Parallel-Processing Astronomical Image Analysis Tools for *HST* & *SIRTF*

Kenneth John Mighell

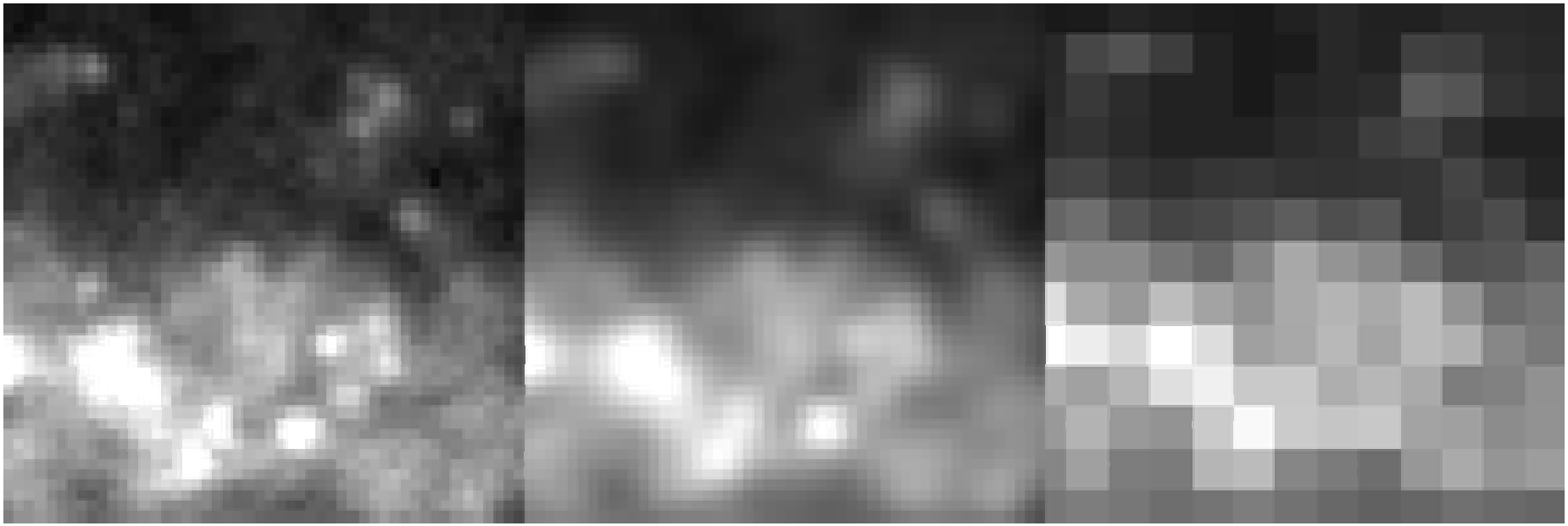
National Optical Astronomy Observatory

Outline

- Parallel-Processing Astronomical Image Analysis Tools for *HST* & *SIRTF*
  (Interagency order # **S-13811-G**)

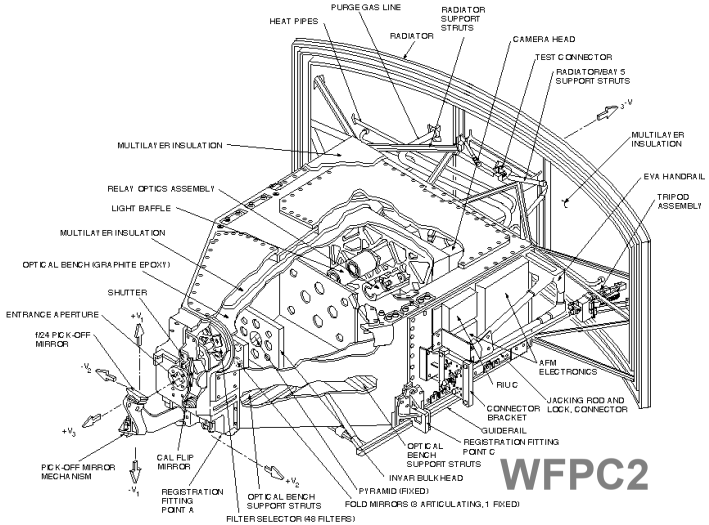
- Fault-Tolerant Parallel-Processing Astronomical Image-Analysis Tools
  (Interagency order # **NNG05EB61I**)

Image Formation Process



Fruchter & Hook 2000 (ADASS IX)

