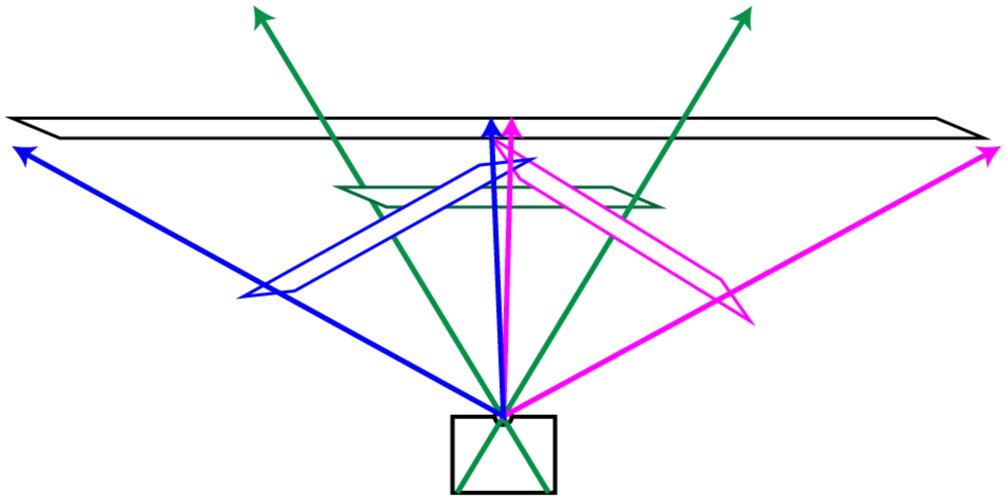
RANSAC Algorithm

AIG Talk 2/22/21

Ryan Levy

Context – Computer Vision

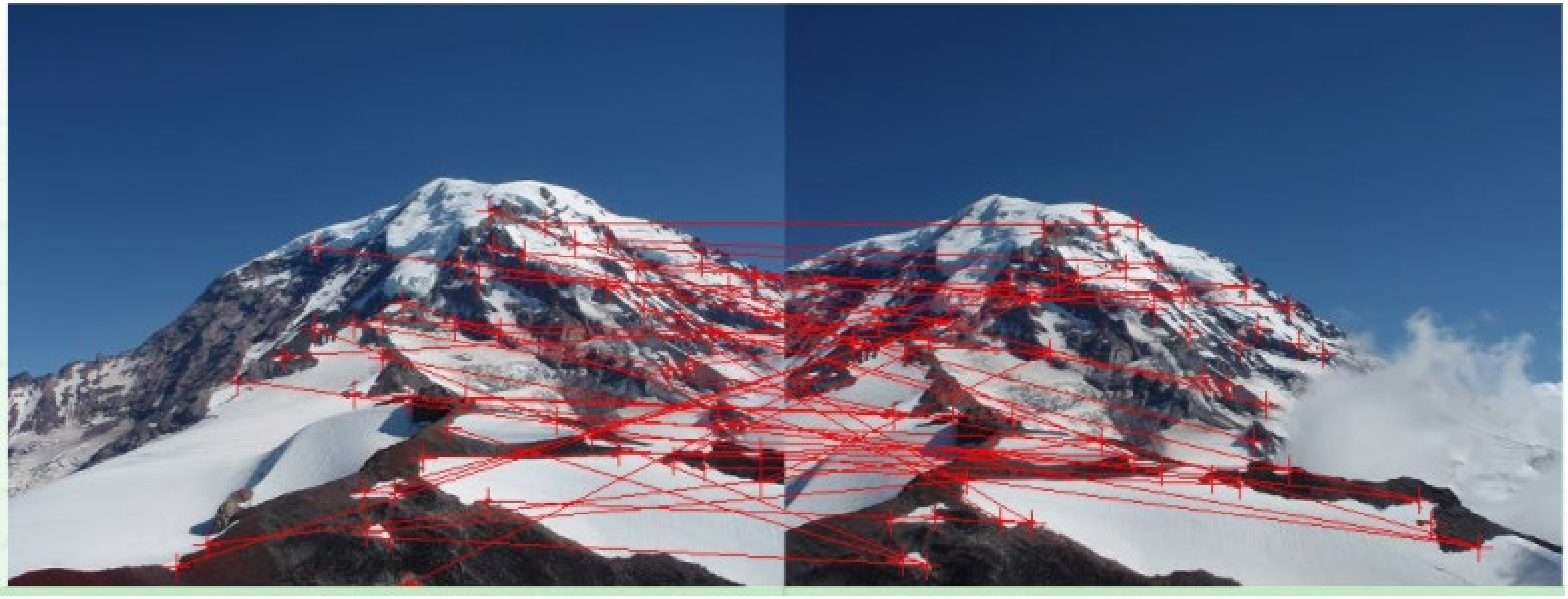
- Often two images will come from the same projection plane but with different transformation
  - Panorama Stitching



