IMCOM: optimized image coaddition pipeline for Roman weak lensing cosmology

Speaker: Kaili Cao (The Ohio State University)

Co-Authors: Christopher M. Hirata (OSU), Katherine Laliotis (OSU), Masaya Yamamoto (Duke → Princeton), Emily Macbeth (OSU), Michael A. Troxel (Duke)

Acknowledgements: Roman HLIS Cosmology PIT, NASA Goddard Space Flight Center, Ohio Supercomputer Center

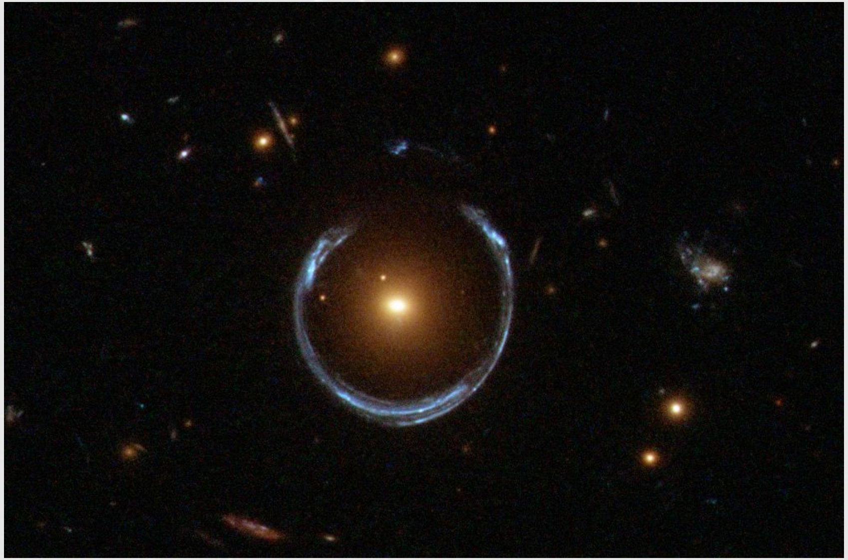
Simulated Image

Outline of this IMCOM (IMage COMbination) overview

- Motivation
  - About gravitational lensing
  - When to use IMCOM?
- Introduction to coaddition
  - Why do we coadd images?
  - How do we coadd images?
- IMCOM vs. Drizzle
  - With vs. without control over output PSF
  - “Stress test” with point sources
- Work in process
  - New implementation: pyimcom
  - New linear algebra strategies
  - Fine-tuning hyperparameters
- Discussion
  - Key takeaways
  - Other ongoing projects

Background: Gravitational lensing and cosmology

- Strong lensing (left)
- Weak lensing (right)