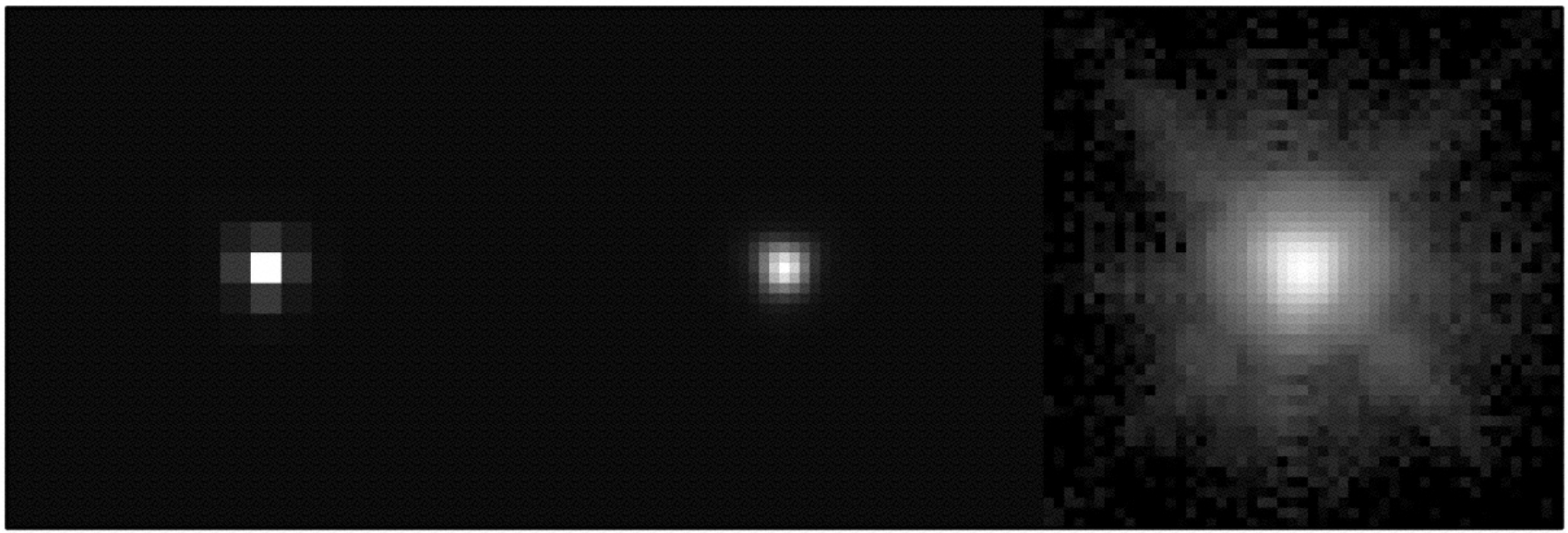
SKY ⇒ TELESCOPE ⇒ DETECTOR



Under-Sampled Point Spread Functions

HST WFPC2: WFC F555W 0.10 arcsecond pixels

1.7 arcsecond



Lauer 1999, PASP, 111, 1434

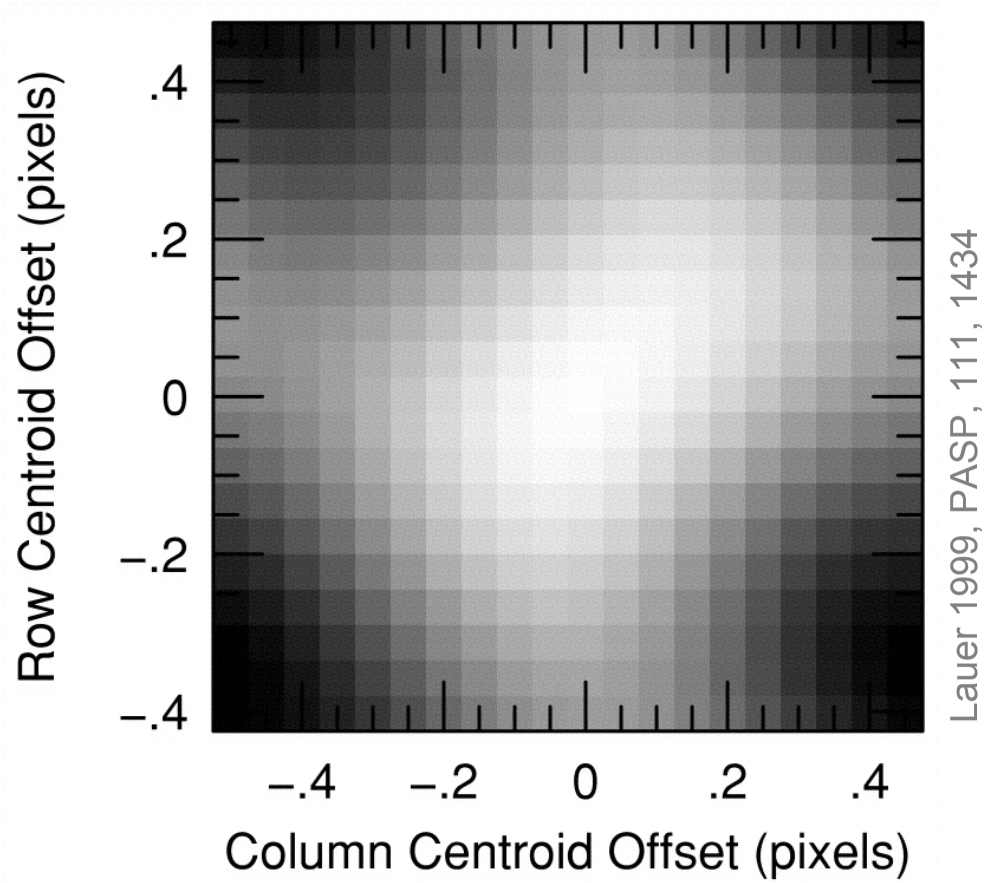
original sampling
linear stretch

3x3 supersampled
linear stretch

3x3 supersampled
logarithmic stretch

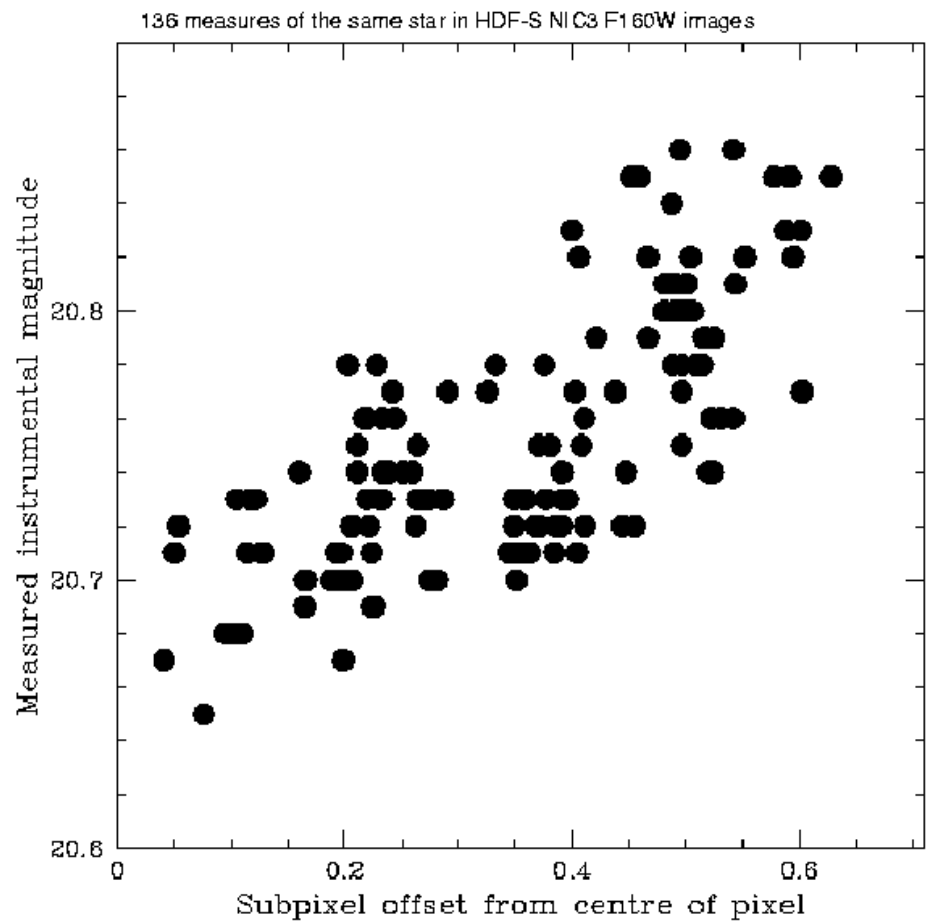
Non-Uniform Pixel Response Function

Photometric Error: NIC3 F160W



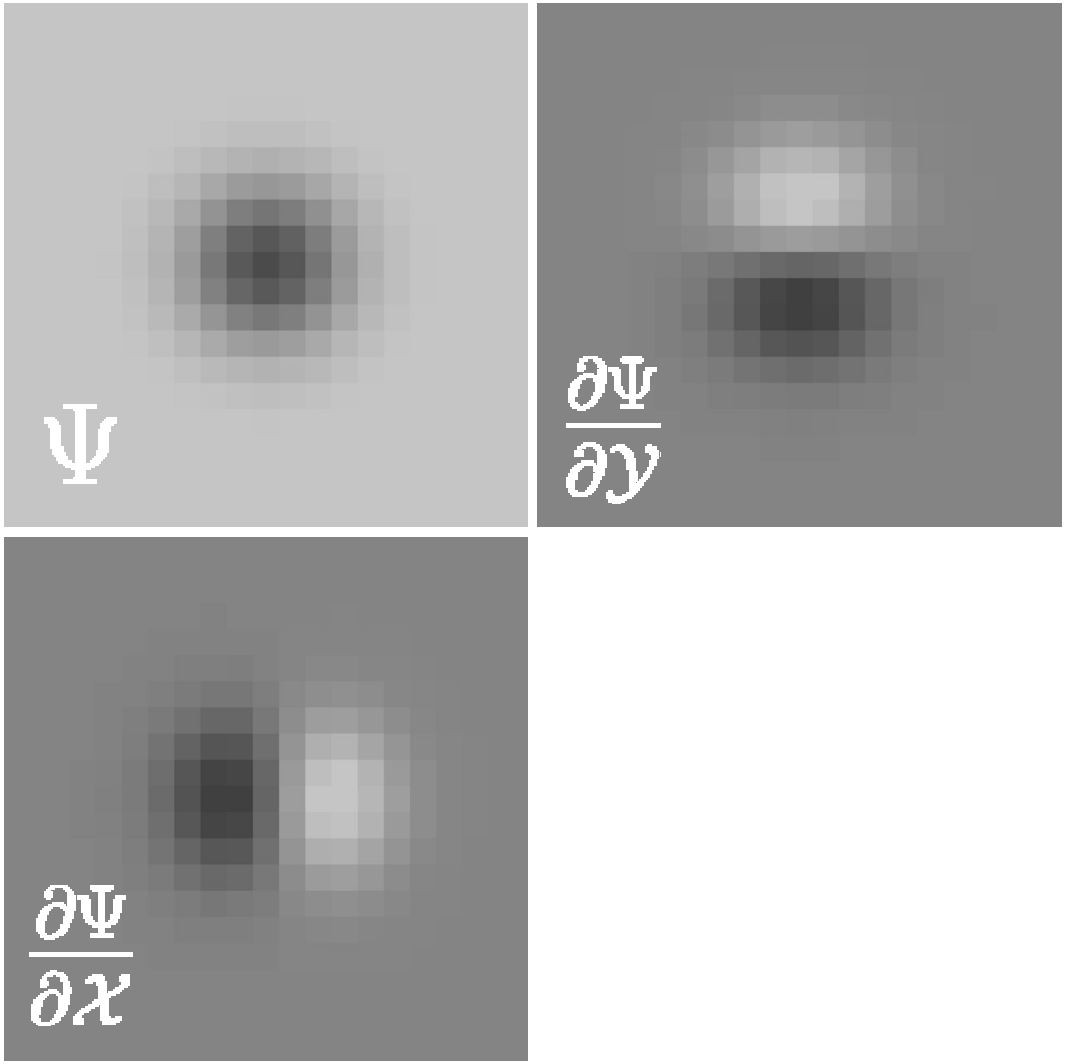
Lauer 1999, PASP, 111, 1434

white=0.12 mag excess, black=0.09 mag deficit



Hook & Fruchter 2000 (ADASS IX)

Derivatives of the PSF



Numerical Derivatives

The mathematics of determining the position partial derivatives of the observational model with respect to the x and y direction vectors is exactly the same with analytical or numerical PSFs. The implementation methodology, however, is significantly different. The position partial derivatives of numerical PSFs can be determined using numerical differentiation techniques on the numerical PSF. Numerical experiments have shown that the following five-point differentiation formula works well with numerical PSFs:

$$f'(x_i) \approx \frac{1}{12}\left[f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})\right]$$

How does one move a PSF?

