

What's New for *Radiance* in 2026

Greg Ward, Anywhere Software

Taoning Wang, LBNL

Minor Improvements & Bug Fixes

- Fixed **gensdaymtx** bug yielding suns 40x too bright
- Fixed bug in **genssky** that was not parsing location
- Fixed issues with **rt pict** under Windows
 - thanks to Mehlika Inanici for pointing out problems
- Added metadata i/o to **ra_bmp**
- TW made some significant Windows speed-ups, including better random number generator
- Sped up IES light calculations in “source.cal”
- Helped Alstan update *Spectral Materials Database*
- Changed **oconv** “set overflow” error to warning

Major Improvements

- Created **rxfluxmtx** tool based on C++ library that includes view options (avoiding **vwrays**)
- New **-bj** option for **rfluxmtx** and **rcontrib** to blur bins
- New **pvsum -m** option to optimize memory caching
- Yongqing Zhao added vectorizing and multi-processing to **dctimestep** for animation (view) sequences, with cross-platform support
- New version of **evalglare** from Jan Wienold
- TW added experimental multiprocessing support for **rcontrib** under Windows, previously only on Unix

Created **rxfluxmtx** Tool

- Superset of **rfluxmtx** options
 - Adds view and pixel options, more efficient than running **vwrays**
- C++ class avoids separate **rcontrib** process
 - Theoretically more efficient, but tests inconclusive
- ASCII output and stdout unsupported
 - Requirement for memory-mapped output and recovery
- Limit is lifted on number of receiver surfaces
 - **rfluxmtx** passes modifier names using **rcontrib** command-line
- Not available under Windows w/o WSL

New rxfluxmtx Tool

- Original **rfluxmtx** command:

```
vwrays -fd -pj .8 -vf seat1.vf -c 8 -x 1024 -y 1024 \  
| rfluxmtx -v -n 10 -fdc @rcopts.txt \  
`vwrays -vf seat1.vf -x 1024 -y 1024 -d` -c 8 \    gets dimensions  
- window/window_kf.rad base_model.rad \  
> mtx/seat1/comp%03d.hdr
```

- Simpler **rxfluxmtx** equivalent:

```
rxfluxmtx -v -n 10 -pj .8 -vf seat1.vf -fc -c 8 -x 1024 -y 1024 \  
@rcopts.txt -o mtx/seat1/comp%03d.hdr \  
window/window_kf.rad base_model.rad
```

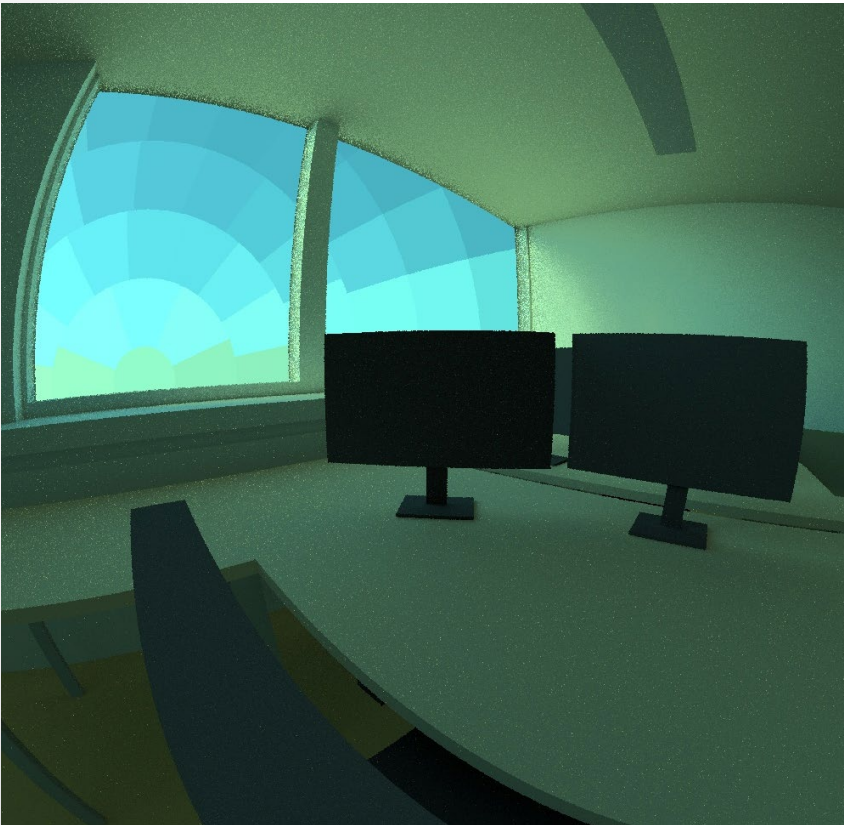
Most X's of any *Radiance* tool!

New **-bj** Bin Blur Option for Matrix Computation

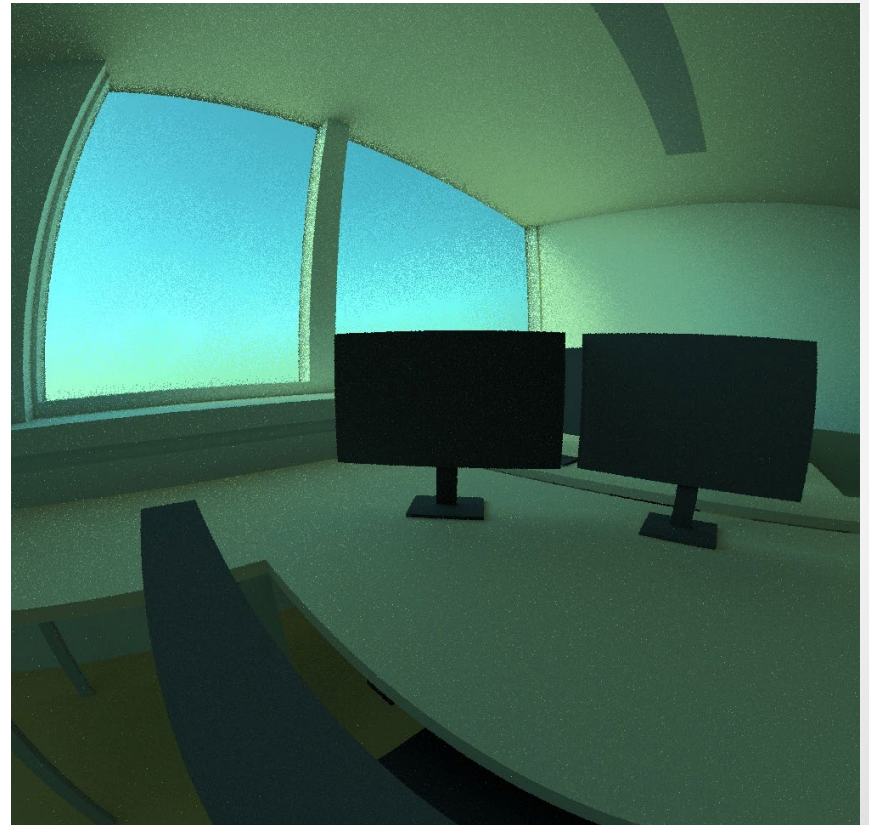
- Added to **rcontrib**, **rfluxmtx**, and **rxfluxmtx**
- Jitters bin intersections the specified amount (0-1)
 - Previous behavior is new default: **-bj 0**
- Any time ray intersects a binned receiver, jitter is applied before bin test
- With **-bj 1**, all bin boundaries are turned to noise
 - Similar to penumbras

Bin Jitter Comparison

rfluxmtx -bj 0



rfluxmtx -bj 1



New **pvsum -m** Option to Optimize Memory

- Reminder: **pvsum** is like **dctimestep** but adds handling of spectral images and matrices in various data formats
 - Multi-processing with memory sharing on Unix only
 - Described at last year's workshop
- New **pvsum -m** option specifies memory cache size
 - Adds loop to prevent overrunning memory and using swap
 - Recommend that **-N** option is set to number of physical cores if used in combination

Yongqing Zhao Sped up **dctimestep** for Images

- Adds multi-processing to **dctimestep** (**-N** option) for image summation
- Initially announced by Yongqing on *Radiance Discourse* in April 2026
 - First version not cross-platform and used too much disk space
- After several iterations and much back-and-forth testing and analysis, we had a memory-efficient, cross-platform version
- Considerably faster than **pvsum** for RGBE inputs!

New version of evalglare from Jan Wienold

- Improved center-of-source calculation
- Better below-horizontal position index calculation
- Release as part of the current HEAD (6.1a)

Added **rcontrib** multi-processing for Windows

- Existing **-n** option enabled for Windows by Taoning
- Employs multiple “slave” processes run by original invocation, similar to Unix model but without fork() call requirement
- Incorporated into latest alpha (HEAD) code
 - Let us know if it works for you, more testers needed

Update to Photosphere

- Erich Phillips has applied OpenAI's Codex tool to create a cross-platform version of Photosphere
 - Leta Eydelberg made original port from Mac to Windows circa 2009
 - linked to cross-platform *wxWidgets* GUI rather than 32-bit *Carbon*
 - Mac port-back was stalled by *wxWidgets* 64-bit *Cocoa* update and Life
- Erich has thus far:
 - Resurrected original Windows XP version on VM
 - Updated old code to work with new compiler under Windows 10
 - Ported to latest macOS running on Apple Silicon
- Still to do:
 - Debug and fix issues found under macOS
 - Update with changes to GUI and libraries since 2016
 - Port updates back to Windows, then onto Linux
 - Release as freeware to the civilized world, if it still exists

Speaking of AI...

- Lots of good applications for narrow AI, but government regulation needs to catch up
 - Cyber-hacking (e.g., OpenAI attacking HuggingFace)
 - Virus creation (AI built DNA for virus to attack e. coli)
 - Deep-fakes (November election, Hello!)
 - Psychological harms/dependency (remember social media?)
- Worst of all, super-human intelligence is the goal of every AI company, which experts say carries a **15% chance of human extinction**
 - Recursive self-improvement starting to happen
 - They have little control over what these models will do
- Joined **PauseAI-US** non-profit in January
 - SF group coordinator as of August, after leadership moved to DC

PauseAI-US.org

Contacting Congress in 2 minutes:

Step 1: Scan the QR codes on this slide

Step 2: Email your representatives

- You can use the template, or write your own

Step 3: Call your representatives

- You can use the talking points listed, or your own

AI Risk Evaluation Act:

pauseai-us.org/RiskEvalAct



Other AI Safety Bills:

www.pauseai-us.org/takeaction/



Questions?

...