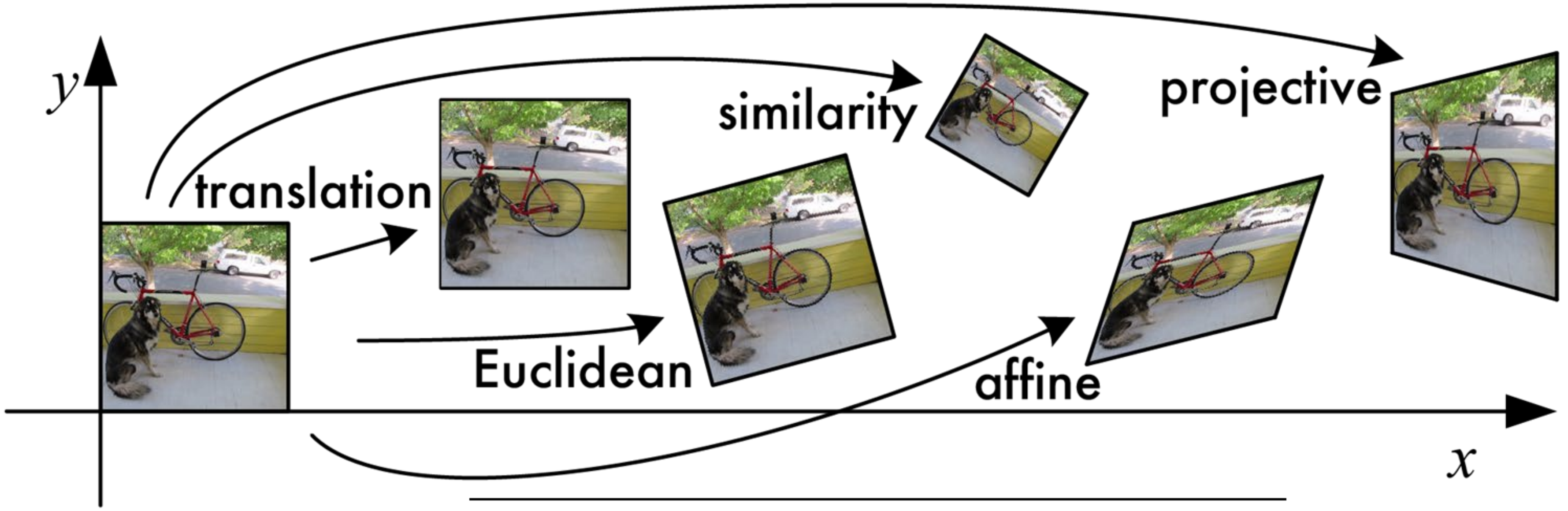
http://dx.doi.org/10.1561/0600000009

Context – Determine Transform

- Use (other) algorithms to perform pattern matching
  - Match ‘unique’ features
- Today’s goal: Match corresponding features without outliers



Context – Transformations



Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\begin{array}{c\|c} \mathbf{I} & \mathbf{t} \end{array}\right]_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \mathbf{R} & \mathbf{t} \end{array}\right]_{2\times 3}$	3	lengths	
similarity	$\left[\begin{array}{c\|c} s\mathbf{R} & \mathbf{t} \end{array}\right]_{2\times 3}$	4	angles	
affine	$\left[\begin{array}{c} \mathbf{A} \end{array}\right]_{2\times 3}$	6	parallelism	
projective	$\left[\begin{array}{c} \tilde{\mathbf{H}} \end{array}\right]_{3\times 3}$	8	straight lines	

$$v = \begin{pmatrix} v_x \\ v_y \\ w \end{pmatrix}$$

Math Background

$$v' = Hv$$

(overdetermined)
Least Squares Problem!

