NAME

pvsum - sum component Radiance pictures based on vector or matrix

SYNOPSIS

pvsum [**−o ospec**][**−o{flc}**][**−N nproc**] **ispec** [**mtx**]

DESCRIPTION

Pvsum is an efficient tool for summing Radiance pictures into one or more output pictures. Similar to *dctimestep(1),* multiple frames are produced if the input *mtx* has more than one column. The number of rows in this matrix corresponds to the number of component pictures specified in *ispec* with an incorporated "%d" or similar format string. The first row in the matrix corresponds to picture 0, and counting proceeds to one less than the number of matrix rows.

If no *mtx* is specified on the command line, the required data is read from the standard input. The input matrix must either have a single component or match the number of components in the input pictures, and the output pictures will match the latter component count. (I.e., grayscale float pictures have a component count of 1, RGB and XYZ pictures a count of 3, and spectral pictures typically have 6 or more components.)

By default, the output pixel type will match that of the input pictures, either floating point or common-exponent byte format (i.e., RGBE, XYZE, or Radiance_spectra). If float output is preferred, specify the *−of* option. If common-exponent byte format is preferred, use the *−oc* option.

If a *−o ospec* argument begins with an exclamation point ('!') and contains a "%d" format string or similar, then a separate command will be executed for each output stream. If no *−o* option is given, all data is sent to the standard output, which may be a sequence of Radiance pictures as understood by *ra_rgbe(1)* in the case of a multi-column input matrix. (Note that there is currently no Radiance tool that fully handles a concatenated series of float or spectral pictures.)

The *−N* option may be used on Unix systems to specify the number of processes to employ in the summations. This setting has an upper limit equal to the count of matrix columns, but the optimal number of processes depends on several factors. Setting the process count above the number of physical cores may offer some benefit on large input collections if their total size significantly exceeds the available system RAM. Experimentation with this setting is therefore encouraged.

EXAMPLES

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

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

To compute a set of hourly spectral pictures at SFO airport from a weather tape and set of Tregenza component pictures using 10 processes:

```
gensdaymtx -of sfo.epw | pvsum -o timestep%04d.hsr -N 10 tregcomp%03d.hsr
```

NOTES

This tool overlaps with *dctimestep,* but provides some missing capabilities. Foremost, *pvsum* reads and can produce spectral pictures and matrices, whereas *dctimestep* expects and requires 3-component pictures and matrices throughout. In addition, *pvsum* accelerates picture sums on Unix systems with more memory and CPU cores. Operations were simplified by focusing on the Daylight Coefficient command form, where the DC matrix is represented as a collection of pictures. Finally, *pvsum* offers more flexible floating-point support and can output to commands as well as files.

AUTHOR

Greg Ward

SEE ALSO

dcglare(1), dctimestep(1), gendaymtx(1), gensdaymtx(1), genskyvec(1), getinfo(1), mkillum(1), ra_rgbe(1), rcollate(1), rcomb(1), rcontrib(1), rcrop(1), rfluxmtx(1), rmtxop(1), rtrace(1), vwrays(1)