Analytical PSFs: Just compute the PSF at the desired location in the observational model.

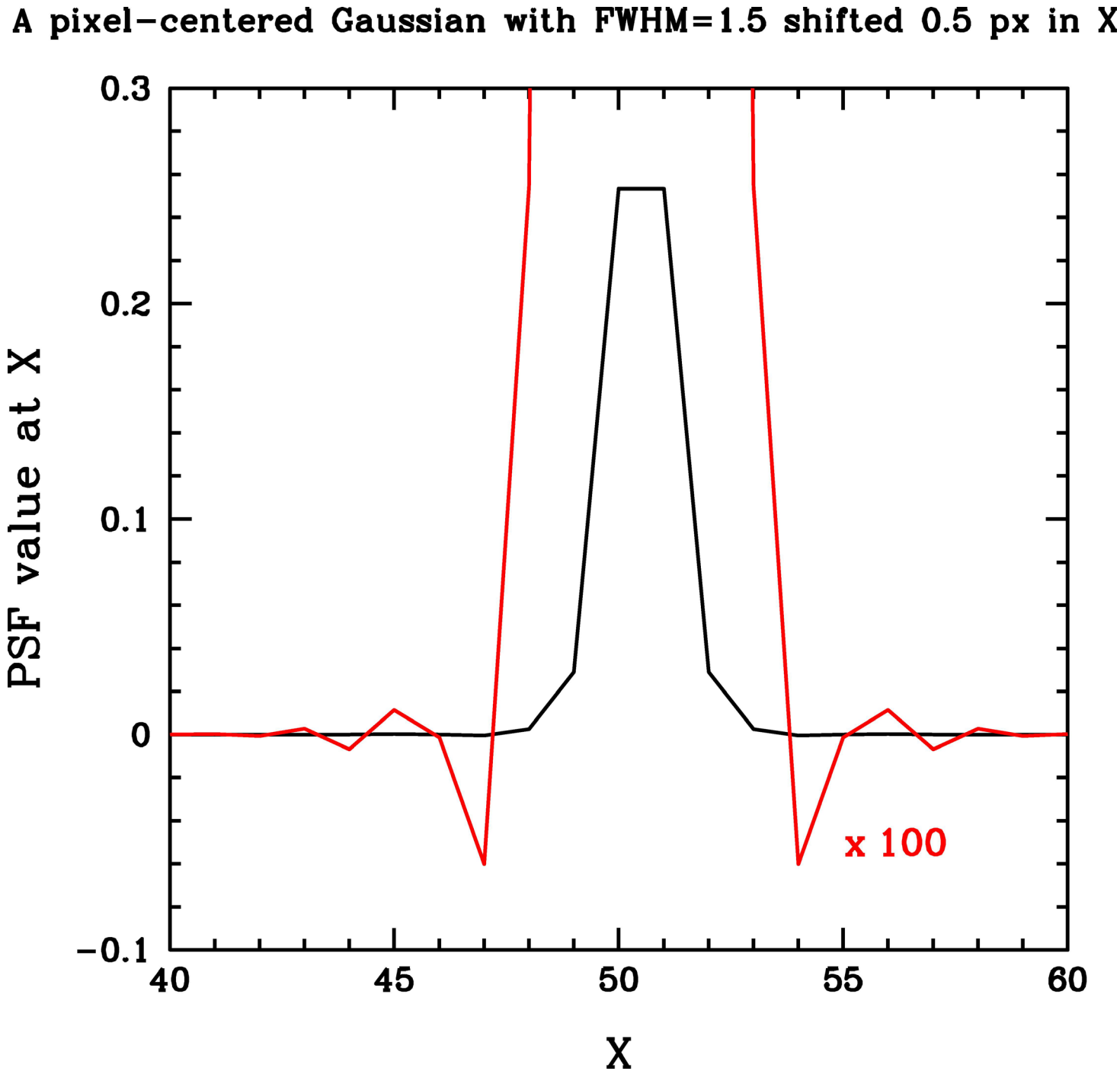
Numerical PSFs: Take the reference numerical PSF and shift it to the desired location using a perfect 2-d interpolation function. OK… but how is that done in practice?

Solution

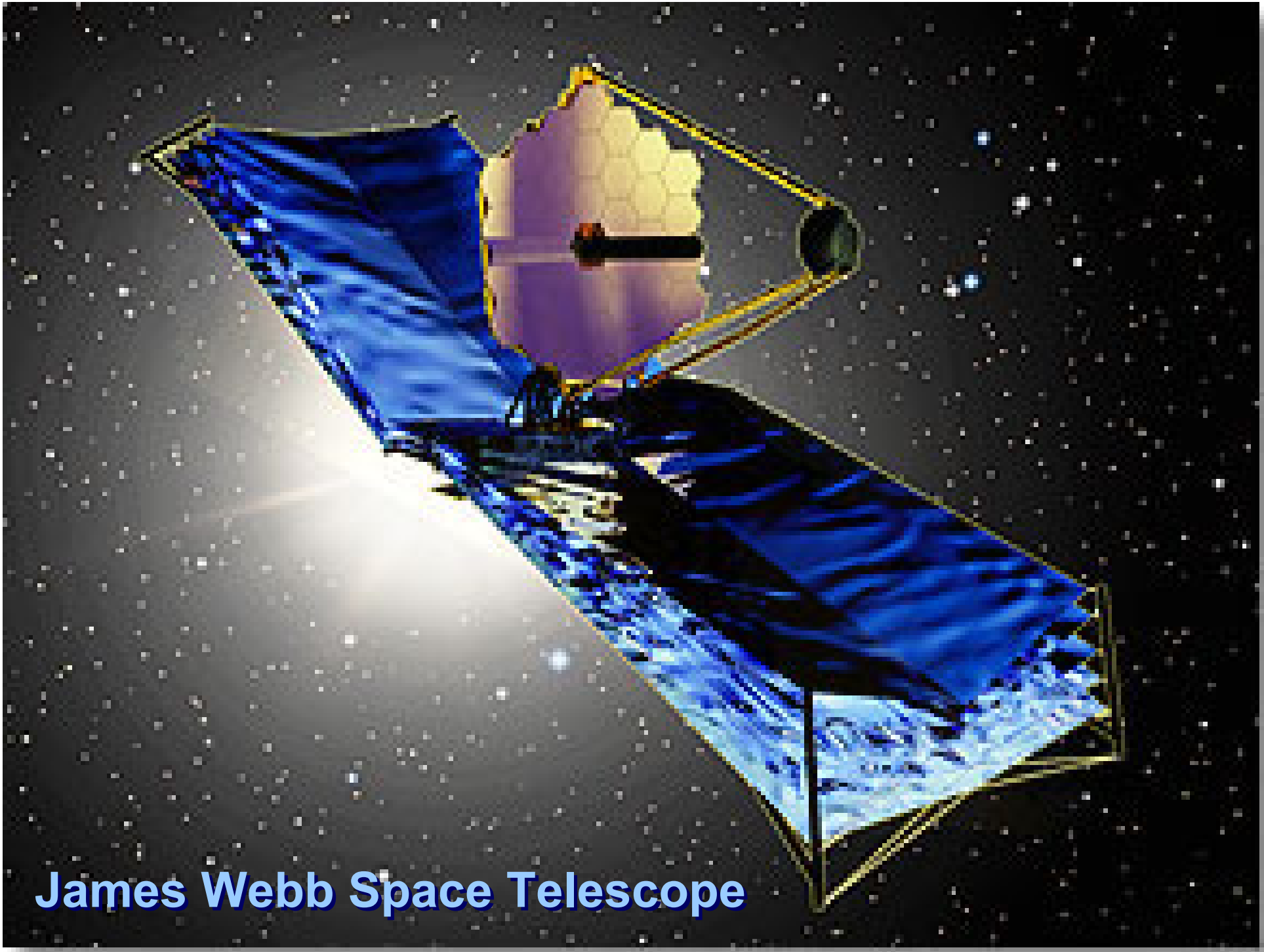
Use the following 21-pixel-wide damped sinc function:

$$f^{\text{shifted}}(\delta x) \equiv \sum_{i=-10}^{10} f(x_i) \frac{\sin\left(\pi(x_i - \delta x)\right)}{\pi(x_i - \delta x)} \exp\left(-\frac{(x_i - \delta x)^2}{(3.25)^2}\right)$$

Note: The 2-d sinc function is separable in x and y.



