CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

UNIVERSITY OF TORONTO

Astrometry.net

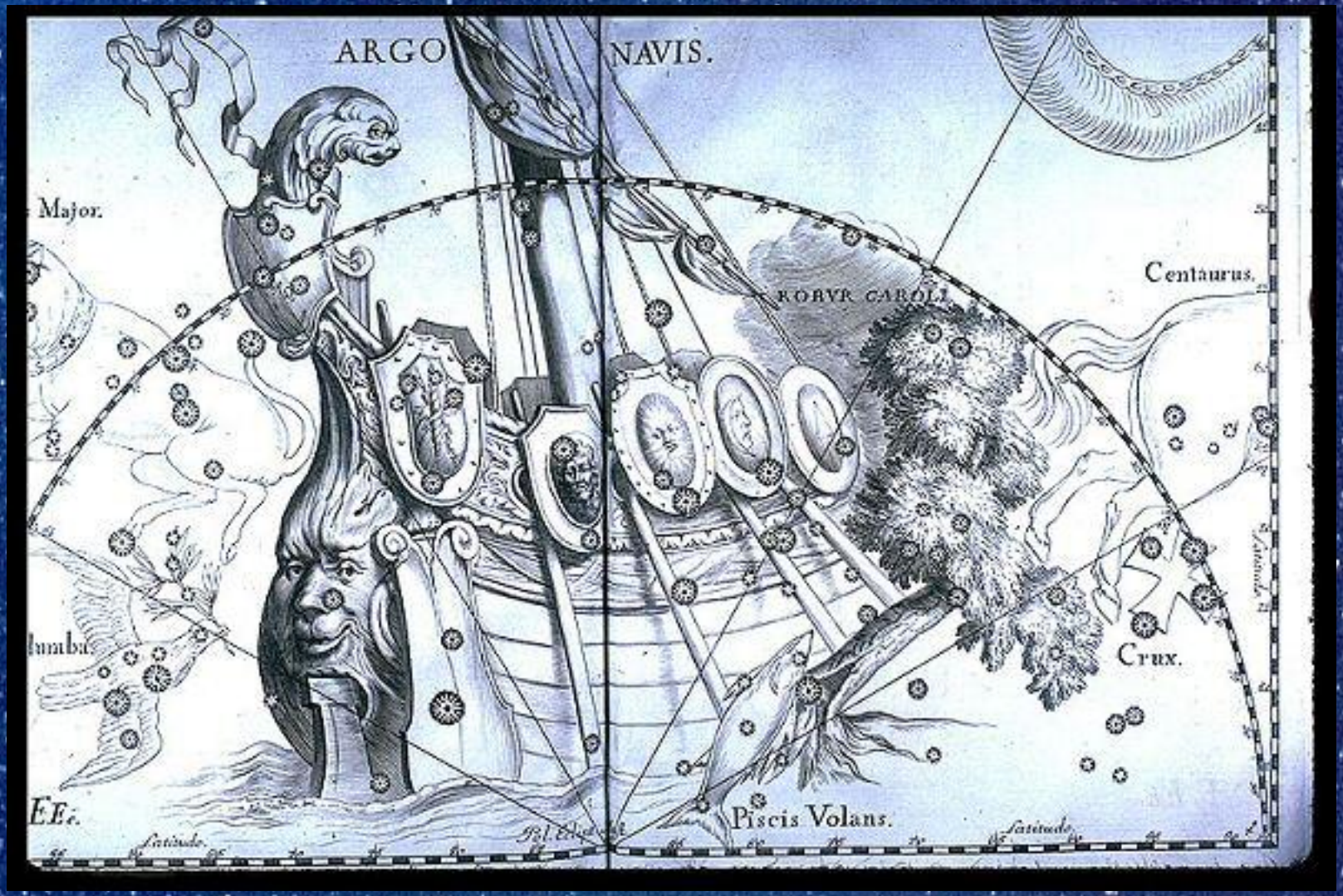
Real-time star pattern matching algorithms for pointing of suborbital and space telescopes

Lorenzo Moncelsi, School of Physics and Astronomy

outline

- the past
- today: in-flight requirements
- BLAST & others:
  - Lost In Space Pyramid (LISP)
  - Run-Of-Mill Algorithm (ROMA)
- the future(?): Astrometry.net

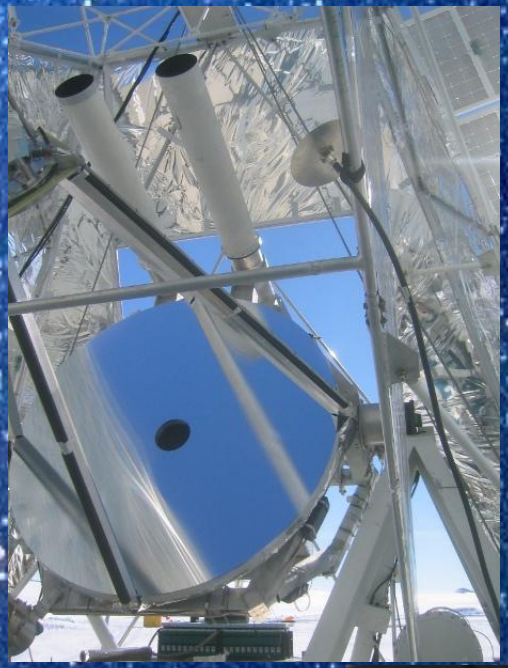
the past


ARGO NAVIS.
Major.
ROBVR CAROLI
Centaurus.
Crux.
Piscis Volans.

today: in-flight requirements

- pointing solution with accuracy < 5″ (1/720 deg)
- real-time pointing solution at 1 Hz
- completely autonomous (no human intervention)
- for balloon experiments: daytime float light conditions
- low power (~30W, your Wii) and Ethernet connection
- low performance computer: 500 MHz – 1GB RAM

star trackers/cameras



- 1.5 megapixel CCD
- 200 mm f/2 lens
- 2° × 2.5° field of the sky
- 600 nm red filter
- ~200 ms exposures
- sensitive to stars down to mag ~ 9

[100 times fainter than the faintest stars visible at night in an urban neighborhood with naked eye]

typical frame -___-

blob finder

- nothing fancy, but fast and reliable
- what we do:
  - mask off the hot/bad/dead pixels
  - find blobs (real space, not Fourier)
  - centroid them with sub-pixel accuracy (Gaussian fit)
  - output where the blobs are on the CCD (x,y)
  - sort them by intensity
- typical numbers: 2/3 up to ~20 blobs per frame

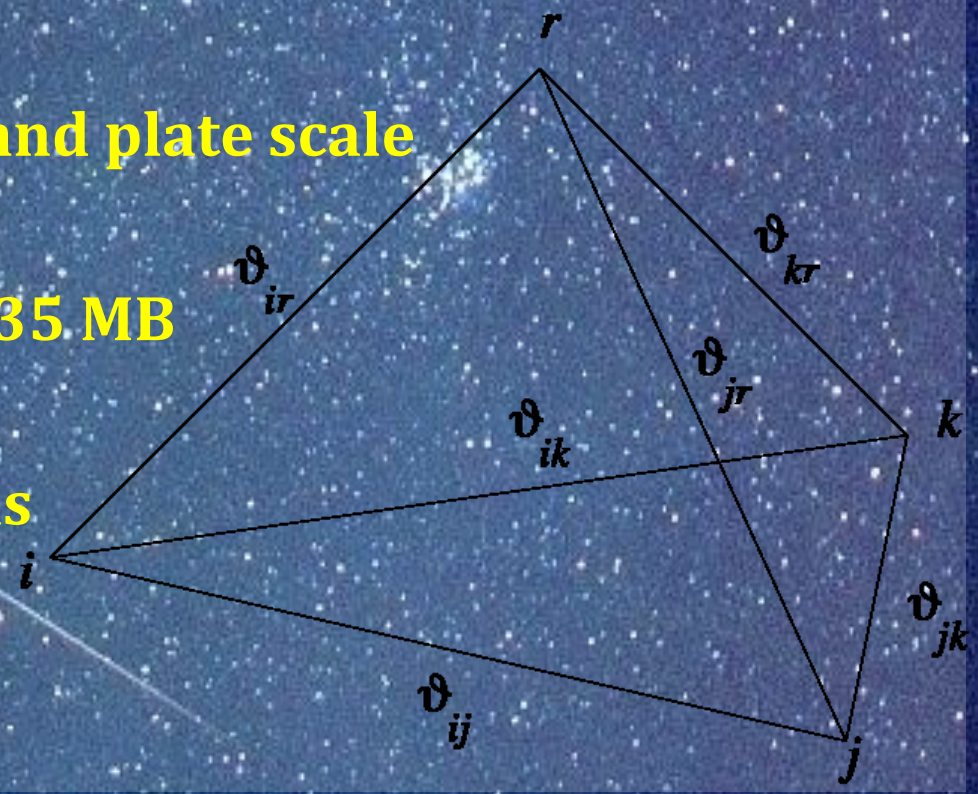
star pattern matching: I input and output

- **INPUT:**
  - **list of positions of blobs, sorted by intensity [required]**
  - **angular tolerance [required]**
  - **plate scale, i.e. deg/pixel [currently required]**
  - guess of recent pointing position [optional]
  - lat & long of telescope [optional]
  - rotation of the CCD about its axis [optional]
- **OUTUT: 4D parameter space**
  - **pointing on celestial sphere: (RA, DEC) or (AZ, EL)**
  - **rotation**
  - **plate scale**

star pattern matching: II
Lost In Space Pyramid (LISP)

overview

- the telescope is "lost", no idea where we are looking at
- we feed LISP with blobs and plate scale
- mag 8 all-sky catalogue: 35 MB
- search for 4 star polygons
- k-vector indexing

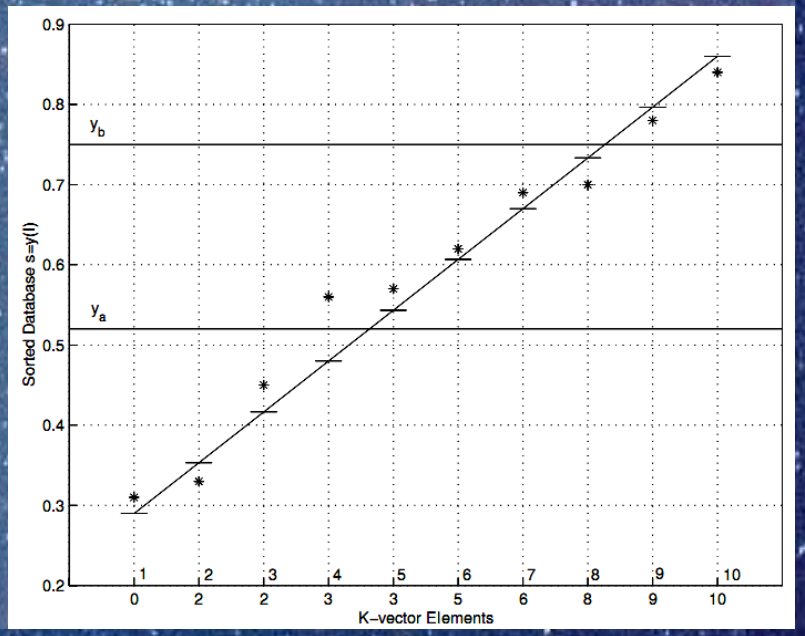


[Mortari et al. 2004]

star pattern matching: II
Lost In Space Pyramid (LISP)

indexing

- **build a structural data base with k-vector approach**
  - **sorted data array of interstar distances α for all star pairs admissible within the star camera FoV, over the whole sky**
  - **fit a slightly steeper straight line**
  - **given [α-t/2,α+t/2], search-less task to yield the 2 range indexes [k_1,k_2]**
  - **the k-indexing allows easy access to the master catalogue**



[Mortari et al. 2004]