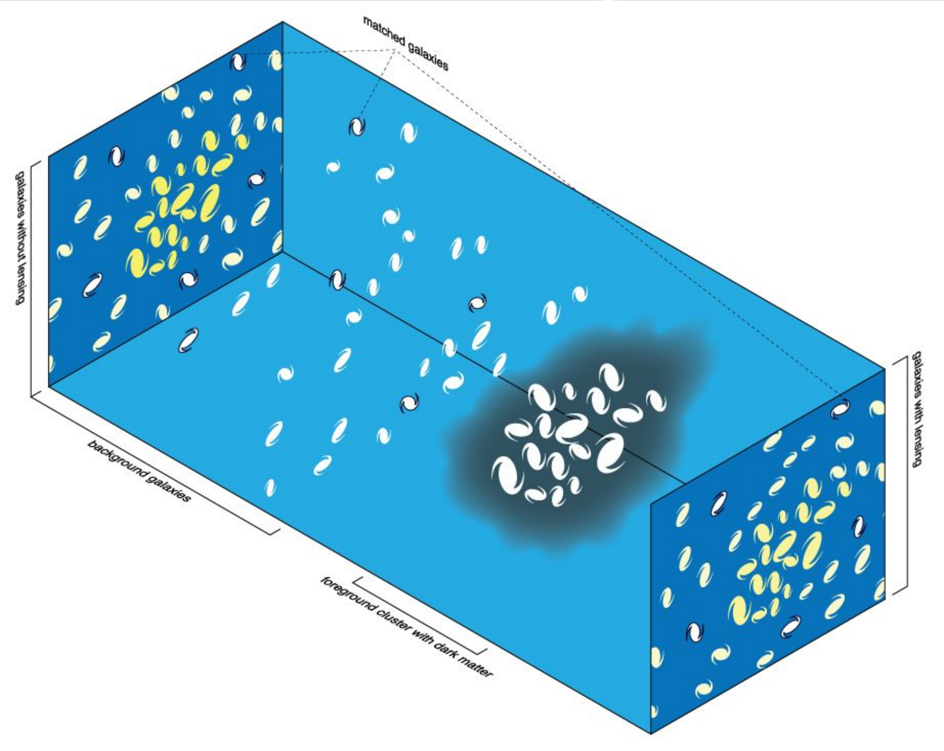


Image credits: ESA/Hubble & NASA, Bulwersator (strong lensing) / Michael Sachs (weak lensing)

When to use IMCOM? – Whenever you need coaddition.

- Weak lensing cosmology
  - A survey design tool for Roman HLWAS
  - A segment of our data analysis pipeline
  - Can be useful for other surveys as well

- Other research areas
  - Whenever you need to coadd images – and can measure PSFs.
  - For example, asteroseismology with Roman GBTDS
    - *Ecclesiastes* 4:9: “Two are better than one,
      because they have a good return for their labor.”

Why do we coadd images? – “Oversample, we must!”

- Roman point spread functions (PSFs)
  - = Airy disk (~λ/D) + linear obscuration + diffraction spikes + detector effects
  - An example in H158 band shown below

