

Lecture 15

Robust Estimation : RANSAC

RECALL: Parameter Estimation:

Let's say we have found point matches between two images, and we think they are related by some parametric transformation (e.g. translation; scaled Euclidean; affine). How do we estimate the parameters of that transformation?

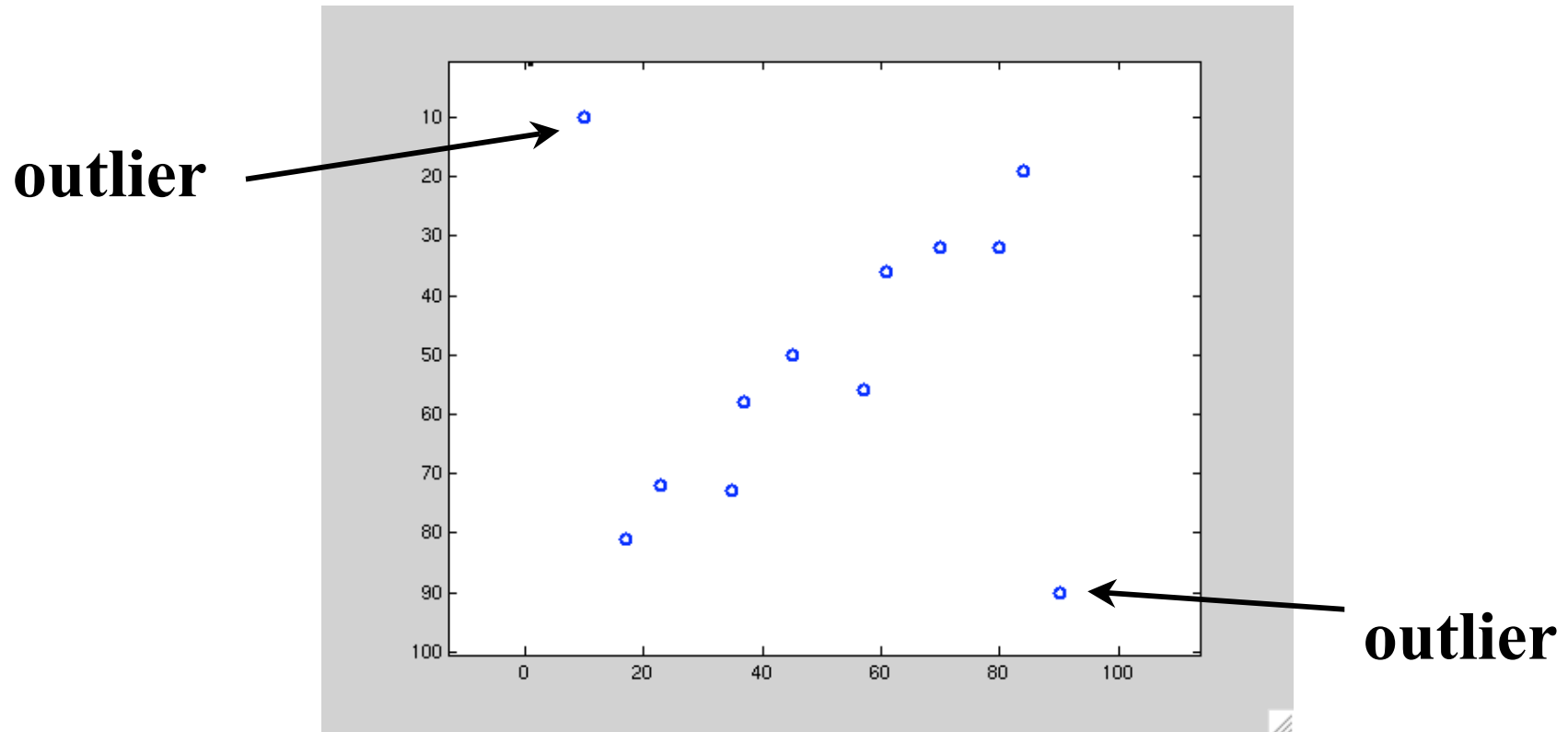
General Strategy

- Least-Squares estimation from point correspondences

But there are problems with that approach....

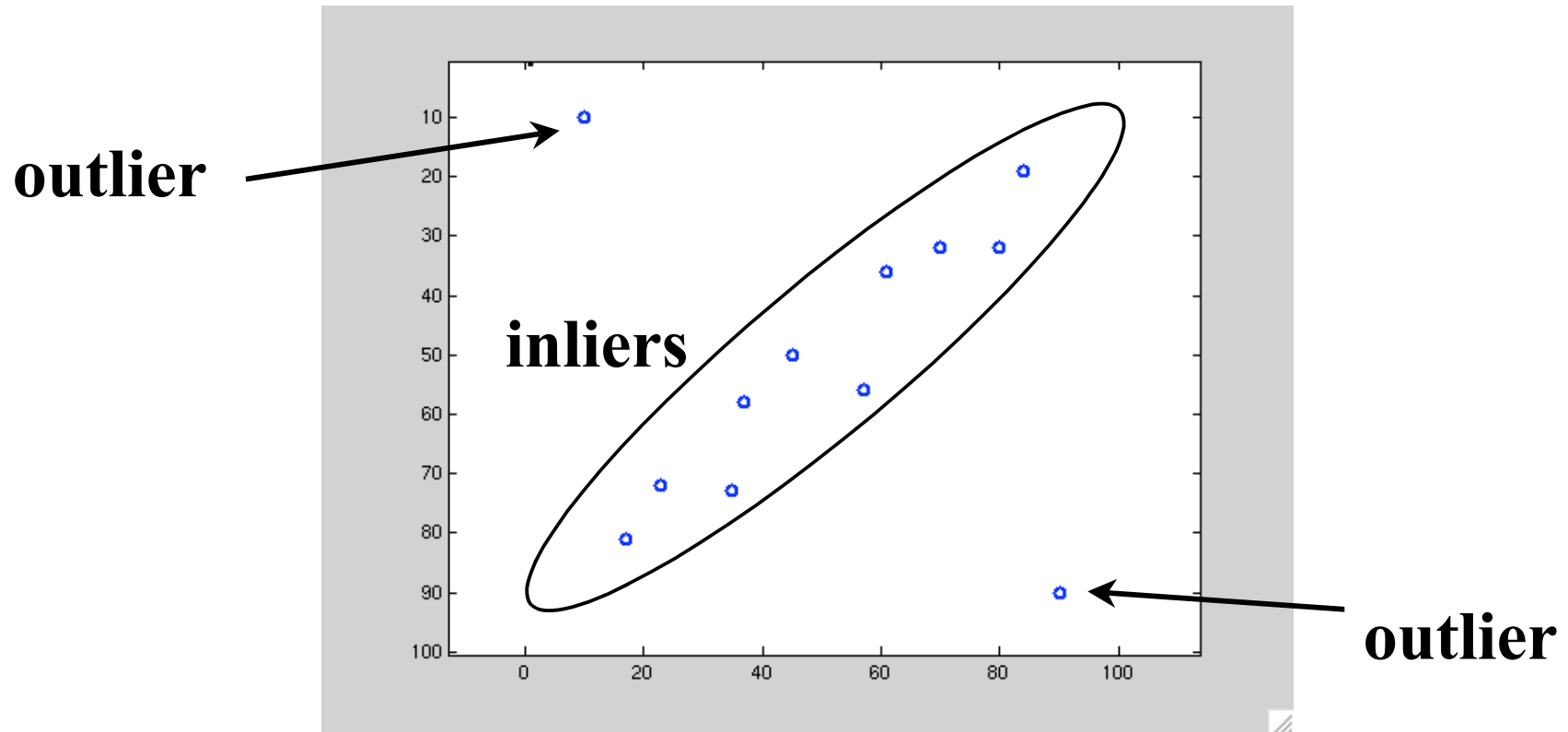
Problem : Outliers

Loosely speaking, outliers are points that don't "fit" the model.