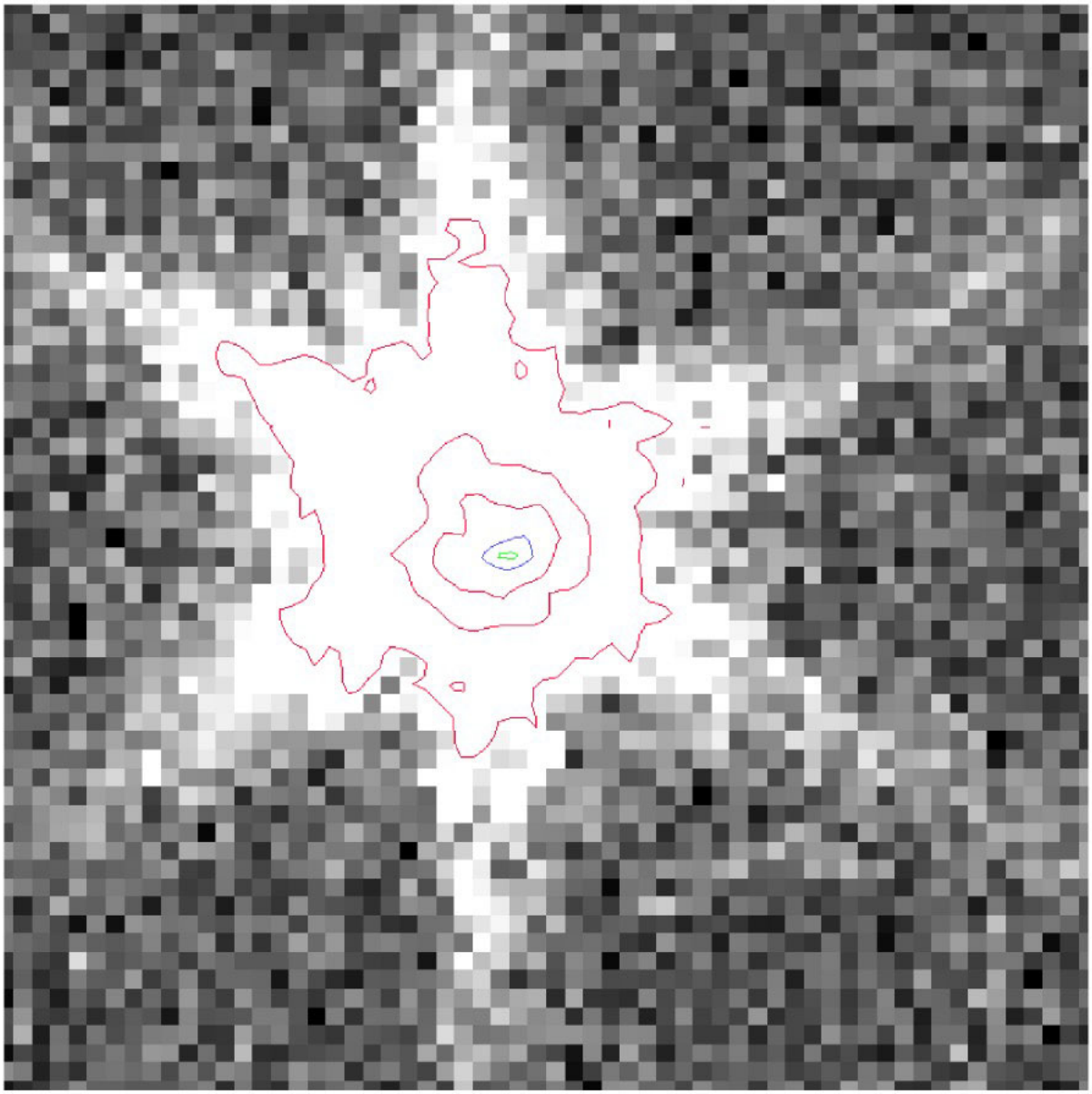
Problem: Negative values caused by shifting an undersampled PSF


