

NAME

rvu - generate RADIANCE images interactively

SYNOPSIS

```
rvu [ rpict options ] [ -n nproc ] [ -o dev ] [ -b ] [ -pe exposure ] [ $EVAR ] [ @file ] octree
rvu [ options ] -defaults
rvu -devices
```

DESCRIPTION

Rvu generates RADIANCE images using *octree*. (The octree may be given as the output of a command enclosed in quotes and preceded by a '!'.) Options specify the viewing parameters as well as giving some control over the calculation. Options may be given on the command line and/or read from the environment and/or read from a file. A command argument beginning with a dollar sign ('\$') is immediately replaced by the contents of the given environment variable. A command argument beginning with an at sign('@') is immediately replaced by the contents of the given file. The options are the same as for *rpict*(1), with a few notable exceptions. The *-pd*, *-r*, *-z*, *-S*, *-P*, *-PP* and *-t* options are not supported, and *-o* specifies which output device is being used instead of the output file. The *-x*, *-y* and *-pa* options are unnecessary, since *rvu* scales the display image to the specified output device. Additionally, the *-b* option improves the display on greyscale monitors, and *-pe* may be used to set an initial exposure value.

The *-n* option may be used to specify multiple processes, to accelerate rendering.

In the second form, the default values for the options are printed with a brief explanation. In the third form, the list of supported output devices is displayed.

rvu starts rendering the image from the selected viewpoint and gradually improves the resolution of the display until interrupted by keyboard input. *rvu* then issues a prompt (usually ':') and accepts a command line from the user. *rvu* may also stop its calculation and wait for command input if the resolution of the display has reached the resolution of the graphics device. At this point, it will give the 'done:' prompt and await further instructions. If *rvu* runs out of memory due to lack of resources to store its computed image, it will give the 'out of memory:' prompt. At this prompt, the user can save the image, quit, or even restart a new image, although this is not generally recommended on virtual memory machines for efficiency reasons.

rvu is not meant to be a rendering program, and we strongly recommend that *rpict*(1) be used instead for that purpose. Since *rpict*(1) does not store its image in memory or update any display of its output, it is much faster and less wasteful of its resources than *rvu*. *rvu* is intended as a quick interactive program for deciding viewpoints and debugging scene descriptions and is not suited for producing polished images.

COMMANDS

Once the program starts, a number of commands can be used to control it. A command is given by its name, which can be abbreviated, followed by its arguments.

aim [*mag* [*x y z*]]

Zoom in by *mag* on point *x y z*. The view point is held constant; only the view direction and size are changed. If *x y z* is missing, the cursor is used to select the view center. A negative magnification factor means zoom out. The default factor is one.

^C Interrupt. Go to the command line.

exposure [*spec*]

Adjust exposure. The number *spec* is a multiplier used to compensate the average exposure. A value of 1 renormalizes the image to the computed average, which is usually done immediately after startup. If *spec* begins with a '+' or '-', the compensation is interpreted in f-stops (ie. the power of two). If *spec* begins with an '=', an absolute setting is performed. An '=' by itself permits interactive display and setting of the exposure. If *spec* begins with an '@', the exposure is adjusted to present similar visibility to what would be experienced in the real environment. If *spec* is absent, or an '@' is followed by nothing, then the cursor is used to pick a specific image location for normalization.

focus [*distance*]

Set focus distance for depth-of-field sampling. If a distance in world coordinates is absent, then the cursor is used to choose a point in the scene on which to focus. (The focus distance setting does not affect rendering in *rvu*, but can be used in *rpict* with the *-pd* option to simulate depth-of-field on views saved from *rvu*.)

frame [*xmin ymin xmax ymax*]

Set frame for refinement. If coordinates are absent, the cursor is used to pick frame boundaries. If "all" is specified, the frame is reset to the entire image.

free

Free cached object structures and associated data. This command may be useful when memory is low and a completely different view is being generated from the one previous.

last [*file*]

Restore the previous view. If a view or picture *file* is specified, the parameters are taken from the last view entry in the file.

