TURNING BACK TIME

Computational Techniques for Star Trail Removal from Long-Exposure Astrophotography

Nikhil Gandudi Suresh | Avanti Bhandarkar

01

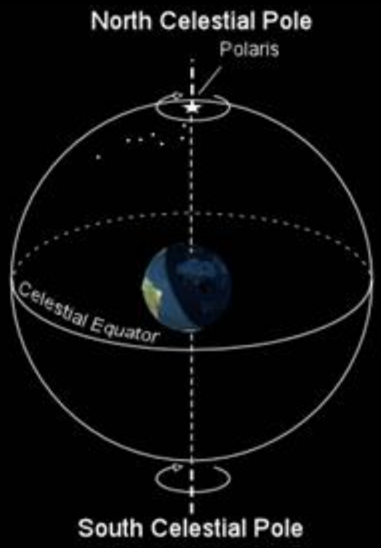
BACKGROUND & PROBLEM STATEMENT

Most Astrophotography Today

+

=

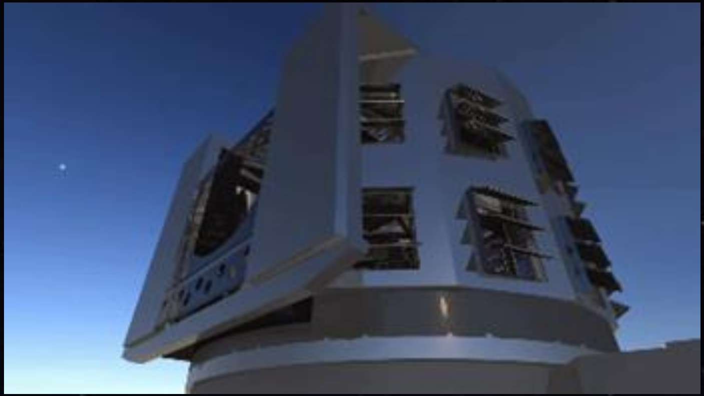
Milky Way Photography Post Processing Tutorial (Lightroom, Photoshop & Starry Landscape Stacker)


North Celestial Pole
Polaris
Celestial Equator
South Celestial Pole


Polaris
Celestial
North Pole

Time lapse and star trails

Star Trackers: in observatories, telescopes and cameras



Vera C. Rubin Observatory rotating dome (youtube.com)

Telescope tracking timelapse (youtube.com)

Move Shoot Move Nomad Star Tracker Stress Test. Over 9 Hours Analyzed (youtube.com

Image captured with (left) and without (right) an automatic star tracker

But, star trackers are not always accessible to amateur astrophotographers

Benro Polaris ($999)

Sky-watcher Star Adventurer (5.7 lbs/2.6 kgs)

