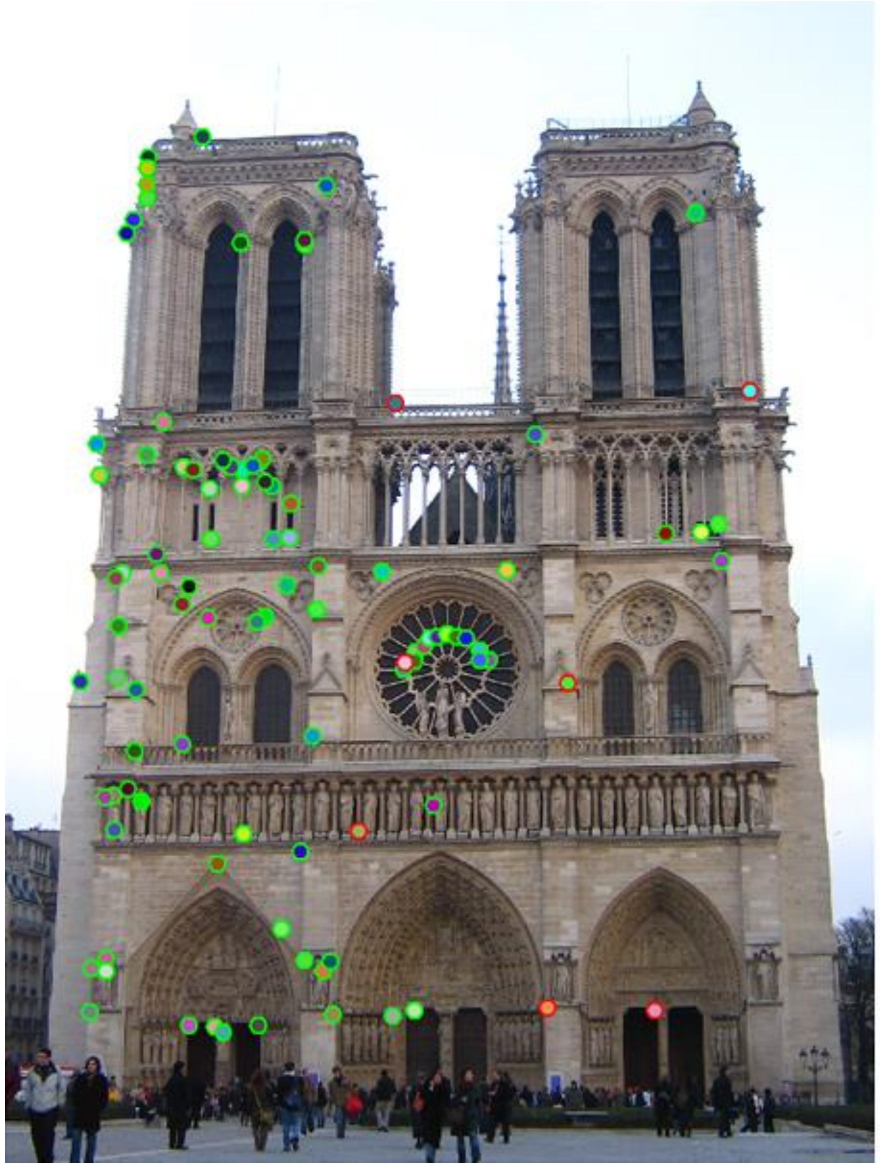
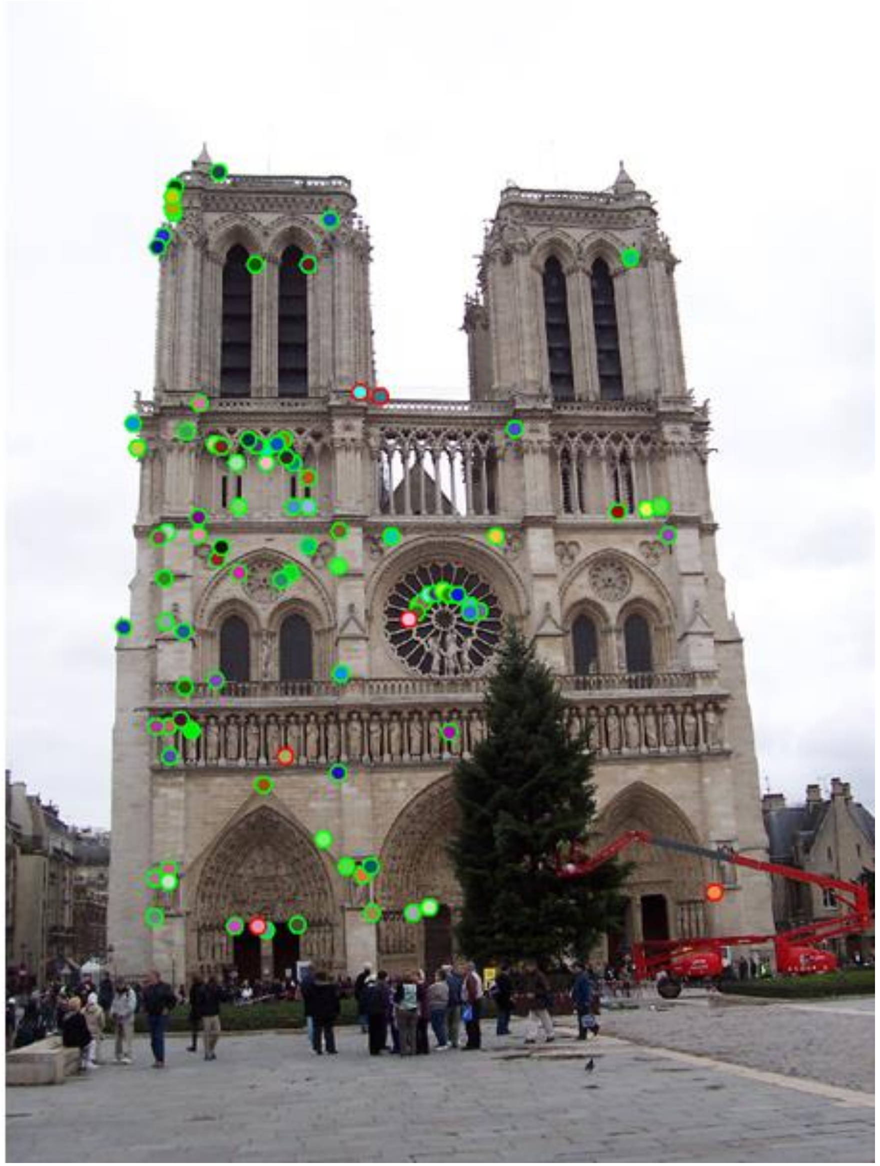
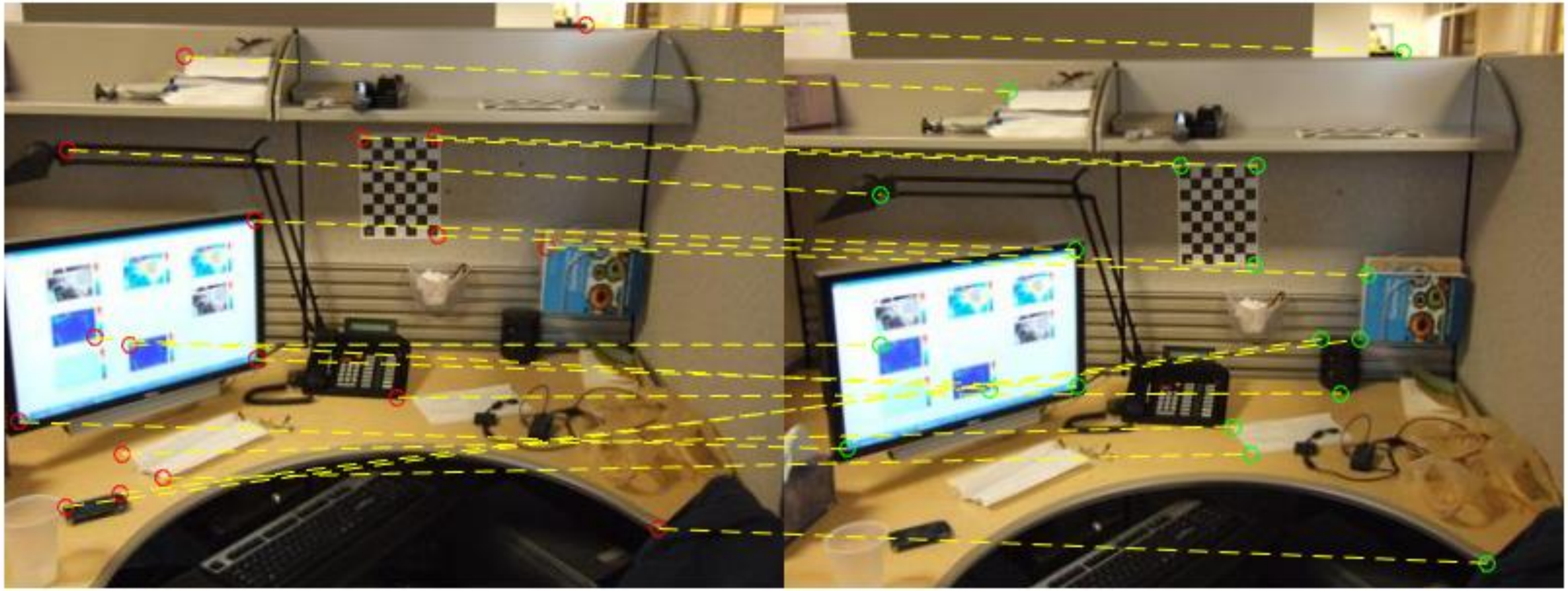
RANSAC
Matching Points

Given matches, what Next?

