Robust fitting

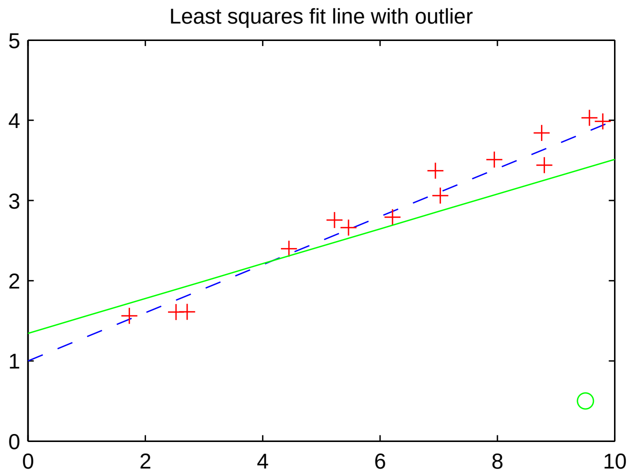
Outliers

In many problems (and especially in vision problems) we have *outliers* in our data. These are points that were incorrectly measured or obtained, and they can have a significant effect on best fit estimates.

Least squares is very sensitive to outliers because the error is *squared*. An outlying point that is far from the bulk of the data therefore has a huge effect on the cost function

$$E_2 = \sum_i e_i^2.$$

Effect of outlier point

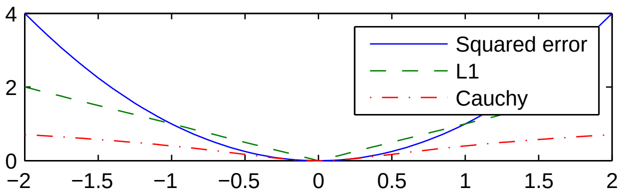


Reweighted least squares

Could limit the effect that a point can have on cost. For example, reweight squared errors by mapping them through a function $\rho(\cdot)$:

$$E_2^\star = \sum_i \rho(e_i).$$

Here $\rho(\cdot)$ is called a *biweight* function. Conventional least squares corresponds to $\rho(\delta) = \delta^2$. However, could use other functions:



Solving IRLS

The problem has now become more difficult, and we can no longer obtain a closed form for the parameter estimates. A common approach is to use *iteratively reweighted least squares*, which searches for a solution using gradient information.

[see any standard text on statistics/operations research]

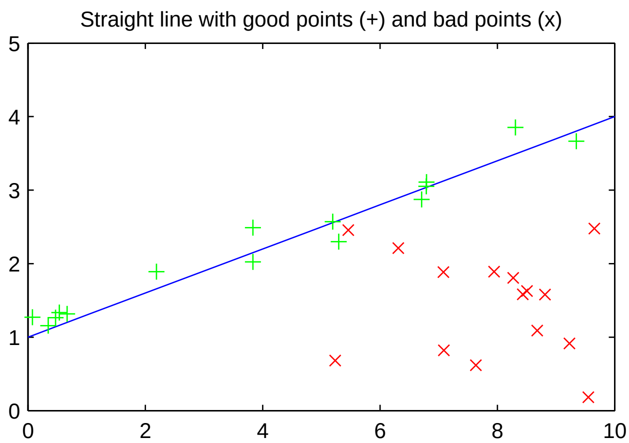
RANSAC

Computer vision researchers typically prefer to use a method called RANSAC (RANdom SAmple Consensus) for addressing this type of problem. This method can deal with a huge number of outliers.

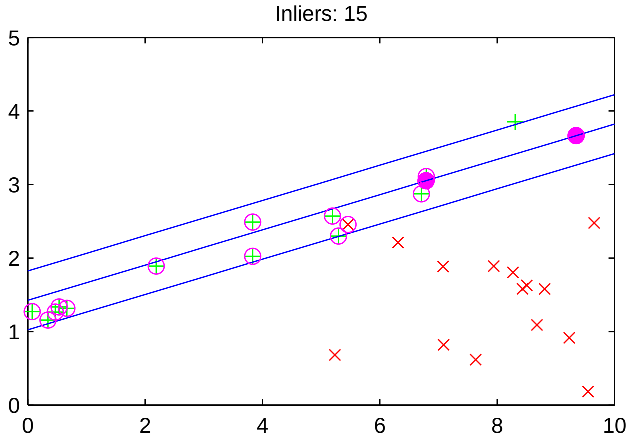
Suppose again that we want to fit a straight line to the data. A line is parameterised by two unknowns, so it can be estimated from just two data points. The RANSAC algorithm proceeds as follows:

1. Select two points randomly from the given data
2. Find the straight line through these points
3. Calculate error between *all* the points and estimated straight line. Count how many points are inliers (errors within an acceptable threshold). Store count and line parameters
4. Return to step 1

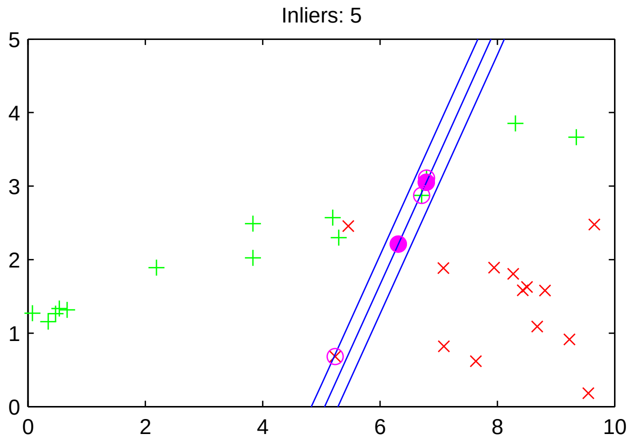
Example: Straight line fitting



Example: Straight line fitting (good random sample)



Example: Straight line fitting (bad random sample)

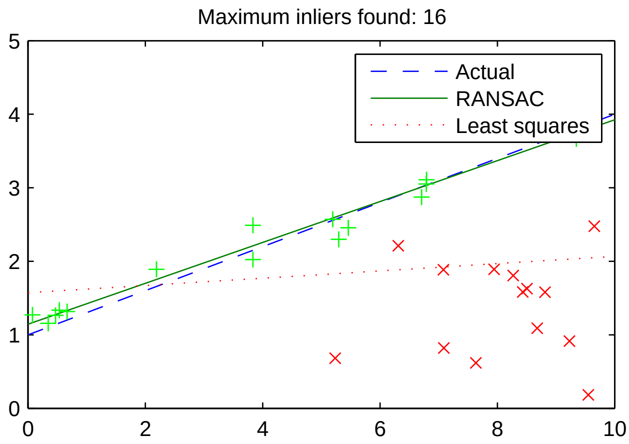


RANSAC solution

After enough iterations, the line with the highest number of inliers is probably a good fit: with enough trials the random choice will eventually give two inliers, the straight line fit will be good, and a large number of points will fall within the acceptable threshold.

How many iterations are required? The answer depends on the fraction of outliers in the data. Suppose 50% of points are outliers (a *huge* number of corrupted points). Randomly choosing two points, the probability of both being inliers is 0.25. The sampling is therefore a Bernoulli trial with a $p = 0.25$ chance of success. Binomial distribution: in n trials expect no successes with probability $(1-p)^n$, so expect at least one success with probability $1-(1-0.25)^n$. For 95% confidence need at least $n = 11$ iterations.

Example: Straight line fitting (RANSAC result)

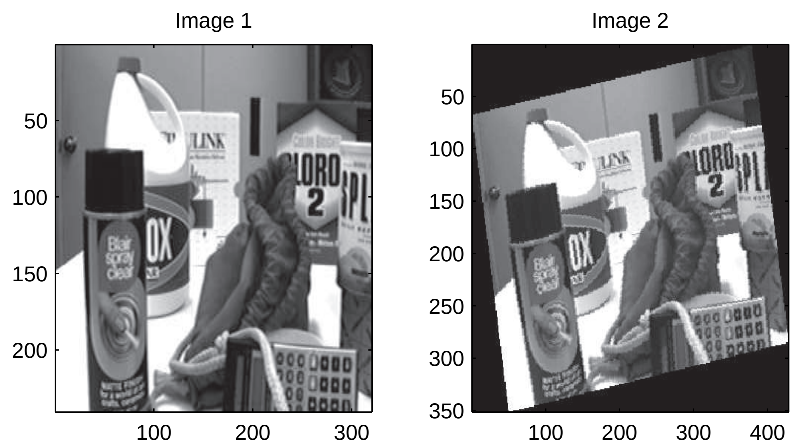


More difficult problems

RANSAC can be used for more complicated problems — only requirement is a parametric form for the relationship being estimated.

However, a larger number of parameters means that more points are required to estimate the transformation, so the number of points in each random sample has to increase. This reduces the probability of any single sample being comprised purely of inliers, so more RANSAC iterations have to be done to ensure a constant probability of success. In practice the process becomes unfeasible for parametric mappings with about 8 unknowns.