Solution: Computational Deblurring of Long Exposure Images

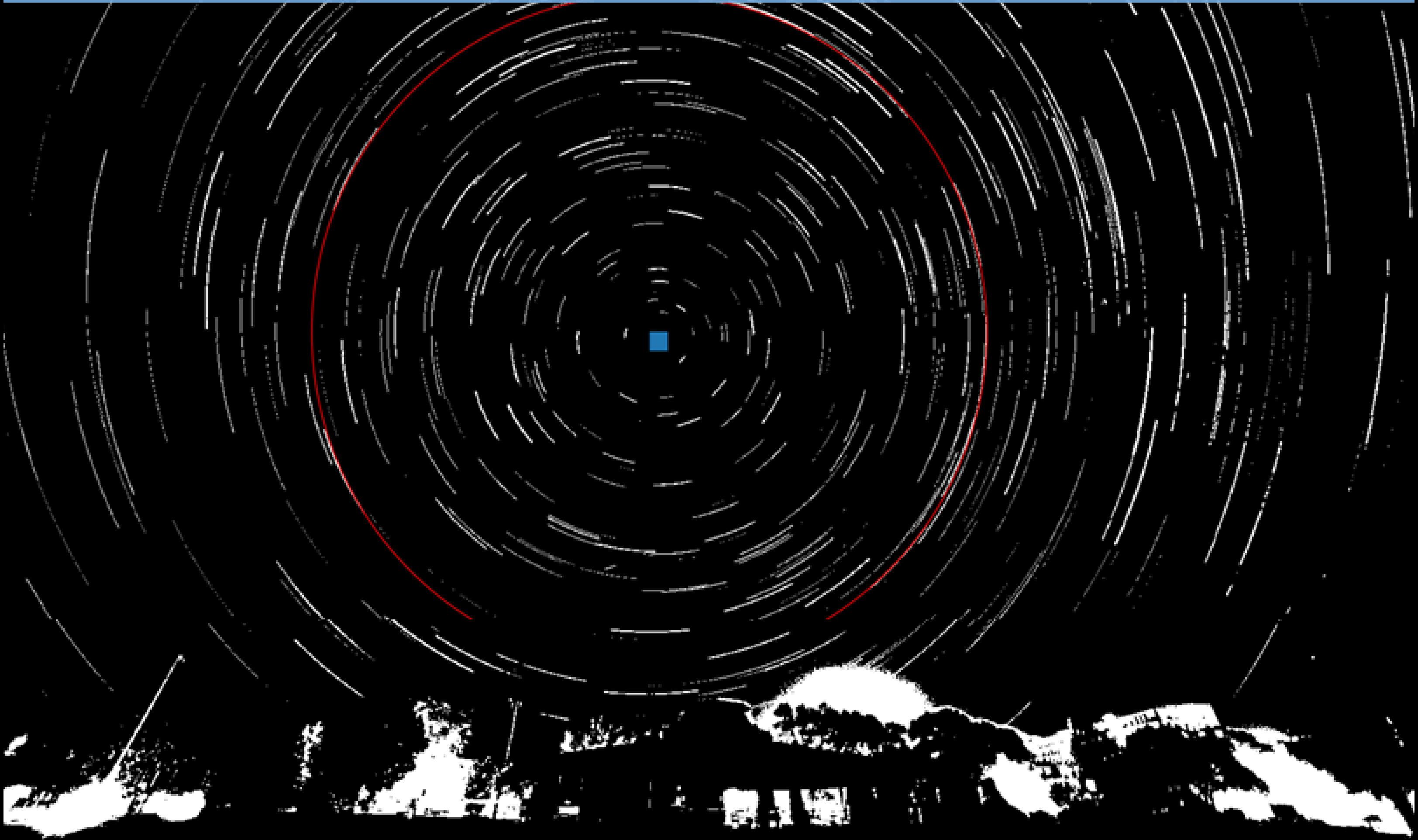
02

DATA COLLECTION, GENERATION & PREPROCESSING

Finding the center of rotation

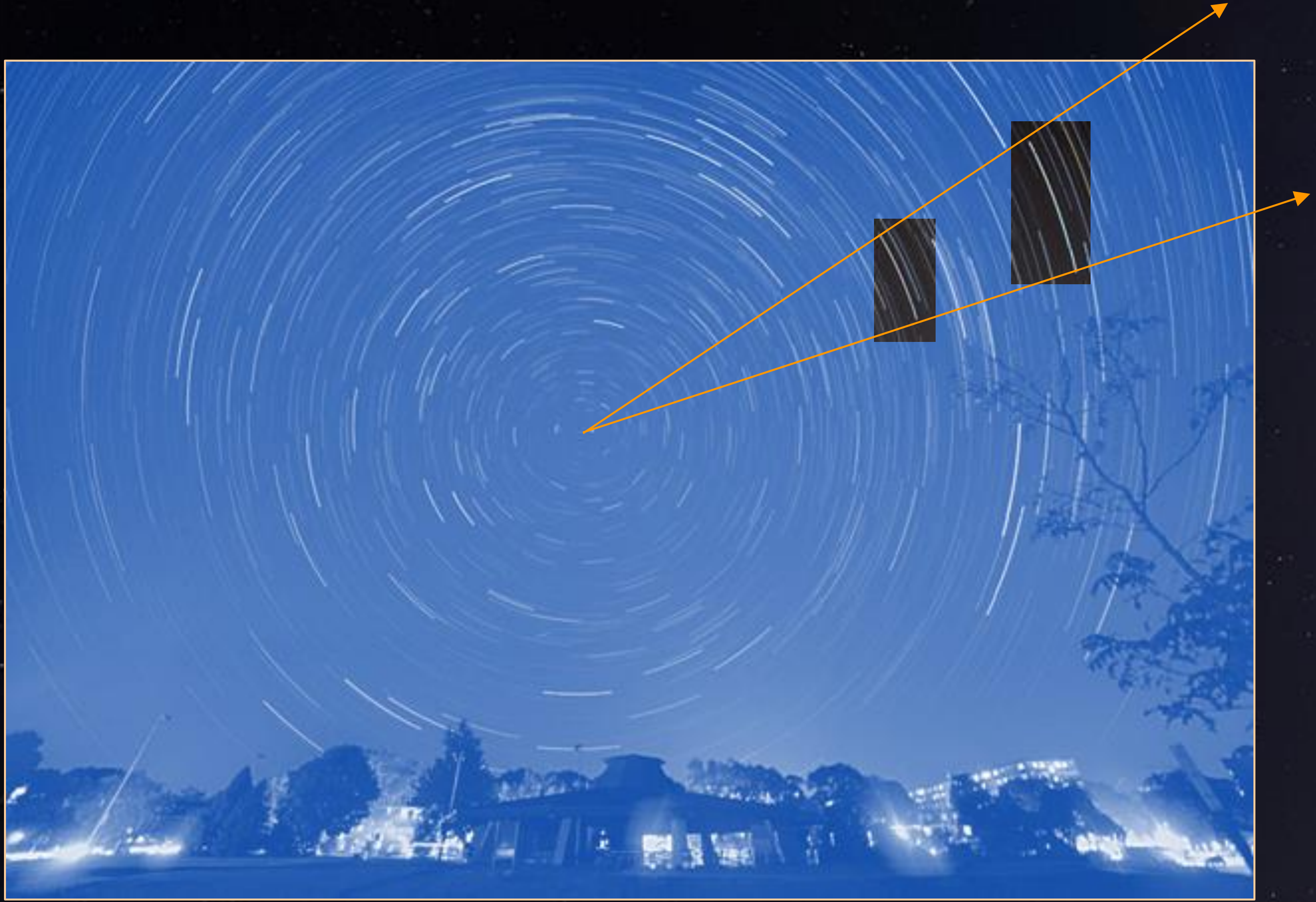
Center

Center of rotation detection using Hough transform



Closer Look

All stars travel the same angular distance



Convert to Polar Coordinates

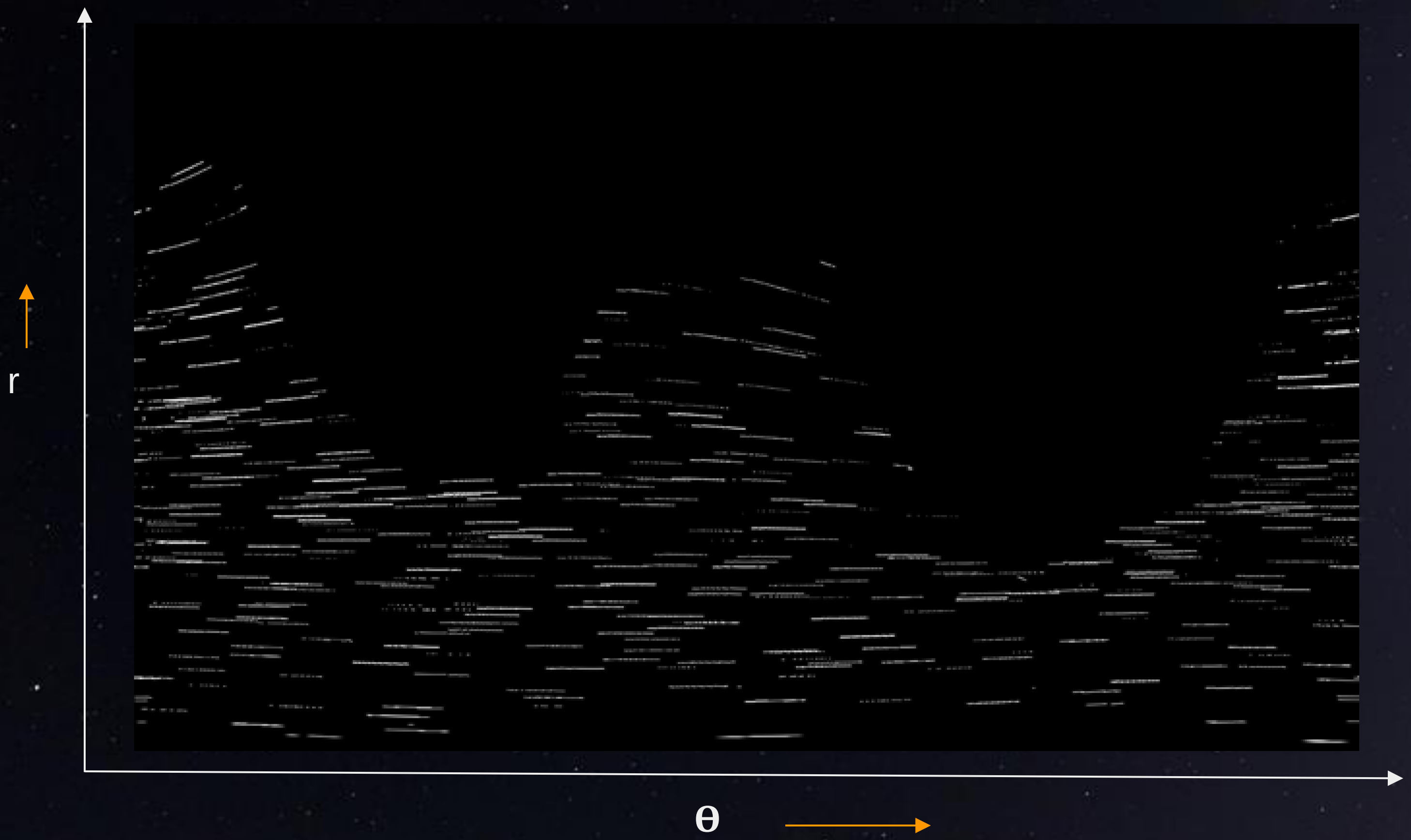
Setup a polar coordinate system with either celestial North/South pole as the center.