$$S = \sum_i (v_i' - Hv_i)^2$$

$$\frac{dS}{dH_{ij}} = -2v_i^T v_i' + 2v_i^T v_i H_{ij}$$

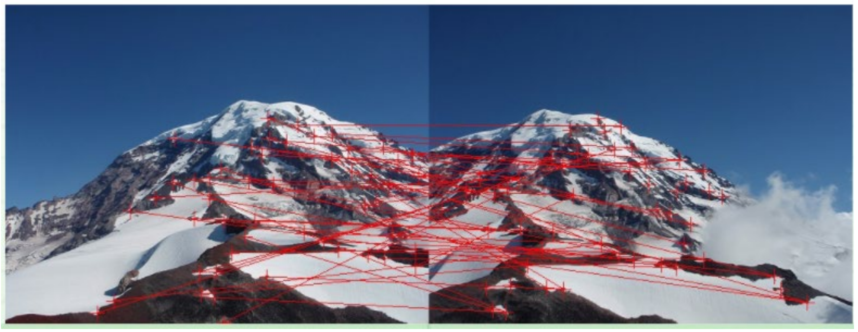
$$\Rightarrow H_{ij} = \left(v_i^T v_i\right)^{-1} v_i^T v_i'$$

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \quad v = \begin{pmatrix} v_x \\ v_y \\ w \end{pmatrix}$$