James Webb Space Telescope

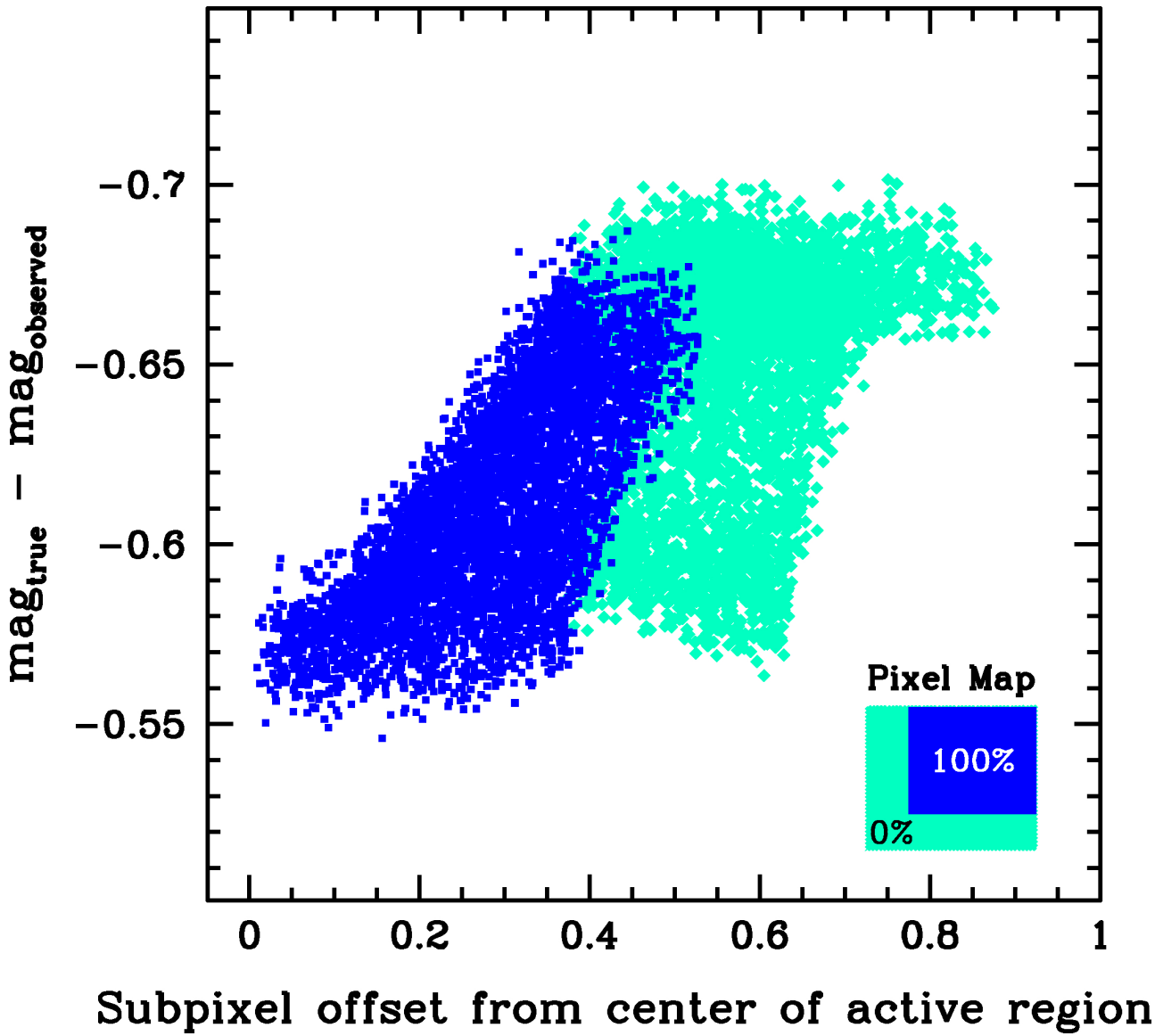
Next Generation Space Telescope

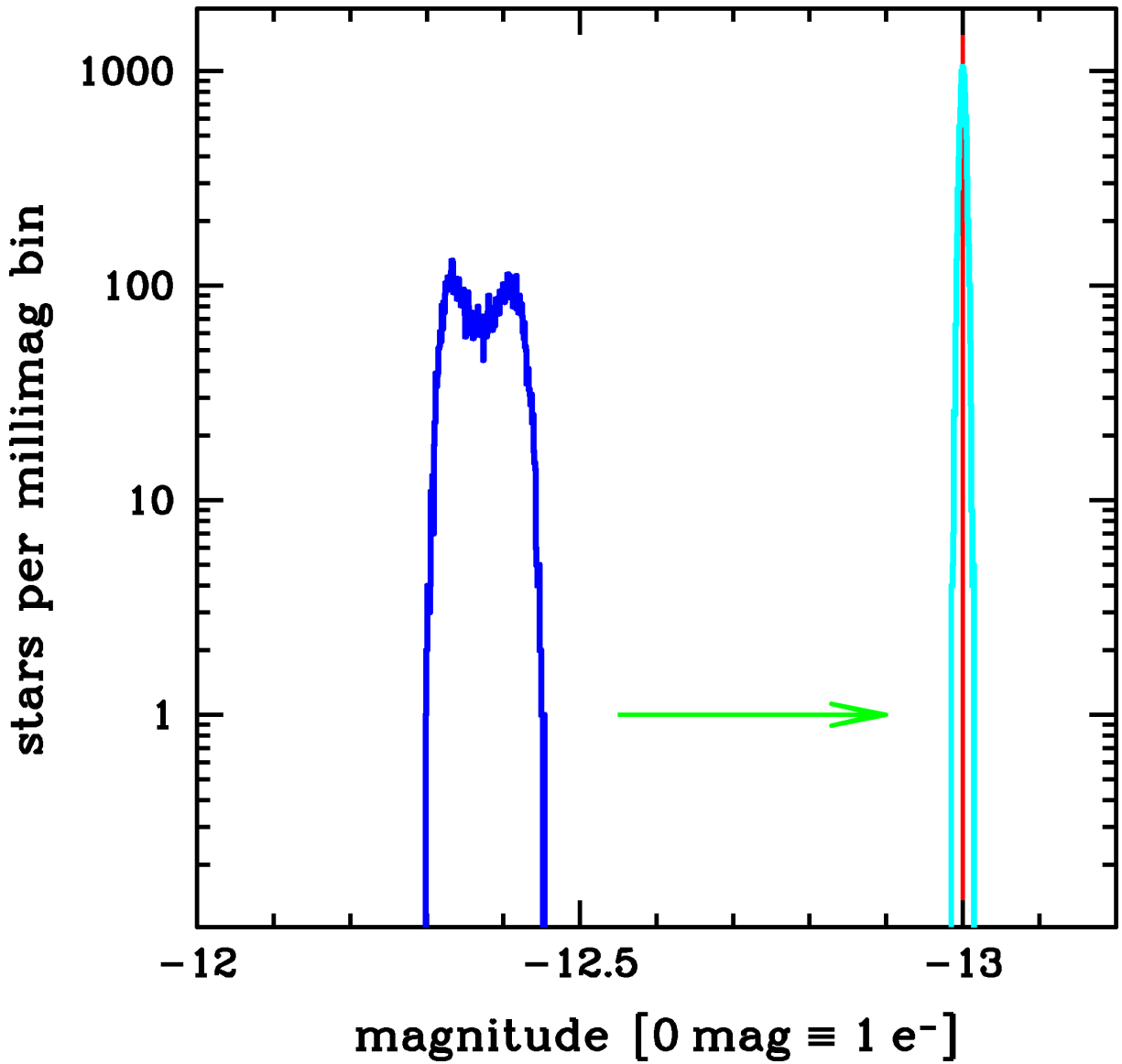
8-m TRW-concept

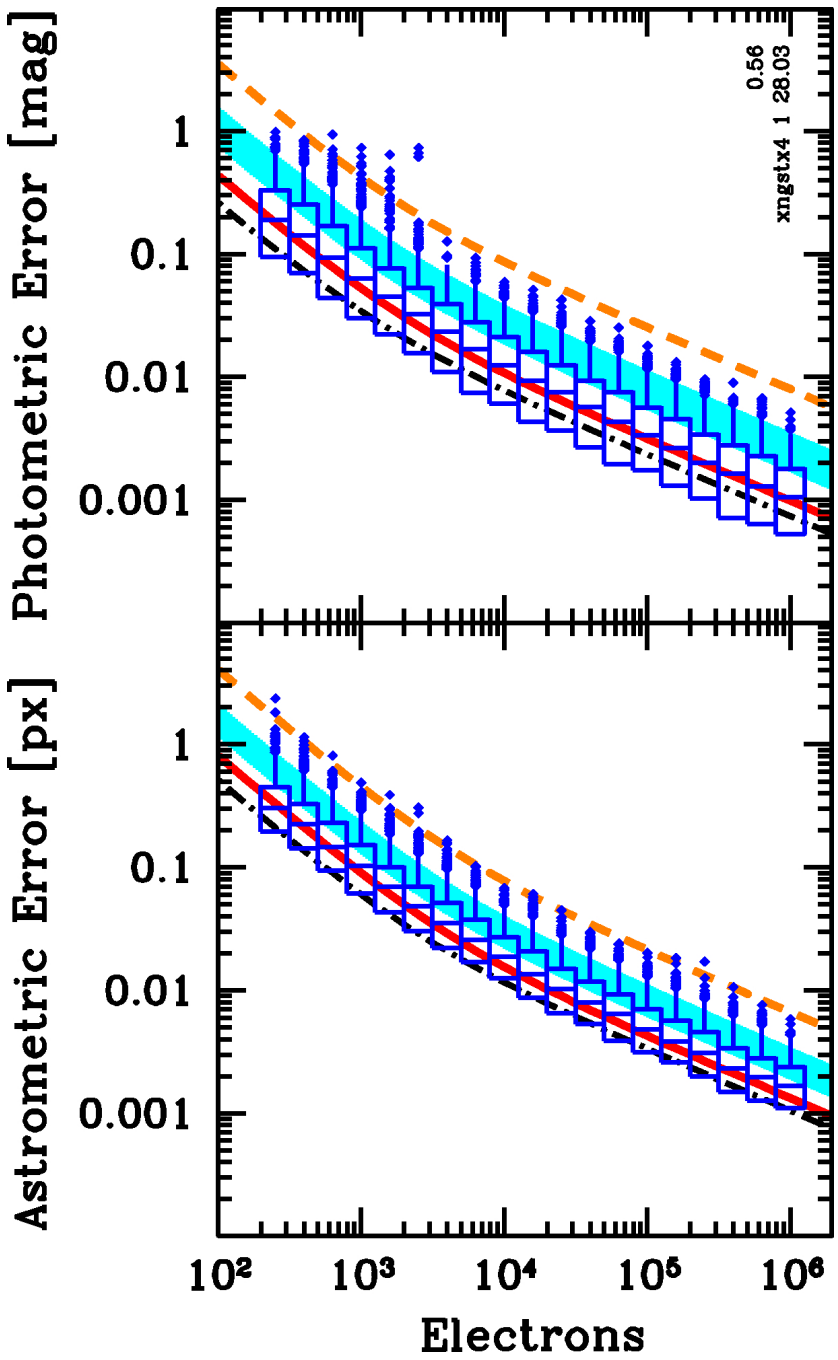
contours: 90%, 50%, 10%, 1%, 0.1% of peak



0.0128 arcsec/pixel β=31.60 PSF by John Krist


−0.7
−0.65
−0.6
−0.55
0
0.2
0.4
0.6
0.8
1
$mag_{true} - mag_{observed}$
Subpixel offset from center of active region
Pixel Map
100%
0%


1000
100
10
1
stars per millimag bin
−12
−12.5
−13
magnitude [0 mag ≡ 1 e⁻]


Photometric Error [mag]
Astrometric Error [px]
1
0.1
0.01
0.001
0.56
xngstx4 1 28.03
10^2
10^3
10^4
10^5
10^6
Electrons

The MATPHOT Algorithm for Accurate and Precise Stellar Photometry and Astrometry Using Discrete Point Spread Functions

Mighell, K. J. (NOAO)

American Astronomical Society Meeting # 205, Poster #153.09, 2005 January 13, San Diego, CA
http://www.aas.org/publications/baas/v36n5/aas205/920.htm