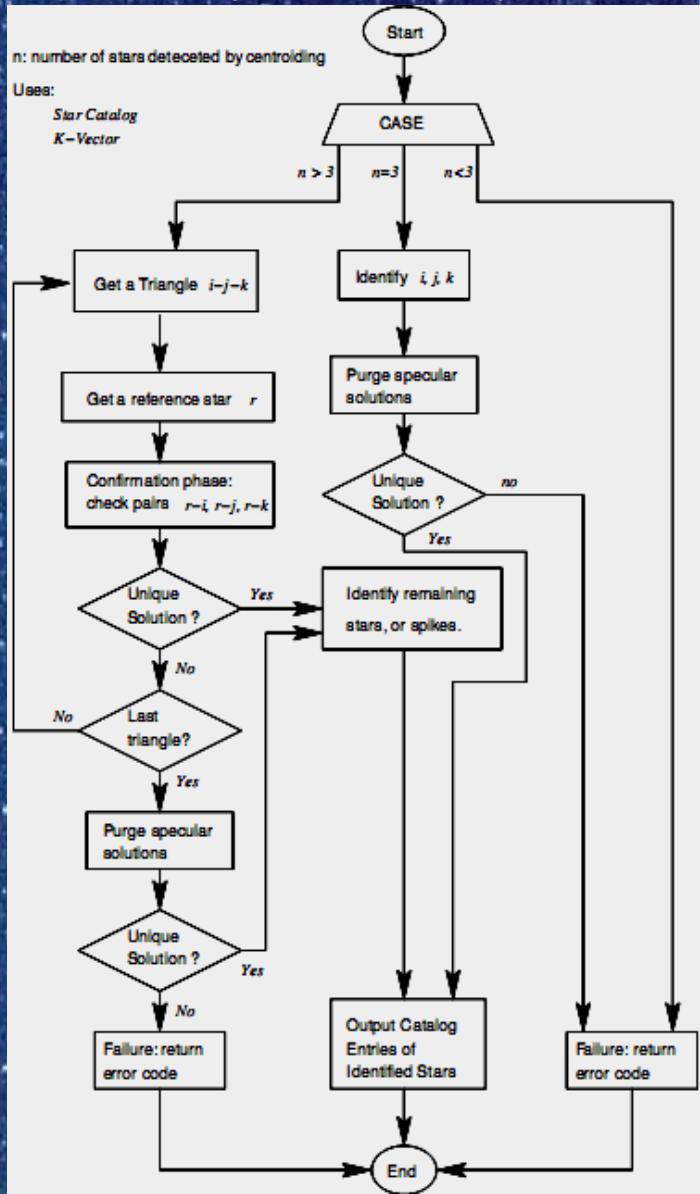
star pattern matching: II
Lost In Space Pyramid (LISP)

method

- if (blobs == 3) simply check uniqueness of triangle
- if (blobs > 3)
  - look for a unique triangle
  - if found, scan remaining stars
  - if pyramid found and unique, done!
  - else, scan other triangles
  - if no pyramid found, return unique triangle
- uniqueness tested against a-priori level of confidence, i.e. compute:

"expected frequency that false matches between measured stars, to within measurement precision, are matched by random pattern combinations in the catalog, assuming a uniform star density"

[Mortari et al. 2004]



star pattern matching: II
Lost In Space Pyramid (LISP)

performance

- performs well (<0.1s) with a few bright blobs in the FoV
- performs ok (~1s) with many blobs, of which a few bright
- performs poorly (~10s) with many faint blobs
- reason: 35 MB catalogue only includes mag 8 stars
- there are ways to improve this, but…
  - daytime operation
  - ROMA
  - suggestions? ☺

star pattern matching: III
Run-Of-Mill Algorithm (ROMA)

- only runs with a pointing "guess"
- all-sky deeper catalogue: mag 10, ~100 MB
- loads all the stars in the relevant patch of sky
- nothing fancy, brute-force approach
- compare distance between 2 brightest blobs/stars, within tolerance
- if matched, a third blob is added…and so forth…
- this is the routine algorithm, because we have many pointing sensors

star pattern matching: VI
Astrometry.net

UNIVERSITY OF TORONTO

courtesy of Dustin Lang!

[Lang et al. 2009]

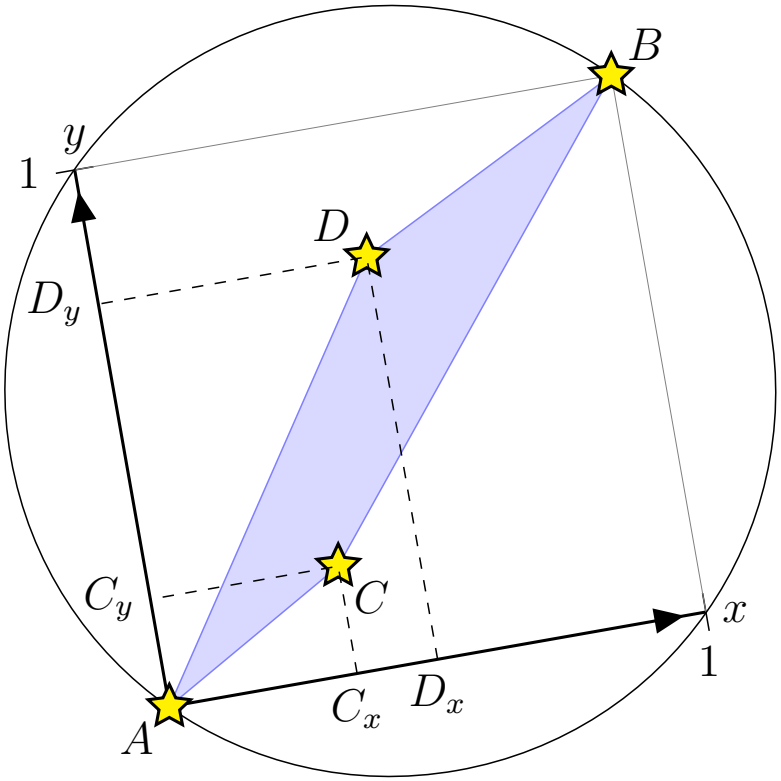
Astrometry.net: The general approach

Two-stage approach:

- Use a search heuristic to generate hypothetical alignments of the image on the sky
- Use Bayesian decision theory to reject false hypotheses

Search heuristic: try to match sets of 4 stars in the image to a large database (or index) created from an astrometric reference catalog.

Astrometry.net: The geometric feature



- Four-star features
- Two most distant stars are labelled *A*, *B*
- They establish a local coordinate frame
- Two others stars are labelled *C* and *D*
- Their positions in the local coordinate frame become the feature descriptor, (C_x, C_y, D_x, D_y)

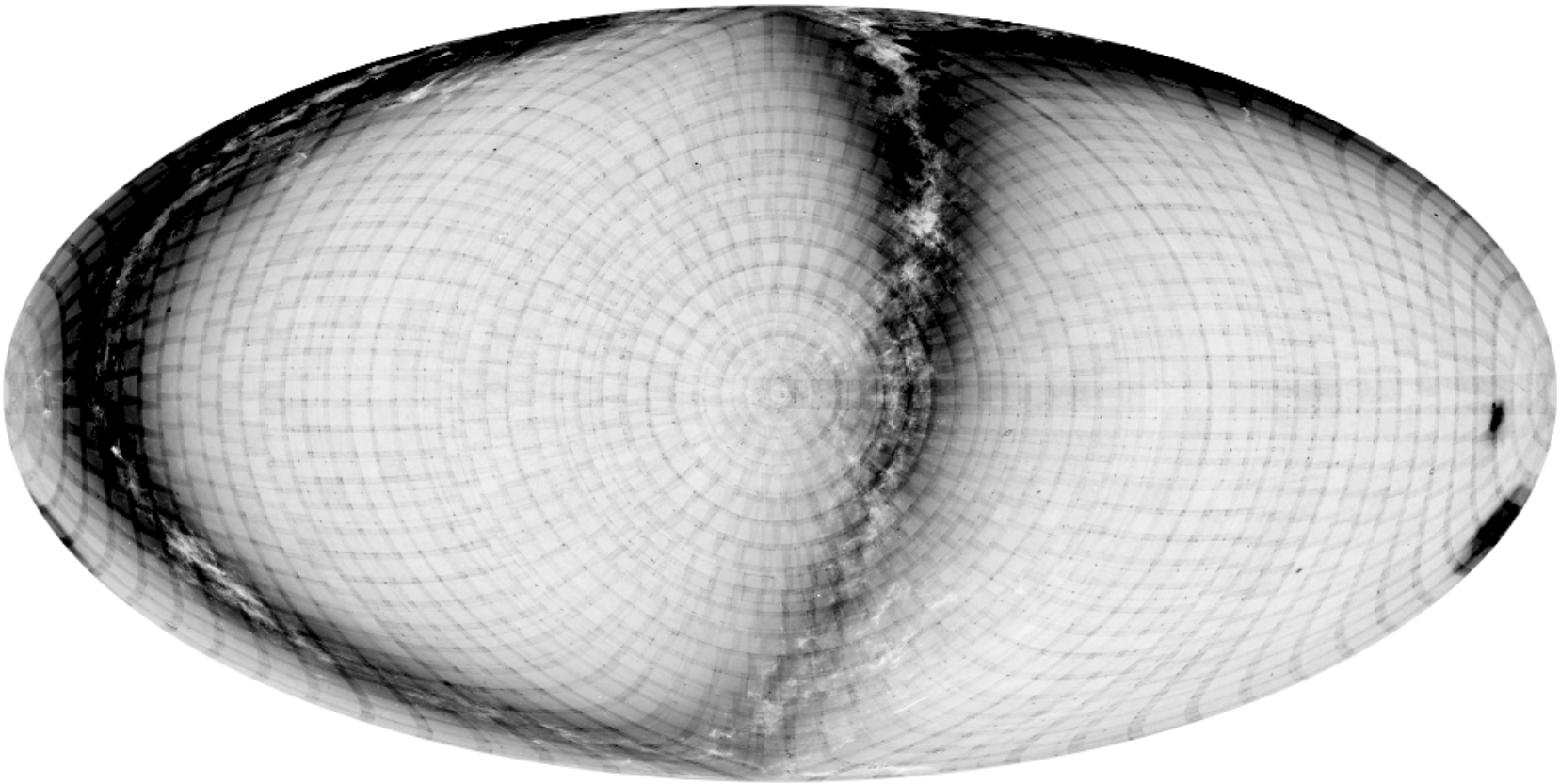
Astrometry.net: Test time

Given a new image:

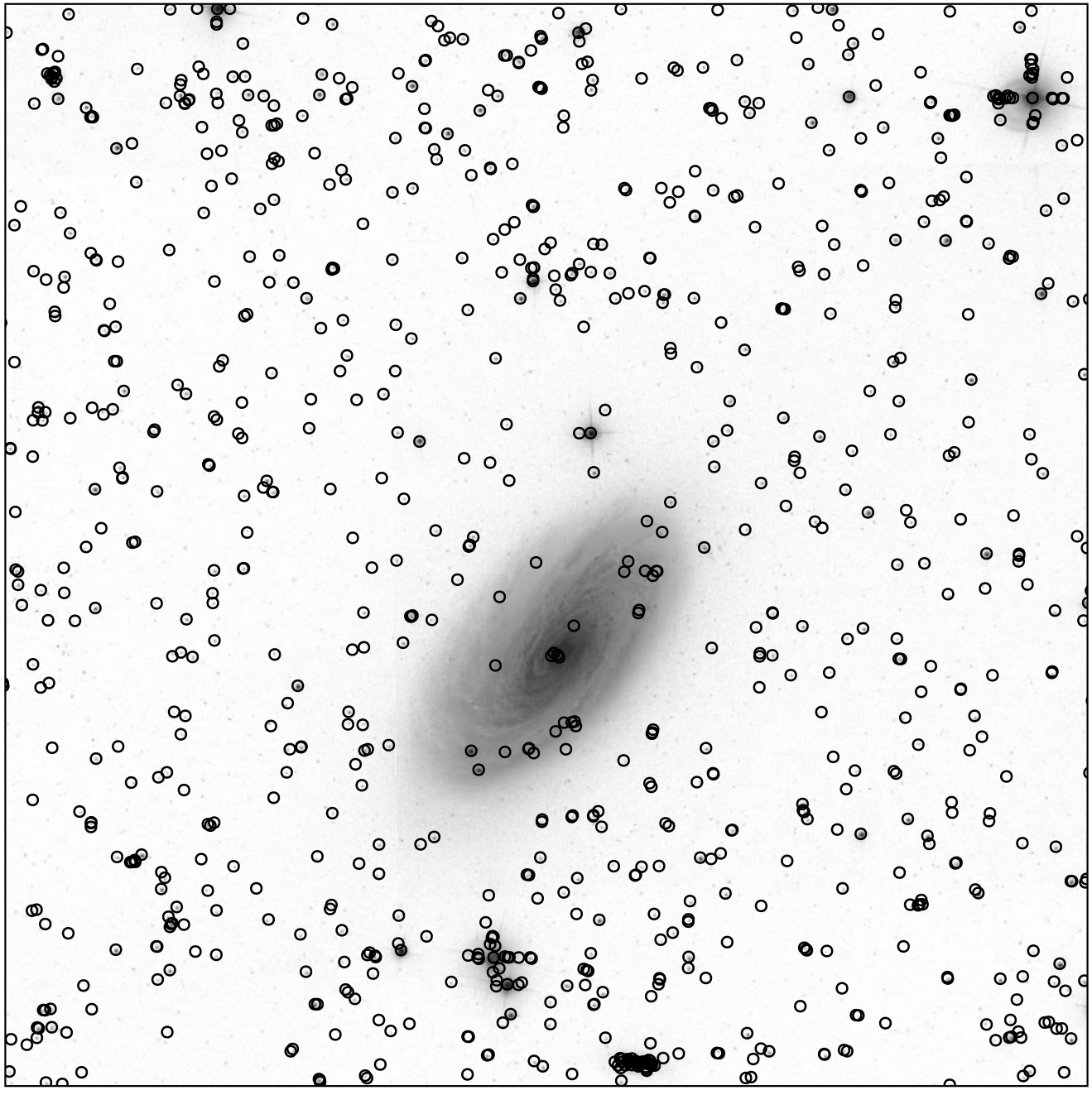
- Find stars
- Repeat many times:
  - Choose 4 stars
  - Compute the shape of those four stars
  - Search in the index for nearby shapes in (4-D) shape space
  - For each matching feature, look up the stars it was made from and compute transformation
  - Check each match to filter out false positives

Astrometry.net: Uniform indexing

- Stars in the reference catalog are very non-uniform
- We want to be able to recognize images anywhere on the sky equally well: Need to uniformize.

