Kernel Based Moving Object Detection (KB-MOD)

A. Becker, D. Halperin, B. Howe, A. Whitaker

UNIVERSITY OF WASHINGTON

NSF

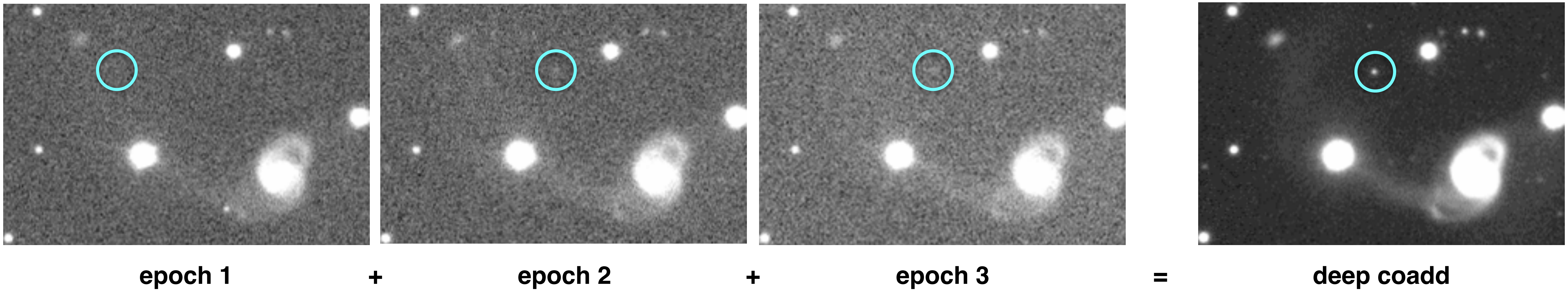
Searching for Faint Objects in Imaging Data

Image(x,y) = Signal(x,y) + Noise(x,y)

“Faint” means low signal–to–noise ratio (S ~ N), i.e. below the detection threshold (S ≫ N) in a single image

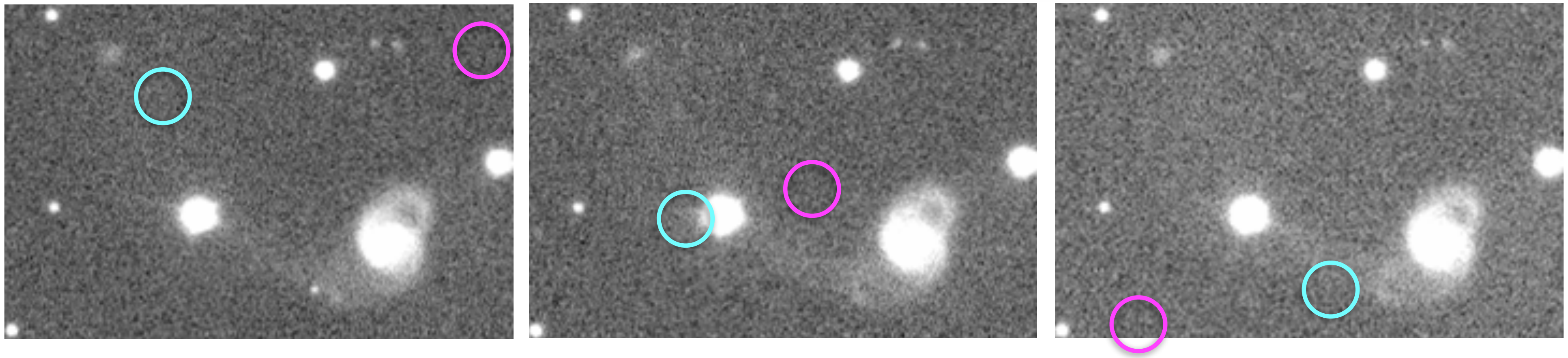
To accumulate signal on a given object, stack multiple images (which form a time-series)

S/N increases like sqrt(number of images)