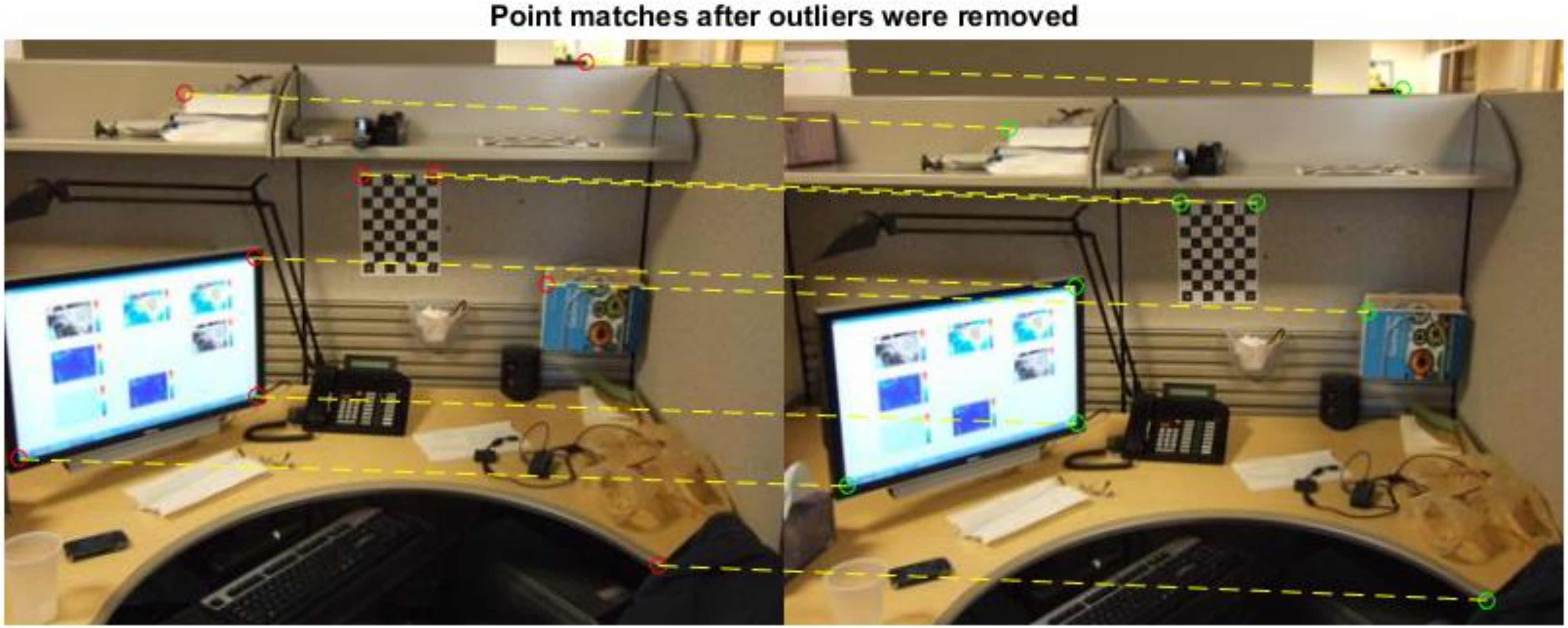




Stereo images with outliers



Stereo images without outliers



Fitting and Alignment

Fitting:

Find the parameters of a model that best fit the data.

Alignment:

Find the parameters of the transformation that best aligns matched points.

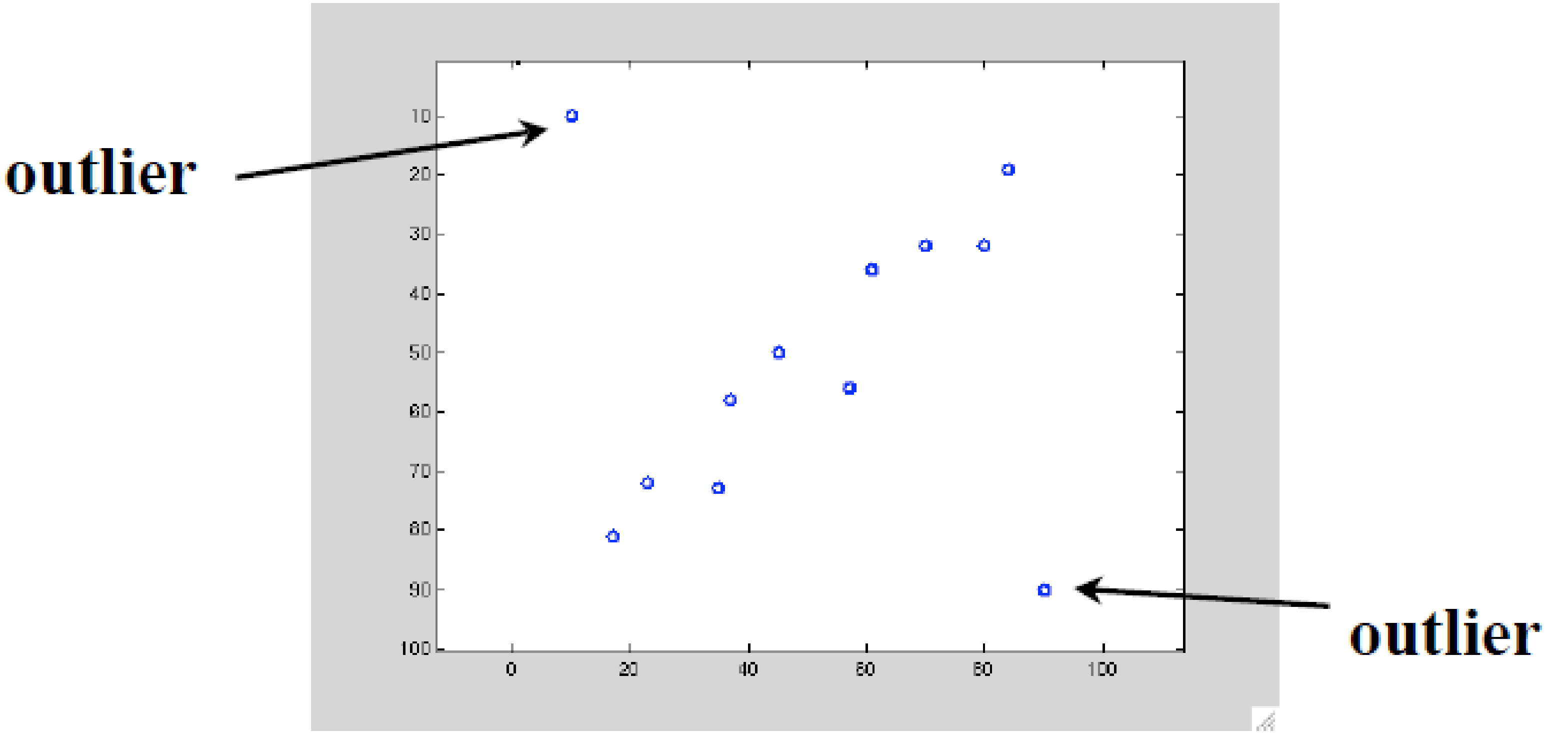
General Strategy:

- Least-Squares estimation from point correspondences

But there are problems with that approach….

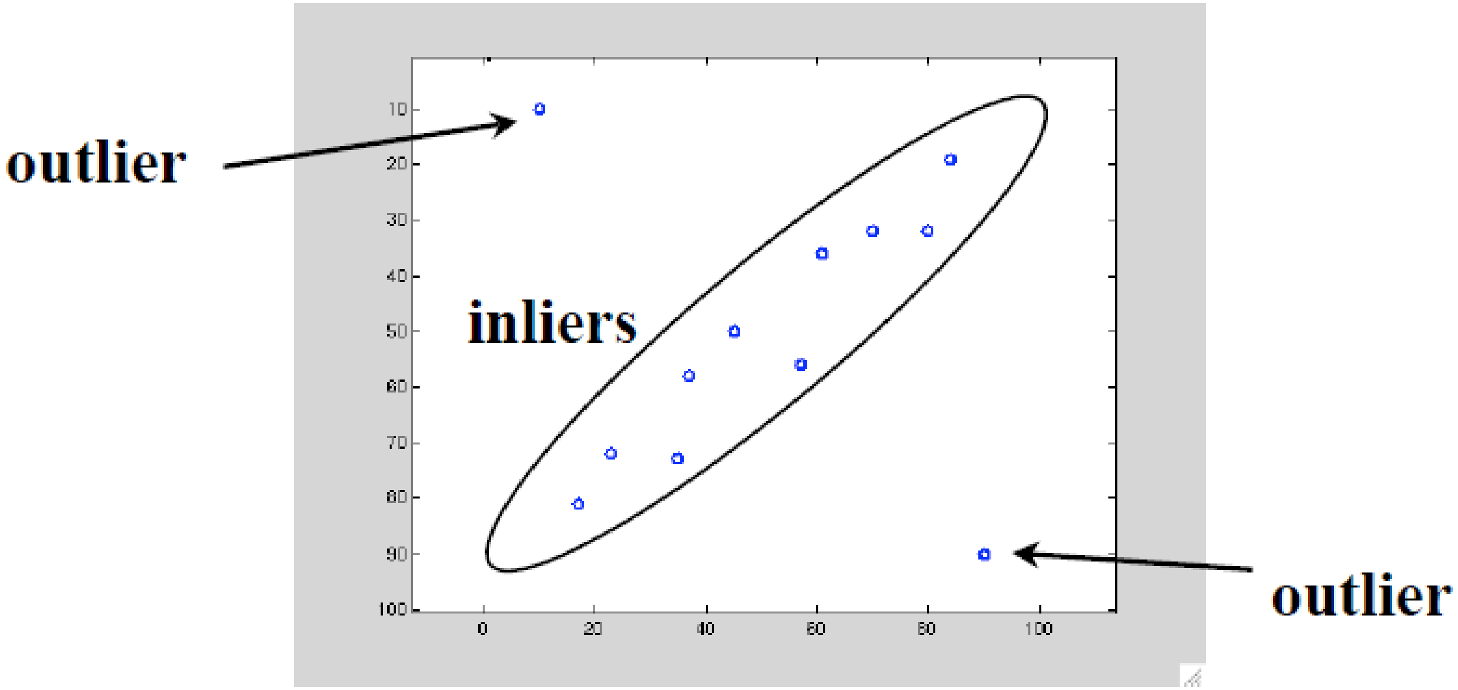
Problem : Outliers

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



Bad Data => Outliers

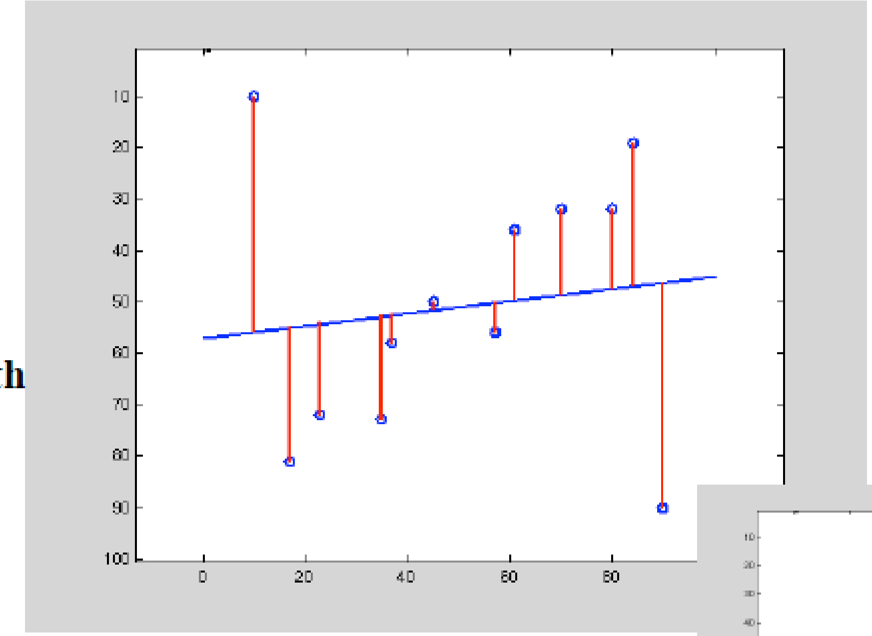
- Loosely speaking, outliers are points that don't "fit" the model. Points that do fit are called "inliers"



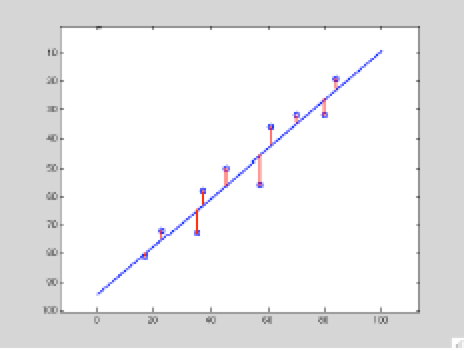
Problem with Outliers

- 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.

Robust Estimation

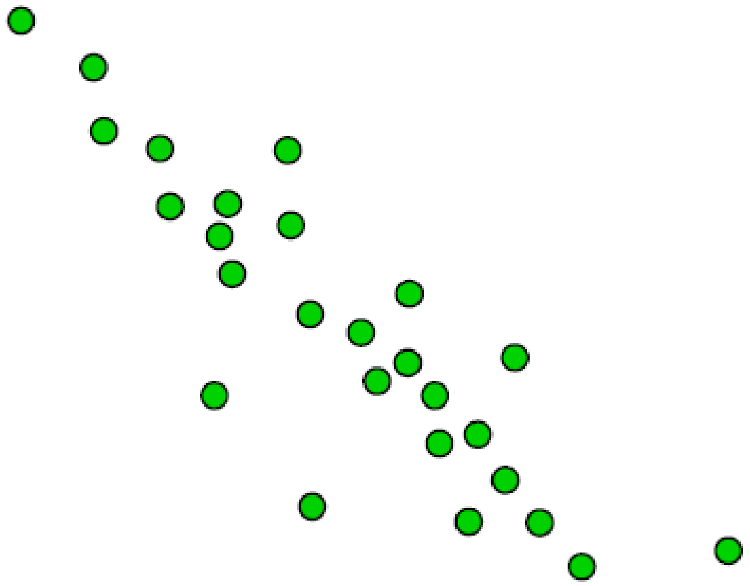
- View estimation as a two-stage process:
  - Classify data points as outliers or inliers
  - Fit model to inliers while ignoring outliers

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.

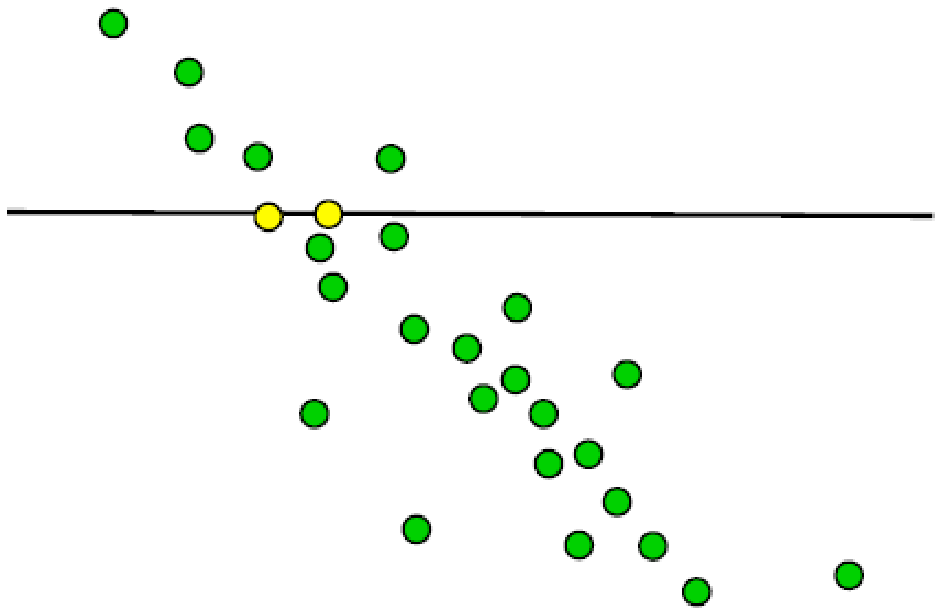
RANSAC

- RANdom SAmple Consensus
- Approach: we want to avoid the impact of outliers, so let's look for "inliers", and use only those.
- Intuition: if an outlier is chosen to compute the current fit, then the resul.ng line won't have much support from rest of the points.

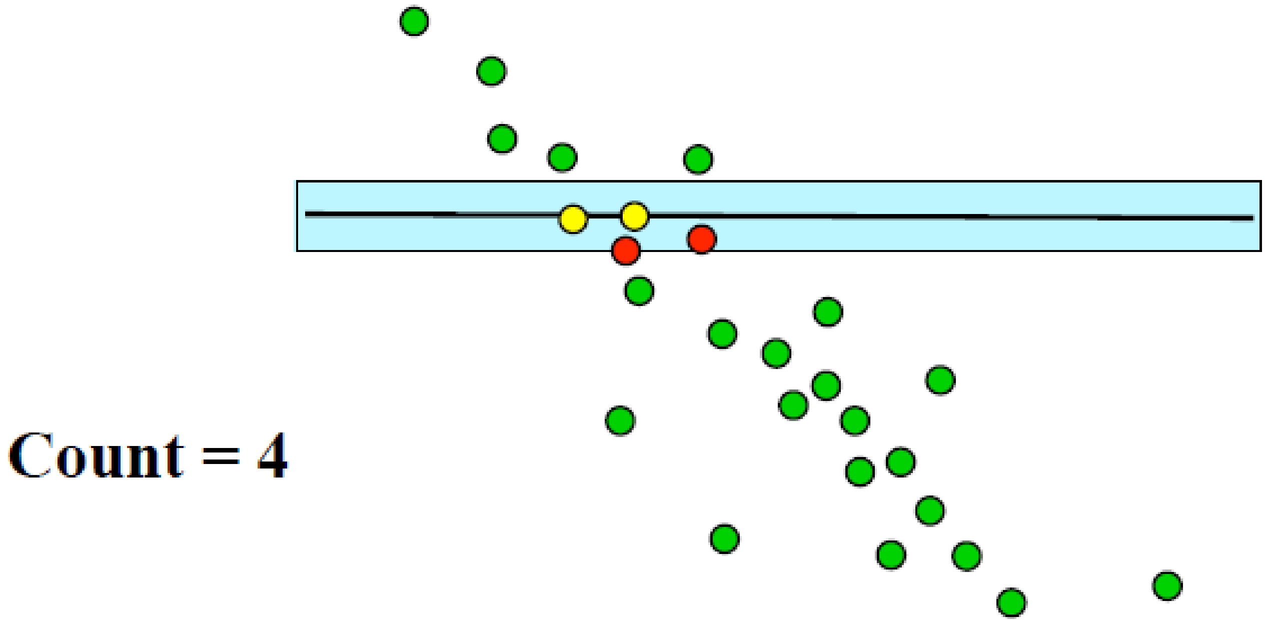
RANSAC Procedure



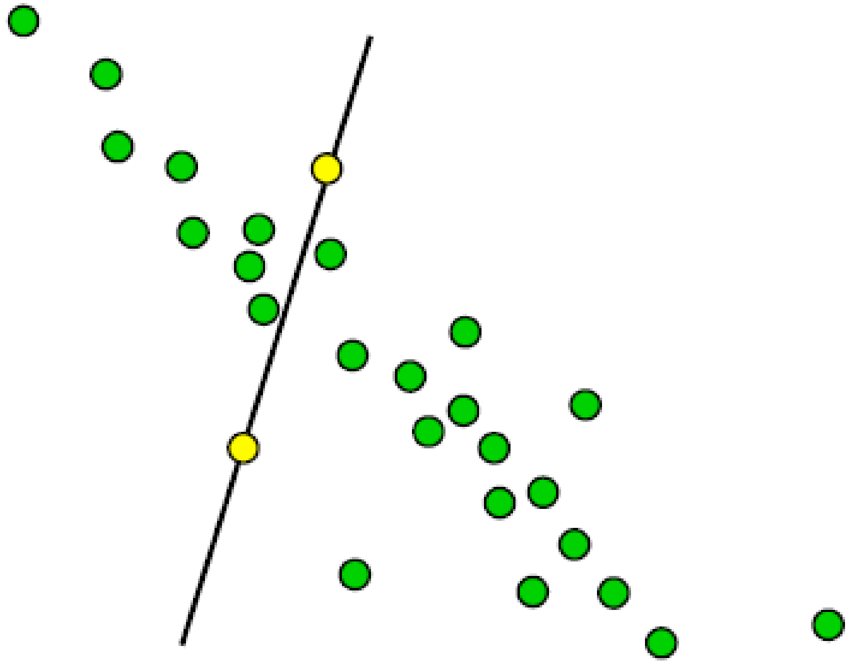
RANSAC Procedure



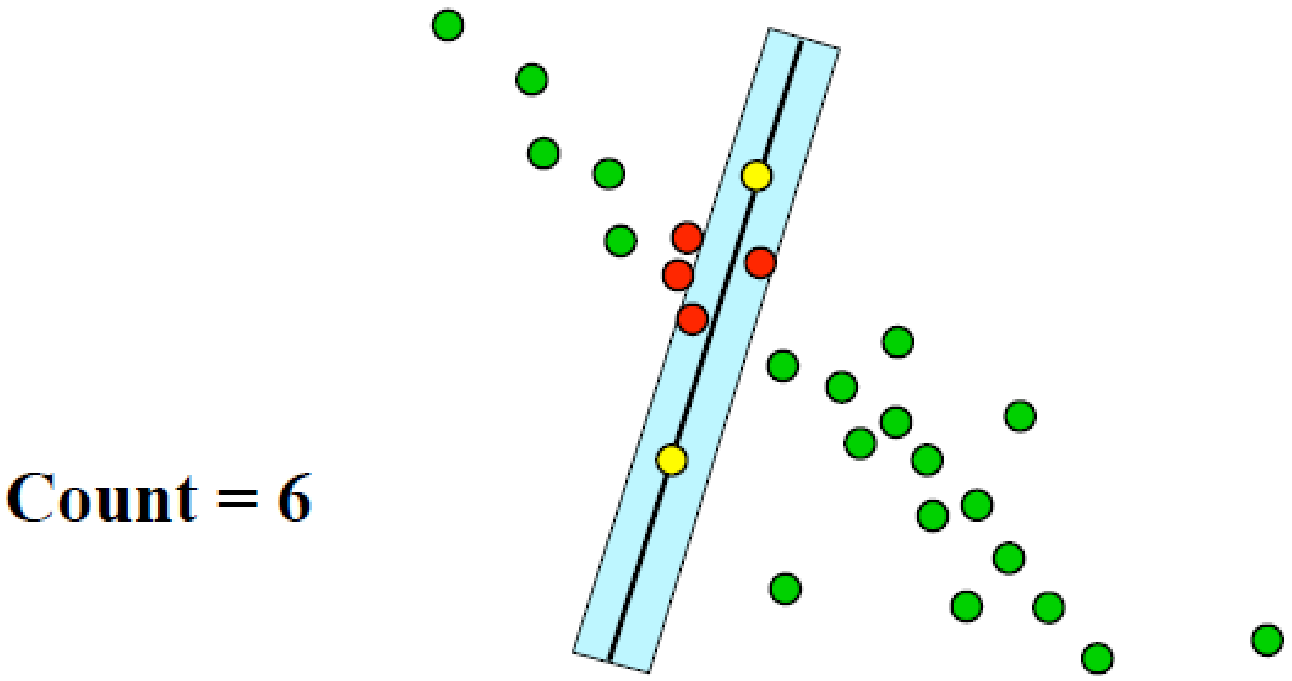
RANSAC Procedure



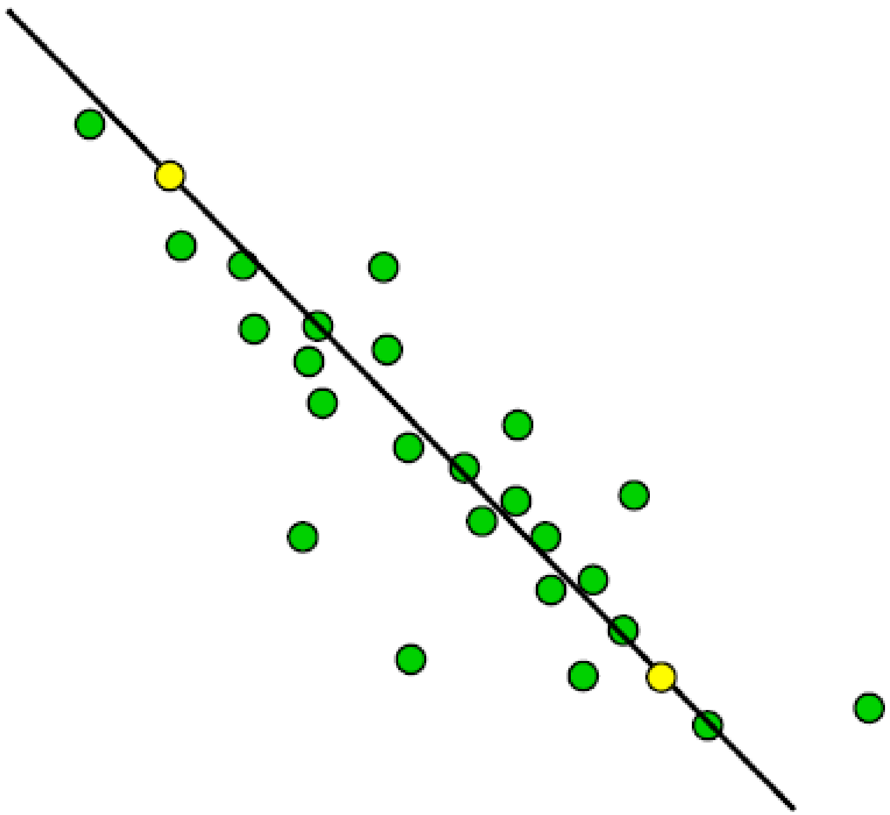
RANSAC Procedure



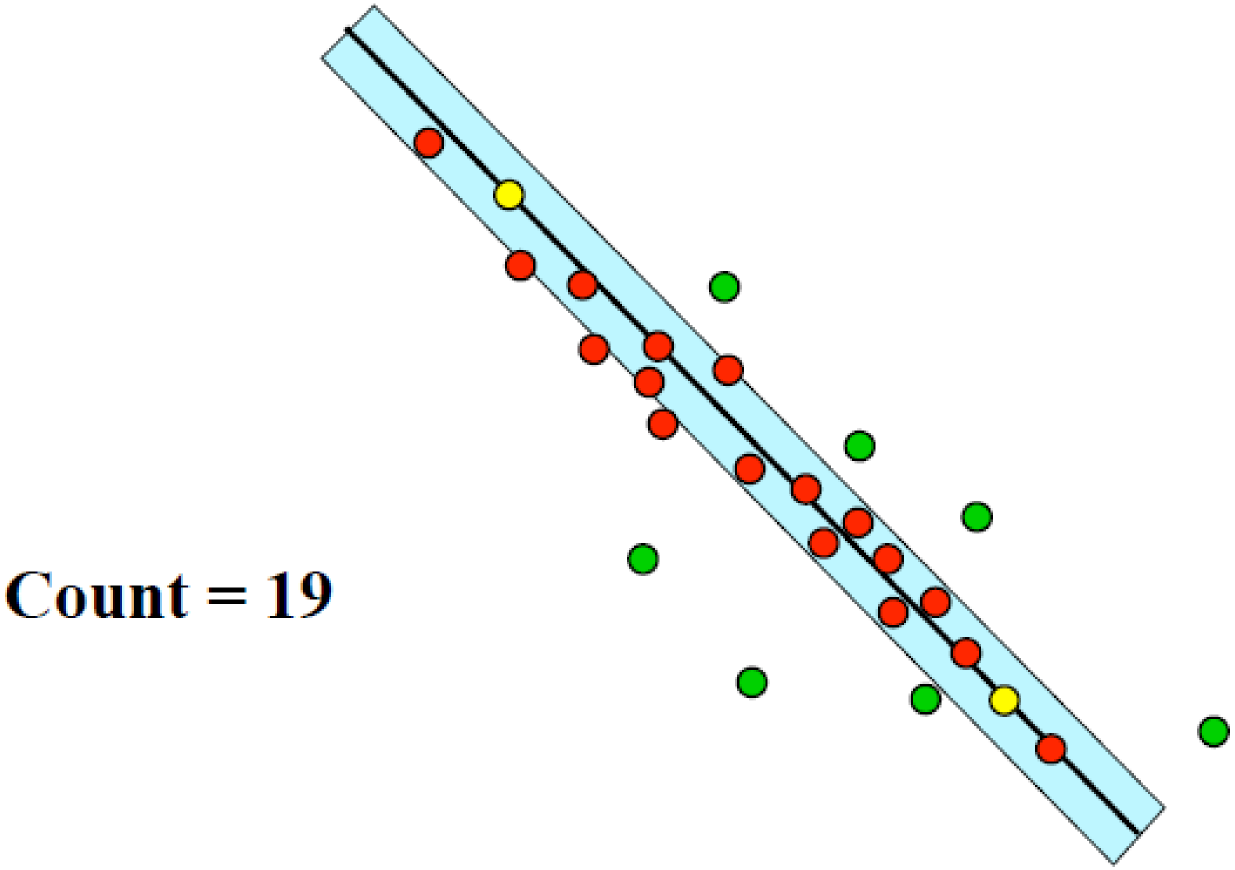
RANSAC Procedure



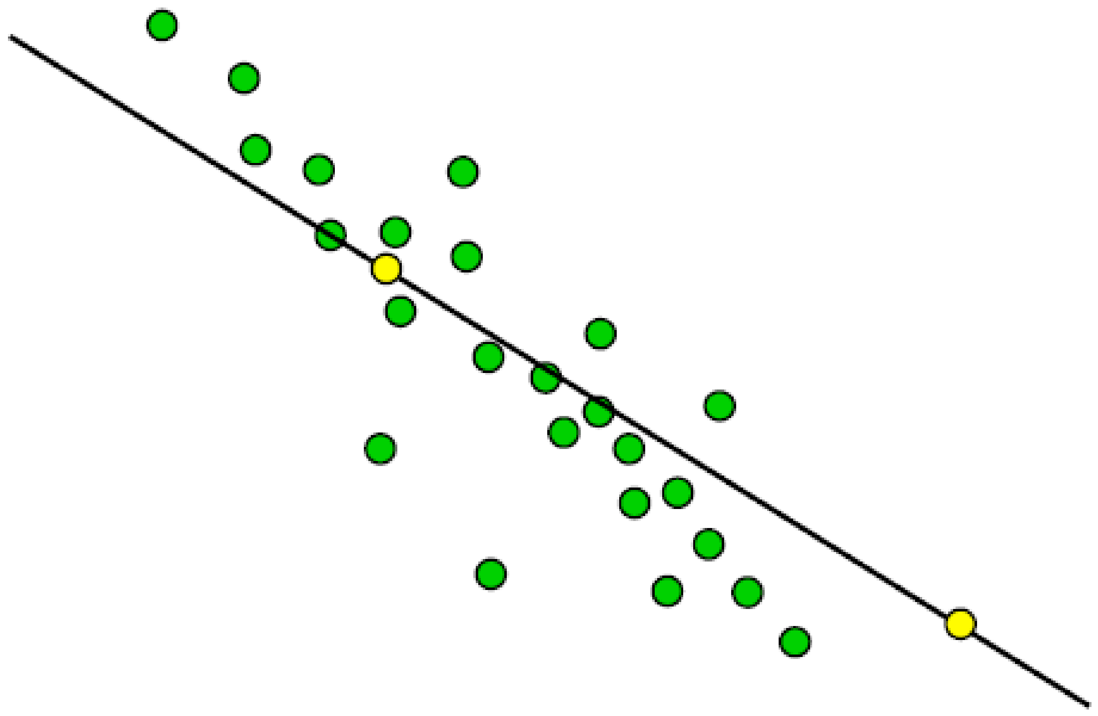
RANSAC Procedure



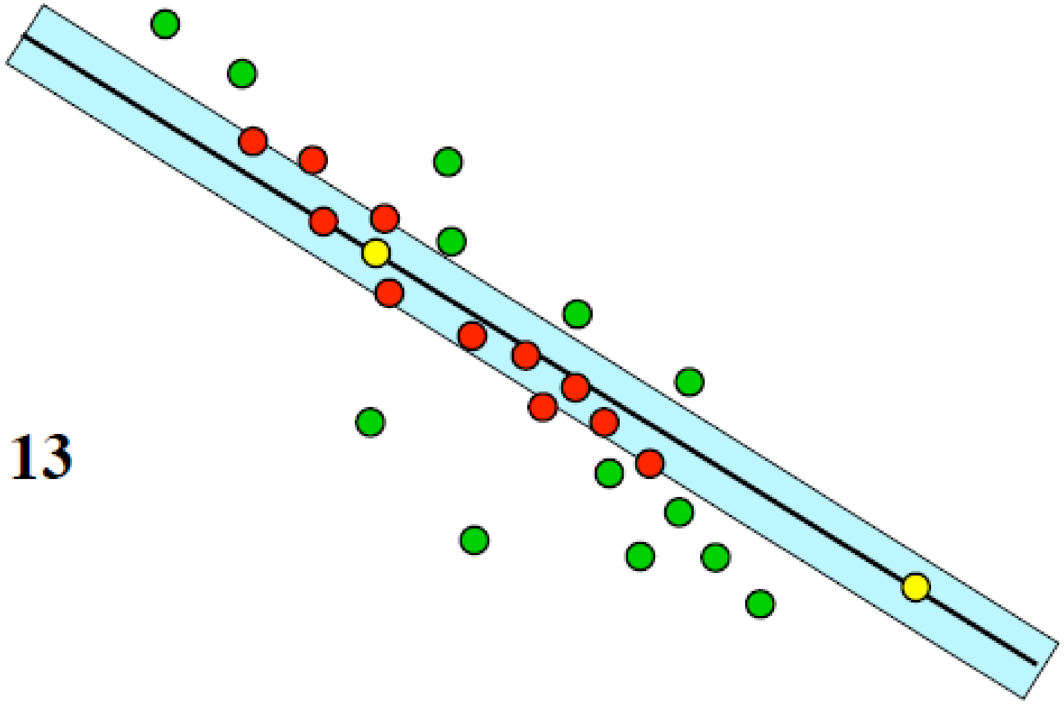
RANSAC Procedure



RANSAC Procedure

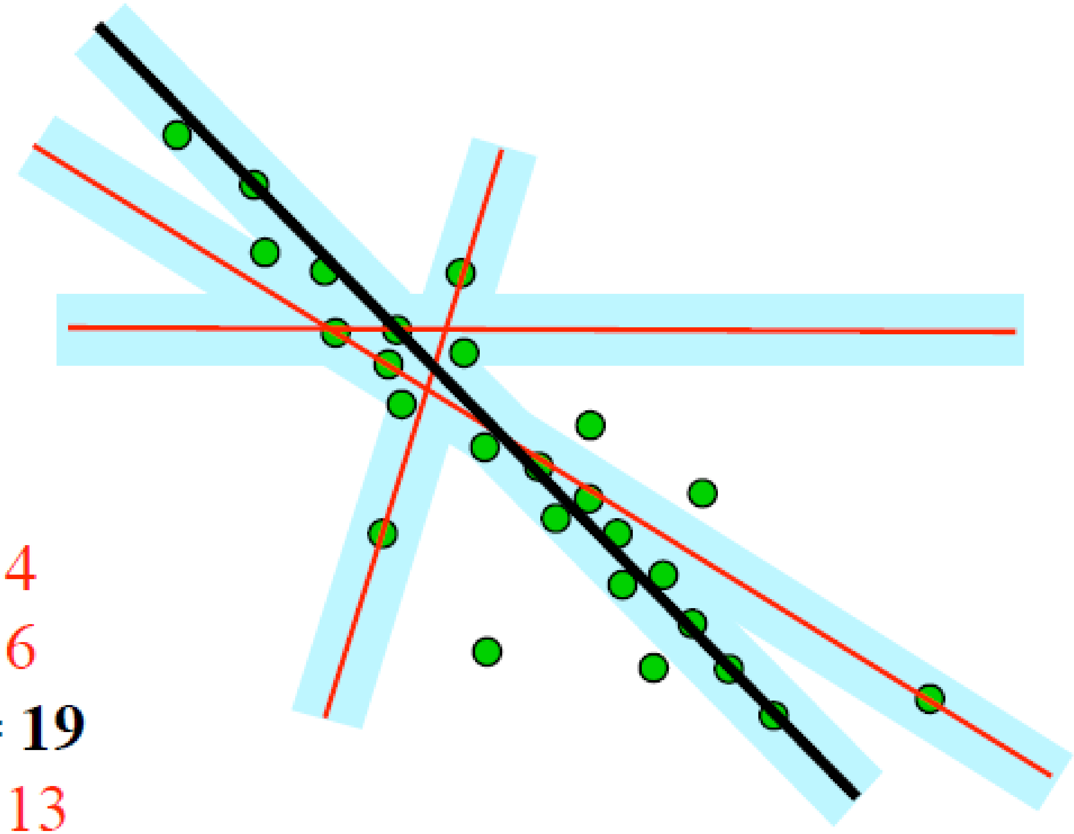


RANSAC Procedure



Count = 13

RANSAC Procedure

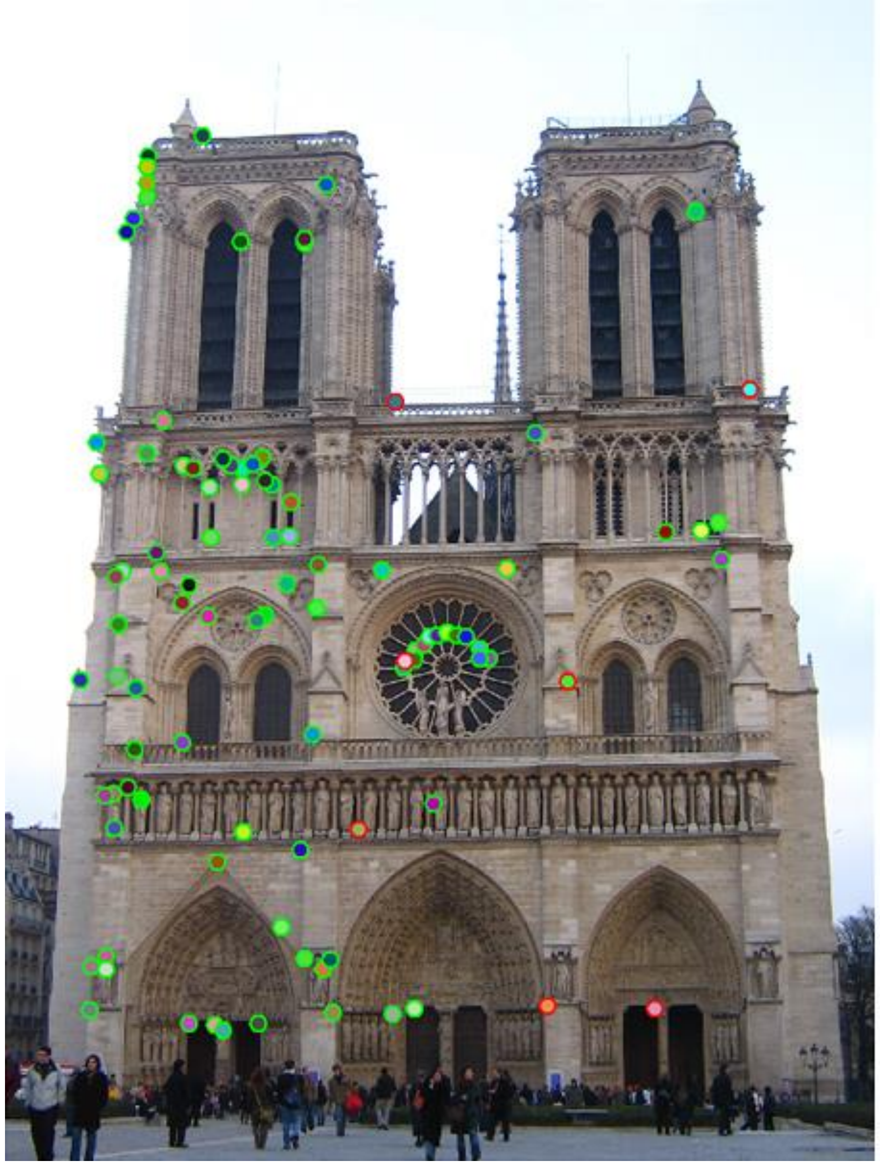