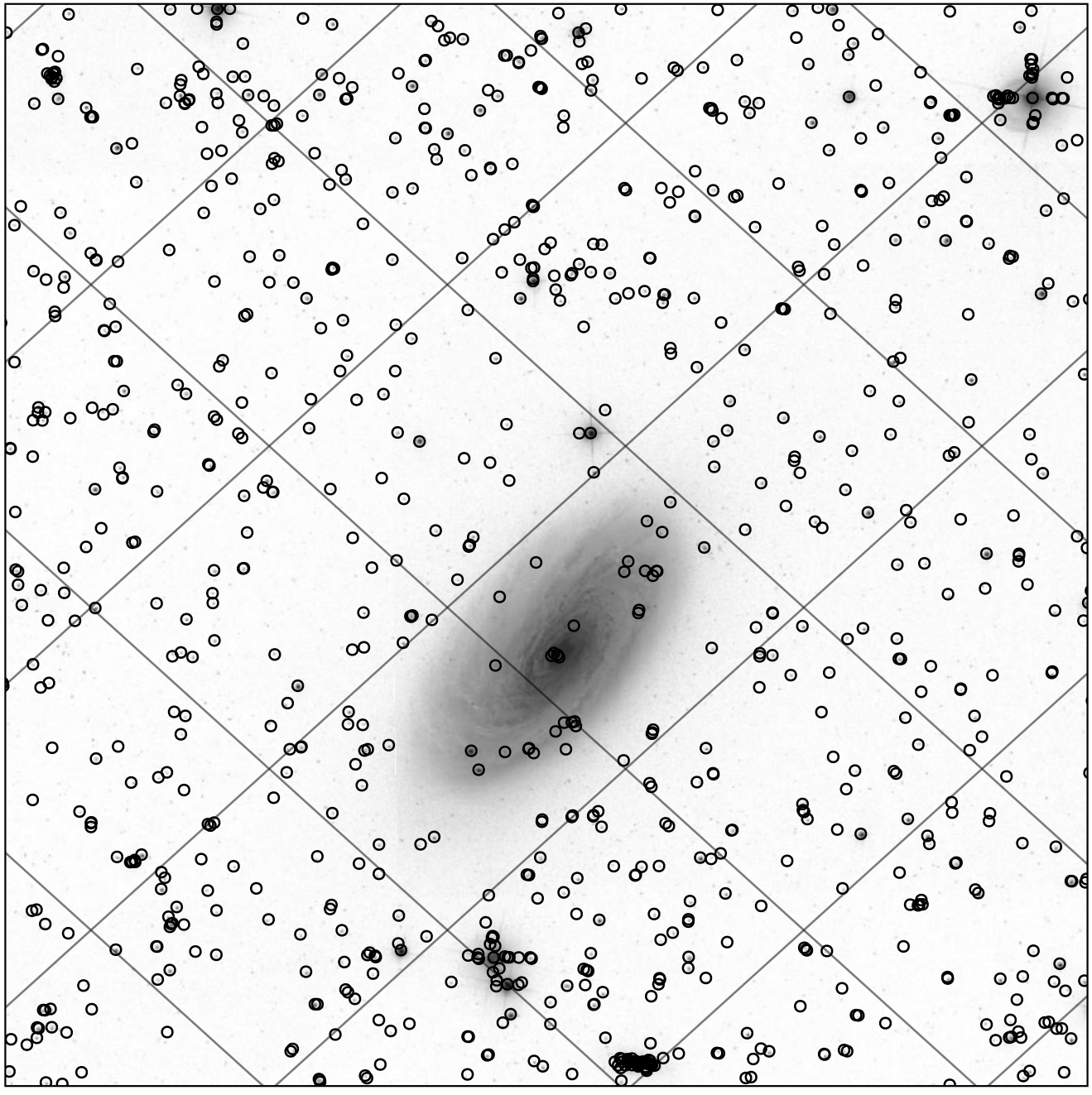


Astrometry.net: Indexing



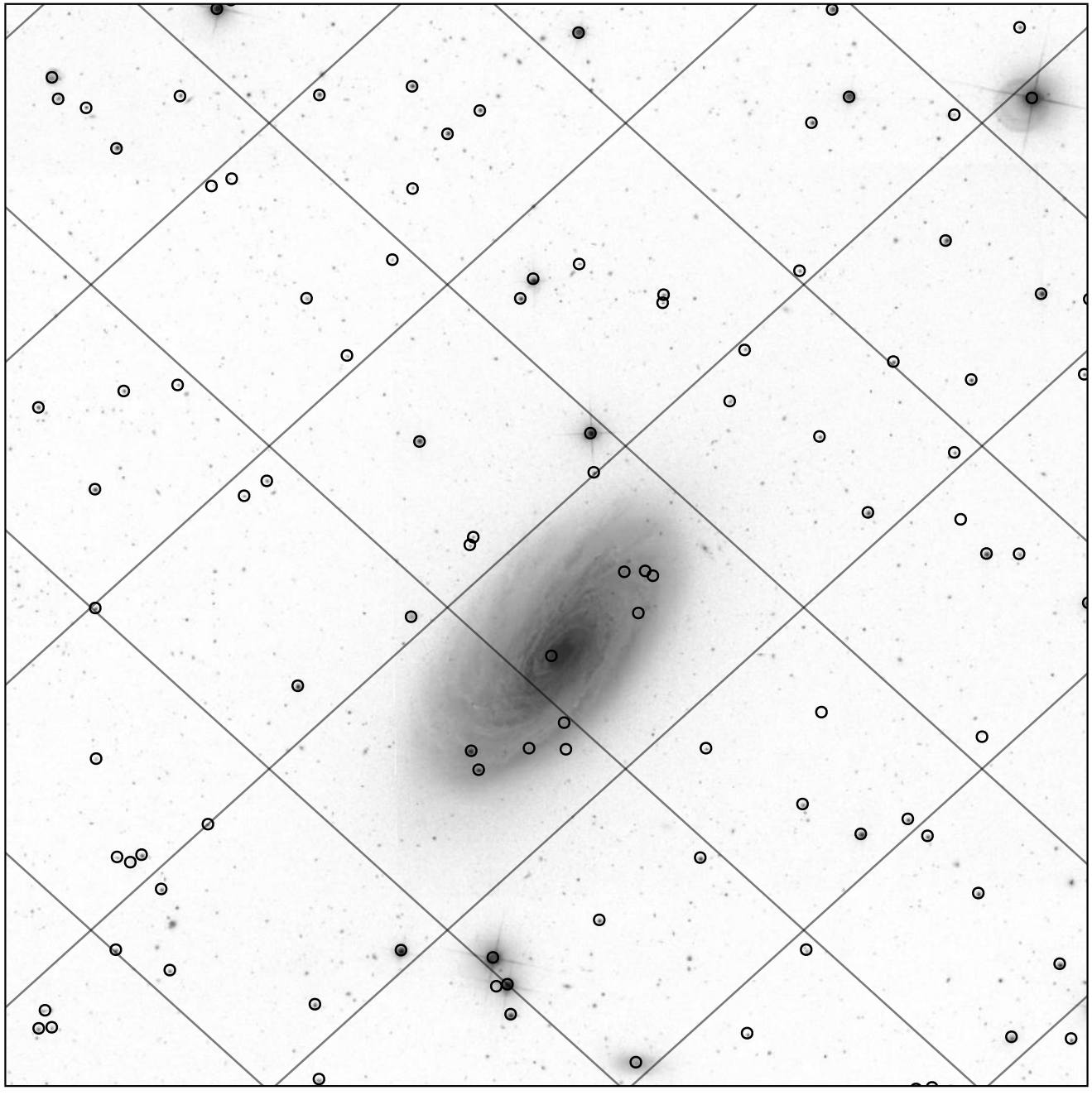
- Start with reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

Astrometry.net: Indexing



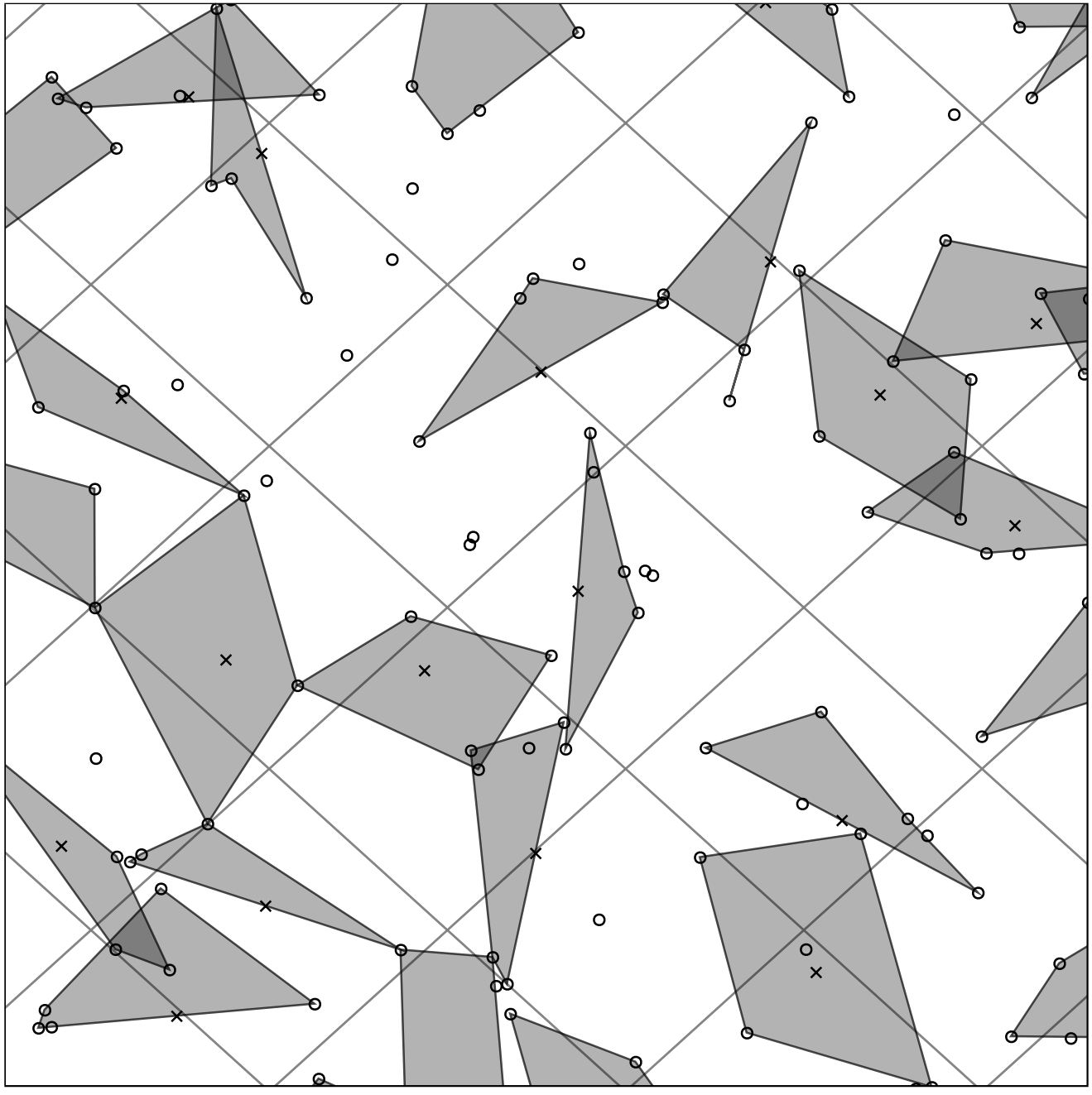
- Start with reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