For stationary objects, use a simple average

Searching for Faint *Moving* Objects in Imaging Data



epoch 1

epoch 2

epoch 3

In this case, a simple image coadd will not suffice

Requires an image “shift and stack” along motion vector

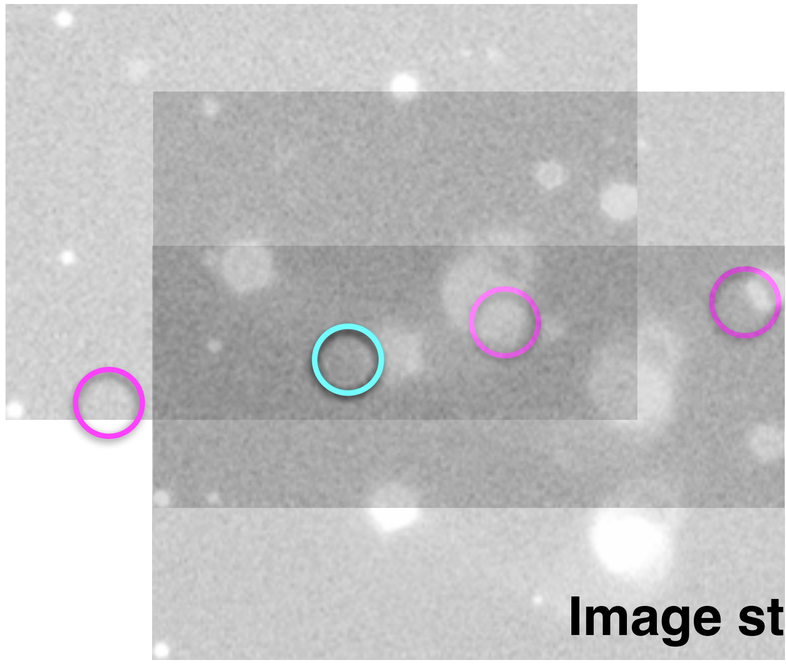


Image stack for cyan trajectory

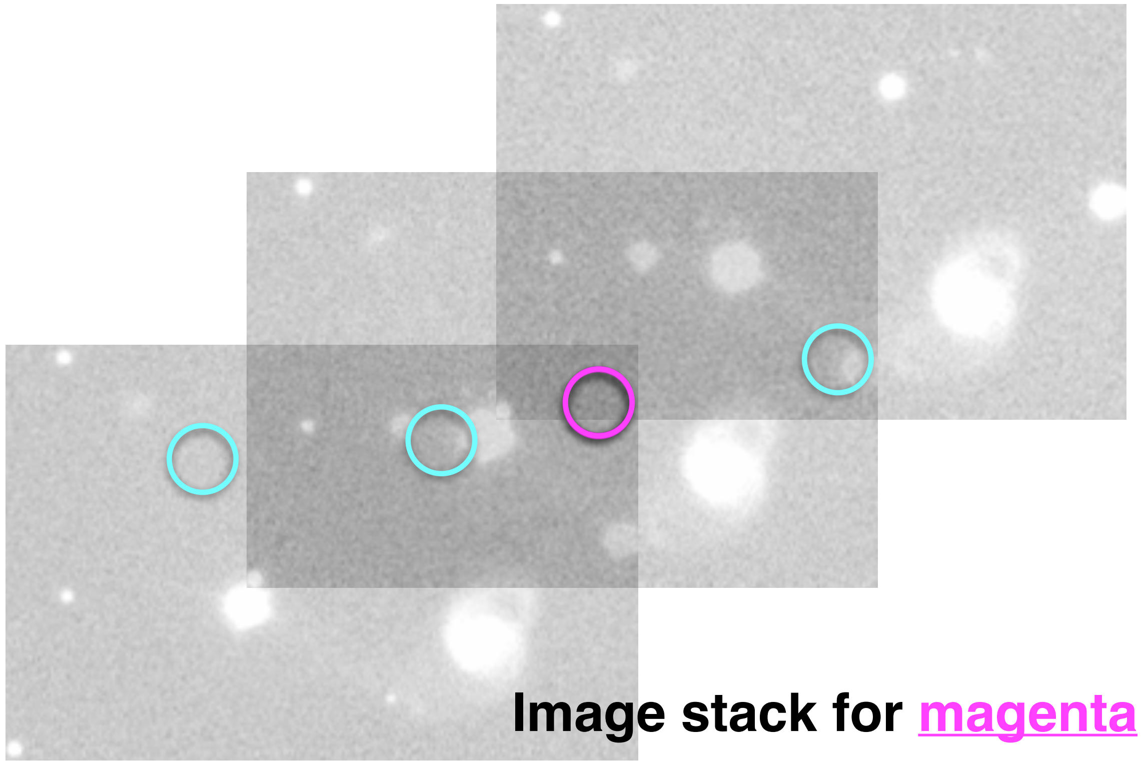
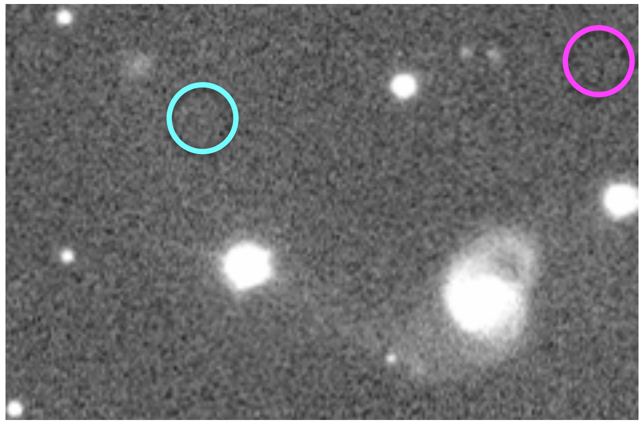
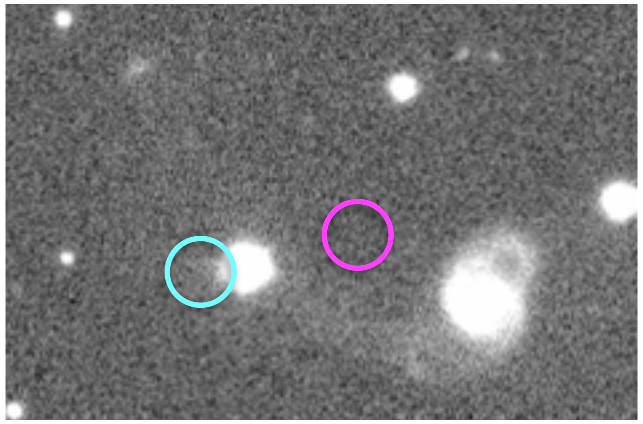


Image stack for magenta

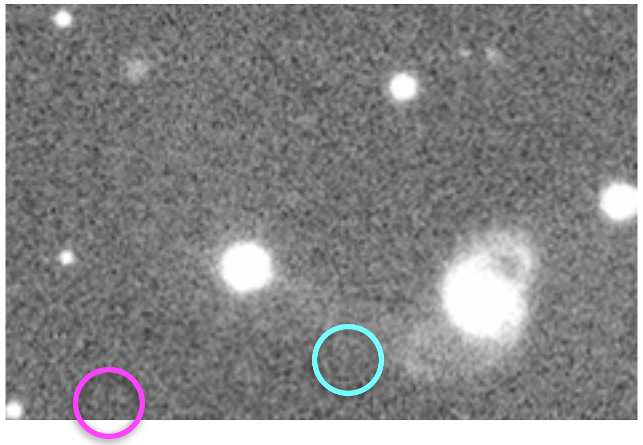
Searching for Faint *Moving* Objects in Imaging Data



epoch 1



epoch 2



epoch 3

In this case, a simple image coadd will not suffice

Requires an image “shift and stack” along motion vector

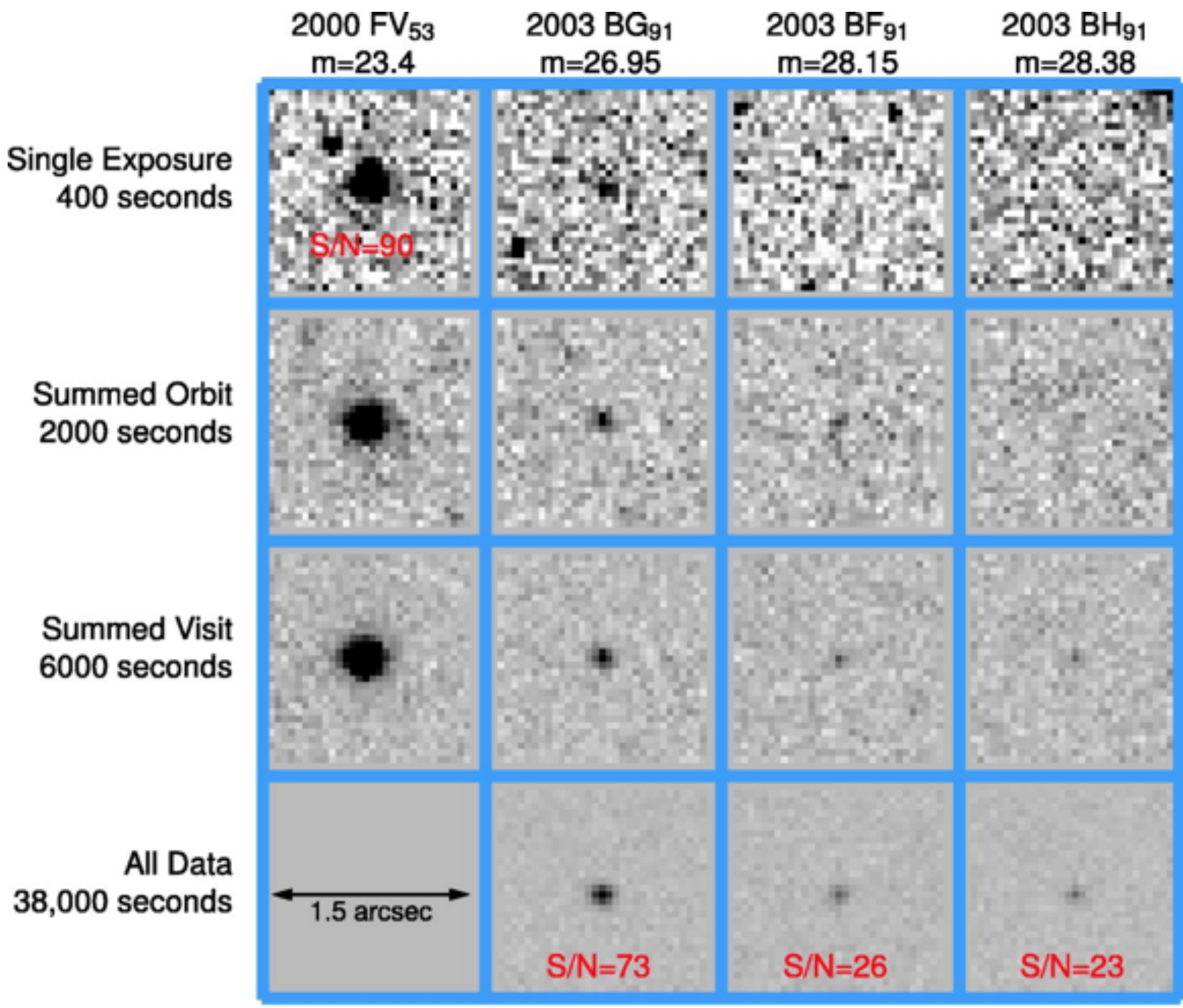
Every possible vector requires a different shift

