

NAME

ranimove - render a RADIANCE animation with motion

SYNOPSIS

ranimove [**-s**] [**-e**] [**-w**] [**-f beg,end**] [**-n nprocs**] [**-t sec**] [**-d jnd**] ***rnmfile***

DESCRIPTION

Ranimove is a program for progressive animation rendering. Variables in the given *rnmfile* indicate input files, output file names, and various other controls and options.

Normally, progress reports are written to the standard output, but the **-s** option tells *ranimove* to do its work silently. The **-e** option tells *ranimove* 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 and non-fatal rendering problems.

Normally, *ranimove* will produce one animation frame for each view given in the specified view file. If the **-f** option is specified, the animation will resume at the given frame, and continue to the end of the sequence, or to the second frame if one is given (separated from the first by a comma but no space).

The **-n** option specifies the number of processes to use for rendering. The default value is 1, which is appropriate for most machines that have a single central processing unit (CPU). If you are running on a machine with multiple CPUs, a larger value up to the number of processors may be used to improve rendering speed, depending on the system load.

Because *ranimove* renders each frame progressively, it needs some criteria for when to proceed to the next frame in the animation. The **-t** option is used to specify the maximum number of seconds to spend on any one frame. The default value for this option is 60 seconds. Additionally, the **-d** option may be used to specify a termination threshold in just-noticeable-differences. If the error can be reduced below this number of JNDs over the whole frame before the time allocation is spent, *ranimove* will then proceed to the next frame. A value of 2.0 JNDs is the point at which 75% of the people will notice a difference, and this is the level usually selected for such a termination test. There is no default value for this option, which means that rendering will proceed until the time allocation is spent for each frame, regardless. If **-t** is set to 0, *ranimove* will spend as much time as it takes to reduce the visible error below the value set by the **-d** option.

Ranimove renders each frame in three stages. In the first stage, a low-quality image is rendered using one ray sample per 16 pixels. In the second stage, pixels from the previous frame are extrapolated to their corresponding positions in this one, based on the given camera and object movements. A set of heuristics is applied to prevent errors in specular highlights and shadows, avoiding some of the errors typical with the *pinterp(1)* program. In the third stage, additional high-quality samples are used to refine important regions of the image that are judged to have visible errors. This proceeds until the stopping criteria specified by the **-t** and **-d** options are met, when the frame is filtered and written to the designated picture file.

The chief differences between this program and *ranimate(1)* are that motion blur is computed for objects as well as camera movement, and its progressive rendering allows better control over the tradeoff between frame accuracy and rendering time. Fewer controls are provided for managing the picture files produced by *ranimove*, and no facilities for distributed rendering are available other than executing *ranimove* on different machines using the **-f** option to manually partition the work.

Animation variable assignments appear one per line in *rnmfile*. 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. Comments in *rnmfile* start with a pound sign (#) and proceed to the end of line.

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

OCTREE The name of the base octree file, which should be generated by the *ocomv(1)* command using the **-f** option. There is no default value for this variable. If no *RIF* variable is given, the octree must be specified.