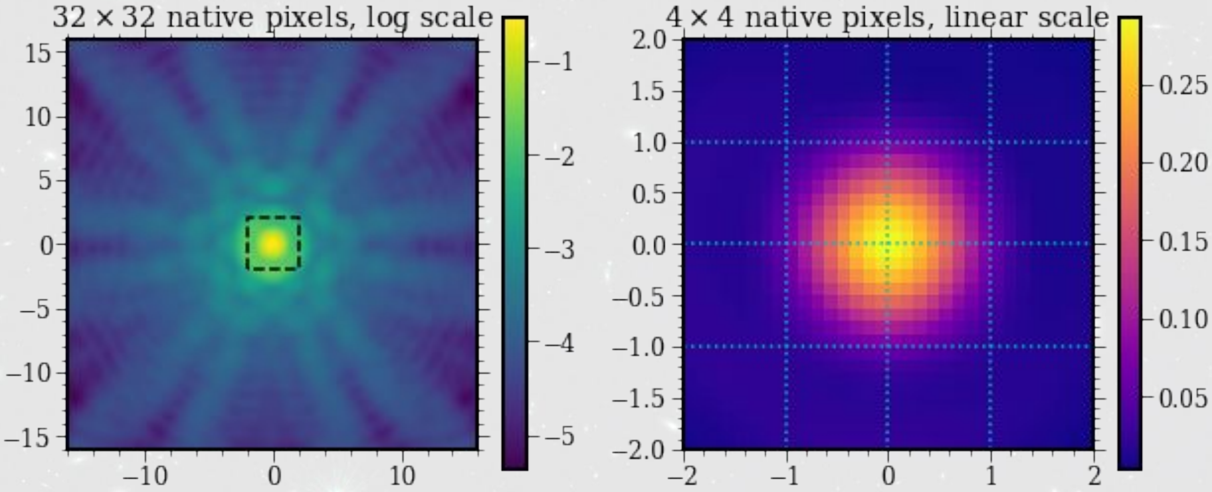


Image credits: Lucasfilm Ltd., Janson_G

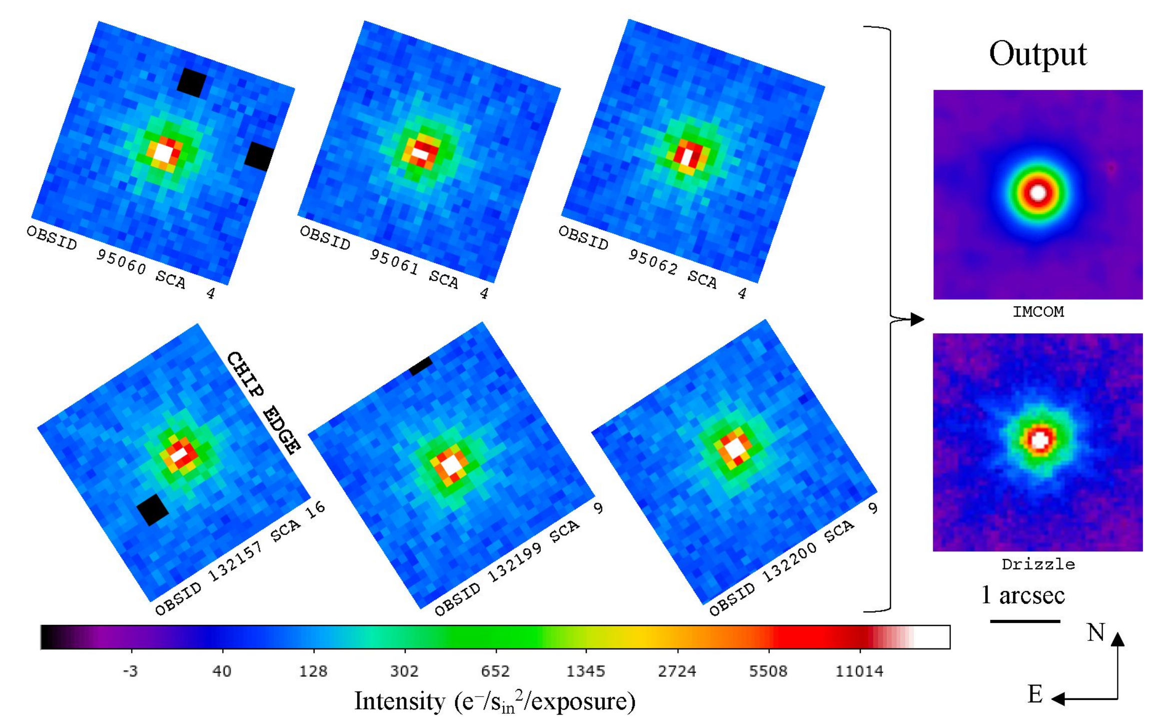
How do we coadd images? – Through linear combinations.

- Linear combination of input **signals**: $H_\alpha = \sum_{i=0}^{n-1} T_{\alpha i} I_i$
  - H : output signal, α : output pixel index, $\boldsymbol{R}_\alpha$: output pixel position
  - I : input signal, i : input pixel index, $\boldsymbol{r}_i$: input pixel position
  - T : coaddition weight, n : number of (selected) input pixels

- Linear combination of input **PSFs**: $\mathrm{PSF}_{\alpha,\mathrm{out}}(\boldsymbol{R}_\alpha - \boldsymbol{s}) = \sum_{i=0}^{n-1} T_{\alpha i} G_i(\boldsymbol{r}_i - \boldsymbol{s})$
  - $\mathrm{PSF}_{\mathrm{out}}$: reconstructed output PSF, Γ : target output PSF
  - IMCOM minimizes discrepancy between $\mathrm{PSF}_{\mathrm{out}}$ and Γ (“leakage”).

