Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Canned Sunshine: Precomputed Daylight Coefficients for Climate Based Modelling using Photon Mapping

Roland Schregle
RS SciComp
Switzerland

roland.schregle@gmail.com

Lars Oliver Grobe, Stephen Wasilewski
RG Envelopes and Solar Energy
Lucerne University of Applied Arts and Sciences,
Switzerland

larsoliver.grobe@hslu.ch
stephen.wasilewski@hslu.ch

2022 International RADIANCE Workshop
3-5 August, 2022
Toronto, Canada

FH Zentralschweiz

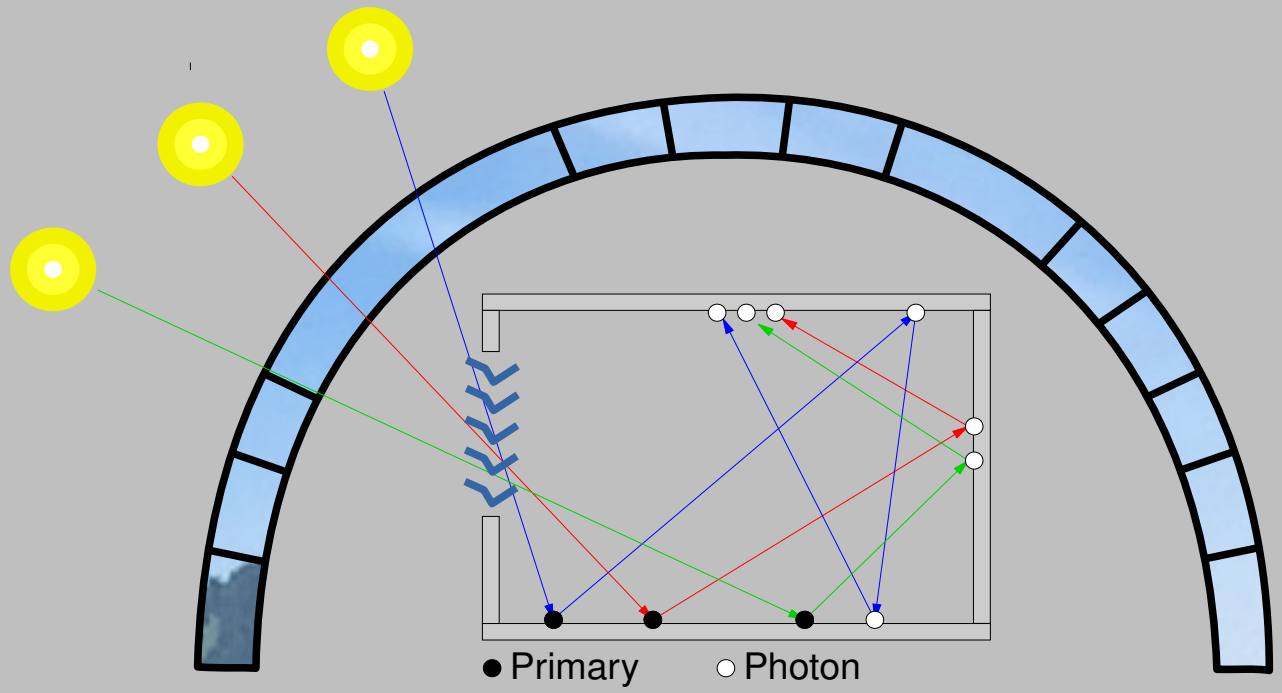
EFQM
Recognised for excellence
4 star

Overview: Why Precompute Contributions?

- ***rcontrib*** calculates geometric transfer terms (coefficients/contributions) between sky and sun positions for climate based daylight modelling.
- No irradiance cache, potentially redundant ray evaluations.
- Slow with shading/redirecting components, data-based BSDFs

Overview: Contribution Photon Map (2015)

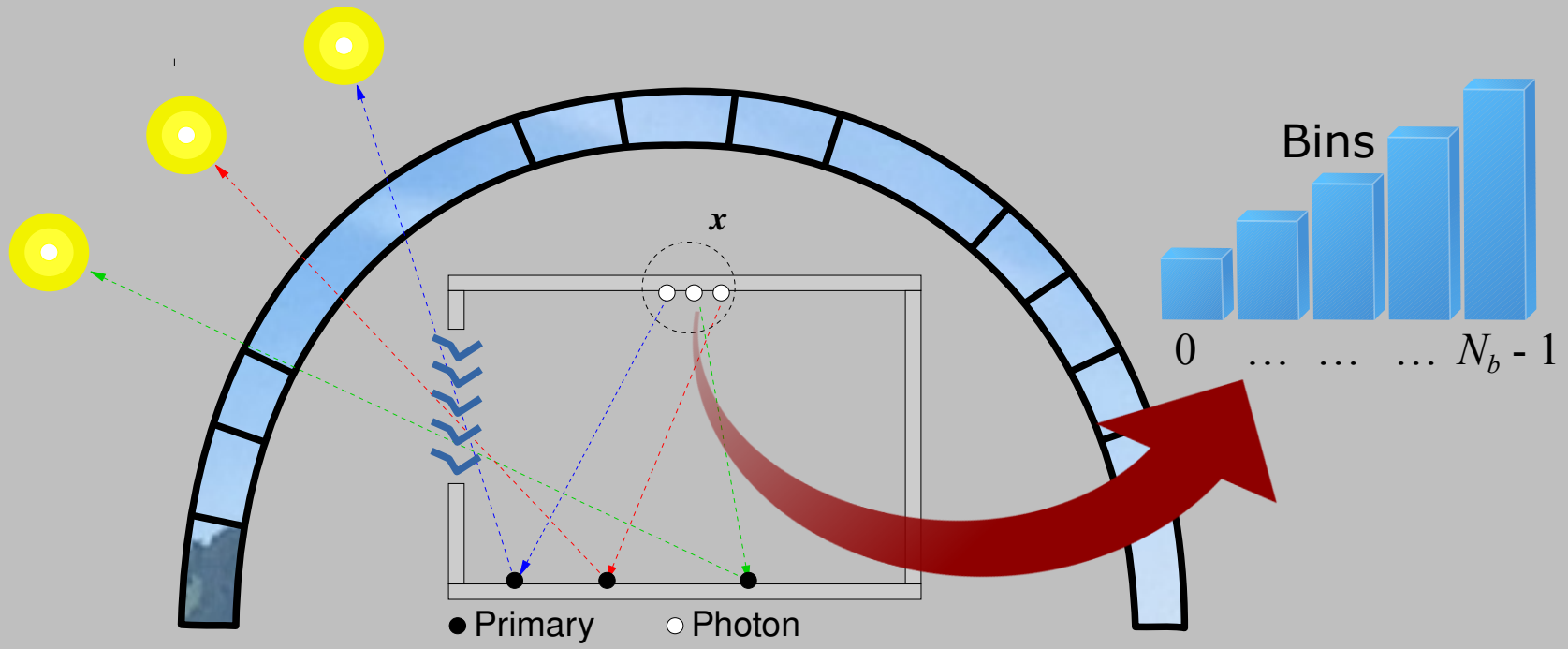
- ***mkpmap*** emits photons from light sources, stores:
  - Flux Φ_p ([W] or [lm], normalised in coefficient mode)
  - Source idx
  - Primary incident direction ψ_p (optional, increases mem footprint)



Overview: Contribution Photon Map (2015)

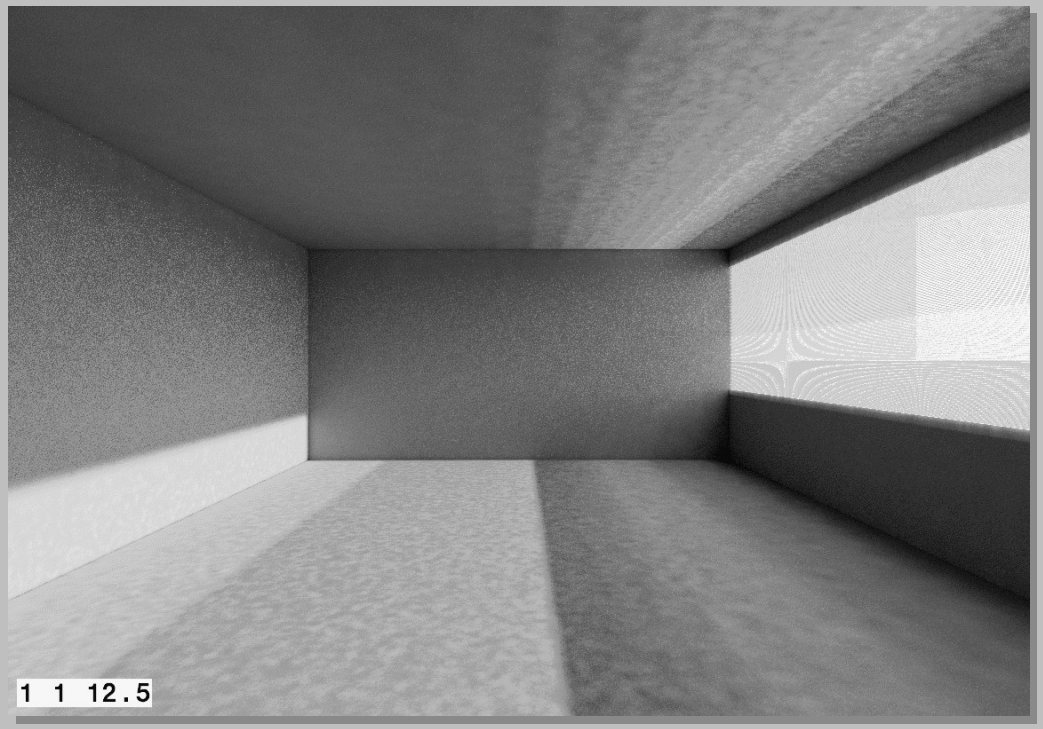
- ***rcontrib*** locates N_p photons around sensor pos, accumulates flux Φ_p in N_b bins acc. to primary dir ψ_p
- Contrib $E(\omega_i)$ for bin $i \in [0, N_b - 1]$ proportional to photon density:

$$E\left(\omega_i\right) \approx \sum_{p=1}^{N_p} \frac{\Phi_p}{\pi r^2} \quad \left\{\forall\ p : \psi_p \in \omega_i\right\},\ N_p \gg N_b\,(!)$$



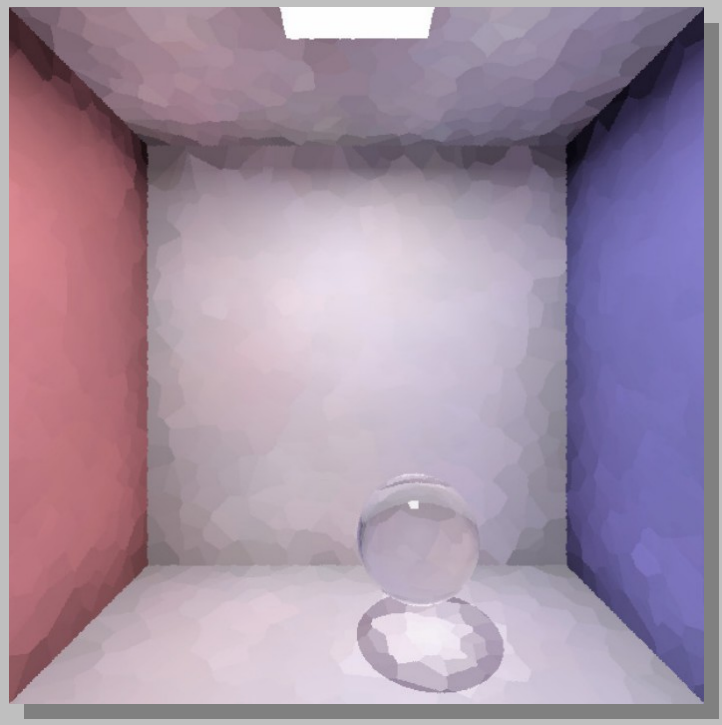
Overview: Contribution Photon Map (2015)

