Roman Grism Simulations with Multiple Orders and Distortions

Austen Gabrielpillai (Code 665), Isak Wold (Code 660), James Rhoads (Code 665), Sangeeta Malhotra (Code 661), Anton Koekemoer (STScI)

Introduction: Slitless Spectroscopy (Grism)

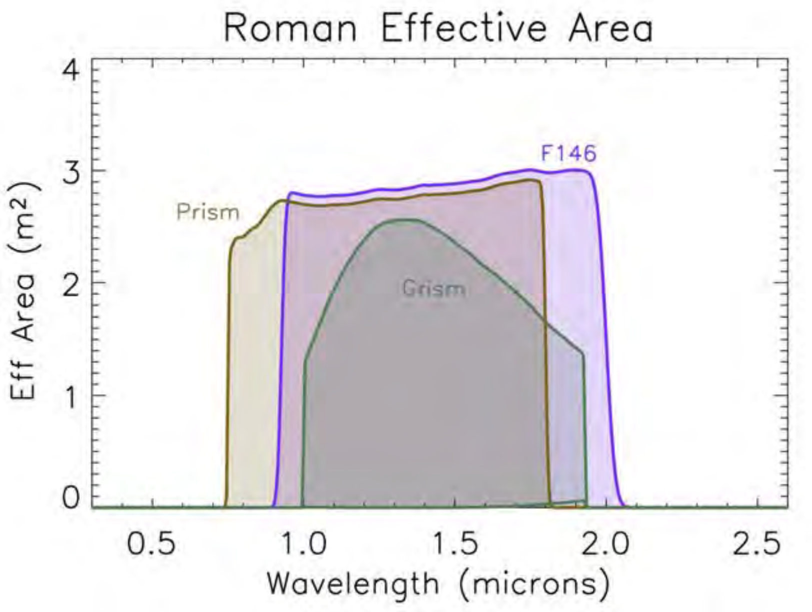
diffraction grating + prism = grism

Disperse the spectra of all objects in the field-of-view

Pro: Great tool for identifying emission line and high-redshift galaxies

Con: Sources can contaminate each other (mitigated in a variety of ways)

Roman Space Telescope's grism spans 1.0 - 1.93 micron, capable of resolving z > 7 Lya emitters



https://roman.gsfc.nasa.gov/science/WFI_technical.html

The Simulation: Overview

Goal: create mock Roman grism foreground simulations and inject synthetic Lya emitters between 7.5 < z < 10.5

Three step process and pipelines:

1. **Pre-processing** - datacube as a function of wavelength using HST images
2. **Simulation** - Assign fluxes to detector coordinates while accounting for distortions
3. **Source injection** - add emission line galaxies into simulated foreground

Simulation built ground up in Python (utilizing just-in-time compilation)

The Simulation: Data Cube

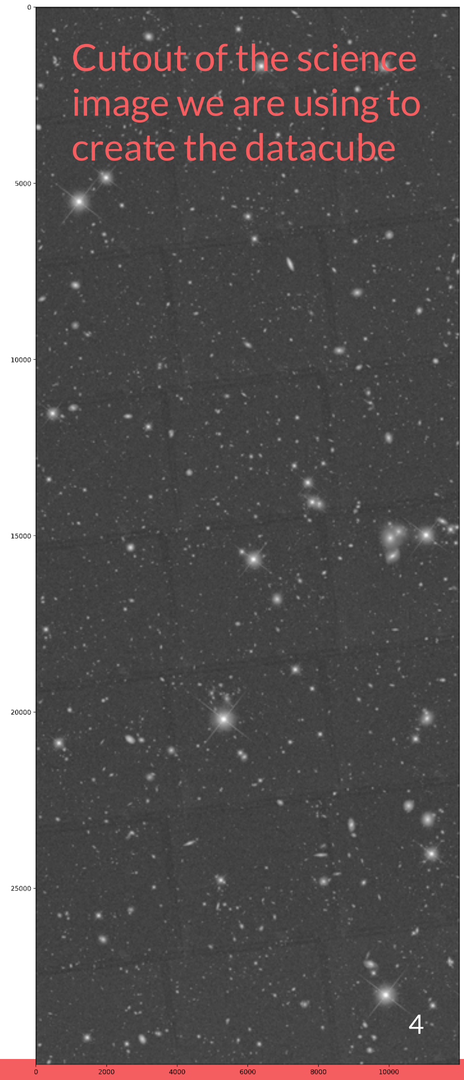
Input data cube using HST images and Spectral Energy Distributions (SEDs) from COSMOS field

Source image: 0.03”/pixel resolution, approximately 6’ x 15’