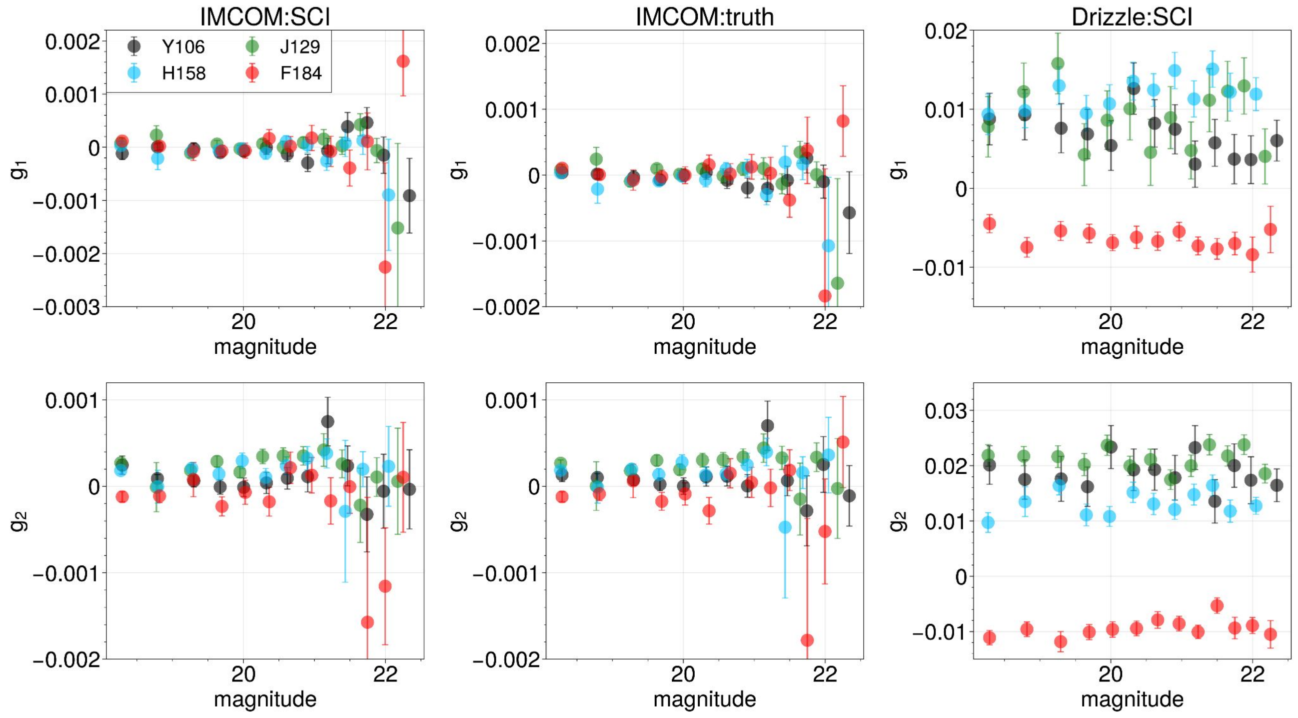
IMCOM vs. Drizzle: With vs. without control over output PSF

- Hirata *et al.* (2024) Figure 9
- Drizzle (Fruchter & Hook 2002): What your telescope sees is what you get
- IMCOM (Rowe *et al.* 2011): What you get is closest to what you want to see



IMCOM vs. Drizzle: “Stress test” with point sources

- Yamamoto *et al.* (2024) Figure 10
- “SCI”: simulated science images
- “truth”: simulated noiseless images
- g_1, g_2: ellipticities (expected to be 0 for point sources)



From fluffy-garbanzo + furry-parakeet to pyimcom

Image credits: PDPics (garbanzo) / Bru-nO (parakeet)