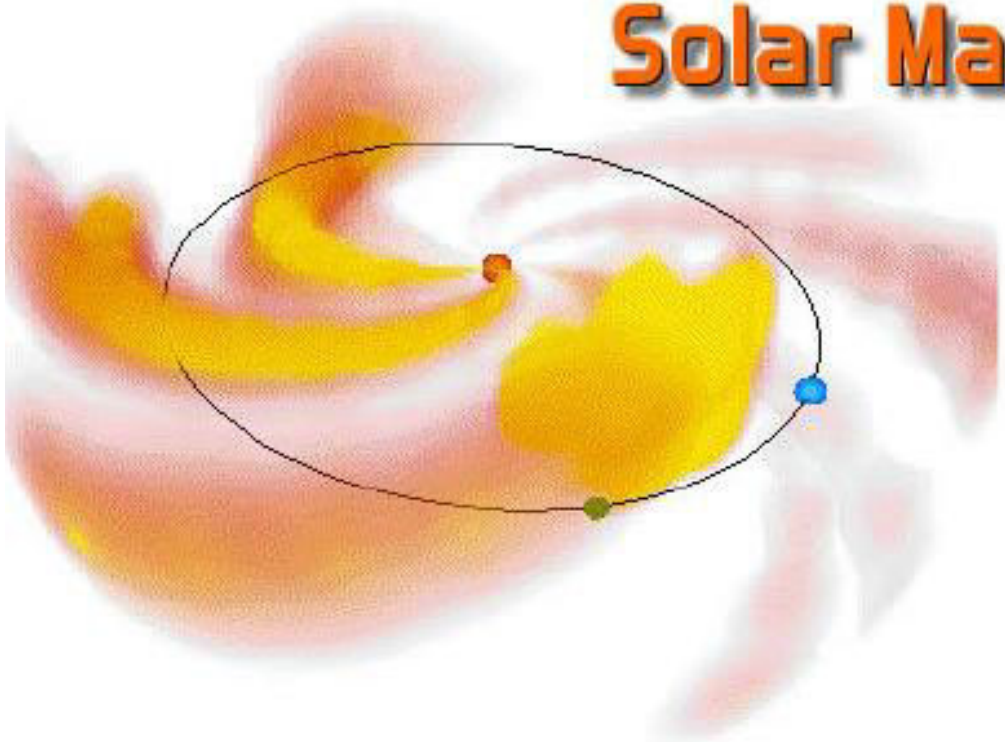
ABSTRACT

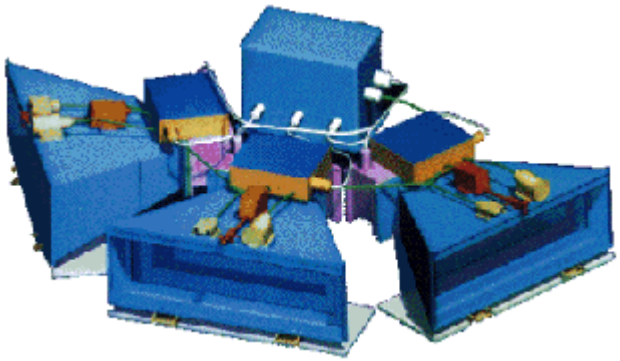
I describe the key features of my MATPHOT algorithm for accurate and precise stellar photometry and astrometry using discrete Point Spread Functions. A discrete Point Spread Function (PSF) is a sampled version of a continuous two-dimensional PSF. The shape information about the photon scattering pattern of a discrete PSF is typically encoded using a numerical table (matrix) or a FITS image file. The MATPHOT algorithm shifts discrete PSFs within an observational model using a 21-pixel-wide damped sinc function and position partial derivatives are computed using a five-point numerical differentiation formula. The MATPHOT algorithm achieves accurate and precise stellar photometry and astrometry of undersampled CCD observations by using supersampled discrete PSFs that are sampled 2, 3, or more times more finely than the observational data. I have written a C-language computer program called MPD which is based on the current implementation of the MATPHOT algorithm; all source code and documentation for MPD and support software is freely available at the following website: http://www.noao.edu/staff/mighell/matphot . I demonstrate the use of MPD and present a detailed MATPHOT analysis of simulated James Webb Space Telescope observations which demonstrates that millipixel relative astrometry and millimag photometric accuracy is achievable with very complicated space-based discrete PSFs. This work was supported by a grant from the National Aeronautics and Space Administration (NASA), Interagency Order No. S-13811-G, which was awarded by the Applied Information Systems Research (AISR) Program of NASA's Science Mission Directorate.

Relevance to NASA

The MATPHOT algorithm can be applied to stellar imaging observations from current and future NASA astrophysical missions with imaging instrumentation with complicated Point Spread Functions like the *Hubble Space Telescope*, the *Spitzer Space Telescope*, and the *Solar Mass Ejection Imager* (*SMEI*).