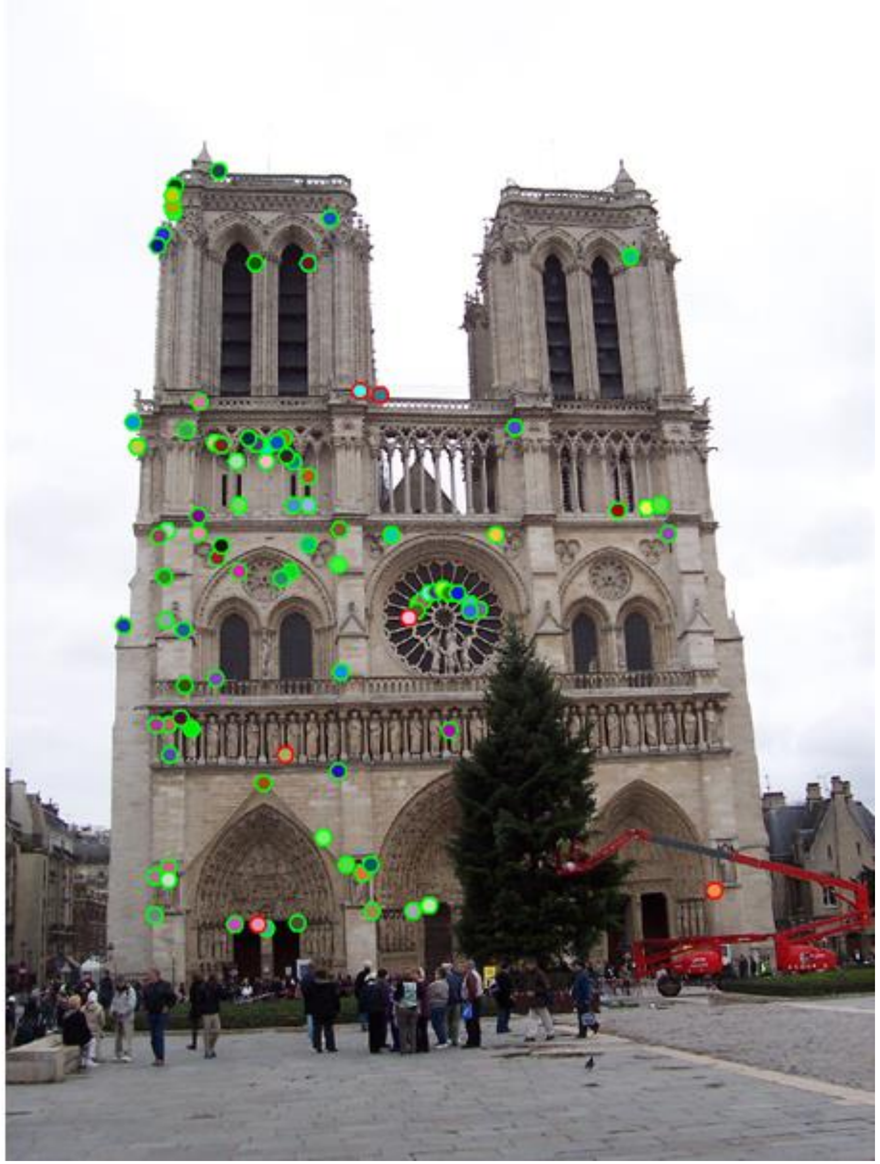
Count = 4
Count = 6
Count = 19
Count = 13

RANSAC - *Loop*

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Find *inliers* to this transformation
3. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers
4. Keep the transformation with the largest number of inliers

Given matches, what Next?

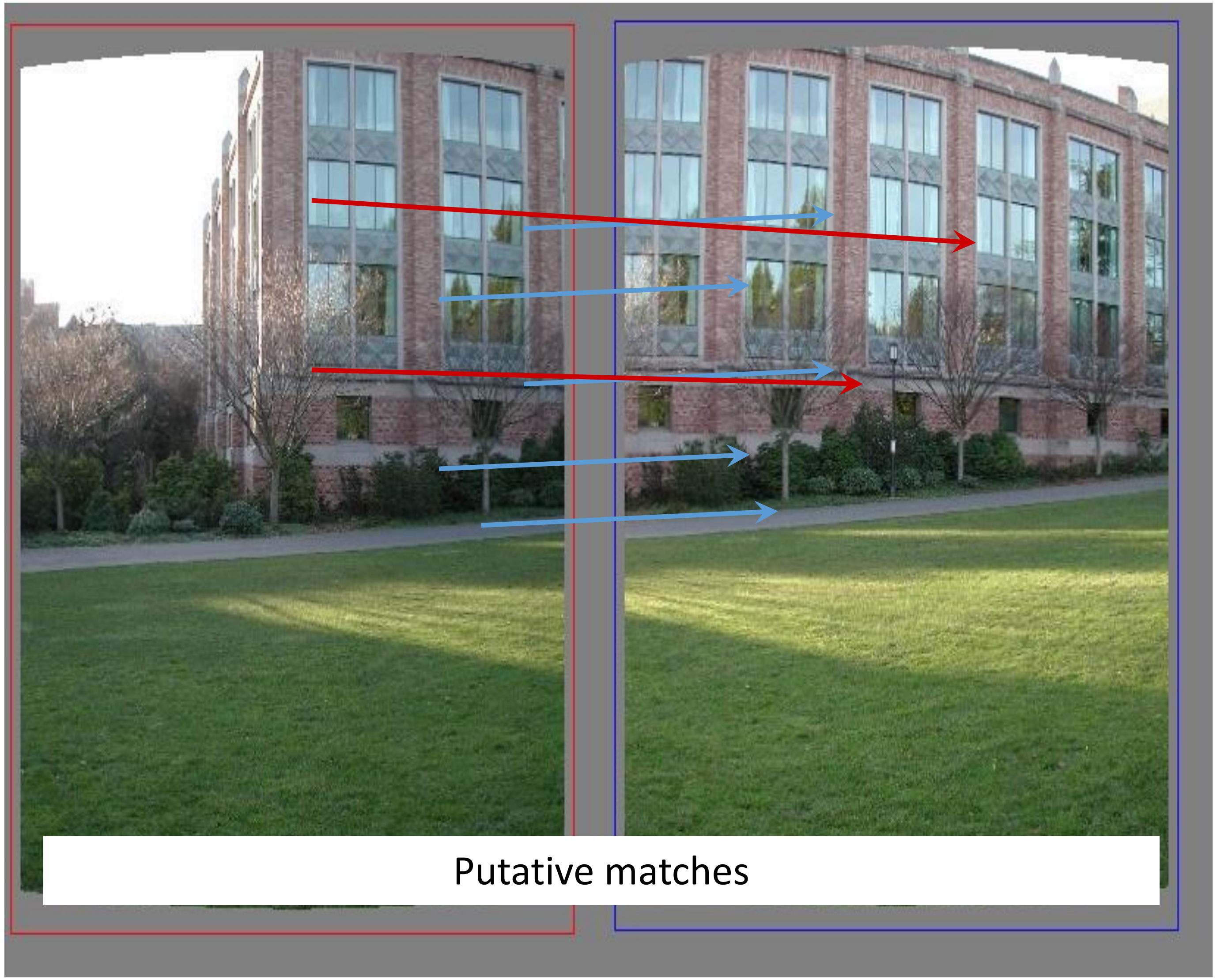




Algorithm 15.4: RANSAC: fitting lines using random sample consensus

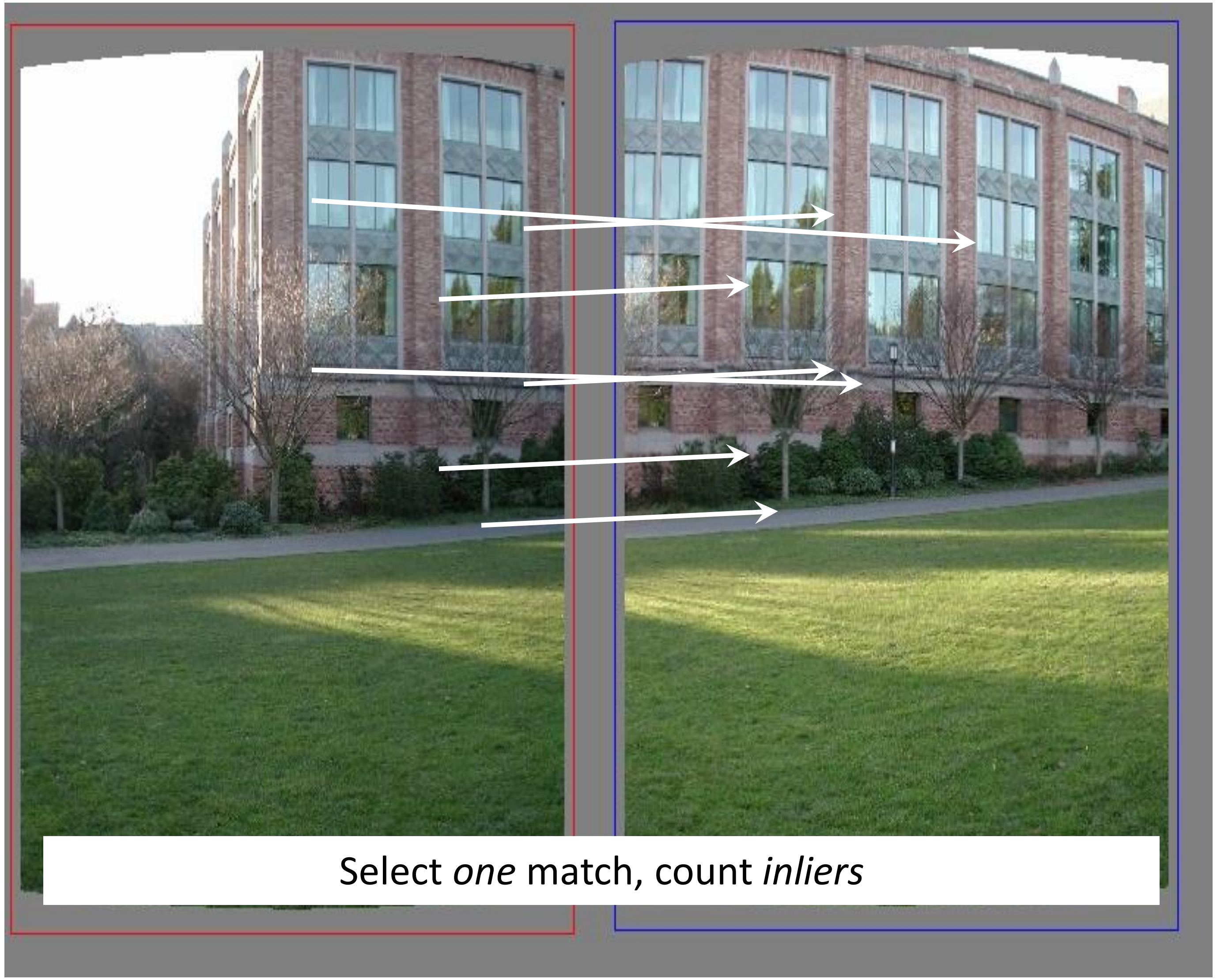
```
Determine:
    n — the smallest number of points required
    k — the number of iterations required
    t — the threshold used to identify a point that fits well
    d — the number of nearby points required
      to assert a model fits well
Until k iterations have occurred
    Draw a sample of n points from the data
      uniformly and at random
    Fit to that set of n points
    For each data point outside the sample
      Test the distance from the point to the line
        against t; if the distance from the point to the line
        is less than t, the point is close
    end
    If there are d 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
```

