What's New in Radiance 5.3

Greg Ward, Anyhere Software

Bug Fixes & Incidental Improvements

- Object picking in **rvu** now ignores transparent surfaces
- Fixed issue with large file handling in **rcollate**
- Output series now indexed from 0 in **dctimestep**
- Improvements to **bsdf2klems** accuracy
- Added min() and max() functions to .cal library
- Added functionality to **getinfo** with -d and +d
- Improved performance of **rmtxop** program
- Added automatic byte-swapping for float/double

More Significant Additions

- Created **radcompare** program as part of larger effort to include unit testing in automatic builds
- Added **rtrace** -orxRX options to separate reflected and unreflected radiance values and distances
- Created **rsplit** program to compliment **rlam** and enable layered image storage with enhanced **rtpict**
- Created **rcode_depth**, **rcode_norm**, **rcode_ident**, and **rcode2bmp** conversion tools to support new depth, normal, and identifier file formats
- Added **gendaymtx** -A option to produce average sky vectors

New **radcompare** Utility

- Takes two *Radiance* files and determines equivalence
- Identifies floating-point values and allows for some discrepancy within specified tolerances
- Smart about ignoring white space and identifying numerical data in text files
- Understands most *Radiance*-specific binary files
  - Reads information headers and looks for differences there as well
  - Some files types compared as binary, not allowing for discrepancies
    - octrees, triangle meshes, ID maps, scattering interpolants
- Provides robust regression testing by allowing variation in results, especially important for HDR image comparison

Simple **radcompare** Examples

rmtxop -ff -c .3 .9 .2 test.mtx -c .7 .2 .3 -t test.mtx > rmtxop.mtx
radcompare ref/rmtxop.mtx rmtxop.mtx
radcompare: warning - headers are different lengths

rmtxop -v -ff -c .3 .7 .2 test.mtx -c .7 .2 .3 -t test.mtx > rmtxop2.mtx
radcompare -v ref/rmtxop.mtx rmtxop2.mtx
radcompare: warning - headers are different lengths
radcompare: input file type is float
radcompare: comparing inputs as 32-bit IEEE floats
radcompare: relative RMS difference between 'ref/rmtxop.mtx' and 'rmtxop2.mtx' of 0.011296 exceeds limit of 0.01

New **rtrace** -orxRX output

- `-or` outputs reflected radiance component
- `-ox` outputs unreflected radiance component
- `-oR` outputs reflected distance
- `-oX` outputs unreflected distance
- `-ov` is always the sum of -or and -ox
- Effective depth (`-ol`) is either `-oR` or `-oX` depending on more significant radiance component
- New options are not compatible with `-I+` or `-i+`

Standard **rtrace** -ov Image

Example **rtrace** -or Image

Example **rtrace** -ox Image

New **rsplit** utility

- Acts like inverse to **rlam**
  - Rather than pulling data together into a single stream, it splits it apart
- Most useful for separating output from **rtrace** into component files
- Example to save image and z-buffer from **rtrace**:

```
vwrays -ff -vf view | rtrace -fff `vwrays -vf view -d` -olv octree \
    | rsplit -ih -iH -of depth.zbf -oh -oH -of3 - \
    | pvalue -r -df > image.hdr
```

- The initial float value for each pixel is the depth, and each one is written in binary to the file "depth.zbf", which has no header
- The next three float values are the RGB radiance, and are sent to stdout with the header and resolution to be converted by **pvalue**

New *Radiance* File Types

- We can now split all the outputs from **rtrace**, but do we want to store them as 32-bit/component floats?
- The **rcode_depth** utility encodes and decodes 16-bit depth values that have better characteristics than short-integer or half-float representations
- The **rcode_norm** utility encodes and decodes 32-bit normal vectors that take 1/3 as much space as floats while still providing excellent precision
- The **rcode_ident** utility uses an indexed format for storing string identifiers (e.g., material names)
- Layered image output is now supported by **rtpict**

Why Layered Images?

