RANSAC in 2011
(30 years after)

Jiří Matas

Center for Machine Perception,
Department of Cybernetics,
Faculty of Electrical Engineering
Czech Technical University Prague

unless stated otherwise, slide credit goes to Ondra Chum
Robust Model Estimation, Inlier – Outlier Separation

1. Finding Features
2. Matching Features
3. Extracting Epipolar Geometry

INLIERS
good matches

OUTLIERS
mismatches
You all know what RANSAC is, don’t you?
The Acronym “RANSAC”

RANSAC is an independent, non-governmental research organization dedicated to increasing the security of weapons of mass destruction (WMD) and reducing proliferation risks. RANSAC’s priority is supporting the cooperative threat reduction agenda between the U.S., Russia, and the other former Soviet states and promoting its expansion to address global proliferation dangers.
**RANSAC [Fischler, Bolles ’81]**

**In:** $U = \{x_i\}$  
set of data points, $|U| = N$

$f(S) : S \rightarrow p$  
function $f$ computes model parameters $p$ given a sample $S$ from $U$

$\rho(p, x)$  
the cost function for a single data point $x$

**Out:** $p^*$  
$p^*$, parameters of the model maximizing the cost function

$k := 0$

Repeat until $P\{\text{better solution exists}\} < \eta$ (a function of $C^*$ and no. of steps $k$)

$k := k + 1$

I. Hypothesis

(1) select randomly set $S_k \subset U$, sample size $|S_k| = m$

(2) compute parameters $p_k = f(S_k)$

II. Verification

(3) compute cost $C_k = \sum_{x \in U} \rho(p_k, x)$

(4) if $C^* < C_k$ then $C^* := C_k$, $p^* := p_k$

end
RANSAC

- Select sample of $m$ points at random
RANSAC

• Select sample of \( m \) points at random

• Calculate model parameters that fit the data in the sample
RANSAC

- Select sample of m points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
RANSAC

- Select sample of $m$ points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- Select data that support current hypothesis
RANSAC

- Select sample of m points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- Select data that support current hypothesis
- Repeat sampling
RANSAC

1. Select sample of m points at random
2. Calculate model parameters that fit the data in the sample
3. Calculate error function for each data point
4. Select data that support current hypothesis
5. Repeat sampling
RANSAC

**ALL-INLIER SAMPLE**

RANSAC time complexity

\[ t = k(t_M + \bar{m}_S N) \]

- \( k \) ... number of samples drawn
- \( N \) ... number of data points
- \( t_M \) ... time to compute a single model
- \( m_S \) ... average number of models per sample
the “gold standard” algorithm:

**In:** \( U = \{x_i\} \)  
set of data points, \(|U| = N\)

\( f(S) : S \rightarrow p \)  
function \( f \) computes model parameters \( p \) given a sample \( S \) from \( U \)

\( \rho(p, x) \)  
the cost function for a single data point \( x \)

**Out:** \( p^* \)  
\( p^* \), parameters of the model maximizing the cost function

\( k := 0 \)

Repeat until \( P\{\text{better solution exists}\} < \eta \) (a function of \( C^* \) and no. of steps \( k \))

\( k := k + 1 \)

I. Hypothesis

(1) select randomly set \( S_k \subset U \), sample size \(|S_k| = m\)

(2) compute parameters \( p_k = f(S_k) \)

II. Verification

(3) compute cost

(4) if \( C^* < C_k \) then \( C^* := C_k, p^* := p_k \)

end Repeat

\( p^{\text{out}} = \) least square fit on the set of inliers to \( p^* \)
RANSAC: case closed?

- The algorithm is extremely simple
- Can we stop the talk here?
RANSAC: case closed?

- The algorithm is extremely simple, we stop the talk here?

  NO!