Convert to Polar Coordinates

Convert to Polar Coordinates

Convert to Polar Coordinates

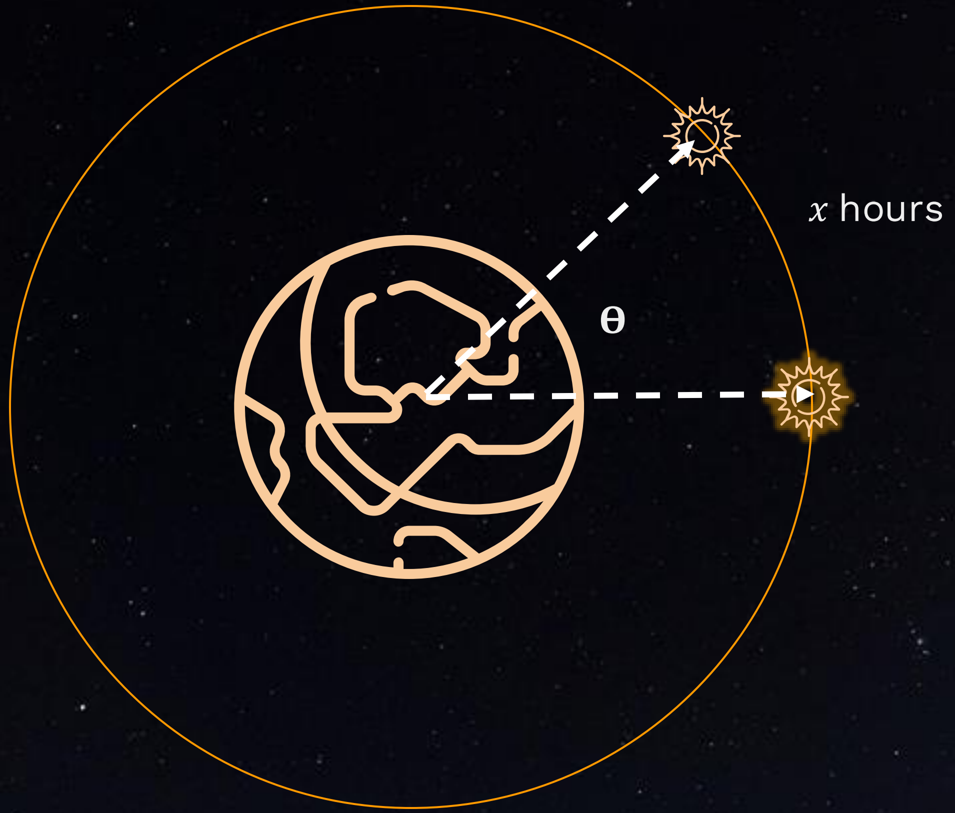
Final Polar Image



Finding the De-blurring Kernel

360 deg → 24 hours

Finding the De-blurring Kernel



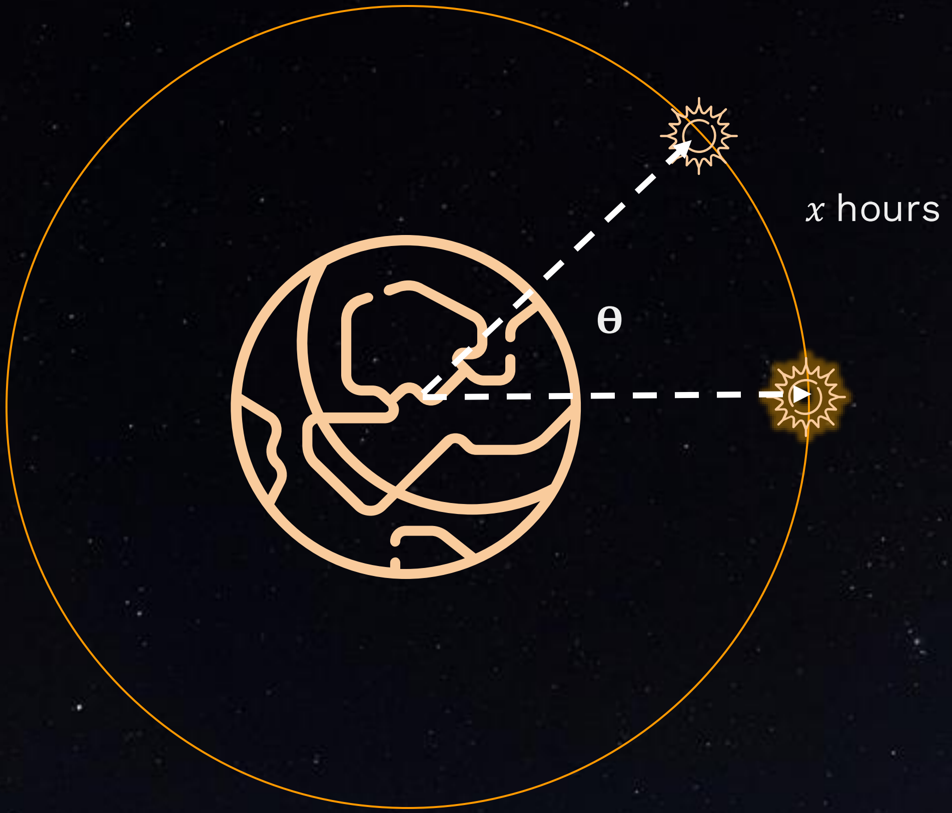
360 deg → 24 hours

θdeg → x hours

$$\theta = \frac{x \ * 360}{24}$$

$$\theta = 15x$$

Finding the De-blurring Kernel



360 deg → 24 hours

θdeg → x hours

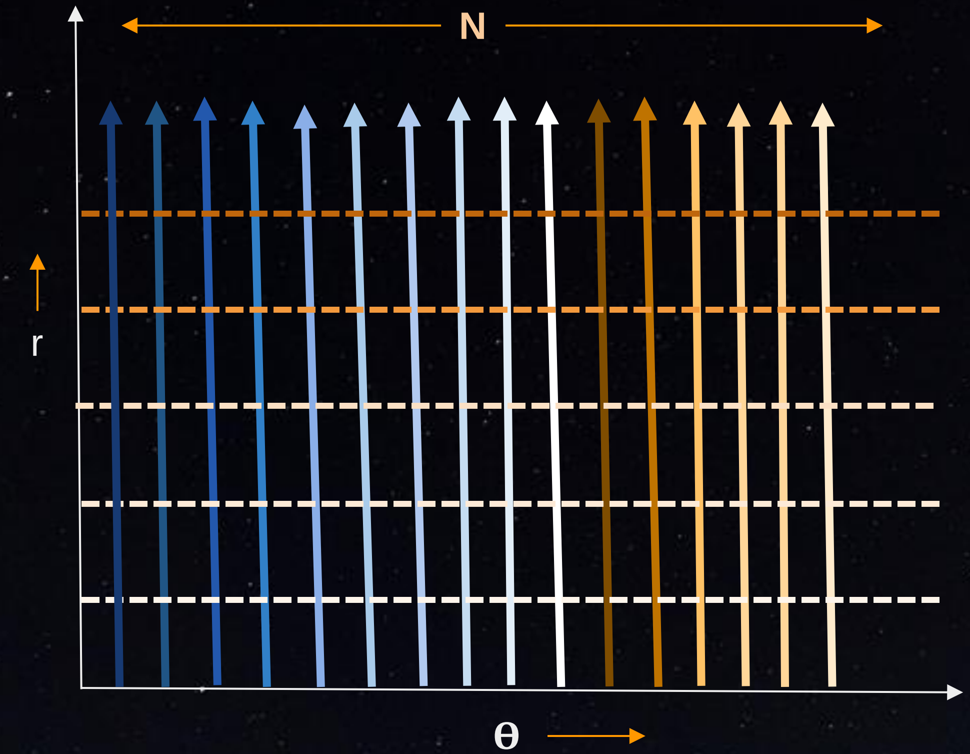
$$\theta = \frac{x \ * 360}{24}$$

$$\theta = 15x$$

$i.e,$ **1 hour** of exposure equals **15 deg** on the image.

In other words, **1 deg** is subtended in **4 minutes**.

Finding the De-blurring Kernel



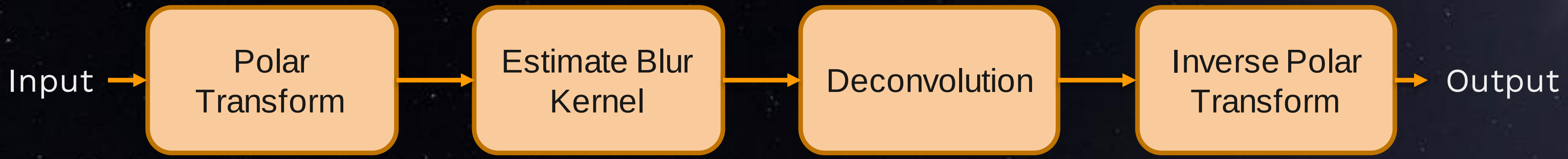
Blur Kernel

360 deg → N

θ deg → ?? (say L)

$$L = \frac{N}{360}\theta$$