- Sometimes referred to as "deep pixels," layered image formats provide a way to store more than just RGB color; OpenEXR is a good example
- Efforts began with Rob Shakespeare and Bill Thompson to support their low-vision analysis work
- Initially, we planned to make an enhanced HDR format to add depth, world position, and surface normals to cloud-shared *Radiance* renderings
- Realized that a single-file deep pixel format would break the Unix toolbox model that has served us so well, but there was another way….

The **rtpict** tool (1)

- Last year, I created the *Perl* script **rtpict** to run **vwrays** with **rtrace** (as others have done) to get something like **rpict** with multiprocessing
- Adding support for layered image output was relatively straightforward, e.g.:
  - **rtpict** -n 4 -ovLn layered -vf view octree
- The above command writes 3 separate files in the directory "layered":
  - radiance.hdr - a flattened HDR image for this view
  - d_firstsurf.dpt - 16-bit encoded first intersection depths
  - perturbed.nrm - 32-bit encoded surface normals at each pixel

The **rtpict** tool (2)

- Layer files are always given the same name and are uncompressed to facilitate random access
  - Standard compression tools generally do better than compressed formats

rtpict -o options and output files

v	radiance.hdr
r	r_refl.hdr
x	r_unrefl.hdr
l	d_effective.dpt
L	d_firstsurf.dpt
R	d_refl.dpt
X	d_unrefl.dpt
n	perturbed.nrm
N	unperturbed.nrm
s	surface.idx
m	modifier.idx
M	material.idx

rtpict Subtlelties

- Replicates **rpict** parameters and defaults, except:
  - `-r, -ro, -S, -P,` and `-PP` are *unsupported*
  - `-t, -ps, -pt, -pd,` and `-pm` are silently *ignored*
- Results may look better with `-u+` sampling
- Some -o* options are not supported
  - Avoid `-otTwVWcp` (trace, weight, contribution, coefficient, (u,v) coordinates, intersection point)
  - Intersection point is supported indirectly by **rcode_depth**
- Using `-i+` with `-ov` writes "irradiance.hdr"
- Add to existing layers by re-using output directory
  - Recommend setting `-pj 0` in this case
- Include `-d` ref/unit when encoding depth layer(s)
- No tiny source sampling on image plane
  - Little-know feature of **rpict** avoids some aliasing artifacts

Details of **rcode_depth**

- Uses a 16-bit hybrid linear/reciprocal encoding
  - Similar to linear/log depth encoding in holodeck
  - Explicit 0 and ∞ representations
- Specify a *reference depth* approximating the "far" distance in local scene
  - Indicates cross-over between linear and reciprocal encoding
  - Defaults to 1.0 in anonymous units
  - Codes -32768 to -1 cover linear range from 0 to reference depth
  - Codes 0 to 32767 cover reciprocal range from reference to ∞
- Option `-d` ref_dist/unit includes distance units

```
rtrace -oL -x xr -y yr octree | rcode_depth -d 8/meter > L.dpt
```

- View parameters stored in header enable **rcode_depth** to decode world intersection points

Details of **rcode_norm**

- Uses same 32-bit normal encoding as ambient files
  - Implementation in src/common/dircode.c
- Average error ~8 seconds of arc
  - Maximum error is 21 seconds of arc
- Takes 1/3 as much space as raw float normals

