RANSAC

- = Random Sample Consensus
  - Hypothesize and test.
- Used for Parametric Matching
  - Want to match two things.
  - Hypothesized match can be described by parameters (e.g., translation, affine, ...)
- Match enough features to determine a hypothesis. See if it is good. Repeat.

Parametric Grouping: Grouping Points into Lines

Basic Facts about Lines

\[(x, y)\text{ is on line if } (x, y). (a, b) = c \Rightarrow ax + by = c\]

Distance from \((x, y)\) to line is

\[(a, b) \cdot (x, y) = ax + by \text{ provided } a^2 + b^2 = 1\]
Line Grouping Problem

This is difficult because of:

- Extraneous data: Clutter
- Missing data
- Noise
Precise Definition?

- Find a line that is close to as many points as possible.
  - Close could mean within $\varepsilon$ pixels.
- Find k lines so that every point is close to one of them.
  - Close could mean with $\varepsilon$ pixels.
  - Or, could minimize sum of squares distance from each point to nearest line.

Brute Force Approach

- Try every possibility
  - Every line (infinite)
  - Fit a line to every subset of points (exponential).
- Discrete sampling
  - Could sample slope and offset uniformly.
  - Sample random lines
  - Random lines likely to be good.
RANSAC: Random Sample Consensus

- Generate Lines using Pairs of Points

How many samples?
Suppose $p$ is fraction of points from line.
$n$ points needed to define hypothesis (2 for lines)
$k$ samples chosen.
Probability one sample correct is:

$$1 - (1 - p^n)^k$$

RANSAC for Lines: Continued

- Decide how good a line is:
  - Count number of points within $\varepsilon$ of line.
    - Parameter $\varepsilon$ measures the amount of noise expected.
  - Other possibilities. For example, for these points, also look at how far they are.

- Pick the best line.
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
- If there are $d$ or more points close to the line then there is a good fit. Refit the line using all those points.

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

(Forsyth & Ponce)

RANSAC for Image Matching
Image Matching

- Detect features in each image (e.g., use blob detection).
- Randomly select enough matches to determine a transformation that will align the images.
  - Eg., if we use an affine transformation, we need 3 matching points. Pick three random points in image one, and match each to a random point in image 2. $O(n^6)$ possible matches.
- Apply this transformation to all points in image 1.
- Count number of points that are transformed “near” (say within 2 pixels) of a point in image 2.
- Pick transformation that matches the most points.

Improvements

- Problem: $O(n^6)$ matches is a lot. Only one in $O(n^3)$ will be right.
- Solution: for each point in image 1, use SIFT descriptors to find point in image 2 that provides best match.
- If most of these matches are correct, we now have much higher chance of finding good matches, with a small chance that we miss some.