Pipeline



Input
Polar Transform
Estimate Blur Kernel
Deconvolution
Inverse Polar Transform
Output

We can use data collected from public, image sharing sites

19 minutes

30 minutes

60 minutes

80 minutes

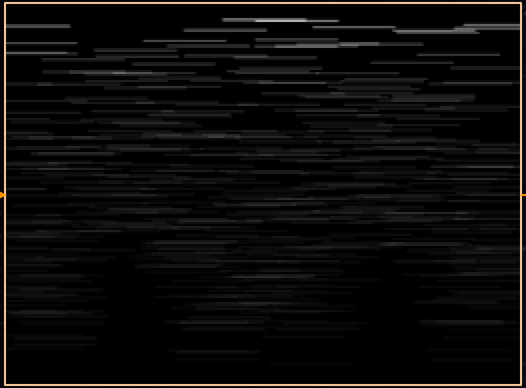
Image Credits: Adrian Mag and Kenny Eliason @ Unsplash

But these have no ground truth...

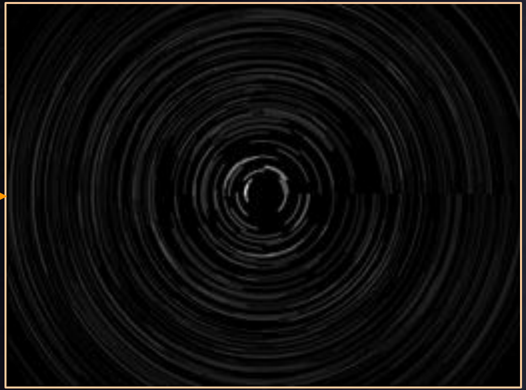
How can we validate our deconvolution algorithms?

Simulate Star Trails – Generate Our Own Ground Truth

Randomly generated 'stars'



Polar image, blurred with a known kernel



Cartesian image, rotated by a known number of degrees

Parameters that we can control in the forward model:

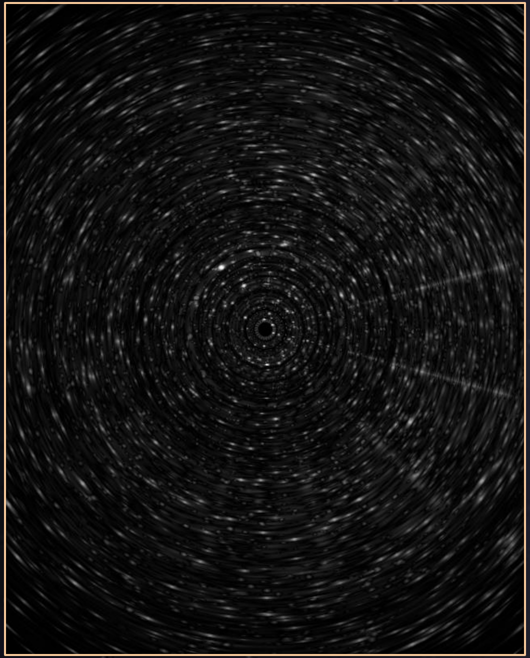
- Degree of blur / time of synthetic exposure
- Density of star trails
- Star intensity