Astrometry.net: Indexing



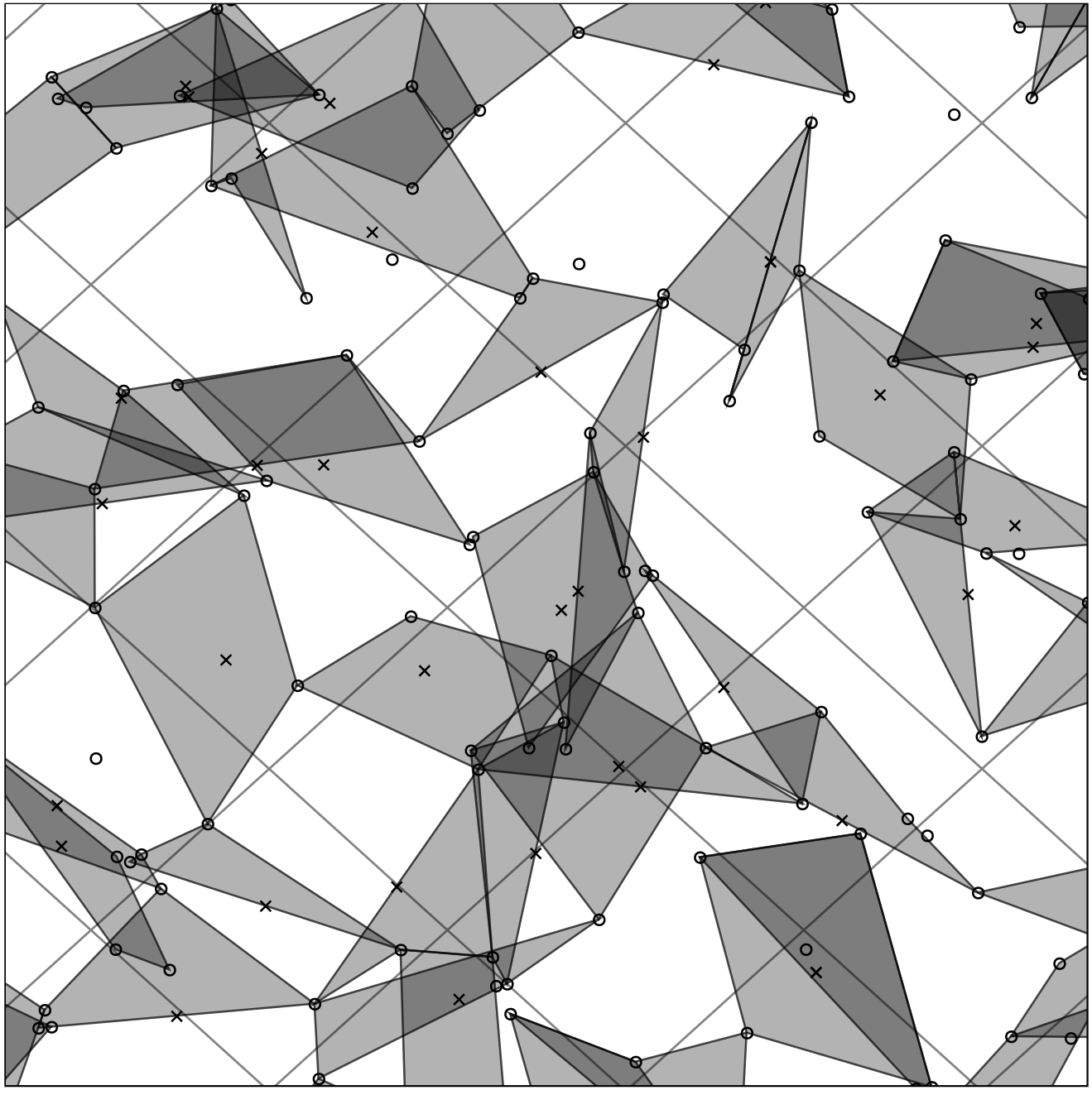
- Start with reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

Astrometry.net: Indexing



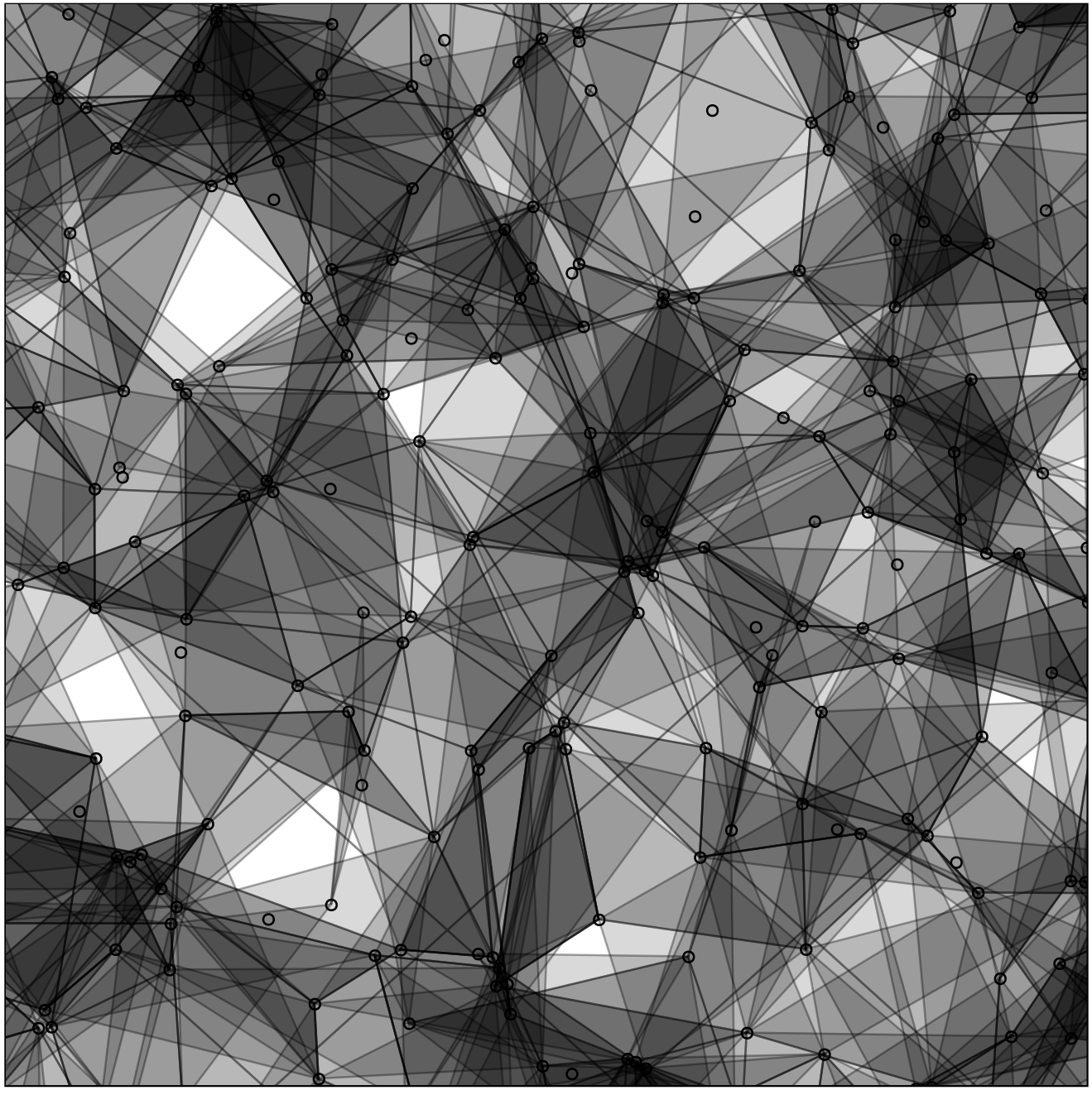
- Start with reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

Astrometry.net: Indexing



- Start with reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

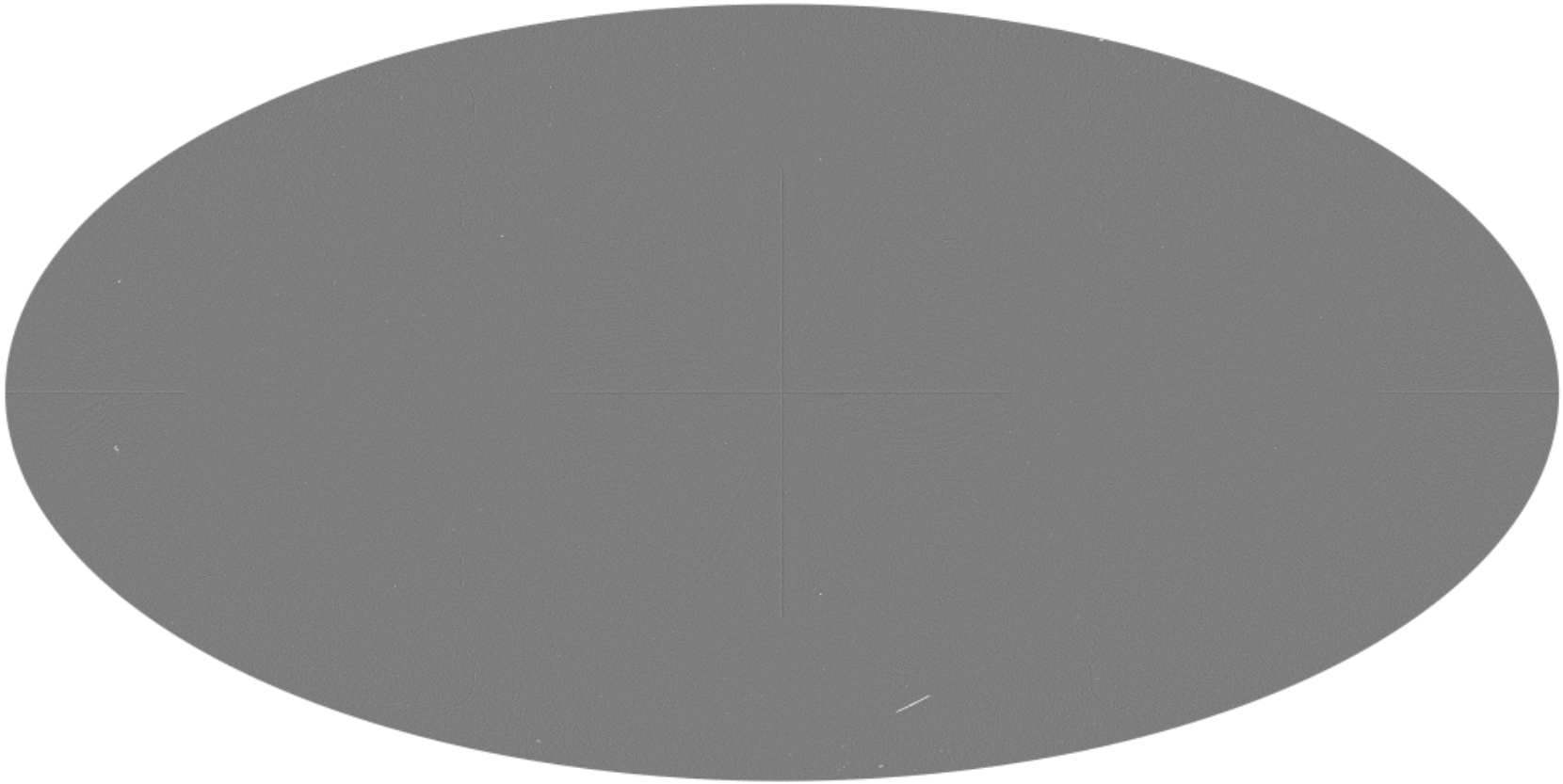
Astrometry.net: Indexing



- Start with reference catalog
- Place a grid over the sky
- Select n brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build N geometric features in each cell

Astrometry.net: Uniform indexing

- The resulting index has a uniform density of features

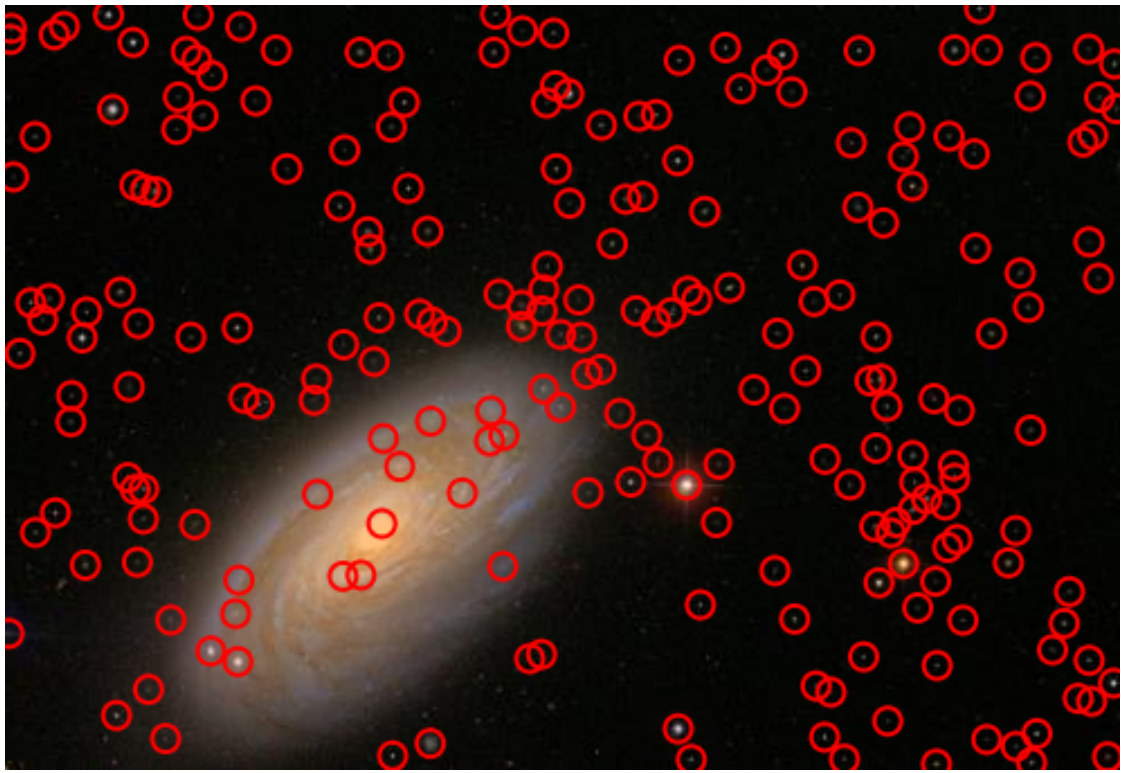


Astrometry.net: Test time [1/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index

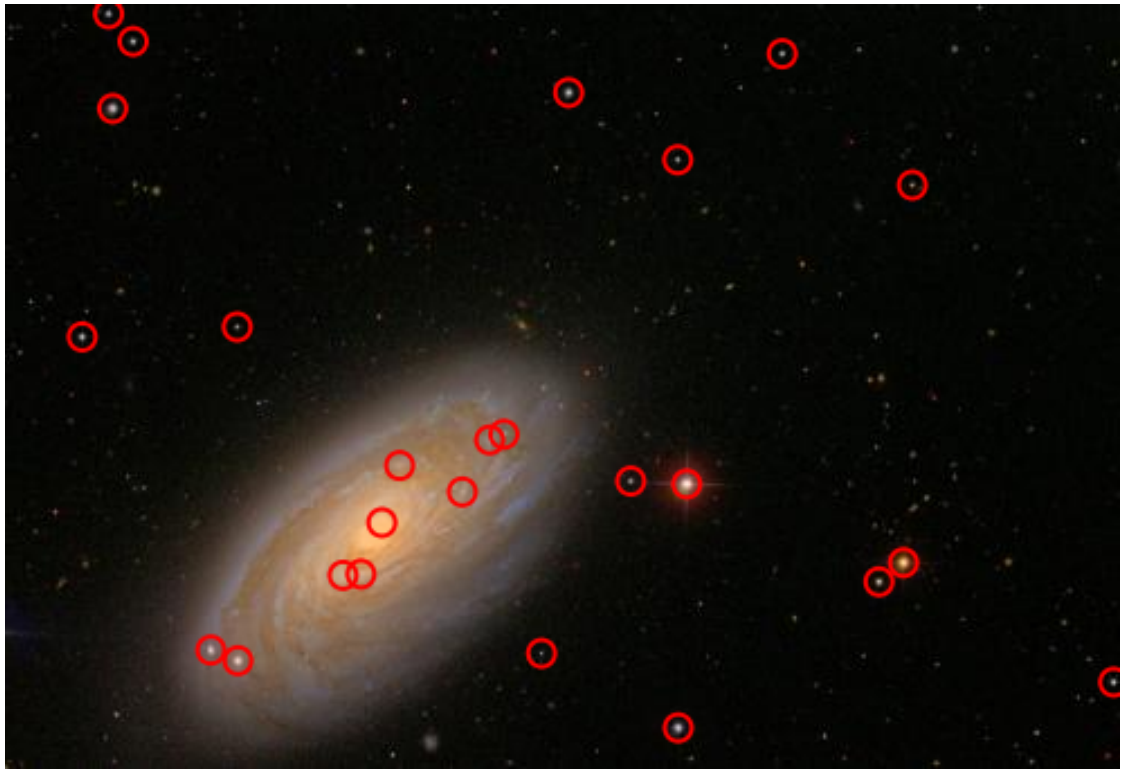
Astrometry.net: Test time [2/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



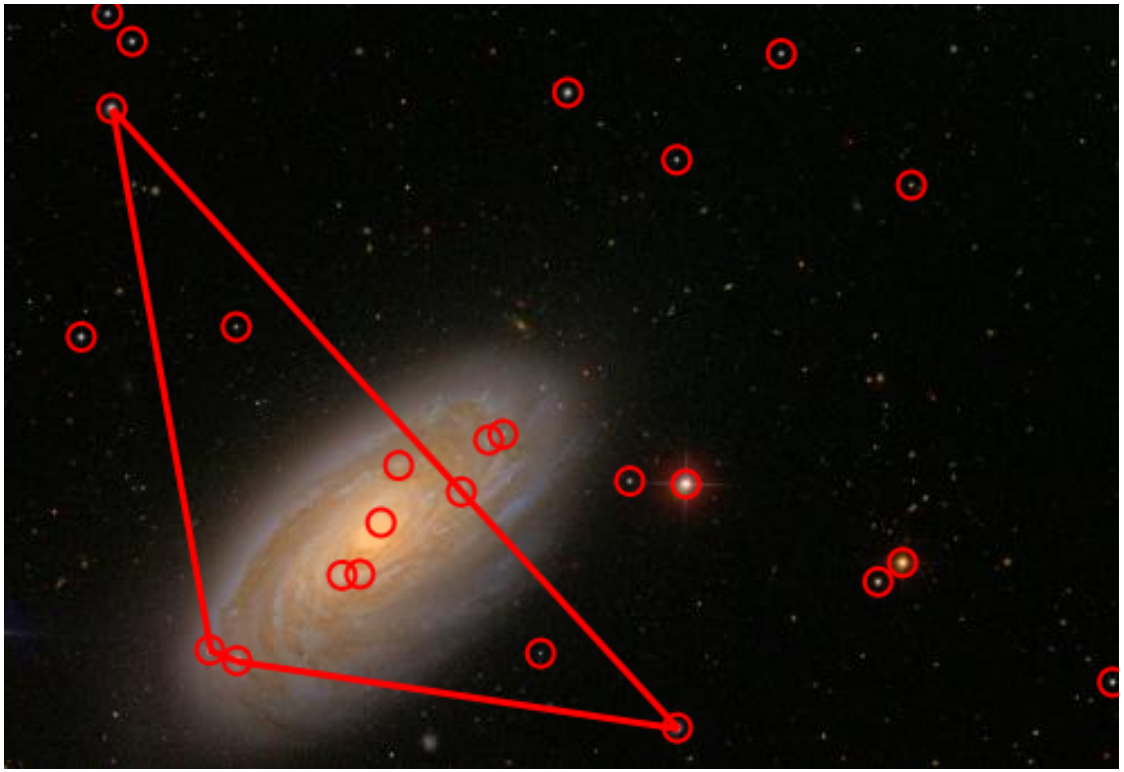
Astrometry.net: Test time [3/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



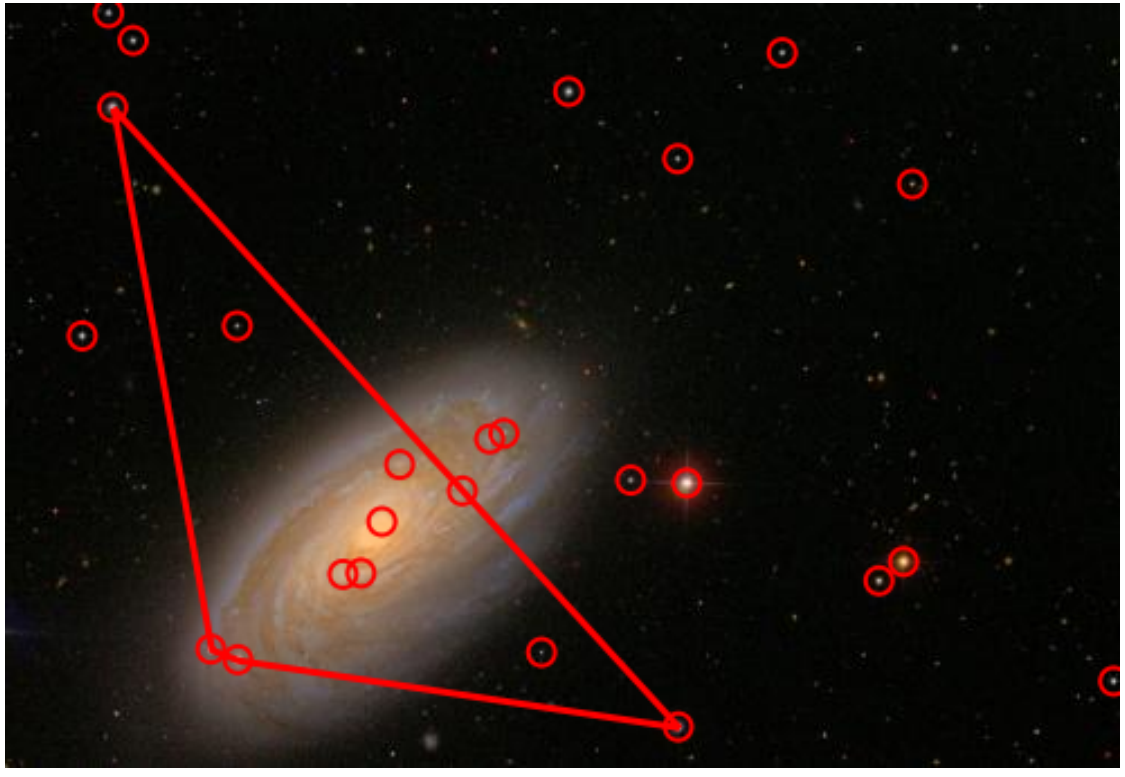
Astrometry.net: Test time [4/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index

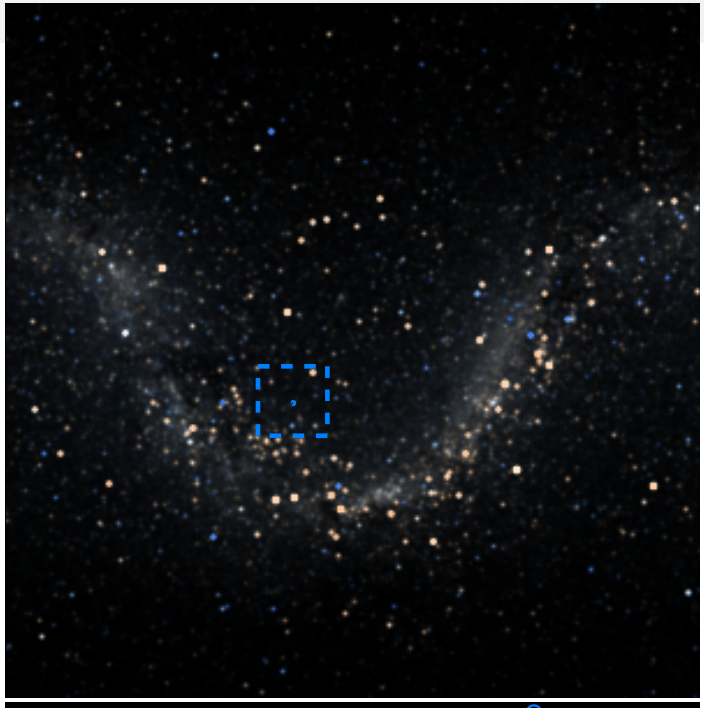


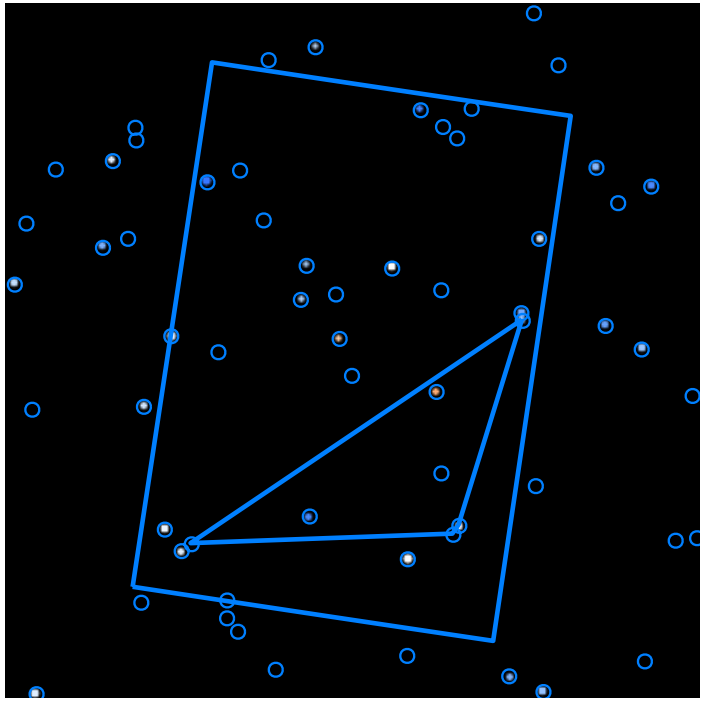
Astrometry.net: Test time [5/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



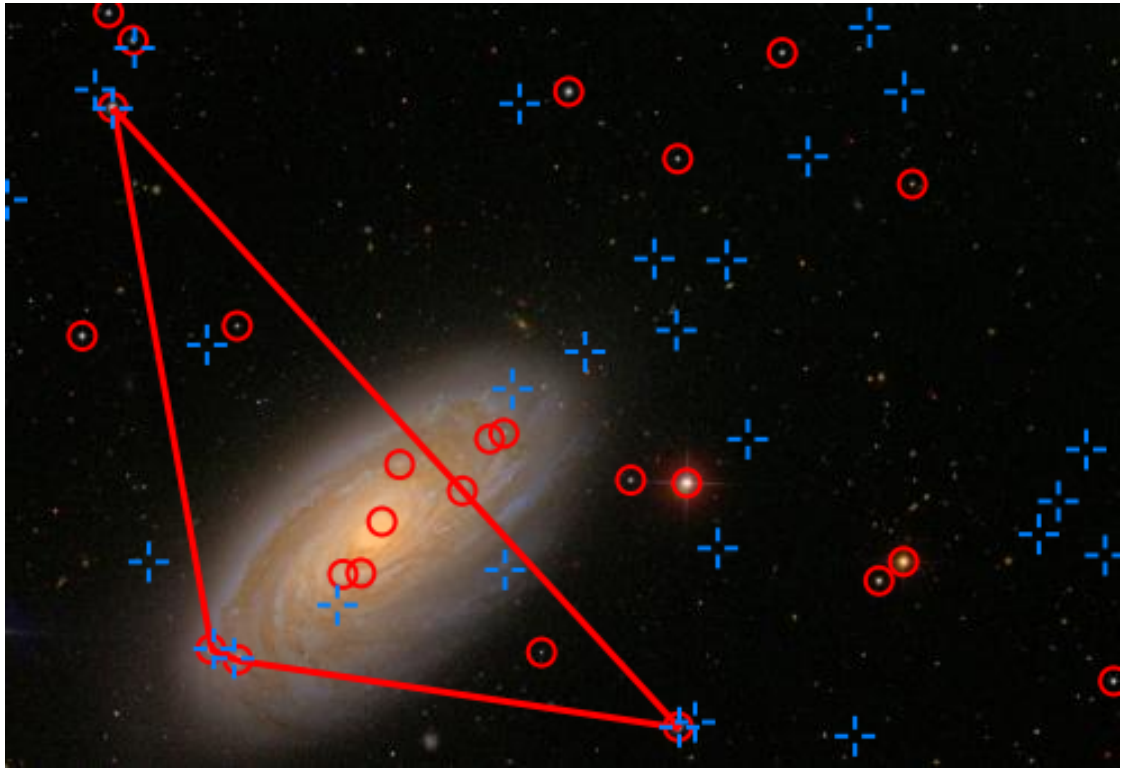
Match #1



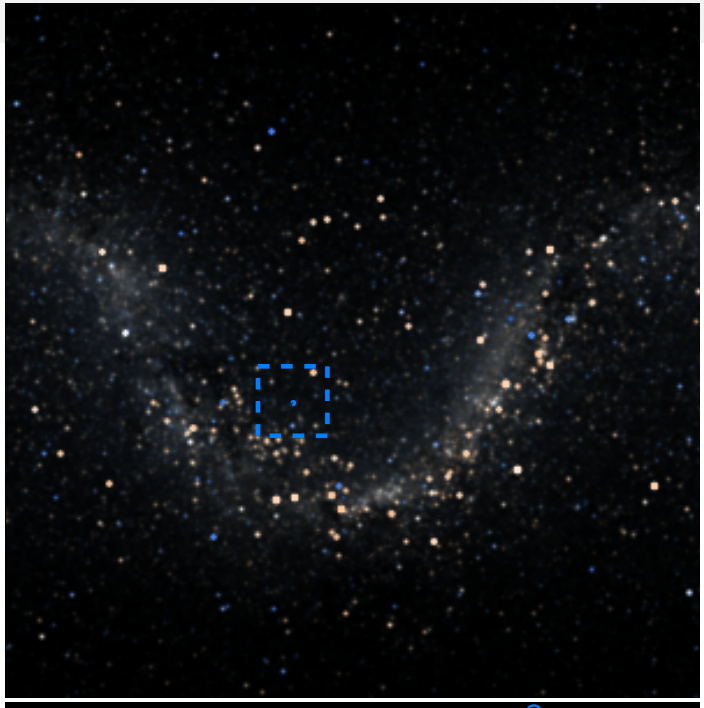


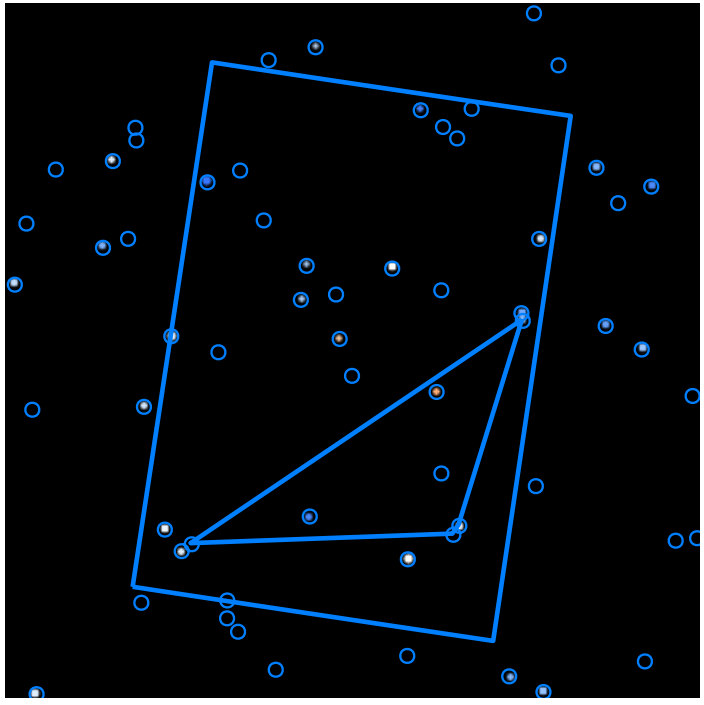
Astrometry.net: Test time [6/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



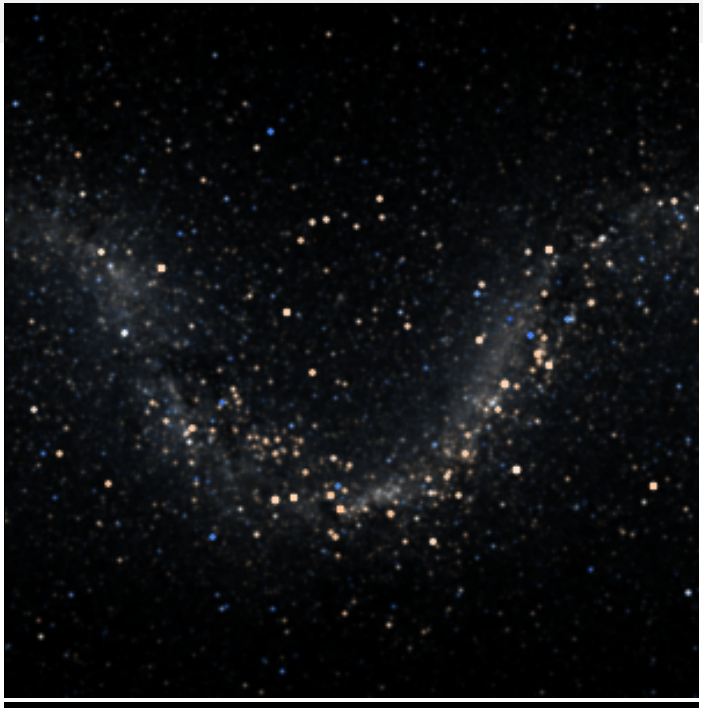
Match #1

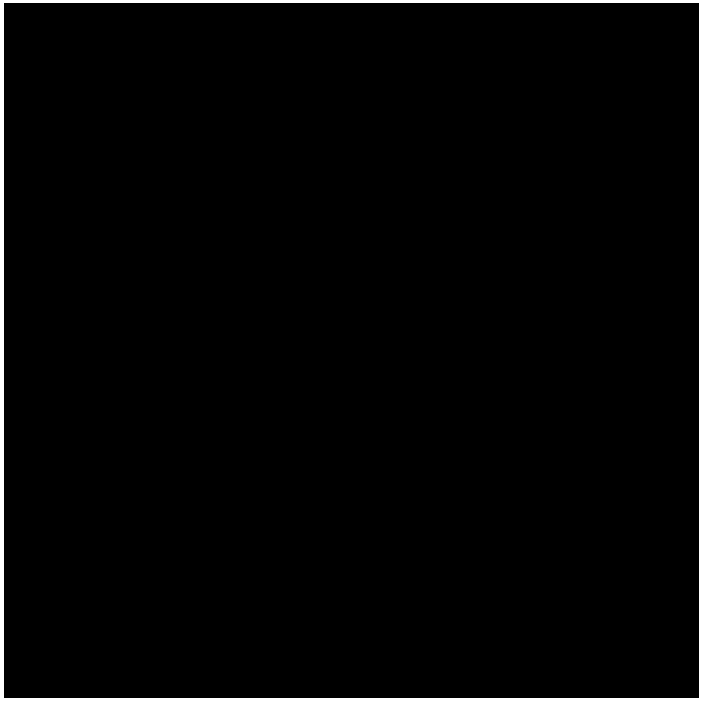


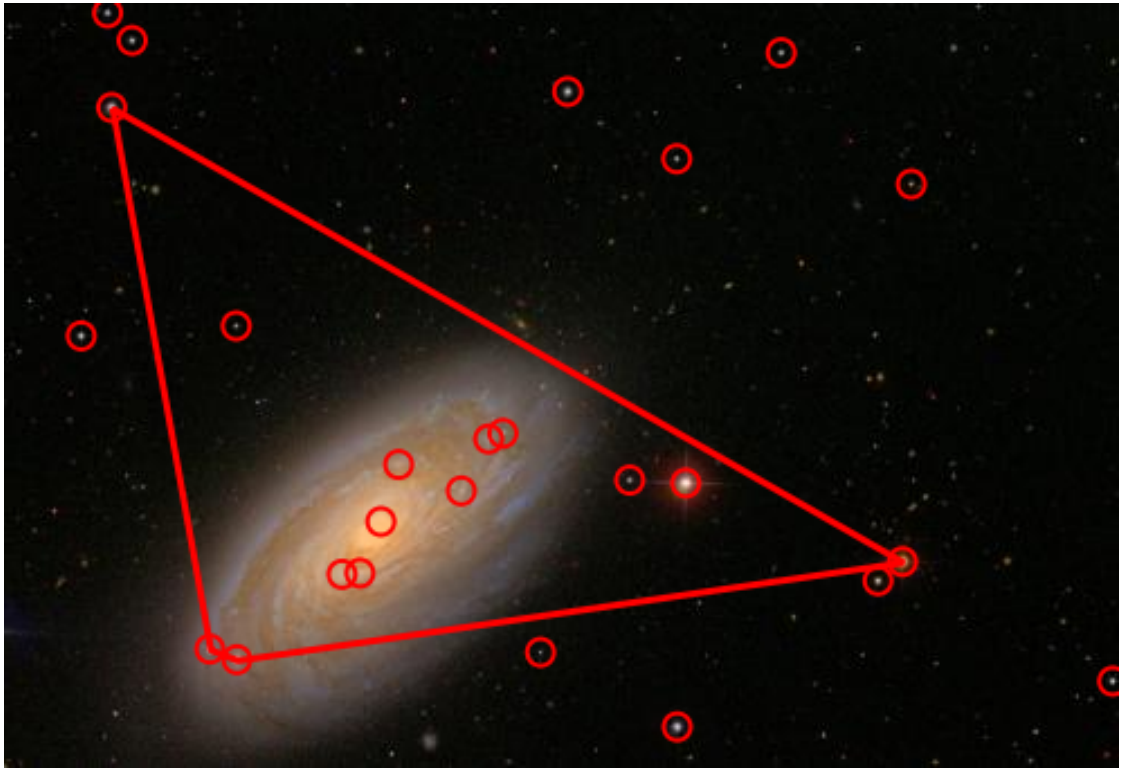


Astrometry.net: Test time [7/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index