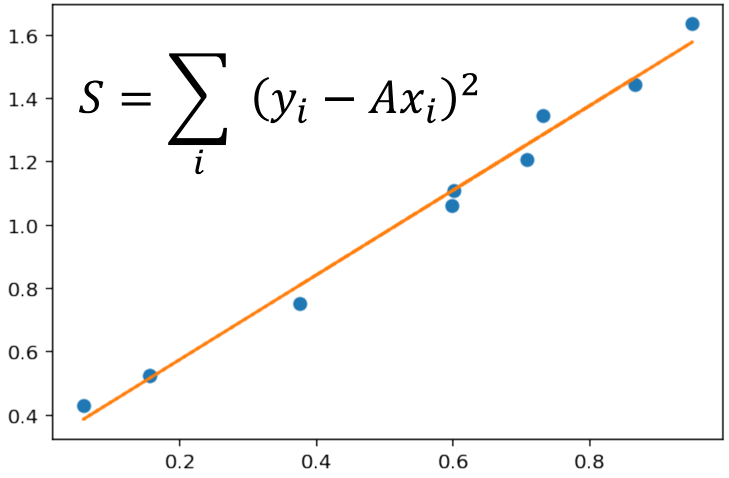
8 free parameters

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{pmatrix}$$

6 free parameters

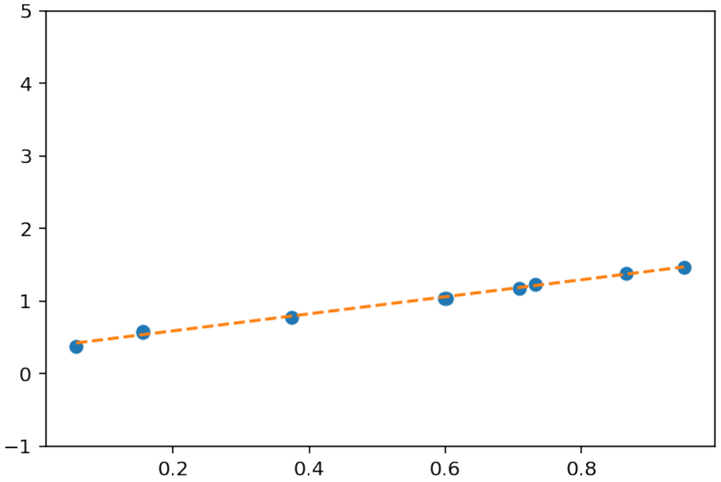


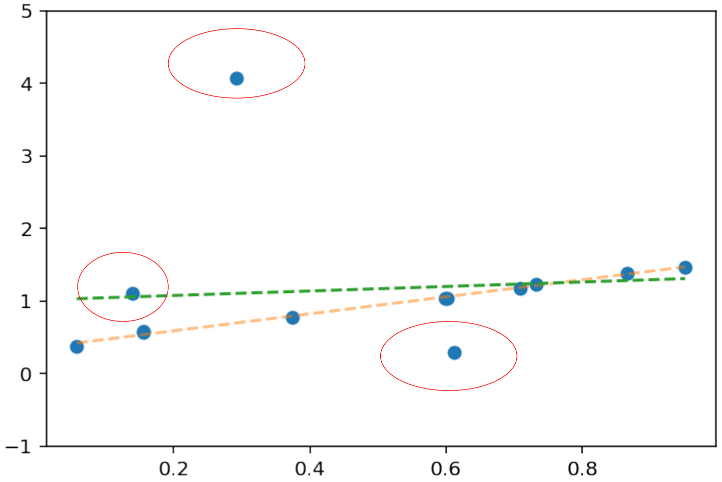
v' $\qquad$ v



Problem - Outliers

- Naïve matching has lots of outliers
- Study RANSAC with linear fits





RANSAC

- Big idea: Try a bunch of fits, the one that is the best is the best!
  - Inliers vs outliers
  - **RAN**dom **SA**mple **C**onsensus

```
for n in range(numTrials):
   pick random set of points
   solve the model using those parameters
   count number of inliers
pick the model with the most inliers
```

Graphics and Image Processing — J. D. Foley, Editor

Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography

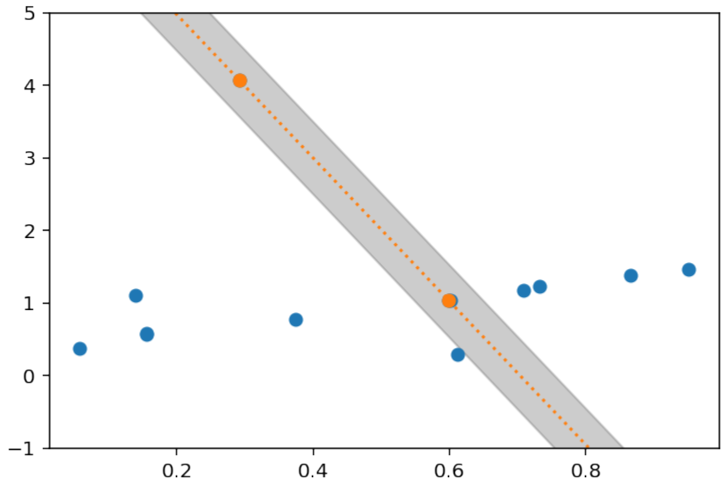
Martin A. Fischler and Robert C. Bolles
SRI International

A new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data is introduced. RANSAC is capable of interpreting/smoothing data containing a significant percentage of gross errors, and is thus ideally suited for applications in automated image analysis where interpretation is based on the data provided by error-prone feature detectors. A major portion of this paper describes the application of RANSAC to the Location Determination Problem (LDP): Given an image depicting a set of landmarks with known locations, determine that point in space from which the image was obtained. In response to a RANSAC requirement, new results are derived on the minimum number of landmarks needed to obtain a solution, and algorithms are presented for computing these minimum-landmark solutions in closed form. These results provide the basis for an automatic system that can solve the LDP under difficult viewing

