tests, pictures).
Sylvain Bergeon (debug).
Todd A. Prader (truetype font support - Font3d, not used at all).

Many people ask me about **ChromaGraphics**. It's just a group name like « POV-Team » or « The Stone Soup Group » (creators of Fractint).

We have developed a lot of raytracing programs, animations, and fractals since 1991, and we are generating a lot of pictures for our own pleasure. I think the quality of our work is quite good, and we have made some fantastic creations with our products. But it's very hard for a French person to communicate with everyone; first, because of the language, and second, because there aren't a lot of net connections here. We need to find out what you want in this program, and learn to speak and write more and more of your language (since it's easier for us to learn English than for you to understand the French language ! Sorry...).

11. CONTACT AND FEEDBACK

11.1. General information for contact or feedback

I'd like to see what you do with my software, just to know what features are most used, and the level of quality it can manage.

For feedback any technical questions and bugs, please report in the a sheet of paper, a fax or an email the following point.

You : lastname, address, zipcode, city, country, e-mail address, phone, fax.

Your computer : cpu clock, memory, type of bios, bus type hdd free space, video card, memory, vesa type, mouse type and version, resident softwares, dos type and version, Windows 95.

Your POVLAB version : register number, modeler version, path to POVLAB, parameters in MODELEUR.MEM, pattern file name for interface, interface resolution, buttons bar on left or right, how much colors for the interface, number of objects and light sources when problems.

After that, describe in details the problem.

11.2. e-mail

Send me e-mail from anywhere, about anything (support, bugs, new releases, pictures, scenes, comments, etc.). Please tell me before sending large files.

11.3. Postal Mail

If you want to send large files, postal mail is preferred. You should only send me the registration form.

11.4. Image and Scene Files

I would really like to see what you've done with POVLAB. I just need to see pictures, so you can keep the .SCN for yourself. I want to put a POVLAB gallery and « image of the month » in POVLAB's home page. So, send me your pictures (by e-mail if <300 kb).

11.5. Where to Find Out More about POVLAB and Utilities

The first best way is to go to the ***POVLAB's Home Page*** on the Internet, at :

<http://www.cyberstation/~dolivier/povlab.html>

Here you'll find utilities, links, bugs, pictures , a full illustrated tutorial and tips about POVLAB.

The other way is to go to ***Persistence Of Vision Home Page*** on Internet, at :

<http://www.povray.org>

Then go to the ftp site, directory *pub/povray/utilies/modellers/povlab*. Here there's the latest version, with another directory with plugins for POVLAB.
You can search with robots to or **FTP Search** for POVLAB sites.

Finally, check the **newsgroup** *comp.graphics.rendering.raytracing*, THE newsgroup related to the POV-Ray raytracer. Here you'll find the POV-Team, a lot of POV-Ray users, questions and comments, tips, links, discussions, me, and more.....

12. UPDATES AND HOW TO REGISTER

12.1. General Information

There are two ways to register POVLAB. Please read the following carefully.

12.2. Limitations in the shareware version

The limitations in the shareware version are :

- You cannot continue an aborted rendering (+c option for POV-Ray disable).
- You cannot render a image larger than 640x480 pixel wide.
- You will not have the output for the .POV source generated via POVLAB.

All the rest is totally the same as the registered version.

12.3. Why Register

All features requested by registered users will receive a higher priority than other requests.

12.4. How to Register ?

There are two ways you can register. You need to fill out the form at the end of the documentation and send that to me, either by postal mail or fax.

I will send you a registration code by postal mail, that you will need to enter in the program. A full registered version of POVLAB (3½ HD disks only) will arrive with it.

If you have any problems reading the documentation (because you don't have Word for Windows 6.0), please let me know. I'll try to send it to you in another format (RTF, TXT, WRI, WordPerfect, Works). Tell me the format you want me to convert to.

12.5. Bank Transfer

This is the best way to pay for your registration, since it involves the lowest fees, takes the shortest time, and the banks get as little money as possible.

Get your bank to transfer FF 300 (or \$60 US) to my bank account. Please make sure that no further costs are charged to us. Don't forget to fax or mail me the registration form.

Bank : LA POSTE
CENTRE DE CHEQUES POSTAUX
33900 Bordeaux
France

Phone : (33) 56 48 78 44

Account Nr : 7 083 44 H BOR

Account Owner : Mr. Denis Olivier
58, rue Robert Ballion
33700 Merignac
France

As you can see, I don't receive postal mail at the same adress, but unless there is an unusual problem, it will be transfered. Please write the check to DENIS OLIVIER.

12.6. Check

Please, fill this in with just my name (Denis Olivier, but NOT *ChromaGraphics*), and put international currency only (as \$US, FF, DM, £ and all international convertible currencies - check with your bank).

12.7. Cash

For those of you who are brave enough to entrust cash to the postal service, you can put either FF 300 or US\$60 in an envelope, together with the registration form, and send it to my address. I disclaim responsibility if there are any problems with this method.

But finally, it's the best solution, for you and me. For you, because everything arrives at the same time and there's no problem coordinating the registration form and the payment, and for me, because with the extra money I can register software in the US with companies that don't support credit cards (and maybe have enough left over to send \$20 to the TOTO fan club !).

12.8. Very low-cost Updates

You need to meet the following conditions :

- Already be registered for POVLAB.
- Have an old version that only differs from the new version by the minor version number. Example : version 2.2 and 2.1 qualifies, 2.2 and 3.0 do not.
- Fill out the order form and specify that it's a very low-cost update.
- YOU NEED TO : send me the old registration form that I sent to you originally.

12.9. Low-cost Updates

You need to meet the following conditions :

- Already be registered for POVLAB.
- You have an old version that differs from the new version by the major version number. Example : version 2.2 and 3.0 qualifies (2.2 and 2.7 would qualify for a very low cost upgrade).
- Fill out the order form and specify that it's a low-cost update.

- YOU NEED TO : send me the old registration form that I sent to you originally.

12.10. Full « PRO » Version

When you register POVLAB, you will receive a full, unlimited version, with all the scenes, textures, patterns, and raw objects. You can also have full technical support, by postal mail or internet.

12.11. Order Form

Please **PRINT CLEARLY** and send with your money to :

ChromaGraphics
DENIS OLIVIER
58, rue Robert Ballion
33700 Mérignac
FRANCE

ORDER FORM FOR POVLAB

- ☐ I would like a full PRO copy of POVLAB. I understand that I will receive a copy of the software with a registration number, that I may freely use for any personal purpose that I wish.

Firstname Lastname

Address

Country.....*Age

Tel.No Fax.No

If you are on some nets, your ID/e-mail adress

Add your name in the auto electronic mailing list (tips, new versions, beta) ?.....

- ☐ Paid by :

- ☐ Bank transfer in francs
- ☐ Cheque in your money equivalent to FF 300 or US \$60
- ☐ Cash (in Francs or US\$) enclosed

- ☐ Pay for :

- ☐ US\$ 15/FF 50 : Very low cost update
- ☐ US\$ 35/FF 150 : Low cost update
- ☐ US\$ 65/FF 300 : Full version, I wasn't registered

- ☐ Total FF..... or \$US

For update, your registration number is.....

- ☐ Please specify exactly what should appear in the registered version (if you don't specify, your lastname and firstname from above will be used) :

String (max. 50 characters, 2 words only)

- ☐ Where did you find POVLAB ?

- ☐ What version of POVLAB do you already have.....You're registered for .

- ☐ Date and sign

* Items marked are optional

I'd still like to hear from you if you have comments about this program. Please use the space below, or attach some more pages. THANK YOU. Suggestions :