

NAME

ranimate - compute a RADIANCE animation

SYNOPSIS

ranimate [*-s*] [*-n*] [*-e*] [*-w*] *ranfile*

DESCRIPTION

Ranimate is an executive program that reads the given *ranfile* and makes appropriate calls to *rad(1)*, *rpict(1)*, *pinterp(1)*, and/or *pfilt(1)* to render an animation. Variables in *ranfile* indicate input files, process servers (execution hosts), output directories and file names, and various other controls and options.

Normally, commands are echoed to the standard output as they are executed. The *-s* option tells *ranimate* to do its work silently. The *-n* option tells *ranimate* not to take any action (ie. not to actually execute any commands). The *-e* option tells *ranimate* to explicate all variables used for the animation, including default values not specified in the input file, and print them on the standard output.

The *-w* option turns off warnings about multiply and misassigned variables.

Normally, *ranimate* will produce one animation frame for each view given in the specified view file. If an animation has ended or been killed in an incomplete state, however, *ranimate* will attempt to pick up where the earlier process left off. If the process is still running, or was started on another machine, *ranimate* will report this information and exit.

Animation variable assignments appear one per line in *ranfile*. The name of the variable is followed by an equals sign ('=') and its value(s). The end of line may be escaped with a backslash ('\'), though it is not usually necessary since additional variable values may be given in multiple assignments. Variables that should have only one value are given in upper case. Variables that may have multiple values are given in lower case. Variables may be abbreviated by their first three letters, except for "host", which must have all four. Comments in *ranfile* start with a pound sign ('#') and proceed to the end of line.

The animation variables, their interpretations and default values are given below.

DIRECTORY

The name of the animation directory. All temporary files generated during the animation will be placed in this directory, which will be created by *ranimate* if it does not exist. A file named "STATUS" will also be created there, and will contain current information about the animation process. This variable has no default value, and its setting is required.

OCTREE The name of the octree file for a static scene walk-through animation. There is no default value for this variable, and any setting will be ignored if the *ANIMATE* variable is also set (see below).

ANIMATE

The scene generation command for a dynamic animation. This command, if given, will be executed with the frame number as the final argument, and on its standard output it must produce the complete octree for that frame. Care must be taken that this command does not create any temporary files that might collide with same-named files created by other animation commands running in parallel. Also, the command should produce no output to the standard error, unless there is a fatal condition. (I.e., switch all warnings off; see the BUGS section, below.) There is no default animation command, and either this variable or the *OCTREE* variable must be set.

VIEWFILE

This variable names a file from which *ranimate* may extract the view for each frame in the animation. This file should contain one valid view per frame, starting with frame 1 on line 1, regardless of the setting of the *START* variable. An exception is made for a view file with only a single view, which is used for every frame of a dynamic scene animation. This variable is required, and there is no default value.

START The initial frame number in this animation sequence. The minimum value is 1, and if a later starting frame is given, *ranimate* assumes that the earlier frames are included in some other *ranfile*, which has been previously executed. (See the *NEXTANIM* variable, below.) The

default value is 1.

END The final frame number in this sequence. The minimum value is equal to the *START* frame, and the default value is computed from the number of views in the given *VIEWFILE*.

EXPOSURE

This variable tells *ranimate* how to adjust the exposure for each frame. As in *pfilt*, the exposure setting may be given either as a multiplier or as a number of f-stop adjustments (eg. +2 or -1.5). Alternatively, a file name may be given, which *ranimate* will interpret as having one exposure value per line per frame, beginning with frame 1 at line 1. (See also the *VIEWFILE* variable, above.) There is no default value for this variable. If it is not given, an average level will be computed by *pfilt* for each frame.

BASENAME

The base output file name for the final frames. This string will be passed to the *-o* and *-z* options of *rpict*, along with appropriate suffixes, and thus should contain a *printf(3)* style integer field to distinguish one frame number from another. The final frames will use this name with a ".pic" suffix. The default value is the assigned *DIRECTORY* followed by *"/frame%03d"*.

host A host to use for command execution. This variable may be assigned a host name, followed by an optional number of parallel processes, followed by an optional directory (relative to the user's home directory on that machine), followed by an alternate user name. Multiple *host* assignments may appear. It is not advisable to specify more than one process on a single-CPU host, as this just tends to slow things down. The default value is "localhost", which starts a single process in the current directory of the local machine.