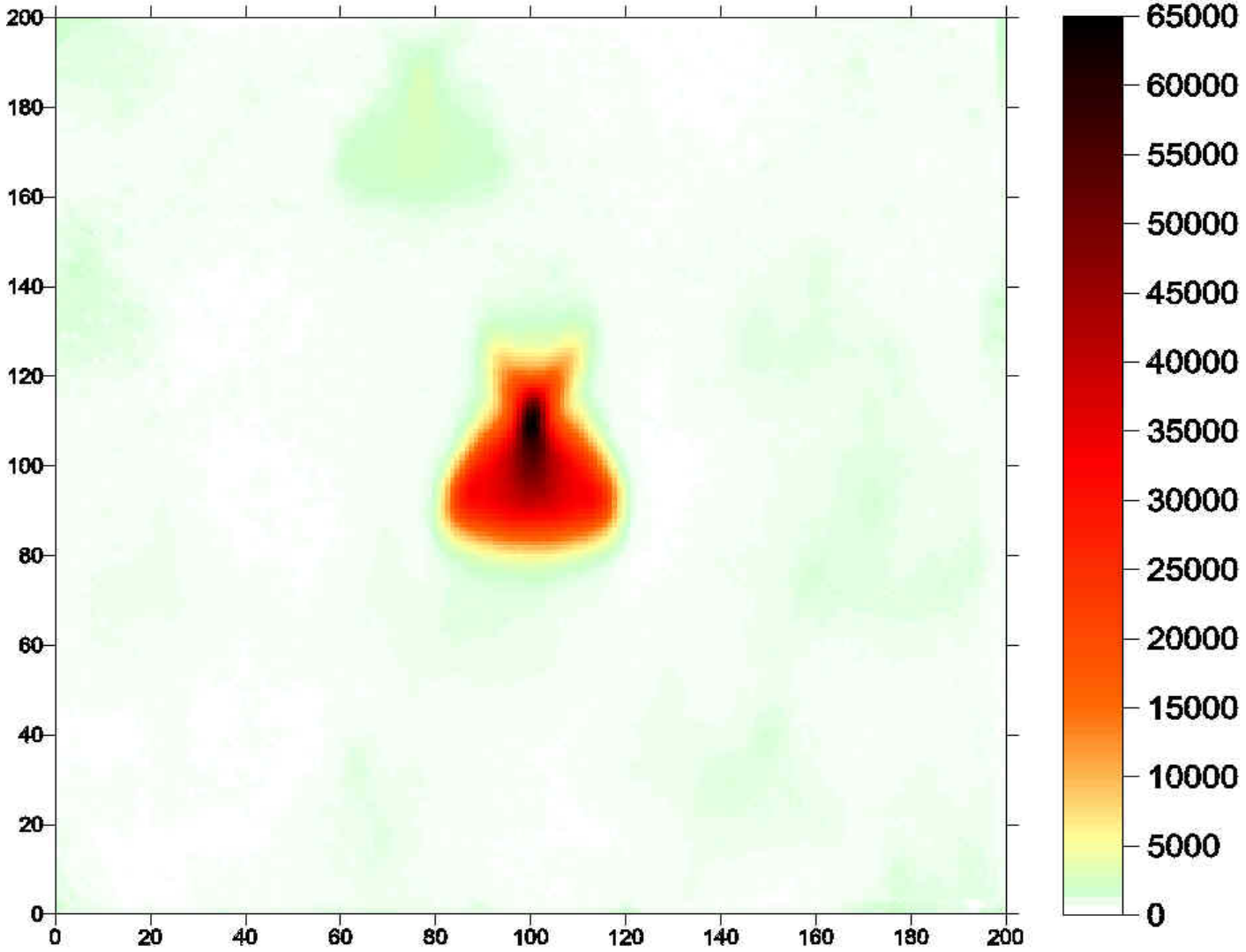
Solar Mass Ejection Imager





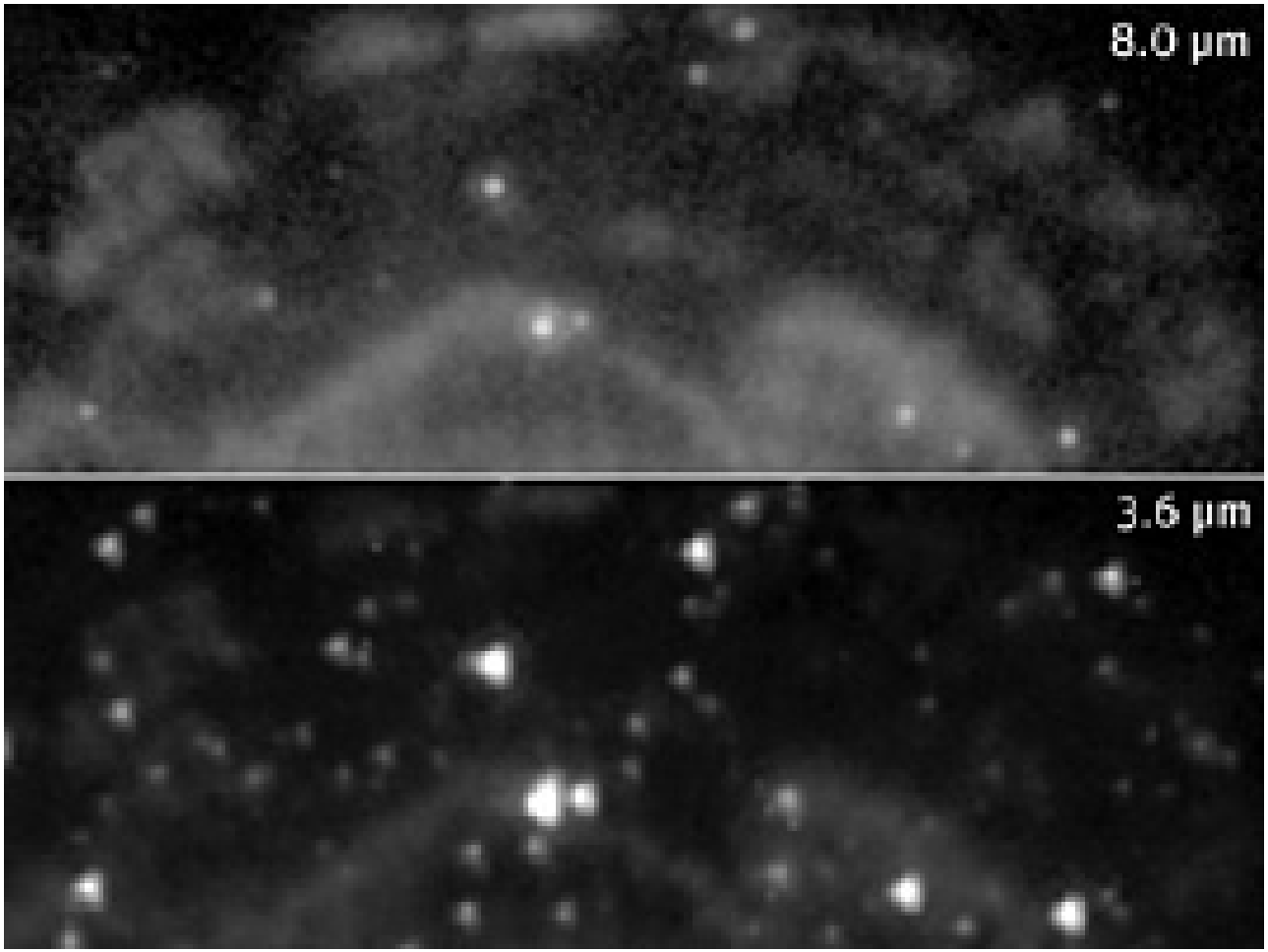
Andy Buffington (SMEI):

Can you fit a fish?



Spitzer observations of the Ring Nebula (M57)

Spitzer suffers from trefoil aberration below 5.4 microns



Spitzer composite of M57
http://skyandtelescope.com/news/article_1463_1.asp

This work is supported by a grant from the **Applied Information Systems Research Program (AISRP)** of the Science Mission Directorate of the National Aeronautics and Space Administration (NASA) [NRA 01-OSS-01: Interagency Order No. S-13811-G] .

Fault-Tolerant Parallel-Processing Astronomical Image-Analysis Tools

Kenneth John Mighell

National Optical Astronomy Observatory

The Friendly Janitor Problem

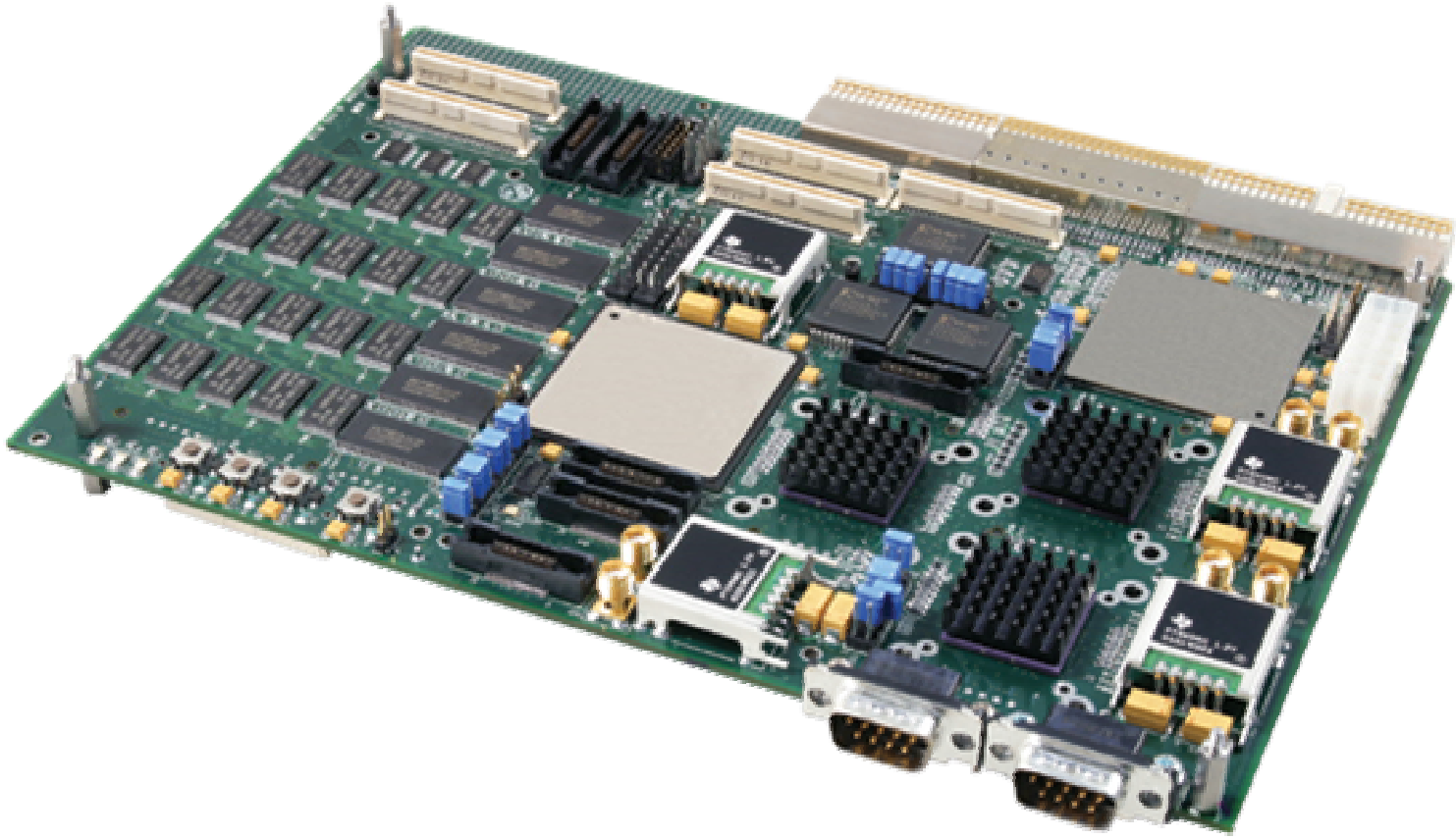
Suppose a friendly janitor walks into a room containing all the routers and connections for a large cluster of networked computers. Upon seeing the unsightly tangle of CAT5 cables, the janitor starts ripping them out to untangle the mess. ***Would a parallel-processing application running on the cluster be able to survive such a catastrophic failure?***

Fault Tolerance

- Most parallel-processing applications today are not capable of surviving major network communication failures.
- A well-designed fault-tolerant application should be able to survive a “helpful” janitor (on the ground) or a blast of high-energy particles from a solar flare (in orbit).