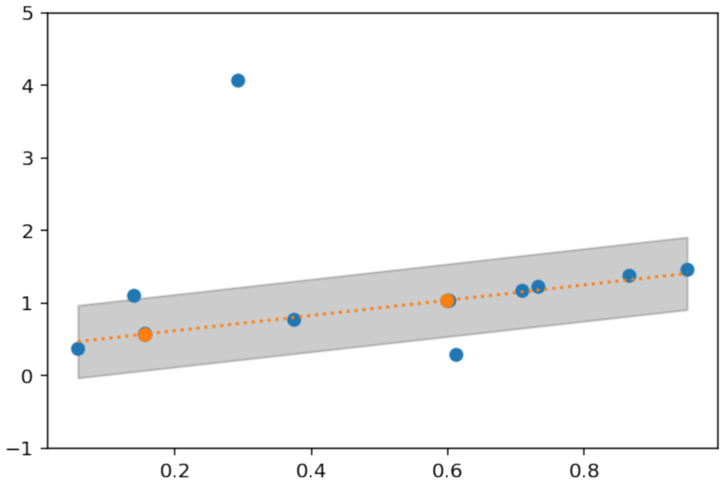
1981 ACM 0001-0782/81/0600-0381 $00.75

RANSAC

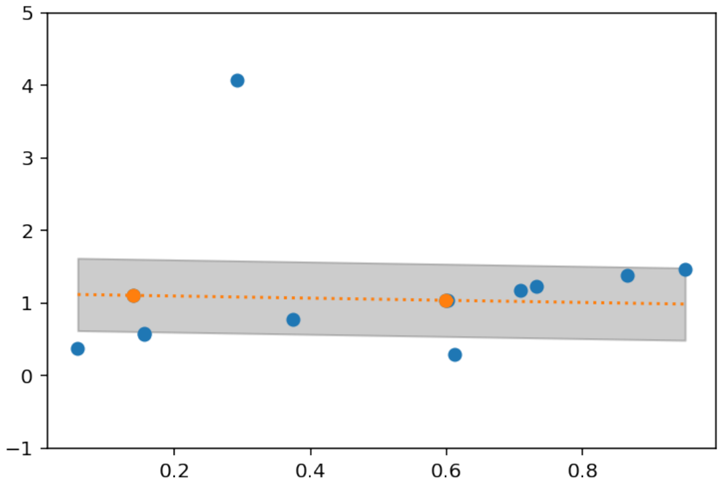
```
for n in range(numTrials):
    pick random set of points
    solve the model using those parameters
    count number of inliers
pick the model with the most inliers
```



1 inliers



8 inliers

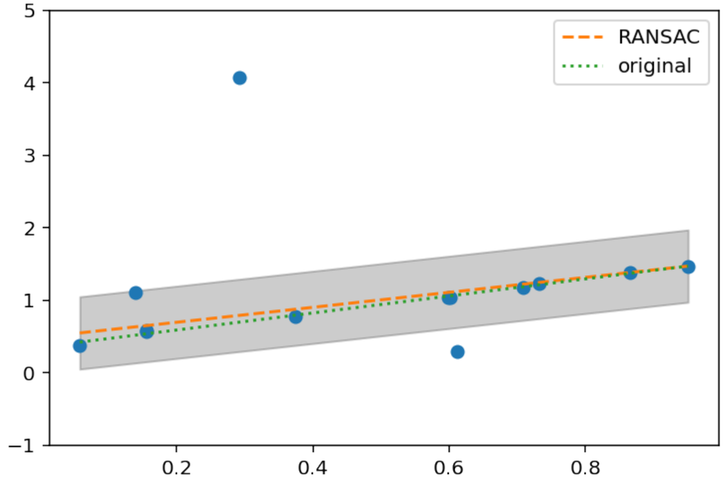


8 inliers

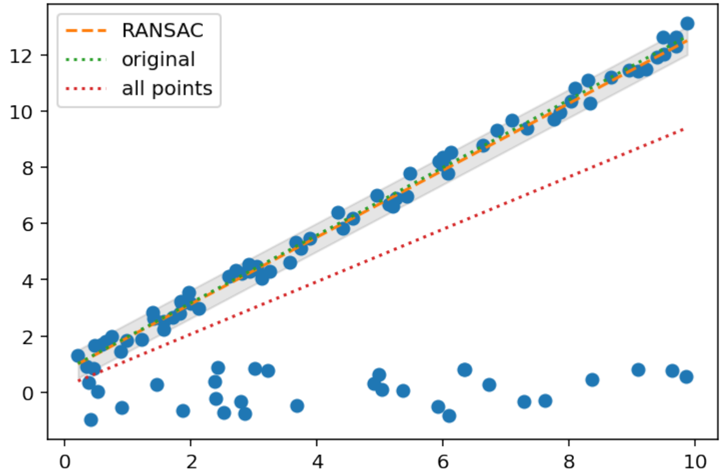
Here total possible trials:

$$\binom{13}{2} = 78$$

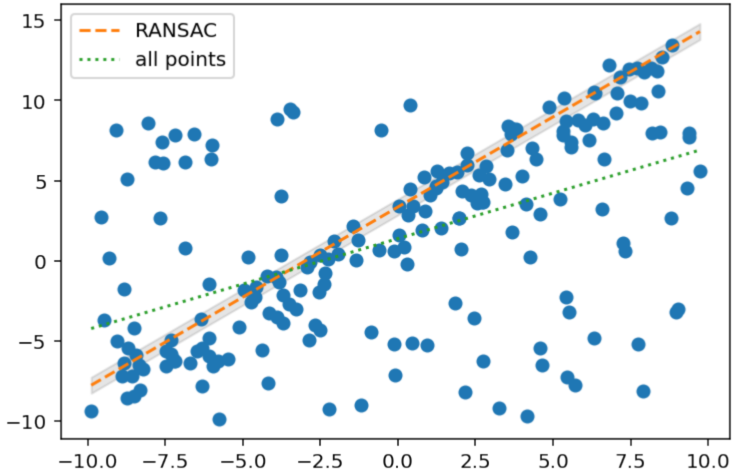
RANSAC Example



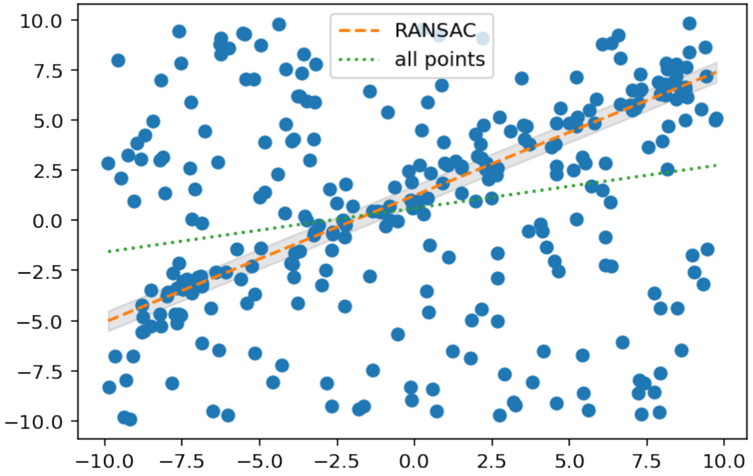

```
nTrials = 100
Delta   = 0.5
N       = 13
N_out   = 3
```




```
nTrials = 100
Delta   = 0.5
N       = 100
N_out   = 30
```




```
nTrials = 100
Delta   = 0.5
N       = 200
N_out   = 100
```




```
nTrials = 100
Delta   = 0.5
N       = 300
N_out   = 200
```

That was too easy?

- Pros
  - Simple/straightforward
  - Very fast
  - Handles large number of outliers
  - # of datapoint insensitive
- Cons
  - KNOBS
    - Inlier/outlier selection
    - (secretly: model choice)
    - Number of trials
      - Computational time(!)

- Trial Selection:
  - $N_{trials} \geq \frac{\log[1-p_s]}{\log[1-(p_g)^s]}$
    - p_s –probability of success (~0.99)
    - p_g - probability of good points (% inliers)
    - s - # parameters
- We had $s = 2$, and at worst $p_g = \frac{1}{3}$ $N_{trials} \geq 39$
- Now imagine $s = 6$ with $p_g = \frac{1}{3}$ $N_{trials} \geq 3355$(!)