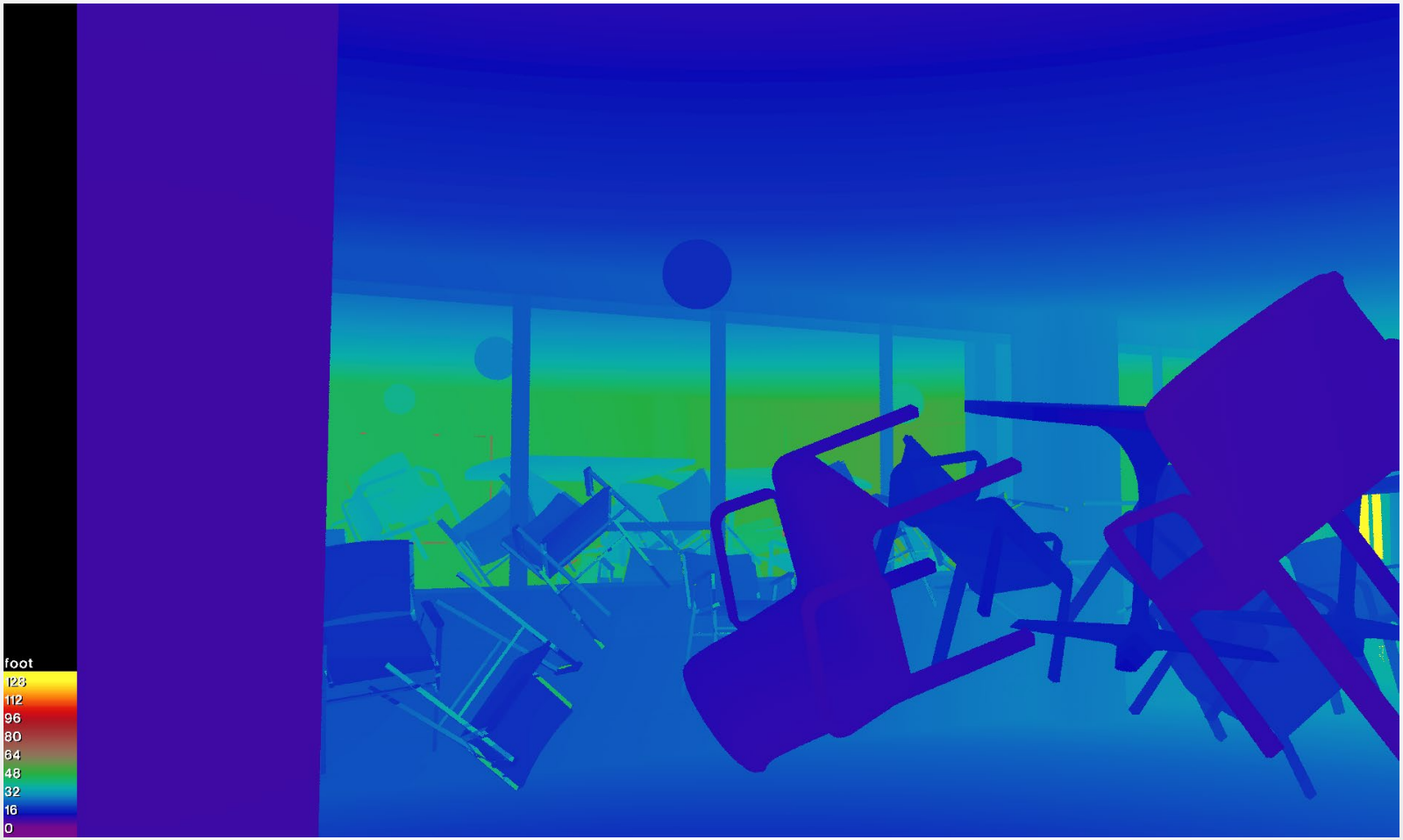
Details of **rcode_ident**

- Creates index table with identifiers encountered on input
- Index width can be 8, 16, or 24 bits
  - Corresponds to maxima of 256, 64K, or 16M unique identifiers
  - **rtpict** always uses 16-bit index
- Like other rcode formats, files are uncompressed, but compress very well using **gzip** or similar

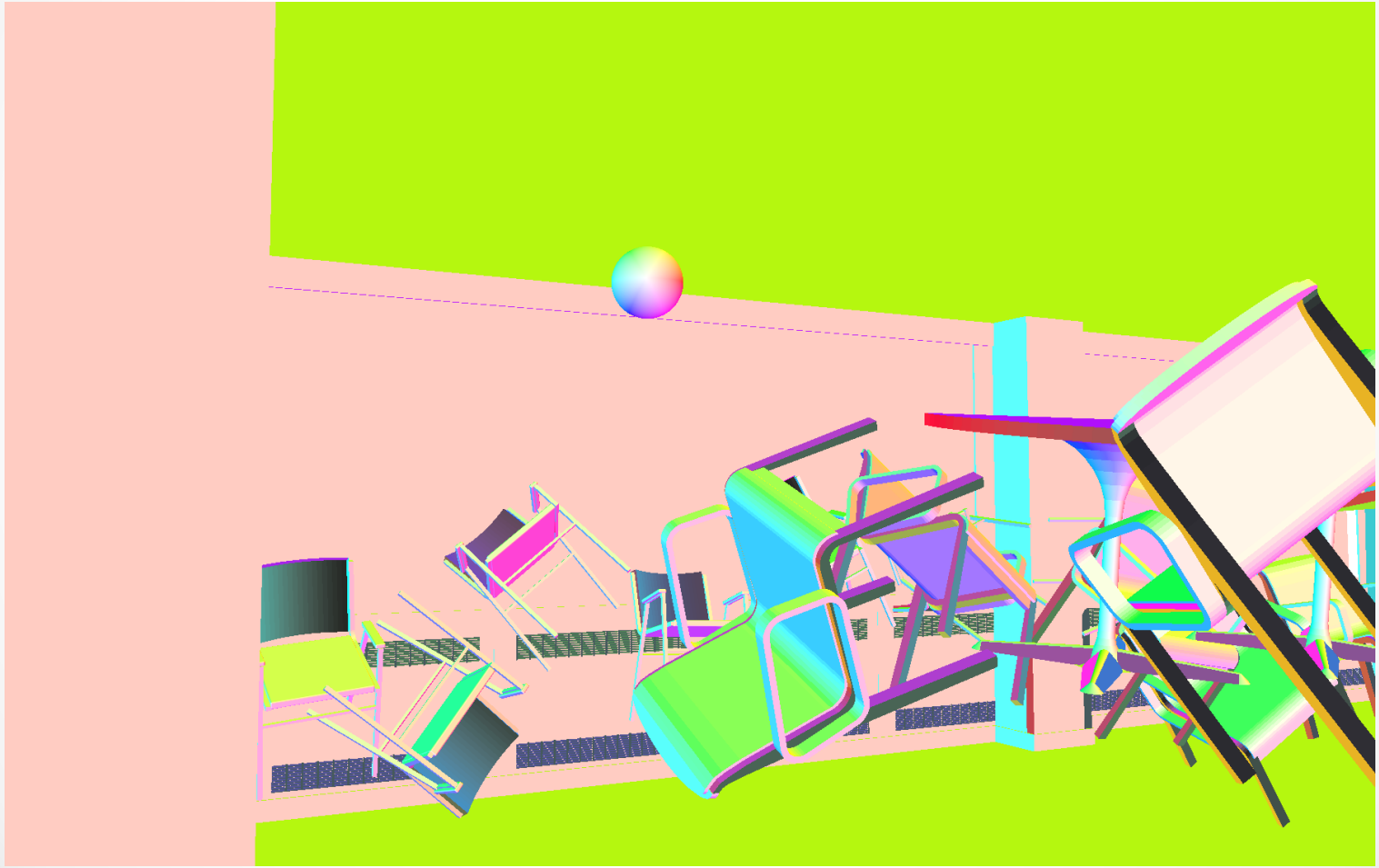
Related **rcode2bmp** Tool

- Takes a list of depth, normal, and identifier files and produces BMP images for quick visualization
- *Perl* script uses a different trick for each data type
  - Depths are visualized by calling **falsecolor**
  - Normals use a HSV color wheel encoding relative to view direction
  - Identifiers are assigned random colors
- Also handles HDR files, calling **`ra_bmp`** `-e auto`