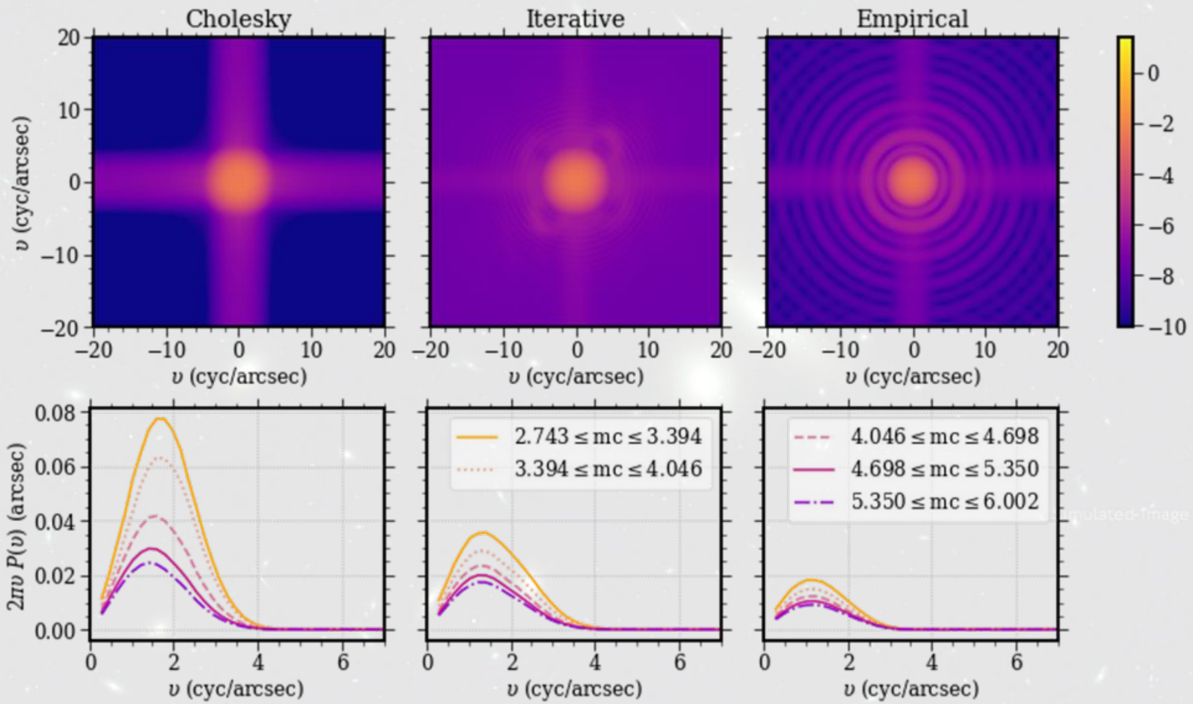
pyimcom: unified, object-oriented implementation

No. of core-hours per block (1 arcmin × 1 arcmin)

LA strategy	Hirata+ 24	Cholesky	Iterative
No. of cores	2 or 3	1 or 1.25	1 or 1.25
Y106 (hours)	41.54	7.87 ± 2.07	24.88 ± 7.05
J129 (hours)	47.53	13.11 ± 6.21	38.46 ± 9.16
H158 (hours)	61.02	11.74 ± 5.37	35.66 ± 9.68
F184 (hours)	58.51	34.26 ± 22.55	29.78 ± 6.75

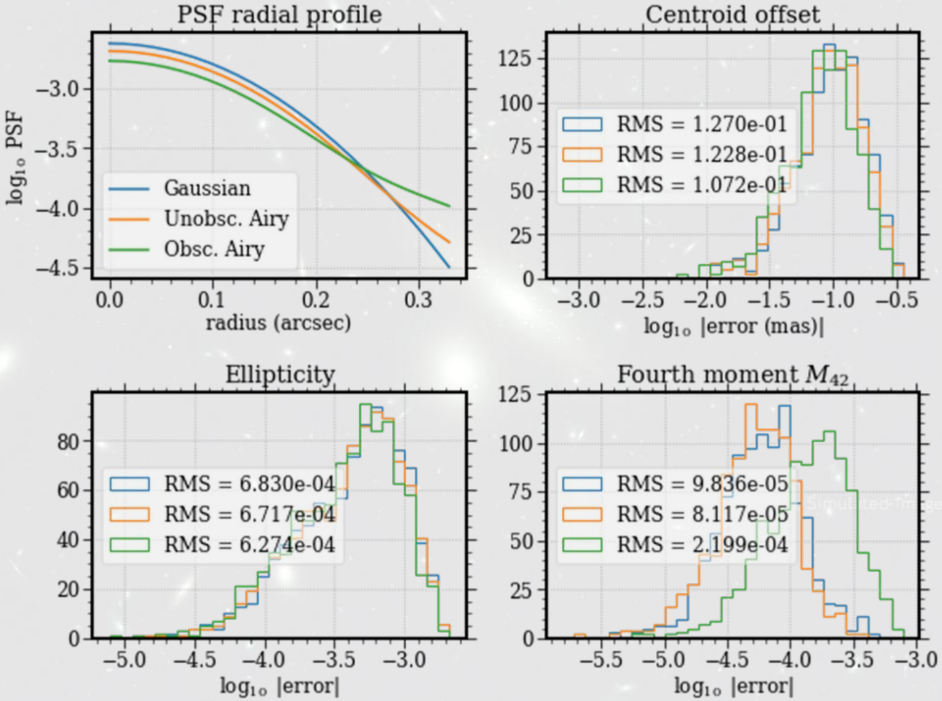
Work in process: New linear algebra strategies

