

NAME

rcrop - crop RADIANCE matrix or picture

SYNOPSIS

rcrop *row0 col0 nrows ncols* [*input* [*output*]]

DESCRIPTION

Rcrop copies the indicated section of a matrix or RADIANCE picture. The four required arguments are the initial row (scanline) to copy, the initial column, the number of rows to copy, and the number of columns to include per row. Note that rows are numbered from 0 at the top of a matrix or picture, and columns are changing fastest in the input, which can have any number of components per matrix element.

The input must have a valid header, including dimensions and format, and a header will always be produced. This is similar behavior to *rmtxop(1)*. If the input header is missing, one can usually be added with an appropriate call to *rcollate(1)*. If an output header is not desired, it can be removed using *getinfo(1)*.

One cannot specify crop boundaries outside the matrix dimension limits. If a 0 value is specified for the number of rows to copy, then all rows will be copied, starting from row0. Similarly, a 0 value for the number of columns to copy implies all columns \geq col0 will be copied.

Using *rcrop* on a RADIANCE picture rather than *pcompos(1)* corrects the view parameters if present, and preserves the exposure in a way that is compatible with *evalglare(1)*. However, one must remember that rows correspond to Y coordinates, and the orientation is reversed from a standard picture, where Y=0 is the bottom scanline.

All matrix formats are supported, including text (ASCII), binary float and double. The *rcrop* tool also works on encoded normal vector files produced by *rcode_norm(1)* and encoded depth files produced by *rcode_depth(1)*. Data is copied verbatim, i.e., there is no internal conversion to another representation, although white space may change in text files.

If no input file is given, data will be taken from the standard input. If no output file is specified, data will be copied to the standard output.

EXAMPLES

To crop a matrix down to 15 rows by 19 columns, starting at row 10 and column 12:

```
rcrop 10 12 15 19 input.mtx > output.mtx
```

To extract column 35 from a matrix:

```
rcrop 0 35 0 1 orig.mtx > col35.mtx
```

To crop a picture to (X by Y) = (1500 by 1000), starting from upper-left corner:

```
rcrop 0 0 1000 1500 orig.hdr cropped.hdr
```

Two ways of computing the inner product of row 15 in matrix 1 by column 27 in matrix 2 and extracting the result without a header, the second command being more efficient:

```
rmtxop -fa mat1.mtx mat2.mtx | rcrop 15 27 1 1 | getinfo -
```

```
rmtxop -fa '!rcrop 15 0 1 0 mat1.mtx' '!rcrop 0 27 0 1 mat2.mtx' | getinfo -
```

AUTHOR

Greg Ward

SEE ALSO

evalglare(1), *pcompos(1)*, *rcalc(1)*, *rcode_depth(1)*, *rcode_norm(1)*, *rcollate(1)*, *rfluxmtx(1)*, *rmtxop(1)*, *rsplit(1)*