RIF This variable specifies a *rad* input file to use as a source of rendering options and other variable settings. If given, *ranimate* will execute *rad* and create an options file to later pass to *rpict* or *rtrace*. Besides prepending the *render* variable, *ranimate* will also extract default settings for the common variables: *OCTREE*, *RESOLUTION*, *EXPOSURE* and *pfilt*. Following the file name, overriding variable settings may be given, which will be passed to *rad* on the command line. Settings with spaces in them should be enclosed in quotes. The execution of *rad* will also update the contents of the octree, if necessary. There is no default value for this variable.

DISKSPACE

Specify the amount of disk space (in megabytes) available on the destination file system for temporary file storage. *Ranimate* will coordinate its batch operations based on this amount of storage, assuming that there is either enough additional space for all the final frames, or that the given *TRANSFER* command will move the finished frames to some other location (see below). The default value is 100 megabytes.

ARCHIVE

After each batch rendering is finished and checked for completeness, *ranimate* will execute the given command, passing the names of all the original pictures and z-buffer files generated by *rpict*. (The command is executed in the destination directory, and file names will be simple.) Normally, the archive command copies the original files to a tape device or somewhere that they can be retrieved in the event of failure in the frame interpolation stages. After the archive command has successfully completed, the original renderings are removed. There is no default value for this variable, meaning that the original unfiltered frames will simply be removed. Note that the last one or two rendered frames may not be copied, archived or removed in case there is another sequence picking up where this one left off.

TRANSFER

The command to transfer the completed animation frames. The shell changes to the destination directory and appends the names of all the finished frames to this command before it is executed. Normally, the transfer command does something such as convert the frames to another format and/or copy them to tape or some other destination device before removing them. If this variable is not given, the final frames are left where they are. (See *BASENAME*,

above.)

RSH The command to use instead of *rsh(1)* to execute commands remotely on another machine. The arguments and behavior of this program must be identical to the UNIX *rsh* command, except that the *-l* option will always be used to specify an alternate user name rather than the *user@host* convention. The *-l* option may or may not appear, but the *-n* option will always be used, and the expected starting directory will be that of the remote user, just as with *rsh*.

NEXTANIM

This variable specifies the next *ranfile* to use after this sequence is completed. This offers a convenient means to continue an animation that requires different control options in different segments. It is important in this case to correctly set the *START* and *END* variables in each *ranfile* so that the segments do not overlap frames.

OVERSAMPLE

This variable sets the multiplier of the original image size relative to the final size given by the *RESOLUTION* variable. This determines the quality of anti-aliasing in the final frames. A value of 1 means no anti-aliasing, and a value of 3 produces very good anti-aliasing. The default value is 2. (A fractional value may be used for previews, causing low resolution frames with large, blocky pixels to be produced.)

INTERPOLATE

This variable sets the number of frames to interpolate between each rendered frame in a static scene walk-through. Z-buffers for each rendered frame will be generated by *rpict*, and *pinterp* will be called to perform the actual "tweening." This results in a potentially large savings in rendering time, but should be used with caution since certain information may be lost or inaccurate, such as specular highlights and reflections, and objects may even break apart if too few renderings are used to interpolate too much motion. The default value for this variable is 0, meaning no interpolation. Interpolation is also switched off if the *ANIMATE* variable is specified.

MBLUR This variable specifies the number of additional samples to be taken at each final frame for motion blurring. An optional shutter speed may be given as a second argument, which indicates the fraction of the between-frame time during which the camera shutter is open. (The default value is 1, meaning the shutter is open the whole time.) The *pmbur(1)* command is used to generate the given number of additional views for *pinterp* to average together. The default value is 0, meaning no motion blurring. This option does not currently work with the *ANIMATE* variable, since *pinterp* only works for static environments.

RTRACE This boolean variable tells *ranimate* whether or not to employ *rtrace* during frame interpolation using the *-fr* option to *pinterp*. If set to True, then the same rendering options and static octree are passed to *rtrace* as are normally used by *rpict*. The default value is False. Note that this variable only applies to static environment walk-throughs (i.e., no *ANIMATE* command).

RESOLUTION

This variable specifies the desired final picture resolution. If only a single number is given, this value will be used for both the horizontal and vertical picture dimensions. If two numbers are given, the first is the horizontal resolution and the second is the vertical resolution. If three numbers are given, the third is taken as the pixel aspect ratio for the final picture (a real value). If the pixel aspect ratio is zero, the exact dimensions given will be those produced. Otherwise, they will be used as a frame in which the final image must fit. The default value for this variable is 640.

render This variable may be used to specify additional options to *rpict* or *rtrace*. These options will appear after the options set automatically by *rad*, and thus will override the default values.

pinterp This variable may be used to specify additional options to *pinterp*, which is used to interpolate frames for a static scene walk-through. (See the *pinterp* man page, and the *INTERPOLATE* variable.) Do not use this variable to set the *pinterp -fr* option, but use the *RTRACE* setting instead.

pfilt This variable may be used to specify additional options to *pfilt*. If this variable is given in the *ranfile*, then *pfilt* will always be used. (Normally, *pfilt* is called only if *pinterp* is not needed or automatic exposure is required.) See the *pfilt* manual page for details.

EXAMPLES

A minimal input file for *ranimate* might look like this:

```

:
sample.ran
:
# The rad input file for our static scene:
RIF= tutor.rif
# The spool directory:
DIRECTORY= anim1
# The view file containing one view per frame:
VIEWFILE= anim1.vf
# The amount of temporary disk space available:
DISKSPACE= 50# megabytes

```

Note that most of the variables are not set in this file. If we only want to see what default values *ranimate* would use without actually executing anything, we can invoke it thus:

```
ranimate -n -e sample.ran
```

This will print the variables we have given as well as default values *ranimate* has assigned for us. Also, we will see the list of commands that *ranimate* would have executed had the *-n* option not been present.

Usually, we execute *ranimate* in the background, redirecting the standard output and standard error to a file:

```
ranimate sample.ran >& sample.err &
```

If we decide that the default values *ranimate* has chosen for our variables are not all appropriate, we can add some more assignments to the file:

```

host= rays 3 ~greg/obj/tutor ray    # execute as ray on multi-host "rays"
host= thishost                      # execute one copy on this host also
INTERP= 3                           # render every fourth frame
RES= 1024                            # shoot for 1024x resolution
MBLUR= 5 .25                         # apply camera motion blur
EXP= anim1.exp                       # adjust exposure according to file
pfilt= -r .9                         # use Gaussian filtering
ARCHIVE= tar cf /dev/nrtape          # save original renderings to tape

```

Note the use of abbreviation for variable names.

FILES

\$(DIRECTORY)/STATUS animation status file \$(DIRECTORY)/* other temporary files
\$(BASENAME).pic final animation frames

AUTHOR

Greg Ward

BUGS

Due to the difficulty of controlling processes on multiple execution hosts, the *-n* option of *ranimate* is not useful in the same way as *rad* for generating a script of executable commands to render the sequence. It may give an idea of the sequence of events, but certain temporary files and so forth will not be in the correct state if the user attempts to create a separate batch script.

If multiple processors are available on a given host and the *RTRACE* variable is set to True, then the *-PP* option of *rtrace* should be employed, but it is not. There is no easy way around this problem, but it has only minor consequences in most cases. (The *-PP* option is used for *rpict*, however.)

The current implementation of the remote shell does not return the exit status of the remote process, which makes it difficult to determine for sure if there has been a serious error or not. Because of this, *ranimate* normally turns off warnings on all rendering processes, and takes any output to standard error from a remote command as a sign that a fatal error has occurred. (This also precludes the use of the *-t* option to report rendering progress.) If the error was caused by a process server going down, the server is removed from the active list and frame recovery takes place. Otherwise, *ranimate* quits at that point in the animation.

The current execution environment, in particular the RAYPATH variable, will not be passed during remote command execution, so it is necessary to set whatever variables are important in the remote startup script (e.g., ".cshrc" for the C-shell). This requirement may be circumvented by substituting the *on(1)* command for *rsh(1)* using the *RSH* control variable, or by writing a custom remote execution script.

If a different remote user name is used, *ranimate* first attempts to change to the original user's directory with a command of the form *cd uname* . This works under *csh(1)*, but may fail under other shells such as *sh(1)*.

If multiple hosts with different floating point formats are used, *pinterp* will fail because the Z-buffer files will be inconsistent. (Recall that *pinterp* is called if INTERPOLATE > 0 and/or MBLUR > 0.) Since most modern machines use IEEE floating point, this is not usually a problem, but it is something to keep in mind.

SEE ALSO

pfilt(1), *pinterp(1)*, *pmbur(1)*, *rad(1)*, *rpict(1)*, *rsh(1)*, *rtrace(1)*