- **Pros (vs. *rcontrib classic*)**: More efficient with data-based BSDFs; binning flexible, as done in ***rcontrib***
- **Cons:** marginally faster, redundant lookups (very large N_p), needs *huge* photon maps (≥ 1G photons) → out-of-core, page from disk



Precomputed Global Photons (~2001)

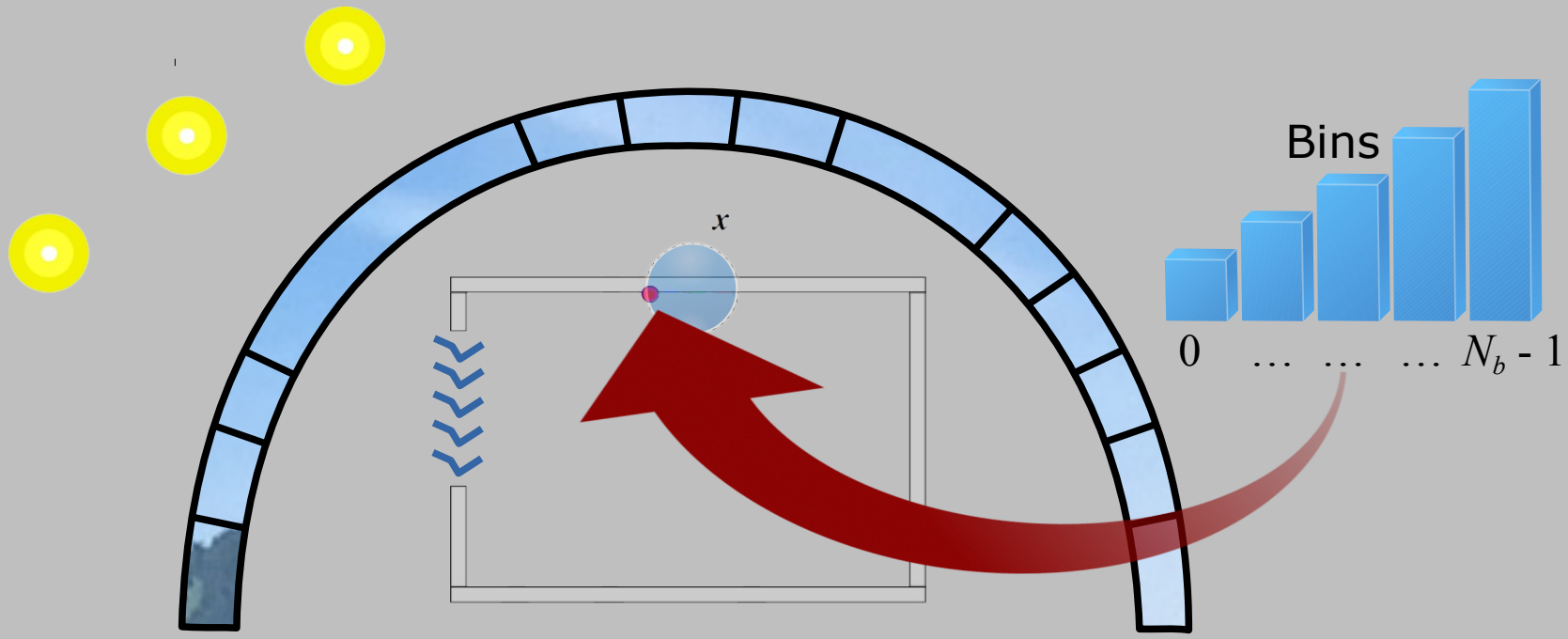
- ***mkpmap*** selects subset of photons for precomputation
- Locates N_p nearest photons at precomp. photon pos, evaluates irradiance E from photon density, stores with precomp photon
- ***rtrace*/*rtrace*/*rvu*** looks up single closest photon to sensor
  → Approximate, but fast
  (fewer redundant photon lookups)
- Idea: apply same principle to contribution photon map
  → Store *vector* of binned contribs per precomputed photon



Precomp. photons directly visualised with -ab -1

Precomputed Contribution Photons: Concept

- Precompute contributions, store for selected photons in ***mkpmap***
- Locate *single closest* precomputed photon in ***rcontrib***, accumulate contributions for each modifier
- But… we now store a *vector* of contributions per photon!

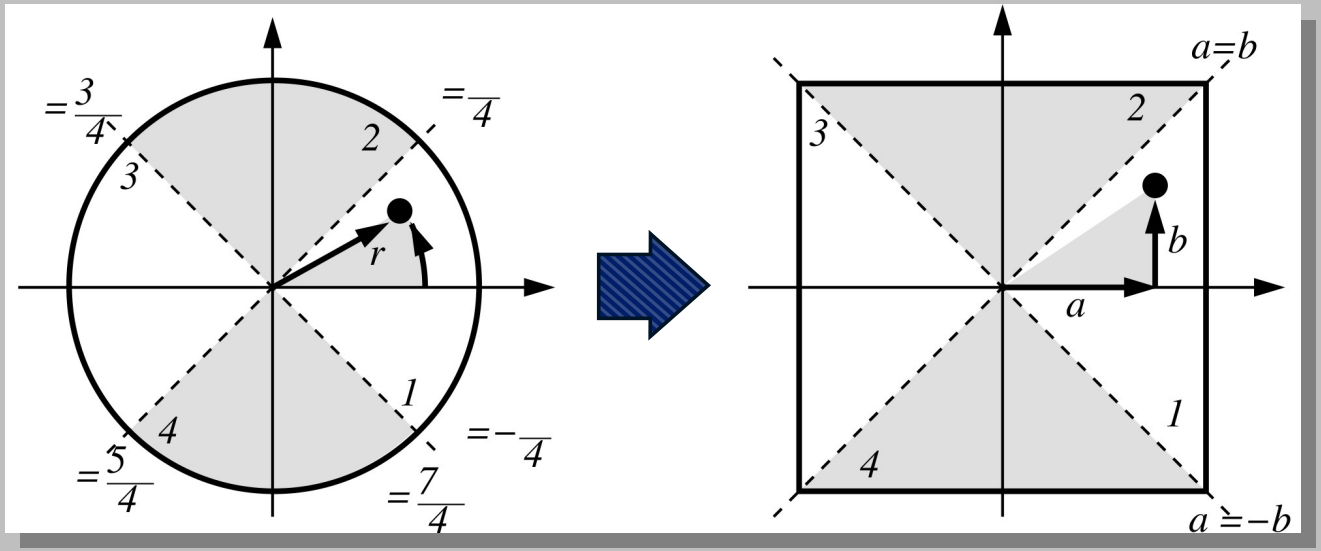


Precomputed Contribution Photons: Data Volume

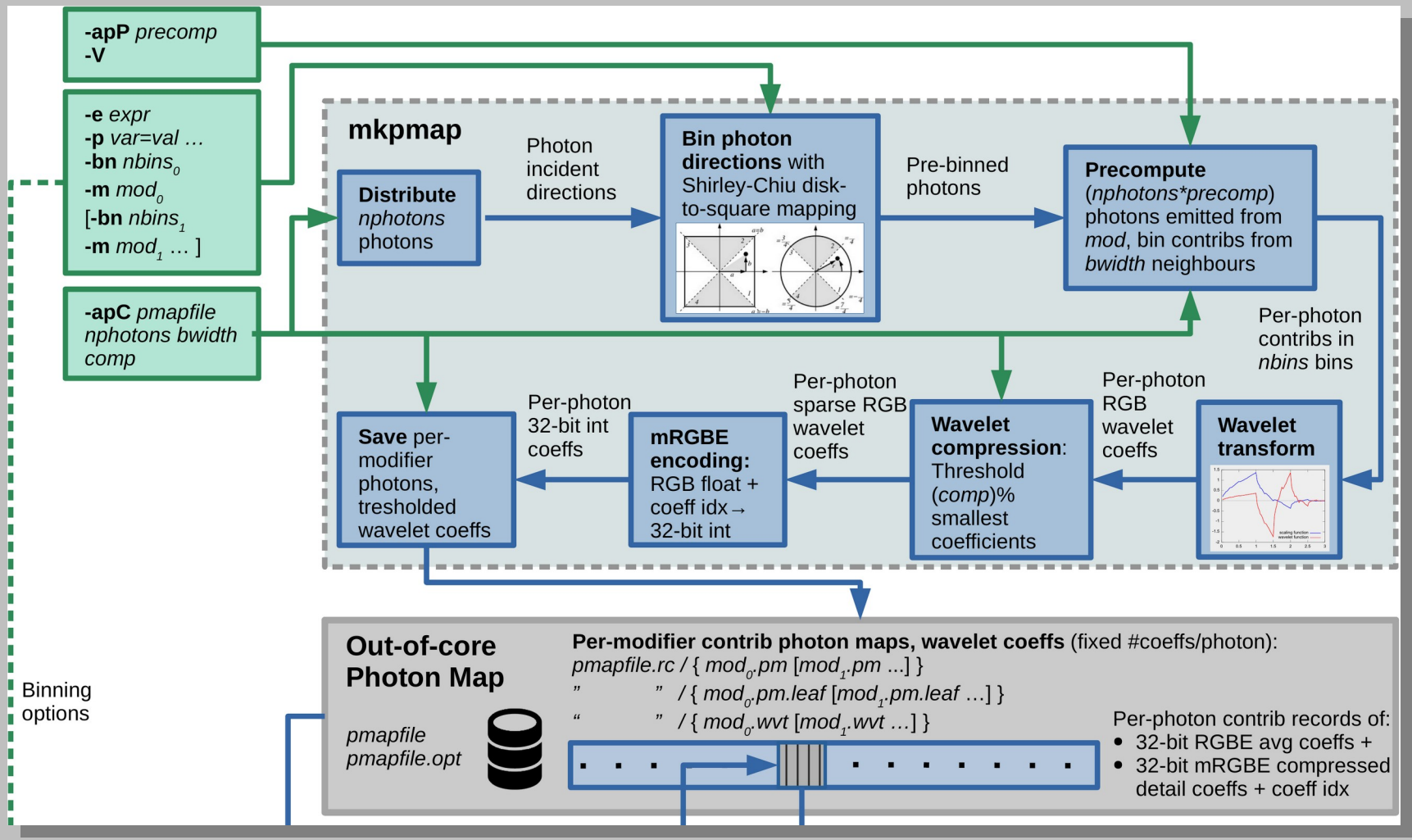
- Need to store binned contribs *per photon* → HUGE data volume
- Example: 10M precomp. photons, Reinhart MF:4 (2305 bins), 32-bit RGBE encoding → 92.2 Gb contributions on disk!
- Need powerful (lossy) compression → wavelets
- Page contribs from disk in ***rcontrib*** → only out-of-core supported
- Cache photons *and* decoded contribs → hide latency of I/O, inv. wavelet transform
- Compact mRGBE encoding for wavelet coeffs (modified, mini…)