- RIF** This variable specifies a *rad(1)* input file to use as a source of rendering options and other variable settings. If given, *animate* will execute *rad* and create an options file to control rendering parameters. *Ranimate* will also extract default settings for the common variables: *OCTREE*, *RESOLUTION*, and *EXPOSURE*. 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.
- move** This variable specifies an object (or objects) with a specific motion and/or rendering priority. Four value arguments are expected for each appearance of this variable. The first is the name of a parent move object, or "void" if none. If given, the object's transformation will be prepended to that of its parent. The second argument is the name of this object, which will be used to name surfaces it contains, and as a parent identifier for any child objects that reference it. The third argument is the transformation string or file for this object. If this argument is enclosed in quotes and begins with a hyphen ('-'), then it will be interpreted as a static transform specification a la *xform(1)*. Otherwise, the argument will be taken as the name of a file that contains one such transform specification per line, corresponding to frames in the animation. A period ('.') may be given if no object transformation is desired. The fourth argument is the name of a *RADIANCE* scene file (or files) to be given to *xform* for transformation. If this argument begins with an exclamation point ('!'), then it will be interpreted as a command rather than a file. A final word corresponding to the frame number will be appended to the command, and its output will be passed to the input of *xform* for each frame. An optional fifth argument specifies the rendering priority for this object. Values greater than 1.0 will result in preferential rendering of this object over other portions of the image when it appears in a frame. Values less than 1.0 will cause the rendering to neglect this object in favor of other parts of the image. A value of 3.0 can be interpreted as saying the viewer is three times more likely to look at this object than the background. A file may be given rather than a floating point value, and this file must contain one floating point number per line, corresponding to priorities for frames in the animation.
- VIEWFILE** This variable names a file from which *ranimove* 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. An exception is made for a view file with only a single view, which is used for every frame of the animation. In this case, the *END* variable must also be specified. This variable is required, and there is no default value.
- END** The final frame number in the animation. The default value is computed from the number of views in the given *VIEWFILE*. Normally, this variable will only be given if the view is static.
- EXPOSURE** This variable tells *animate* 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 *animate* 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, no exposure adjustments will be made.
- BASENAME** The base output file name for the final frames. This string should contain a *printf(3)* style integer field to distinguish one frame number from another. The final frames will use this name with a ".hdr" suffix. The default value is "frame%03d".
- MBLUR** This variable specifies the fraction of a frame time that the shutter is simulated as being open for motion blur. Motion blur is computed by *ranimove* using image-based rendering methods in a rather crude fashion. A much better result is achieved by setting *MBLUR=0* and using the *pmblur2(1)* command as a post-process, assigning the *ZNAME* and *MNAME* variables described below. The default motion blur is 0.

RATE This variable specifies the animation frame rate, in frames per second. This is needed to compute the animation error visibility. The default value is 8.

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.

lowq This variable may be used to specify rendering options for the initial, low-quality ray samples. It may be given either as a list of rendering parameter settings, or as variable settings for the *rad* command, in which case the *RIF* variable must also be specified.

highq This variable may be used to specify rendering options for the final, high-quality ray samples. It may be given either as a list of rendering parameter settings, or as variable settings for the *rad* command, in which case the *RIF* variable must also be specified.

oconv This variable may be used to specify special options for *oconv*. See the *oconv(1)* manual page for a list of valid options. (The *-f* option is specified by default.)

ZNAME The base file name for the depth buffer output frames. This string should contain a *printf(3)* style integer field to distinguish one frame number from another. The final frames will use this name with a ".zbf" suffix. This is needed by the *pmblur2* command to compute a better motion blur. There is no default setting for this variable.

MNAME The base file name for the motion vector output frames. This string should contain a *printf(3)* style integer field to distinguish one frame number from another. The final frames will use this name with a ".mvo" suffix. This file is used together with the image and depth buffer by *pmblur2* to compute a better motion blur. In this case, the *MBLUR* variable should be unset or zero. The file will contain 3 16-bit unsigned values per pixel. The first two are the x and y offsets to each pixel in the previous frame, with an offset of 32768 being no movement. The third value is 0 if the previous frame's pixel was on a different object, and 32768 if it was on the same object. There is no default setting for this variable.

EXAMPLES

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

```

:
sample.rnm
:
# The rad input file for our static scene:
RIF= tutor.rif
# The view file containing one view per frame:
VIEWFILE= anim1.vf
# Our central character and its motion:
move= void myguy myguy.xf myguy.rad 2.0

```

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

```
ranimove -n 0 -e sample.rnm
```

This will print the variables we have given as well as default values *ranimove* has assigned for us.

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

```
ranimove sample.rnm >& sample.err &
```

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

```
RES= 1024                # shoot for 1024x resolution
MBLUR= .25               # apply camera motion blur
RATE= 15                 # 15 frames/second
EXP= anim1.exp           # adjust exposure according to file
lowq= QUAL=Low           # low quality ray sampling
highq= QUAL=Med          # high quality ray sampling
```

Note the use of abbreviation for variable names.

AUTHOR

Greg Ward

SEE ALSO

fieldcomb(1), oconv(1), pfilt(1), pinterp(1), pmblur2(1), pvalue(1), rad(1), ran2tiff(1), ranimate(1), rpict(1), xform(1)