For discovery, don’t know beforehand which vectors are populated

Which means you need to try them all!

Traditional “Shift-and-Stack” Techniques

- Bernstein 2004: Image coaddition along 10^{14} tracks. 3 new objects found.
- Must scale up detection threshold given the large number of “experiments”
- Limit analyses to slowest moving objects (assume linear motion)
- KB-MOD uses an improved order of operations to speed up technique by x100
- Answering the question “does a point source exist at this location” can be reduced to a single pixel access operation in an image (or a database)



Bernstein et al. 2004

KB-MOD Applied to Sloan Digital Sky Survey

SDSS Stripe 82 region

300 square degrees

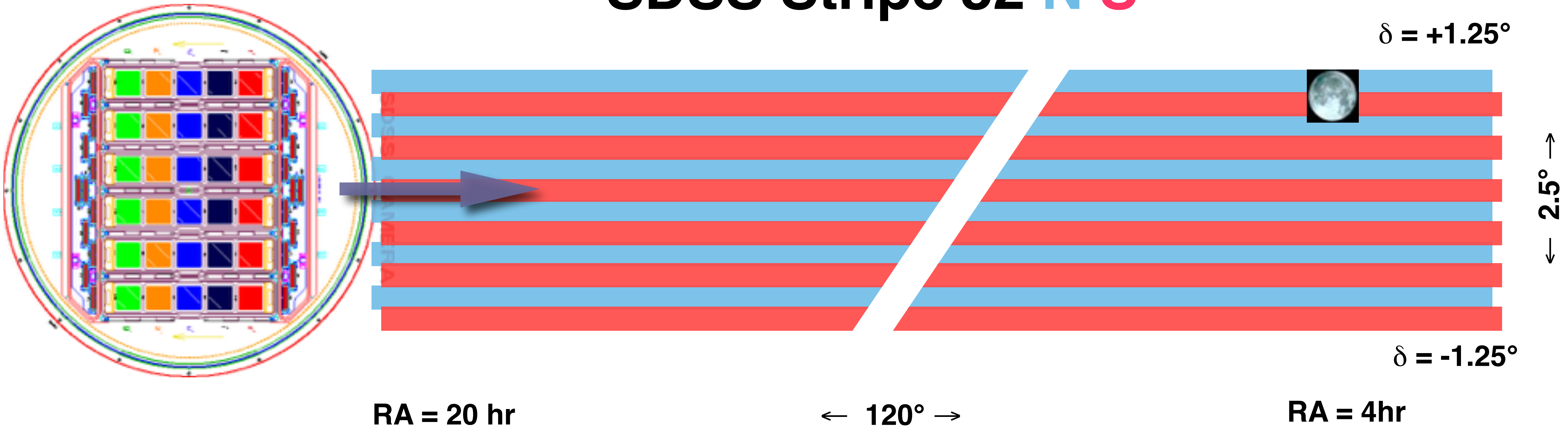
~70 epochs/position between 1998 and 2007

5-band photometry

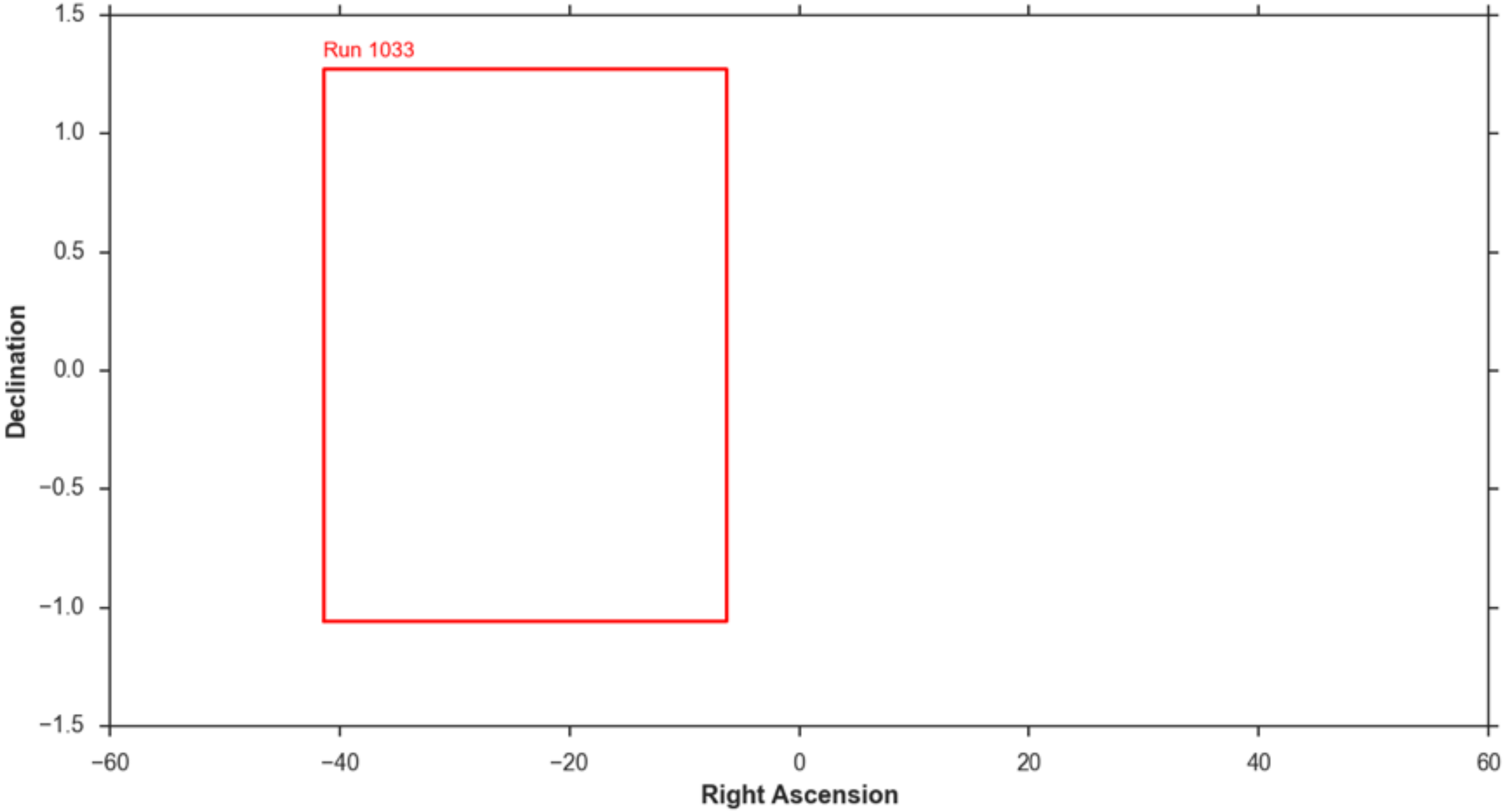
3 of which we will use (gri)

Previously (and successfully) mined for slow moving objects

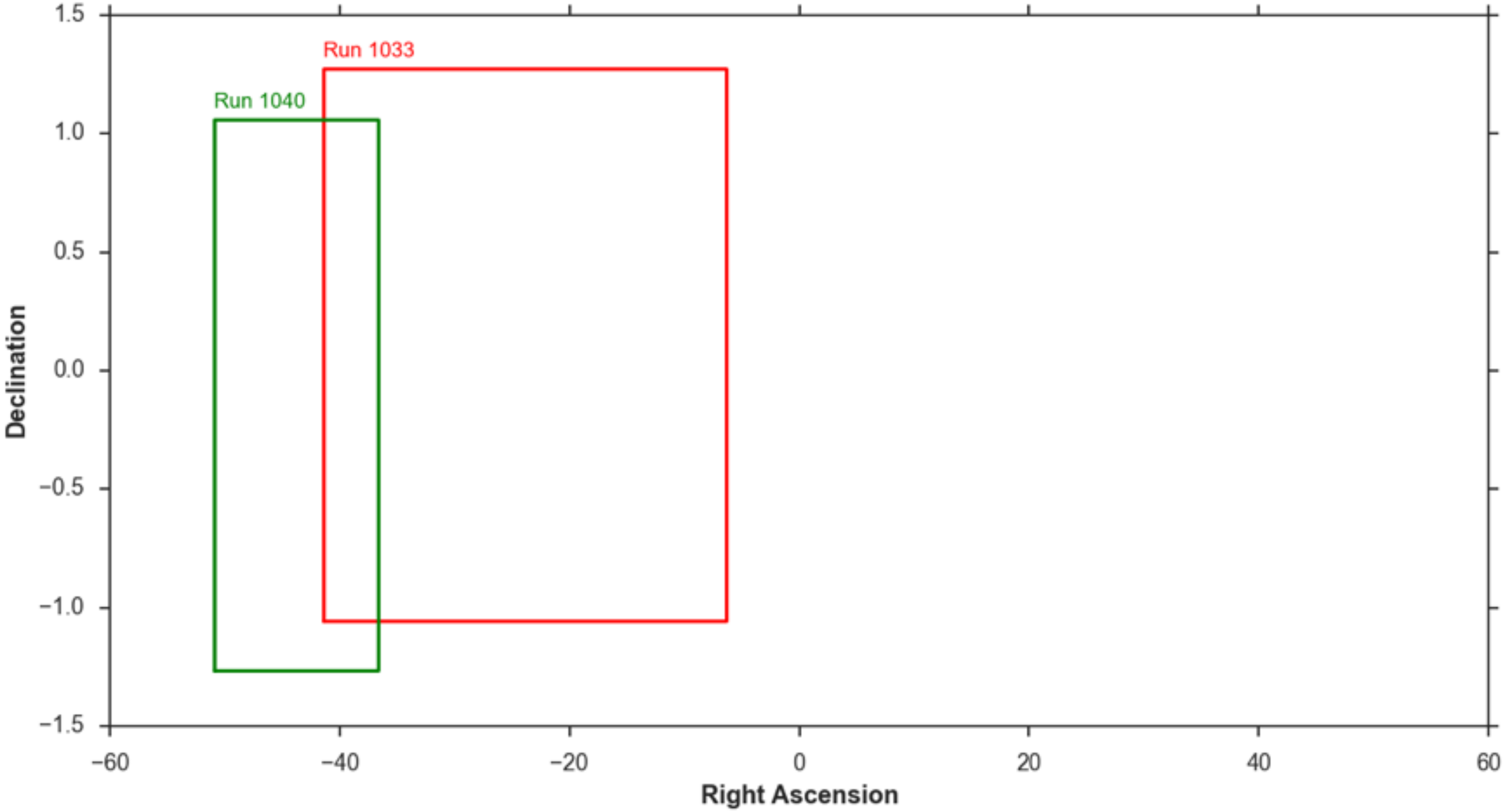
SDSS Stripe 82 N S



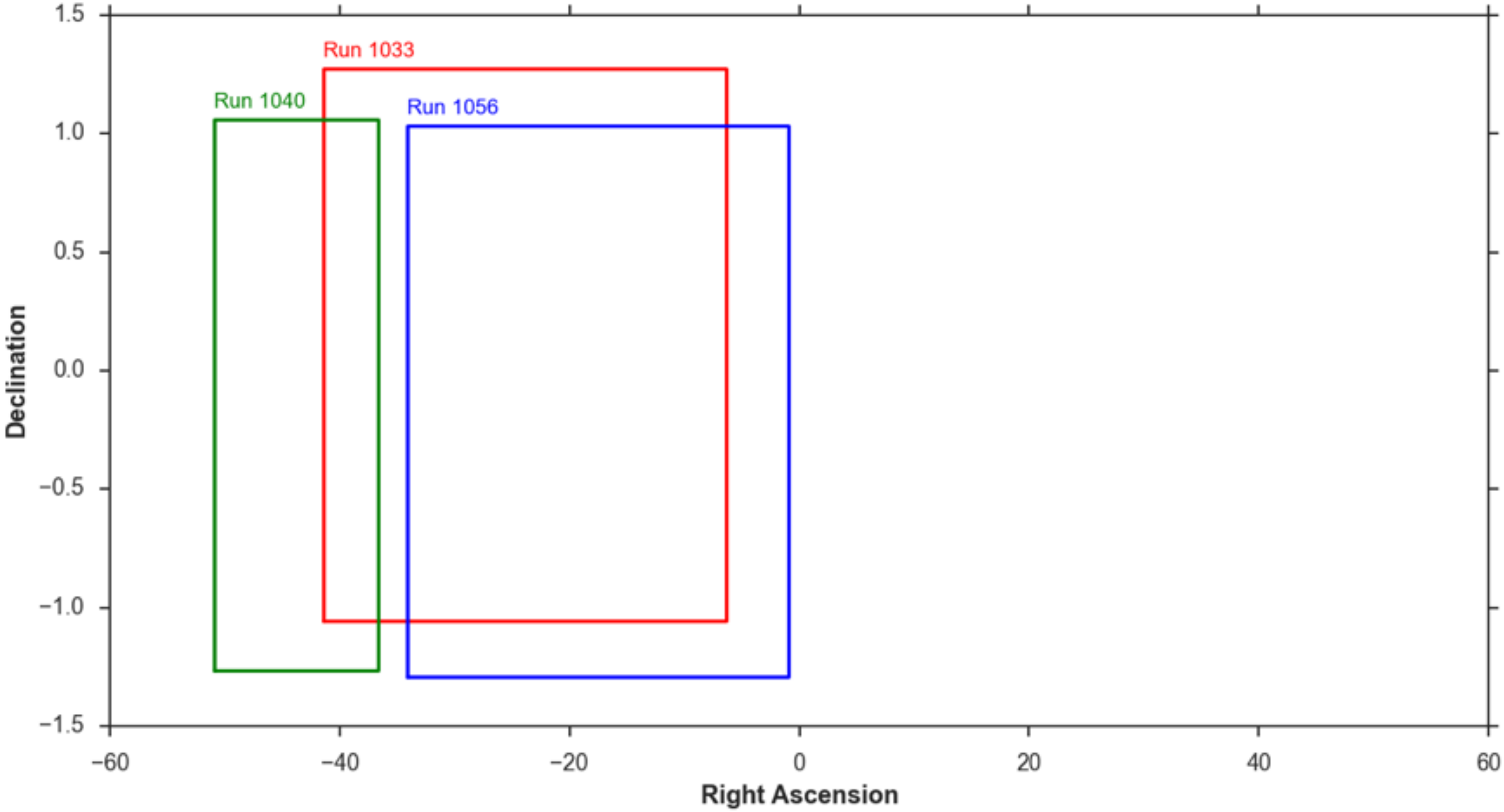
Spatial Structure in the Data



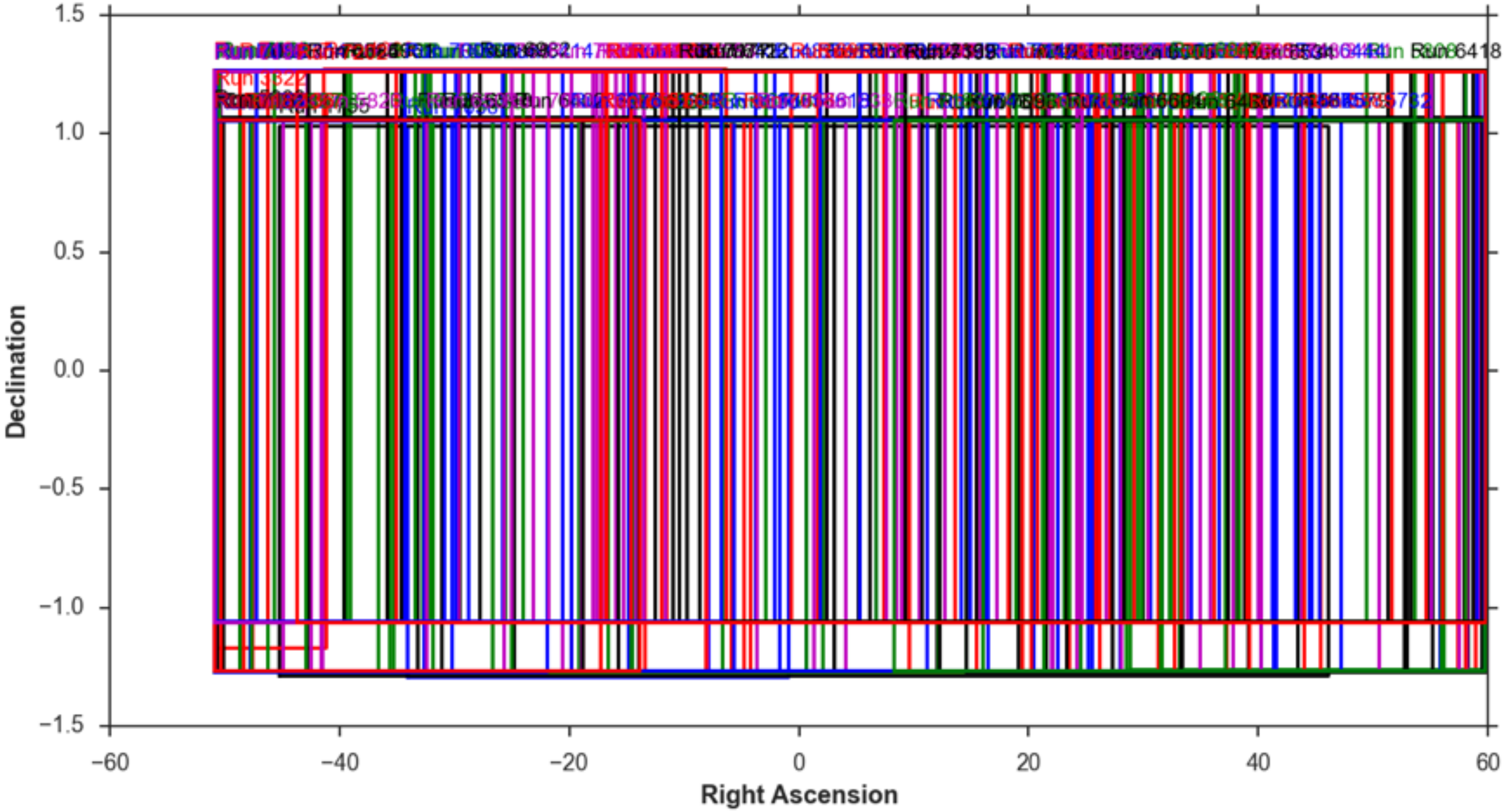
Spatial Structure in the Data



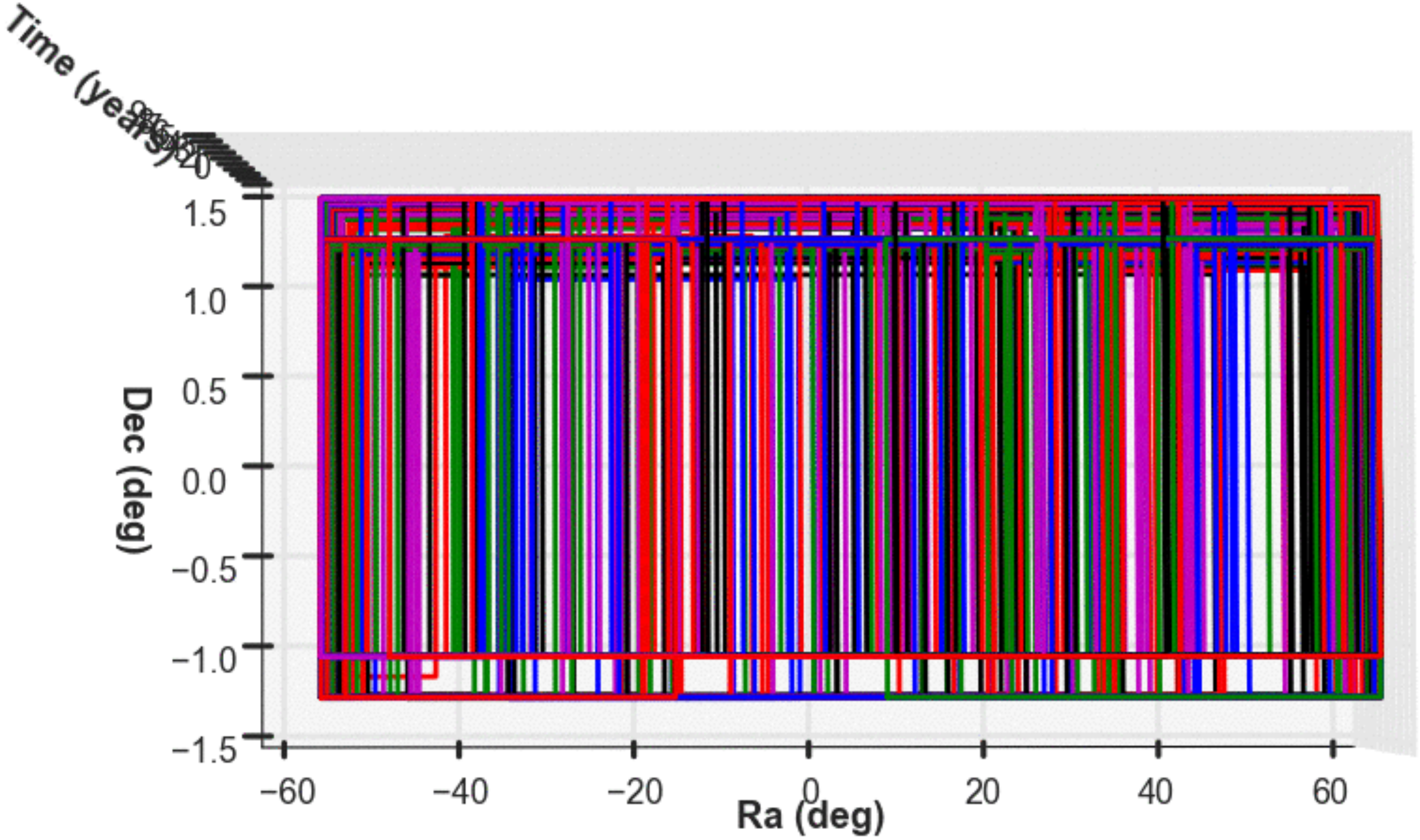
Spatial Structure in the Data



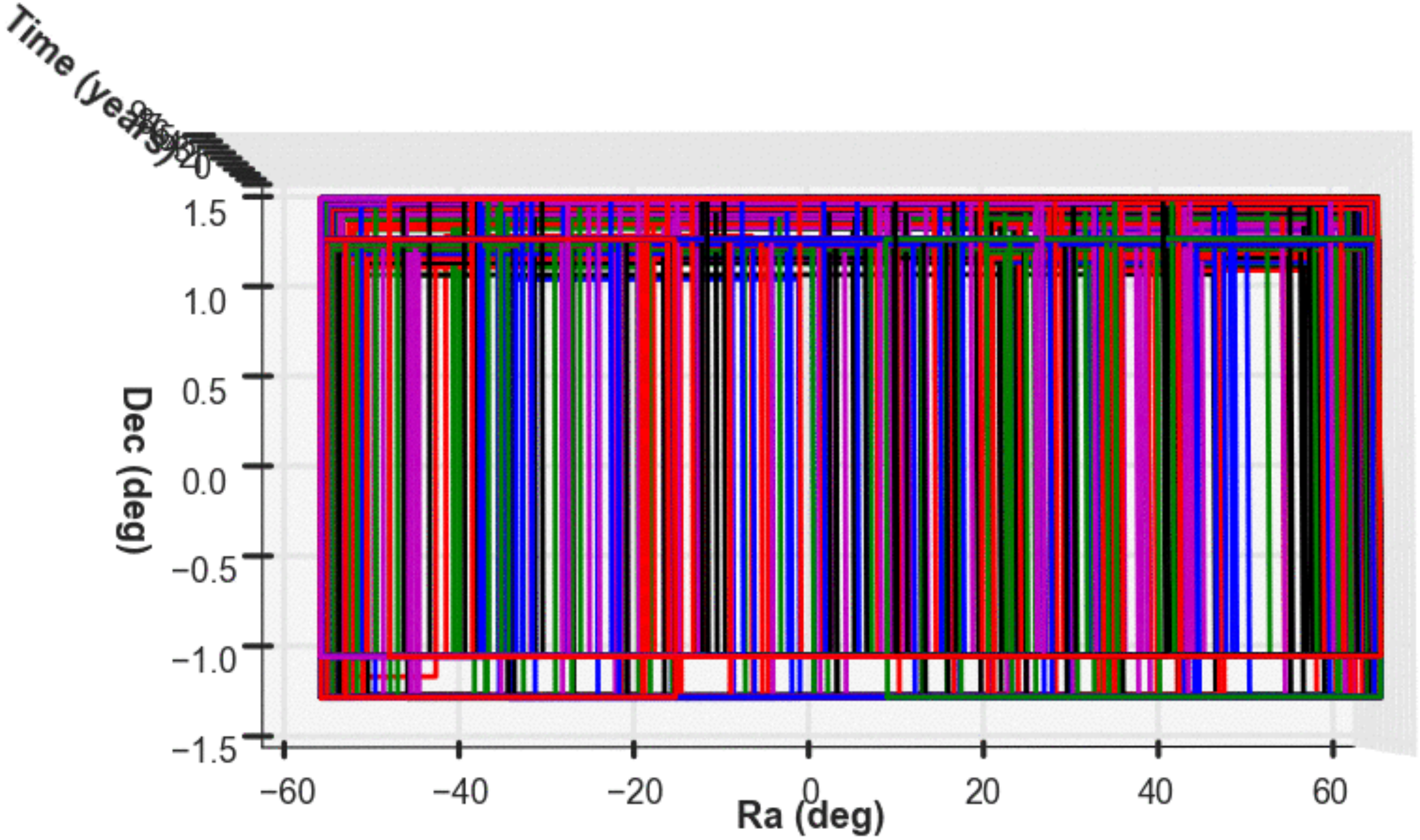