Precomputed Contribution Photons: Binning

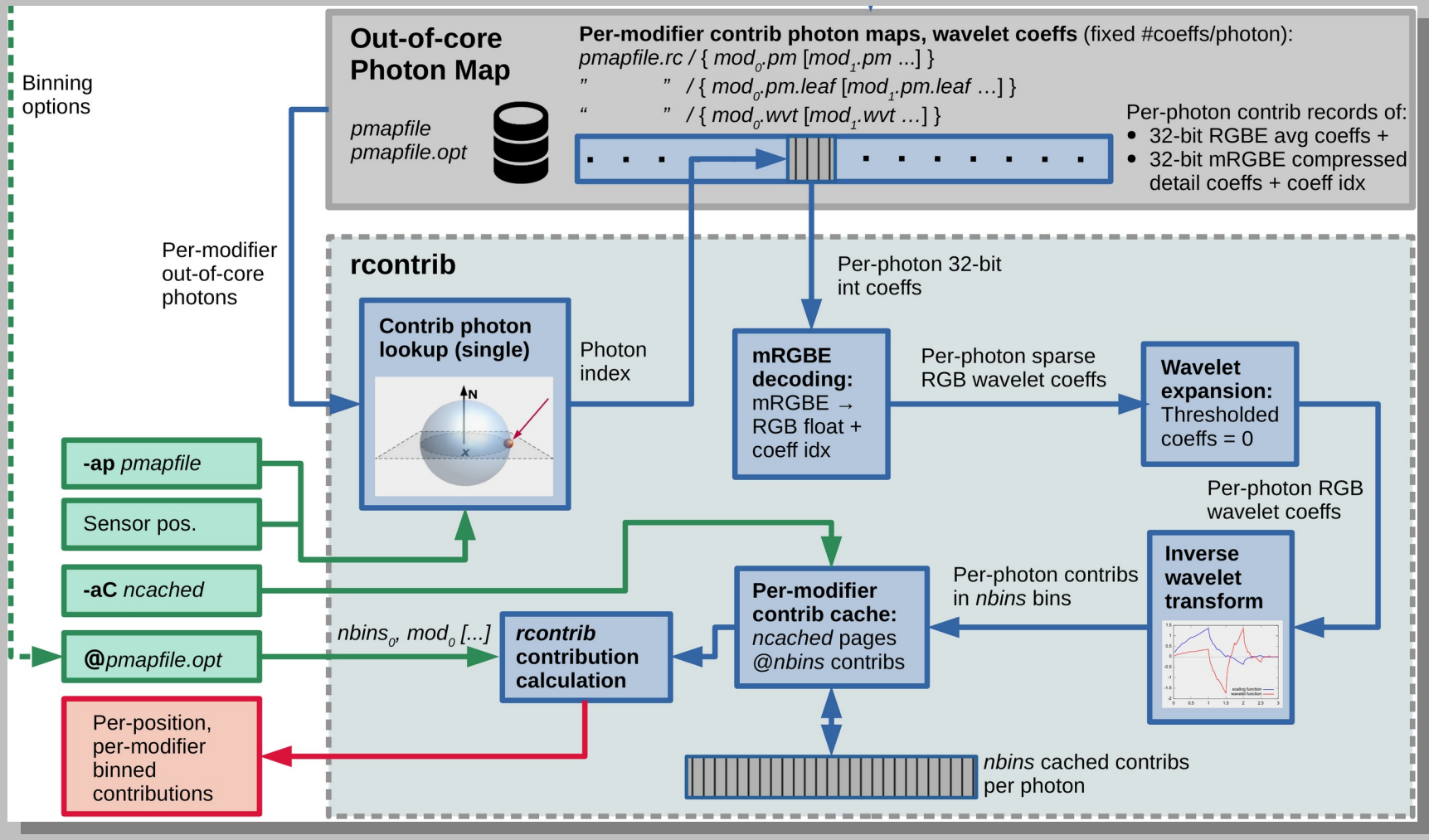
- ***mkpmap*** bins photons using Shirley-Chiu disk-to-square mapping → 2D matrix ideal domain for wavelet transform
- **Pros:** Photon lookups for large N_p only in precomp
- **Cons:** Binning now "frozen" in photon map
- Binning params passed to ***rcontrib*** via *@optionFile* for consistency



Precomputed Contribution Photons: Overview (*mkpmap*)

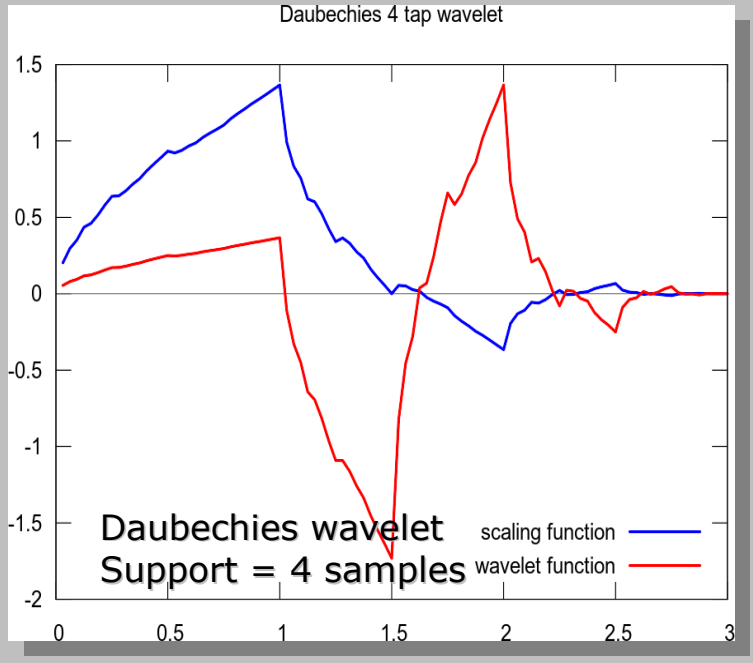


Precomputed Contribution Photons: Overview (*rcontrib*)



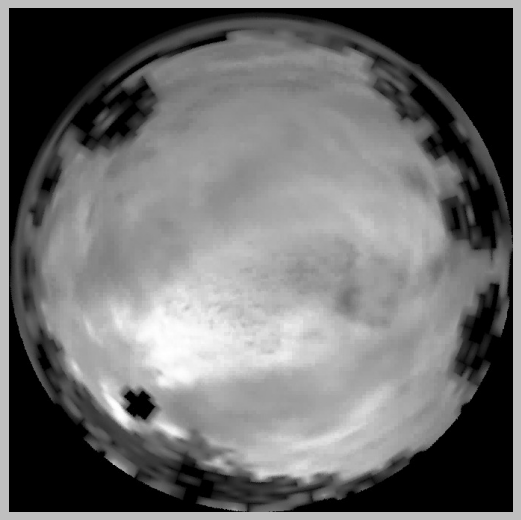
Wavelet Transform

- Applications:
  - Wavelet Radiosity [Gortler, 1993]
  - Sph. Wavelets, BSDFs [Schröder 1995, Lalonde 1997, Wu 2019]
  - Image processing, compression [JPEG2000]
  - Fingerprint identification [AFIS, 2010]
- 2D Wavelet transform over Shirley-Chiu square (=matrix)
- Decomposes contribs into:
  - *approximation* coeffs → low freq,
  - *detail* coeffs → high freq
- Multiple frequency bands
  → *multiresolution analysis*
- Finite support → no ringing artefacts, peaks preserved
- Headaches guaranteed! 

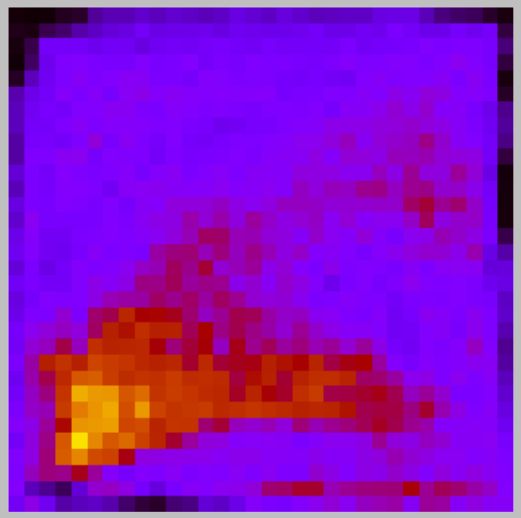


Wavelet Transform

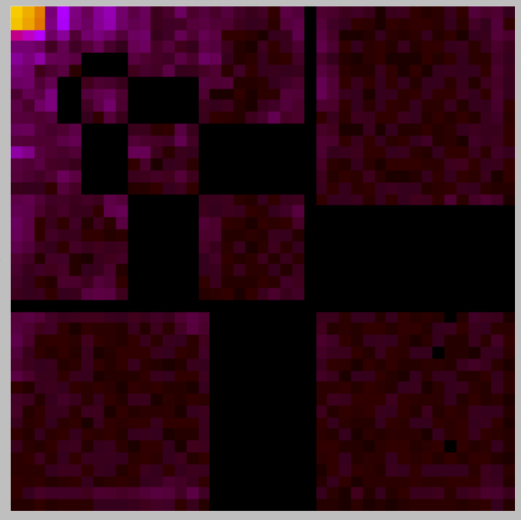
- Alternating horiz. / vert. passes → Decomposition over both axes
- RGB transformed independently → 3-tuple coefficients
- Matrix dimensions $l \times l = N_b$, where $l > 3$ arbitrary (vs. powers of 2)
  → Need boundary extension → extra *padding* coefficients