- Power spectra of simulated input white noise in Y106 band
- Upper: 2D spectra, averaged over 16^2 blocks
- Lower: Azimuthally averaged spectra, binned by mean coverage (“mc”)



Work in process: Fine-tuning, e.g., choice of target PSF

- Injected point sources in K213 band
- Three PSFs with same FWHM
  - Simple Gaussian
  - Unobscured Airy disk*
  - Obscured Airy disk*
  - (*: Smoothed by a Gaussian.)
- Three shape measurements
  - Centroid (first moment)
  - Ellipticity (second moments)
  - Fourth moment (one of them; Zhang *et al.* 2023)



Discussion: Key takeaways and other ongoing projects

- Key takeaways
  - IMCOM is a linear image coaddition algorithm which provides control over reconstructed PSF in output images.
  - The new implementation is about an order of magnitude faster. It also provides alternative linear algebra strategies.
  - IMCOM is being fine-tuned for best weak lensing science yield.
- Other ongoing projects
  - Characterization and mitigation of biasing from read noise
  - Application of shear pipelines to IMCOM extended sources
  - Removal of large diffraction spikes via PSF splitting technique
  - Propagation of astrometry/flux calibration/PSF modeling errors

Simulated Image

Thank you!

Repository: https://github.com/kailicao/pyimcom

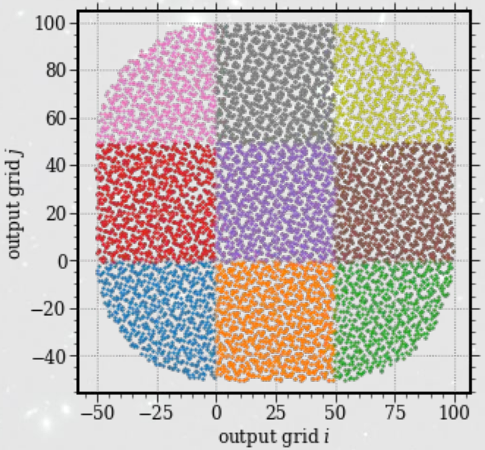
Speaker: Kaili Cao (Email: cao.1191@osu.edu)

