university of toronto
department of computer science

Astrometry.net

Making the Sky Searchable:
From Pixels to WCS

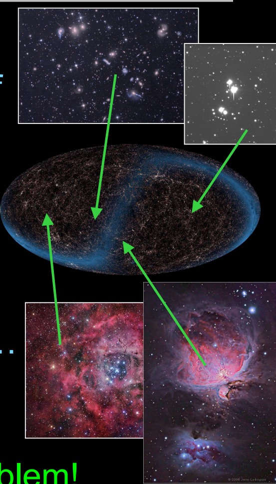
Sam Roweis
University of Toronto & Google

Dustin Lang & Keir Mierle
University of Toronto

Jon Barron, Michael Blanton & David Hogg
New York University

“Auto Calibration” of Astronomical Data

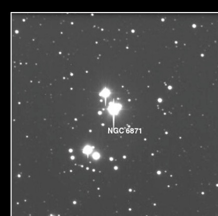
- Vision: Take every astronomical image ever taken in the history of the world and put correct astrometric headers on them.
- We want to include all modern professional telescope surveys plus all amateur photos, satellite images, historical plate archives…



- Astrometry is “meta-data”. Solve a really big data fusion problem!

Infer the Meta-Data from the Pixels!

- Start with just the pixels.
- Automatically infer the viewing parameters θ (plate scale, location of the image on the sky, bandpass, exposure date, telescope model,…)
- You think I’m joking, but I’m totally serious.
- All of your pixels are belong to us!

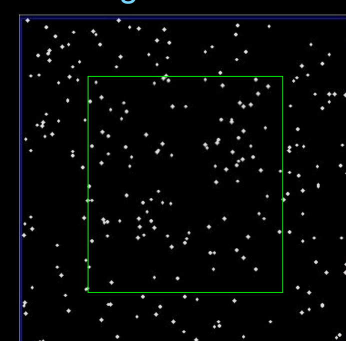


7 arcminutes

NGC 6871
1964

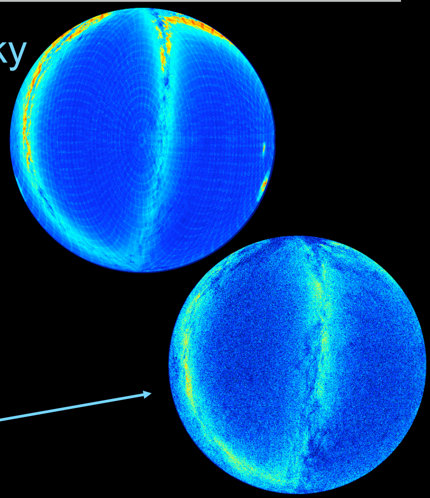
Simple model based on star catalogues

- We start with a catalogue of stars in the sky, and from it build a simple model which is used to calibrate (‘solve’) new test images.
- Goal: we can spend as much time as we want building the model but solving should be fast.
- Challenges:
  1) The sky is big.
  2) Both catalogues and pictures are noisy.



Catalogues: USNO-B 1.0 + TYCHO-2

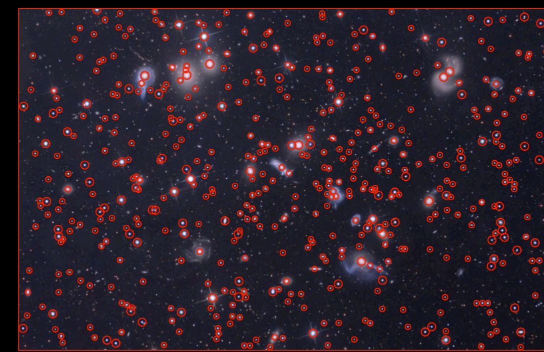
- USNO-B is an all-sky catalogue compiled from scans of old Schmidt plates. Contains about 10^9 objects, both stars and galaxies.
- TYCHO-2 is a tiny subset of 2.5M brightest stars.



Blind Calibration (1)

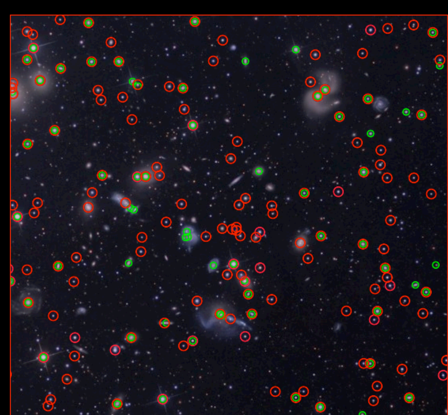
Preprocess the image to estimate the PSF, then identify "source locations" to finite precision, and work only with those.

Effectively, this creates a finite, but enormous set of images (about 1e300).

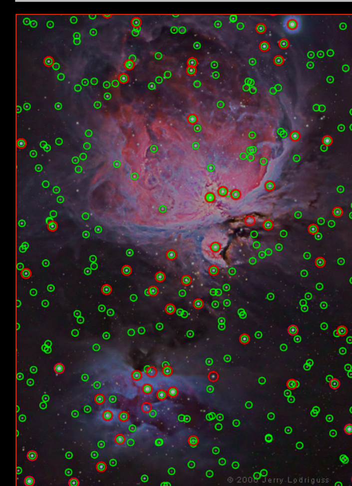


Blind Calibration (2)

Discretize space of hypotheses by considering sets of matchings between image sources and catalogue stars.



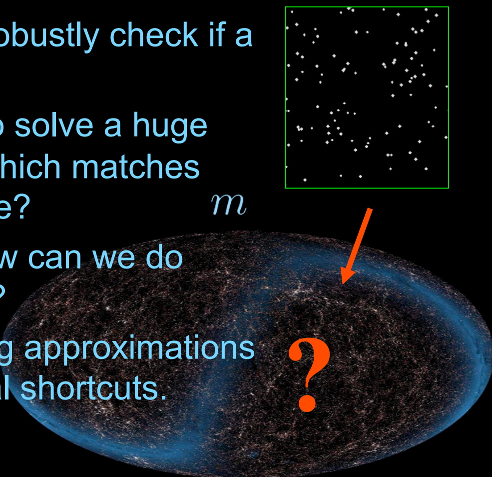
Verifying (checking) hypotheses



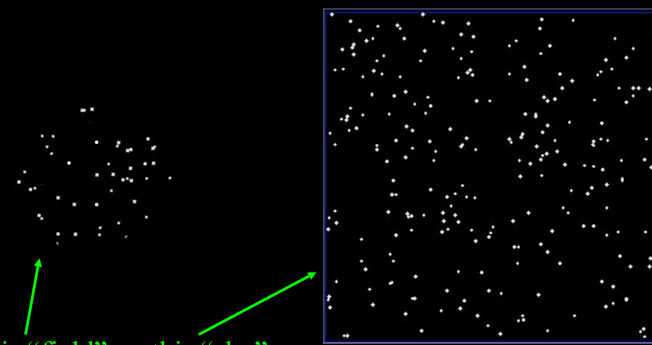
- For each potential match we find, we need to estimate the probability that it is correct: do we really have the correct alignment on the sky?
- Look at the number of catalog-object "matches" and compute the log-odds of getting that many hits if the query image were dropped on a random patch of sky.

Efficient Search for Solution

- We know how to robustly check if a match is correct.
- But we still have to solve a huge search problem: which matches should we examine?
- In other words, how can we do efficient inference?
- Separate modelling approximations from computational shortcuts.

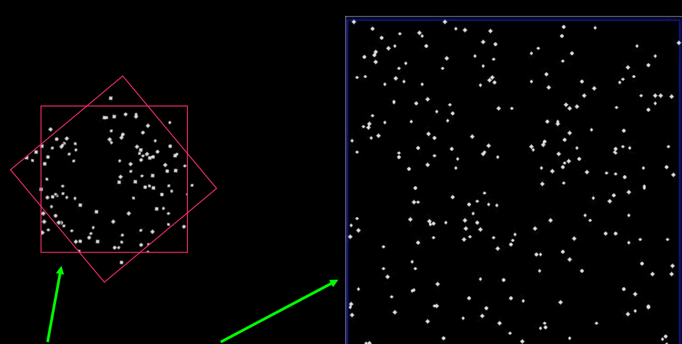


Example (a million times easier)



Find this "field" on this "sky".

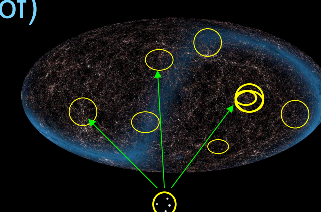
Example (a million times easier)



Find this "field" on this "sky".

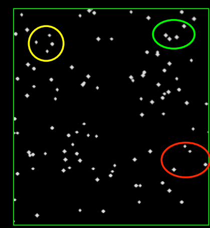
(Inverted) Index of Features

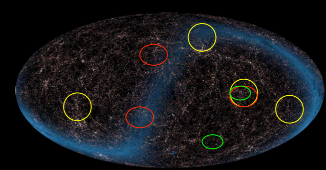
- To solve this problem, we employ the classic idea of an "inverted index".
- We define a set of "features" for any particular view of the sky (image).
- Then we make an (inverted) index, telling us which views on the sky exhibit certain (combinations of) feature values.
- The features in our inverted index act as "hash codes" for locations on the sky. (cf Bloom Filters)



Matching a test image

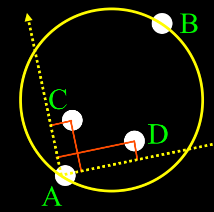
- When we see a new test image, we compute which features are present, and use our inverted index to look up which possible views from the catalogue also have those feature values.
- Each feature generates a candidate list in this way, and by intersecting the lists we can zero in on the true matching view.





"Quads" as Robust Features

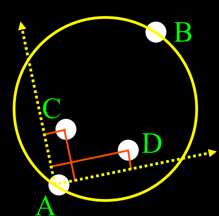
- We encode the relative positions of nearby quadruples of stars (ABCD) using a coordinate system defined by the most widely separated pair (AB).
- Within this coordinate system, the positions of the remaining two stars form a 4-dimensional code for the shape of the quad.
- Swapping AB or CD does not change the shape but it does "reflect" the code, so there is some degeneracy.



Continuous, vector-valued hash codes!

"Quads" as Robust Features

- This geometric hash code is invariant to scale, translation and rotation. It is continuous!
- It also has the property that if stars are uniformly distributed in space, codes are uniformly distributed in 4D.
- We compute codes for most nearby quadruples of stars, but not all; we require C&D to lie in the unit circle with diameter AB.

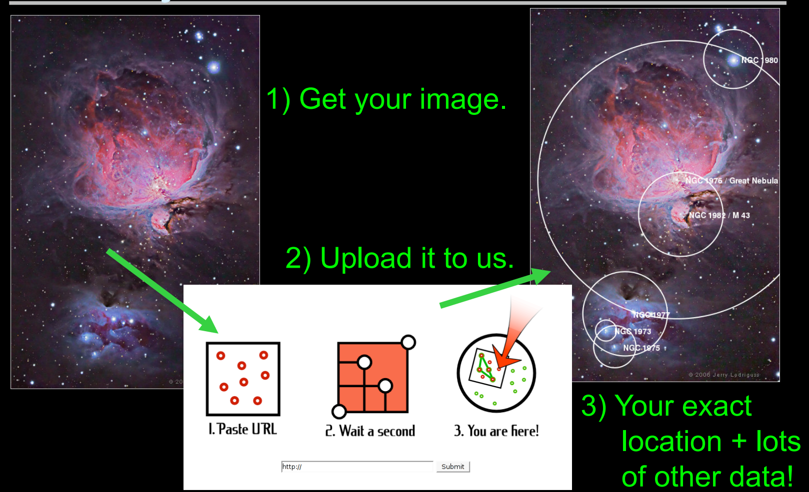


Continuous, vector-valued hash codes!

Summary: inference strategy

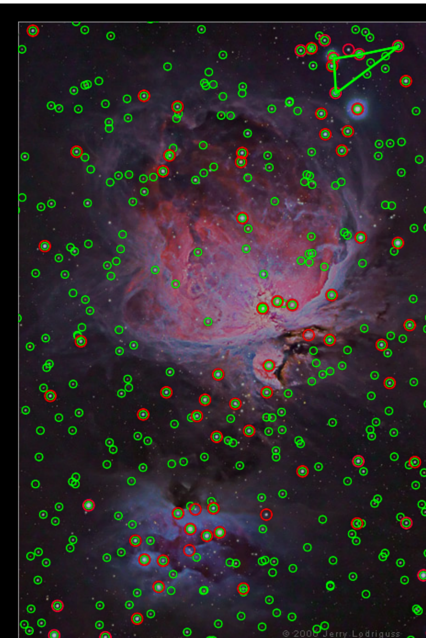
- Identify objects (stars+galaxies) in the image bitmap and create a list of their 2D positions.
- Cycle through all possible valid* quads (brightest first) and compute their corresponding codes.
- Look up the codes (in a code KD-tree) to find matches within some tolerance; this stage incurs some false positive and false negative matches.
- Each code match returns a candidate position & rotation on the sky. For each one, estimate the posterior by verifying the candidate using all objects in the image.

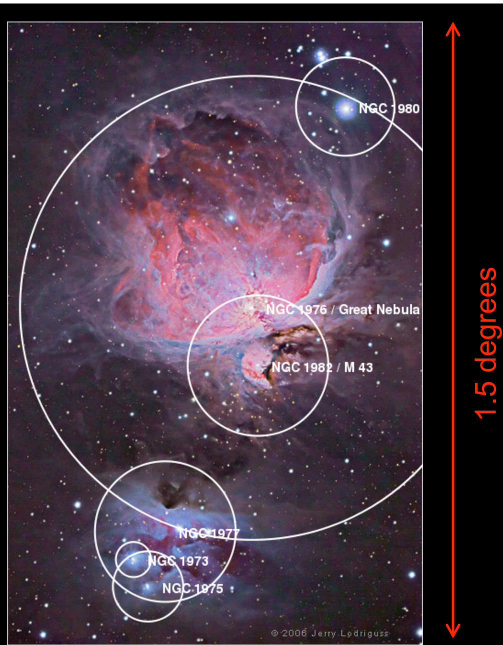
Our System is On the Web



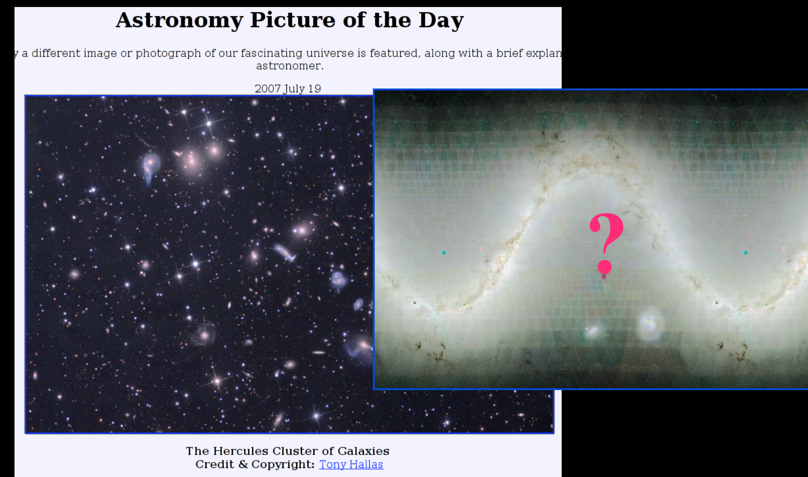
http://www.cs.toronto.edu/~roweis

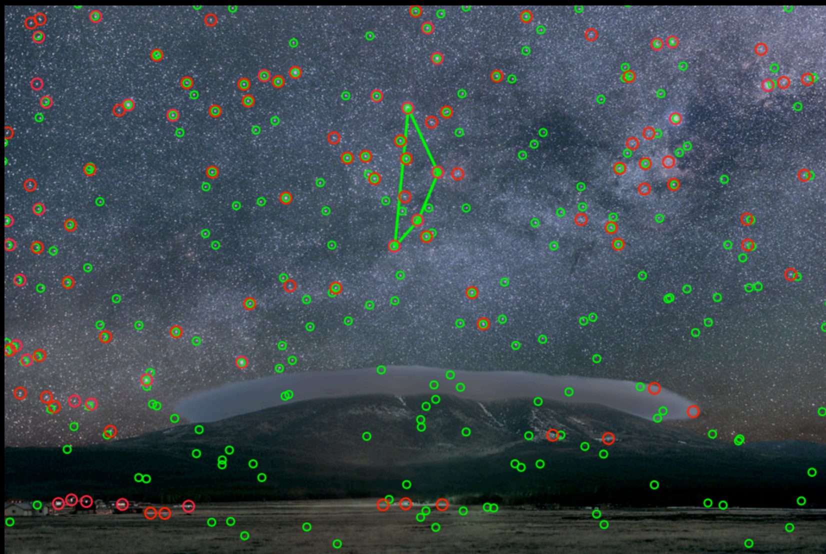
roweis@cs.toronto.edu



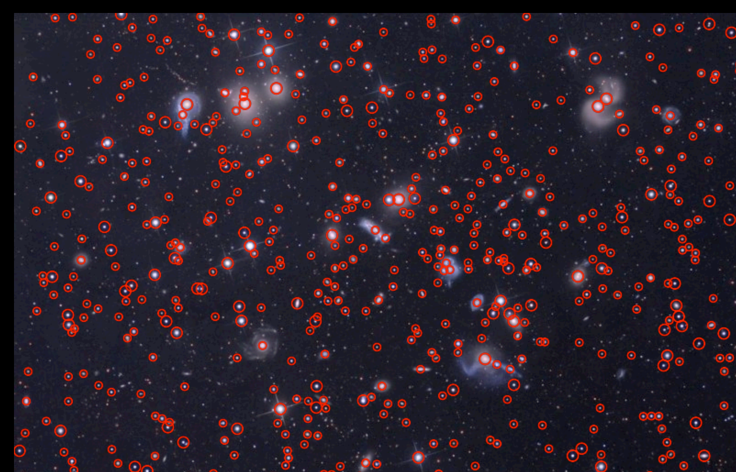


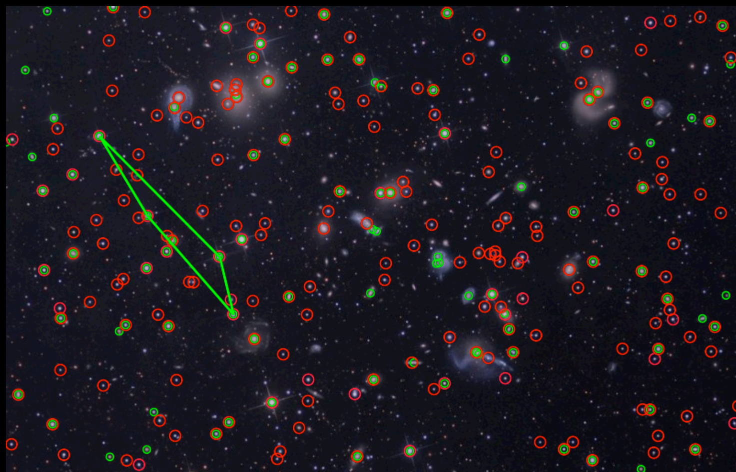
Astronomy Picture of the Day

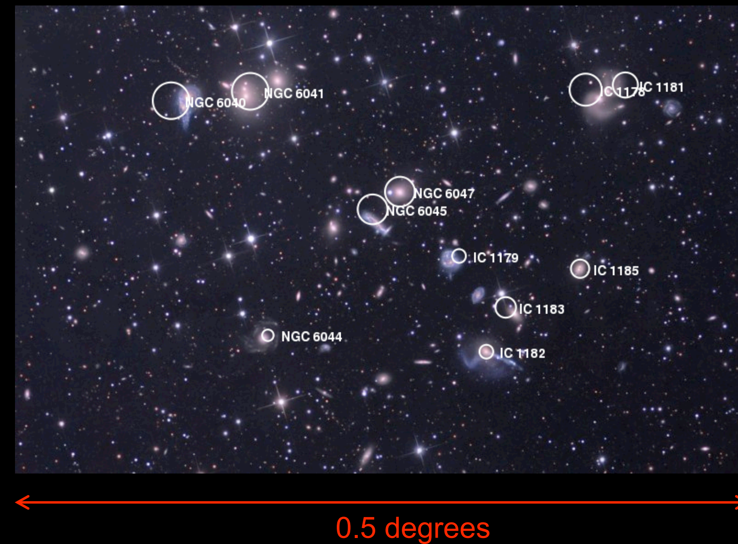


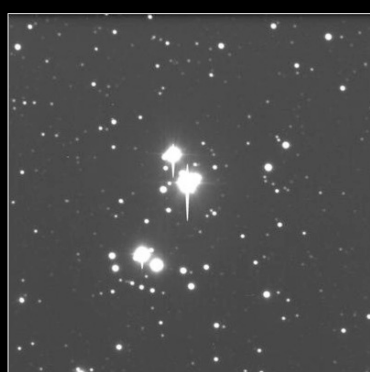


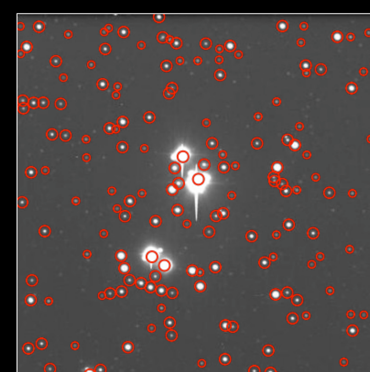

Errai
Alfirk
Alderamin
Cepheus
IC 1396
Deneb
NGC 7822
Navi
Caph
Cassiopeia
Ruchbah
Schedar
Lacerta
57 degrees

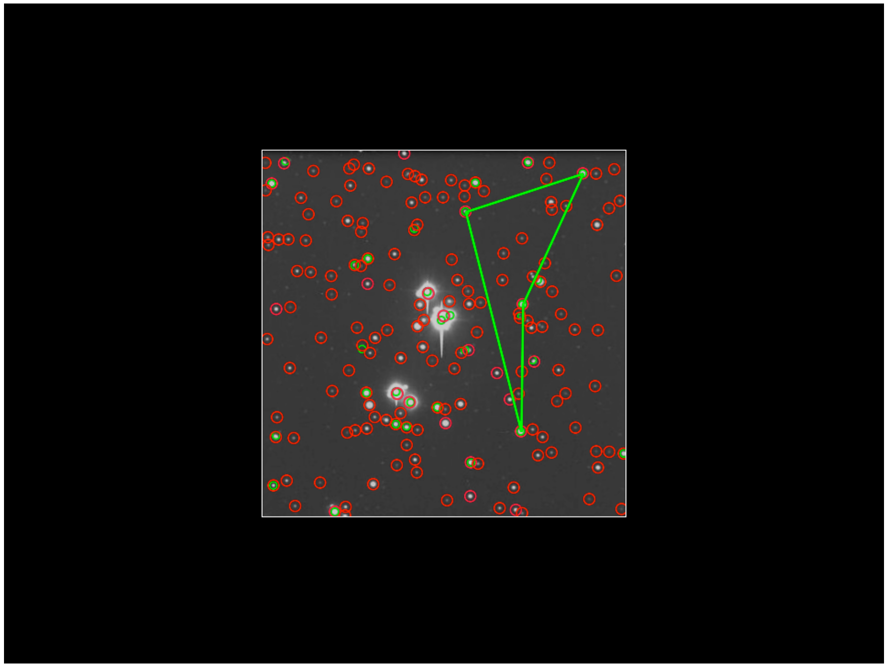


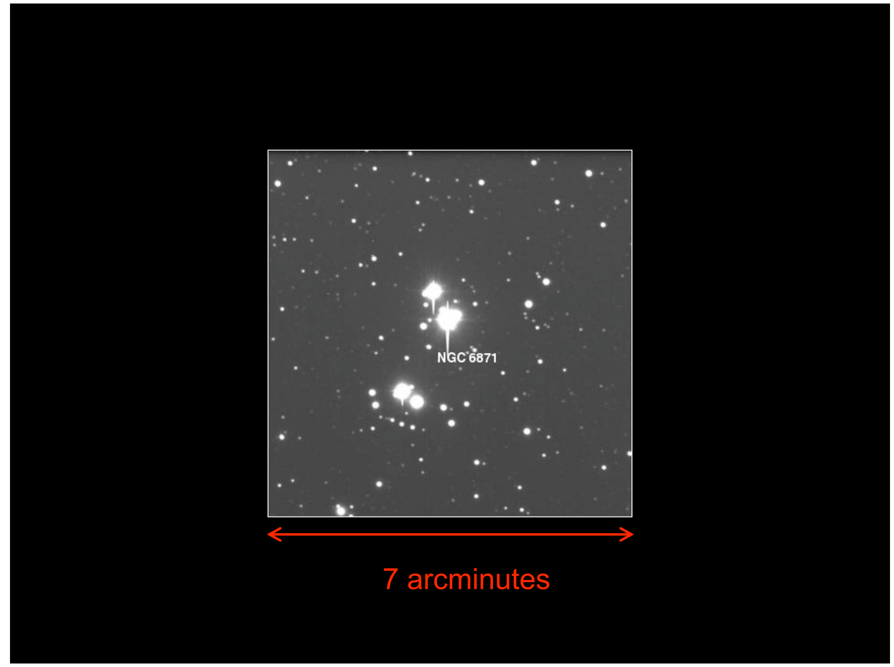


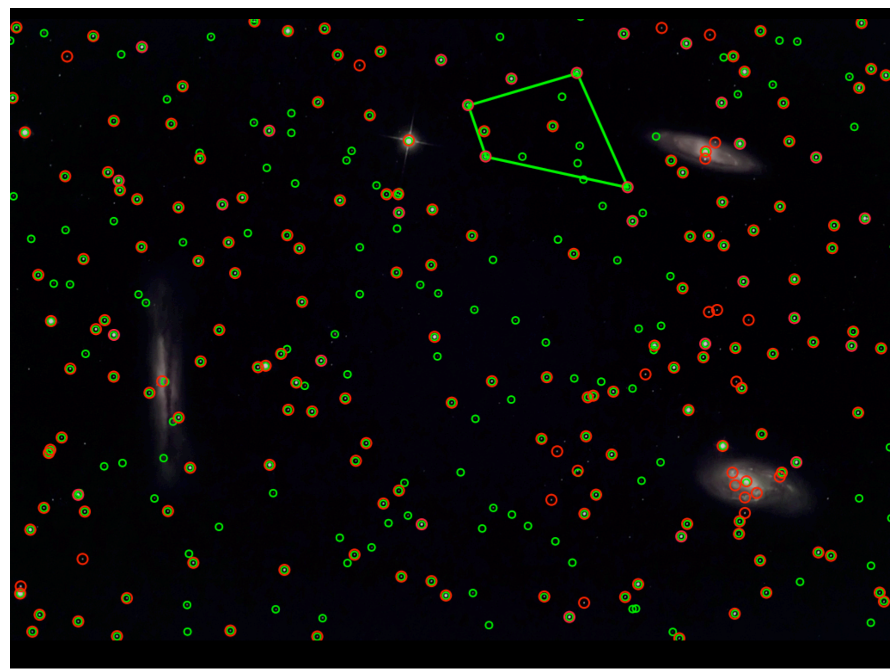

NGC 6040
NGC 6041
IC 1178
IC 1181
NGC 6047
NGC 6045
IC 1179
IC 1185
IC 1183
NGC 6044
IC 1182
0.5 degrees

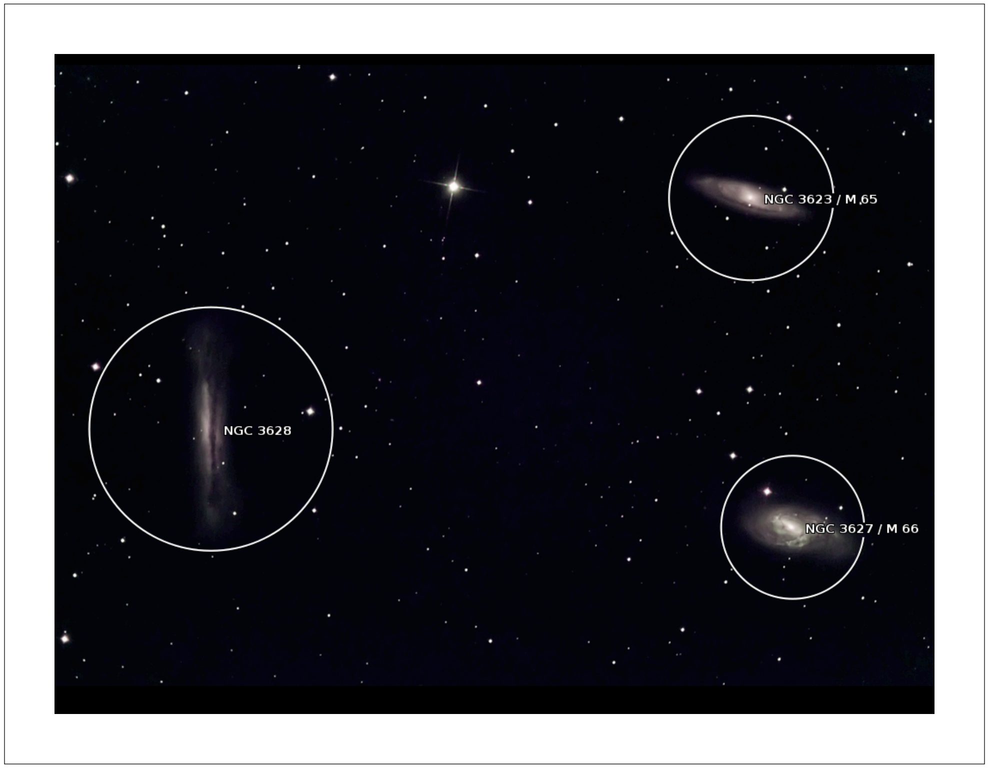


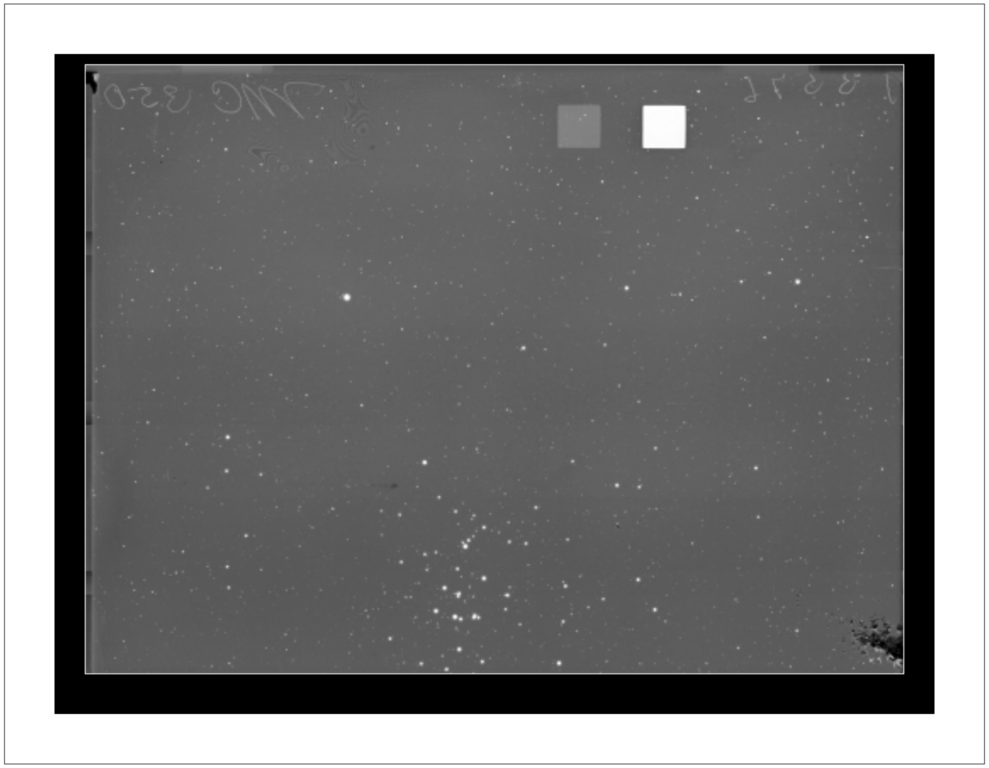


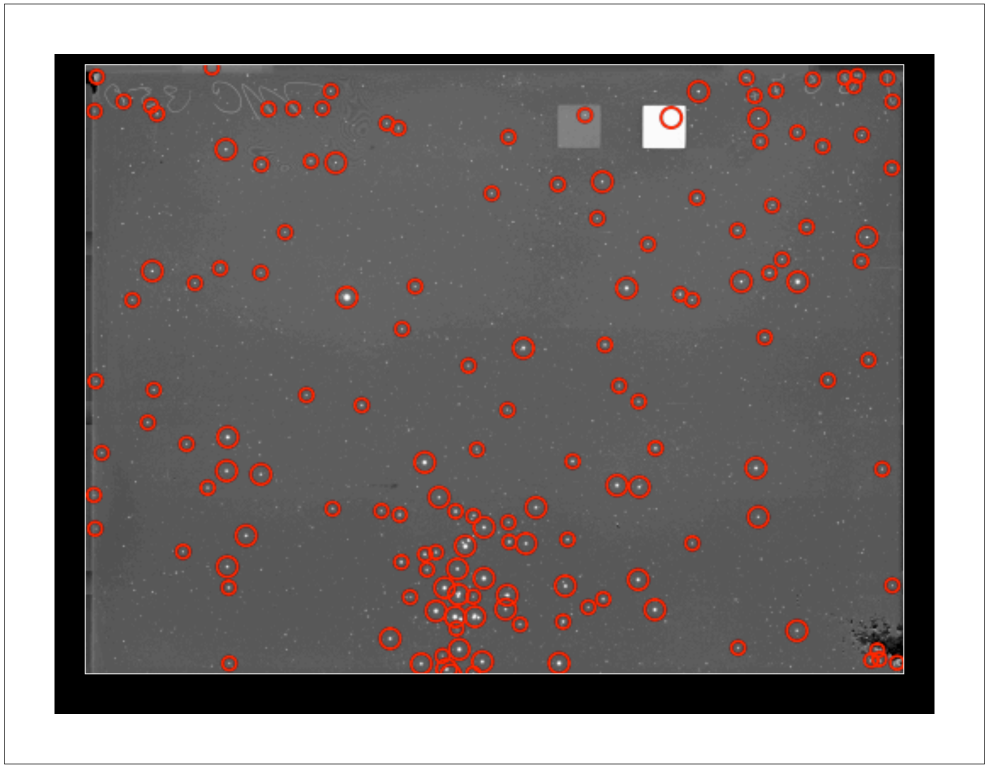


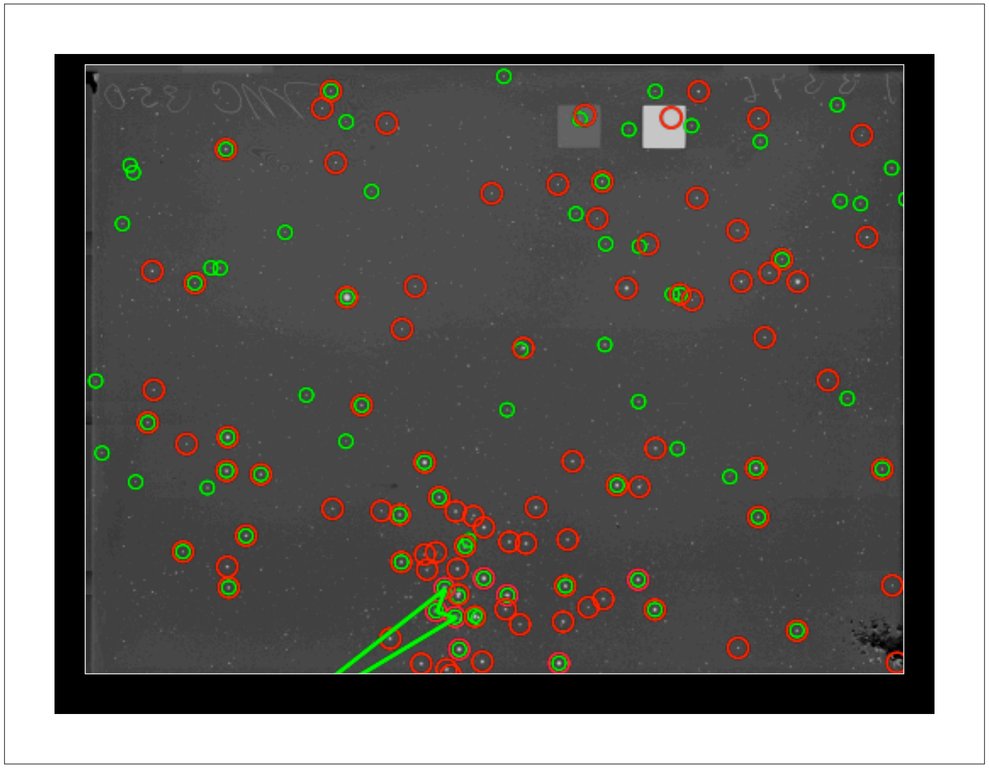

NGC 6871
7 arcminutes

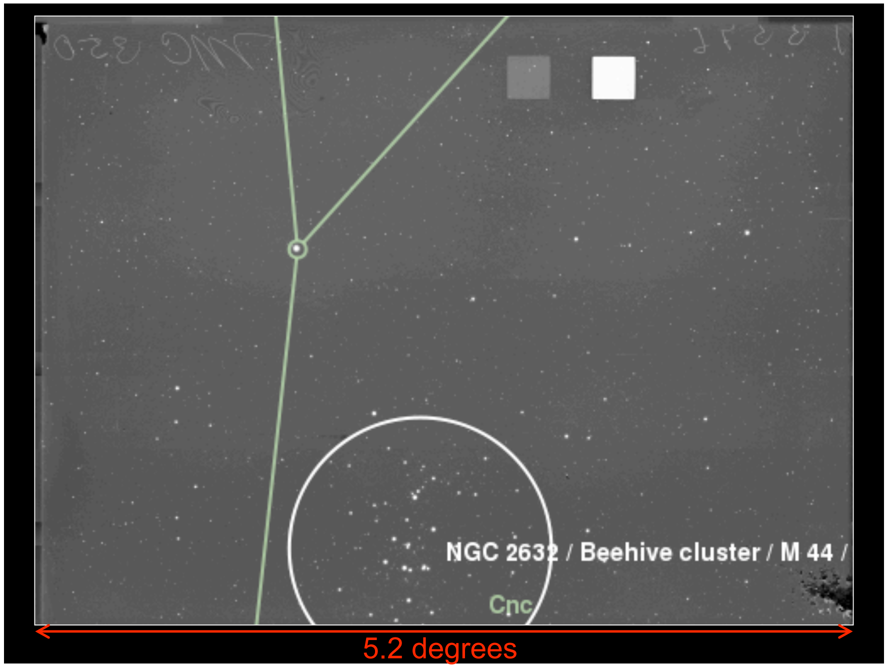


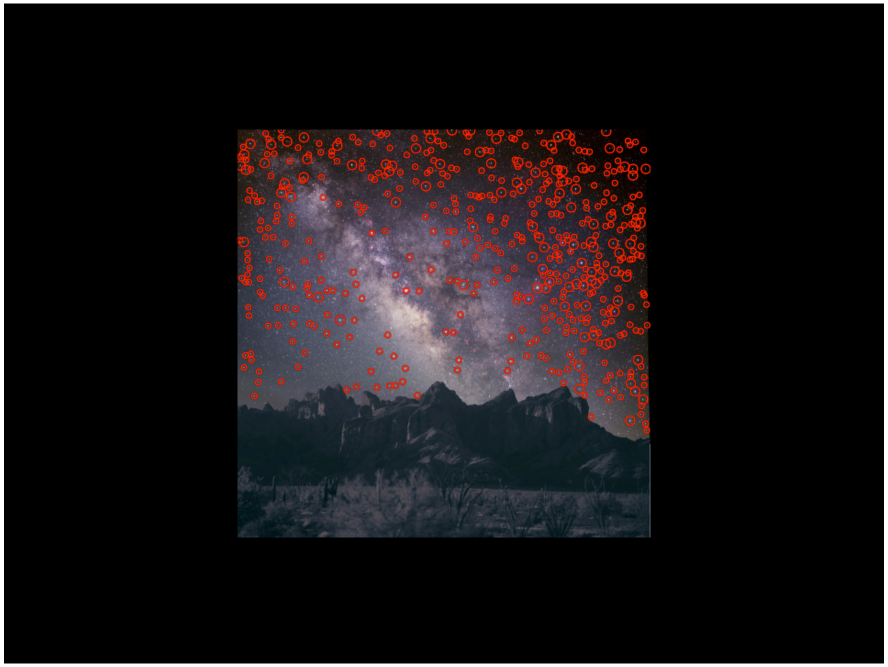
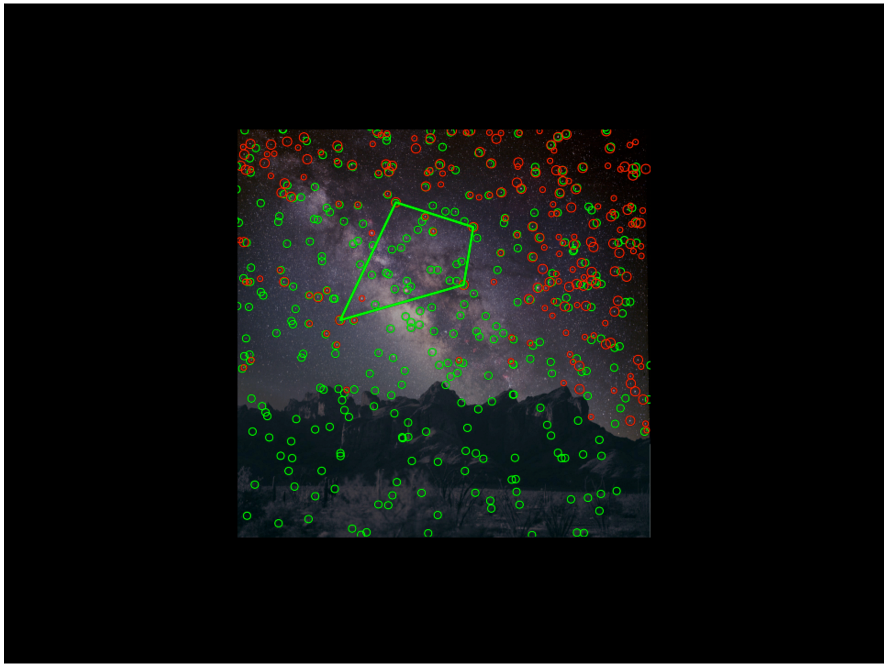

NGC 3623 / M 65
NGC 3628
NGC 3627 / M 66

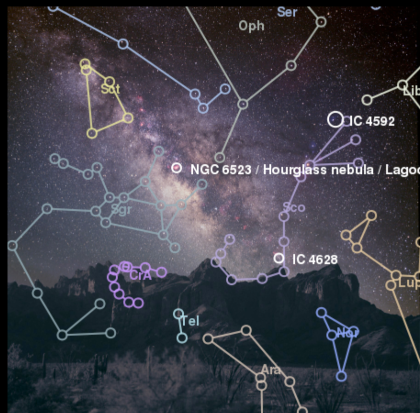




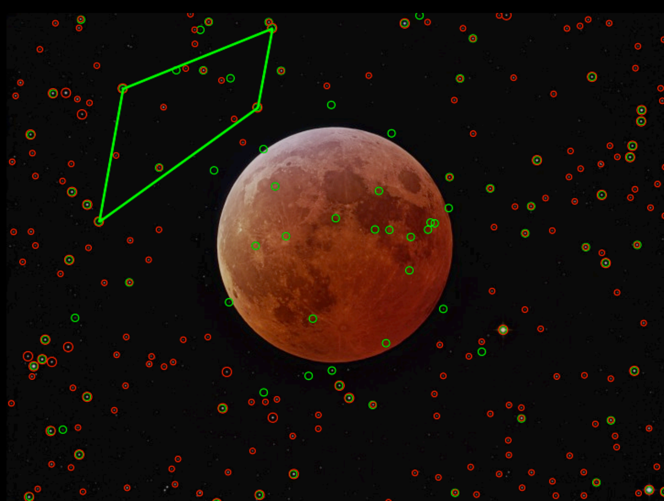



NGC 2632 / Beehive cluster / M 44 /
Cnc
5.2 degrees

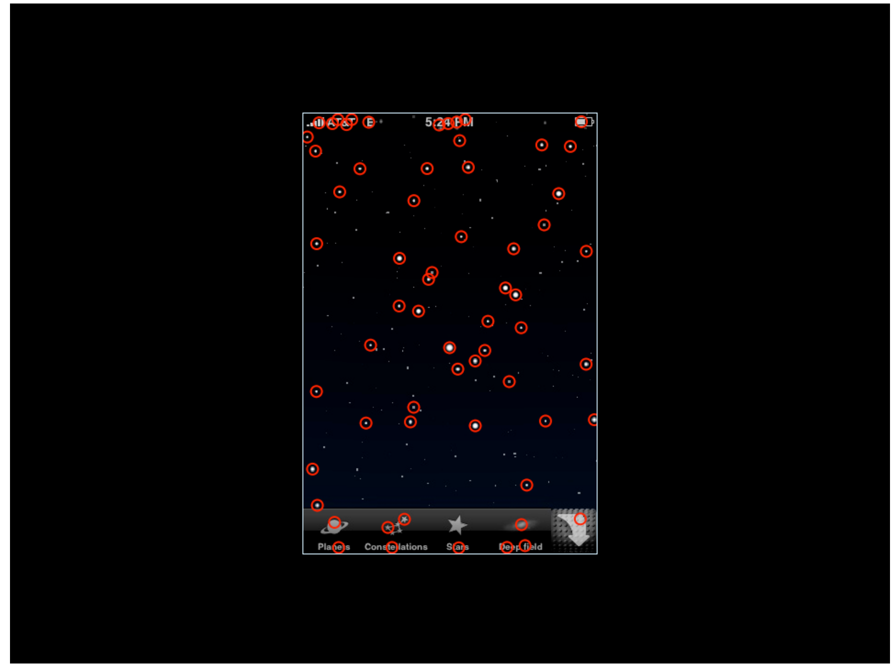



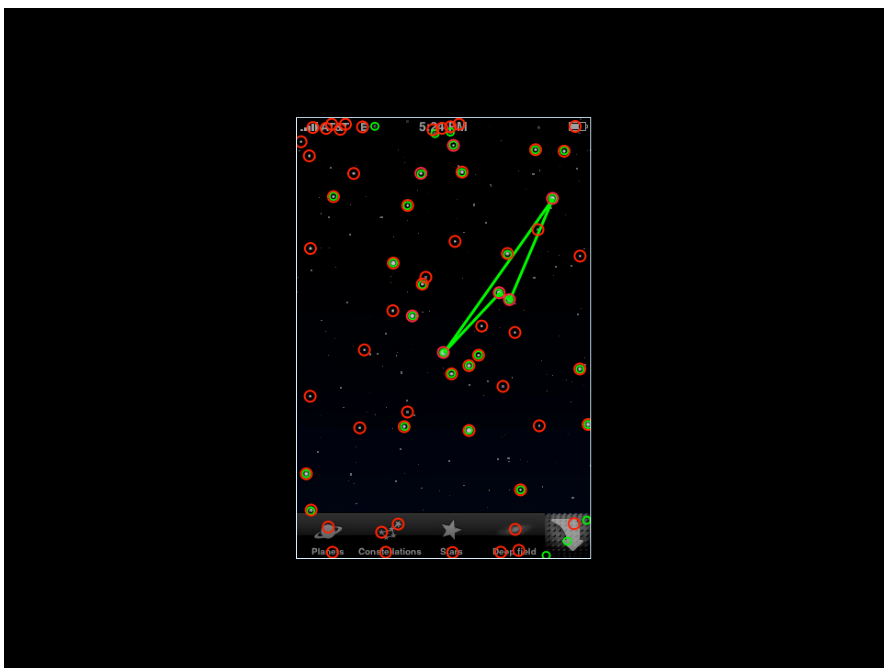


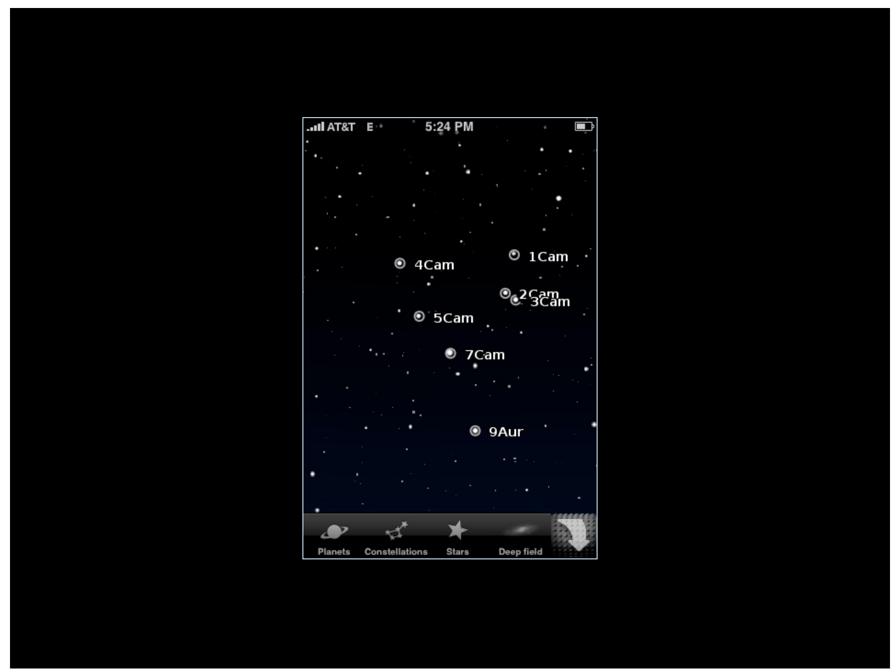
68 degrees

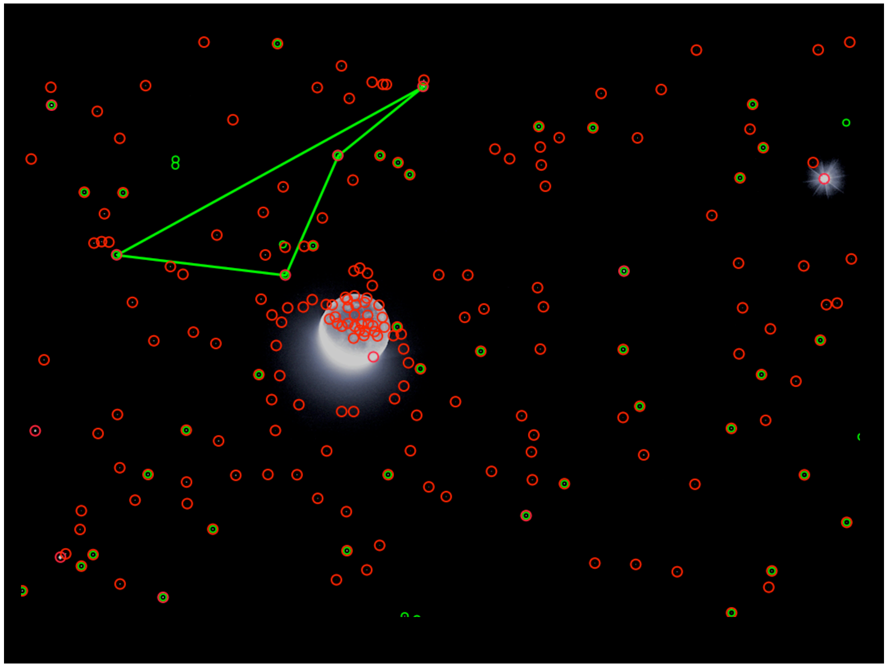

1.4 degrees

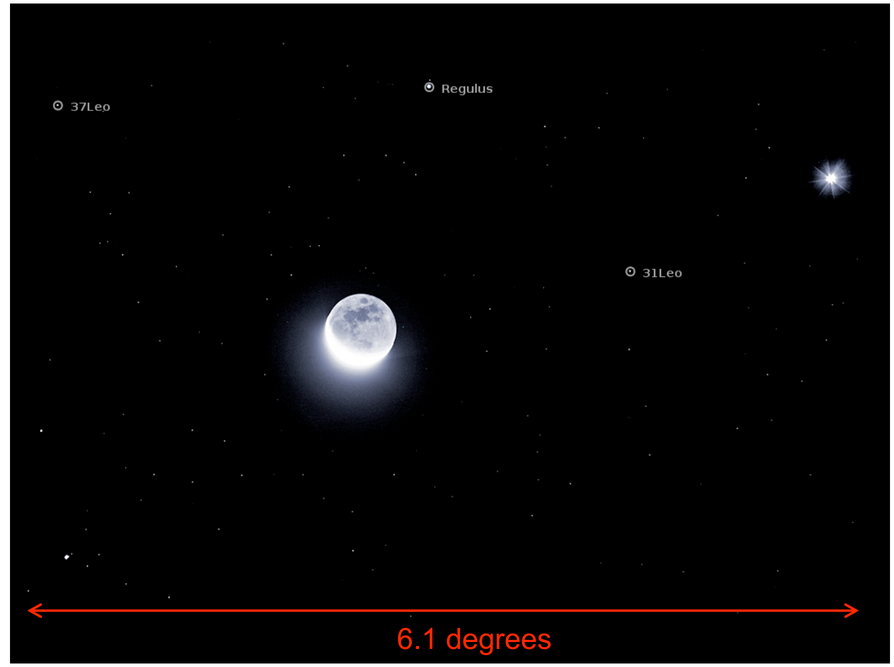

AT&T E
5:24 PM
Planets
Constellations
Stars
Deep field





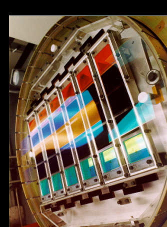

AT&T E
5:24 PM
4Cam
1Cam
2Cam
3Cam
5Cam
7Cam
9Aur
Planets
Constellations
Stars
Deep field





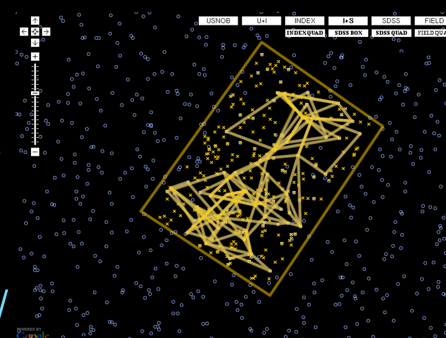
Preliminary Scaleup Results: SDSS

- The Sloan Digital Sky Survey (SDSS) is an all-sky, multi-band survey which includes targeted spectroscopy of interesting objects.
- The telescope is located at Apache Point Observatory.
- Fields are 14x9arcmin corresponding to 2048x1361 pixels.



Preliminary Scaleup Results: SDSS

- 336,554 fields science grade+
- 0 false positives
- 99.84% solved 530 unsolved
- 99.27% solve w/ 60 brightest objs

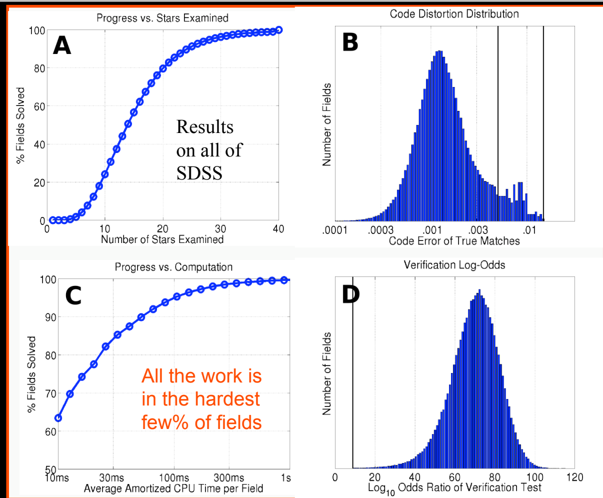


Magnitudes used only to decide search order.

Assume known pixel scale (for speedup of solving only.)

http://www.cs.toronto.edu/~roweis roweis@cs.toronto.edu

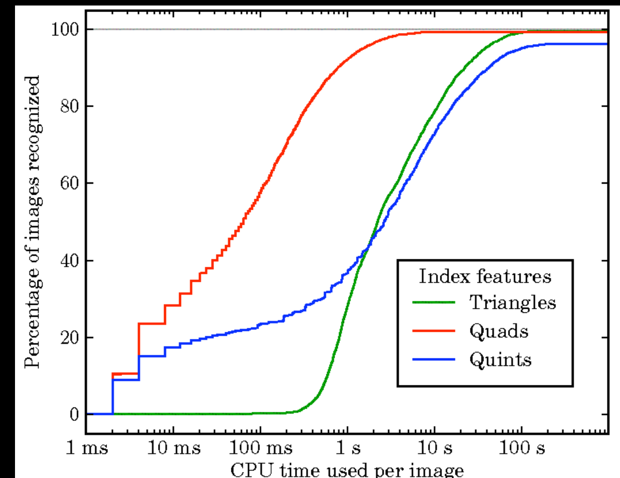
Speed/Memory/Disk



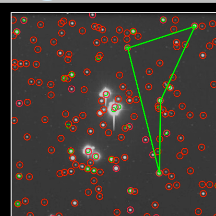
Indexing takes ~12 hours, uses ~ 2 GB of memory and ~100 GB of disk.

Solving a test image almost always takes <<1sec (not including object detection).

Why Quads?



Once we locate the image…

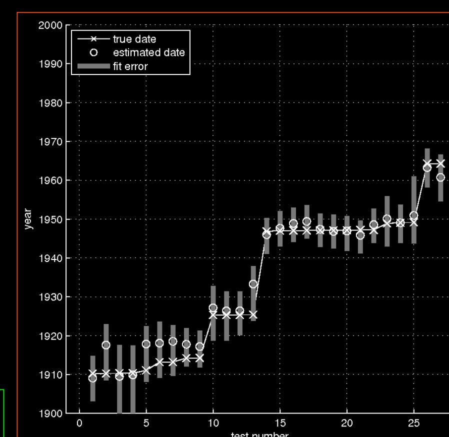


- We can say a lot more about it:
  - Telescope properties: bandpass, plate scale
  - Seeing info: approx. date, distortion model
- If we do this on entire collections, we can improve/extend standard catalogues.
- A lot of modern experiments result in disks full of images/signals; modern engineering should be great an analyzing those disks.

Preliminary Results: Blind Dating

- 27 science quality historical images
- Dates range over 50+ years.
- Mean error in estimated dates is 1.56 years. 100% inliers.

Starting from pixels only, no prior meta-data.

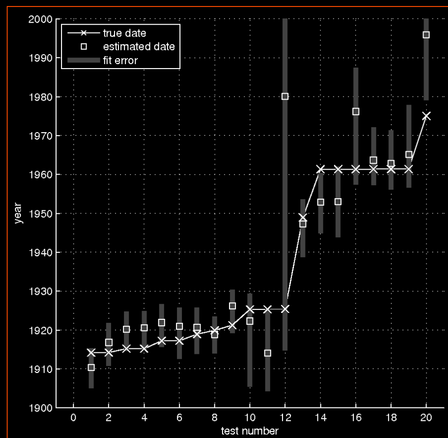


[Barron et.al 2008]

Preliminary Results: Blind Dating

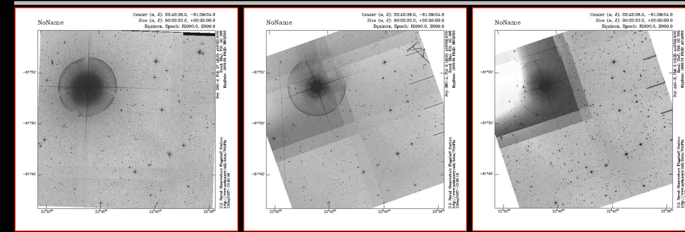
- 20 low quality historical images
- Dates range over 65+ years.
- Mean error in estimated dates is 4.41 years. 80% inliers.

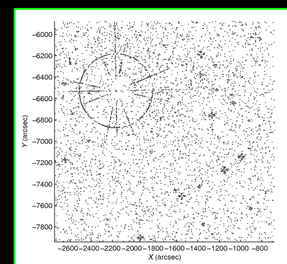
Starting from pixels only, no prior meta-data.

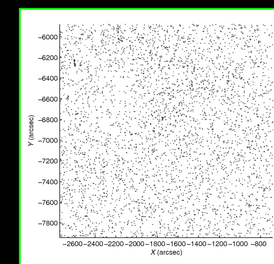


[Barron et.al 2008]

“Cleaning” the USNO Star Catalog



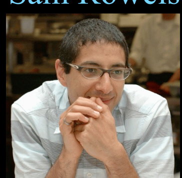




[Cleaning USNO-B, AJ 2007]

The Core Team

Sam Roweis



Astrometry.net

David Hogg

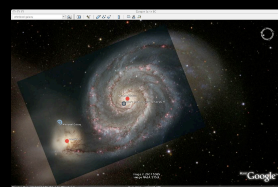
Keir Mierle Dustin Lang Jon Barron Michael Blanton

astrometry.net is open source!

- We have released all our code. Download it from astrometry.net if you want to try the system out yourself.
- We are putting the engine on the web. email alpha@astrometry.net if you want to be an alpha tester for the web service.
- Our internal trac pages are public. Check out trac.astrometry.net if you want to see all the gory details.

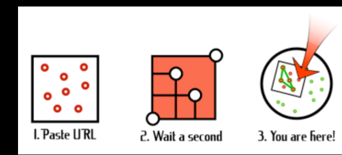
What have we done already

- Built a working prototype.
- Resolved ~ 400,000 Sloan Digital Sky Survey images blind.
- Solved historical plates (~1810).
- Tested a live service (professional & amateur astronomers as users.)
- Created a “picture of the day” layer for upcoming Google Sky.
- All with 2 students and <$80K.
- Run out of money.



What we need to do next

- Algorithms are extremely effective. Prototype is highly successful.
  Now we need to scale up.
- Next steps:
  - Get funding & resources.
  - Work with researchers, amateur & professional astronomers and others to understand/develop needs.
  - Go live and change science!



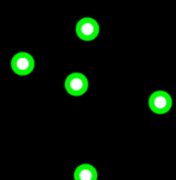
Astrometry.net

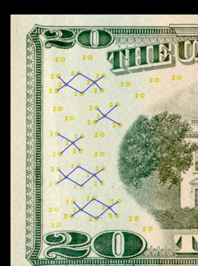
+

You?

The “Eurion” constellation

- Photocopiers and Photoshop are already playing this game!
- A pattern of five yellow dots called the Eurion constellation appears on many currencies.

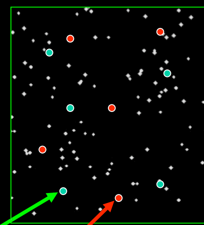




Infer the Meta-Data from the Pixels!

Bad News: Distractors & Dropouts

- Another major challenge: Query images will contain some extra stars that are not in your index catalogue, and some catalogue stars will be missing from the image.



- These "distractors" & "dropouts" mean that naïve matching techniques will not work.

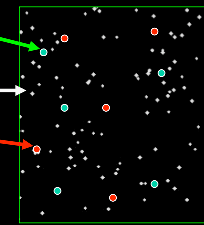
Generative ("Forward") Model

$$p(\text{image}|\theta) = \sum_j F[x_j] + \sum_{i \in I} z_i F[x_i]$$

$z_i = 1$ $z_i = 0$

$x_j \sim \text{Uniform} \quad \#j \sim \text{Poisson}$

$x_i \sim \mathcal{N}(T_\theta[\text{position}_i], \sigma^2)$



θ denotes the image scale, aspect, pointing & rotation

$F[\cdot]$ denotes the point spread function

$T_\theta[\cdot]$ transforms star locations from catalogue (sky) coordinates to image coordinates under θ

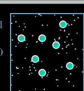
z_i are iid Bernoulli indicators for "dropout" (whose prior depends on scale)

State Estimation ("Inverse Model")

$p(\text{image}|\theta) = \sum_j F[x_j] + \sum_{i \in I} z_i F[x_i]$

$x_j \sim \text{Uniform} \quad \#j \sim \text{Poisson}$

$x_i \sim \mathcal{N}(T_\theta[\text{position}_i], \sigma^2)$



θ denotes the image scale, aspect, pointing & rotation

$F[\cdot]$ denotes the point spread function

$T_\theta[\cdot]$ transforms star locations from catalogue (sky) coordinates to image coordinates under θ

z_i are iid Bernoulli indicators for "dropout"

$$p(\theta|\text{image}) \; ?$$

Great! Now we just have to invert the model to find the parameters given an image. Hmm…

Approximating the Posterior (3)

Break the sum over matchings into three terms:

$p(\text{image}|m^*) + N_q\beta$ ← false positives with at least q stars

(z is the number of image sources not explained by noise)

$$p(m|\text{image}) \approx \frac{p(\text{image}|m)}{p(\text{image}|m^*) + N_q\beta}$$

Assume every element of the second term (false positives with at least q stars) has the same value.

Assume every element of the third term is zero!

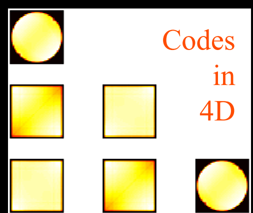
(There are billions of terms, each of order 1e-300.)

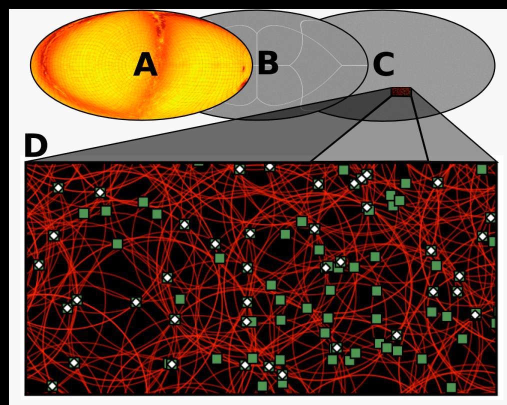
Key quantity is the log odds of false/true prob. mass, which we can estimate pretty well:

$$\log\left[\frac{N_q\beta}{p(\text{image}|m^*)}\right]$$

A Typical Final Index at One Scale

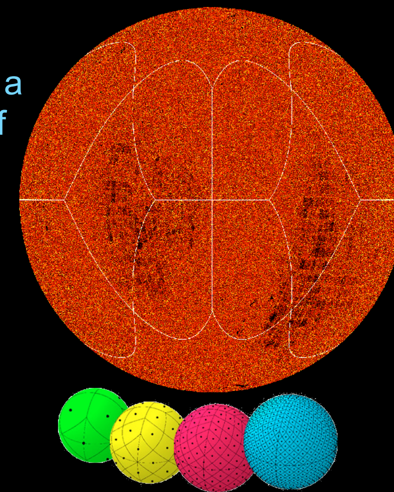
- 144M stars (6 quads/star)
- 205M quads (4-5 arcmin)
- 12 healpixes

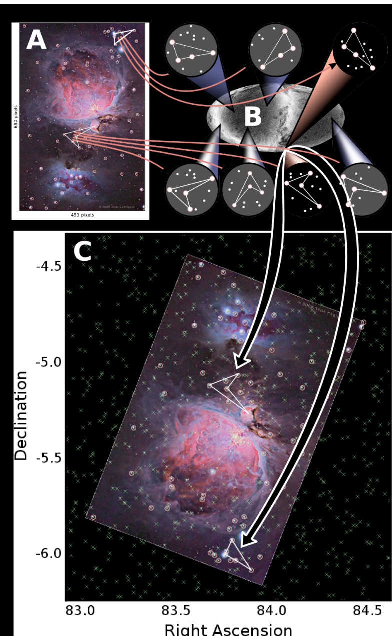




Making a uniform catalogue

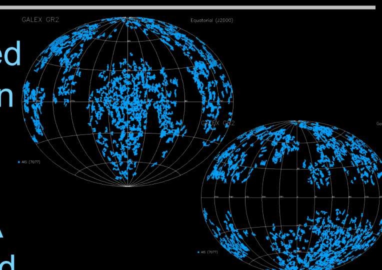
- Starting with USNO+ TYCHO we “cut” to get a spatially uniform set of the ~150M brightest stars & galaxies.
- We do this by laying down a fine “healpix” grid and taking the brightest K unique objects in each pixel.





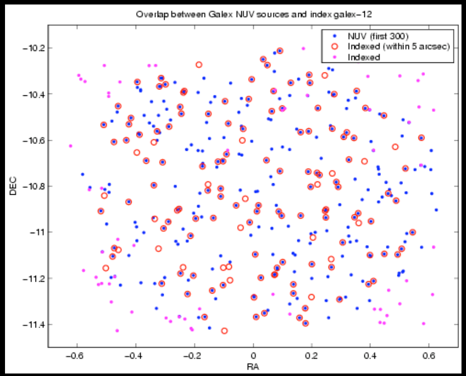
Preliminary Results: GALEX

- GALEX is a space-based telescope, seeing only in the ultraviolet.
- It was launched in April 2003 by Caltech&NASA and is just about finished collecting data now.
- It takes huge (80 arcmin) circular fields with 5arcsec resolution and spectra of all objects.



Preliminary Results: GALEX

- GALEX NUV fields can be solved easily using an index built from bright blue USNO stars.

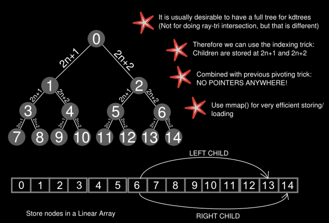


Caching Computation

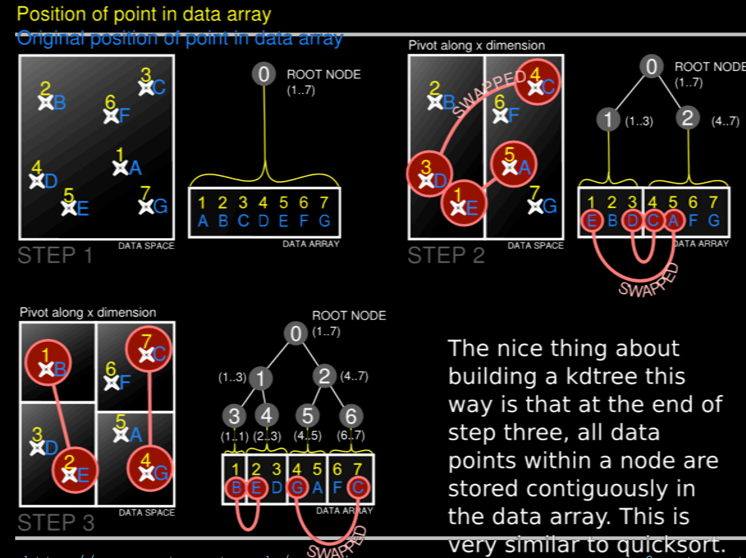
- The idea of an inverted index is that is pushes the computation from search time back to index construction time.
- We actually do perform an exhaustive search of sorts, but it happens during the building of the inverted index and not at search time, so queries can still be fast.
- There are millions of patches of the scale of a test image on the sky (plus rotation), so we need to extract about 30 bits.

Algorithms & Data Structures

- Implementations are all in-core.
- Written in C & Python.
- Parallelization is at the script level, which has many aggregation & storage advantages.
- We make extensive use of mem-mapped files, some fancy AVL lists and a cool new “pointerless” KD-tree implementation. [Mierle & Lang]



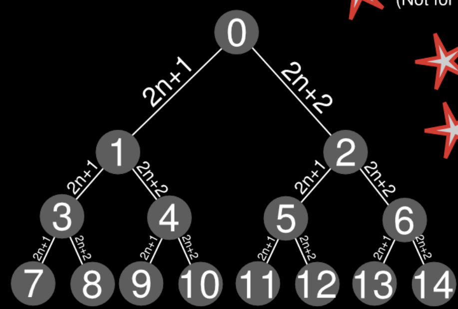
Pointer-Free KD-Trees

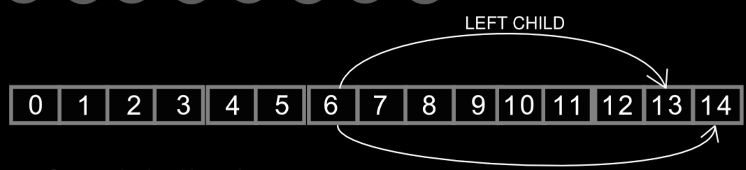


The nice thing about building a kdtree this way is that at the end of step three, all data points within a node are stored contiguously in the data array. This is very similar to quicksort.

Pointer-Free KD-Trees

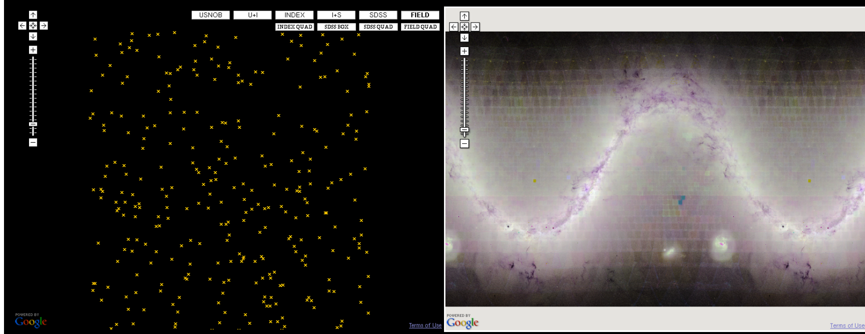
- It is usually desirable to have a full tree for kdtrees (Not for doing ray-tri intersection, but that is different)
- Therefore we can use the indexing trick: Children are stored at 2n+1 and 2n+2
- Combined with previous pivoting trick: NO POINTERS ANYWHERE!
- Use mmap() for very efficient storing/loading





Store nodes in a Linear Array

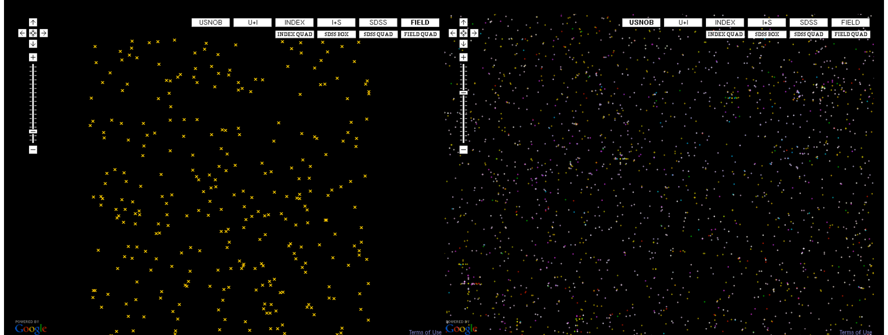
A Real Example from SDSS



Query image (after object detection).

An all-sky catalogue.

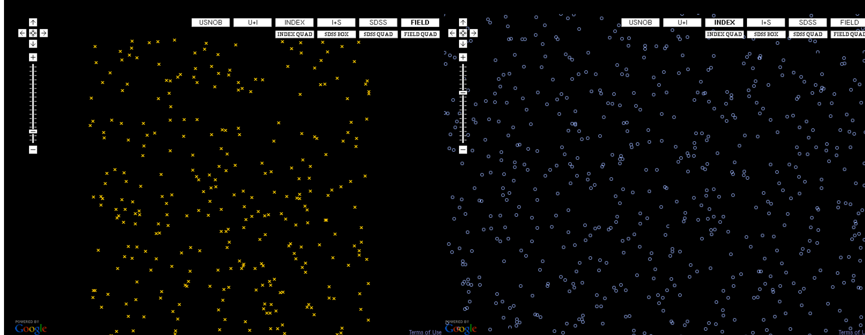
A Real Example from SDSS



Query image (after object detection).

Zoomed in by a factor of ~ 1 million.

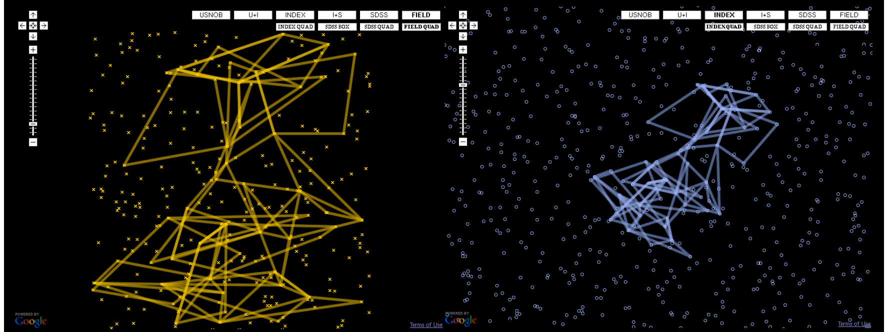
A Real Example from SDSS



Query image (after object detection).

The objects in our index.

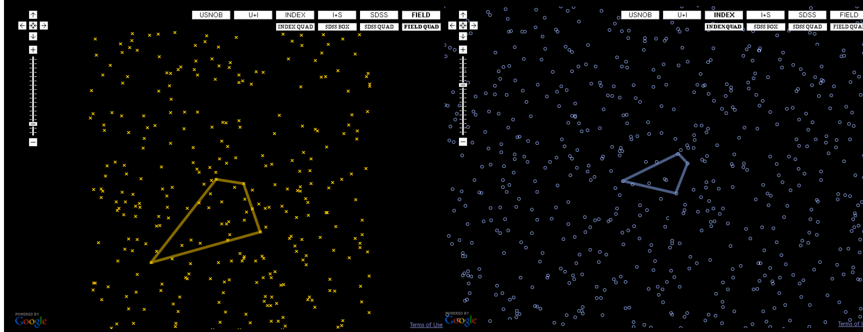
A Real Example from SDSS



All the quads in our index which are present in the query image.

http://www.cs.toronto.edu/~roweis roweis@cs.toronto.edu

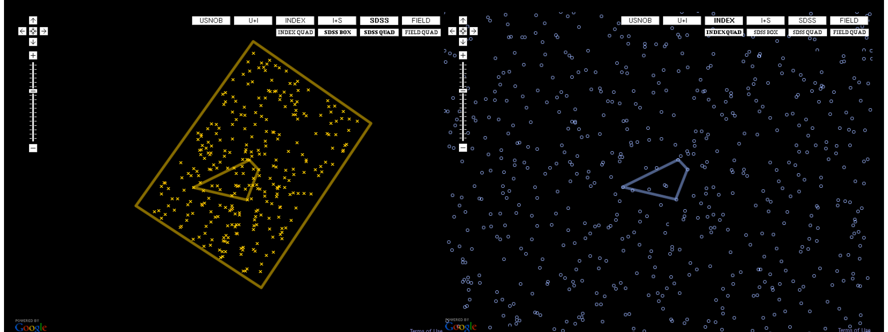
A Real Example from SDSS



A single quad which we happened to try.

http://www.cs.toronto.edu/~roweis roweis@cs.toronto.edu

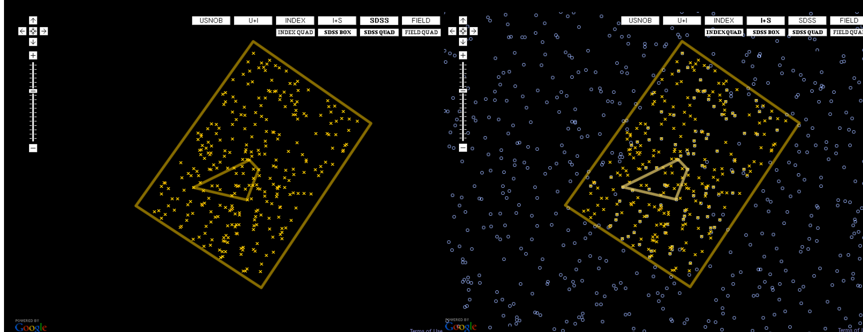
A Real Example from SDSS



The query image scaled, translated & rotated as specified by the quad.

http://www.cs.toronto.edu/~roweis roweis@cs.toronto.edu

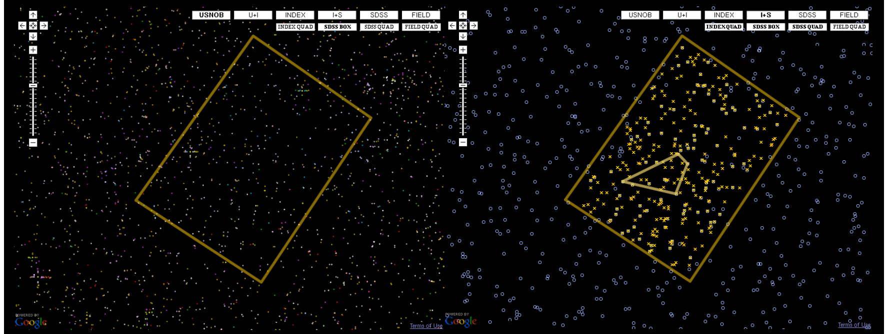
A Real Example from SDSS



The proposed match, on which we run verification.

http://www.cs.toronto.edu/~roweis roweis@cs.toronto.edu

A Real Example from SDSS



The verified answer, overlaid on the original catalogue.

The proposed match, on which we run verification.