HDR sky capture



Shirley-Chiu binning

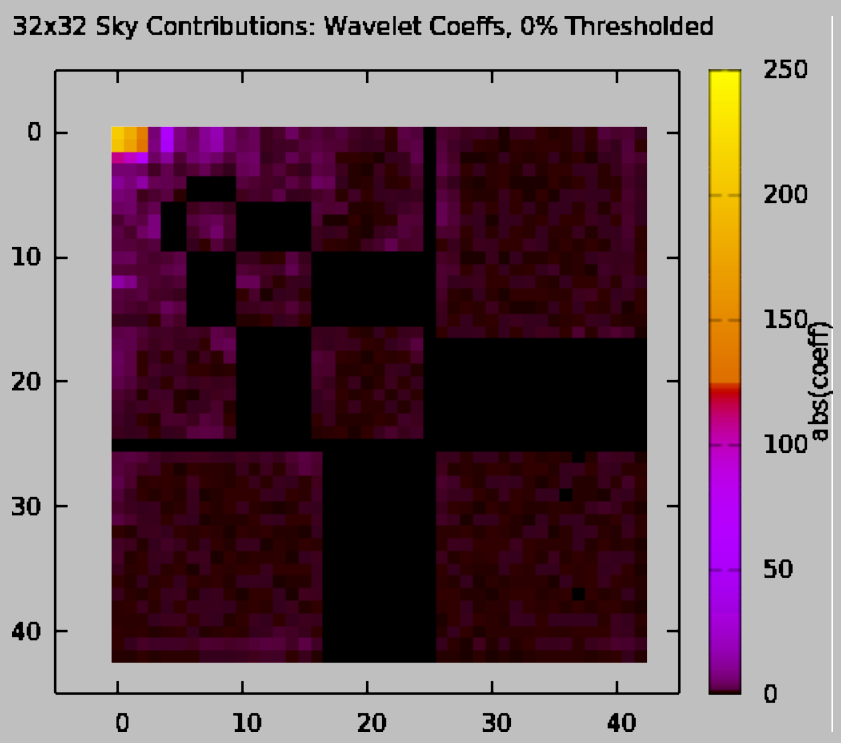


Wavelet coefficients

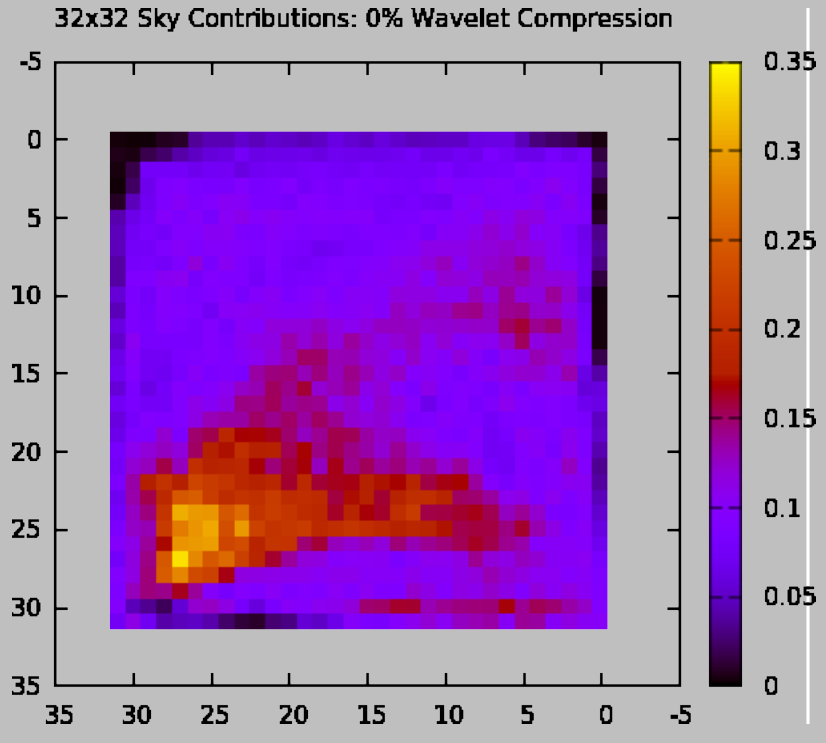
Approx Detail

Wavelet Compression

- Threshold the (*comp*)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



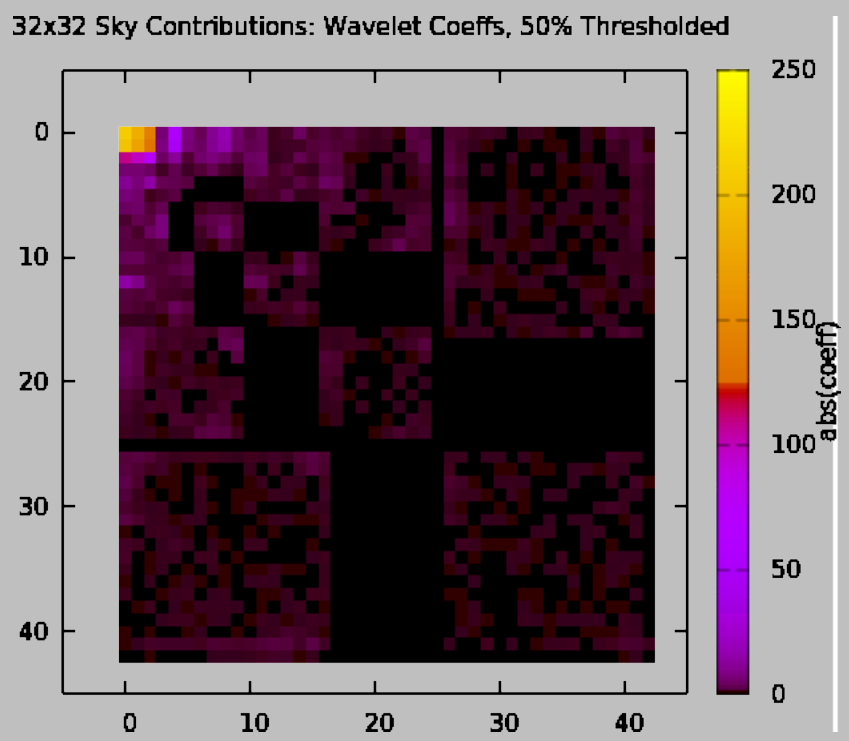
0% thresholded



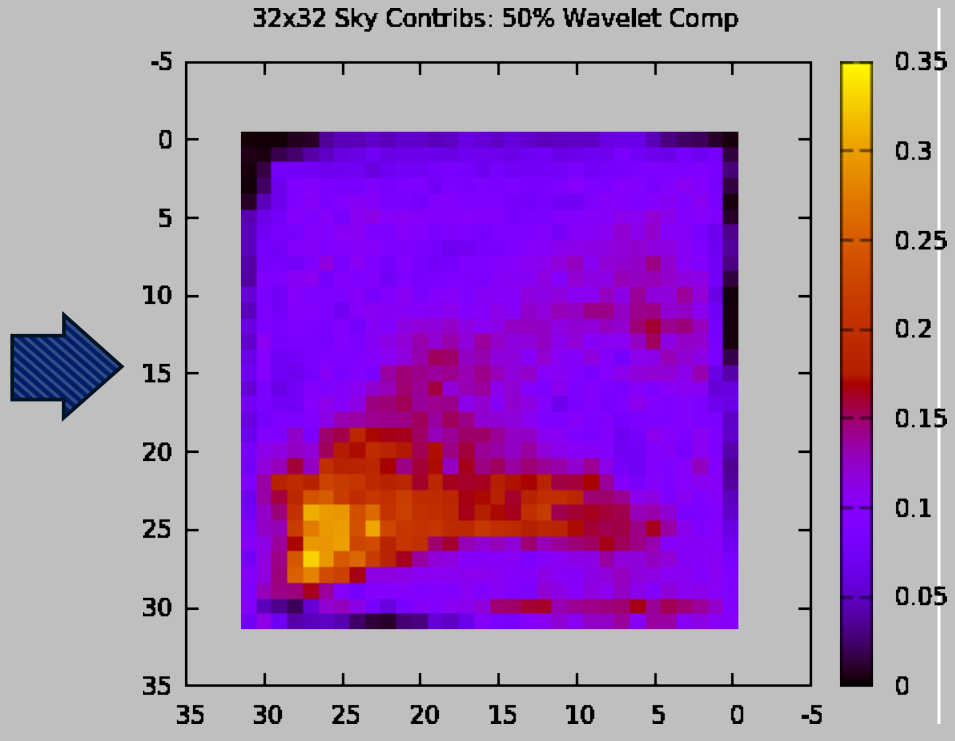
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



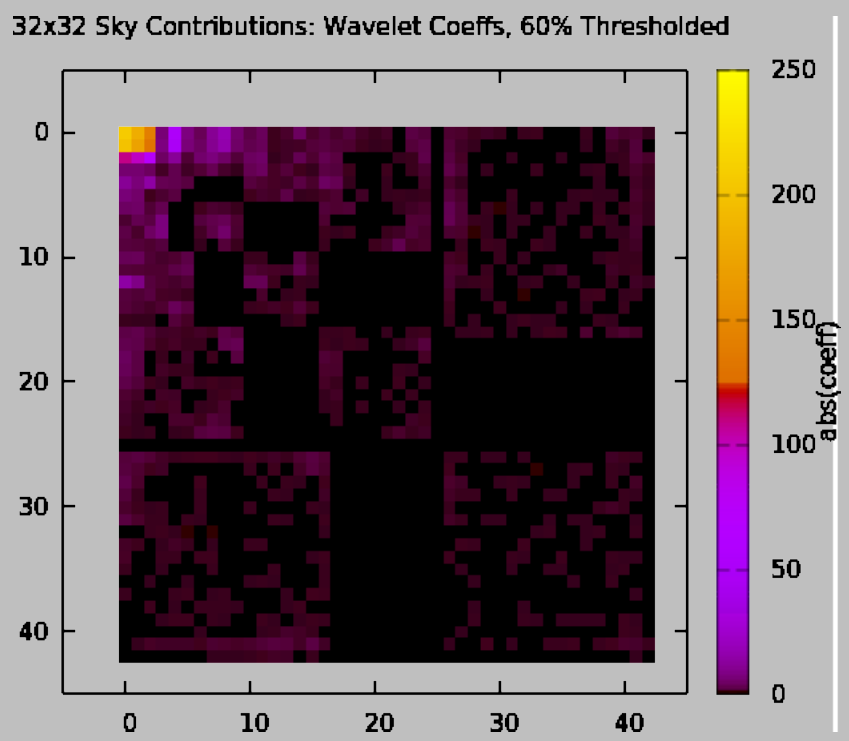
50% thresholded



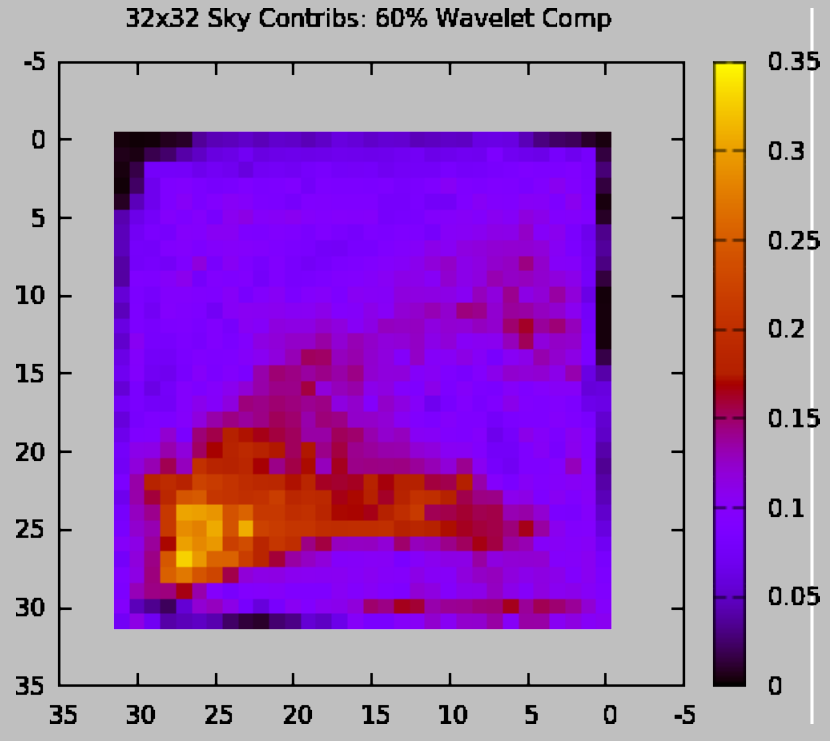
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



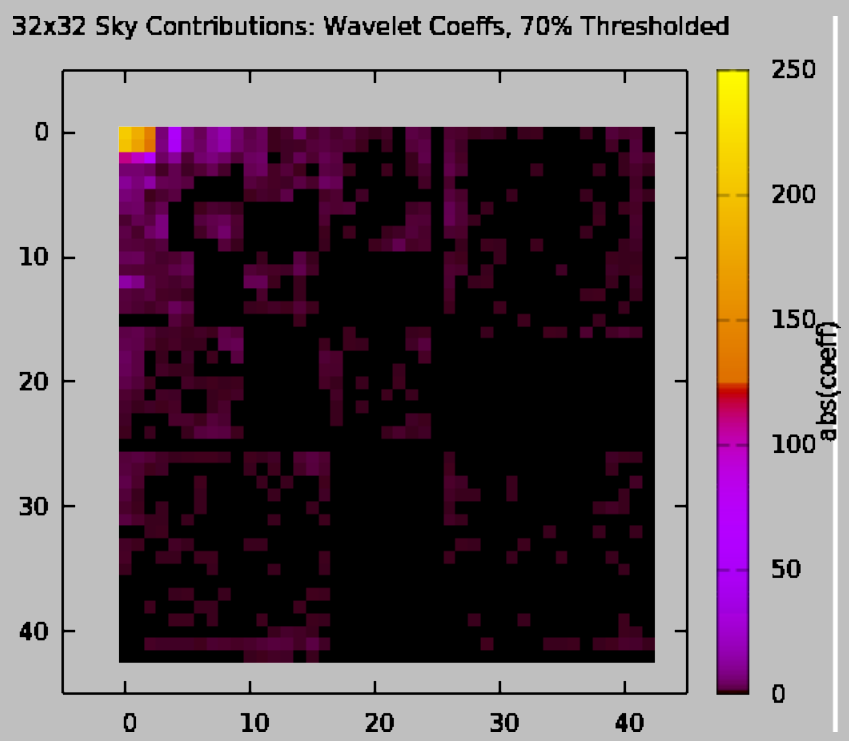
60% thresholded



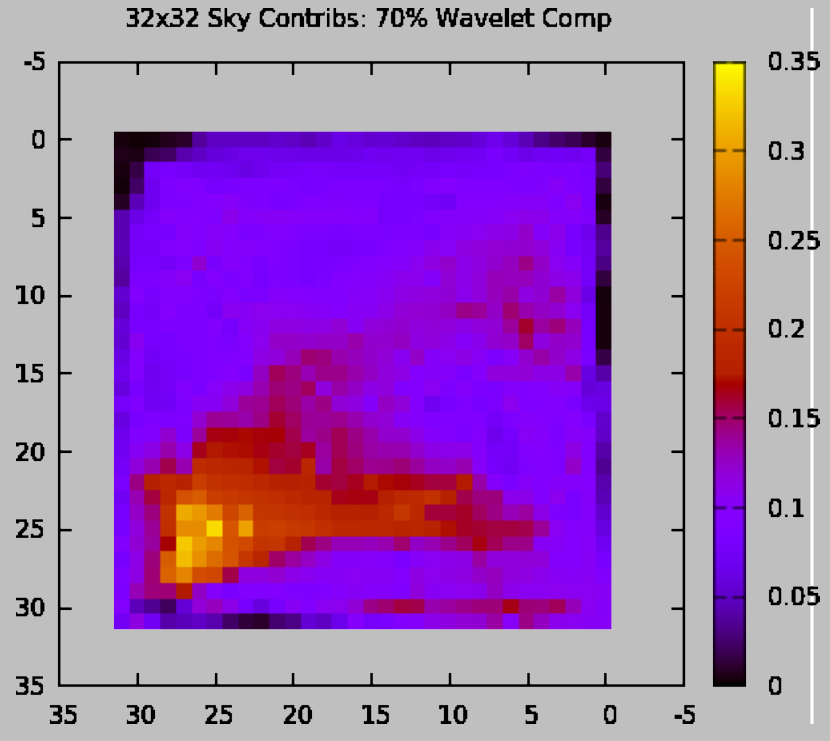
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