Example: Estimating similarity transformation

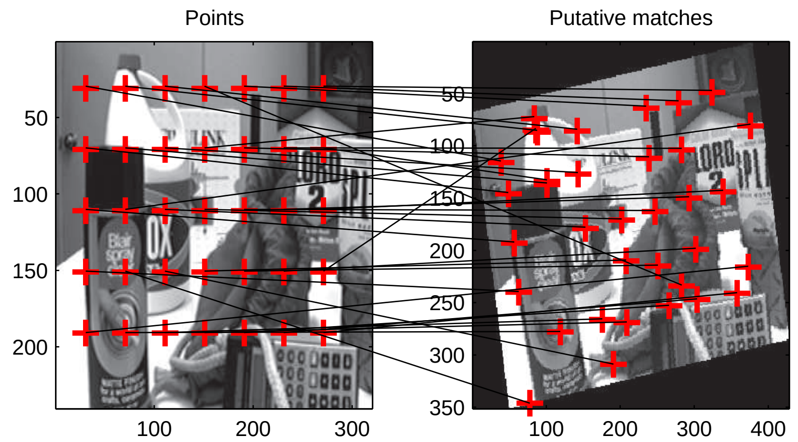


Example: Estimating similarity (formulation)

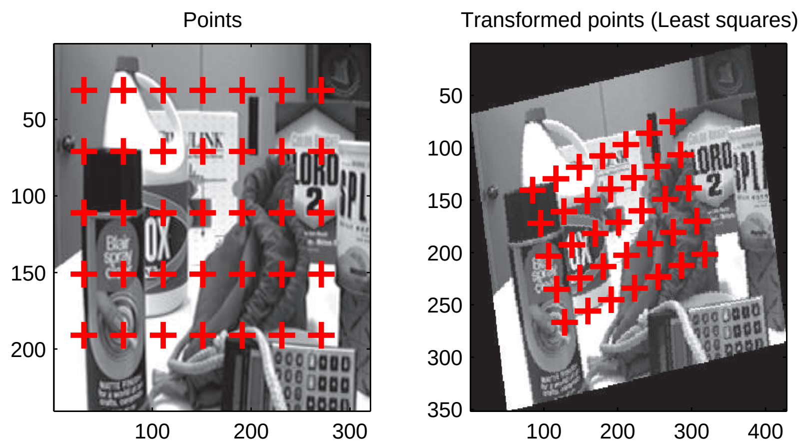
1. Points $\mathbf{x}_1$ in first image and $\mathbf{x}_2$ in second image are related by a similarity transformation (translation, rotation, scaling):
$$\mathbf{x}_2 = s\mathbf{R}\mathbf{x}_1 + \mathbf{t}$$
2. Approach: take region around point in first image and search for best match region in second image. Gives a corresponding point pair $(\mathbf{x}_1, \mathbf{x}_2)$ which should satisfy above equation. Repeat for many points in first image. Some point pairs are outliers from errors in matching

3. Need to estimate four unknown transformation parameters from set of (corrupted) sample pairs

Example: Estimating similarity (matches)



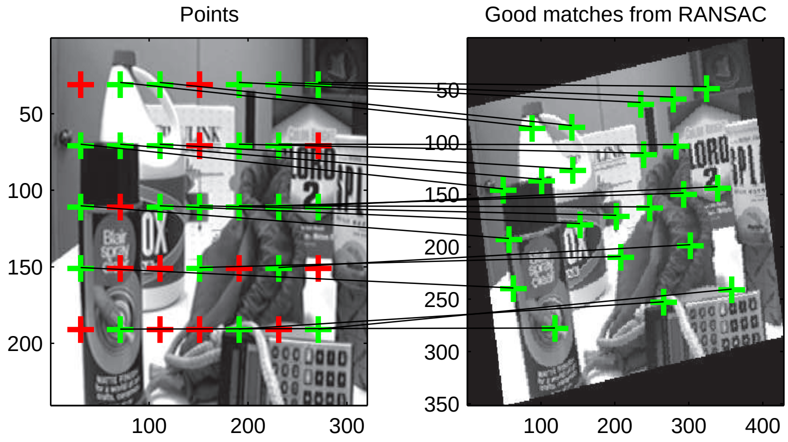
Example: Estimating similarity (Least squares)



Example: Estimating similarity (RANSAC)

1. Four parameters define similarity transformation, which can be estimated from two corresponding point pairs $(\mathbf{x}_1, \mathbf{x}_2)$
2. RANSAC: take random sample of two corresponding point pairs, estimate similarity transformation, map all points from image 1 through the transformation to image 2, and find distances between transformed points and actual points. Count number of points with a small distances, and store with parameters
3. Repeat many times and take solution as instance with highest inlier count

Example: Estimating similarity (RANSAC inliers)



Example: Estimating similarity (RANSAC result)