The Web is My Observatory: Enabling science with heterogeneous images

Dustin Lang

McWilliams Postdoctoral Fellow, Carnegie Mellon University
visiting University of Waterloo

Western Physics & Astronomy Colloquium, 2015-02-05

Talk outline

Talk outline:

- Web astronomy — finding comets by searching the web
- Choose 1:
  - *Enhance!* — calibrating photometry for an Open-Source Sky Survey
  - *Astrometry.net* — recognizing astronomical images

Comet Holmes: Crowd-sourced astronomy

- We did a Yahoo Image Search for "Comet Holmes"
- Got 2300 result images, fed them into *Astrometry.net* to astrometrically calibrate (and vet) them
- 1300 were recognized as pictures of the sky (and their pixel-to-sky mappings were found)
- We fit an elliptical orbit to recover the trajectory of the comet through the Solar system

Comet Holmes: Crowd-sourced astronomy

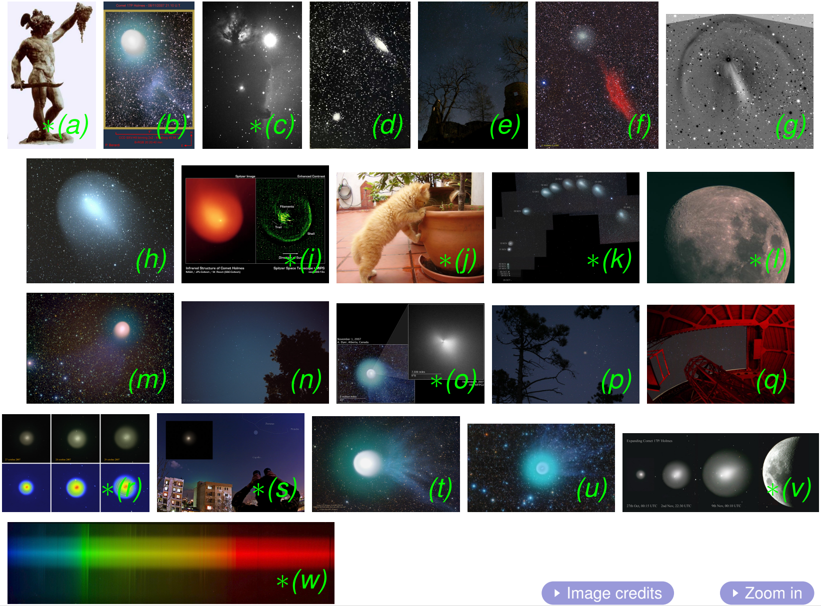
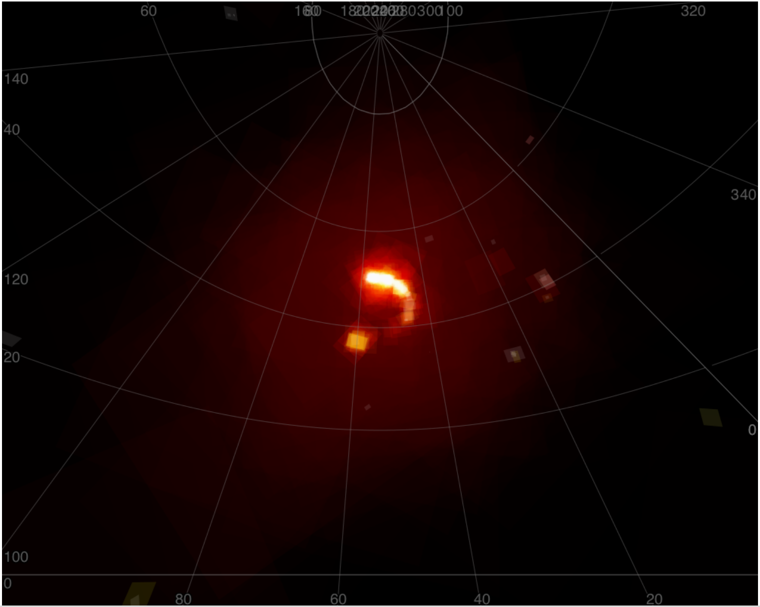


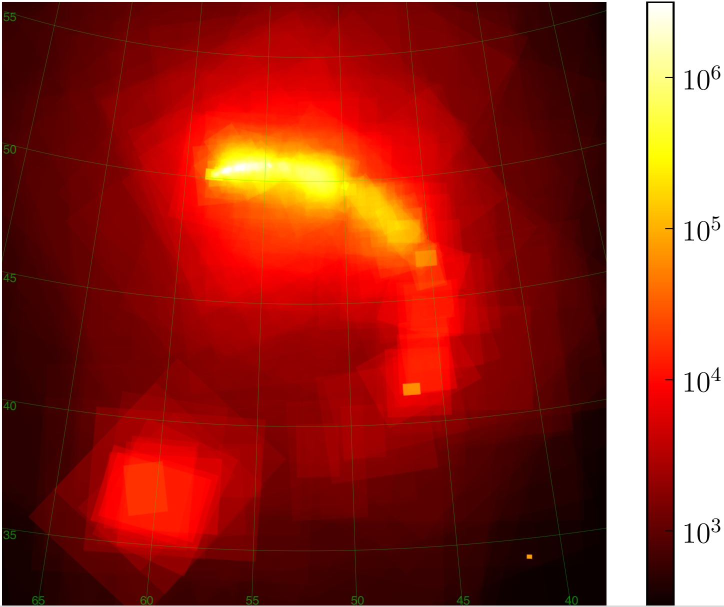
Image credits

Zoom in

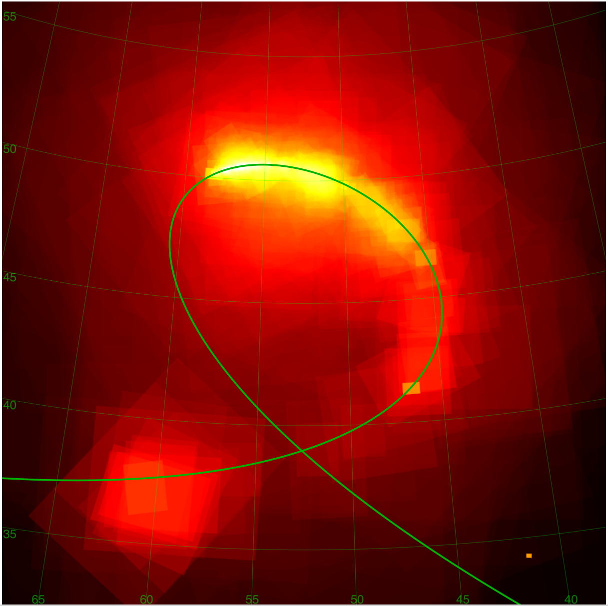
Comet Holmes: Pixel density



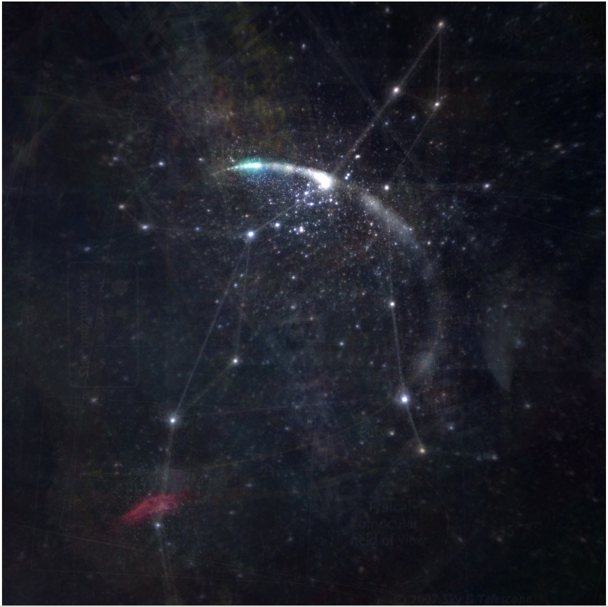
Comet Holmes: Pixel density (pixels/deg^2)



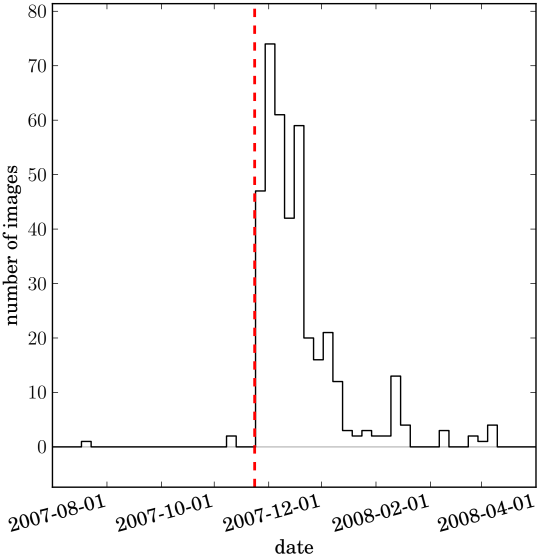
Comet Holmes: Pixel density and true path



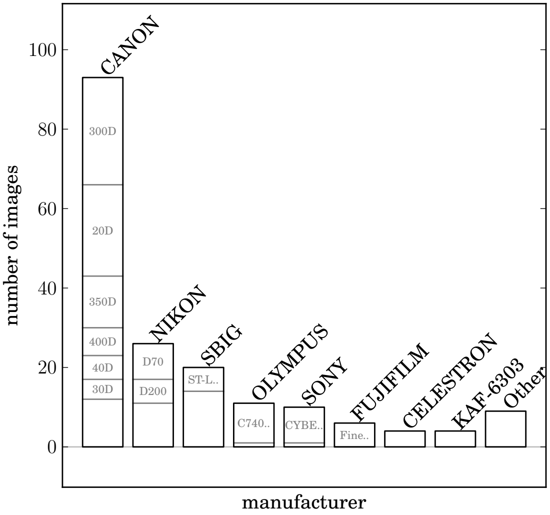
Comet Holmes: Co-adding the images



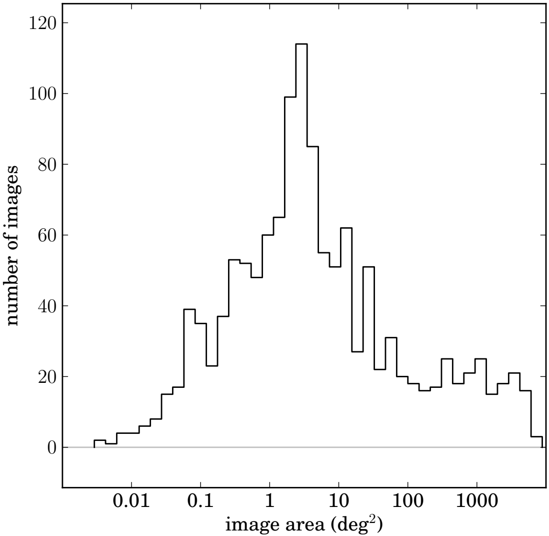
Comet Holmes: Timestamps from EXIF



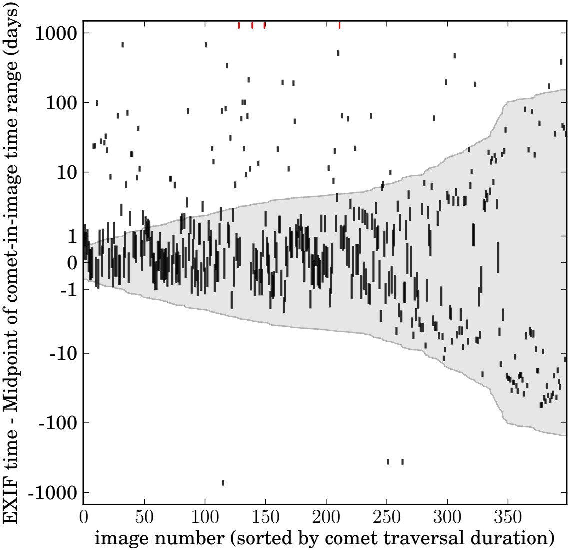
Comet Holmes: Manufacturers from EXIF



Comet Holmes: Image area



Comet Holmes: Accuracy of timestamps



Comet Holmes: Inferring the orbit

- Generative modeling approach—build a model of the comet *and* the photographers that explains the observed data
- The data are the image pointings on the sky—we don't try to locate the comet within the images!
- Given orbital parameters, we trace the trajectory of the comet in (RA,Dec) over time
- Likelihood of an image is proportional to the amount of time the comet spends within the image, *weighted* by when we think the image was taken
- Adjust orbital parameters to maximize the product of all $\sim$ 1300 image likelihoods

Comet Holmes: Inferring the orbit

- Likelihood for a single image pointing:

$$p(\boldsymbol{\alpha}_i|\boldsymbol{\Omega}_i, \boldsymbol{\omega}, \boldsymbol{\theta}) = p_{\mathrm{good}}\, p_{\mathrm{fg}}(\boldsymbol{\alpha}_i|\boldsymbol{\Omega}_i, \boldsymbol{\omega}, \boldsymbol{\theta}) + [1 - p_{\mathrm{good}}]\, p_{\mathrm{bg}}(\boldsymbol{\alpha}_i)$$

  $\boldsymbol{\alpha}_i$ is the pointing of image i (RA,Dec center)
  $\boldsymbol{\Omega}_i$ is the field of view (fixed; given by *Astrometry.net*)
  $\boldsymbol{\omega}$ are the comet's orbital parameters
  $\boldsymbol{\theta}$ are the hyperparameters (η, p_{good}, p_{EXIF})
- This is a mixture model—a "foreground" or "inlier" component plus a "background" or "outlier" component.

Comet Holmes: Inferring the orbit

- Likelihood for a single image pointing:

$$\begin{aligned} p(\boldsymbol{\alpha}_i|\Omega_i,\omega,\theta) &= p_{\rm good}\, p_{\rm fg}(\boldsymbol{\alpha}_i|\Omega_i,\omega,\theta) \\ &\quad +[1-p_{\rm good}]\, p_{\rm bg}(\boldsymbol{\alpha}_i) \end{aligned}$$

- Background component:

$$p_{\rm bg}(\boldsymbol{\alpha}_i) = [4\pi]^{-1}$$

- The image could come from anywhere on the sky, regardless of the comet position or time; describes random images that made it into our data set (via a false positive from Yahoo Image Search)

Comet Holmes: Inferring the orbit

- Foreground component:

$$p_{\mathrm{fg}}(\boldsymbol{\alpha}_i|\Omega_i,\omega,\theta) = \int p(\boldsymbol{\alpha}_i|t_i,\Omega_i,\omega,\theta)\,p(t_i|\Omega_i,\theta)\,dt_i$$

- (we need to marginalize over the time the image was taken, which requires using a *prior*; more on this soon. . .)
- The time-dependent likelihood is:

$$p_{\mathrm{fg}}(\boldsymbol{\alpha}_i|t_i,\Omega_i,\omega,\theta) = \begin{cases} [\eta\,\Omega_i]^{-1} & \text{comet in } \eta \text{ sub-image} \\ 0 & \text{comet not in } \eta \text{ sub-image} \end{cases}$$

- The comet can be anywhere inside the central η fraction of the image; η accounts for photographers putting their subjects near the center of the image.

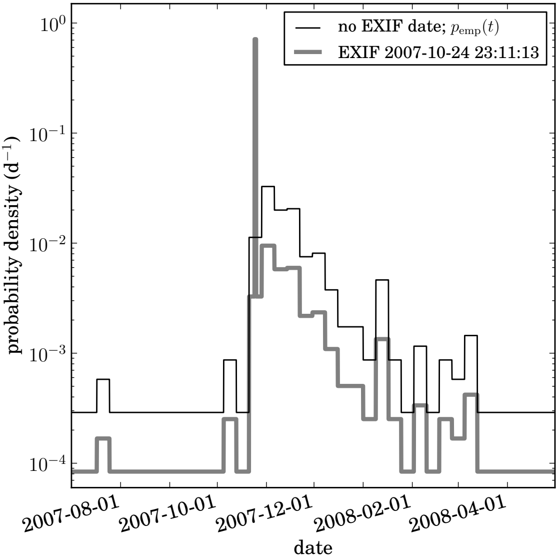
Comet Holmes: Inferring the orbit

- The time prior ends up being crucial; flat is not good
- We cheat by building an empirical prior p_{emp} (we should fit a flexible model instead)
- p_{emp} is the histogram of EXIF timestamps, regularized by adding 1 to each bin and normalizing
- If an image has an EXIF timestamp, we use it but also hedge our bets by mixing in a fraction of p_{emp}:

$$p(t_i|\boldsymbol{\Omega}_i, \theta) = p_{\mathrm{EXIF}}\, p(t_i|t_{\mathrm{EXIF}}) + [1 - p_{\mathrm{EXIF}}]\, p_{\mathrm{emp}}(t_i)$$

- where $p(t_i|t_{\mathrm{EXIF}})$ is 1 for times within 12 hours of the timestamp, 0 otherwise.

Comet Holmes: Inferring the orbit



Comet Holmes: Inferring the orbit

- All together now:

$$p(\boldsymbol{\alpha}_i|\boldsymbol{\Omega}_i, \omega, \theta) = \frac{1 - p_{\mathrm{good}}}{4\pi} + \frac{p_{\mathrm{good}}}{\eta\Omega_i} \int \mathrm{InImage}(\Omega_i, \eta, \boldsymbol{\omega})\, p(t_i|\boldsymbol{\Omega}_i, \theta)\, dt_i$$

- If the image does not have EXIF,

$$p(t_i|\boldsymbol{\Omega}_i, \theta) = p_{\mathrm{emp}}(t_i)$$

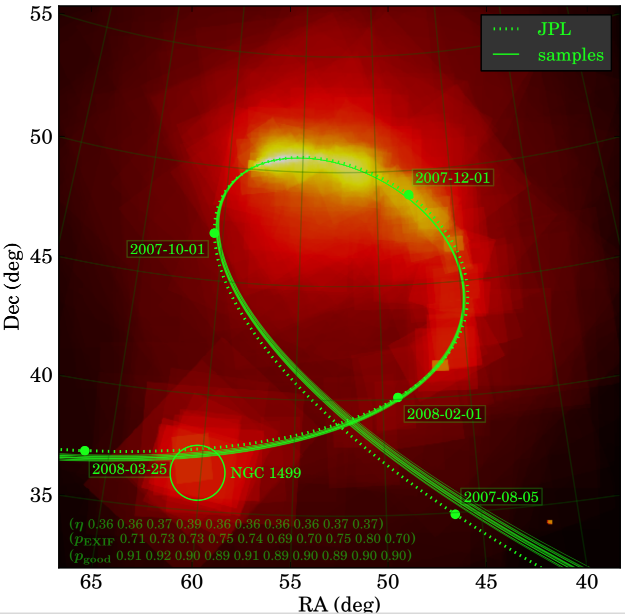
- If it does have EXIF,

$$p(t_i|\boldsymbol{\Omega}_i, \theta) = (1 - p_{\mathrm{EXIF}})\, p_{\mathrm{emp}}(t_i) + p_{\mathrm{EXIF}} \left[\mathrm{abs}(t_i - t_{\mathrm{EXIF}}) < \frac{1}{2}\right]$$

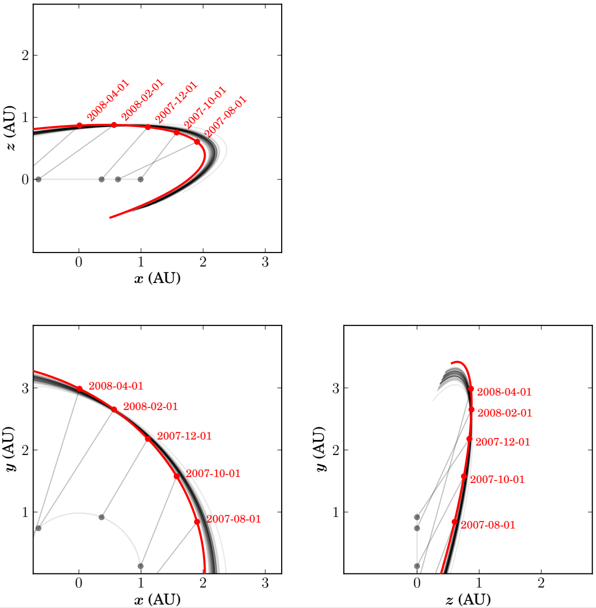
Comet Holmes: Inferring the orbit

- For the inference we use emcee, a great affine-invariant MCMC sampler by Dan Foreman-Mackey
- Initialize at the median time, velocity from images within a week, and heuristic radius (1 AU)
- Results are best shown as a sampling of trajectories

Comet Holmes: Inferring the orbit



Comet Holmes: Inferring the orbit



Comet Holmes: Results

- With just the search phrase “Comet Holmes”, we can get the comet’s path through the sky, with timestamps
- From this, we can determine the orbital elements that describe its 3-d orbit
- . . . and all this without even locating the comet in the images!
- This is a demonstration of the (huge) amount of astronomical imaging available on the web

Comet Holmes: Hyperparams

- We are modeling the behavior of astrophotographers (in pointing their cameras)
- η tells us how people frame their photos: we find it to be about 37%—Comet Holmes gets placed in the middle 60% $\times$ 60% of the image
- p_{EXIF} tells us how accurately people set their camera clocks—a third of images have EXIF timestamps, and about 75% of the time they're correct
- p_{good} tells us about the purity of images we get from Yahoo Image Search and that pass calibration by *Astrometry.net* (this will depend on the uniqueness of the search term!); 90% for "Comet Holmes"

Comet Holmes: Is this Citizen Science?

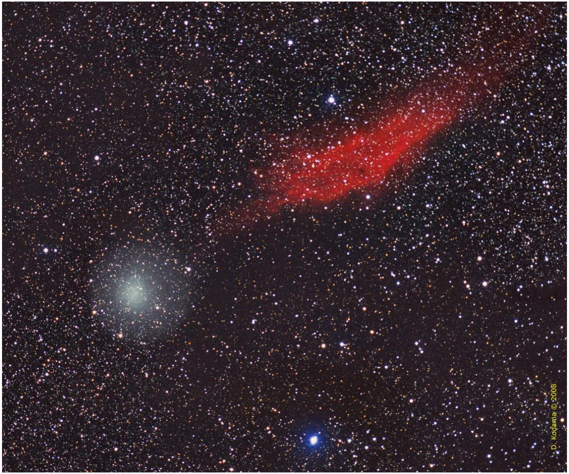
- Different from most “Citizen Science” projects (GalaxyZoo, SETI@Home, AAVSO, MicroFUN): the astrophotographers did not opt in to this study; they were unwitting participants (several years after the fact)
- We didn’t even attempt to get permissions for the > 2000 images we touched in this study — are we justified in using these data?
- We did contact the owners of the example images shown, and they were very enthusiastic

Comet Holmes: Web data issues

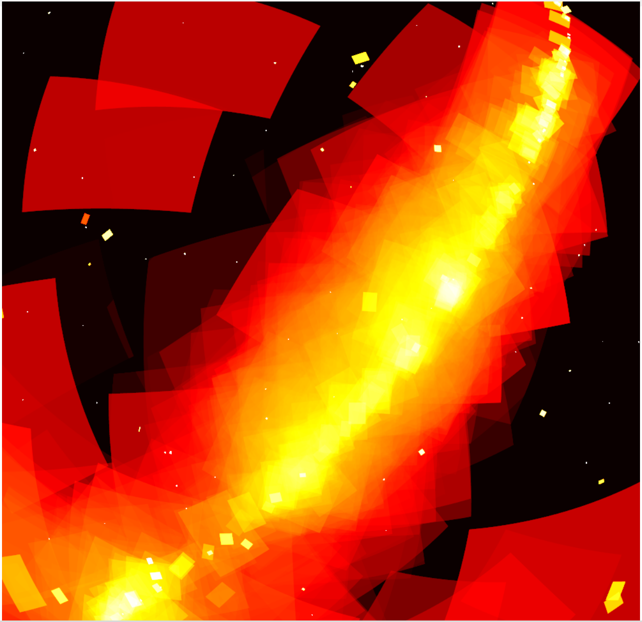
- This project gives a snapshot (as of 2007) of astro images posted on the web
- The web offers a huge volume of image data (opportunities for serendipity in time-domain work) . . .
- . . . but issues of permissions, detailed calibration, image independence, spoofing, . . .
- Commercial image search APIs have an uncertain future: Yahoo! Image Search API is shut down; Google Image Search API is limited; Bing Image Search API lives. Repeatable science?
- Could we build an Open Source Sky Survey from images contributed by amateur astronomers?

Comet Holmes: Photographer model

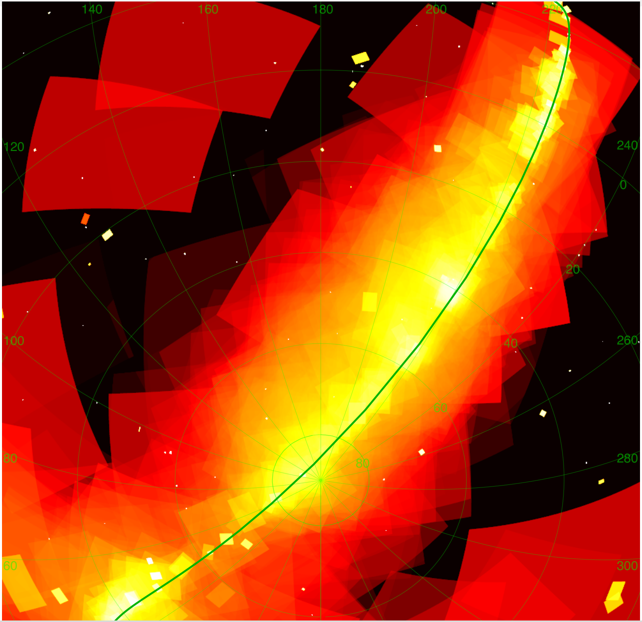
A more sophisticated model of what photographers photograph and how they frame their images could help remove some biases (eg, conjunctions are popular!)



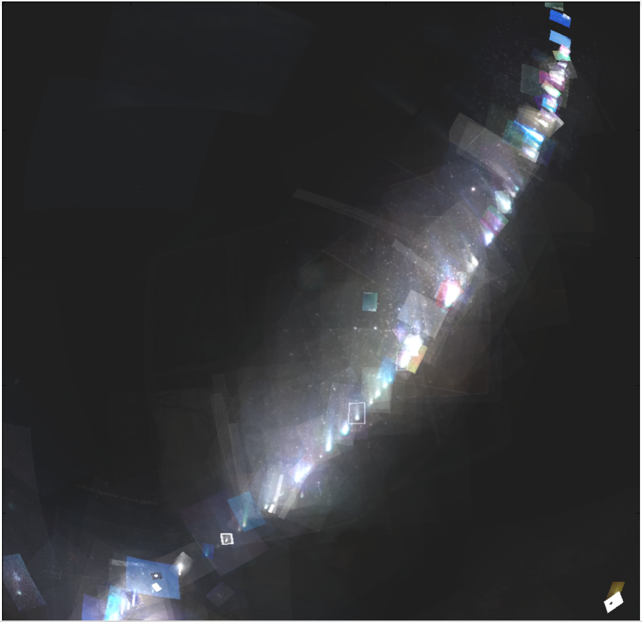
Comet Hyakutake: Eye candy



Comet Hyakutake: Eye candy



Comet Hyakutake: Eye candy



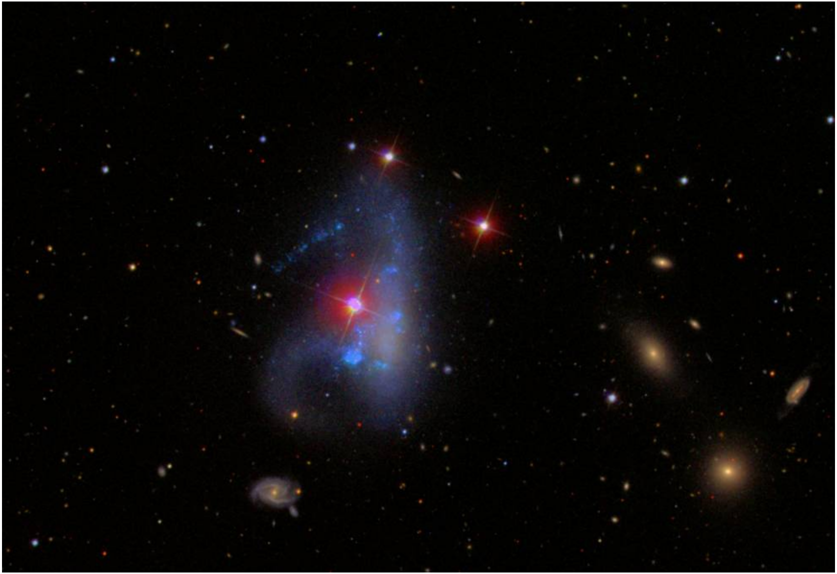
Astrometry.net

Enhance!

Astrometry.net

The goal: recognition of astronomical images

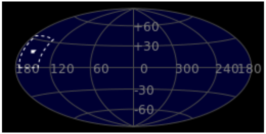
You give us an astronomical image

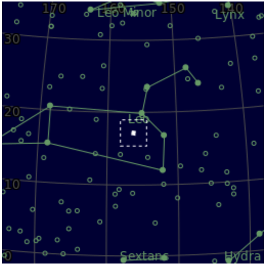


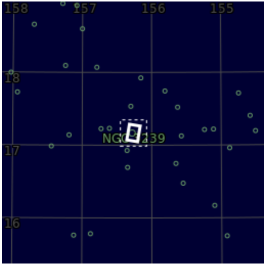
The goal: recognition of astronomical images

We tell you where your telescope was pointing

location, scale, and rotation—World Coordinate System (WCS)



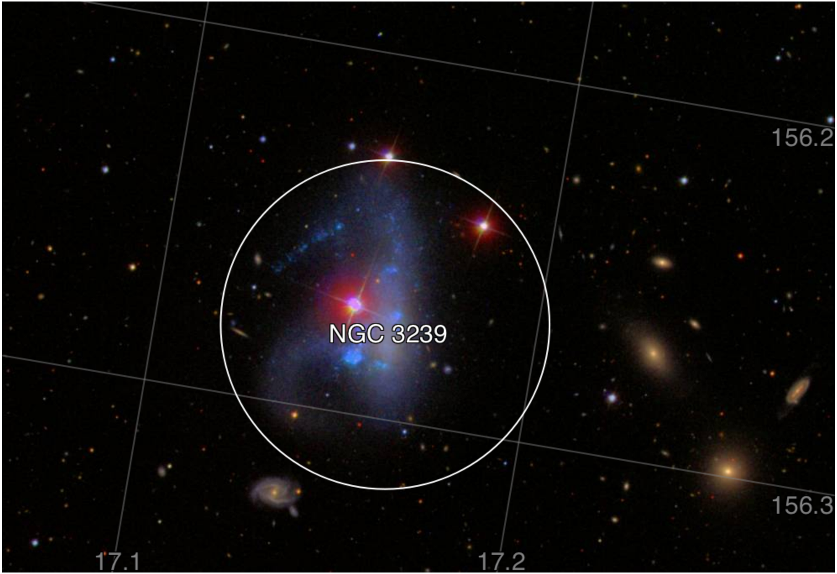





```
SIMPLE  =                    T / Standard FITS file
BITPIX  =                    8 / ASCII or bytes array
NAXIS   =                    0 / Minimal header
CTYPE1  = 'RA---TAN' / TAN (gnomic) projection
CTYPE2  = 'DEC--TAN' / TAN (gnomic) projection
CRVAL1  =        156.232948081 / RA  of reference point
CRVAL2  =        17.2082104082 / DEC of reference point
CRPIX1  =        700.032424927 / X reference pixel
CRPIX2  =        245.767471313 / Y reference pixel
CD1_1   =   -3.67848594254E-05 / Transformation matrix
CD1_2   =     0.00021821611049 / no comment
CD2_1   =     0.00021821611049 / no comment
CD2_2   =    3.67848594254E-05 / no comment
CUNIT1  = 'deg     ' / X pixel scale units
```

The goal: recognition of astronomical images

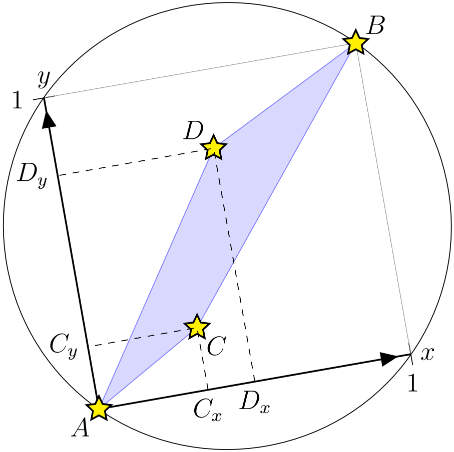
...and tell you what's in your image



Astrometry.net: The approach

- Match features in the image to features in an index
  - Features are based on the geometric arrangement of small groups of stars
  - The index maps features to places on the sky
- Each feature match is a hypothesis about the alignment of the image on the sky
- Verify the hypotheses to reject false matches

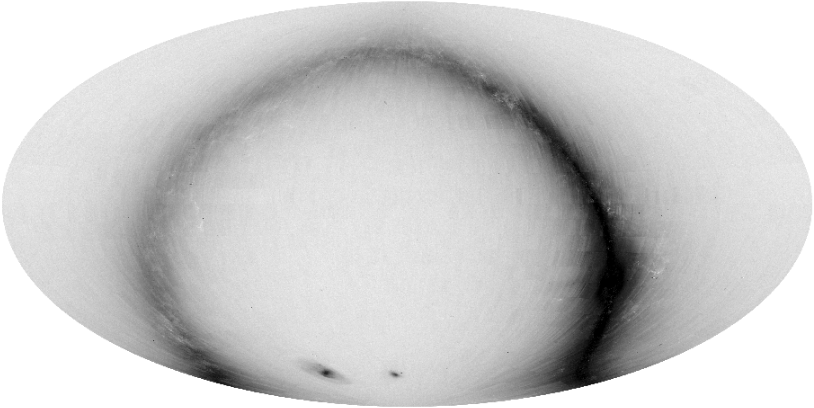
Astrometry.net: The geometric feature



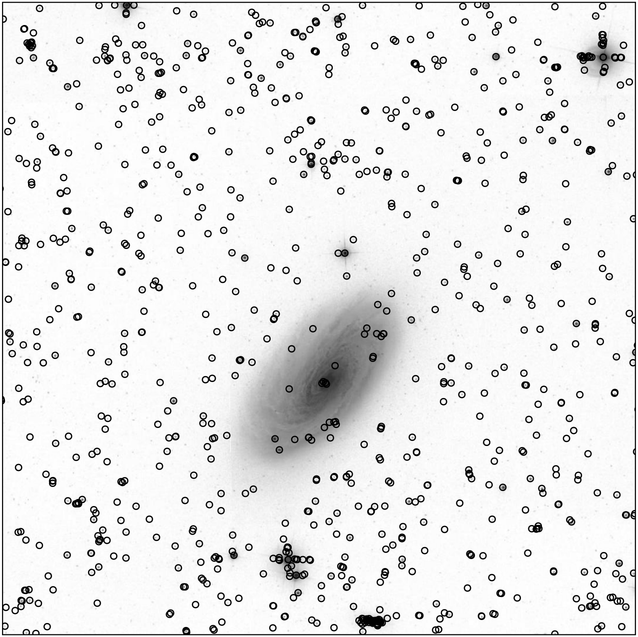
- Four-star features
- Two most distant stars are labelled *A*, *B*
- They establish a local coordinate frame
- Two others stars are labelled *C* and *D*
- Their positions in the local coordinate frame become the feature descriptor, (C_x, C_y, D_x, D_y)
- Has the invariances we need: translation, scale,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

Astrometry.net: Indexing

We start with a reference catalog, 2MASS, containing half a billion objects:

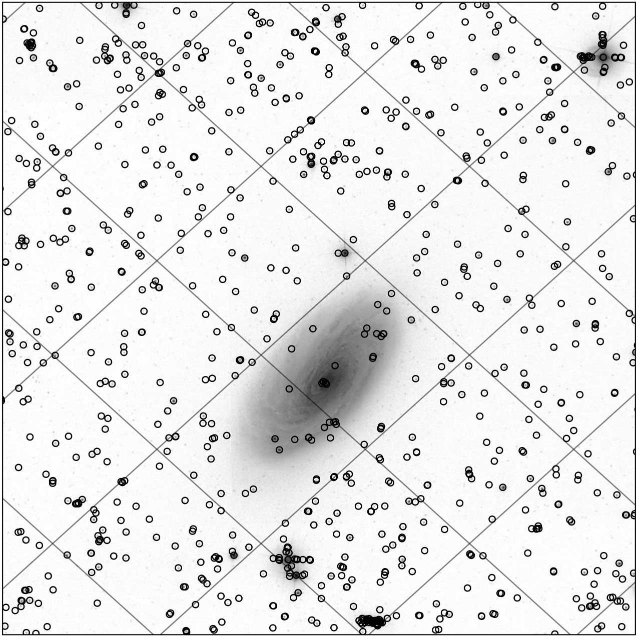


Astrometry.net: Indexing



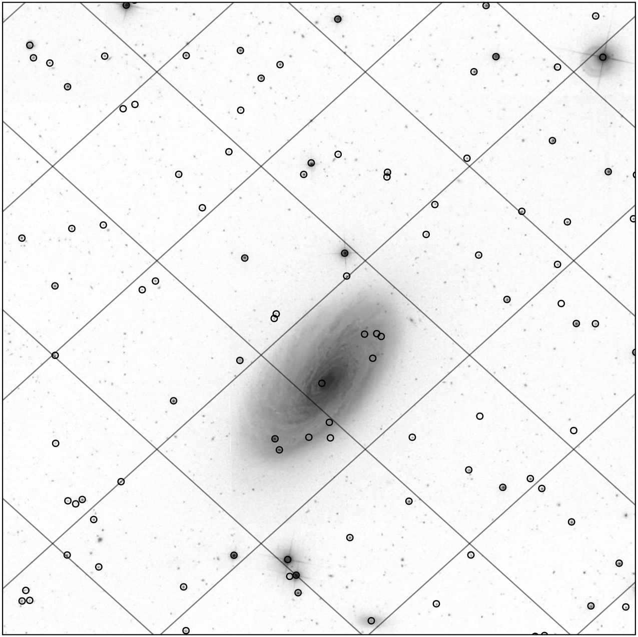
- Start with 2MASS reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- …build *N* geometric features in each cell

Astrometry.net: Indexing



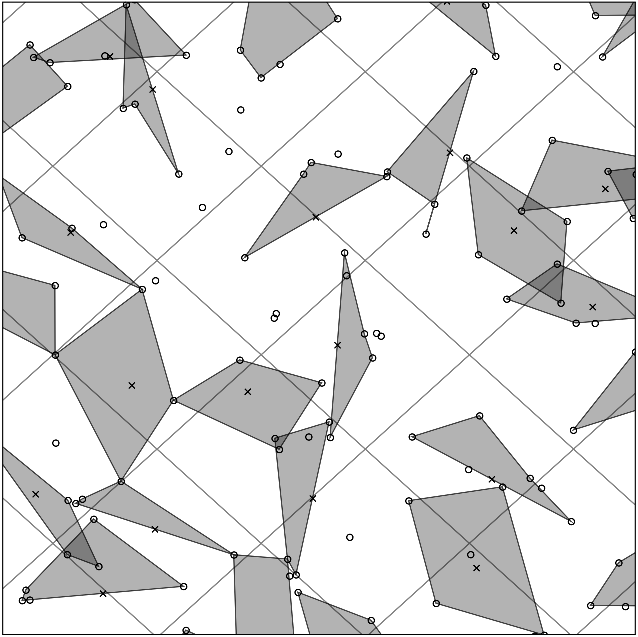
- Start with 2MASS reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- …build *N* geometric features in each cell

Astrometry.net: Indexing



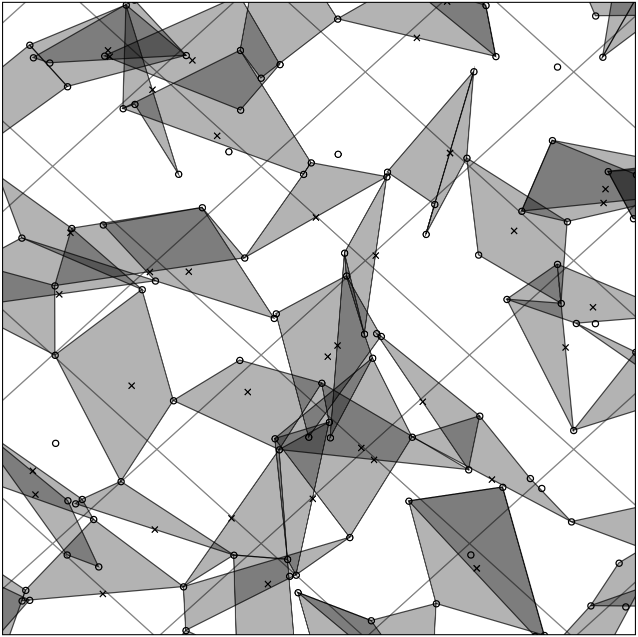
- Start with 2MASS reference catalog
- Place a grid over the sky
- Select n brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- …build N geometric features in each cell

Astrometry.net: Indexing



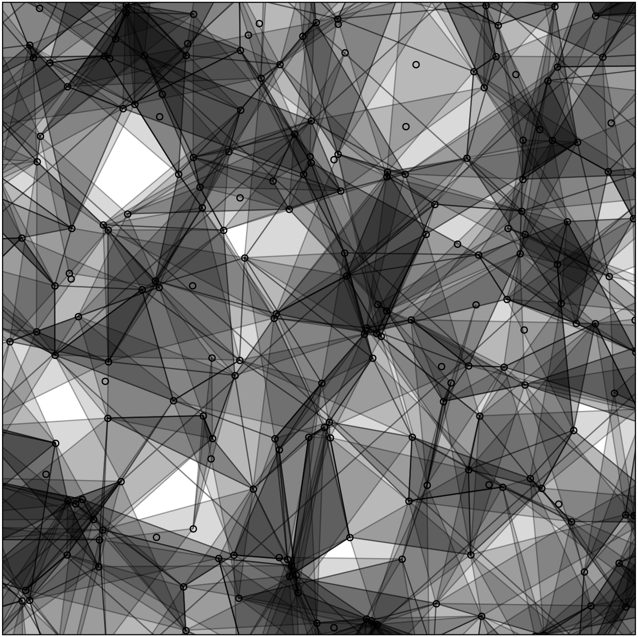
- Start with 2MASS reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

Astrometry.net: Indexing



- Start with 2MASS 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



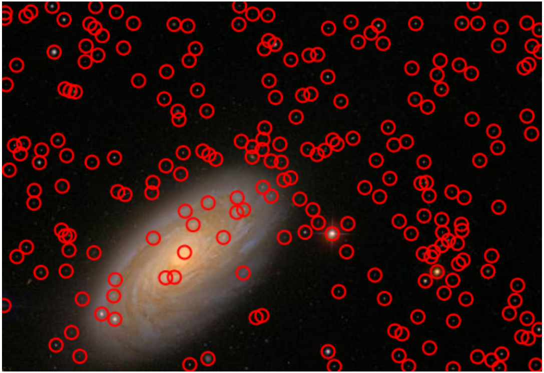
- Start with 2MASS reference catalog
- Place a grid over the sky
- Select *n* brightest stars in each cell
- Build a geometric feature in each cell
- Build another geometric feature in each cell
- . . . build *N* geometric features in each cell

Astrometry.net: Test time [1/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index

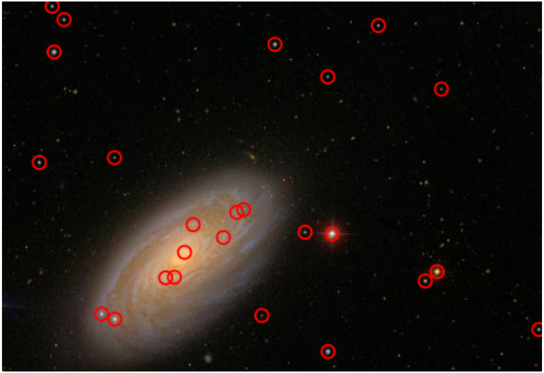
Astrometry.net: Test time [2/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



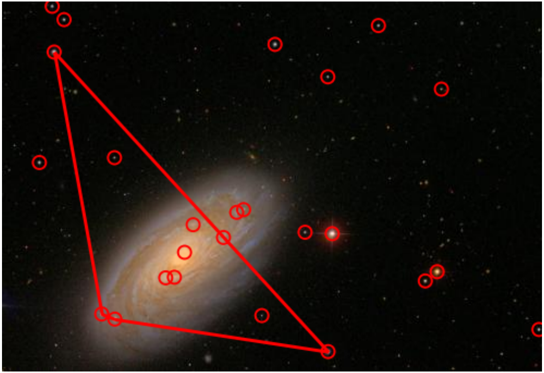
Astrometry.net: Test time [3/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



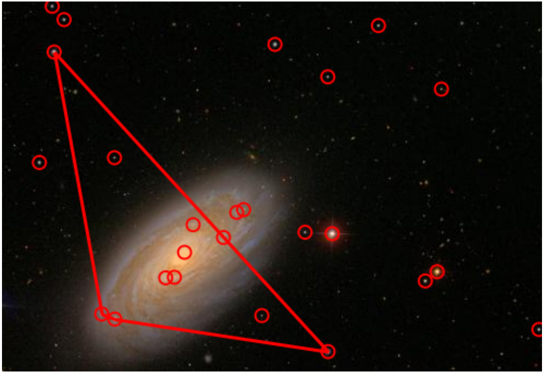
Astrometry.net: Test time [4/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index

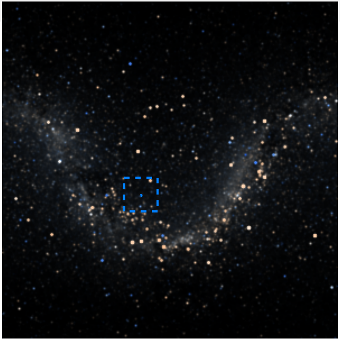


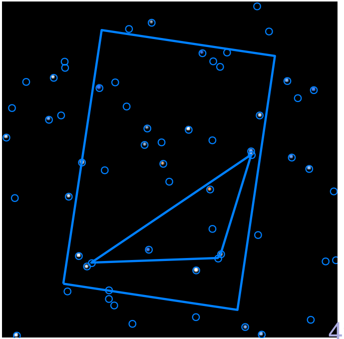
Astrometry.net: Test time [5/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



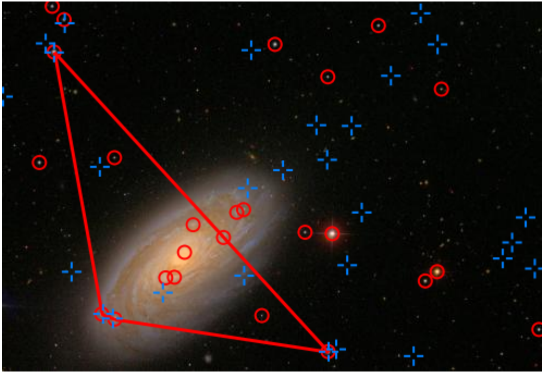
Match #1 of 1



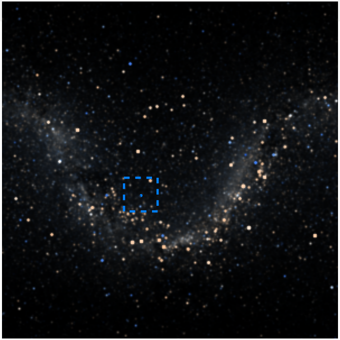


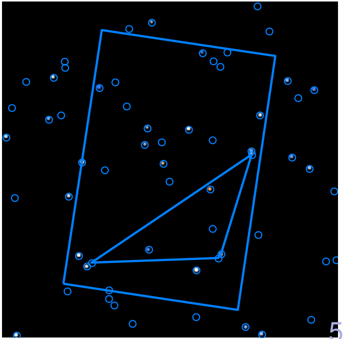
Astrometry.net: Test time [6/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at a geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



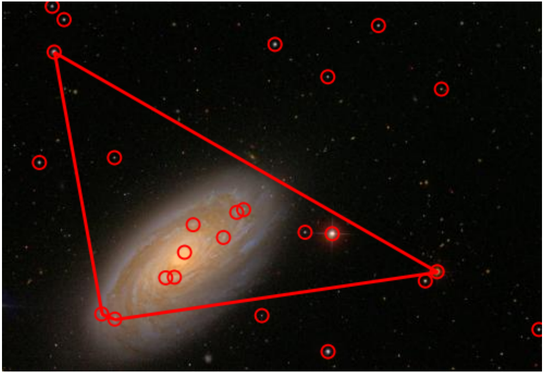
Match #1 of 1

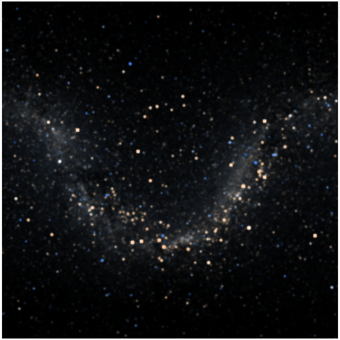




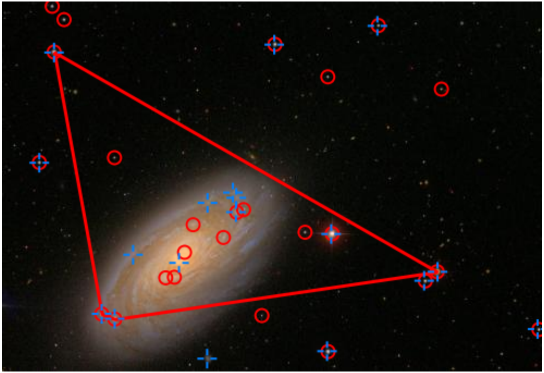
Astrometry.net: Test time [7/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index

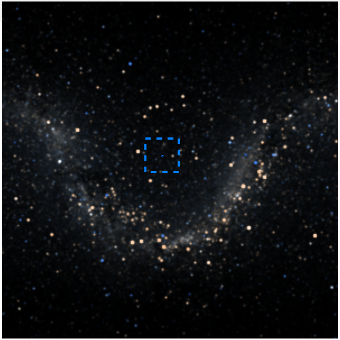


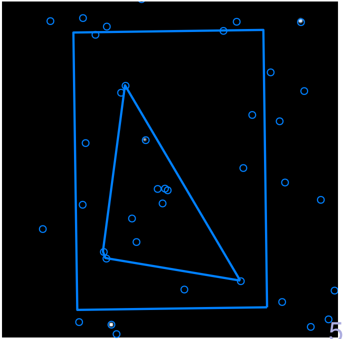


- Detect stars
- Starting with the brightest stars . . .
- Look at another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



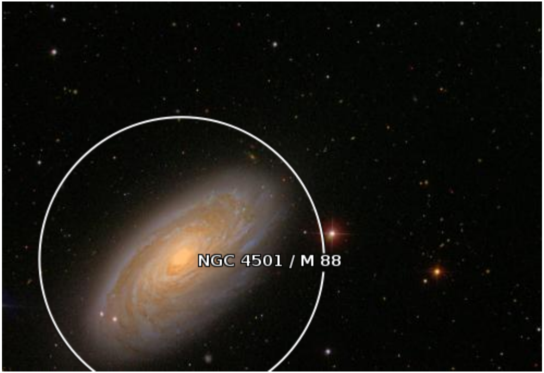
Match #2 of 2



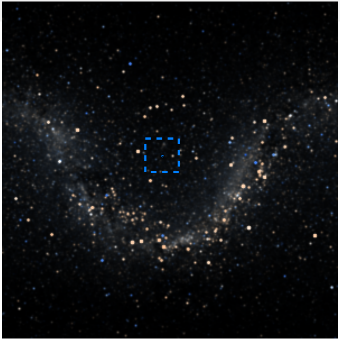


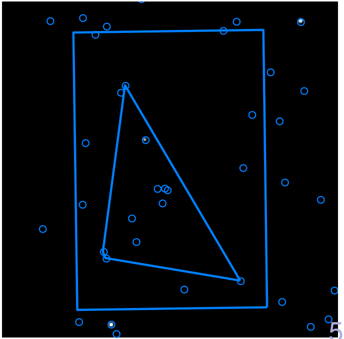
Astrometry.net: Test time [9/9]

- Detect stars
- Starting with the brightest stars . . .
- Look at another geometric feature
- Find matching features in the index
- Check each match by looking for alignment with other stars in the index



Match #2 of 2





Astrometry.net : Demo

Astrometry.net: Performance

- Tested on $330,000$ images from the Sloan Digital Sky Survey (SDSS)
- Correctly recognized over 99.95 %
- The 0.05 % are mostly interesting flaws in the "ground truth" reference catalog
- No false positives: we are either correct or give no-answer
- Most images take less than 1 second of CPU time (given strong hints about the image scale)

Astrometry.net: Users

- Open-source code, web version, and Flickr robot (thousands of users; running since 2007 April)
- Photographic archives: the DASCH project is scanning 500,000 photographic glass plates (taken 1880 to 1985) from the Harvard archives — a data set that would be near-impossible to use otherwise
- LSST; AMS-2 aboard the International Space Station
- Web API and a Python client: *astrobin.com*; *PinPoint*; and observatory control software

Astrometry.net: Summary

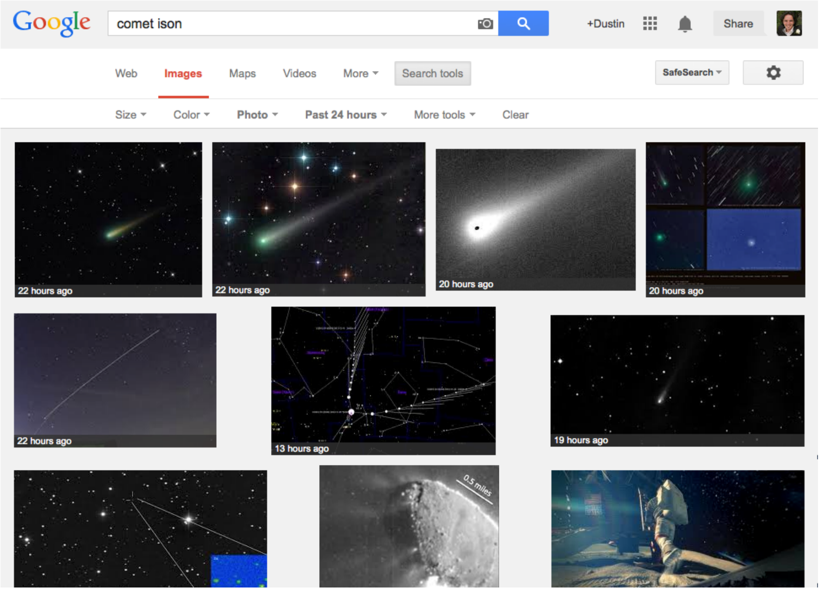
- We use geometric hashing, a classic idea in computer vision
- Uses a feature that captures the relative geometric arrangement of stars
- Match features to pre-computed features in an index
- For each match, predict the positions of others stars
- Accept or reject using Bayesian decision theory

Thanks!

Time for questions and discussion!

Enhance!

The Web has tons of astronomical images



The Web has tons of astronomical images

(And more than a few loonies)

Photometric calibration

- Wouldn't it be great to be able to use all the astronomical images on the Web for science?
- But many pretty pictures have had non-linear mappings applied
- Assuming monotonic non-linear mappings (stretches), ranks are preserved *(brightest pixel remains the brightest)*
- Work with ranks rather than absolute brightness
- Build a combined ranking: images vote on brightness ranking of pixels

Enhance!: setup

- Web Image search for: “NGC 5907” using Bing, Google, and Flickr Image Search APIs
- Run *Astrometry.net* on each
- Define target image (region of sky)
- Resample images to target frame

Enhance!: algorithm

- Initialize Combined image and Vote Count image.
- For each image:
  - Grab overlapping regions in the new image and Combined image
  - argsort (rank order) the pixels in each
  - Weighted average the ranks (using Vote Counts)
  - Reorder the Combined pixels using those ranks
  - Increment Vote Count

For $i = 1, \ldots, N$, do :

$$\sigma = \mathtt{argsort}(c[m_i]) \qquad \tau = \mathtt{argsort}(d_i[m_i])$$

$$\rho = \mathtt{argsort}\left(\frac{v[m_i] * \sigma^{-1} + \tau^{-1}}{v[m_i] + 1}\right)$$

$$c[m_i] = c[m_i]\left[\sigma[\rho^{-1}]\right] \qquad v[m_i] \mathrel{+}= 1$$

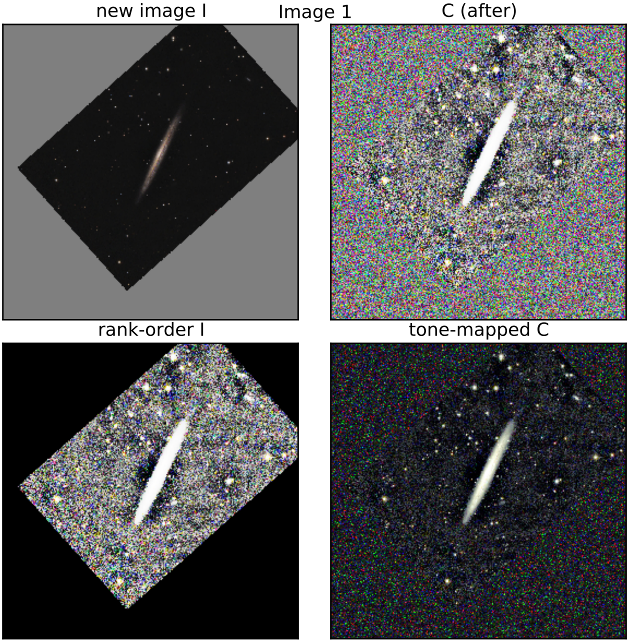


Image 1
new image I
C (after)
rank-order I
tone-mapped C

Image 2

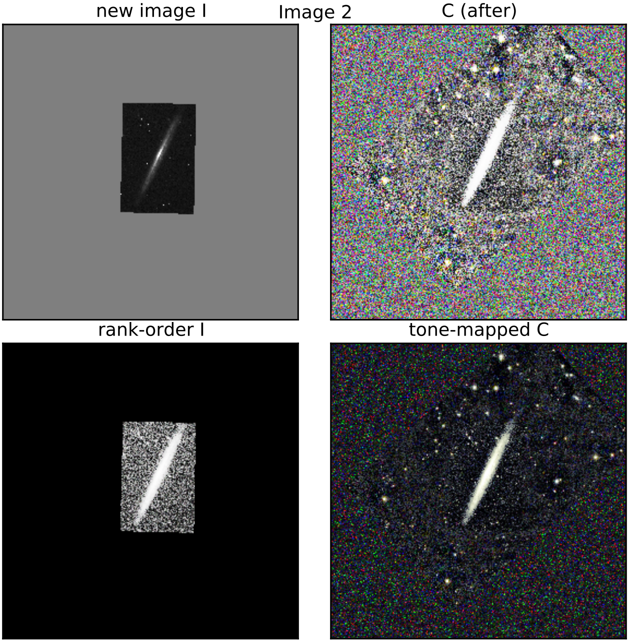


Image 3

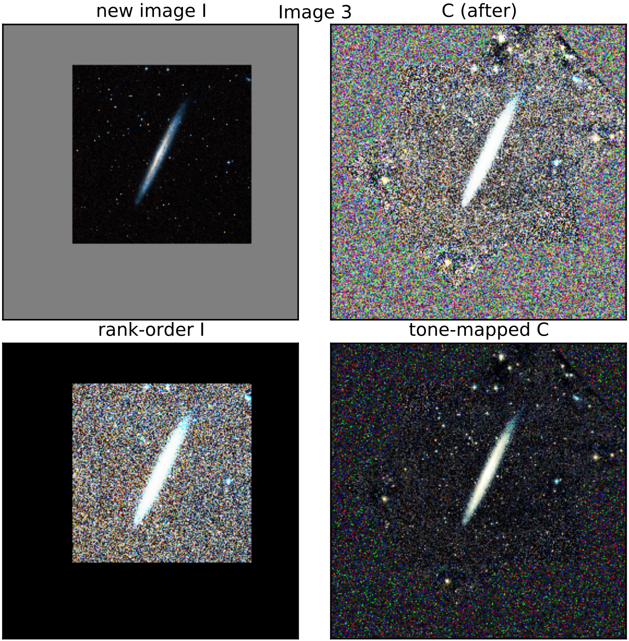


Image 4

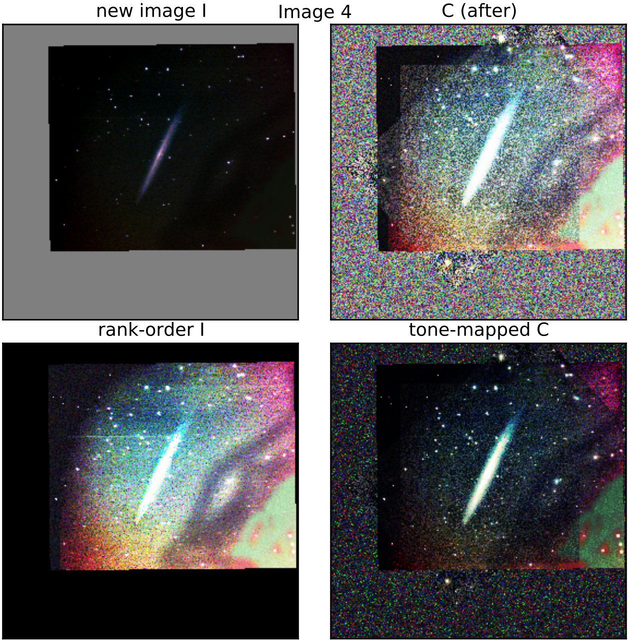


Image 5

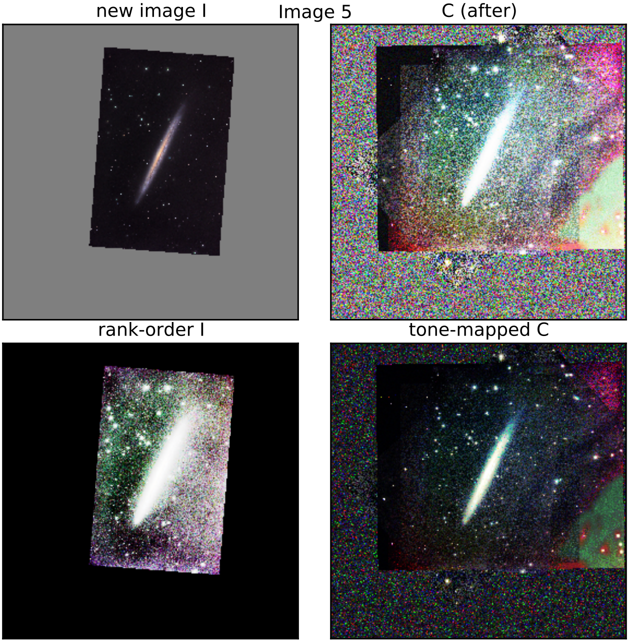


Image 6

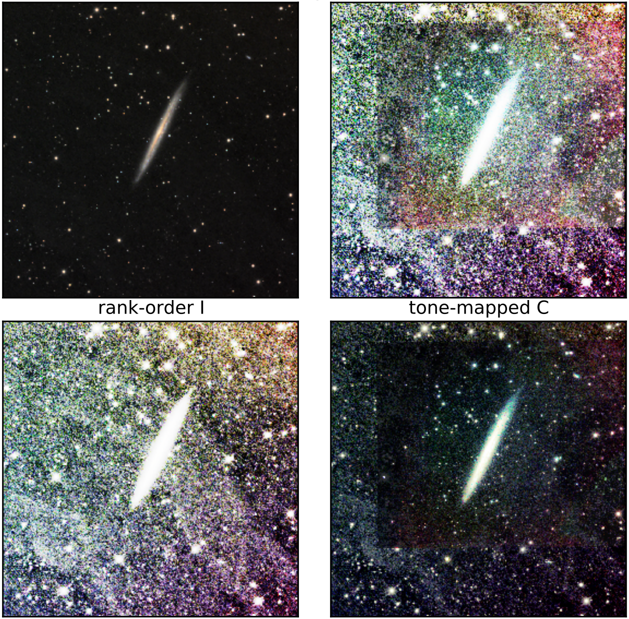


Image 7

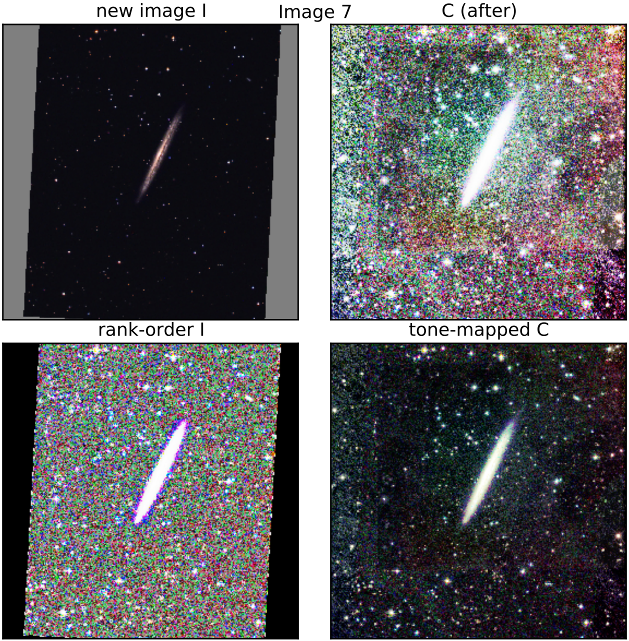


Image 8

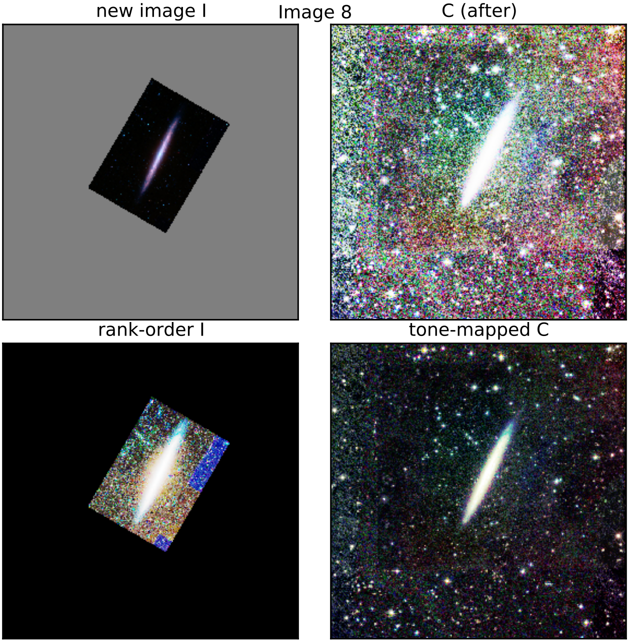


Image 9

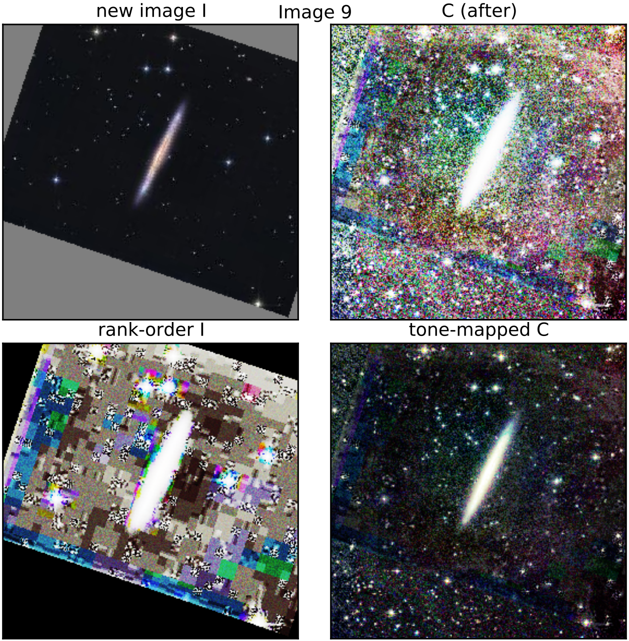
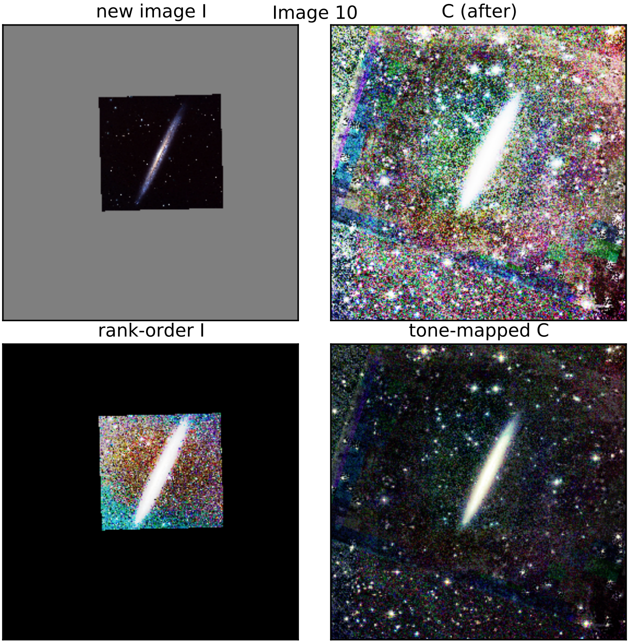


Image 10



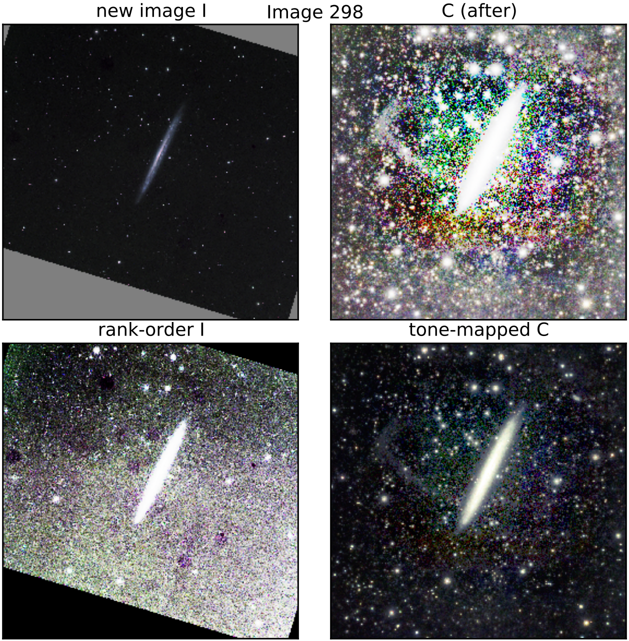
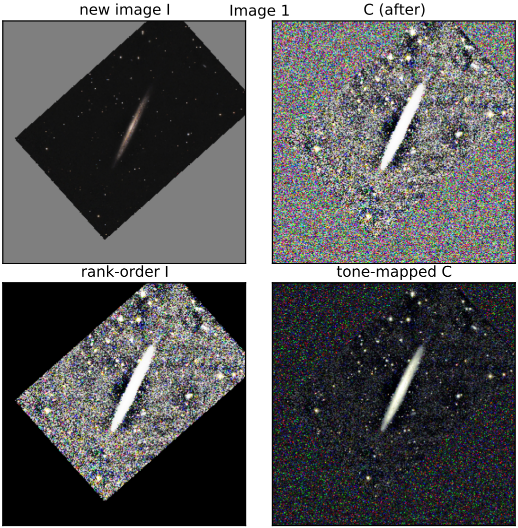


Image 298
new image I
C (after)
rank-order I
tone-mapped C

Enhance!: demo



Enhance!: results

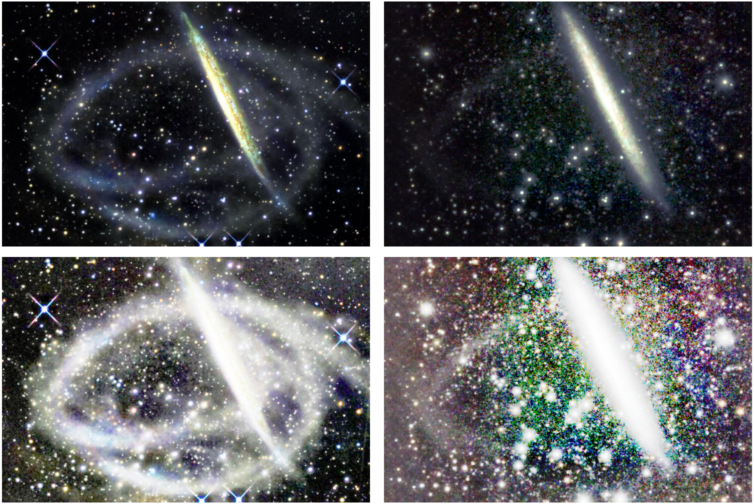


Image Credit & Copyright: R Jay Gabany (Blackbird Observatory); Martínez-Delgado *et al.* 2008

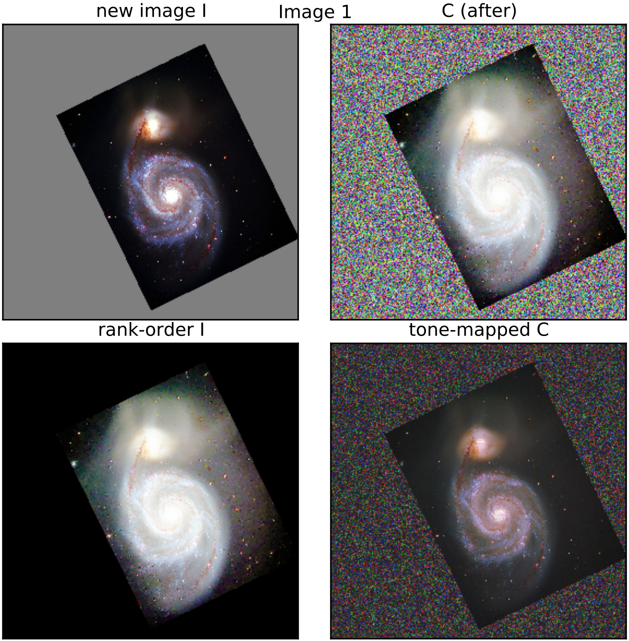


Image 1
new image I
C (after)
rank-order I
tone-mapped C

Image 2

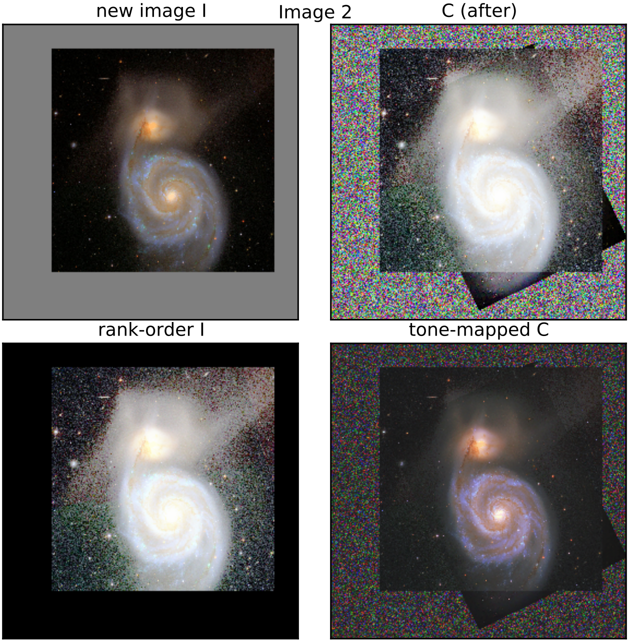


Image 3

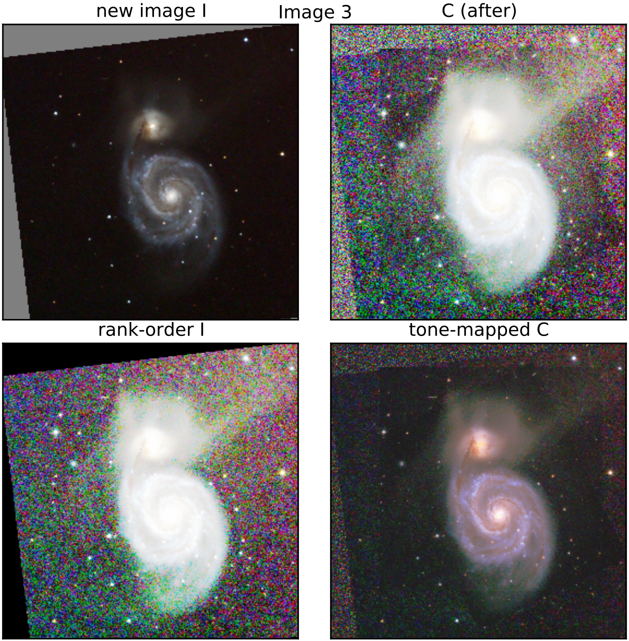


Image 4

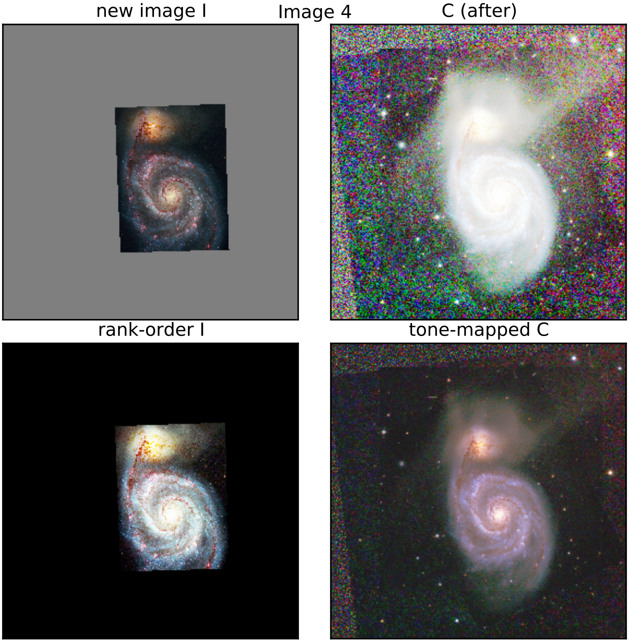
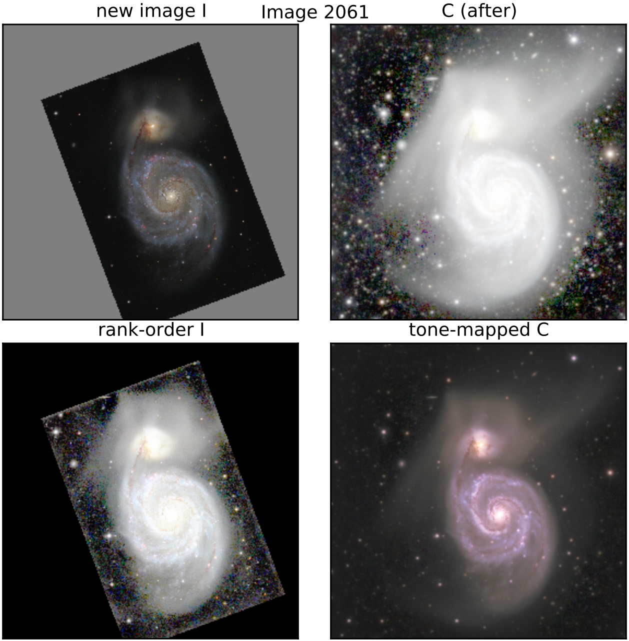
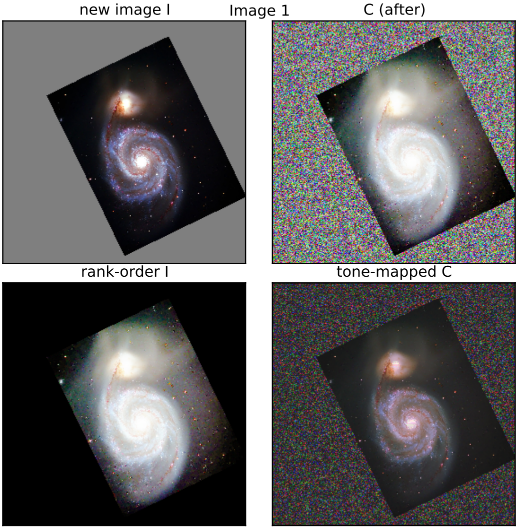


Image 2061



Enhance!: M51



Enhance!: discussion

- With tons of images, *Enhance!* does ok
- Web search *vs* contributed images; open source sky survey
- Compact sources vs surface brightness; PSF
- Could improve weighting (S/N; scale)
- Could improve outlier detection, duplicate detection
- Rank statistics allow us to handle images that have had unknown non-linear operations applied

Thanks!

Time for questions and discussion!

Comet Holmes: Image credits

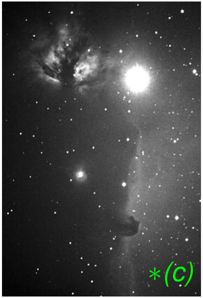
(a) copyright 2000–2005 Gods, Heroes, and Myth (`http://www.gods-heros-myth.com`);
(b) Paolo Berardi;
(c) Amateur Astronomers, Inc. Research Committee (`http://asterism.org`);
(d) Edward Emerson Barnard (in Ball, R. S., 1905, *A Popular Guide to the Heavens*, Van Nostrand);
(e) copyright P.-M. Hedén (`http://www.clearskies.se`);
(f) copyright Dave Kodama (`http://astrocamera.net`);
(g) TOC Observatory (`http://tocobs.org`);
(h) copyright Fay Saunders;
(i) NASA, JPL-Caltech, W. Reach (SSC-Caltech);
(j) copyright Julián Cantarelli;
(k) copyright 2007, 2008 John F. Pane (`http://holmes.johnpane.com`);
(l) Bruce Card, Aldrich Astronomical Society, Worcester MA;
(m) copyright Tyler Allred (`http://allred-astro.com`);
(n) Joe Orman;
(o) NASA, ESA, and H. Weaver (The Johns Hopkins University Applied Physics Laboratory), and A. Dyer, Alberta, Canada;
(p) copyright Vincent Jacques (`http://vjac.free.fr/skyshows`);
(q) Jimmy Westlake, Colorado Mountain College;
(r) Stephane Zoll (`http://astrosurf.com/zoll`);
(s) Babak Tafreshi / TWAN (`http://twanight.org`);
(t) Ivan Eder (`http://eder.csillagaszat.hu`);
(u) Vicent Peris (OAUV), José Luis Lamadrid (CEFCA);

Comet Holmes: Example images zoom-in

Zoom out

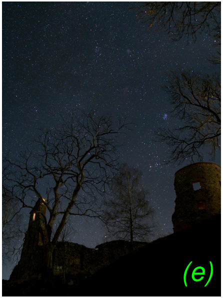
∗(a)

(b)

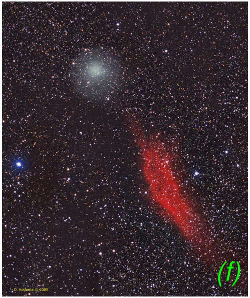


∗(c)

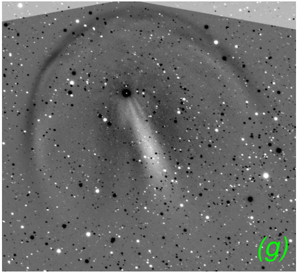
(d)



(e)



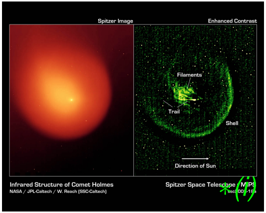
(f)

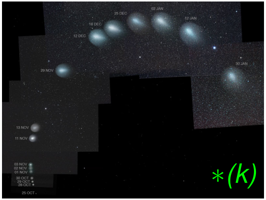


(g)

Comet Holmes: Example images zoom-in

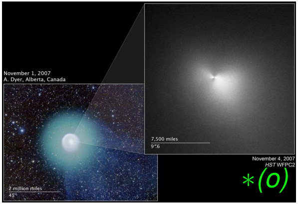
Zoom out

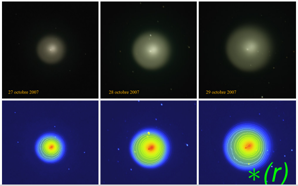




Comet Holmes: Example images zoom-in

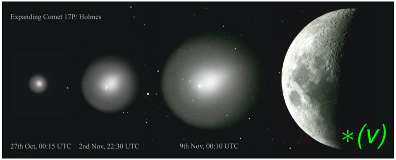
Zoom out

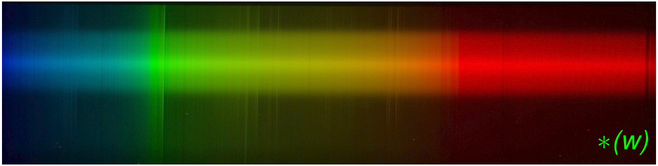




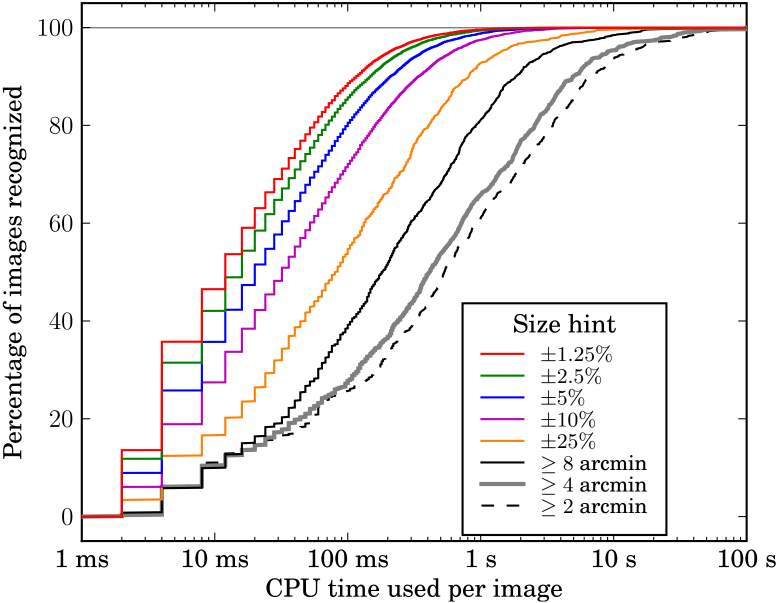
Comet Holmes: Example images zoom-in

Zoom out

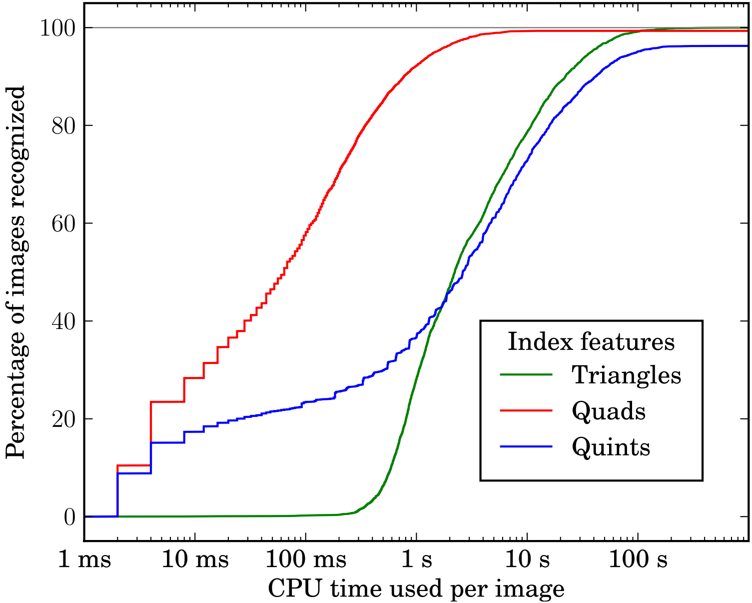




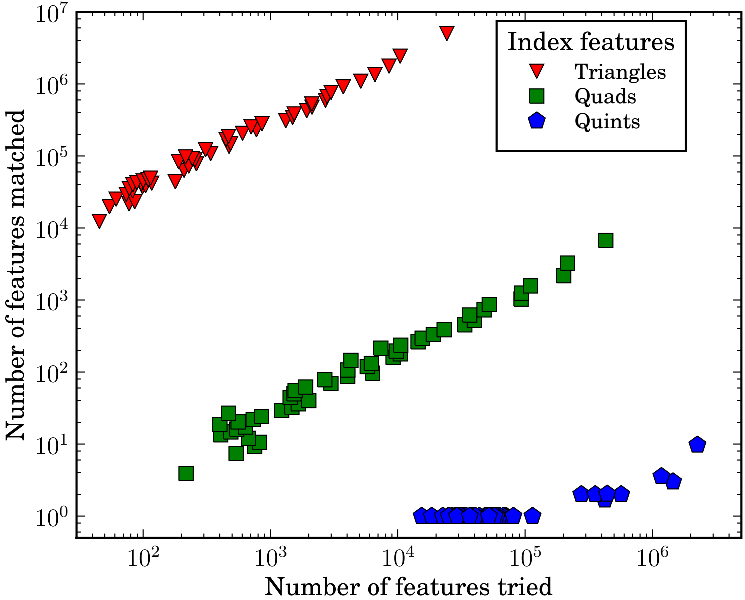
Astrometry.net: Performance — scale hints



Astrometry.net: Performance — triangles & quintuples



Astrometry.net: Performance — triangles & quintuples



Astrometry.net: Other data sets

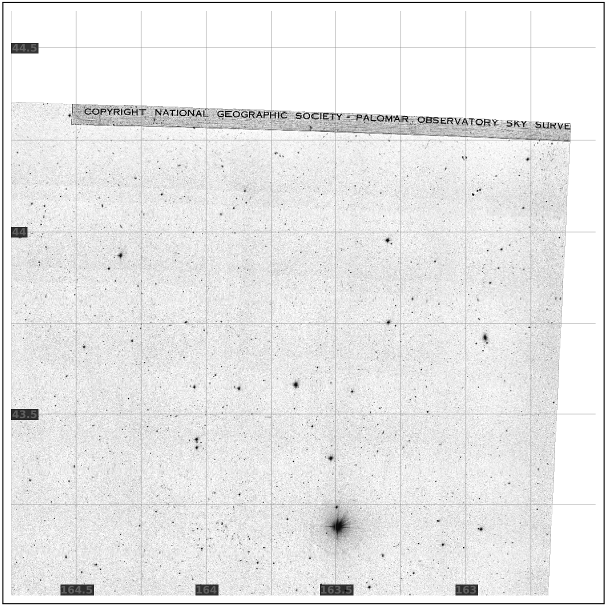
We get excellent results on SDSS images.

We've also tested:

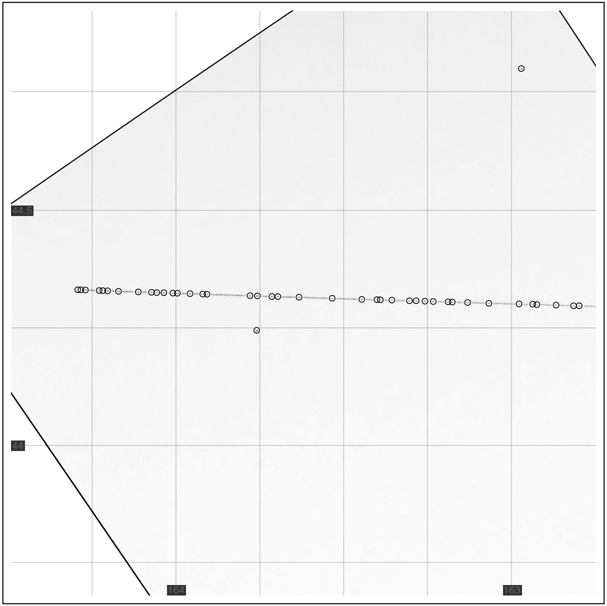
- images from GALEX, a space telescope that measures ultraviolet (UV): 99.7% recognition rate for near-UV
- images from the Hubble Space Telescope's Advanced Camera for Surveys — with a custom-built index — 100% success on a tiny sample of 191 images

False positive example

False positive example



False positive example



False positive example