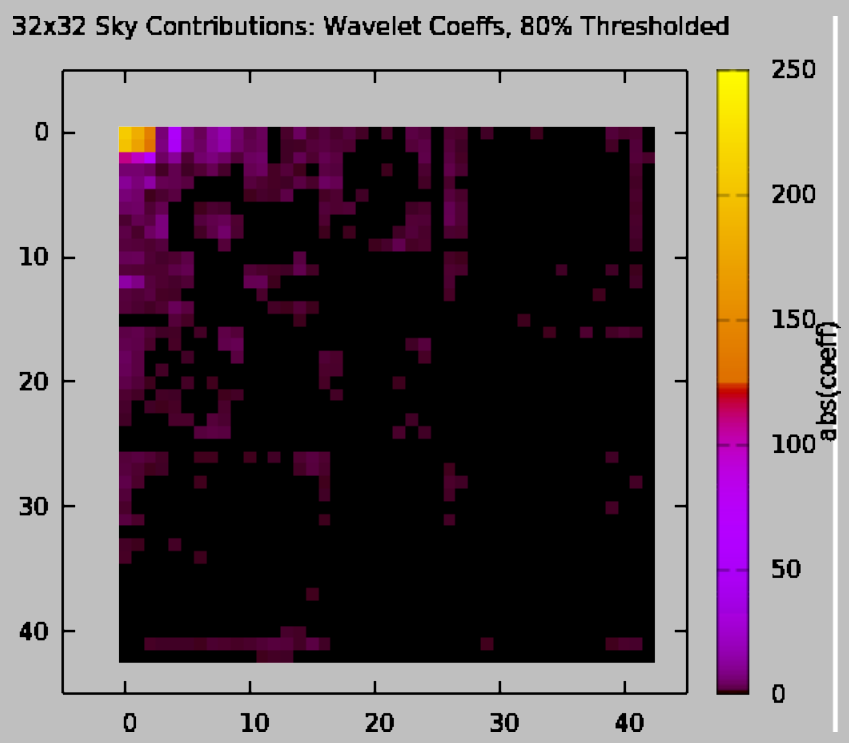
70% thresholded



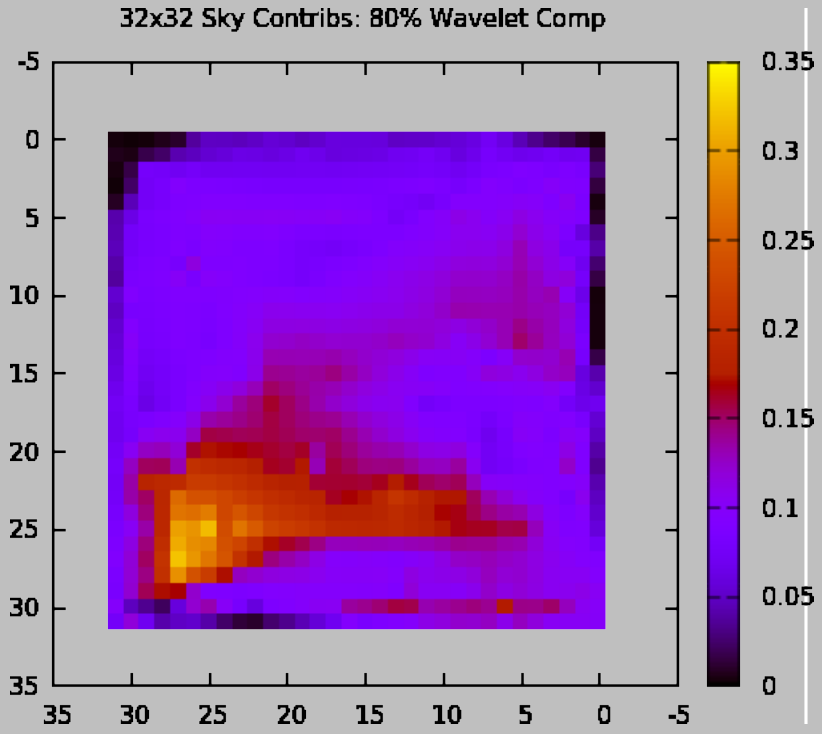
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



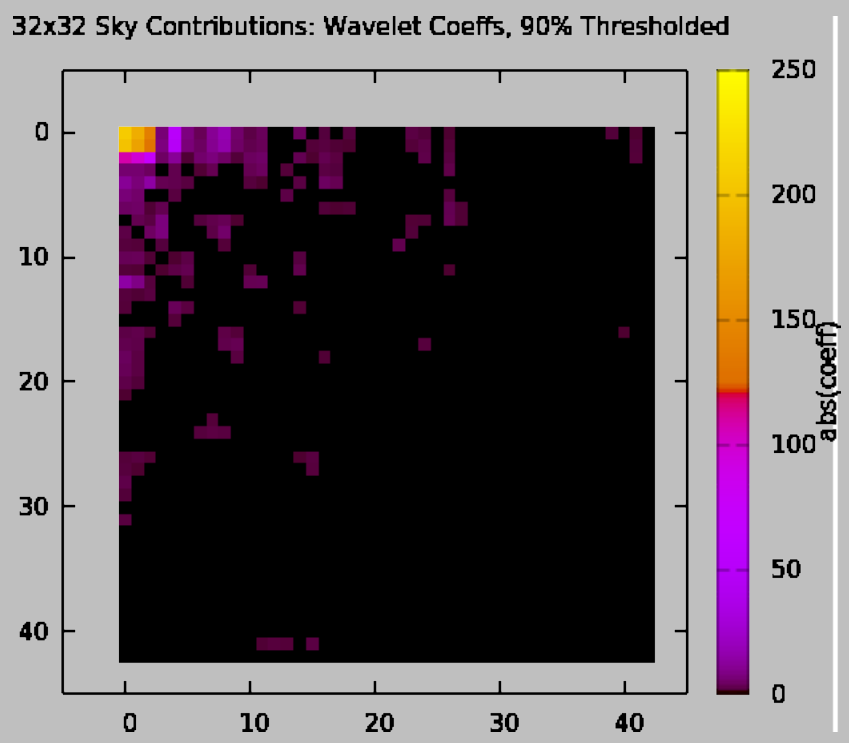
80% thresholded



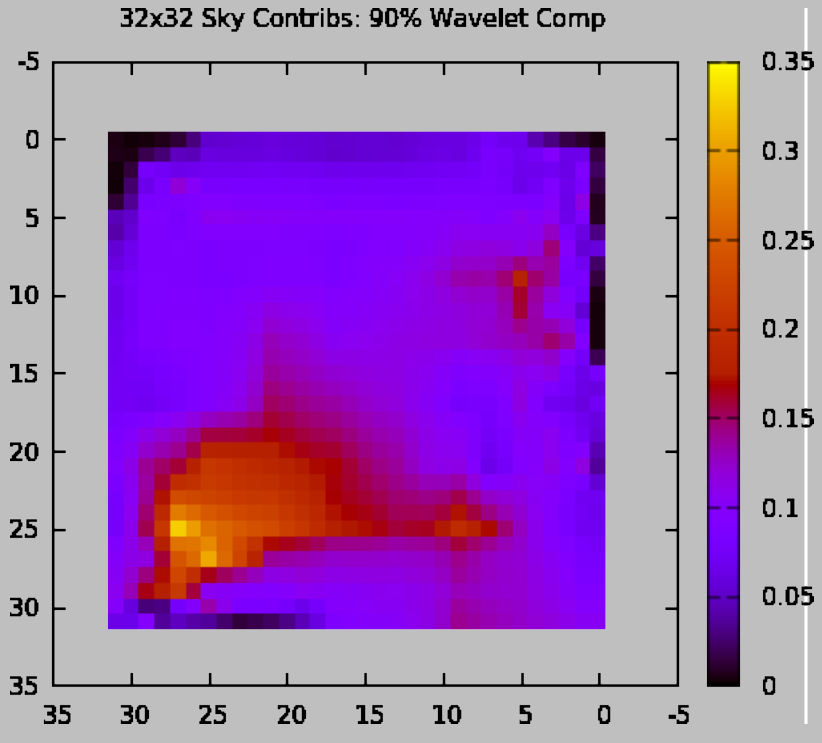
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



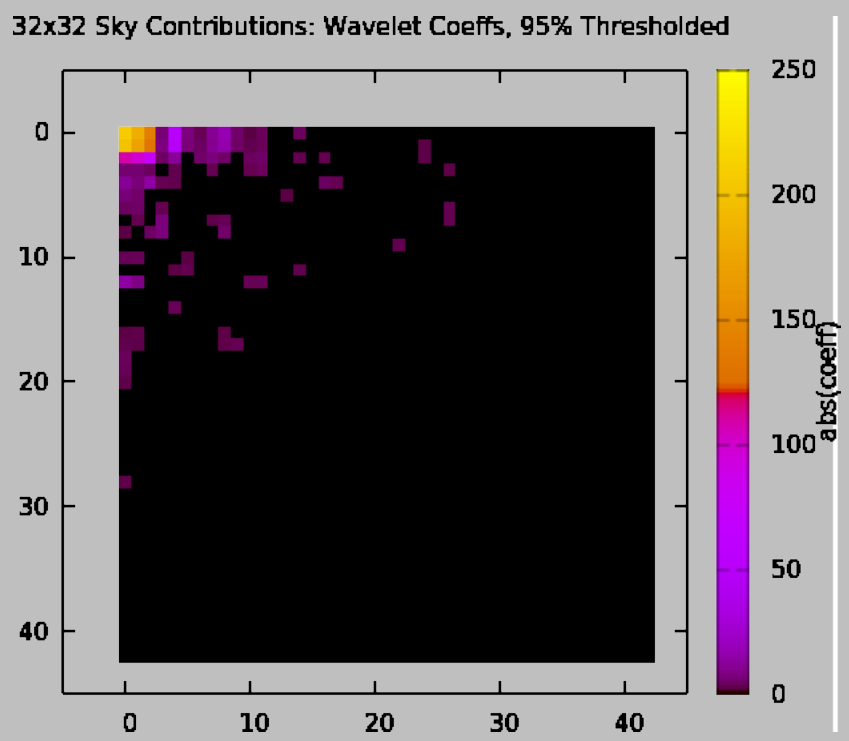
90% thresholded



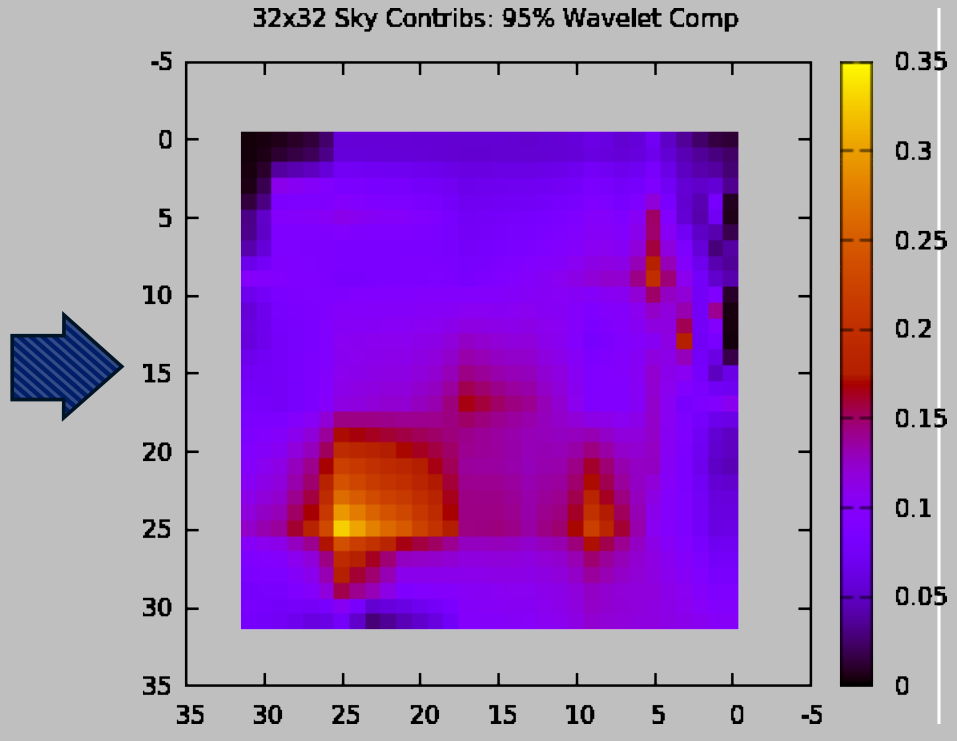
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