Effective Depth Visualiztion

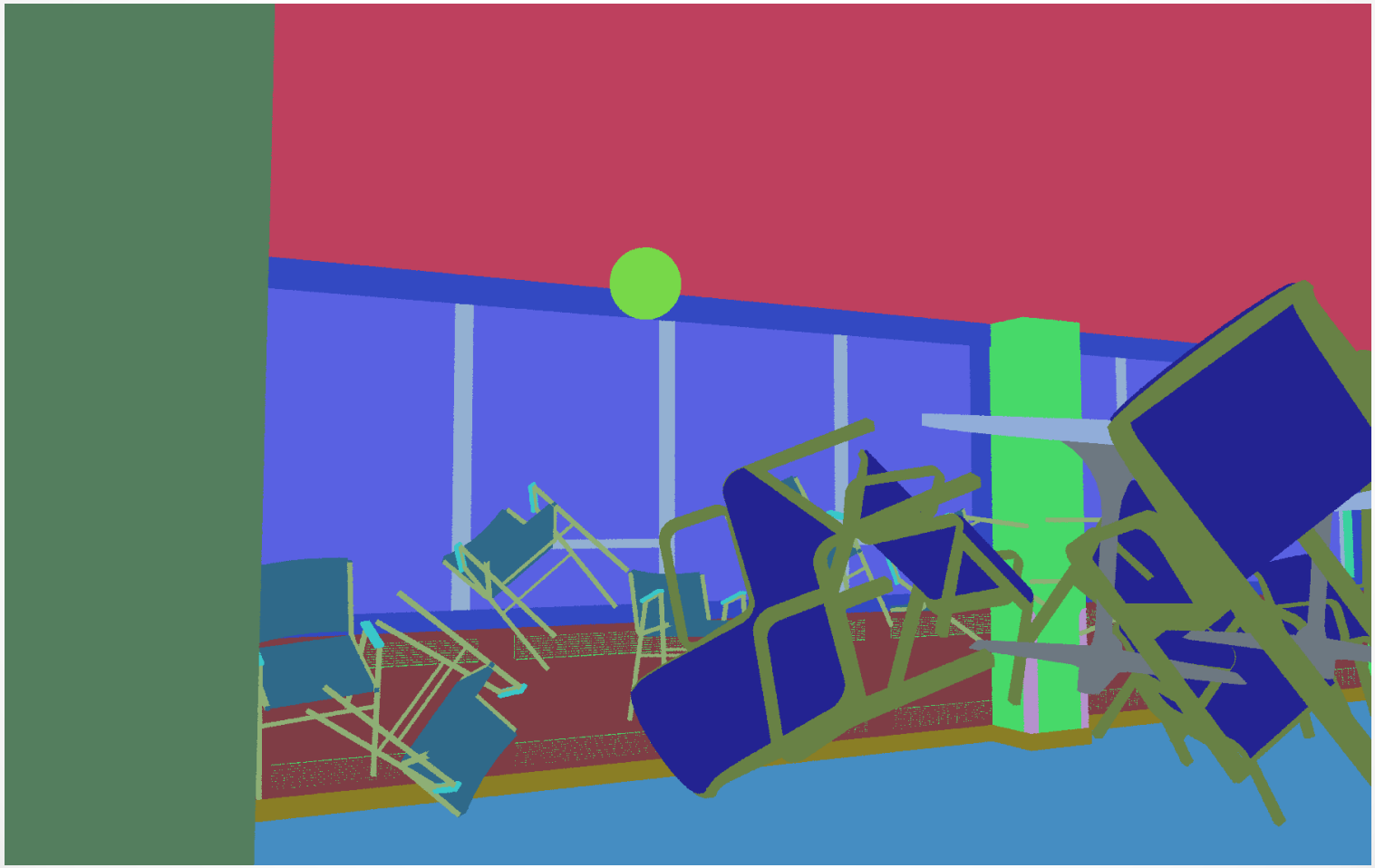


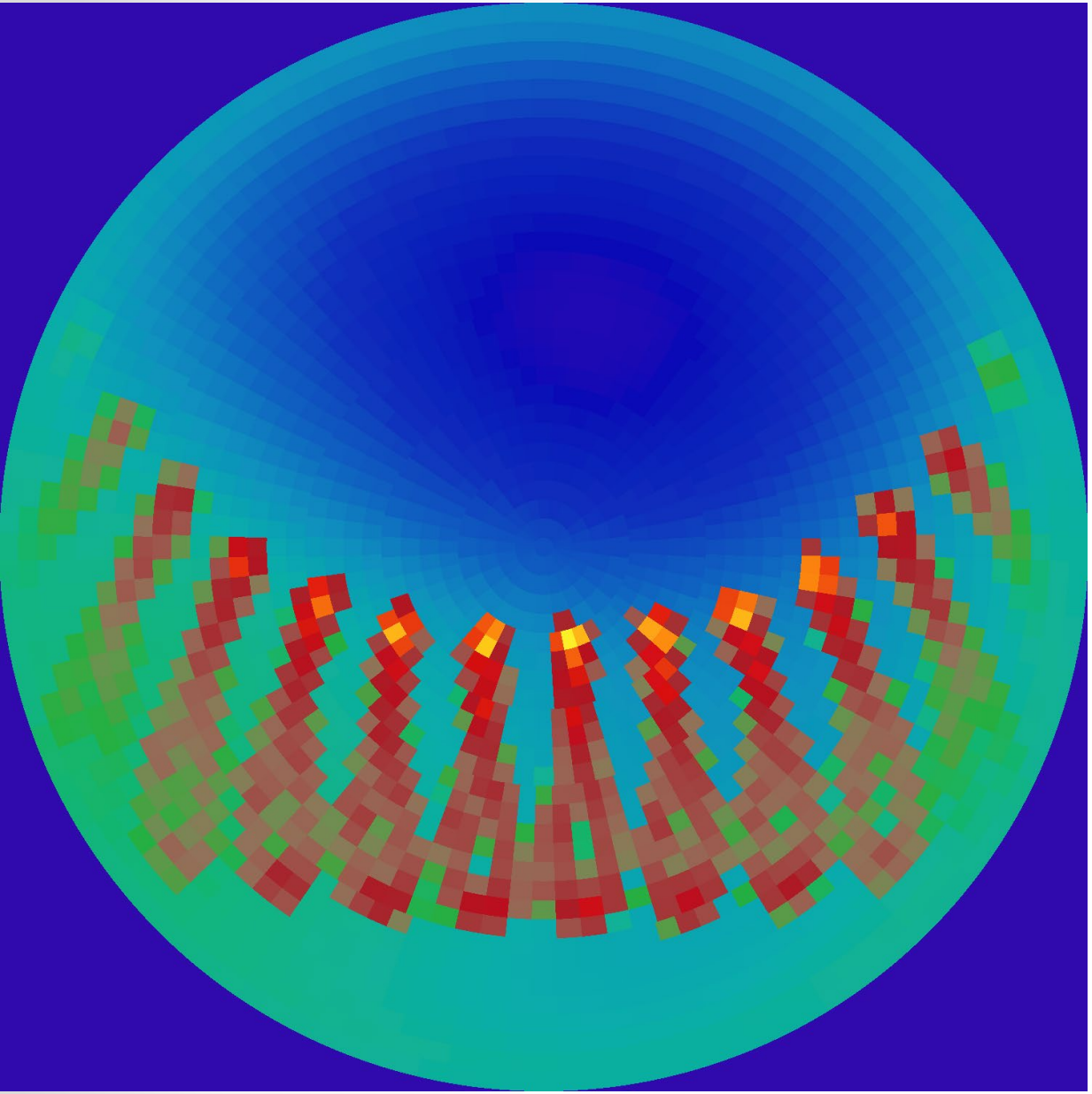
Unperturbed Surface Normals



Darker normals are facing away from camera (reversed)

Material IDs





New option: **gendaymtx -A**

- Averages sky patch values over entire weather tape
- Useful in PV exposure analysis and other applications

Other Happenings…

- Rob Guglielmetti no longer at NREL, but working to produce a new set of installers all the same
  - NREL servers will continue hosting distro for time being
- Official 5.3 release likely in the next month or so
  - Just need to roll it out -- not waiting for anything in particular
- Hope to continue working on unit testing
  - Tests in place for renderers, most generators, and a few utilities
  - Much work left to be done -- funding priorities make it slow going
- Nathaniel Jones has a new annual glare simulation
  - Look forward to learning more about it this afternoon

¿Questions?