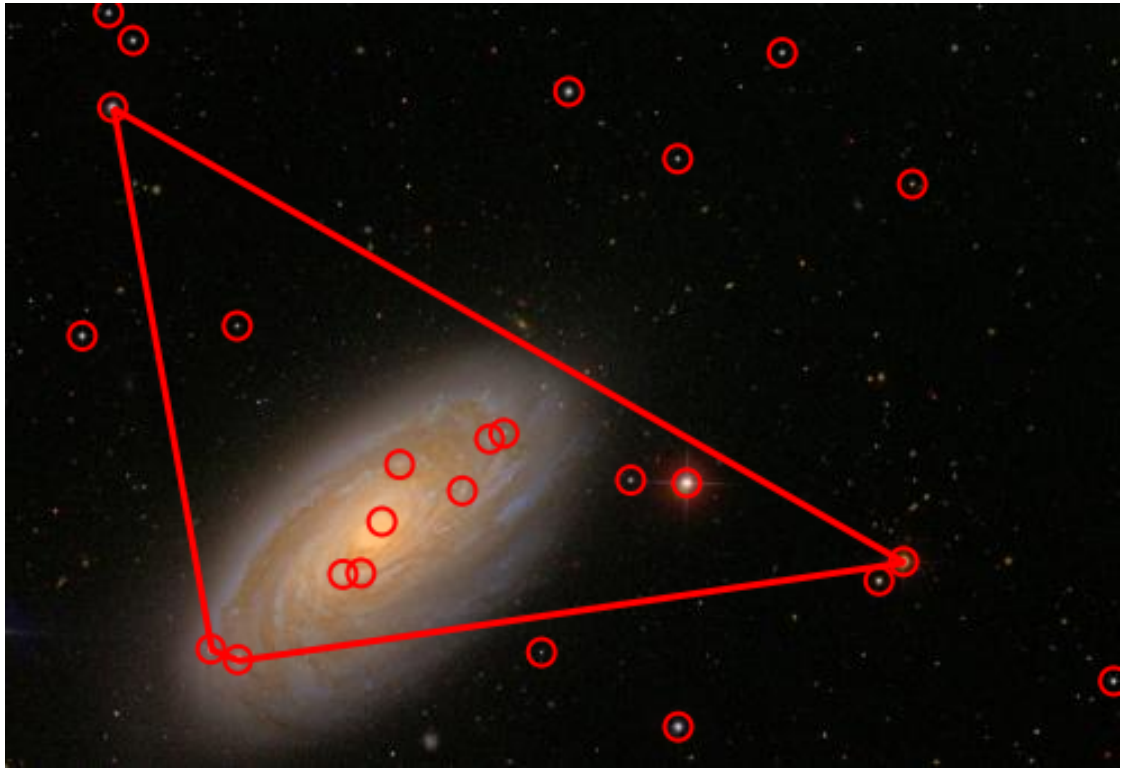




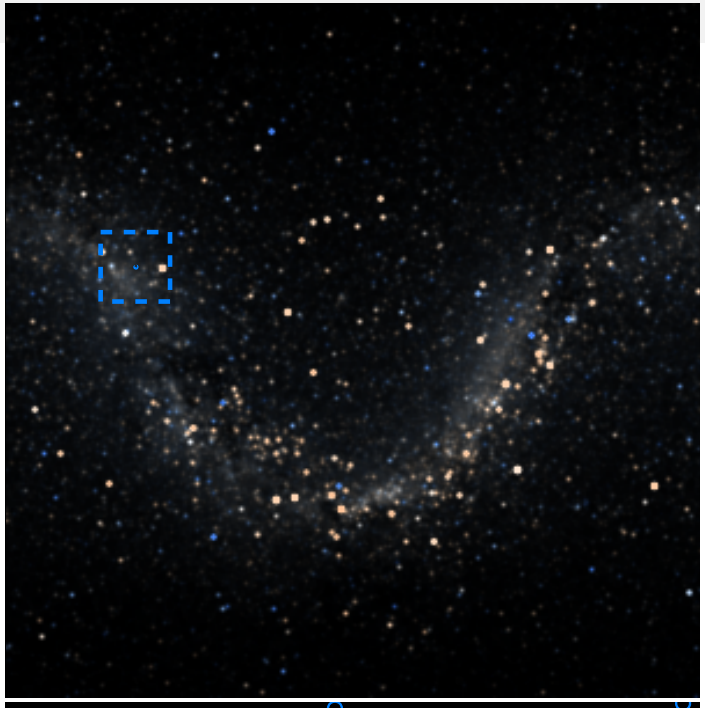


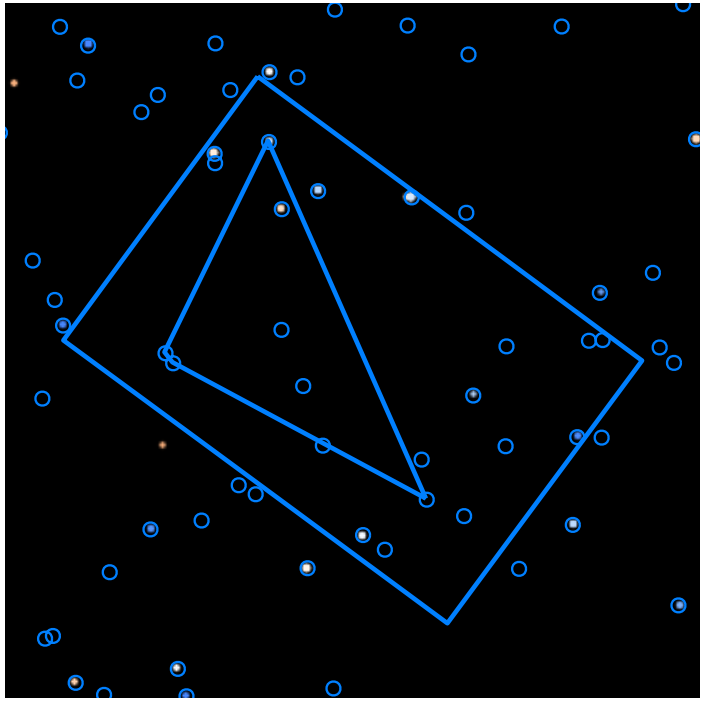
Astrometry.net: Test time [8/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



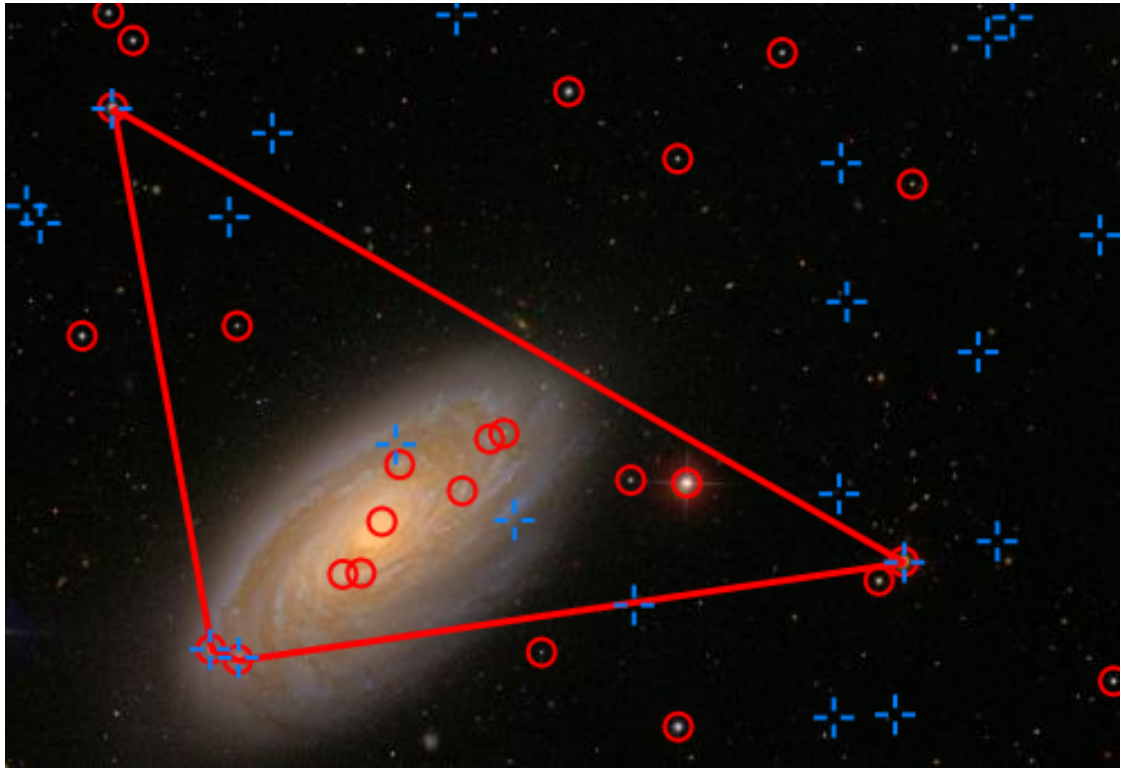
Match #1



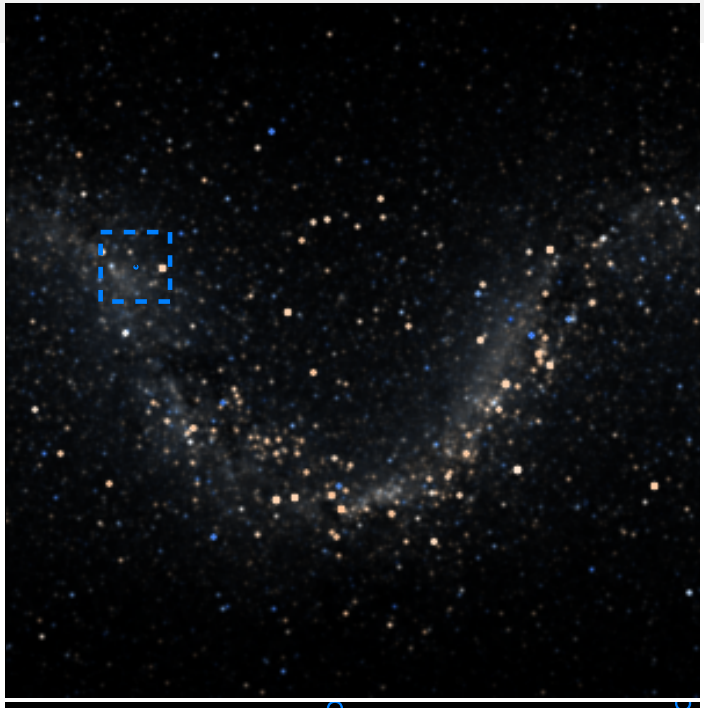


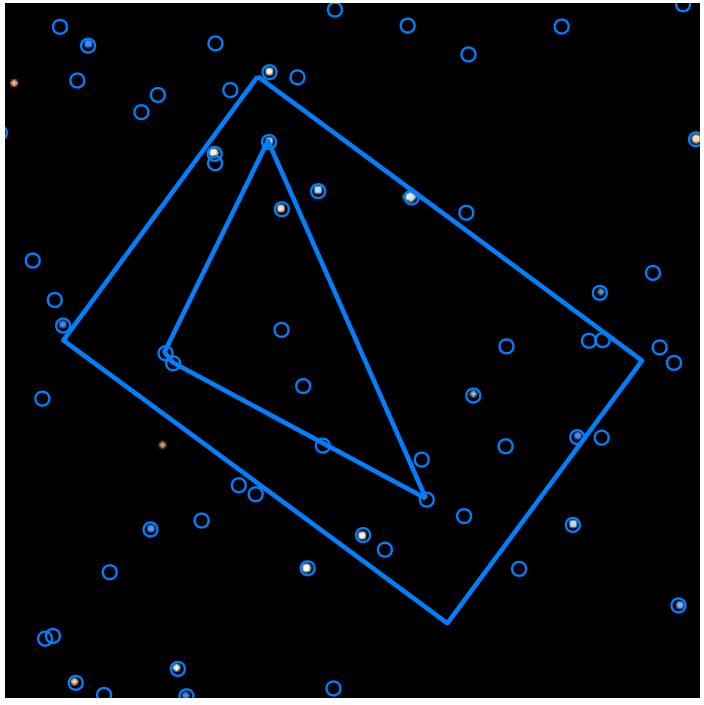
Astrometry.net: Test time [9/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