Spatial Structure in the Data



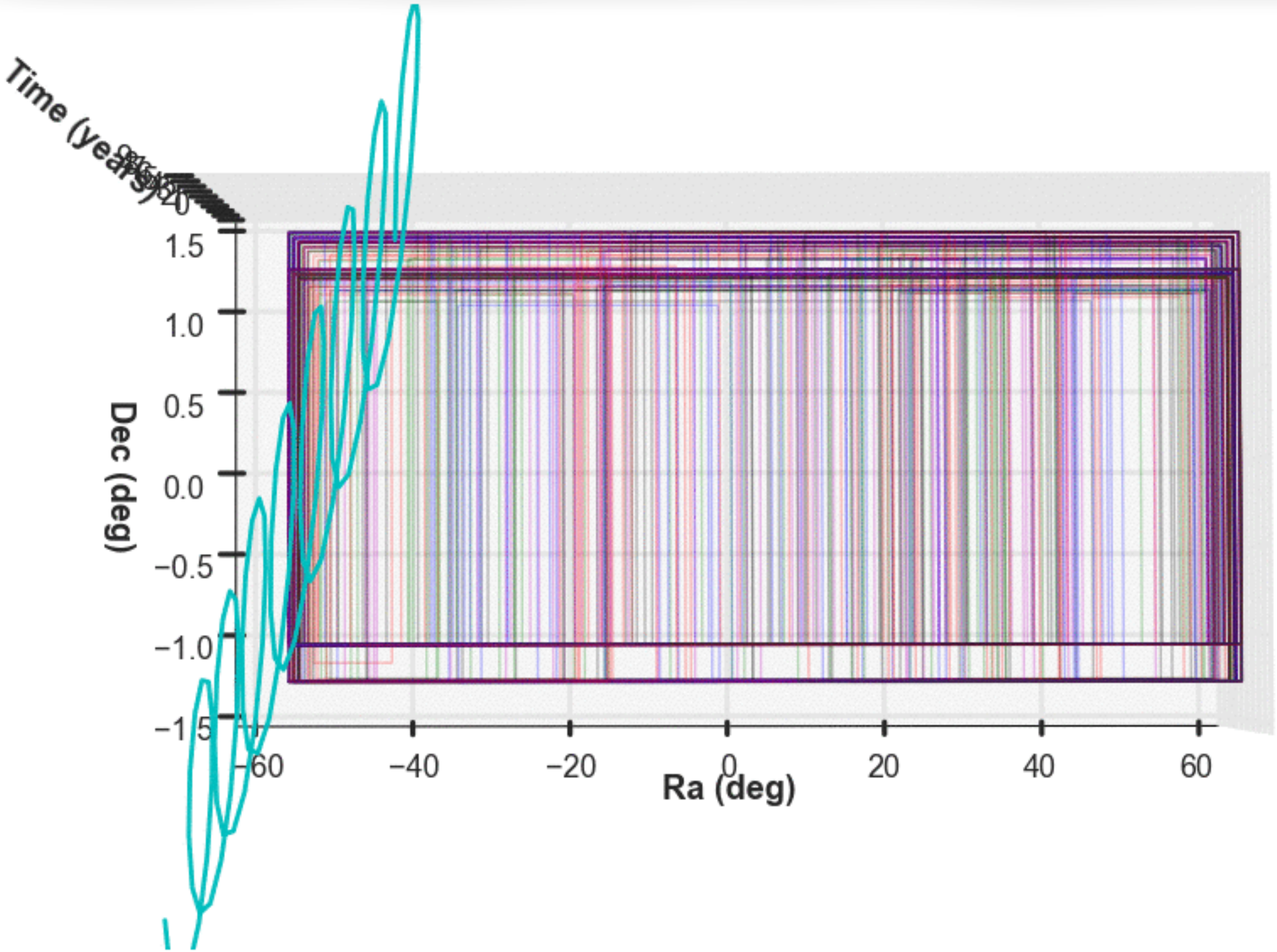
Temporal Structure in the Data



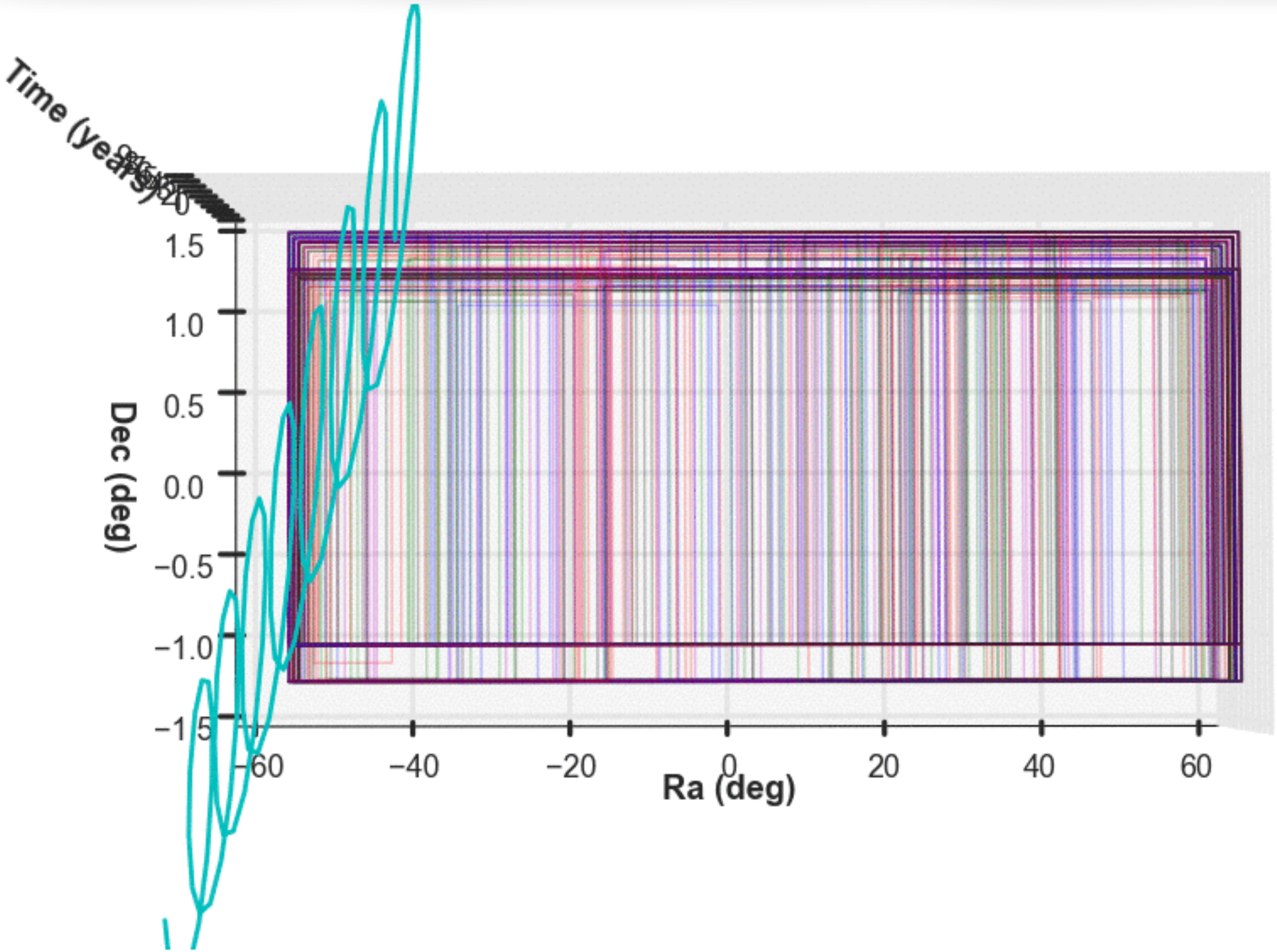
Temporal Structure in the Data



Goal: Rapid Intersect of Spatial-Temporal Trajectories with Imaging Data



Goal: Rapid Intersect of Spatial-Temporal Trajectories with Imaging Data



Database: Maximal Table Design

	Run (Night)	Set (Exposure)	Image (CCD)	Pixel
Primary Key	runId	setId	imageId	pixelId
Foreign Key		runId	setId	imageId
Spatial	bbox	bbox	bbox3d	integerIdx
Temporal	tmin,tmax	tmin,tmax		
Count	3E+02	5E+03	2E+06	5E+12

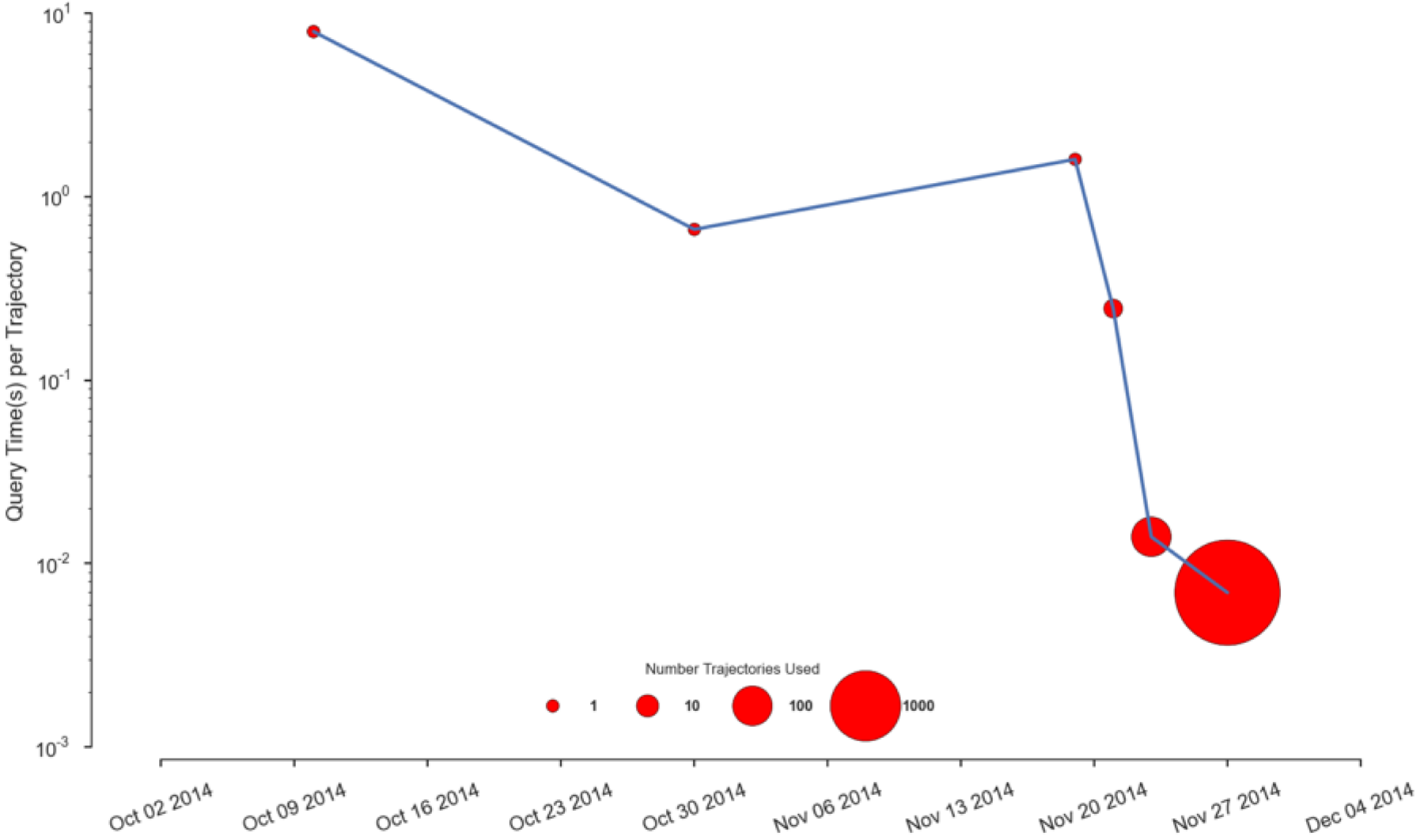
```
CREATE INDEX image_bbox3d_idx ON Image USING gist (bbox3d gist_geometry_ops_nd);

CLUSTER Image USING image_bbox3d_idx;
```

Database: Minimal Table Design

	Run (Night)	Image (CCD)
Primary Key	runId	imageId
Spatial	bbox	bbox3d
Temporal	tmin,tmax	
Count	3E+02	2E+06

Pixel access happens directly from the images

Query Timing vs. Week of Incubator



Outstanding Issues & Future Work

Data ingest remains a significant issue

- 10^4 ingests/second means 2-3 years of ingest for 10^{12} pixels
  - Similar cost for clustering
- Parallel database a possibility (Greenplum)

Since this cost is driven by the Pixels table, perhaps we could leave the pixels out of the database.

- Requires fast on-disk pixel access at the image level
- Read an image once using the intersections from multiple trajectories

LSST will be 4 orders of magnitude more imaging data / pixels

- Scalability is key

To be continued…