$s = 6$

p_g	95%	75%	50%
N_{trials}	4	24	293

Enhancements

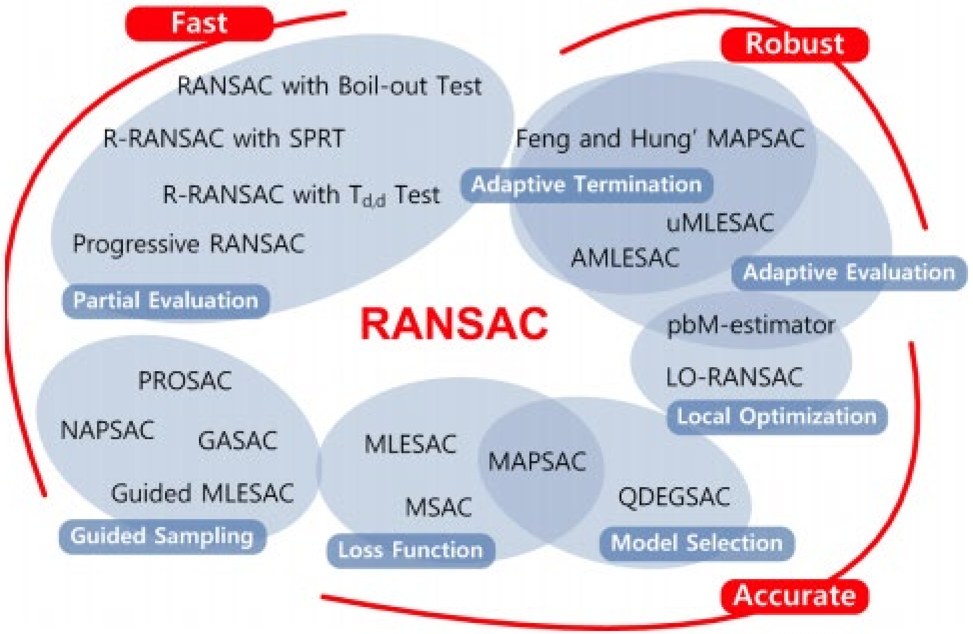


Figure 1: RANSAC Family

Sunglok Choi, et al. Performance Evaluation of RANSAC Family. Proceedings of the British Machine Conference 2009. doi:10.5244/C.23.81.

Guided Sampling

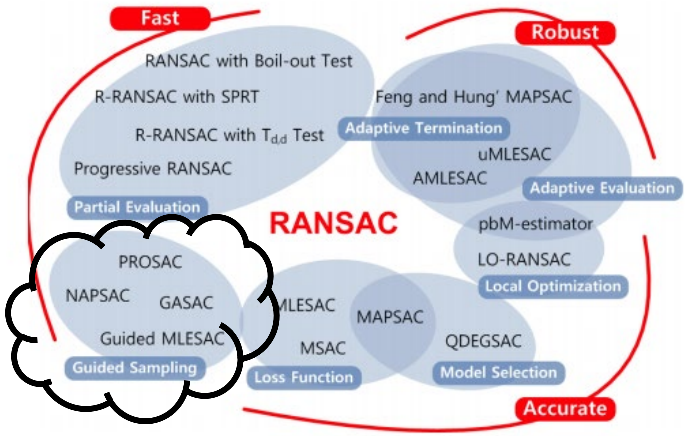


Figure 1: RANSAC Family

- PROSAC
  - Big idea: Group similar points together to get inliers
  - Sort all points by “quality”, always start with high quality points and expand afterword
    - Increases speed

- GASAC
  - Big idea: use a Genetic algorithm to select groups

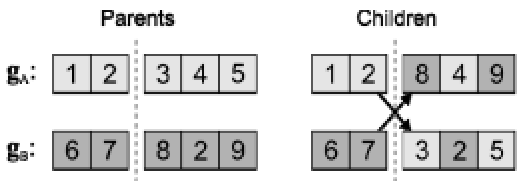


Figure 3: Exchange of chromosome parts using cross-over

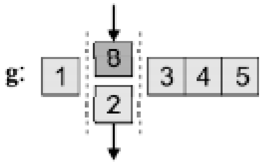


Figure 4: Modification of genes by mutation

Partial Evaluation

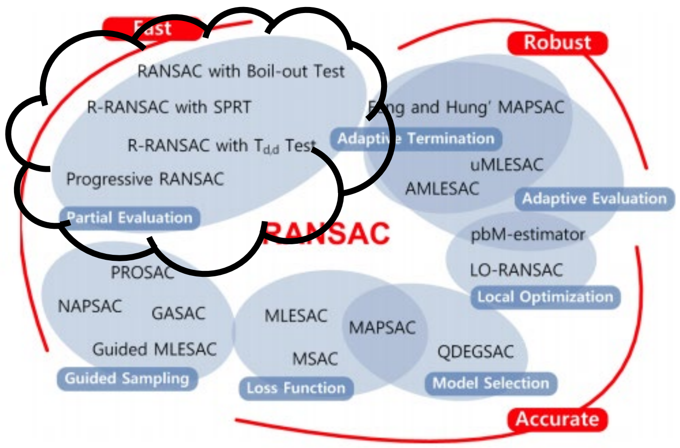


Figure 1: RANSAC Family

- $T_{d,d}$ test
  - Big idea: randomly check $d \ll N$ points to confirm model
  - Randomly select $d \ll N$ points, see if considered inlier to currently model
    - If yes, stop
  - $d = 1$ seems optimal

- Boil-out Test - ?
- Bail-out Test
  - Big idea: Discard models that have too few inliers
  - Fit your model and randomly select points to do an inlier/outlier check
  - Ignore your model if there's too few inlier according to probability distributions

