

NAME

rpiece - render pieces of a RADIANCE picture

SYNOPSIS

```
rpiece [ -v ] [ -x xres ] [ -y yres ] [ -X xdiv ] [ -Y ydiv ] [ -FR syncfile ] [ -T timelim ] [ $EVAR ] [ @file ] [ rpict options ] -o picture octree
```

DESCRIPTION

Rpiece renders a RADIANCE picture a piece at a time, calling *rpict(1)* to do the actual work. This is useful for running multiple *rpict* processes on cooperating machines to render a single picture, which is a shared file specified with the *-o* argument. The overall picture dimensions will be *xres* by *yres* (or smaller, depending on the *-pa* option and other view options), and the picture will be rendered in *xdiv* by *ydiv* pieces.

There are two basic methods for telling *rpiece* which piece(s) of a picture to render. The explicit method is to write on the standard input the *X* and *Y* position of the desired piece(s), where *X* runs from zero to *xdiv-1* and *Y* runs from zero to *ydiv-1*. (The lower left piece of a picture corresponds to (0,0) in this system.) Alternatively, the implicit specification method uses a synchronization file to determine which piece is to be rendered next. Specified with the *-F* option, *syncfile* initially contains the values for *xdiv* and *ydiv*, so the *-X* and *-Y* options are unnecessary. (However, they are used if *syncfile* does not exist.) The first *rpiece* process puts a lock on *syncfile* and modifies its contents before starting work on the first piece of the image. It writes the *X* and *Y* position of the piece it will work on, so the next *rpiece* process to modify *syncfile* will start on the next piece. (When it finishes with its piece, it appends the index to the end of *syncfile*.) This procedure continues until all the pieces are done, at which point all of the *rpiece* processes will terminate.

The *-R* option may be used instead of *-F* if some of the pieces were not properly finished by previous (killed) runs of *rpiece*. This option should be used by at most one *rpiece* process, which must be started first and with *no other rpiece processes running* or else it will rerender the same pieces other processes have begun. Once the recover process is started, you may start other *rpiece* processes using the *-F* option to run simultaneously. If some processes die during execution, leaving one or more half-finished pieces in the picture even though the other processes think the work is done, you may run a single *rpiece* with the *-R* option by itself to repair the holes.

The *-v* flag switches on verbose mode, where *rpiece* reports to the standard output after each piece begins and after each piece is finished.

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.

EXAMPLE

First *rpiece* process is started on the machine "goober":

```
goober% echo 1 8 > syncfile
goober% echo -F syncfile -x 1024 -y 1024 -vf view -o picture octree > args
goober% rpiece @args &
```

Second *rpiece* processes is started on the machine "sucker":

```
sucker% rpiece @args &
```

NOTES

Due to NFS file buffering, the network lock manager is employed to guarantee consistency in the output file even though non-overlapping writes are used. This would tend to slow the process down if *rpiece* were to wait for this I/O to complete before starting on the next piece, so *rpiece* forks separate processes to hang around waiting for I/O completion. The number of processes thus designated is set by the MAXFORK macro in the program (compiled in the src/util directory). If the fork call is slow on a system, it may actually be better to set MAXFORK to zero. In other cases, the network lock manager may be so slow that this value should be increased to get the best utilization.

The output picture is not run-length encoded, and can be quite large. The approximate size (in kilobytes) can be computed by the simple formula:

$$\text{filesize} = \text{xres} * \text{yres} / 256$$

Make sure that there is enough space on the filesystem to hold the entire picture before beginning. Once the picture is finished, the *ra_rgb(1)* program with the *-r* option may be used to convert to a run-length encoded picture for more efficient storage, although *pfilt(1)* or any of the other Radiance picture filters will do the same thing.

The ALRM signal may be used to gracefully terminate an *rpiece* process after it finishes the current piece. This permits other currently running or subsequently started *rpiece* process(es) to continue rendering the picture without loss. The *-T* option will send the ALRM signal to *rpiece* after the specified number of (decimal) hours. This is the best way to force a time limit on the computation, since information will not be lost, though the process may continue for some time afterwards to finish its current piece.

BUGS

This program may not work on some systems whose NFS lock manager is unreliable. In particular, some System V derivative UNIX systems often have problems with the network lock manager. If the output is scrambled or *rpict* aborts with some ambient file related problem, you should just remove the ambient file and go back to normal rendering.

AUTHOR

Greg Ward

SEE ALSO

getinfo(1), *pfilt(1)*, *ra_rgb(1)*, *rpict(1)*, *rtpict(1)*, *rxpiece(1)*, *ximage(1)*