03

DECONVOLUTION METHODS & RESULTS ON SIMULATED IMAGES

Deconvolution Methods

Original



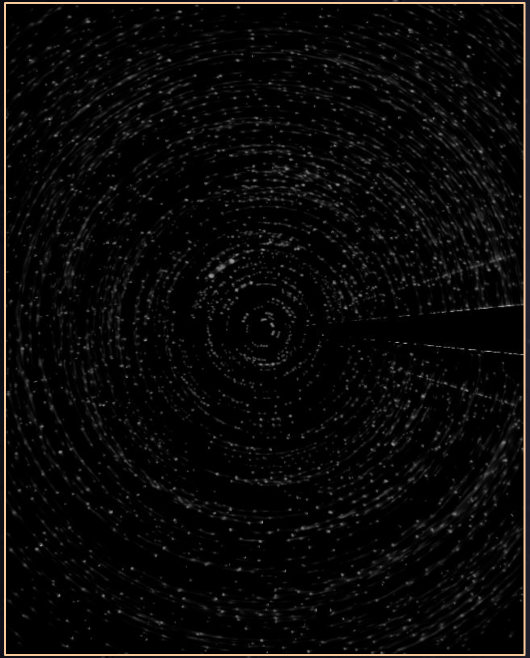
Wiener Deconvolution

Deconvolution Methods

Original

Wiener Deconvolution

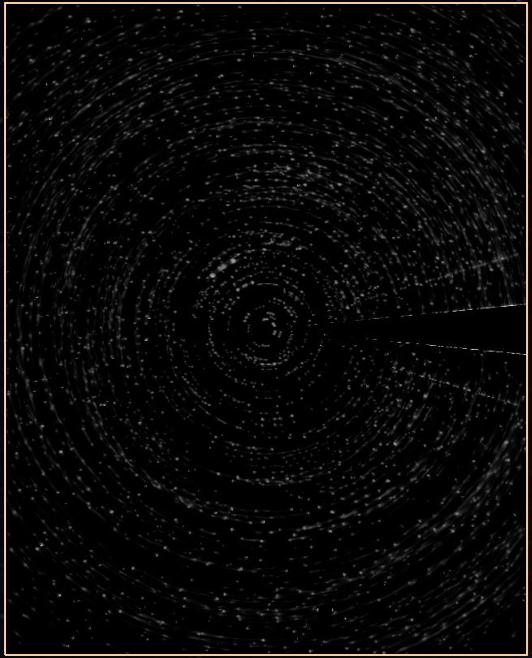
Best Method



Richardson Lucy

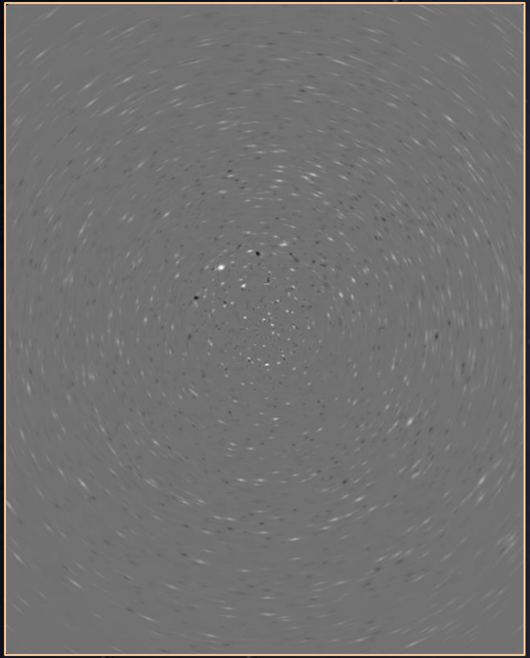
Deconvolution Methods

Original



Richardson Lucy

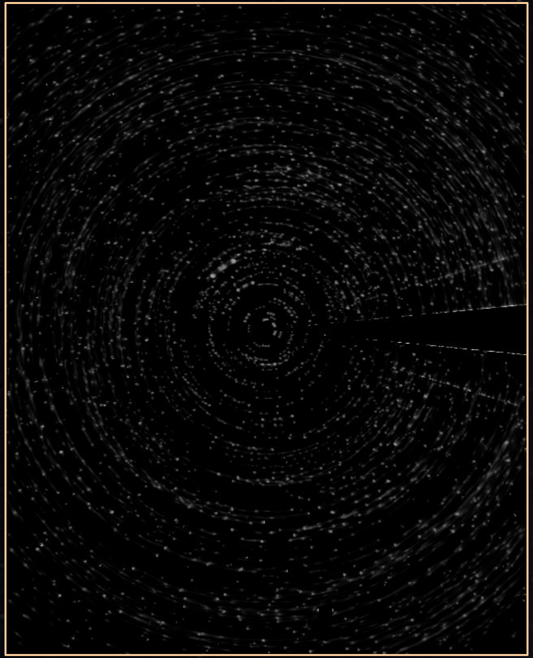
Best Method



Vanilla Gradient Descent

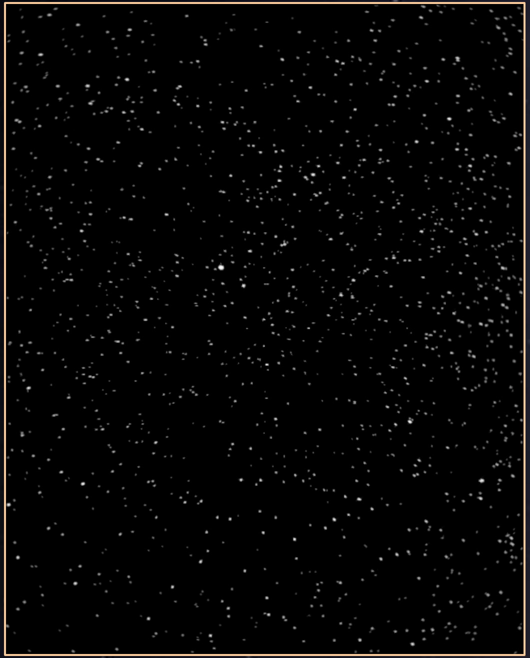
Deconvolution Methods