Likelihood Extensions

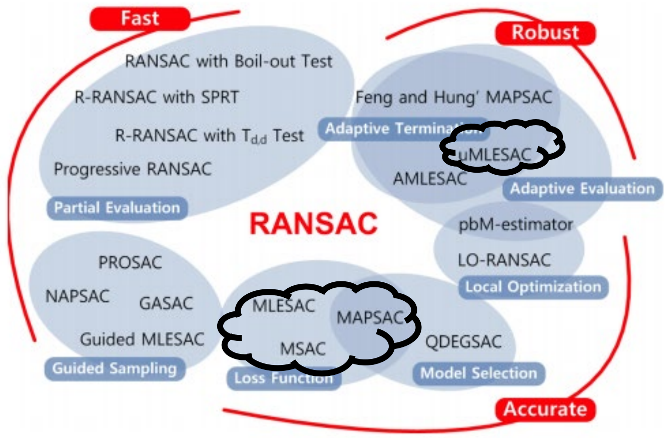
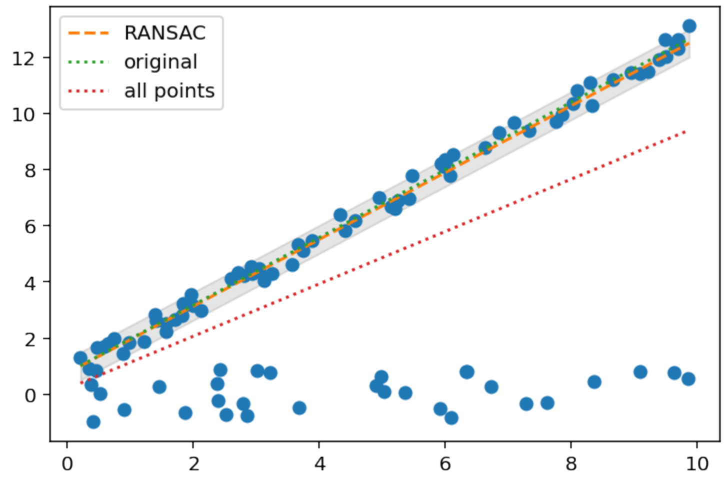


Figure 1: RANSAC Family

- MLESAC
  - Big idea: instead of inliers choose maximum likelihood
- MAPSAC
  - Big idea: MLESAC + Bayesian update, i.e., posterior probability maximization
- uMLESAC
  - Big idea: use ideas from MAPSAC with adapative termination
  - Uses $T_{d,d}$ test

Summary



- RANSAC is an algorithm for solving least squares problems with outliers
  - Robust, fast, and simple algorithm
- Many improvements, focused on speed as well as accuracy

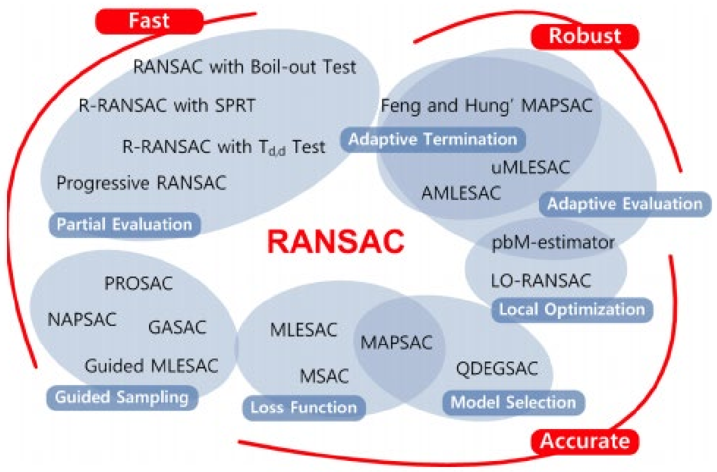


Figure 1: RANSAC Family

References

- Ancient Secrets of Computer Vision
- Richard Szeliski (2007), "Image Alignment and Stitching: A Tutorial", Foundations and Trends® in Computer Graphics and Vision: Vol. 2: No. 1, pp 1-104. http://dx.doi.org/10.1561/0600000009
- A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus

References

- Ancient Secrets of Computer Vision
- Richard Szeliski (2007), "Image Alignment and Stitching: A Tutorial", Foundations and Trends® in Computer Graphics and Vision: Vol. 2: No. 1, pp 1-104. http://dx.doi.org/10.1561/0600000009
- A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus