

NAME

dctimestep - compute annual simulation time-step(s) via matrix multiplication

SYNOPSIS

```
dctimestep [ -n nsteps ] [ -h ] [ -o ospec ] [ -x xres ] [ -y yres ] [ -i{fld} ] [ -o{fldlc} ] DCspec [ skyf ]
dctimestep [ -n nsteps ] [ -h ] [ -o ospec ] [ -i{fld} ] [ -o{fld} ] Vspec Tbsdf Dmat.dat [ skyf ]
```

DESCRIPTION

Dctimestep has two invocation forms. In the first form, *dctimestep* is given a daylight coefficient specification and an optional sky vector or matrix, which may be read from the standard input if unspecified. The daylight coefficients are multiplied against these sky values and the results are written to the standard output. This may be a list of color values or a combined Radiance picture, as explained below.

In the second form, *dctimestep* takes four input files, forming a matrix expression. The first argument is the View matrix that specifies how window output directions are related to some set of measured values, such as an array of illuminance points or pictures. This matrix is usually computed by *rfluxmtx(1)* or *rcontrib(1)* for a particular set of windows or skylight openings. The second argument is the window transmission matrix, or BSDF, given as a matrix or a standard XML description. The third argument is the Daylight matrix file that defines how sky patches relate to input directions on the same opening. This is usually computed using *rfluxmtx* with separate runs for each window or skylight orientation. The last file is the sky contribution vector or matrix, typically computed by *genskyvec(1)* or *gendaymtx(1)*, and may be passed on the standard input.

If the input sky data lacks a header, the *-n* option may be used to indicate the number of time steps, which will be 1 for a sky vector. The sky input file must otherwise contain the number of columns (time steps) specified in each sky patch row, whether it is read from the standard input or from a file. Input starts from the first patch at the first time step, then the first patch at the second time step, and so on. Note that all matrix elements are RGB triplets, so the actual size of the sky vector or matrix is three times the number of steps times the number of sky patches. The *-if* or *-id* option may be used to specify that sky data is in float or double format, respectively, which is more efficient for large matrices. These options are unnecessary when the sky input includes a header.

Any of the matrix or vector files may be read from a command instead of a file by using quotes and a beginning exclamation point ('!').

The standard output of *dctimestep* is either a color vector with as many RGB triplets as there are rows in the View matrix, or a combined *Radiance* picture. Which output is produced depends on the first argument. A regular file name will be loaded and interpreted as a matrix to generate a color results vector. A file specification containing a '%d' format string will be interpreted as a list of *Radiance* component pictures, which will be summed according to the computed vector.

The *-o* option may be used to specify a file or a set of output files to use rather than the standard output. If the given specification contains a '%d' format string, this will be replaced by the time step index, starting from 0. In this way, multiple output pictures may be produced, or separate result vectors (one per time step). If input is a matrix rather than a set of pictures, the *-x* and/or *-y* options may be necessary to set the output picture size. If only one dimension is specified, the other is computed based on the number of rows in the result vectors.

A header will normally be produced on the output, unless the *-h* option is specified. The *-of*, *-od*, or *-oc* option may be used to specify IEEE float, double, or RGBE (picture) output data, respectively.

EXAMPLES

To compute workplane illuminances at 3:30pm on Feb 10th:

```
gensky 2 10 15:30 | genskyvec | dctimestep workplaneDC.dmx > Ill_02-10-1530.dat
```

To compute a picture at 10am on the equinox from a set of component pictures:

```
gensky 3 21 10 | genskyvec | dctimestep dcomp%03d.hdr > view_03-21-10.hdr
```

To compute a set of illuminance contributions for Window 1 on the Winter solstice at 2pm:

```
gensky 12 21 14 | genskyvec | dctimestep IllPts.vmx Blinds20.xml Window1.dmx > Ill_12-21-14.dat
```

To compute Window2's contribution to an interior view at 12 noon on the Summer solstice:

```
gensky 6 21 12 | genskyvec | dctimestep view%03d.hdr Blinds30.xml Window2.dmx > view_6-21-12.hdr
```

To generate an hourly matrix of sensor value contributions from Skylight3 using a 3-phase calculation, where output columns are time steps:

```
gendaymtx -of Tampa.wea | dctimestep WPpts.vmx shade3.xml Skylight3.dmx > wp_win3.dat
```

Generate a series of pictures corresponding to timesteps in an annual simulation:

```
gendaymtx NYCty.wea | dctimestep -o tstep%04d.hdr dcomp%03d.hdr
```

To multiply an irradiance view matrix through a pair of XML window layers using a given exterior daylight matrix and sky vector:

```
dctimestep Illum.vmx "!rmtxop -ff Blinds1.xml Windo1.xml" Exter.dmx Jan20.sky
```

To multiply two matrices into a IEEE-float result with header:

```
dctimestep -of Inp1.fmx Inp2.fmx > Inp1xInp2.fmx
```

NOTES

Dctimestep optimizes its matrix concatenation by checking for all-zero rows or columns, thus avoiding unnecessary vector multiplications. This can improve performance when a daylight matrix contains zero-filled column vectors corresponding to hours of darkness.

It rarely makes sense to specify the *-od* output option with *dctimestep*, since matrix operations are carried out using 32-bit "float" values. This takes less memory, but can also be less accurate than an equivalent invocation of *rmtxop(1)*, which performs all operations on 64-bit "double" values.

AUTHOR

Greg Ward

SEE ALSO

gendaymtx(1), genskyvec(1), getinfo(1), mkillum(1), rcollate(1), rcontrib(1), rcrop(1), rfluxmtx(1), rmtxop(1), rtrace(1), vwrays(1)