Original



Richardson Lucy

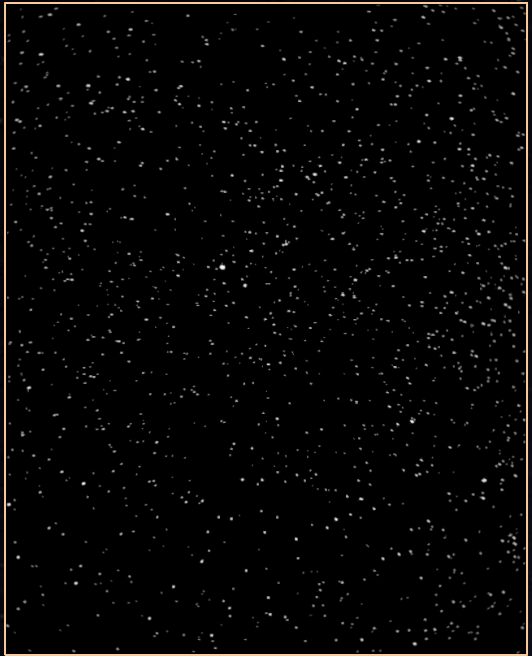
Best Method



FISTA

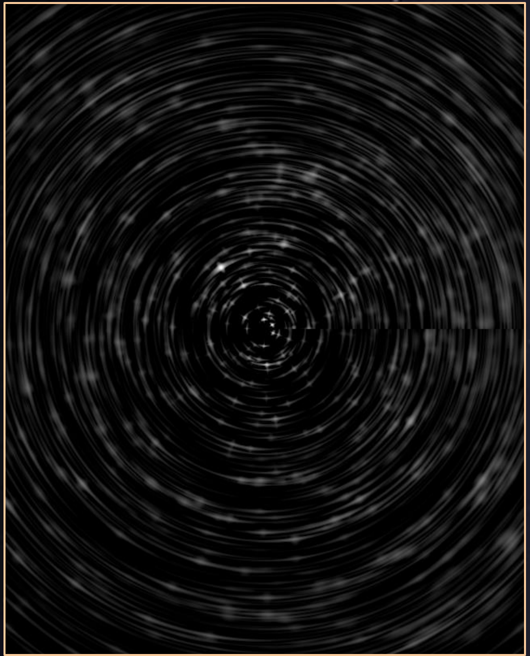
Deconvolution Methods

Original



FISTA

Best Method



FISTA + TV

Metrics

Peak Signal to Noise Ratio

PSNR is the ratio between the maximum possible power of a signal and the power of any noise that might distort it.

$$MSE = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M * N}$$

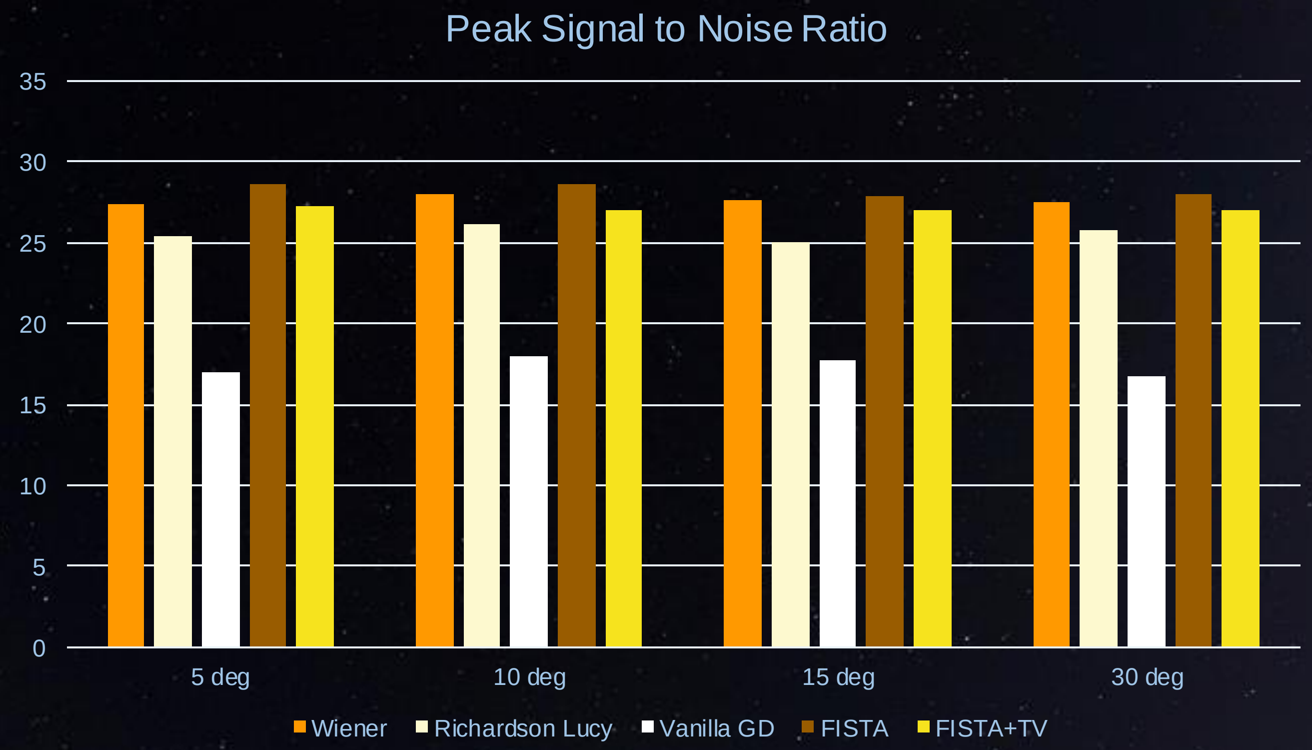
$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right)$$