Bad Data => Outliers

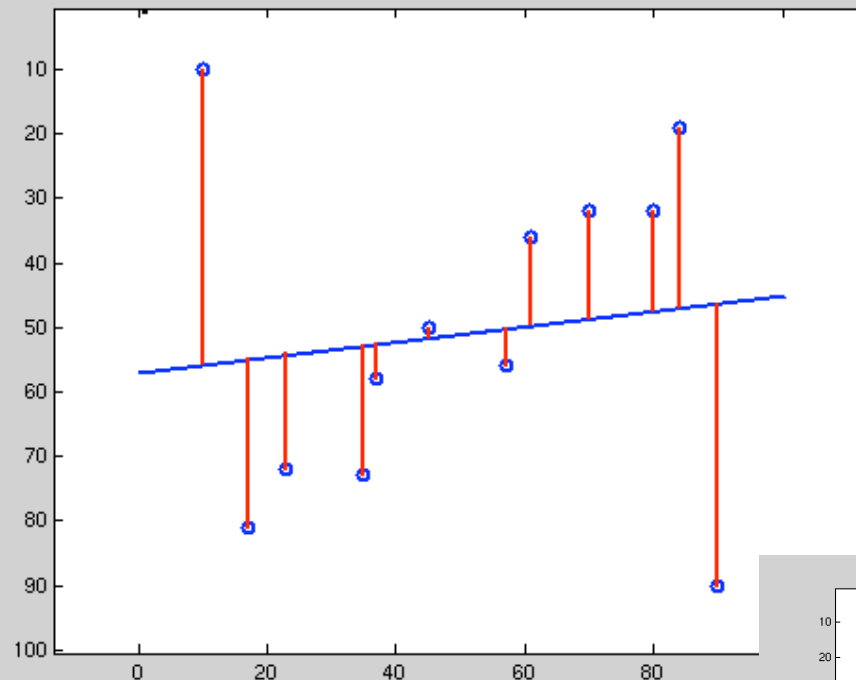
Loosely speaking, outliers are points that don't "fit" the model.
Points that do fit are called "inliers"



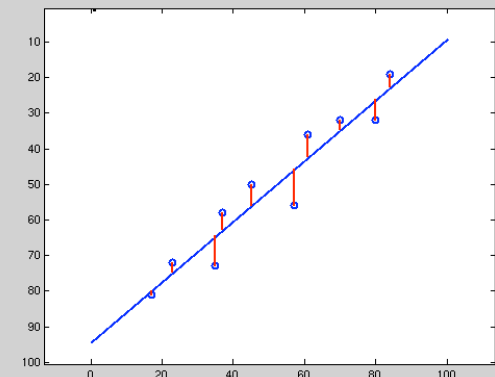
Problem with Outliers

Least squares estimation is sensitive to outliers,
so that a few outliers can greatly skew the result.

Least squares
regression with
outliers

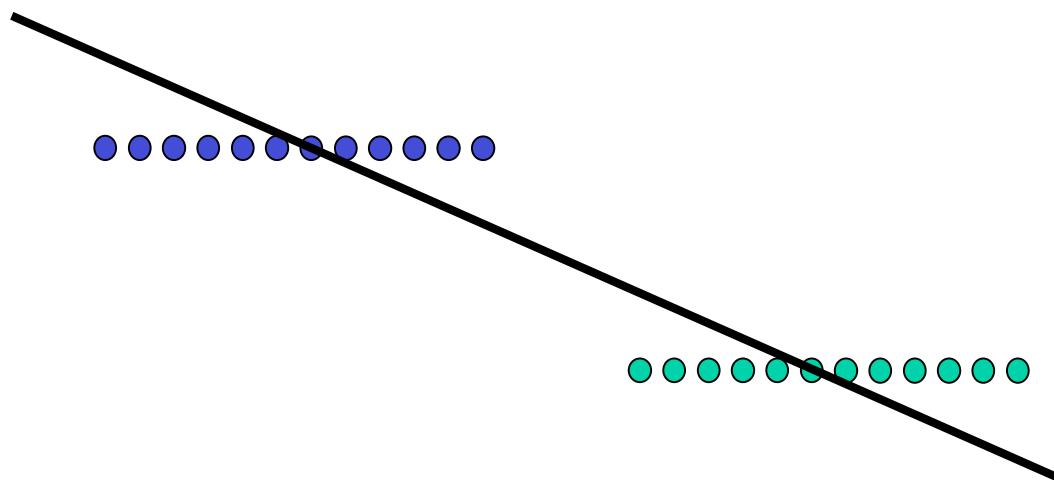


compare



**Solution: Estimation methods
that are robust to outliers.**

Outliers aren't the Only Problem



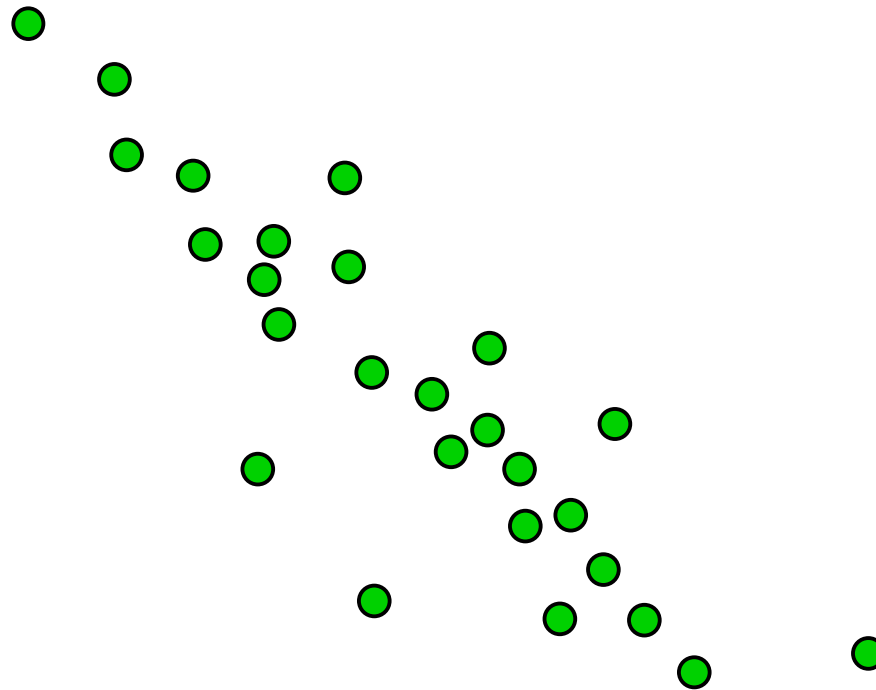
Multiple structures can also skew the results. (the fit procedure implicitly assumes there is only one instance of the model in the data).

Robust Estimation

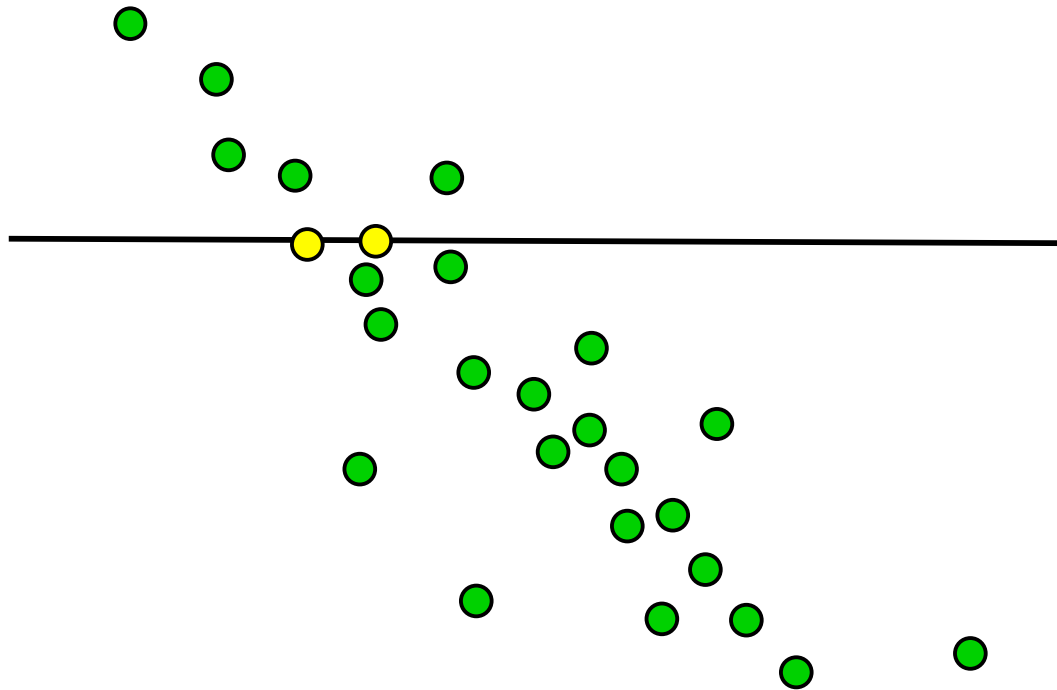
- View estimation as a two-stage process:
 - Classify data points as outliers or inliers
 - Fit model to inliers while ignoring outliers
- Example technique: RANSAC
(**RAN**dom **SA**mples **C**onsensus)

M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. of the ACM* **24**: 381--395.