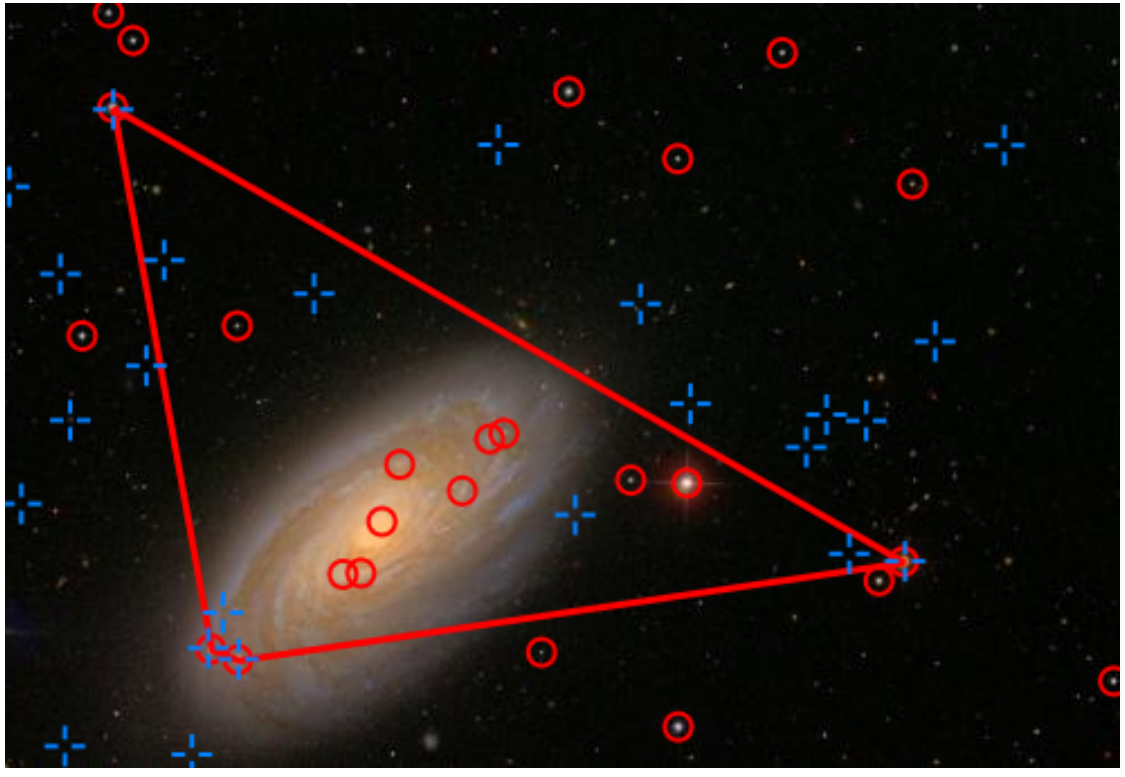
Match #1



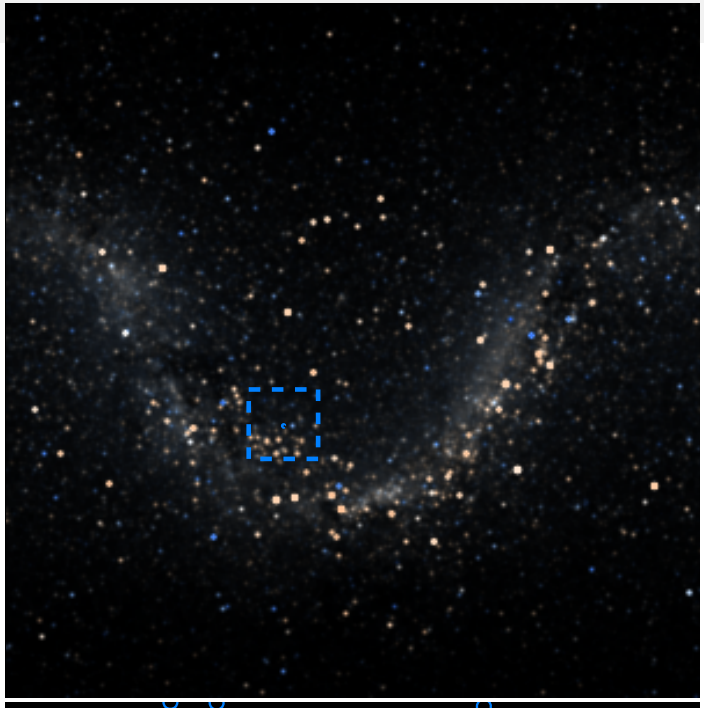


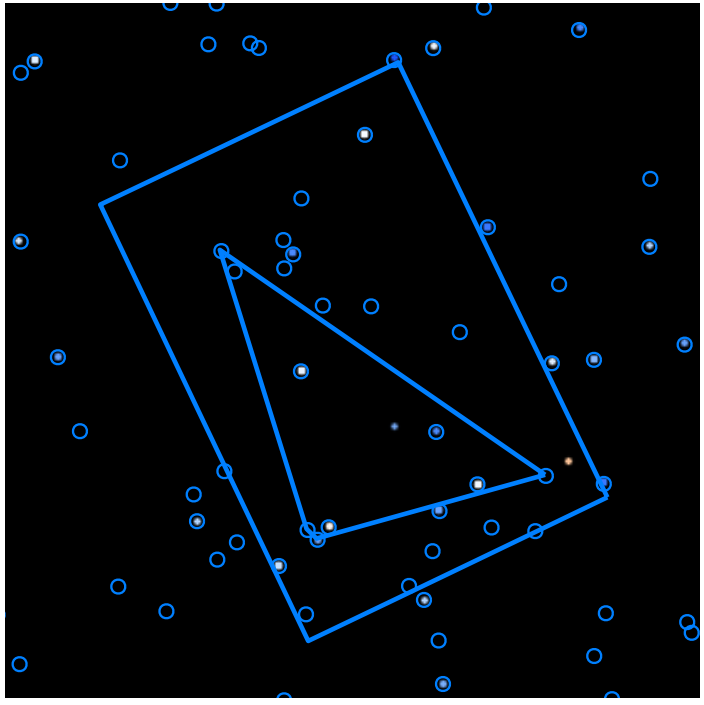
Astrometry.net: Test time [10/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



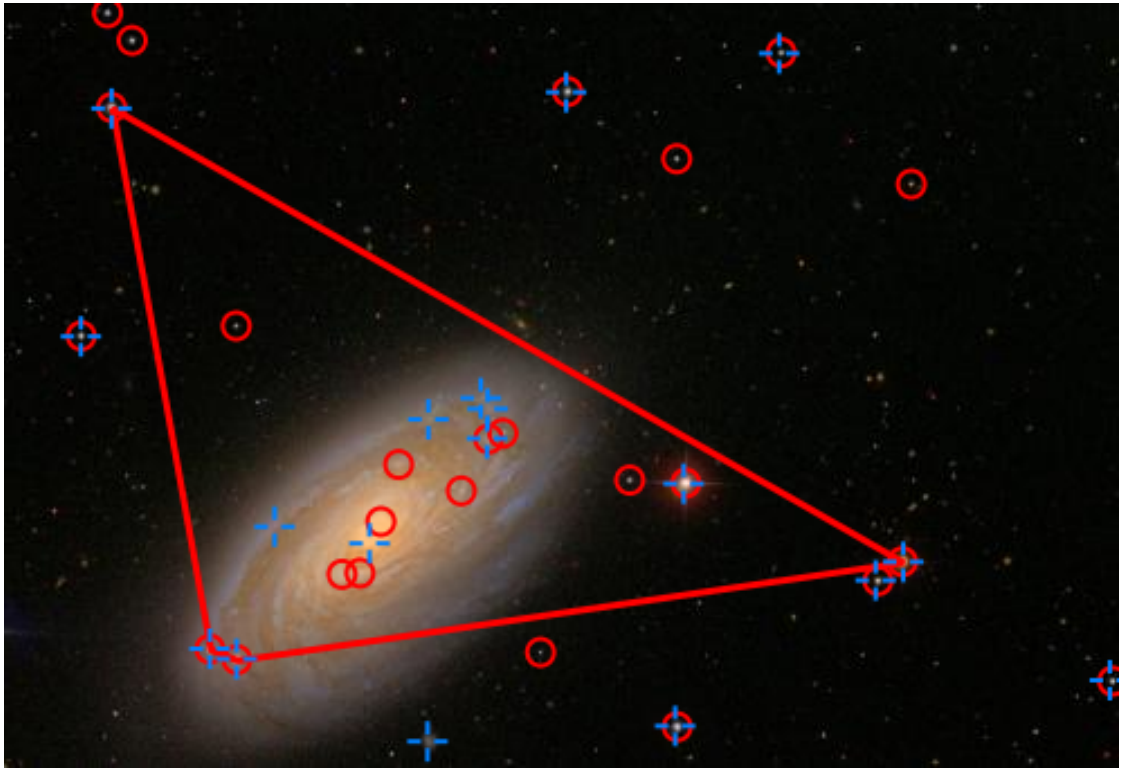
Match #2



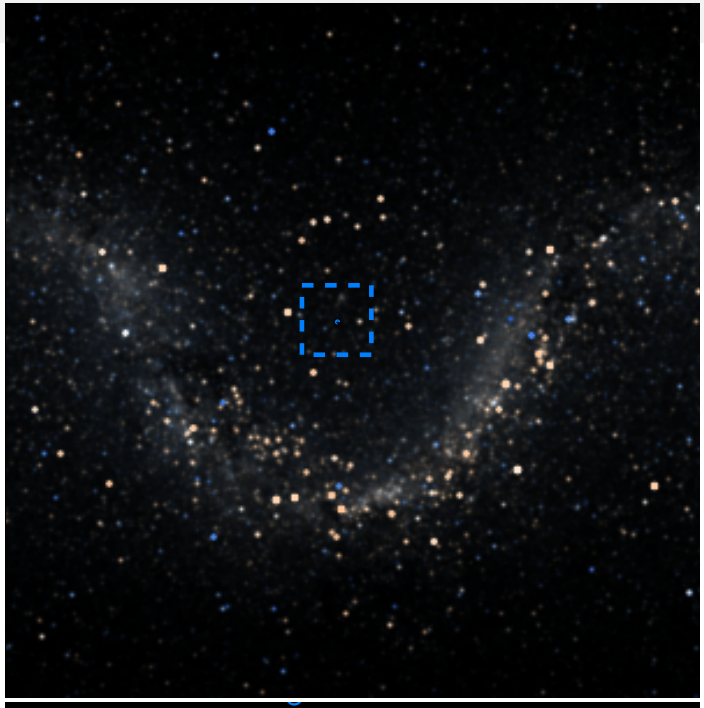


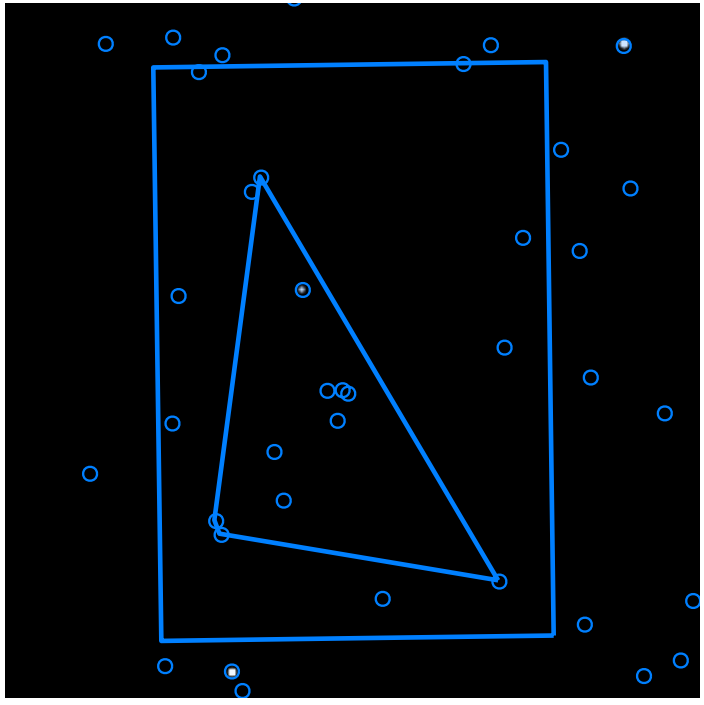
Astrometry.net: Test time [11/11]

- Detect stars (source extraction)
- Starting with the brightest stars . . .
- Build another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



Match #3





star pattern matching: VI
Astrometry.net

UNIVERSITY OF TORONTO

Astrometry.net

performance

- custom index with stars brighter than 10 mag, 50 MB to be loaded in RAM
- 556/558 (99.6%) frames recognized correctly. One of those 2 has only 2 stars
- performance: 23 ms/frame on average with MacBook (Core2 duo 2.4 GHz, using one core)

courtesy of Dustin Lang!

[Lang et al. 2009]