L [*vw* [*rfile*]]

Load parameters for view *vw* from the *rad(1)* input file, *rfile*. Both *vw* and *rfile* must be given the first call, but subsequent calls will use the last *rfile* as a default, and "1" as the default view (ie. the first view appearing in *rfile*). If *rvu* was started by *rad*, then the *rfile* parameter will initially default to the *rad* input file used.

move [*mag* [*x y z*]]

Move camera *mag* times closer to point *x y z*. For a perspective projection (or fisheye view), only the view point is changed; the view direction and size remain constant. The view size must be modified in a parallel projection since it determines magnification. If *x y z* is missing, the cursor is used to select the view center. A negative magnification factor decreases the object size. The default factor is one. Care must be taken to avoid moving behind or inside other objects.

new [*nproc*]

Restart the image, using the specified number of rendering processes. Usually used after the "set" command.

origin [*forward* [*right* [*up*]]]

Move the view origin forward or backwards the distance specified by the *forward* argument, along the view direction vector. If at least two arguments are given, the second distance will be a *right* shift, or left if negative relative to the current view direction. If three arguments are given, the third distance will be a distance of *up* along the view up vector, or down if negative. The view direction and size are unchanged in any of the above cases. With no arguments, the cursor is used to select the view origin, and the direction will be determined by the (reoriented) surface normal. The view type and size will not be altered, but the up vector may be changed if the new direction coincides.

pivot *angle* [*elev* [*mag* [*x y z*]]]

Similar to the "move" command, but pivots the view about a selected point. The *angle* is measured in degrees around the view up vector using the right hand rule, so a positive value pivots the viewer to the right of the selected point. The optional *elev* is the elevation in degrees from the pivot point; positive raises the view point to look downward and negative lowers the view point to look upward.

quit

Quit the program.

^R

Redraw the image. Use when the display gets corrupted. On some displays, occassionally forcing a redraw can improve appearance, as more color information is available and the driver can make a better color table selection.

rotate *angle* [*elev* [*mag*]]]

Rotate the camera horizontally by *angle* degrees using the right-hand rule. A positive value rotates the view towards the left, and a negative value looks to the right. If an elevation is specified, the camera looks upward *elev* degrees. (Negative means look downward.)

set [*var* [*val*]]

Check/change program variable. If *var* is absent, the list of available variables is displayed. If *val* is absent, the current value of the variable is displayed and changed interactively. Otherwise, the variable *var* assumes the value *val*. Variables include: ambient value (av), ambient value weight (aw), ambient bounces (ab), ambient accuracy (aa), ambient divisions (ad), ambient radius (ar), ambient samples (as), black&white (b), back face visibility (bv), direct jitter (dj), direct sampling (ds), direct threshold (dt), direct visibility (dv), irradiance (i), limit weight (lw), limit recursion (lr), medium extinction (me), medium albedo (ma), medium eccentricity (mg), medium sampling (ms), pixel sample (ps), pixel threshold (pt), specular jitter (sj), specular threshold (st), and uncorrelated sampling (u). Once a variable has been changed, the "new" command can be used to recompute the image with the new parameters. If a program variable is not available here, it may show up under some other command or it may be impossible to change once the program is running.

trace [*xbeg ybeg zbeg xdir ydir zdir*]

Trace a ray. If the ray origin and direction are absent, the cursor is used to pick a location in the image to trace. The object intersected and its material, location and value are displayed.

view [*file* [*comments*]]

Check/change view parameters. If *file* is present, the view parameters are appended to a file, followed by *comments* if any. Alternatively, view options may be given directly on the command line instead of an output view file. Otherwise, view parameters are displayed and changed interactively.

V [*vw* [*rfile*]]

Append the current view as view *vw* in the rad file *rfile*. Compliment to *L* command. Note that the view is simply appended to the file, and previous views with the same name should be removed before using the file with *rad*.

write [*file*]

Write picture to *file*. If argument is missing, the current file name is used.

^Z

Stop the program. The screen will be redrawn when the program resumes.

ENVIRONMENT

RAYPATH	the directories to check for auxiliary files.
DISPLAY_GAMMA	the value to use for monitor gamma correction.

AUTHOR

Greg Ward

SEE ALSO

getinfo(1), lookamb(1), mkpmap(1), oconv(1), pfilt(1), rad(1), rpict(1), rtpict(1), rtrace(1)