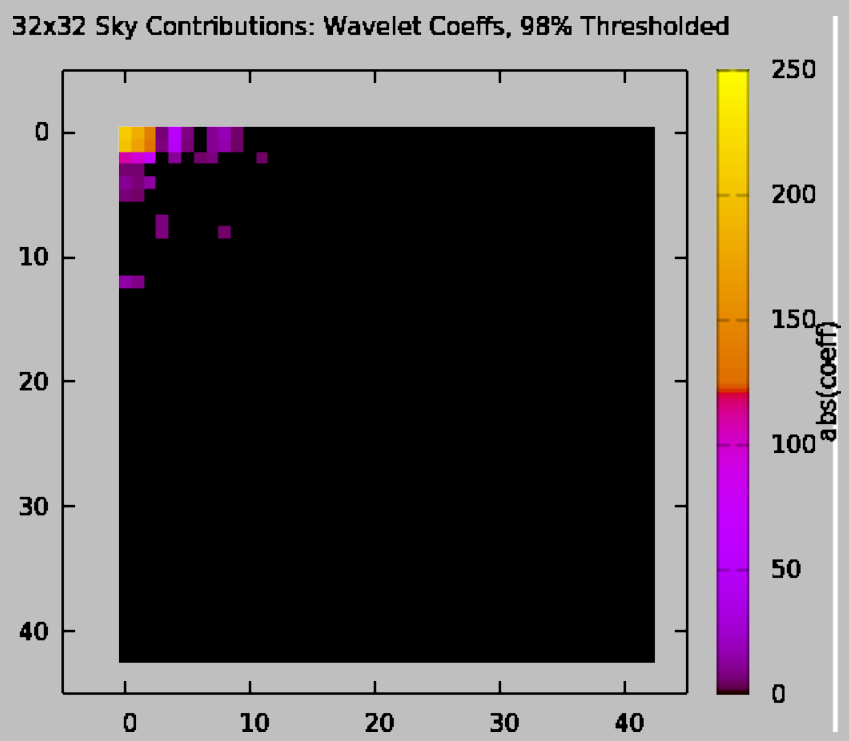
95% thresholded



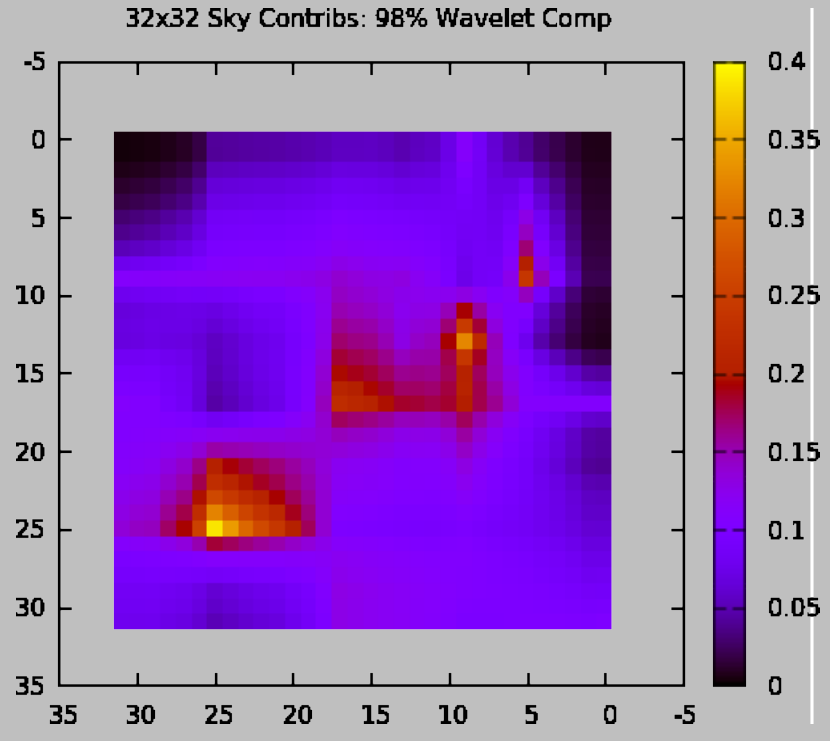
Reconstructed

Wavelet Compression

- Threshold the ($comp$)% smallest *detail* coefficients
  → not stored, set to 0 when reconstructing via inv. wavelet xform



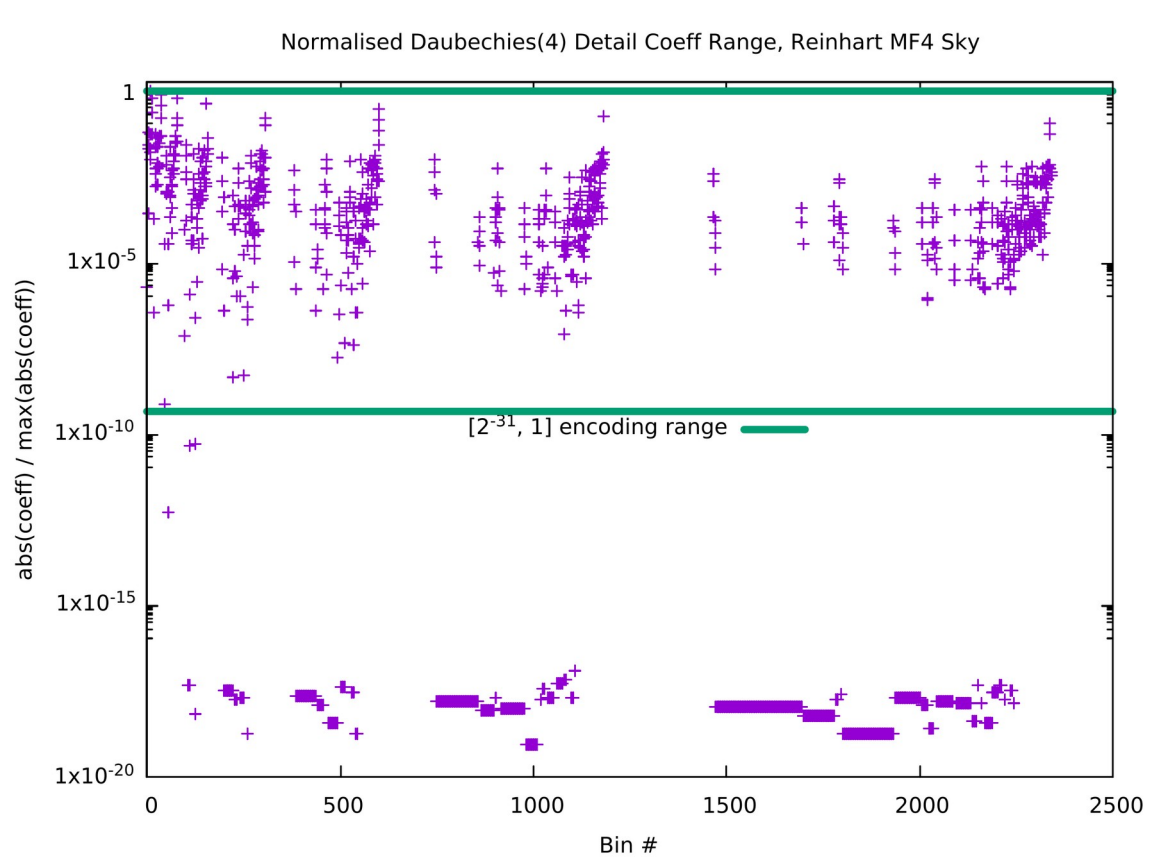
98% thresholded



Reconstructed

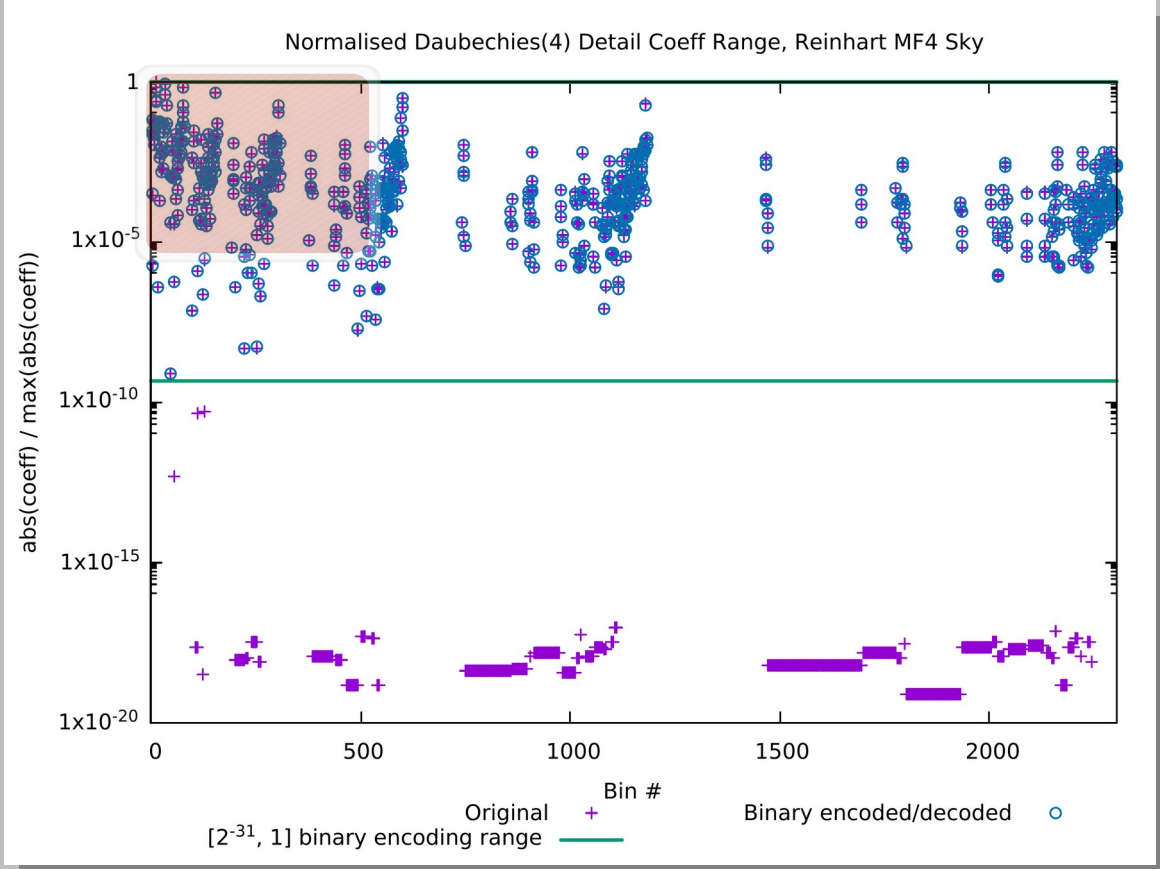
mRGBE Encoding for Wavelet Coefficients

- Wavelet detail coeff range limited → Reduced precision encoding
- Normalise coefficients to maximise encoding range