Structural Similarity

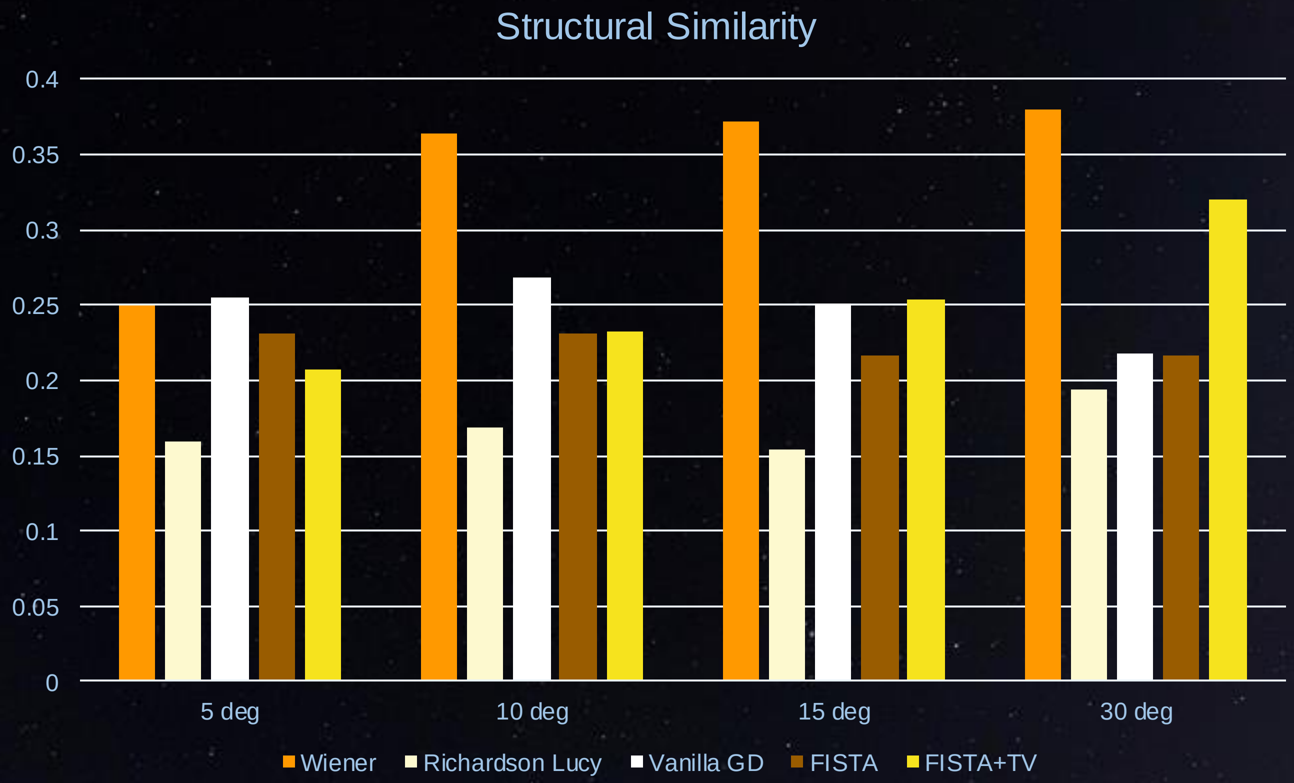
SSIM measures the difference between select properties (luminance, contrast and structure) of pixels in a pair of images.

$$\mathrm{SSIM}(I, I_w) = \frac{(2\mu\mu_w + C_1)(2\sigma(I, I_w) + C_2)}{(\mu^2 + \mu_w^2 + C_1)(\sigma(I)^2 + \sigma(I_w)^2 + C_2)},$$

Metrics


Peak Signal to Noise Ratio
35
30
25
20
15
10
5
0
5 deg
10 deg
15 deg
30 deg
Wiener
Richardson Lucy
Vanilla GD
FISTA
FISTA+TV

Metrics


Structural Similarity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
5 deg
10 deg
15 deg
30 deg
Wiener
Richardson Lucy
Vanilla GD
FISTA
FISTA+TV

04

RESULTS ON A REAL IMAGE

Deconvolution Pipeline

Polar image

Weiner deconvolution
Richardson-Lucy
Gradient Descent
FISTA

FISTA + Tikhonov
FISTA + TV
ADMM
ADMM + TV

Reconstructed stars

FISTA + TV worked the best

Image Recovery Pipeline

Input image

-

Dilated star trail mask

=

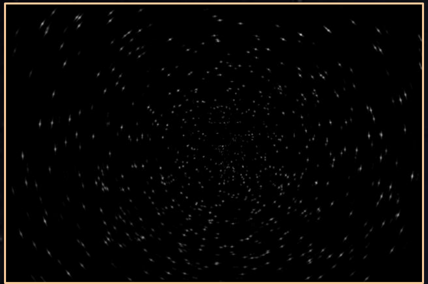
Image excluding star trails

In-painted star trails

Foreground mask
(generated during K-Means
clustering)

Gaussian blur (large sigma)
applied to in-painted image
(excluding foreground mask)

x



Reconstructed stars

=

Output image

Input image

Output image

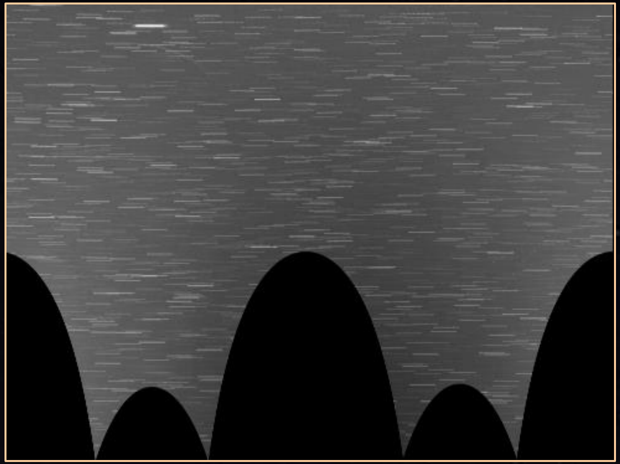
05

CHALLENGES & DRAWBACKS

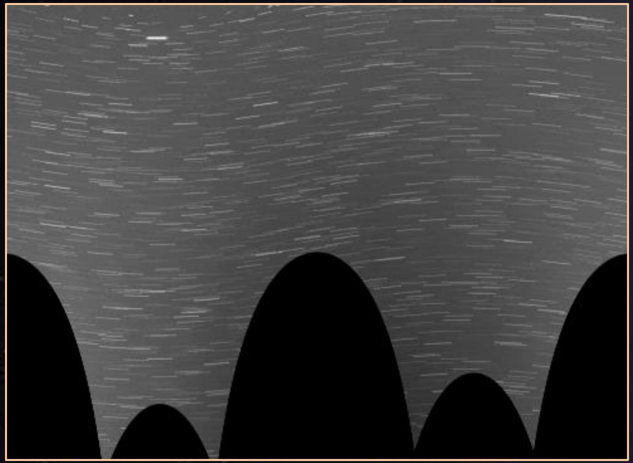
Challenge 1 - Finding the Center

Finding the center is very crucial and Hough transform is not always reliable.

Implemented alternate method which incorporates user feedback while finding the center.