Ransac Procedure

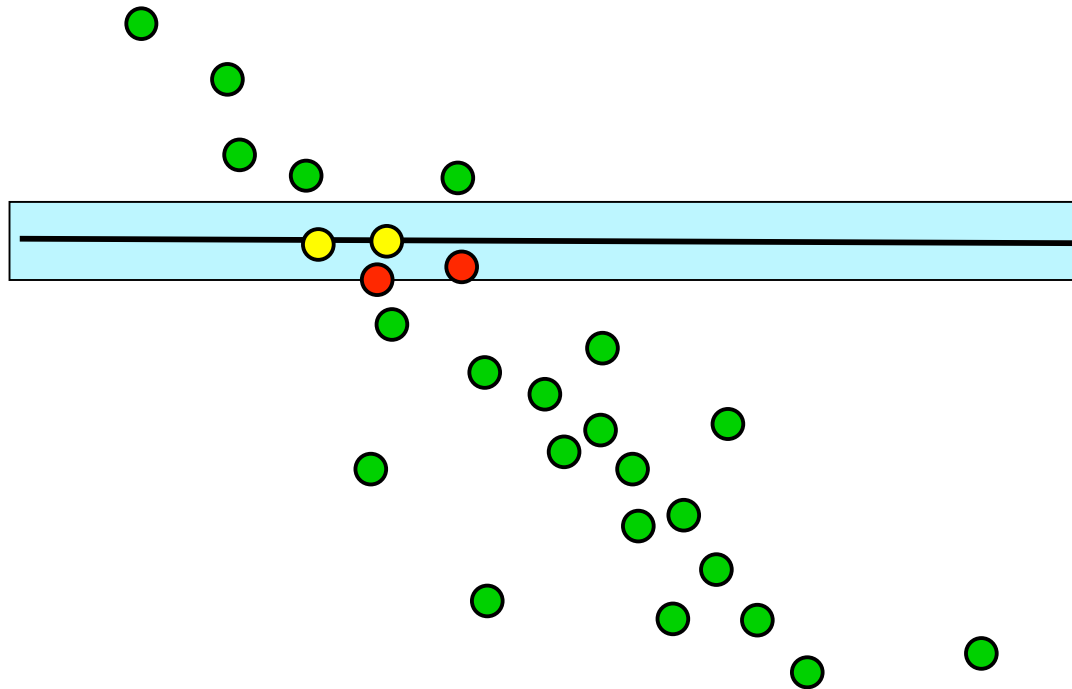


Ransac Procedure

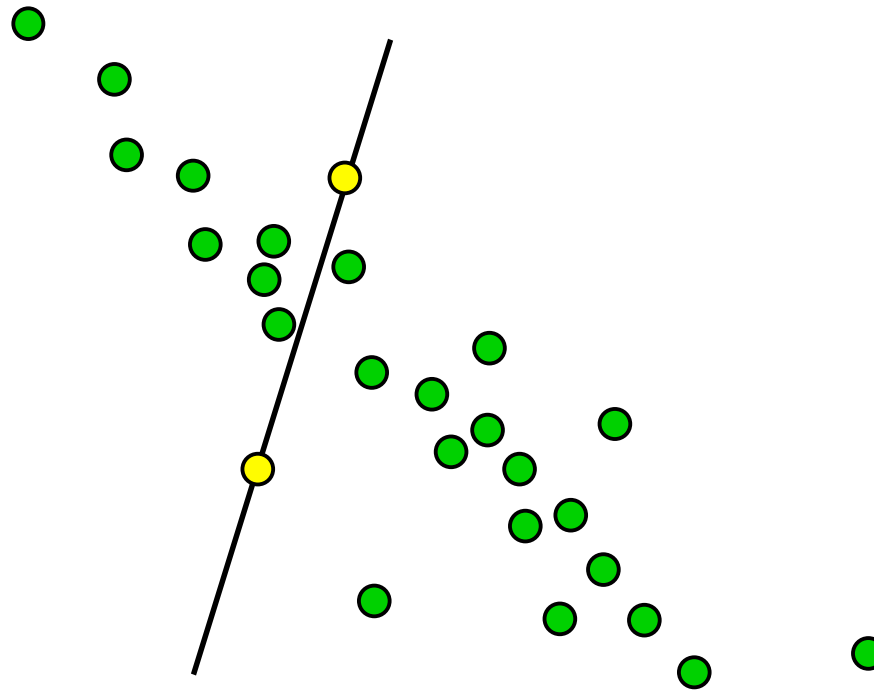


Ransac Procedure

Count = 4

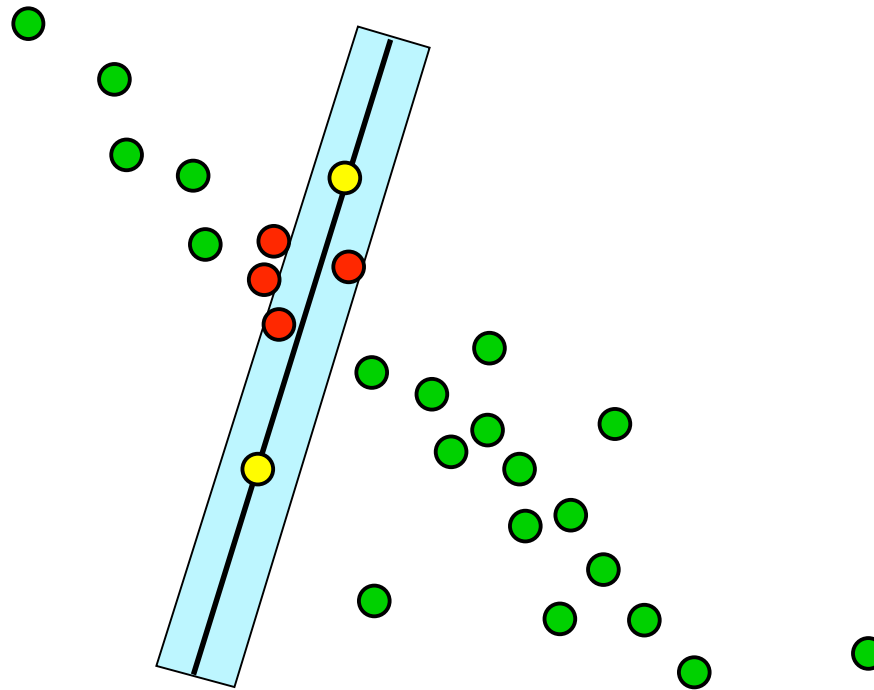


Ransac Procedure

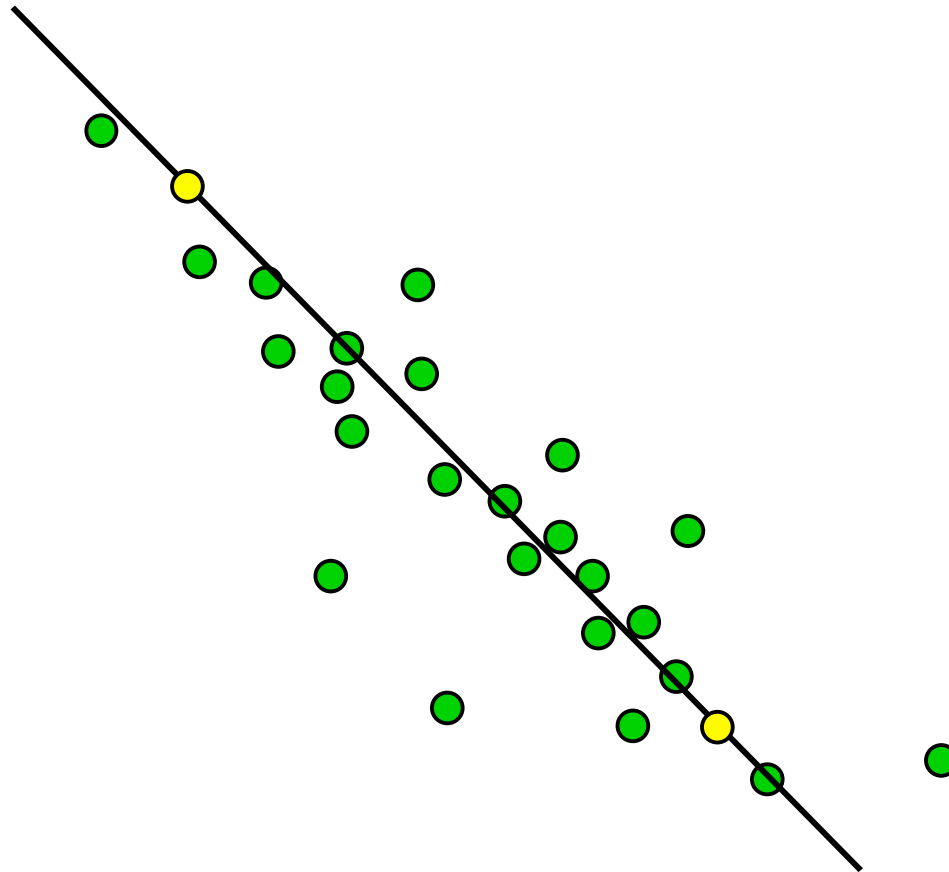


Ransac Procedure

Count = 6

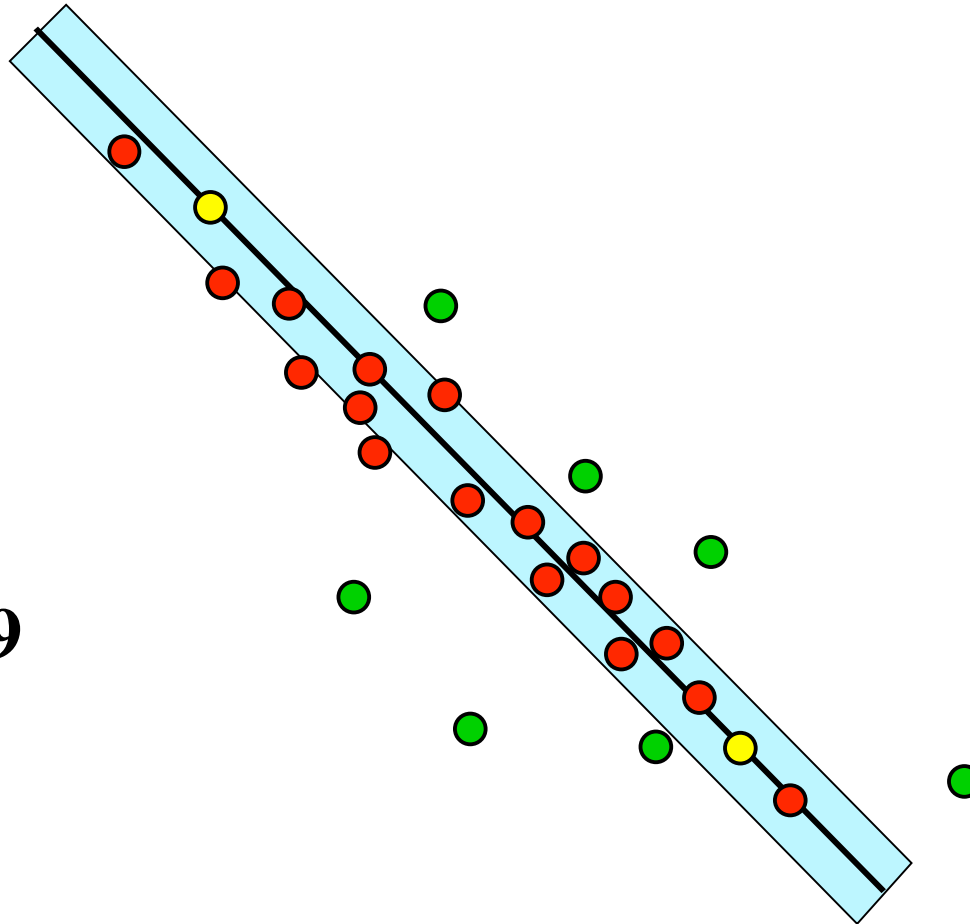


Ransac Procedure

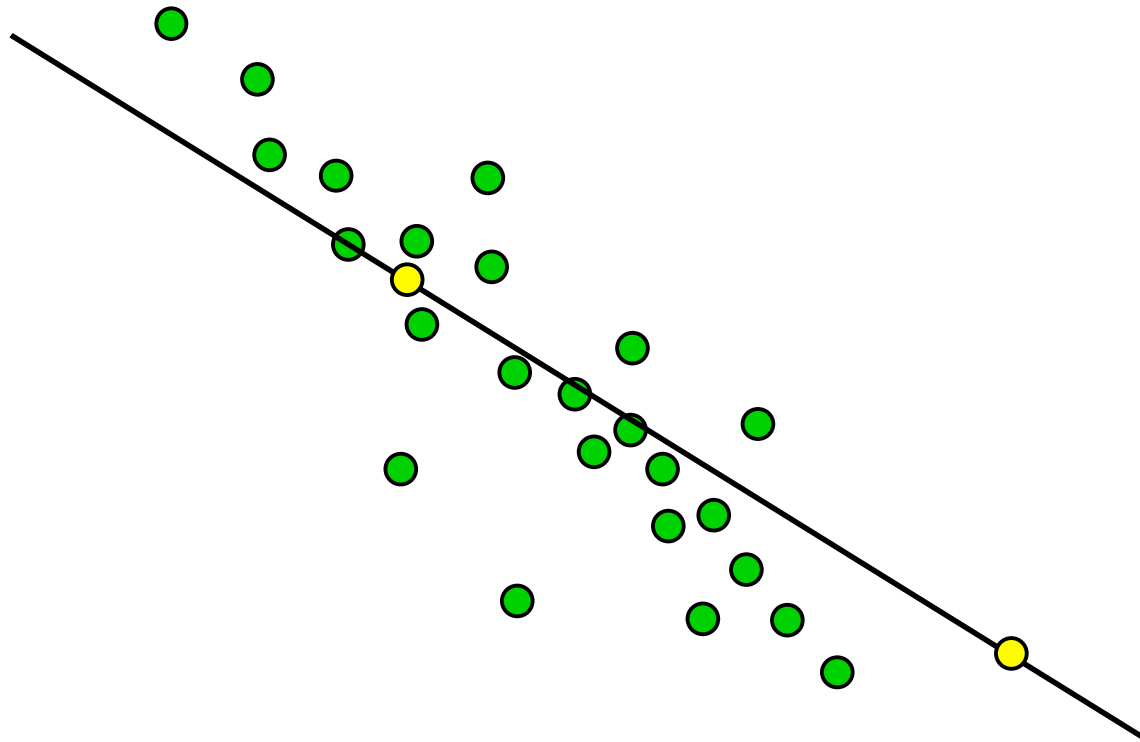


Ransac Procedure

Count = 19

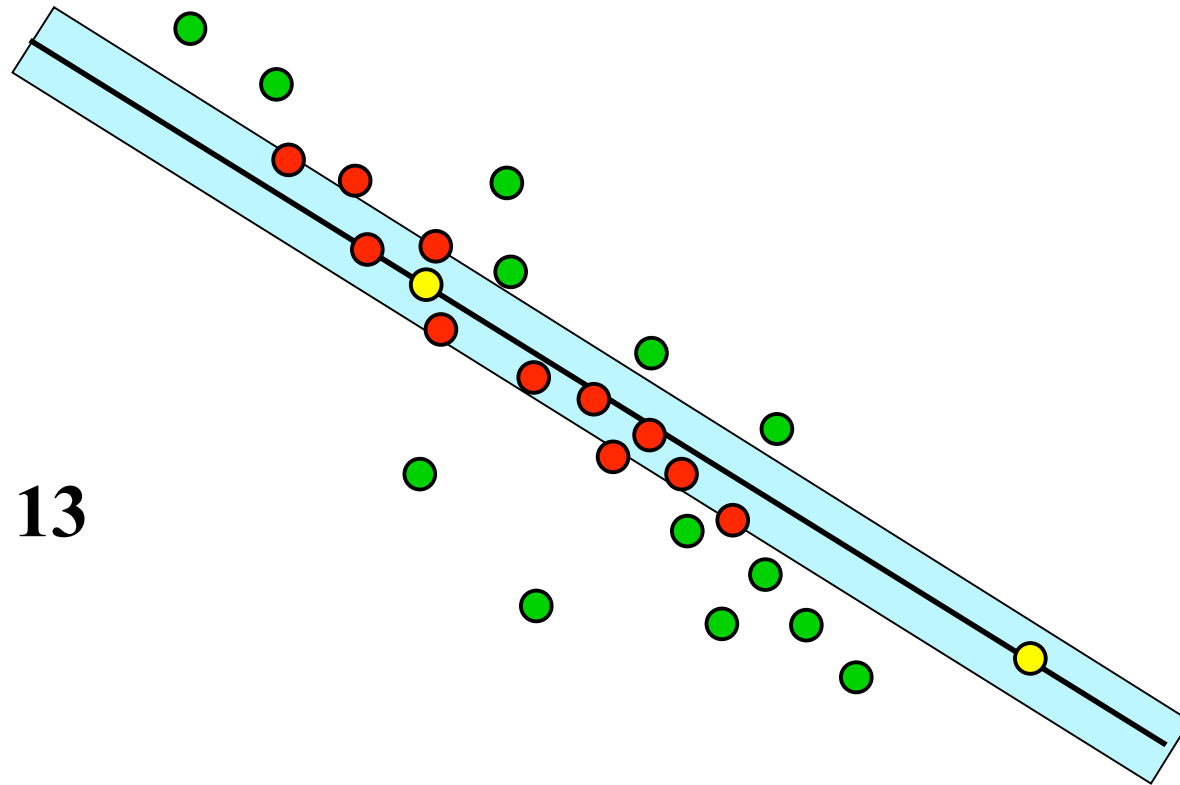


Ransac Procedure



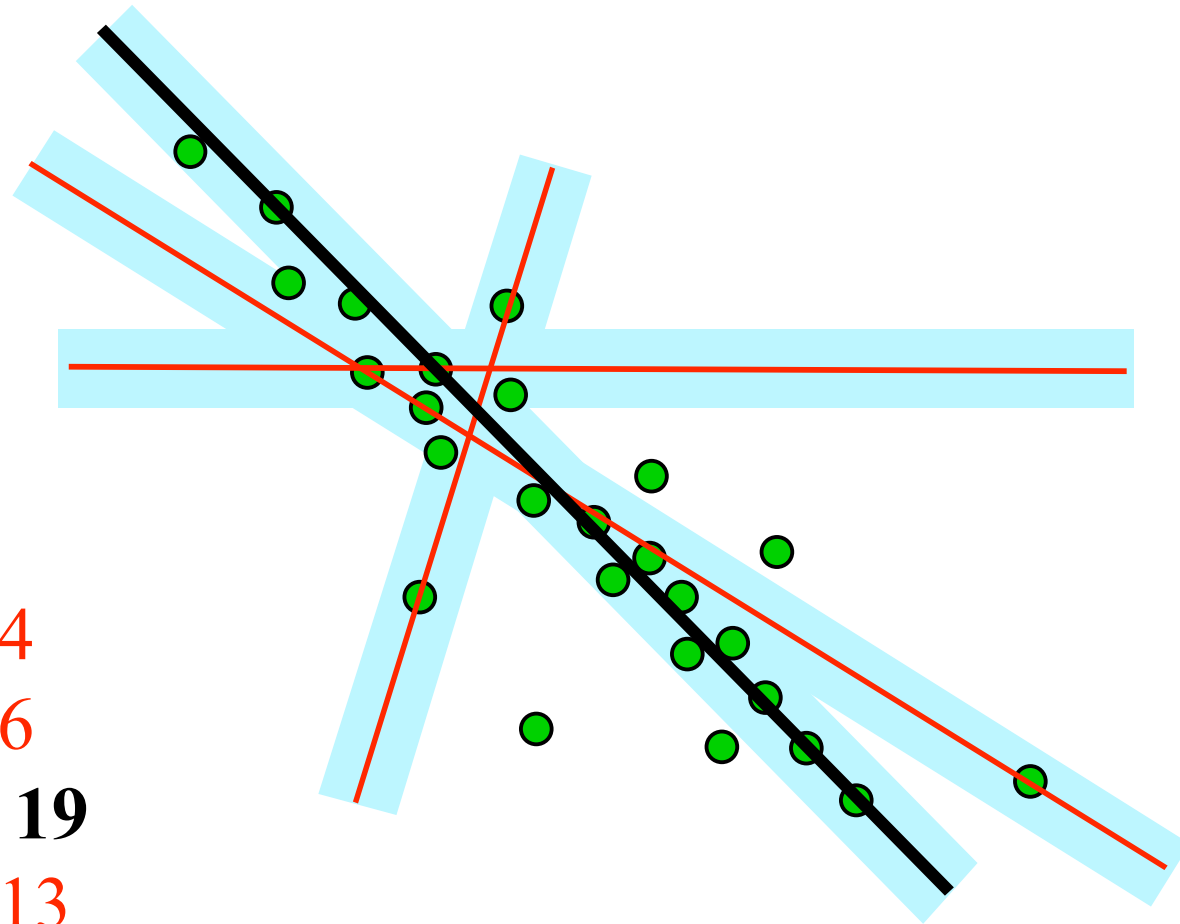
Ransac Procedure

Count = 13



Ransac Procedure

Count = 4
Count = 6
Count = 19
Count = 13



Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:

- s** — the smallest number of points required
- N** — the number of iterations required
- d** — the threshold used to identify a point that fits well
- T** — the number of nearby points required
to assert a model fits well

Until **N** iterations have occurred

Draw a sample of **s** points from the data
uniformly and at random

Fit to that set of **s** points

For each data point outside the sample

Test the distance from the point to the line
against **d** if the distance from the point to the line
is less than **d** the point is close

end

If there are **T** or more points close to the line
then there is a good fit. Refit the line using all
these points.

end

Use the best fit from this collection, using the
fitting error as a criterion

(Forsyth & Ponce)

How Many Samples to Choose?

e = probability that a point is an outlier

s = number of points in a sample

N = number of samples (we want to compute this)

p = desired probability that we get a good sample

Solve the following for N :

$$1 - (1 - (1 - e)^s)^N = p$$

Where in the world did that come from?

How Many Samples to Choose?

e = probability that a point is an outlier

s = number of points in a sample

N = number of samples (we want to compute this)