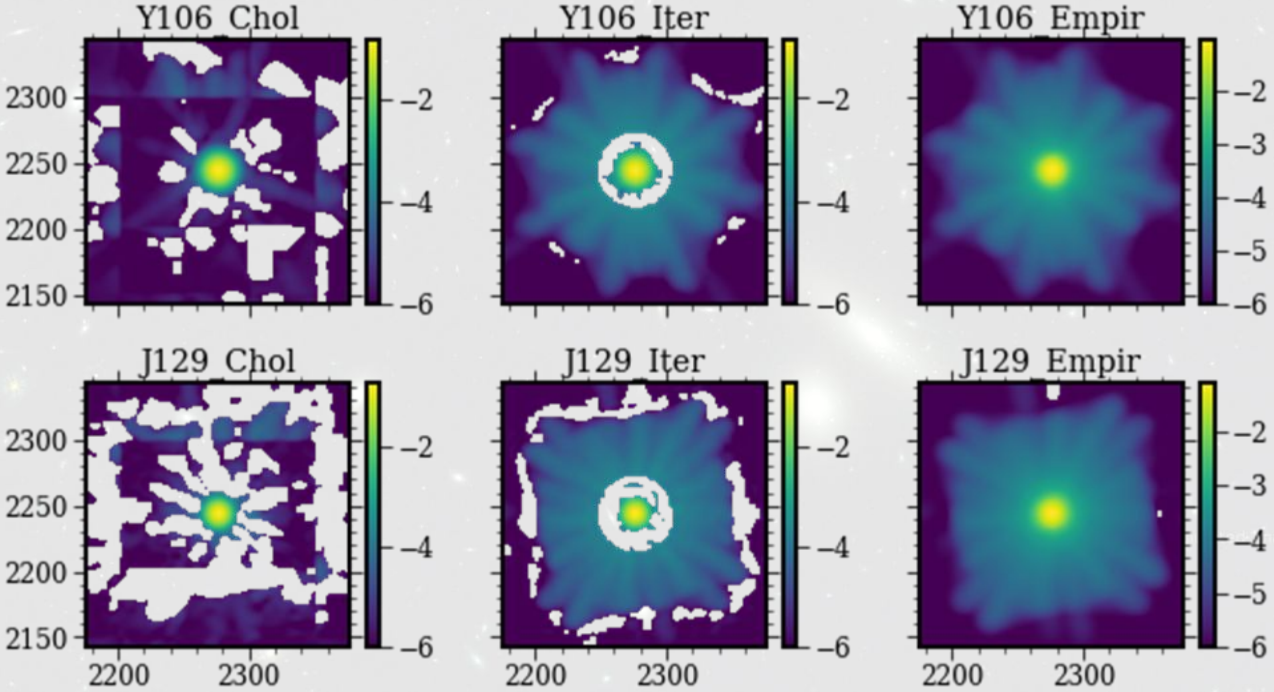
Co-Authors: Christopher M. Hirata (OSU), Katherine Laliotis (OSU), Masaya Yamamoto (Duke → Princeton), Emily Macbeth (OSU), Michael A. Troxel (Duke)

Acknowledgements: Roman HLIS Cosmology PIT, NASA Goddard Space Flight Center, Ohio Supercomputer Center

Backup: About postage stamp boundary effects

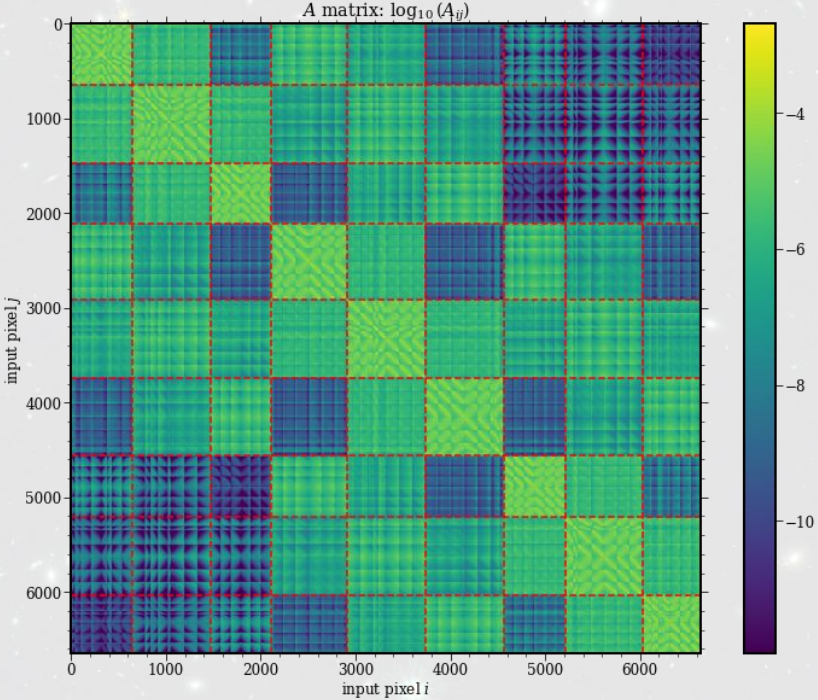
TL;DL: They are caused by input pixel selection.