NMP ST – 8: COTS BASED HIGH PERFOMANCE COMPUTING

- ≥3 COTS CPUs each ≥ 1 billion instructions/sec
- capable of supporting ≥100 nodes
- bandwidth ≥1 billion bits/sec
- COTS standard OS for distributed systems
- 100 MIPS/Watt (sustained, peak: 300)
- radiation tolerant with minimal shielding
- 5-year mission lifetime reliability:
  - Goal: 99.999%
  - Minimum: 99.9%



Maxwell Technologies SCS750 Prototype

Status Quo is Changing

The evolution of parallel computer architectures has recently created new trends and challenges for both parallel application developers and end users.

- **Monolithic systems → clusters of PCs**
- **Proprietary ($$$$$) → COTS-based ($$$)**
- **10^3 CPUs today → 10^5 CPUs soon**

Problems with the Status Quo

- Clusters of PCs, unfortunately, provide a less integrated environment and therefore often have **inferior I/O behavior** than previous architectures.
- Grid and metacomputing efforts yield further increase in the number of processors available to parallel applications at the cost of **increasing** the **physical distance** between computational nodes.

New Challenges for MPI (1)

- An MPI application utilizing thousands of processors faces many **scalability issues** that can dramatically impact the overall performance of any parallel application:
  - **Process Control**
  - **Resource Exhaustion**
  - **Latency Awareness and Management**
  - **Fault Tolerance**
  - **Optimized Collective Operations for Common Communication Patterns**

New Challenges for MPI (2)

- **Network layer transmission errors**
  - Highly improbable for moderate-sized clusters but can no longer be ignored when dealing with large scale computations.
- **Process failures**
  - The probability that a parallel application will encounter a process failure during its run increases with the number of processors it uses.

New Challenges for MPI (3)

- If a parallel application is to survive a process failure without having to restart from the beginning; it either must **regularly write checkpoint files** (and restart the application from the last consistent checkpoint) or the application must be able to **adaptively handle process failures** during runtime.

Need for a New MPI Implementation

- **No MPI implementation available** today is capable of addressing all these issues simultaneously.
- **A new MPI implementation is needed**: one that is capable of providing a framework to address these important issues in emerging networks and architectures.

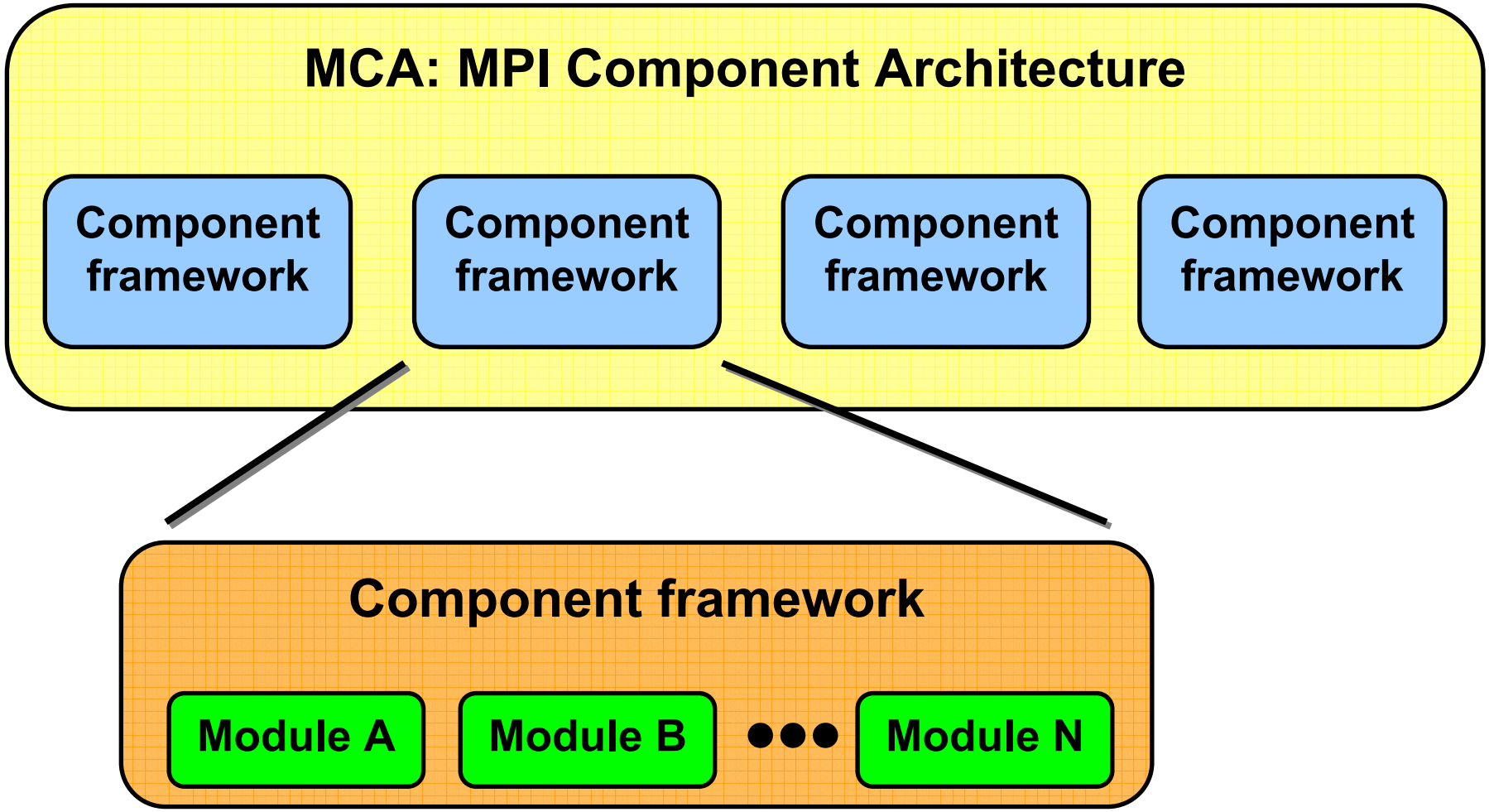
Open MPI Project

- Open MPI is an **all-new**, **production-quality implementation** of the **MPI 2** standard which is fundamentally centered around component concepts.
- Tutorial at SC2004 at Pittsburgh on 09/08/2004
- **First full release** of Open MPI is planned for ~April 2005
- Open MPI has **bandwidths that are 30% higher** for large messages when compared with existing implementations; ping-pong latencies are comparable.

Open MPI: Partners

- Open MPI is built upon prior research and has been influenced by experience gained from the codes bases of mature MPI implementations.
- The initial partners of the Open MPI are:
  - **Advanced Computing Laboratory** (LA-MPI)
    Los Alamos National Laboratory
  - **Open Systems Laboratory** (LAM/MPI)
    University of Indiana
  - **Innovative Computing Laboratory** (FT-MPI)
    University of Tennessee
  - Sandia National Laboratory – Livermore
- New partners soon …

Architecture of Open MPI



Component Framework

- Each component framework is dedicated to a single task, such as providing parallel job control or performing MPI collective operations.
- Upon demand, a *framework* will discover, load, use, and unload *modules.*
- Each framework has different policies and usage scenarios; some will only use one module at a time while others will use all available modules simultaneously.

Modules

- Modules are self-contained software units that can configure, build, and install themselves.
- Modules provide requested services to higher-level tiers and other parts of MPI.
- Modules adhere to the interface prescribed by their parent component framework.

Component Frameworks

- Point-to-point Transport Layer
- Point-to-point Management Layer
- Collective Communication
- Process Topology
- Reduction Operations
- Parallel I/O
- and many more…

Flexibility

The wide variety of framework types allows third party developers to use Open MPI as a

- deployment vehicle for commercial products
- comparison mechanism for different algorithms and techniques
- research platform

Summary

- Open MPI is a new implementation of the MPI standard. It provides new functionality that has not been previously seen in any single open-source, production-quality MPI implementation, including support for all of MPI 2, multiple concurrent user threads, and multiple options for handling process and network failure.
- Open MPI will debut ~April 2005.
- Coming soon to **http://www.open-mpi.org**

This work is supported by a grant from the **Applied Information Systems Research Program (AISRP)** of the Science Mission Directorate of the National Aeronautics and Space Administration (NASA) [NRA 01-OSS-01: Interagency Order No. NNG05EB61I] .