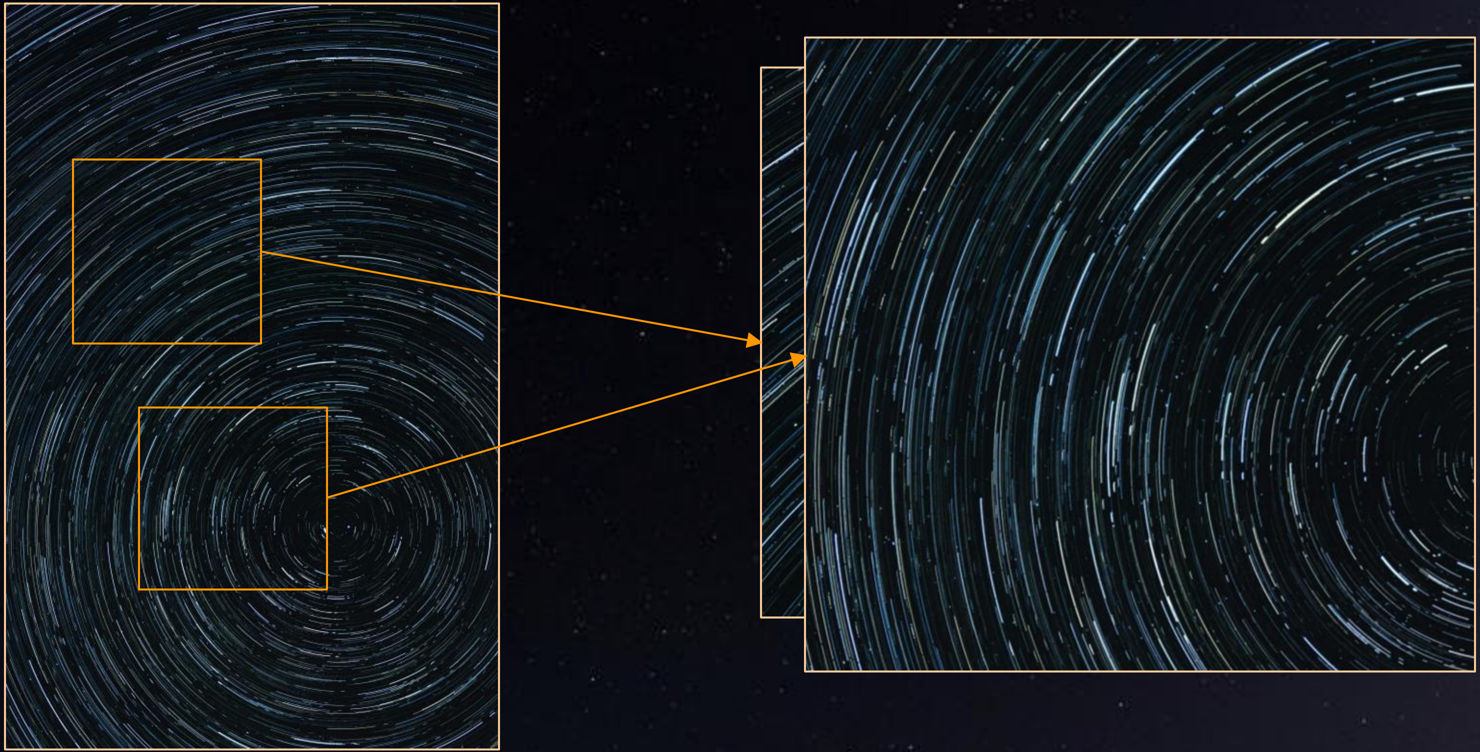
Correct Center



Incorrect Center
(Center off by 50px)

Challenge 2 – Finding “Good” Images

Most star trail images on the web are fusion of multiple short(er) exposure images.
They either do not have total time of exposure mentioned or have a rough estimate mentioned.
Also, most images have discontinuous trails because of time intervals between frames.



Drawback 1 - Reliance on Concentric Trails

Our method relies heavily on the assumption that the trails are concentric and centered around the North or South celestial pole.

Our method will not work for vortex (left) or path (right) type star trails.

Drawback 2 - Poor Performance on Dense Trails

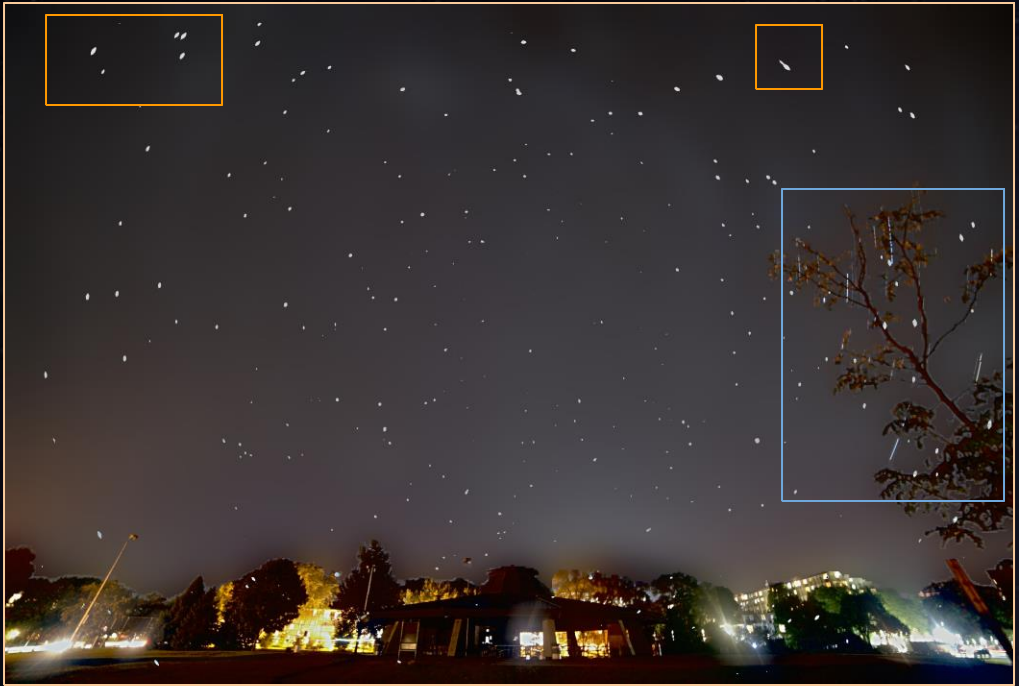
The K-Means clustering algorithm used to separate trails from the foreground during preprocessing performs poorly on dense / overlapping trails, making them difficult to deconvolve using our method.

Dense trail input image (left) and corresponding clustered output (right)

Drawback 3 - Partial Retention of Star Trails

Reasons:

- Fine foreground objects --> inaccurate K-Means clustering --> inaccurate mask generation
- Deconvolution artifacts that look like star trails/motion blurs



Thank You

Questions?