

**NAME**

rhoptimize - optimize beam locations in holodeck file

**SYNOPSIS**

**rhoptimize** [ **-u** ] **src\_holo** [ **dest\_holo** ]

**DESCRIPTION**

*Rhoptimize* optimizes beam positions and eliminates fragment waste in the holodeck file *src\_holo*, writing the result to *dest\_holo*, or back into *src\_holo* if only one argument is given. This may improve rendering speed on large holodecks, which tend to have widely dispersed beam positions that cause delays due to long file seeks. It may also reduce the size of the file, since large holodecks become fragmented as they fill up with new ray samples. (Use the *rhinfo(1)* program to determine holodeck file fragmentation.) The *-u* option adds a check to make sure that each ray sample is unique, i.e., the same sample is never given twice. This check is usually unnecessary, but may eliminate redundant samples from some holodeck files.

**EXAMPLE**

To optimize the beam order in old.hdk and write the results to new.hdk:

```
rhoptimize old.hdk new.hdk
```

To optimize beam order in scene.hdk:

```
rhoptimize scene.hdk
```

**NOTES**

If *rhoptimize* is given only one file argument, it creates a temporary file in the same directory, and moves it onto the original file once it has successfully completed its operation. If the operation fails for some reason, the temporary file is removed and the original holodeck is left unchanged.

This program generally takes several minutes to run and offers no progress reports.

**FILES**

rho????.hdk      temporary file to hold new holodeck

**AUTHOR**

Greg Ward Larson

**ACKNOWLEDGMENT**

This work was supported by Silicon Graphics, Inc.

**SEE ALSO**

getinfo(1), pfilt(1), psign(1), rhcopy(1), rhinfo(1), rholo(1), rpict(1)