p = desired probability that we get a good sample

$$1 - \underbrace{(1 - (1 - e)^s)}_{}^N = p$$

**Probability that choosing
one point yields an inlier**

How Many Samples to Choose?

e = probability that a point is an outlier

s = number of points in a sample

N = number of samples (we want to compute this)

p = desired probability that we get a good sample

$$1 - \underbrace{(1 - (1 - e)^s)}_{\text{Probability of choosing } s \text{ inliers in a row (sample only contains inliers)}}^N = p$$

**Probability of choosing
 s inliers in a row (sample
only contains inliers)**

How Many Samples to Choose?

e = probability that a point is an outlier

s = number of points in a sample

N = number of samples (we want to compute this)

p = desired probability that we get a good sample

$$1 - \underbrace{(1 - (1 - e)^s)}_{\text{Probability that one or more points in the sample were outliers (sample is contaminated)}}^N = p$$

**Probability that one or more
points in the sample were outliers
(sample is contaminated).**

How Many Samples to Choose?

e = probability that a point is an outlier

s = number of points in a sample

N = number of samples (we want to compute this)

p = desired probability that we get a good sample

$$1 - \underbrace{(1 - (1 - e)^s)}_{\text{Probability that } N \text{ samples were contaminated.}}^N = p$$

**Probability that N samples
were contaminated.**

How Many Samples to Choose?

e = probability that a point is an outlier

s = number of points in a sample

N = number of samples (we want to compute this)

p = desired probability that we get a good sample

$$\underbrace{1 - (1 - (1 - e)^s)^N}_{\text{Probability that at least one sample was not contaminated}} = p$$

**Probability that at least
one sample was not
contaminated
(at least one sample of s
points is composed of only
inliers).**

How many samples?