RANSAC example: Translation



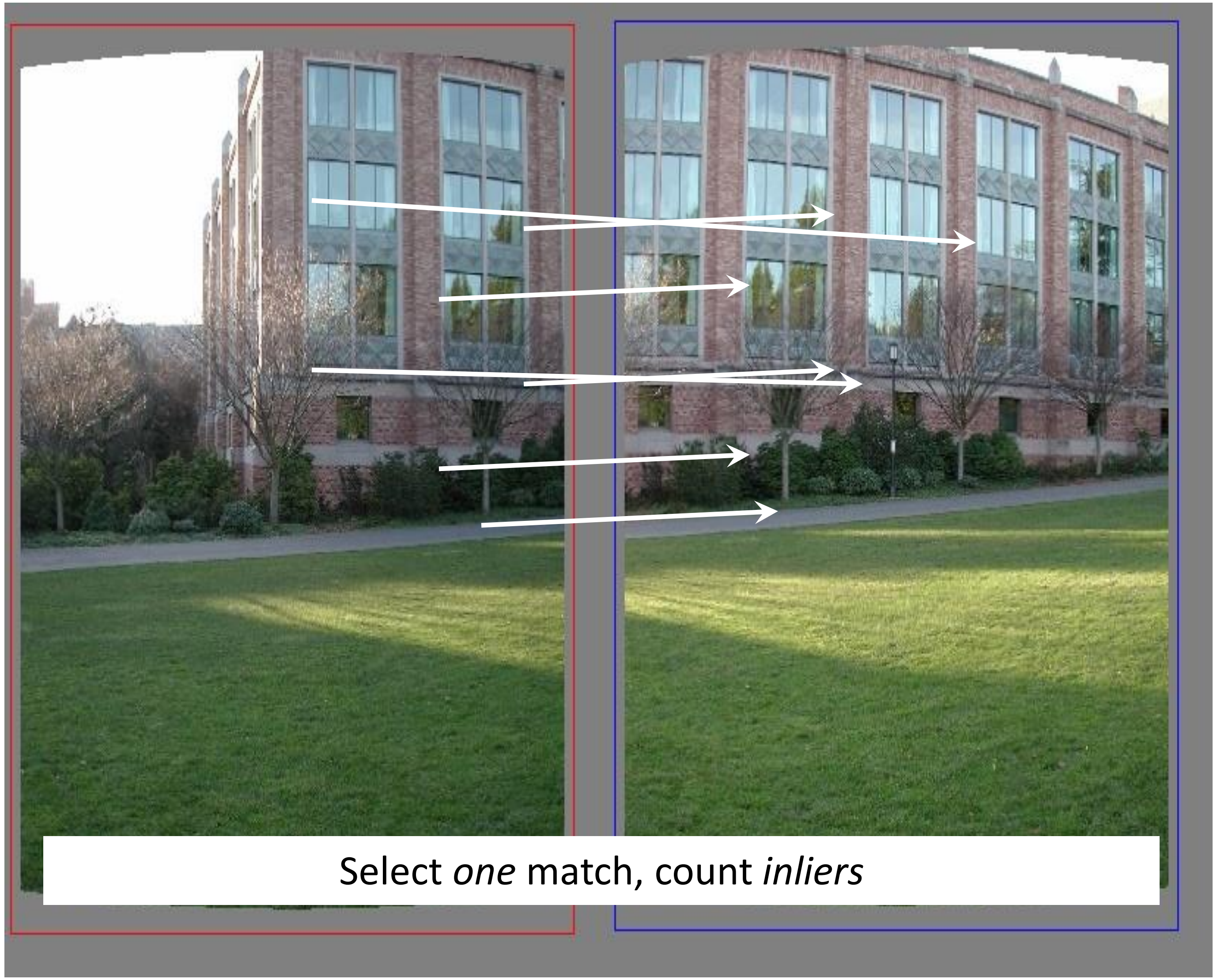
Putative matches

RANSAC example: Translation

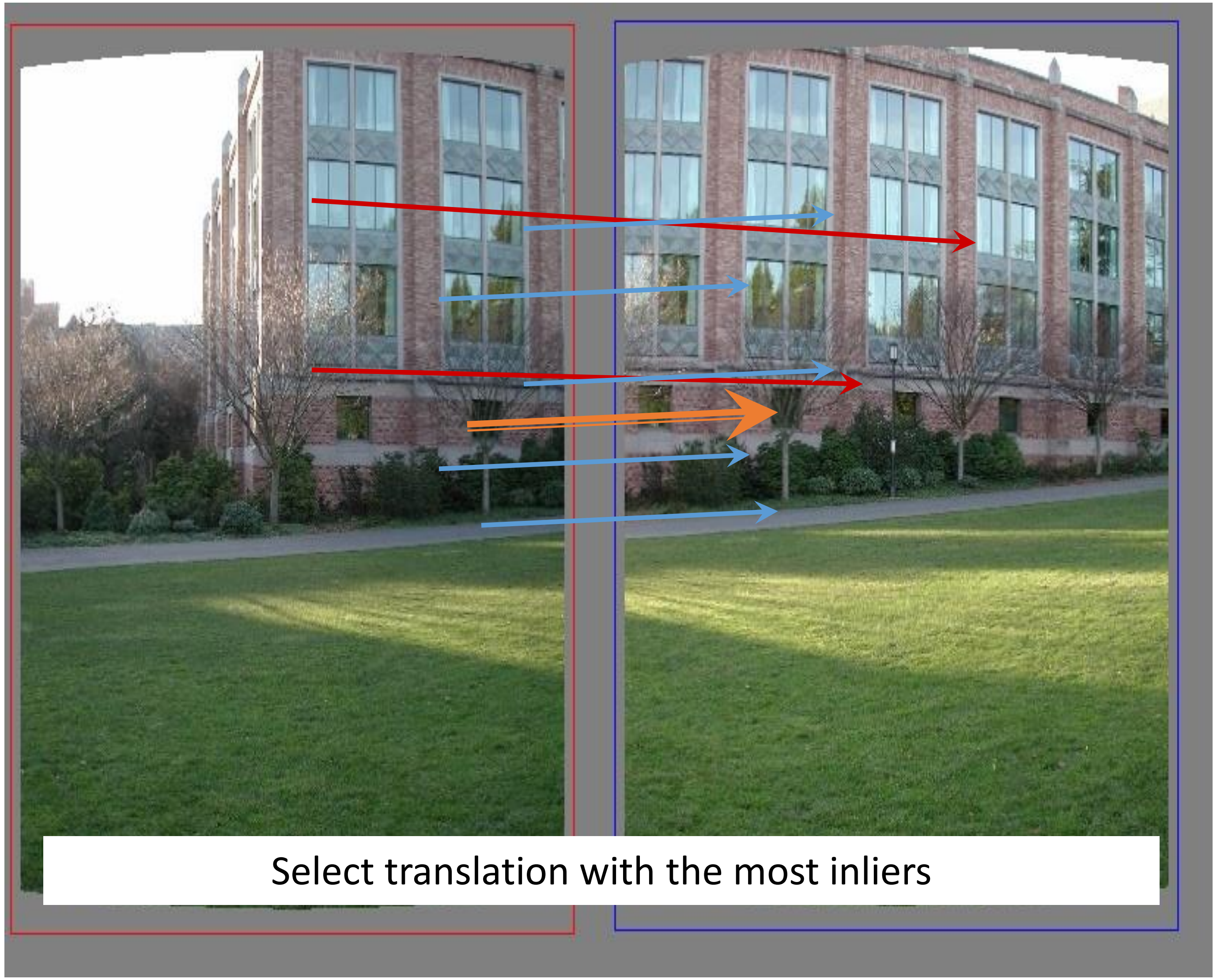


Select *one* match, count *inliers*

RANSAC example: Translation

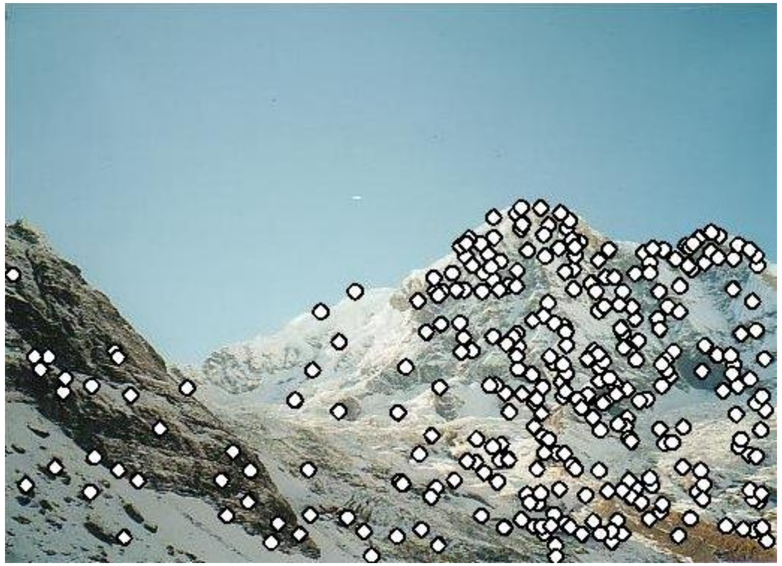


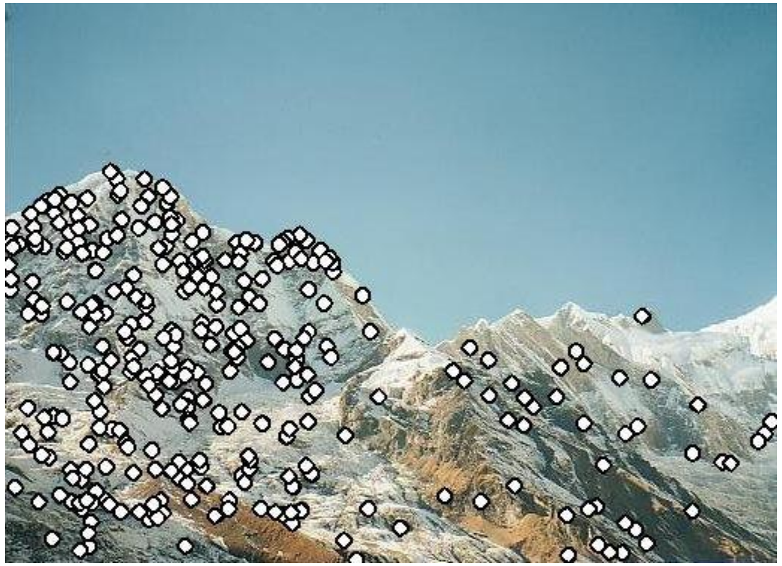
RANSAC example: Translation



Robust feature-based alignment

Robust feature-based alignment



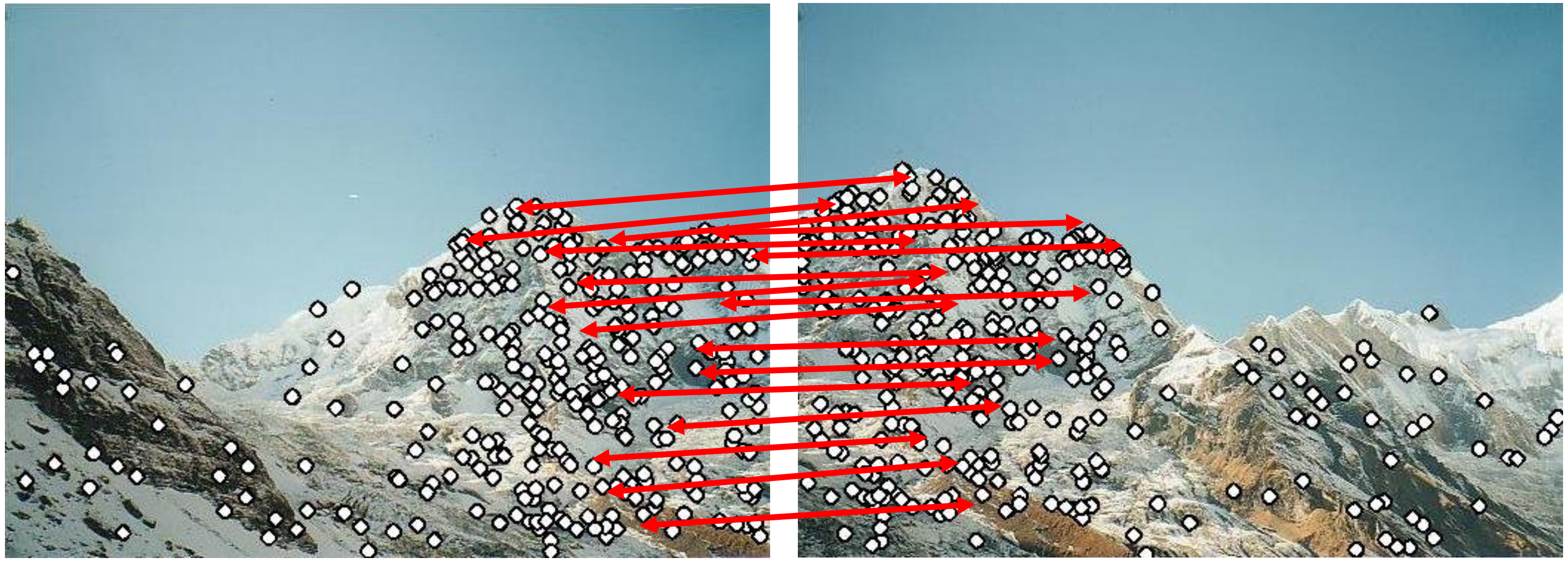


- Extract features

Robust feature-based alignment

- Extract features
- Compute *putative matches*

Robust feature-based alignment



- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T*
  - *Verify* transformation (search for other matches consistent with *T*)

Robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation *T*
  - *Verify* transformation (search for other matches consistent with *T*)

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:

$$1 - \left(1 - (1 - e)^{s}\right)^{N} = p$$

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

$$1 - \left(1 - \left(\underbrace{\mathbf{1 - e}}_{\text{Probability that choosing one point yields an inlier}}\right)^s\right)^N = p$$

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

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

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

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

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

