
**RANdom SAmple COnsensus**

Select *one* match, count *inliers*
Repeat many times.
Keep match with largest set of inliers
Basic Philosophy
(voting scheme)

• Elemental subset (minimum number of points) randomly picked up for each hypothesis.

• The standard deviation of the inlier noise has to be given before by the user!

• **Assumption 1:** Outlier features will not vote consistently for any single model.

• **Assumption 2:** There are enough features to agree on a good model.
RANSAC

Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found
RANSAC for line fitting example

Source: R. Raguram
RANSAC for line fitting example

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesize-and-verify loop

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesize-and-verify loop

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesize-and-verify loop

Source: R. Raguram
RANSAC for line fitting example

The best inlier structure

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesize-and-verify loop

Do least-square fit on the inliers.

Source: R. Raguram
RANSAC
RANdom SAmple Consensus

Sample set = set of points in 2D

Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found
Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found
RANSAC

Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found

$$|O| \, = \, 6$$

Sample set = set of points in 2D

standard deviation of the inlier noise has to be given
Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found
An example:

Images:
Image 1
Image 2

Matches:
Red: good matches
Green: bad matches

- By RANSAC fit a homography (later...) mapping features from image 1 to 2.
- Bad matches will be labeled as outliers (hence rejected).
Fitting helps matching.

this is a robust fit
RANSAC conclusions

Good
• Robust to outliers.
• The number of hypotheses N is taken sufficiently large (hundreds to thousands) that RANSAC gives very similar results every time.

Bad
• Computational time grows quickly with fraction of outliers and number of parameters.
• Not good for getting multiple inlier structures.

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