Choose N so that, with probability p , at least one random sample is free from outliers. e.g. $p=0.99$

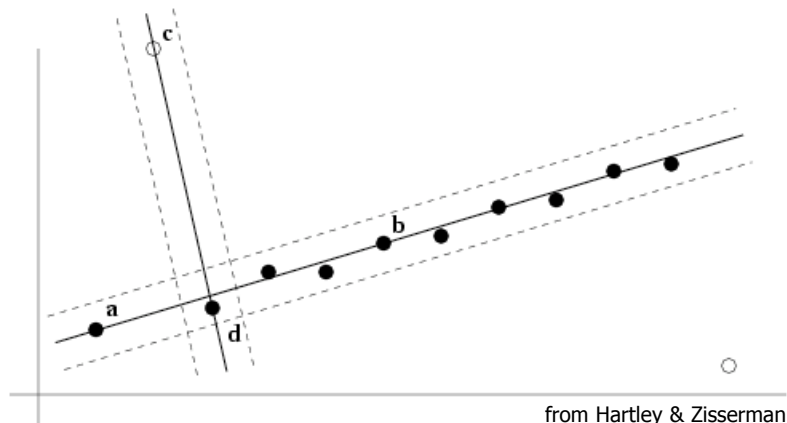
$$(1 - (1 - e)^s)^N = 1 - p$$

$$N = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)}$$

proportion of outliers e							
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Example: N for the line-fitting problem

- $n = 12$ points
- Minimal sample size $S = 2$
- 2 outliers: $e = 1/6 \Rightarrow 20\%$
- So $N = 5$ gives us a 99% chance of getting a pure-inlier sample
 - Compared to $N = 66$ by trying every pair of points



Acceptable consensus set?

- We have seen that we don't have to exhaustively sample subsets of points, we just need to randomly sample N subsets.
- However, typically, we don't even have to sample N sets!
- Early termination: terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e) * (\text{total number of data points})$$

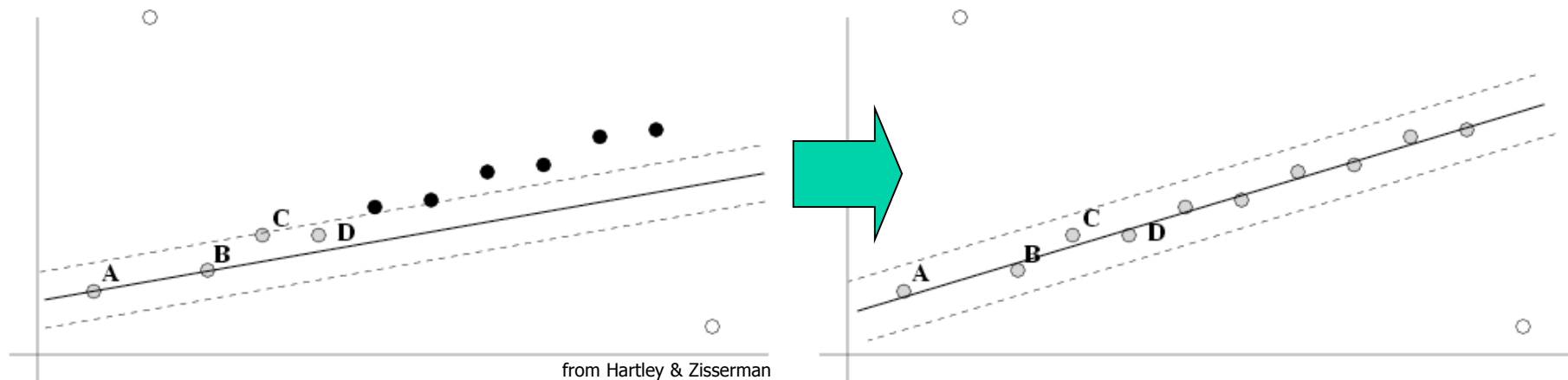
RANSAC: Picking Distance Threshold d

- Usually chosen empirically
- But...when measurement error is known to be Gaussian with mean 0 and variance \mathbf{S}^2 :
 - Sum of squared errors follows a χ^2 distribution with \mathbf{m} DOF, where \mathbf{m} is the DOF of the error measure (the *codimension*)
 - (*dimension + codimension*) = *dimension of parameter space*
 - E.g., $\mathbf{m} = 1$ for line fitting because error is perpendicular distance
 - E.g., $\mathbf{m} = 2$ for point distance
- Examples for probability $\mathbf{p} = 0.95$ that point is inlier

\mathbf{m}	Model	d^2
1	Line, fundamental matrix	$3.84 \mathbf{S}^2$
2	Homography, camera matrix	$5.99 \mathbf{S}^2$

After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers with greatest support
- Improve this initial estimate with Least Squares estimation over all inliers (i.e., standard minimization)
- Find inliers wrt that L.S. line, and compute L.S. one more time.



