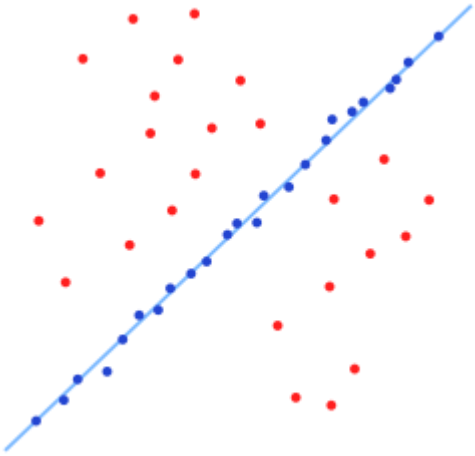
RANSAC - RANdom SAmple Consensus

From Wikipedia

Example

A simple example is fitting of a 2D line to set of observations. Assuming that this set contains both inliers, i.e., points which approximately can be fitted to a line, and outliers, points which cannot be fitted to this line, a simple least squares method for line fitting will in general produce a line with a bad fit to the inliers. The reason is that it is optimally fitted to all points, including the outliers. RANSAC, on the other hand, can produce a model which is only computed from the inliers, provided that the probability of choosing only inliers in the selection of data is sufficiently high. There is no guarantee for this situation, however, and there are a number of algorithm parameters which must be carefully chosen to keep the level of probability reasonably high.

A data set with many outliers for which a line has to be fitted.



Fitted line with RANSAC, outliers have no influence on the result.

Overview

The input to the RANSAC algorithm is a set of observed data values, a parameterized model which can explain or be fitted to the observations, and some confidence parameters.

RANSAC achieves its goal by iteratively selecting a random subset of the original data. These data are *hypothetical inliers* and this hypothesis is then tested as follows:

1. A model is fitted to the hypothetical inliers, i.e. all free parameters of the model are reconstructed from the data set.
2. All other data are then tested against the fitted model and, if a point fits well to the estimated model, also considered as a hypothetical inlier.
3. The estimated model is reasonably good if sufficiently many points have been classified as hypothetical inliers.
4. The model is reestimated from all hypothetical inliers, because it has only been estimated from the initial set of hypothetical inliers.
5. Finally, the model is evaluated by estimating the error of the inliers relative to the model.

This procedure is repeated a fixed number of times, each time producing either a model which is rejected because too few points are classified as inliers or a refined model together with a corresponding error measure. In the latter case, we keep the refined model if its error is lower than the last saved model.

The algorithm

The generic RANSAC algorithm, in pseudocode, works as follows:

```
input:
    data - a set of observations
    model - a model that can be fitted to data
    n - the minimum number of data required to fit the model
    k - the number of iterations performed by the algorithm
    t - a threshold value for determining when a datum fits a model
    d - the number of close data values required to assert that a model fits well
to data
output:
    best_model - model parameters which best fit the data (or nil if no good model
is found)
    best_consensus_set - data point from which this model has been estimated
    best_error - the error of this model relative to the data

iterations := 0
best_model := nil
best_consensus_set := nil
best_error := infinity
while iterations < k
    maybe_inliers := n randomly selected values from data
    maybe_model := model parameters fitted to maybe_inliers
    consensus_set := maybe_inliers

    for every point in data not in maybe_inliers
        if point fits maybe_model with an error smaller than t
            add point to consensus_set

    if the number of elements in consensus_set is > d
        (this implies that we may have found a good model,
        now test how good it is)
        better_model := model parameters fitted to all points in consensus_set
        this_error := a measure of how well better_model fits these points
        if this_error < best_error
            (we have found a model which is better than any of the previous ones,
            keep it until a better one is found)
            best_model := better_model
            best_consensus_set := consensus_set
            best_error := this_error

    increment iterations

return best_model, best_consensus_set, best_error
```