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

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

$$\underbrace{1 - \left(1 - (1 - e)^{s}\right)^{N}}_{\text{Probability that at least one sample was not contaminated (at least one sample of s points is composed of only inliers).}} = p$$

How many samples?

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

$$\left(1-(1-e)^s\right)^N = 1-p$$

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

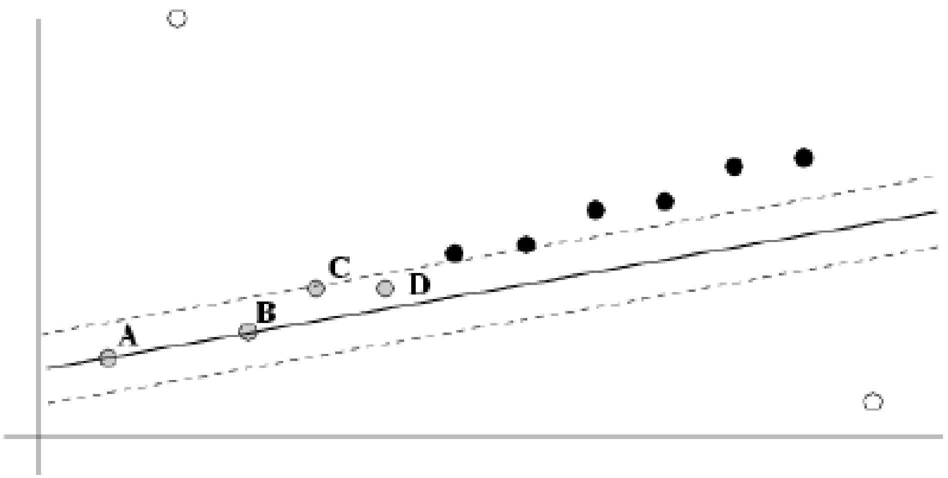
	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

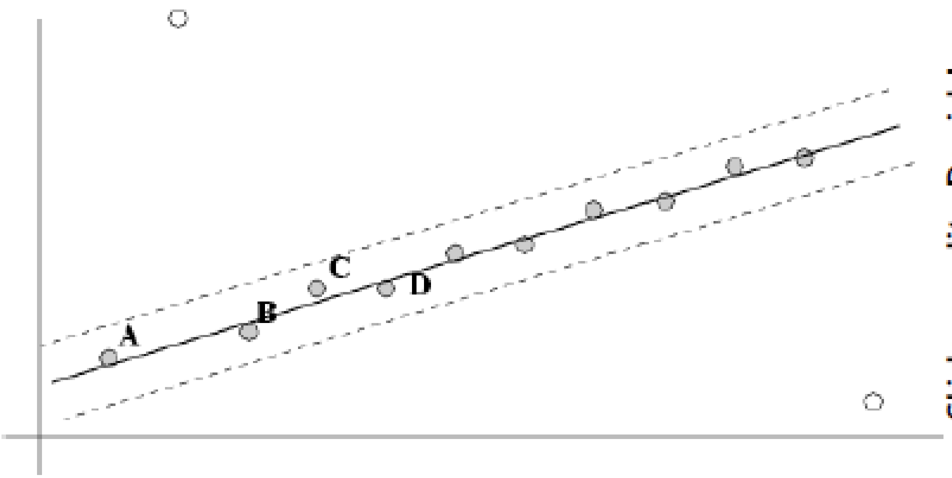
How to choose the parameters?

- Number of sample points **S**
  - Minimum number needed to fit a model
- Outlier ratio **e** (e = #outliers/#datapoints)
- Number of trials **T**
  - Choose T so that, with probability p, at least one random sample is free from outliers
- Distance threshold $delta$
  - Choose $delta$ so that a good point with noise is likely (prob = 0.95) within threshold

After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.





Slide credit: David Lowe

RANSAC : In Summary

Pros

- General method suited for a wide range of model fitting problems
- Easy to implement and easy to calculate its failure rate

Cons

- Computational time grows quickly with fraction of outliers and number of parameters

Common applications

- Computing a homography (e.g., image stitching)
- Estimating fundamental matrix (relating two views)