Practical Example

- Stabilizing aerial imagery using RANSAC
 - find corners in two images
 - hypothesize matches using NCC
 - do RANSAC to find matches consistent with an affine transformation
 - take the inlier set found and estimate a full projective transformation (homography)

Stabilization Application

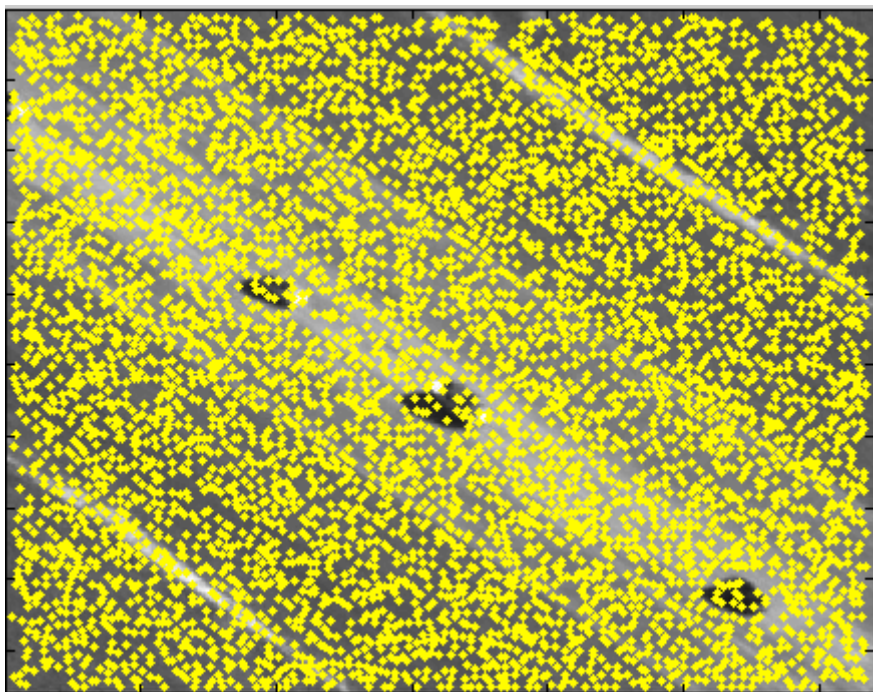
Input: two images from an aerial video sequence.



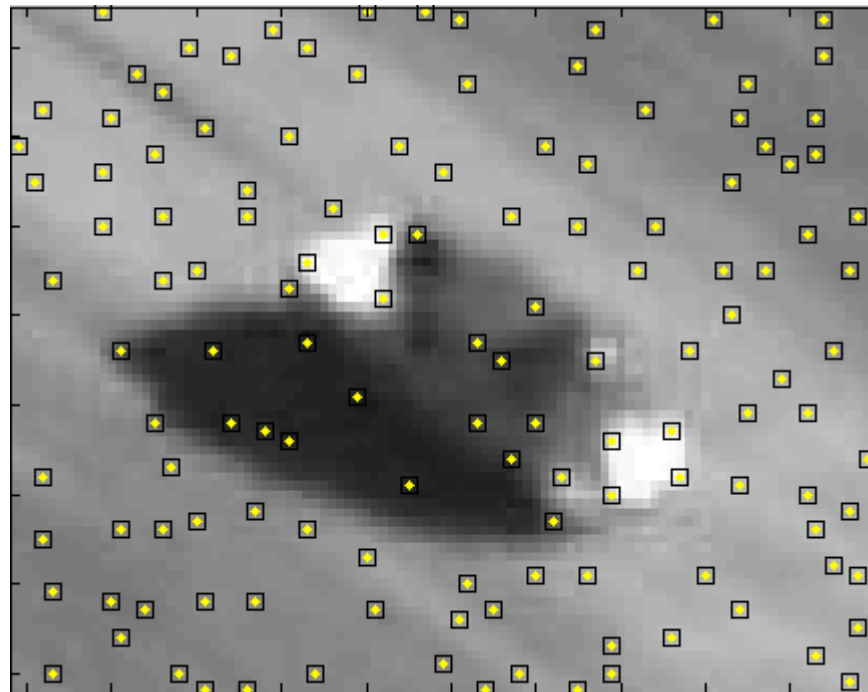
Note that the motion of the camera is “disturbing”

Stabilization Application

Step1: extract Harris corners from both frames. We use a small threshold for R because we want LOTS of corners (fodder for our next step, which is matching).



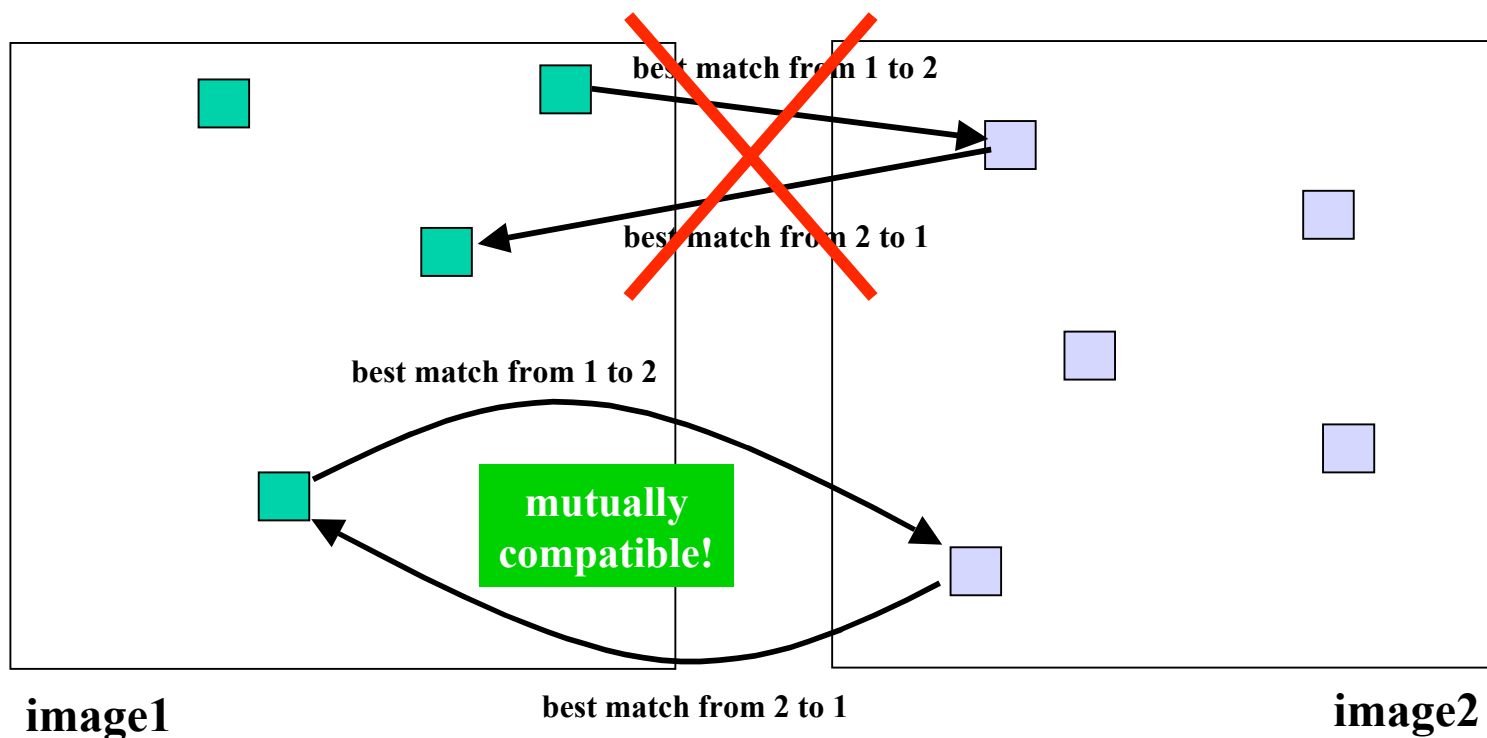
Harris corners for first frame



Detailed view

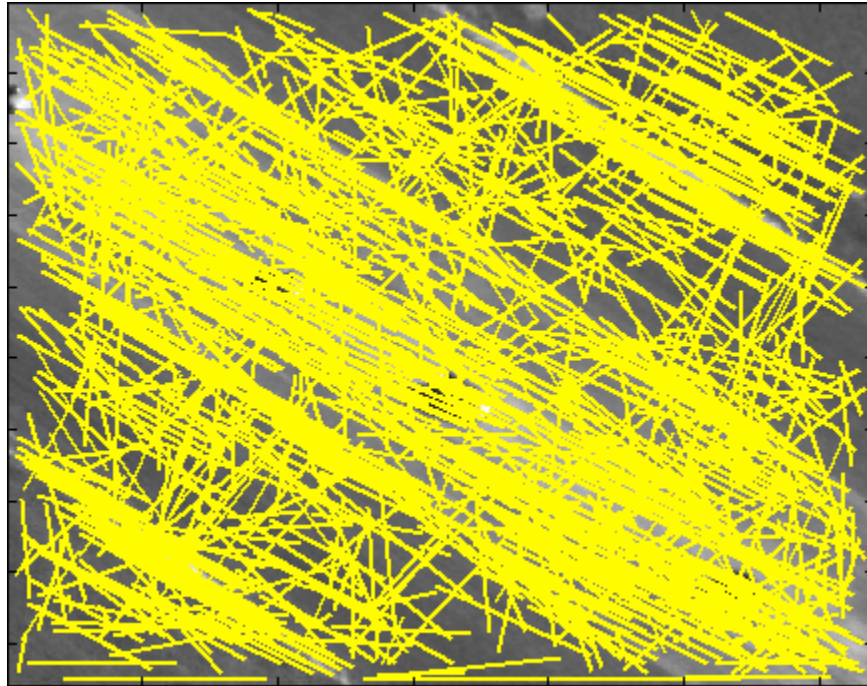
Stabilization Application

Step2: hypothesize matches. For each corner in image 1, look for matching intensity patch in image2 using NCC. Make sure matching pairs have highest NCC match scores in BOTH directions.