- The ("gold") standard algorithm often:
  - does not produce the correct model with the user-defined probability
  - outputs an inaccurate model
  - does not handle degeneracies
  - can be speeded up (by orders of magnitude)
  - does not guarantee maximum running time
  - needs information about the scale of the noise (this issue will not be discussed)
  - multiple models .....
RANSAC: what is (not) covered

Covered:
most practically important developments, (most cited)

Not covered:
- scale selection
- maximum-likelihood link (MLESAC, MSAC)
- methods guaranteeing optimal results:

Hongdong Li: Consensus set maximization with guaranteed global optimality for robust geometry estimation. ICCV 2009:

F Kahl - Computer Vision–ECCV 2008, 2008 - Robust Optimal Pose Estimation
RANSAC: selected papers before 2006


RANSAC since 2006 (variants)

A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus

R Raguram, JM Frahm... - Computer Vision–ECCV 2008, 2008 - Springer
Abstract. The Random Sample Consensus (RANSAC) algorithm is a popular tool for robust estimation problems in computer vision, primarily due to its ability to tolerate a tremendous fraction of outliers. There have been a number of recent efforts that aim to increase the efficiency of...
Cited by 40 - Related articles - All 16 versions

Optimal randomized RANSAC
O Chum... - IEEE transactions on pattern analysis and ..., 2008 - computer.org
Abstract—A randomized model verification strategy for ransac is presented. The proposed method finds, like ransac, a solution that is optimal with user-specified probability. The solution is found in time that is close to the shortest possible and superior to any deterministic verification ...
Cited by 38 - Related articles - All 6 versions

Improved RANSAC performance using simple, iterative minimal-set solvers
Abstract RANSAC is a popular technique for estimating model parameters in the presence of outliers. The best speed is achieved when the minimum possible number of points is used to estimate hypotheses for the model. Many useful problems can be represented using ...
Cited by 1 - Related articles - All 3 versions

Optimal Randomized RANSAC
O rej Chum... - IEEE TRANSACTIONS ON PATTERN ..., 2008 - cmp.felk.cvut.cz
Abstract—A randomized model verification strategy for RANSAC is presented. The proposed method finds, like RANSAC, a solution that is optimal with user-specified probability. The solution is found in time that is close to the shortest possible and superior to any deterministic ...
Related articles - View as HTML

New conditional sampling strategies for speeded-up RANSAC
Virtual mitochondrion: towards an integrated model of oxidative phosphorylation complexes and beyond.

... M Heiske, C Nazaret, S Ransac - Biochemical Society ..., 2010 - ncbi.nlm.nih.gov
The modelling of OXPHOS (oxidative phosphorylation) in order to integrate all kinetic and thermodynamic aspects of chemiosmotic theory has a long history. We briefly review this history and show how new ways of modelling are required to integrate a local model of the individual respiratory ...

All 5 versions

The fitting of electrons in complex I: A stochastic approach
S Ransac, C Arnarez... - Biochimica et Biophysica Acta (BBA)- ..., 2010 - Elsevier
A stochastic approach based on the Gillespie algorithm is particularly well adapted to describe the time course of the redox reactions that occur inside the respiratory chain complexes because they involve the motion of single electrons between the individual unique redox centres of ...

Related articles - All 3 versions

How does antimycin inhibit the bc1 complex? A part-time twin
S Ransac... - Biochimica et Biophysica Acta (BBA)-Bioenergetics, 2010 - Elsevier
Using a stochastic simulation without any other hypotheses, we recently demonstrated the natural emergence of the modified Mitchell Q-cycle in the functioning of the bc1 complex, with few short-circuits and a very low residence time of the reactive semiquinone species in the Q ...

Related articles - All 3 versions

[CITATION] A stochastic approach of the electron transport in the mitochondrial respiratory chain
..., S Ransac - Biochimica et Biophysica Acta (BBA)-Bioenergetics, 2010 - Elsevier
Related articles

[CITATION] S15. 14 Stochastic approach of bc1 complex functioning
S Ransac... - Biochimica et Biophysica Acta (BBA)-Bioenergetics, 2008 - Elsevier
Related articles

[Structural basis of phospholipase activity of Staphylococcus hyicus lipase
..., G Poudroyen, M Nardini, S Ransac... - Journal of molecular ..., 2007 - Elsevier
Staphylococcus hyicus lipase differs from other bacterial lipases in its high phospholipase A1 activity. Here, we present the crystal structure of the S. hyicus lipase at 2.86 Å resolution. The lipase is in an open conformation, with the active site partly covered by a neighbouring ...
Locally optimised RANSAC

Chum, Matas, Kittler:
   Locally Optimized RANSAC, *DAGM*, 2003
Chum, Matas, Obdržálek:
   Enhancing RANSAC by Generalized Model Optimization, *ACCV*, 2004
Locally Optimized RANSAC

It was observed experimentally, that RANSAC takes several times longer than theoretically expected. This is due to the noise on inlier measurement – not every all-inlier sample generates a good hypothesis.

By applying local optimization (LO) to the-best-so-far hypotheses:
(i) a better agreement with theoretical (i.e. optimal) performance
(ii) lower sensitivity to noise and poor conditioning.

Note: the claim in Chum et al. 2003: The LO is shown to be executed so rarely that it has minimal impact on the execution time. Not always true!
The Stopping Criterion ...

- **Observation:** Often, the solution is found later than predicted, i.e. after on average $k$ steps

$$k = \left( \frac{1}{\varepsilon} \right)^m$$

where $\varepsilon$ is the fraction of inliers, $m$ the sample size

- **RANSAC stopping criterion** makes a (hidden) assumption that *any* all-inlier sample leads to the solution.
.... Makes an Invalid Assumption

Not every all-inlier sample gives a model consistent with all inliers

Lower number of inliers is detected

RANSAC runs longer
<table>
<thead>
<tr>
<th>Image</th>
<th>Qty1</th>
<th>std</th>
<th>std</th>
<th>std</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>box</td>
<td>1</td>
<td>28.1±2.2 (24-33)</td>
<td>8.1±11.0 (0-32)</td>
<td>28.3±2.2 (24-33)</td>
<td>28.6±2.1 (24-33)</td>
</tr>
<tr>
<td></td>
<td>Samp</td>
<td>65.4±3.2 (56-77)</td>
<td>18.8±25.5 (0-74)</td>
<td>65.9±5.1 (56-77)</td>
<td>66.6±4.8 (56-77)</td>
</tr>
<tr>
<td></td>
<td>Time (ms)</td>
<td>1.0</td>
<td>1.2</td>
<td>3.9</td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td>Error</td>
<td>2.13±2.85 (0.3-20.4)</td>
<td>15.09±8.46 (0.7-70.7)</td>
<td>2.54±4.59 (0.3-81.5)</td>
<td>2.27±3.86 (0.3-54.3)</td>
</tr>
<tr>
<td></td>
<td>LO count</td>
<td>0.0±0.0 (0)</td>
<td>0.0±0.0 (0)</td>
<td>0.0±0.0 (0)</td>
<td>1.3±0.6 (1-4)</td>
</tr>
</tbody>
</table>

Credit: Karel Lebeda
Solution: Local Optimisation Step

Repeat $k$ times
1. Hypothesis generation
2. Model verification
   2b. If model best-so-far Execute (Local) Optimisation

**Inner RANSAC + Re-weighted least squares:**
- Samples are drawn from the set of data points consistent with the best-so-far hypothesis
- New models are verified on all data points
- Samples can contain more than minimal number of data points since consistent points include almost entirely inliers

\[
\sum_{l=1}^{k} P_l = \sum_{l=1}^{k} \frac{1}{l} \leq \int_{1}^{k} \frac{1}{x} dx + 1 = \log k + 1
\]

**How often?**

**Conclusion:** the LO step 2b is executed rarely, does not influence running time significantly
Validation: Two-view Geometry Estimation

Histograms of the number of inliers returned over 100 executions of RANSAC (top) and LO-RANSAC (bottom)

Result:
(i) variation of the number of inliers significantly reduced
(ii) speed-up up to 3 times (for 7pt EG and 4pt homography est.)
A model from a random sample need not be required to be the solution. Any model that is consistent with a large proportion of inliers moves us closer to a solution, since its inliers can be sampled.

Idea:

1. Estimate approximate models with lower complexity (less data points in the sample) with loose thresholds

2. Followed by LO step estimating full model
Estimation via Approximate Models

Epipolar geometry and radial distortion

Standard:
9 point correspondences define the model

LO Ransac:
1. Approximated by EG with no radial distortion
2. In LO step (2b), estimate the full model from 9 (or more!) points

\[ p = \frac{1}{1 + \lambda|x|^2} \]

Division model
Fitzgibbon CVPR’01

EG from 3 local affine frame (LAF) correspondences
- each region provides 3 points
- 3 LAFs determine EG
- points close to each other (low precision)

Chum, Matas, Obdržálek: Enhancing RANSAC by Generalized Model Optimization, ACCV 2004
Radial Distortion I

Distribution of inliers over 100 runs

A: 7pt LO-RANSAC

B: 9pt RANSAC

C: LO-RANSAC-RD

Distribution of $\lambda$ parameter

The number of samples

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5528</td>
<td>31456</td>
<td>790</td>
</tr>
</tbody>
</table>
Radial Distortion II

Distribution of inliers over 100 runs

A: 7pt LO-RANSAC
B: 9pt RANSAC
C: LO-RANSAC-RD

Distribution of $\lambda$ parameter

Courtyard QY (48% of inliers)

The number of samples

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1861</td>
<td>6863</td>
<td>432</td>
</tr>
</tbody>
</table>
EG from Three Correspondences

518 tentative correspondences, 7% of inliers

<table>
<thead>
<tr>
<th>Method</th>
<th>EG consistent</th>
<th>iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>7pt LO-RANSAC</td>
<td>N/A</td>
<td>≈ 684,000,000</td>
</tr>
<tr>
<td>3LAF RANSAC</td>
<td>25</td>
<td>47,668</td>
</tr>
<tr>
<td>3LAF LO-RANSAC</td>
<td>36</td>
<td>14,880</td>
</tr>
</tbody>
</table>

More than 10,000-fold speed-up
SUMMARY

- Not every all-inlier sample generates model consistent with all inliers due to noise

- Local data driven search initialised by the best-so-far solution saves computational effort

- Using hierarchical estimation (starting with approximate models) speeds up the estimation
LO RANSAC algorithm

It must be possible to simplify this and put it on sound footing!
Degenerate Configurations

Chum, Werner, Matas: EG Estimation unaffected by dominant plane, *CVPR'05*
Degenerate Configurations

The presence of degenerate configuration causes RANSAC to fail in estimating a correct model, instead a model consistent with the degenerate configuration and some outliers is found.

The DEGENSAC algorithm handles scenes with:

- all points in a single plane
- majority of the points in a single plane and the rest off the plane
- no dominant plane present

No a-priori knowledge of the type of the scene is required
Degenerate Configurations

- Infinite number of models passes through a degenerate configuration
- Sample with points from both the degenerate configuration and outlier(s) is a problem – it has higher than random support
Toy Example

Robust fitting of a plane to 3D points

Algorithm:
- Draw samples of three points & fit a plane to them
- Calculate the support of the plane

Problem:
- High number of points lie on a line
- A sample containing two points from the line and one outlier has large support (the whole line)
- Such sample is not degenerate, the problem does not show as a singularity in the model estimation step
- Algorithm may return incorrect plane (that contains the line)
Real Example

The Dominant Plane Problem:
Given many (almost) coplanar points, a few points off the plane, and (possibly) many outliers estimate the epipolar geometry (EG), if possible.

RANSAC run with 95% confidence actually finds the inliers on the lamppost in only 17% of executions.
In the Presence of a Dominant Plane

- RANSAC draws minimal samples of 7 correspondences to hypothesize the epipolar geometry
- When dominant plane is present, samples with more than 4 coplanar correspondences often appear
- 7 or 6 coplanar correspondences: the sample is consistent with a family of fundamental matrices. This case is easily be detected.
- Chum et al. CVPR’05 show that the epipolar geometry hypothesised from 5 coplanar points from the dominant plane and 2 off the plane (a so called H-degenerate sample) has a large RANSAC support.

If at least one of the off-plane correspondences is an outlier, the EG is incorrect.
Different Solutions Returned by RANSAC

(a) 643 inliers ($\varepsilon = 0.67$)
(b) 621 inliers ($\varepsilon = 0.65$)
(c) 615 inliers ($\varepsilon = 0.64$)
(d) 614 inliers ($\varepsilon = 0.64$)
(e) 616 inliers ($\varepsilon = 0.64$)
(f) 615 inliers ($\varepsilon = 0.64$)
Core of the algorithm:

1. Draw samples of 7 correspondences and estimate 1-3 fundamental matrices by the 7-point algorithm
2. Test samples with the largest support so far for H-degeneracy
3. When **H-degeneracy** was detected, use plane-and-parallax algorithm [Irani&Anadan ECCV’97].

Note: the plane-and-parallax needs to draw samples of only 2 correspondences to hypothesize EG, therefore its complexity is negligible compared to RANSAC where 7 correspondences are drawn into a sample.

Alternative to step 2: execute RANSAC detecting degenerate configuration on inliers:

**Frahm, Pollefeys:** RANSAC for (Quasi-)Degenerate Data (QDEGSAC), *CVPR’06*

(and later improvements and generalization)
Stability of the Results

How many times will a correspondence be labeled as an inlier, if we run the experiment 100 times?

- **RANSAC**
- **DEGENSAC**

![Off the plane correspondences](image)

Correspondence number
PROSAC -
Progressive Sampling and Consensus

Progressive Sampling Consensus

Instead of drawing samples randomly, samples containing tentative correspondences are drawn preferentially. The sampling procedure gradually progresses towards uniform sampling of standard RANSAC.

RANSAC tested 106,534 samples, PROSAC tested only 9 samples.
Idea: exploit the fact that all correspondences are not created equal.

In all (?) multi-view problems, the correspondences are from a totally ordered set (they are selected by thresholding of a similarity function)

Challenge:
• organize computation in a way that exploits ordering, and yet is equally robust as RANSAC

PROSAC properties:
- selection of correspondences and RANSAC integrated (no need for threshold on local similarity).
- Same guarantees about the solution as RANSAC
- Depending on the data, can be orders of magnitude faster
The PROSAC Algorithm

Idea:

1. Generate the maximal number of $T_N$ RANSAC samples.
2. Evaluate the samples in order of decreasing quality of the samples. The quality of the sample is given by the lowest quality of correspondence in the sample.

An efficient way (without generating all $T_N$ samples beforehand) how to generate ordered samples of an ‘average’ RANSAC follows. RANSAC, on average, draws

$$T_n = T_N \frac{n!}{m!} = T_N \prod_{i=0}^{m-1} \frac{n-i}{N-i}$$

samples of size $m$ that contain only data points from the set of $n$ top quality data points $\mathcal{U}_n$. There are $T_n$ samples containing only data points from $\mathcal{U}_n$ and $T_{n+1} = \frac{n+1}{n+1-m} T_n$ samples containing only data points from $\mathcal{U}_{n+1}$. Since $\mathcal{U}_{n+1} = \mathcal{U}_n \cup \{ u_{n+1} \}$, there are $T_{n+1} - T_n$ samples that contain a data point $u_{n+1}$ and $m - 1$ data points drawn from $\mathcal{U}_n$.

Efficient algorithm:

```plaintext
for n = m \ldots N
  Generate \lfloor T_{n+1} - T_n \rfloor samples consisting of a data point $u_{n+1}$ and $m - 1$ data points drawn from $\mathcal{U}_n$ at random.
```
EG estimation experiment

- The fraction of inliers in top $n$ correspondences:

<table>
<thead>
<tr>
<th>Method</th>
<th>$k$</th>
<th>min $k$</th>
<th>max $k$</th>
<th>time [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROSAC</td>
<td>9</td>
<td>5</td>
<td>29</td>
<td>0.06</td>
</tr>
<tr>
<td>RANSAC</td>
<td>106,534</td>
<td>97,702</td>
<td>126,069</td>
<td>10.76</td>
</tr>
</tbody>
</table>
## Multiple motion experiment

<table>
<thead>
<tr>
<th></th>
<th>Background</th>
<th>Mug</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>783</td>
<td>166</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>79%</td>
<td>31%</td>
</tr>
<tr>
<td>$I$</td>
<td>617</td>
<td>51.6</td>
</tr>
<tr>
<td>$k$</td>
<td>1.0</td>
<td>18</td>
</tr>
<tr>
<td>time [sec]</td>
<td>0.33</td>
<td>0.12</td>
</tr>
<tr>
<td>PROSAC</td>
<td>15</td>
<td>10,551</td>
</tr>
<tr>
<td>RANSAC</td>
<td>1.10</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Note: What is an inlier does not depend on the similarity of appearance!
PROSAC: Conclusions

- In many cases, PROSAC draws only a few samples.
- The information needed to switch from RANSAC to PROSAC - ordering of features - is typically available.
- Note that the set of data points that is sampled to hypothesise models and the set used in verification need not be the same.
- RANSAC needs protection against degenerate configurations and Local Optimisation (effectively we ran LO-DEGEN-PROSAC). This is important, since often top correspondences lie on the same plane.
Preemptive RANSAC

D. Nistér:
Preemptive RANSAC for live structure and motion estimation.

ICCV 2003
RANSAC Hypothesis Evaluation

Hypothesis Generator

Observation Likelihood

Hypotheses

Observations

1000

500 x 1000 = 500,000
Preemptive RANSAC  

[Nister 2005]  

Depth-first Preemption  

---  

Hypotheses  

Observations  

500  

1000  

500 \times \text{????} = \text{????????}
Breadth-first Preemption

Chunks size 100

Observations 1000

Hypotheses 500

$500 \times 200 = 100,000$

Overhead $\sim 100$ microseconds

[Preemptive RANSAC] [Nister 2005]

slide credit: David Nister
Preemptive RANSAC

Hypothesis Generation

Observed Tracks

slide credit: David Nister
WaldSAC –
Optimal Randomised RANSAC

Chum, Matas: Randomized RANSAC with Td,d test,

Matas, Chum: Optimal Randomised RANSAC,
International Conference on Computer Vision, Beijing, 2005

IEEE Trans. PAMI, 2008, 8, pages 1472-1482
Randomized RANSAC

Instead of verifying all data points in the verification phase, a statistical test is introduced. Only promising hypotheses are fully verified.

RANSAC time complexity \[ t = k(t_M + \bar{m}sN) \]

R-RANSAC time complexity \[ t = \frac{k}{1 - \alpha}(t_M + \bar{m}sV) \]

\[ \alpha \ll 1 \quad \text{and} \quad V \ll N \]

RR with T(d,d) test – simple, mathematically tractable test

RR with SPRT – optimal test (minimises the total running time \( t \)) based on Wald’s Sequential Analysis [1947]

Chum, Matas: Randomized RANSAC with T(d,d) test, BMVC 2002
Matas, Chum: Randomized RANSAC with SPRT, ICCV 2005 (to appear)
RANSAC – Time Complexity

Repeat \( k \) times (\( k \) is a function of \( \eta, I, N \))

1. **Hypothesis generation**
   - Select a sample of \( m \) data points
   - Calculate parameters of the model(s)

2. **Model verification**
   - Find the support (consensus set) by
     - verifying all \( N \) data points

\[
t = k \left( t_M + \overline{m_s} N \right)
\]

- \( t_M \) — time needed to draw a sample
- \( \overline{m_s} \) — average number of models per sample

- \( I \) — the number of inliers
- \( N \) — the number of data points
- \( \eta \) — confidence in the solution
RANSAC time complexity

\[ t = k \left( t_M + \bar{m}sN \right) \]

The number of samples \( k \sim \frac{1}{P} \)

where \( P \) is a probability of drawing an all-inlier sample

\[ P = \frac{\binom{I}{m}}{\binom{N}{m}} \approx \varepsilon^m \]

where \( m \) is size of the sample

and \( \varepsilon \) is the fraction of inliers \( I/N \)
Randomised RANSAC [Matas, Chum 02]

Repeat $k/(1-\alpha)$ times
1. Hypothesis generation
2. Model pre-verification $T_{d,d}$ test
   - Verify $d \ll N$ data points, reject
   - the model if not all $d$ data points
   - are consistent with the model
3. Model verification
   Verify the rest of the data points

$V$ – average number of data points verified
$\alpha$ – probability that a good model is rejected by $T_{d,d}$ test

$$t = \frac{k}{1 - \alpha} \left( t_M + \bar{m}_s V \right)$$
Model Verification is Sequential Decision Making

\[ H_g: P(x_i = 1|H_g) \geq \varepsilon \]
\[ H_b: P(x_i = 1|H_b) = \delta \]

\( x_i = 1 \) \( x_i \) is consistent with the model

where

\( H_g \) - hypothesis of a ‘good’ model (\( \approx \) from an uncontaminated sample)
\( H_b \) - hypothesis of a ‘bad’ model, (\( \approx \) from a contaminated sample)
\( \delta \) - probability of a data point being consistent with an arbitrary model

Optimal (the fastest) test that ensures with probability \( \alpha \) that that \( H_g \) is not incorrectly rejected is the

Sequential probability ratio test (SPRT) [Wald47]
Two important properties of SPRT:

1. probability of rejecting a "good" model $\alpha < 1/A$

2. average number of verifications $V = C \log(A)$

Compute the likelihood ratio

$$\lambda_i = \prod_{j=1}^{i} \frac{P(x_j|H_b)}{P(x_j|H_g)}$$

if $\lambda_i > A$ reject the model

if $i = N$ accept model as 'good'
WaldSAC

Repeat \( k/(1-1/A) \) times
1. Hypothesis generation
2. Model verification
   use SPRT

\[
\begin{align*}
\text{Time} & \quad t_M \\
\bar{m}_s \cdot C \log A \\
C & \approx ((1 - \delta)\log \frac{1 - \delta}{1 - \varepsilon} + \delta \log \frac{\delta}{\varepsilon})^{-1}
\end{align*}
\]

\[
t(A) = \frac{k}{(1 - 1/A)}(t_M + \bar{m}_s C \log A)
\]

In sequential statistical decision problem decision errors are traded off for time. These are two incomparable quantities, hence the constrained optimization.

In WaldSAC, decision errors cost time (more samples) and there is a single minimised quantity, time \( t(A) \), a function of a single parameter \( A \).
Note: the Wald’s test is equivalent to series of $T(d, c)$, where $c = \left[ \text{log } A - d \log \lambda_1 \right] / \log \lambda_0$
Exp. 1: Wide-baseline matching

<table>
<thead>
<tr>
<th></th>
<th>samples</th>
<th>models</th>
<th>V</th>
<th>time</th>
<th>spd-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>2914</td>
<td>7347</td>
<td>110.0</td>
<td>1099504</td>
<td>1.0</td>
</tr>
<tr>
<td>R-R</td>
<td>7825</td>
<td>19737</td>
<td>3.0</td>
<td>841983</td>
<td>1.3</td>
</tr>
<tr>
<td>Wald</td>
<td>3426</td>
<td>8648</td>
<td>8.2</td>
<td>413227</td>
<td>2.7</td>
</tr>
</tbody>
</table>
Exp. 2 Narrow-baseline stereo

<table>
<thead>
<tr>
<th></th>
<th>samples</th>
<th>models</th>
<th>V</th>
<th>time</th>
<th>spd-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>155</td>
<td>367</td>
<td>600.0</td>
<td>235904</td>
<td>1.0</td>
</tr>
<tr>
<td>R-R</td>
<td>247</td>
<td>587</td>
<td>86.6</td>
<td>75539</td>
<td>3.1</td>
</tr>
<tr>
<td>Wald</td>
<td>162</td>
<td>384</td>
<td>23.1</td>
<td>25032</td>
<td>9.4</td>
</tr>
</tbody>
</table>
Randomised Verification in RANSAC: Conclusions

- The same confidence $\eta$ in the solution reached faster (data dependent, $\approx 10\times$)
- No change in the character of the algorithm, it was randomised anyway.
- Optimal strategy derived using Wald‘s theory for known $\varepsilon$ and $\delta$.
- Results with $\varepsilon$ and $\delta$ estimated during the course of RANSAC are not significantly different. Performance of SPRT is insensitive to errors in the estimate.
  - $\delta$ can be learnt, an initial estimate can be obtained by geometric consideration
  - Lower bound on $\varepsilon$ is given by the best-so-far support
  - Note that the properties of WaldSAC are quite different from preemptive RANSAC!
Conclusions

- RANSAC is in principle simple, but (as is often the case), a state-of-the-art implementation is not.

- A public version of RANSAC with PRO-sampling, LO-optimization, DEGENERACY protection and WALD-evaluation will be made public in (hopefully near) future.
Thank you.

Questions, please?