Use segmentation map to identify objects and morphologies

Resample the SEDs to chosen wavelength step (1A)

Create N slices where each object in a slice has its resampled flux



The Simulation: Grism, Distortions and Orders

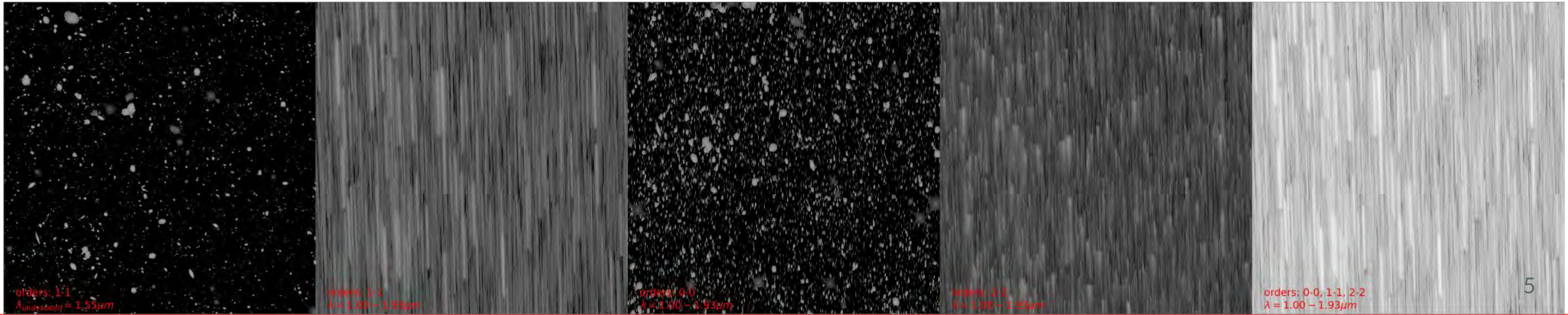
transform image pixel coordinates to sky coordinates via world coordinate system

Modeling main order (1-1) and two off orders (0-0 and 2-2) via engineering team's polynomials

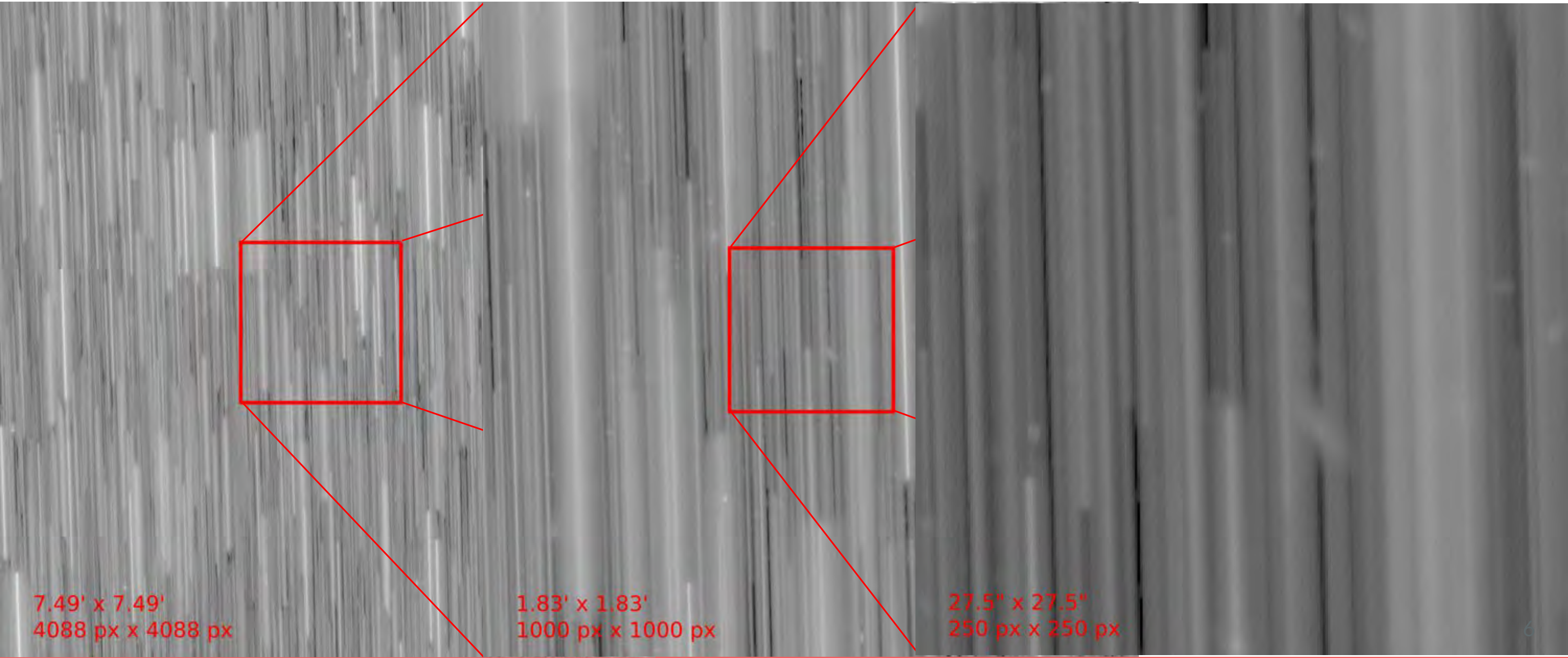
nearest neighbor pixel assignment where flux of a sky pixel is assigned to the closest grism pixel