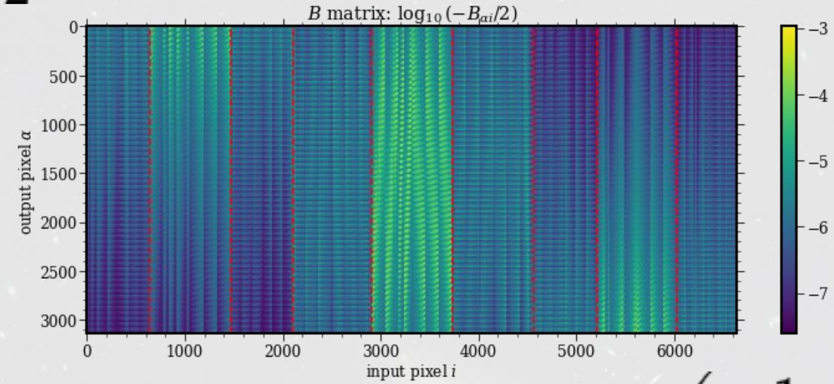


Backup: IMCOM formalism and example matrices

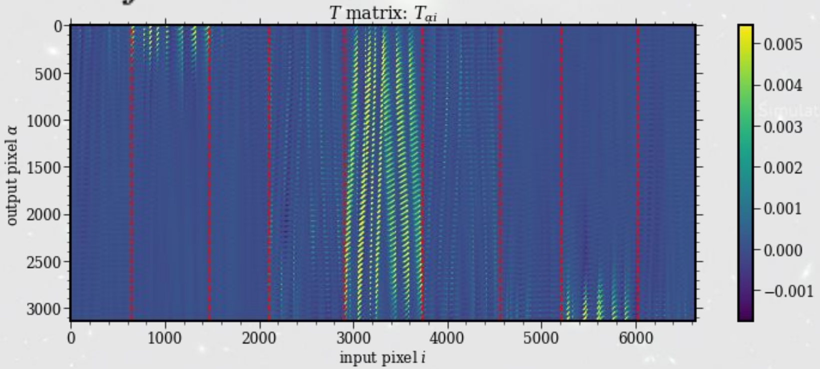
$$A_{ij} = [G_j \otimes G_i](\boldsymbol{r}_i - \boldsymbol{r}_j)$$



$$-\frac{1}{2}B_{\alpha i} = [\Gamma \otimes G_i](\boldsymbol{r}_i - \boldsymbol{R}_\alpha)$$



$$T_{\alpha i} = \sum_j [(\mathbf{A} + \kappa_\alpha \mathbf{I}_{n\times n})^{-1}]_{ij} \left(-\frac{1}{2}B_{\alpha j}\right)$$



Backup: Major operations of the IMCOM software

- Read input data, principally input signals (I_i), pixel positions ($\boldsymbol{r}_i$), and PSFs (G_i); parse configuration to get output pixel positions ($\boldsymbol{R}_\alpha$) and the target PSF (Γ).
- Perform fast Fourier transform (FFT) and inverse FFT to compute PSF overlaps ($G_j \otimes G_i$, $\Gamma \otimes G_i$, and $C = ||\Gamma||^2$).
- Perform interpolations (see Appendix A of Hirata *et al.* 2024 for details) using pixel positions to obtain system matrices ($\mathbf{A}$ and $\mathbf{B}$).
- Solve linear systems to get the optimal Lagrange multiplier κ_α and coaddition weights $T_{\alpha i}$ for each output pixel.
- Compute the output map (H_α), and report diagnostics for its quality (“PSF leakage” U_α / C and “noise amplification” Σ_α).