Stabilization Application

Step2: hypothesize matches.

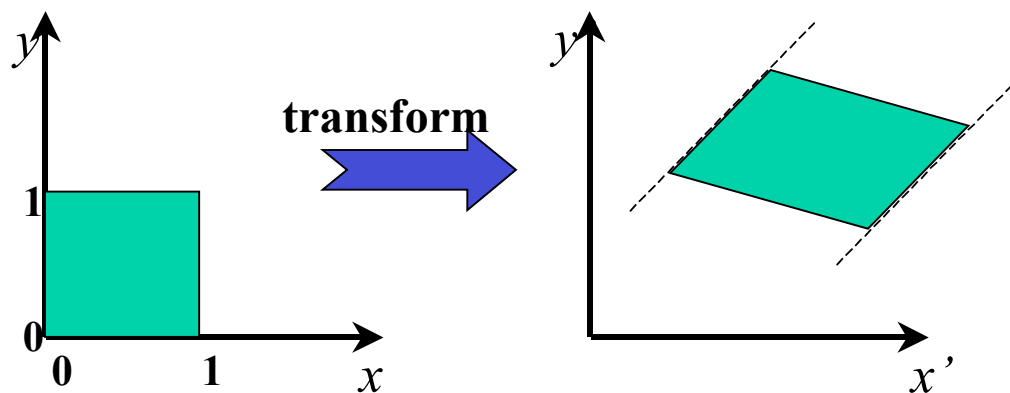


yikes!

As you can see, a lot of false matches get hypothesized. The job of RANSAC will be to clean this mess up.

Stabilization Application

Step3: Use RANSAC to robustly fit best affine transformation to the set of point matches.



$$\begin{aligned}x_i' &= a x_i + c y_i + c \\y_i' &= d x_i + e y_i + f\end{aligned}$$

How many unknowns?

How many point matches are needed?

Stabilization Application

Step3: Use RANSAC to robustly fit best affine transformation to the set of point matches.

Affine transformation has 6 degrees of freedom.

We therefore need 3 point matches [each gives 2 equations]

Randomly sample sets of 3 point matches. For each, compute the unique affine transformation they define. How?

Stabilization Application

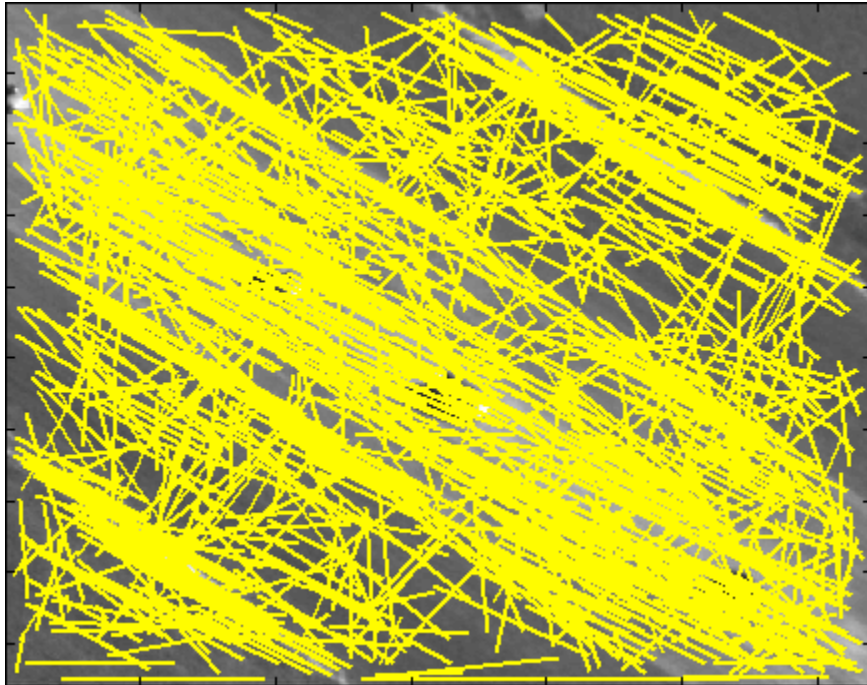
How to compute affine transformation from 3 point matches?
Use Least Squares! (renewed life for a nonrobust approach)

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i & 0 & 0 & 0 \\ \sum x_i y_i & \sum y_i^2 & \sum y_i & 0 & 0 & 0 \\ \sum x_i & \sum y_i & \sum 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum x_i^2 & \sum x_i y_i & \sum x_i \\ 0 & 0 & 0 & \sum x_i y_i & \sum y_i^2 & \sum y_i \\ 0 & 0 & 0 & \sum x_i & \sum y_i & \sum 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \sum x_i x_i' \\ \sum y_i x_i' \\ \sum x_i' \\ \sum x_i y_i' \\ \sum y_i y_i' \\ \sum y_i' \end{bmatrix}$$

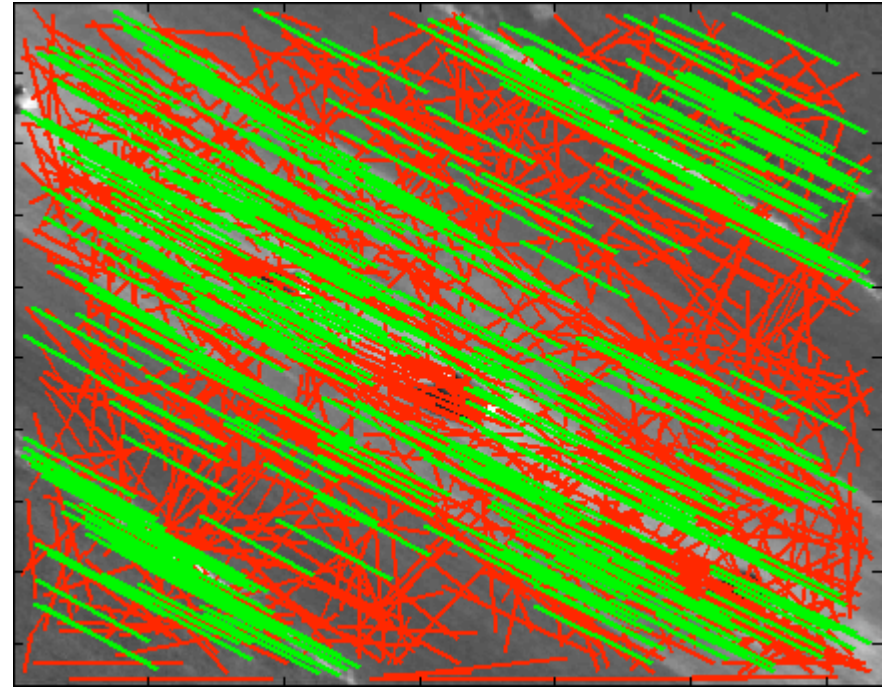
Then transform all points from image1 to image2 using that computed transformation, and see how many other matches confirm the hypothesis.

Repeat N times.

Stabilization Application



original point matches



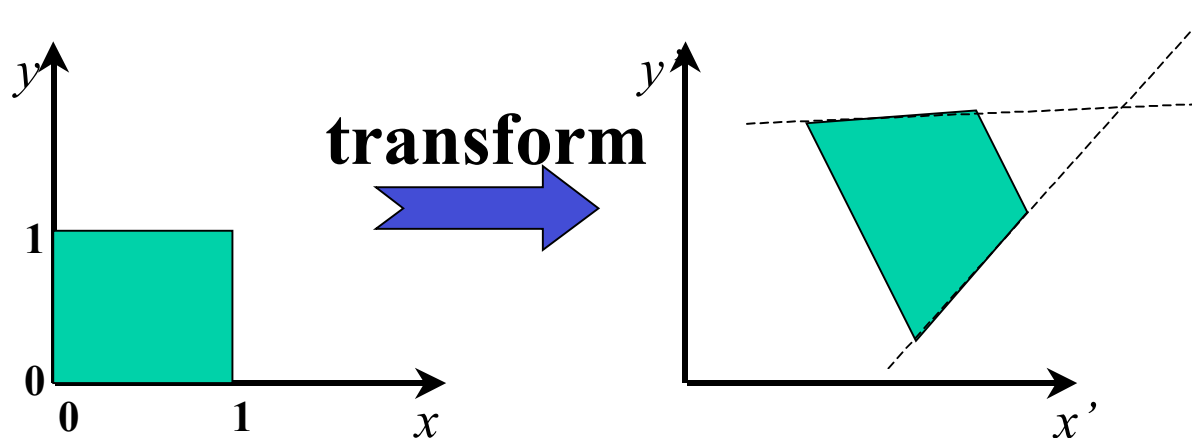
labels from RANSAC

green: inliers

red: outliers

Stabilization Example

Step4: Take inlier set labeled by RANSAC, and now use least squares to estimate a projective transformation that aligns the images. (we will discuss this ad nauseum in a later lecture).