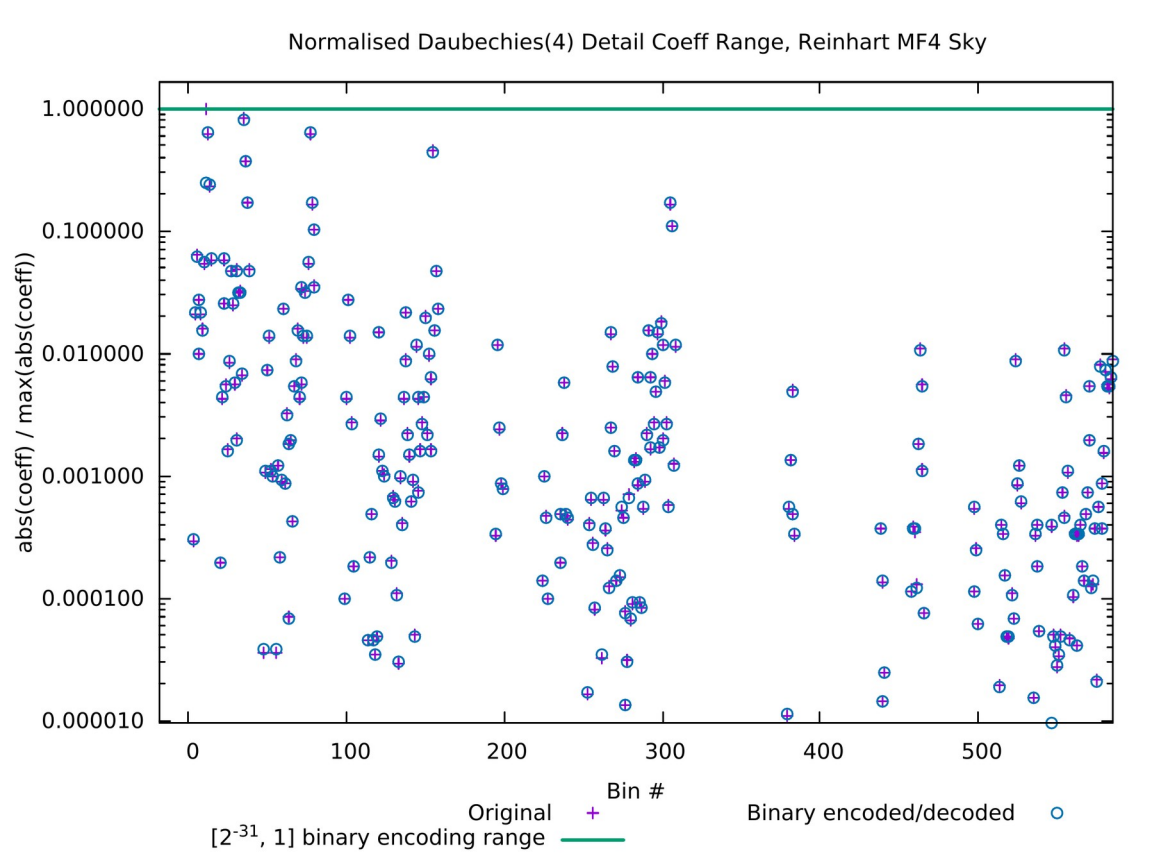
mRGBE Encoding for Wavelet Coefficients

- 5-bit mantissa + 5-bit exponent (base 2)
  → Encoding range $[2^{-31}, 1]$



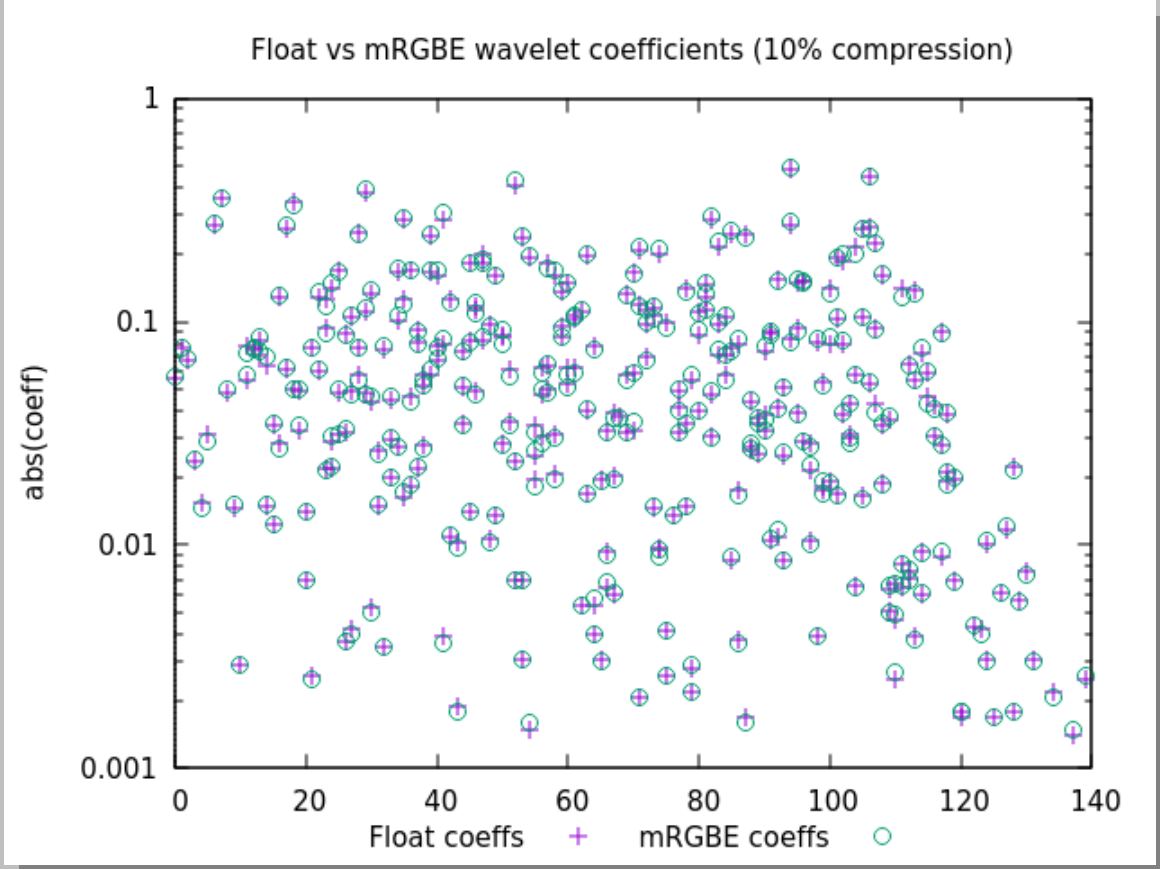
mRGBE Encoding for Wavelet Coefficients

- 5-bit mantissa + 5-bit exponent (base 2)
  → Encoding range $[2^{-31}, 1]$



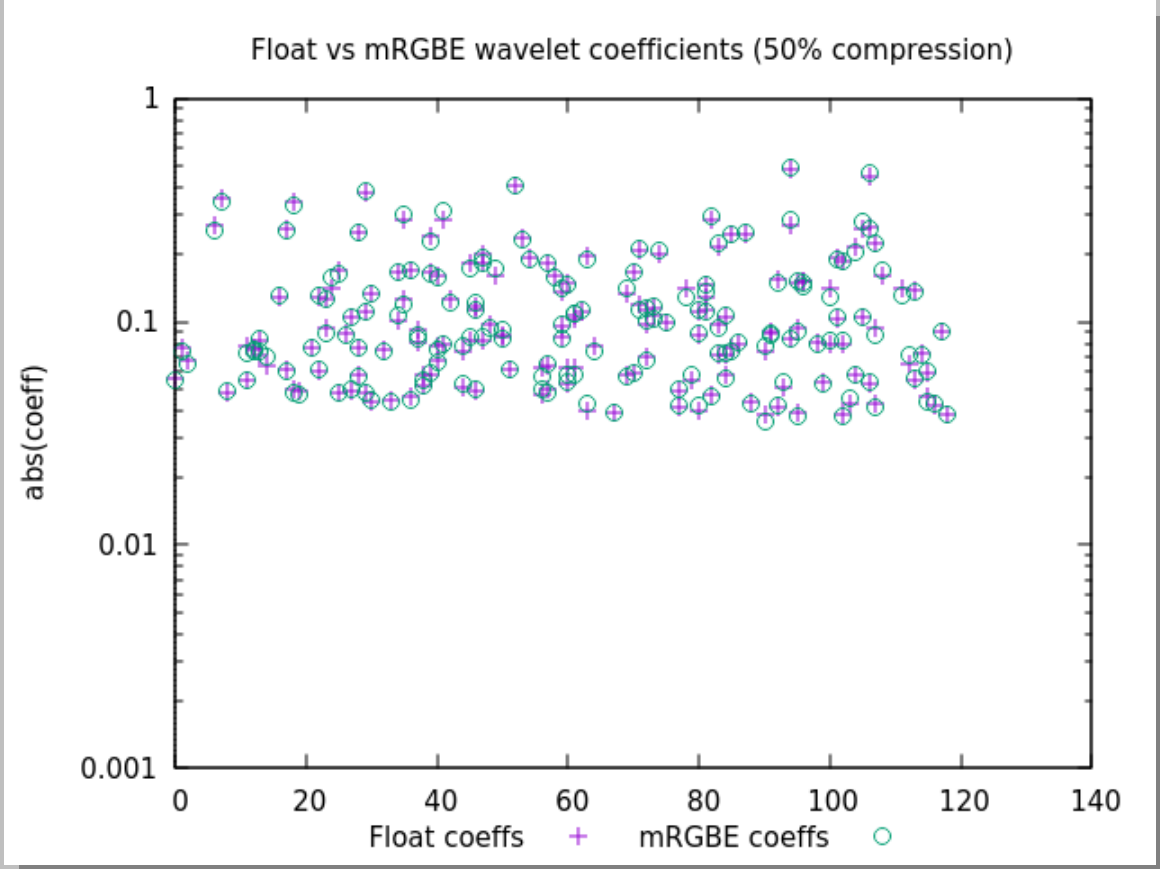
mRGBE Encoding for Wavelet Coefficients

- Offsetting by threshold increases encoding precision
- 10% thresholded → Encoded range [0.001, 1]



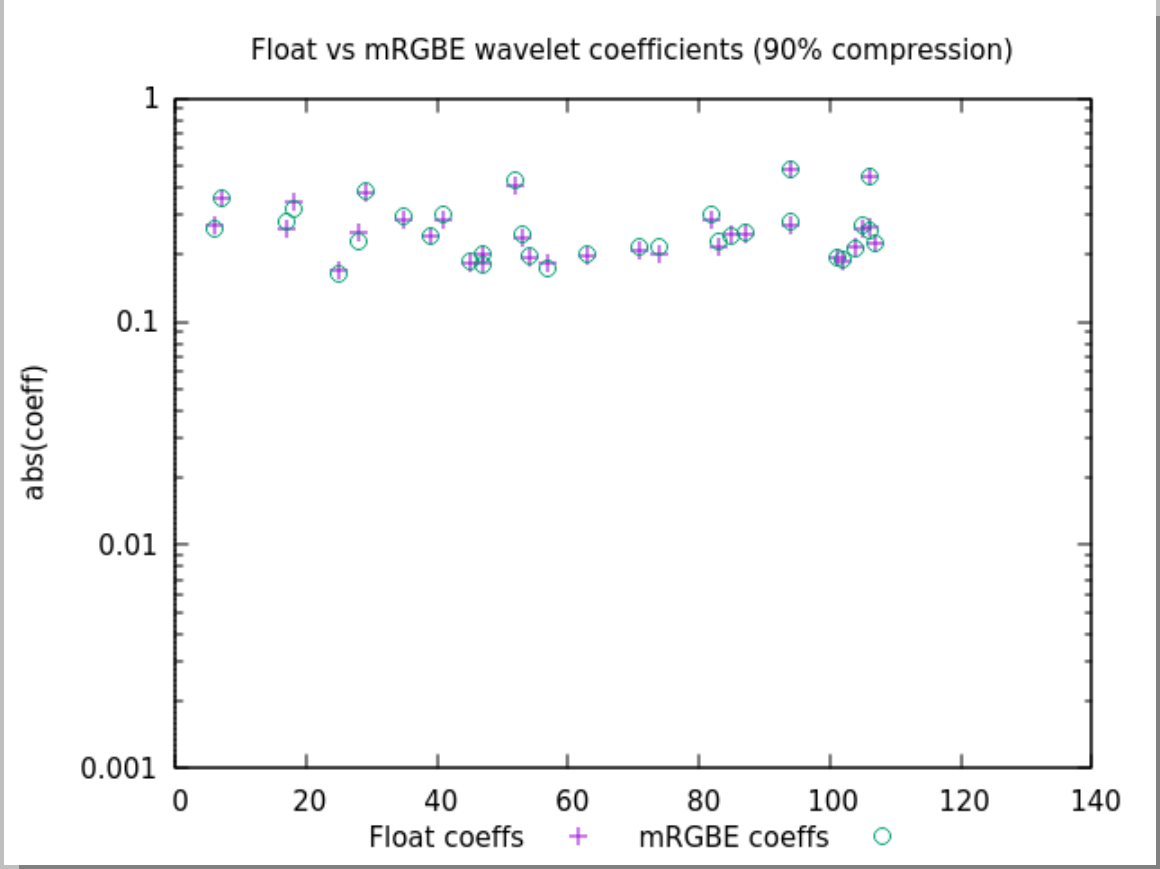
mRGBE Encoding for Wavelet Coefficients

- Offsetting by threshold increases encoding precision
- 50% thresholded → Encoded range [0.04, 1]