(source: 0.03"/pixel, grism: 0.11"/pixel)

Account for position dependent wavelength cutoffs, trace, and dispersion solution



The Simulation: Output and Resolution

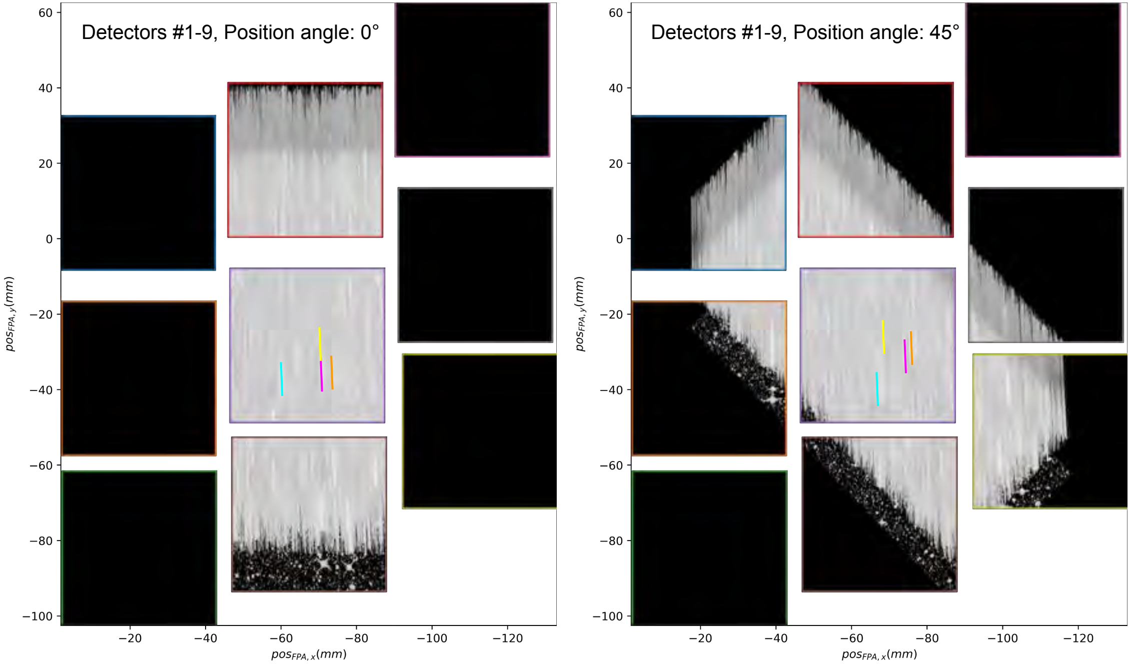


The Simulation: Position Angles

Position angle is defined in the configuration file

Rotate sky coordinates around a chosen midpoint

Sources previously contaminated can be untangled through post-processing



Note: the dataset is only able to span across ~3 detectors, limiting our observational data's FOV

The Simulation: Design Challenges

Can’t run on local machine after switching from 0.06” -> 0.03” resolution

- **Solved**: Obtained access to Discover Supercomputing Cluster (more CPUs, nodes, and RAM)

High resolution input increased computation time, less runs per day

- **Solved**: Used numba module to improve computation speed by a factor of 4 (12 hours across 5 nodes -> 3 hours across 5 nodes per detector)

EAZY SED templates does not account for stars and some bright objects

- **Solving**: Comparing and normalizing output SEDs with H band magnitudes

Finale: Summary

Input: COSMOS field image at 0.03”/pixel resolution

Pre-processing: creates a datacube of N wavelength steps using resampled SEDs

Simulation: Python based. Modeling main and two off orders, distortions, and can choose position angles

Coming soon: data release and data challenge, methods paper

Contact Info

Email: a.gabrielpillai@gmail.com

Twitter: @a_gabrielpillai

Website: aust427@github.io