Projective Transformation

Stabilization Example

Step4: estimate projective transformation that aligns the images.

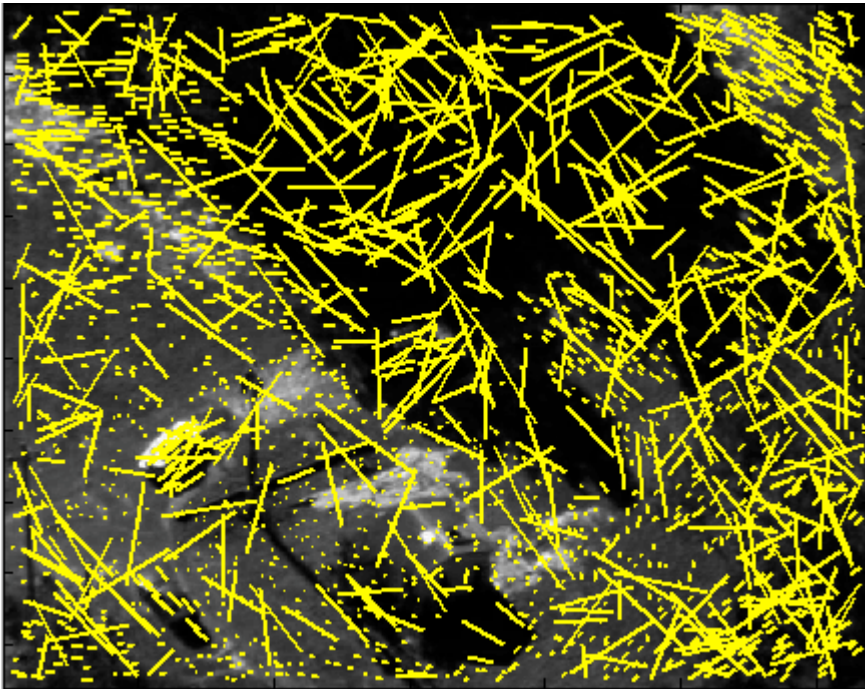


Now it is easier for people (and computers) to see the moving objects.

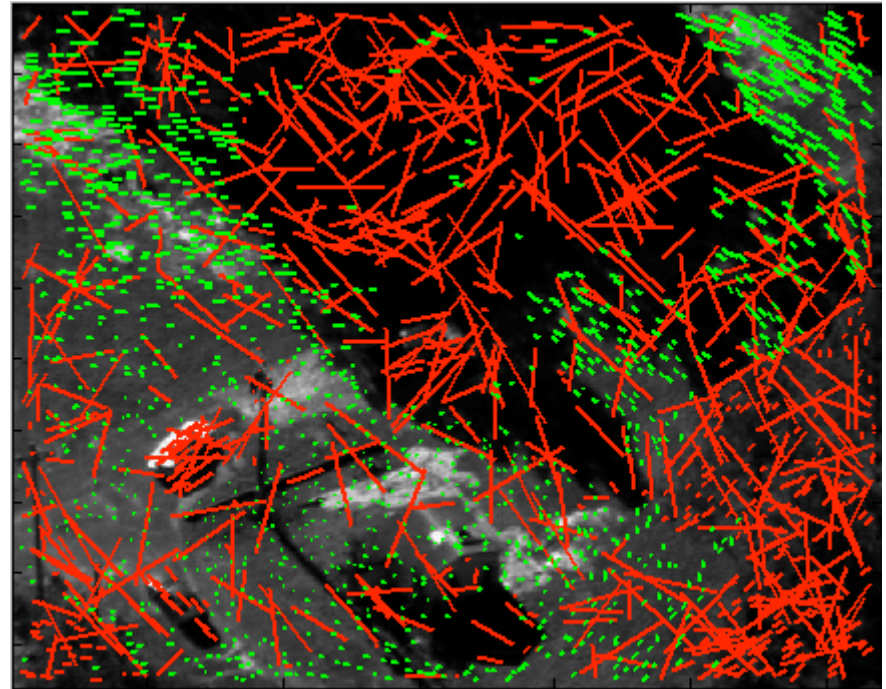
Stabilization Examples



Stabilization Examples



original point matches



labels from RANSAC

green: inliers

red: outliers

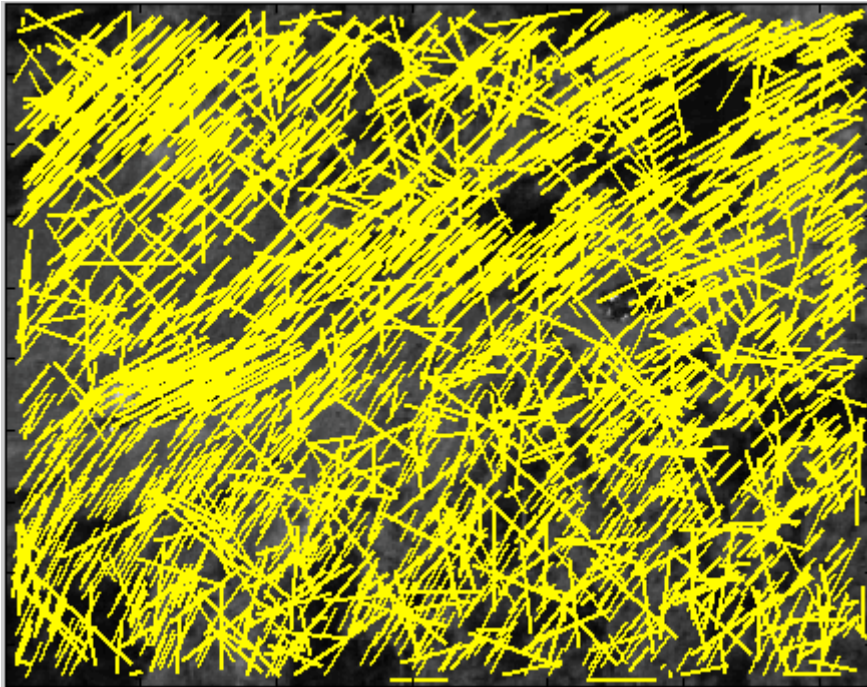
Stabilization Examples



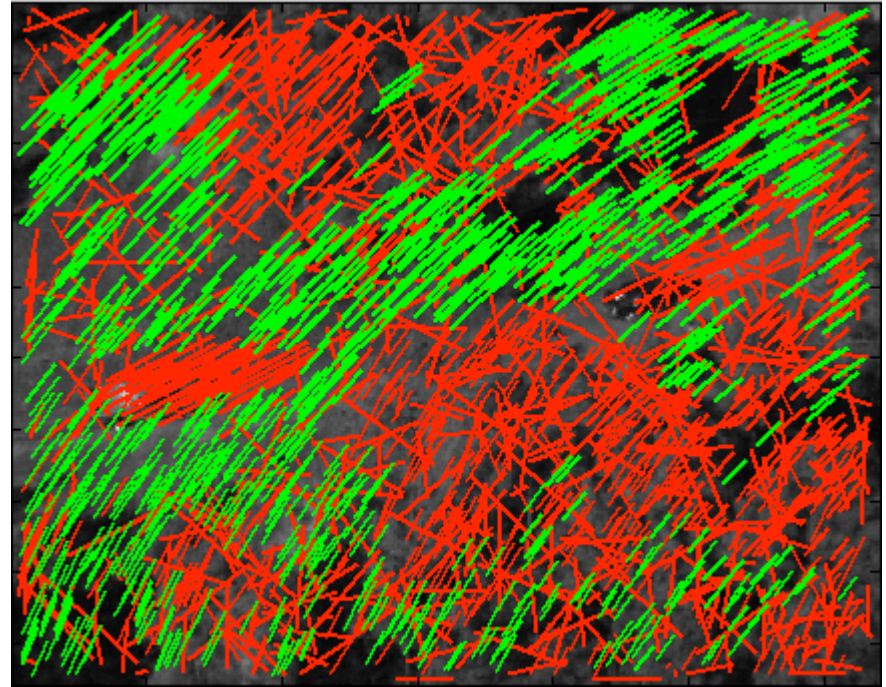
Stabilization Examples



Stabilization Examples



original point matches



labels from RANSAC

green: inliers
red: outliers

Stabilization Examples