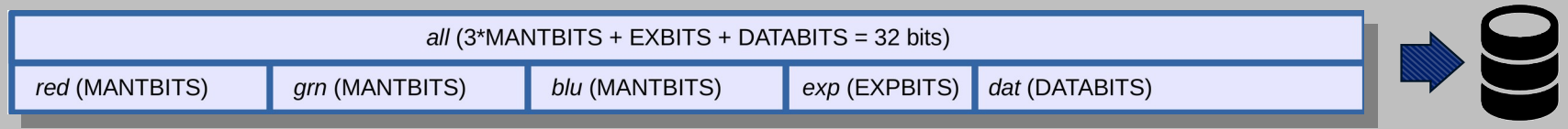
mRGBE Encoding for Wavelet Coefficients

- Offsetting by threshold increases encoding precision
- 90% thresholded → Encoded range [0.12, 1]

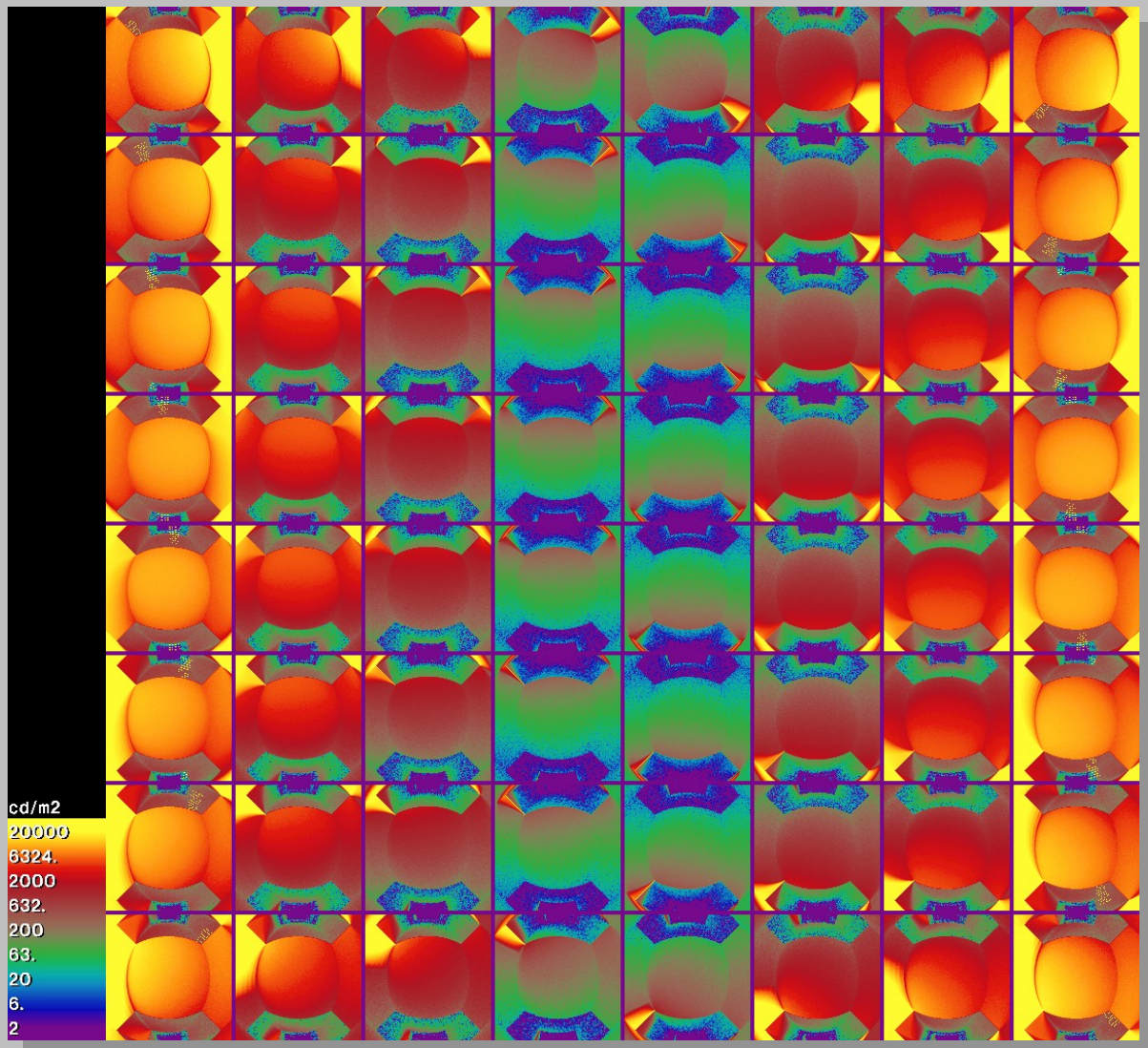


mRGBE Encoding for Wavelet Coefficients

- Compactly encodes thresholded wavelet coeffs on disk
- RGB mantissae + common exponent + wavelet coeff index
- 1D coeff index indicates position in matrix after thresholding
  → *Incremental* encoding minimises overflow!
- Coeffs offset by minimum, normalised to 1 prior to encoding
  → Common normalisation factor stored as std. RGBE
- Default config in 32-bit envelope:
  5 bits / mantissa (incl. 1 sign)
  5 bits exponent (base 2, implicitly negative)
  12 bits incremental coeff index
- Avg deviations ~3% if R:G:B ≤ 10 → Assume low colour saturation



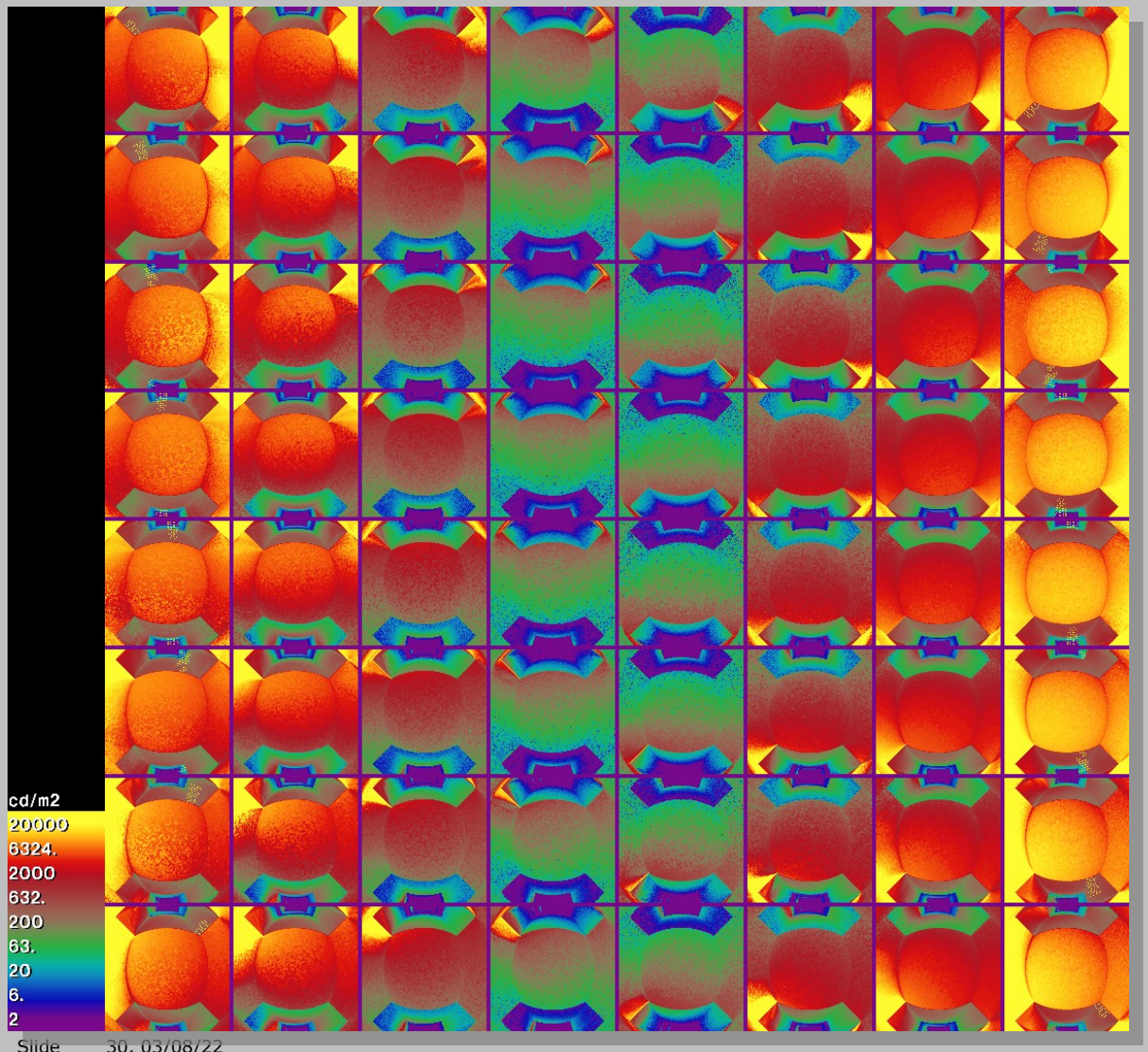
Results: Bilaterally Lit Scene (3970 Suns, 64 Bins)



Whoops…
Binning params not passed to ***rcontrib*** → no suns!

rcontrib classic
-ab 4
~1h @20 cores

Results: Bilaterally Lit Scene (3970 Suns, 64 Bins)

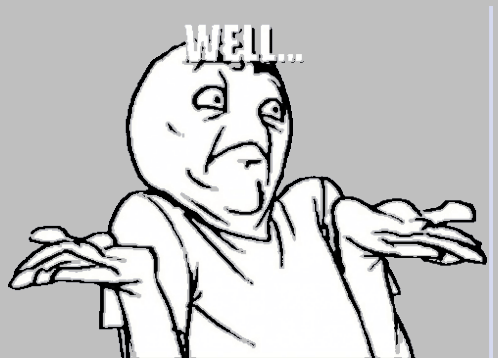


mkpmap
64M contrib photons
64000 precomputed
(N_p = *2400 photons)*
80% compression
~13.5 min @20 cores

rcontrib
-ab -1
~14 sec @20 cores

Conclusions and Outlook

- Precomputation + caching speeds up contribution photon mapping
- Wavelet compression effectively manages data volume
- Technical report available at: http://dx.doi.org/10.13140/RG.2.2.24397.10721/2
- **BUT...** (and this is a big BUTT)
- More testing with "RealWorld" scenes needed
- Improve compression; other wavelets, thresholding strategy?
- Reduce boundary artefacts
- Code release delayed... um, yeah...
- Paper? Hmmm...
- Future work? Uhhh...



Thank you for your attention!

This research was supported by:

Grant #179067 (*Light Fields for Spatio-Temporal Glare Assessment*)