Robust Pose Optimization Made Differentiable

Eric Brachmann

5th International Workshop on Recovering 6D Object Pose @ICCV19
Dr. Eric Brachmann
@eric_brachmann

2012-2017
PhD at
since 2018
Post-Doc at
since 2019
Guest at
Main Research Interests

- Machine learning and projective geometry
- Robust fitting with (differentiable) RANSAC
  - Object poses
  - Camera poses
  - Lines
  - Epipolar Geometry

DSAC – CVPR’17
DSAC++ – CVPR’18
Object Coordinates – ECCV’14
Goal

Pose Estimation Pipeline

RGB(-D) Image \( I \) 

6D Poses \( \hat{h}_o \)

Object Detection 

Object Classification 

Correspondence Prediction 

RANSAC 

Pose Solver 

Pose Scoring 

Pose Loss 

“Learning 6D object pose estimation using 3D object coordinates”, Brachmann et al., ECCV’14 
“iPose: instance-aware 6D pose estimation of partly occluded objects”, Jafari et al., ACCV’18 
“Segmentation-driven 6D Object Pose Estimation”, Hu et al., CVPR’19 
“Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation”, Park et al., ICCV’19 
“DPOD: 6D Pose Object Detector and Refiner”, Zakharov et al., ICCV’19 

...
Why End-to-End?

Pose Estimation Pipeline

RGB(-D) Image $I$

Object Detection

Object Classification

Correspondence Prediction

RANSAC

Pose Solver

Pose Scoring

Pose Loss

6D Camera Pose $\hat{h}$
Why End-to-End?

Comparing reprojection error before and after end-to-end training:

- Improvement
- ±0px
- Degradation
- +10px

[NGRANSAC] “Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses”, Brachmann and Rother, ICCV19
<table>
<thead>
<tr>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object Detection</strong></td>
</tr>
<tr>
<td>Pose Solver</td>
</tr>
</tbody>
</table>

**Visual Learning Lab Heidelberg**
Pose Loss (RGB-D)

Object Detection  Object Classification  Correspondence Prediction  RANSAC  Pose Solver  Pose Scoring  Pose Loss

Input: RGB-D

\[ \ell(t, \mathbf{t}^*) + \alpha \ell(R, R^*) \] with \( \mathbf{h} = (t, R) \)

\[ \|t - \mathbf{t}^*\| \quad \|\log(R^*R^T)\| \] with \( \log: \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3 \)

\[ \theta \]

in OpenCV:
\[ \text{cv2.Rodrigues()} \]
incl. gradients
Pose Loss (RGB)

Object Detection | Object Classification | Correspondence Prediction | RANSAC | Pose Loss

Input: RGB

Z-Err: 5cm 10cm 20cm

\[ \ell_{\pi} (\mathbf{h}, \mathbf{h}^*) = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{v} \in \mathcal{V}} \| C \mathbf{h}^* \mathbf{v} - C \mathbf{h} \mathbf{v} \| \] [Bra16]

\( \mathcal{V} \) ... Model vertices
\( C \) ... Camera calibration matrix

Pose Solver (RGB-D)

Object Detection  Object Classification Correspondence Prediction RANSAC Pose Solver Pose Scoring Pose Loss

Input: RGB-D

Kabsch Algorithm:

\[
\text{cov}[x_i, y_i] = \sum_i (x_i - \bar{x})(y_i - \bar{y})^T
\]

\[
\text{cov}[x_i, y_i] = U \Sigma V^T
\]

\[
\hat{R} = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \text{det}(VU^T) \end{pmatrix} U^T
\]

\[
\hat{t} = \hat{R}\hat{y} - \bar{x}
\]

\[
(\hat{R}, \hat{t}) = \arg\min_R \sum_i \|x_i - (Ry_i - t)\|^2
\]

C++ code with PyTorch integration coming soon.

Pose Solver (RGB)

Object Detection  Object Classification  Correspondence Prediction  RANSAC  Pose Solver  Pose Scoring  Pose Loss

Input: RGB

\( p_i \)

\( y_i \)

\((R, \hat{t}) = \arg\min_{R,t} \sum_i \|p_i - C(Ry_i - t)\|^2\)

Solving Perspective-n-Point:

Initialization ➔ Gauss-Newton

[Gao03] Gao et al., “Complete Solution Classification for the Perspective-Three-Point Problem”, TPAMI’03
Residual vector: \[ r(h)_i = \|p_i - C h y_i \|^2 \]

Update Rule: \[ h^{t+1} = h^t - (J_r^T J_r)^{-1} J_r^T r(h^t) \]

Jacobian: \[ (J_r)_{ij} = \frac{\partial [r(h^t)]_i}{\partial [h^t]_j} \]
Pose Solver (RGB)

Object Detection  Object Classification  Correspondence Prediction  RANSAC  Pose Solver  Pose Scoring  Pose Loss

Initialization  ➔  Gauss-Newton

Residual vector: \( [r(h)]_i = \|p_i - Chy_i \|^2 \)

Update Rule: \( h^{t+1} = h^t - (J_r^T J_r)^{-1} J_r^T r(h^t) \)

Jacobian: \( [J_r]_{ij} = \frac{\partial [r(h^t)]_i}{\partial [h^t]_j} \)

Last update: \( \hat{h} = h^\infty - (J_r^T J_r)^{-1} J_r^T r(h^\infty) \)

Gradients: \( \frac{\partial}{\partial y_i} \hat{h} \approx - (J_r^T J_r)^{-1} J_r^T \frac{\partial}{\partial y_i} r(h^\infty) \)
Residual vector: \[ r(h)_i = \| p_i - C h y_i \|^2 \]

Update Rule: \[ h^{t+1} = h^t - (J_r^T J_r)^{-1} J_r^T r(h^t) \]

Jacobian: \[ [J_r]_{ij} = \frac{\partial [r(h^t)]]}{\partial [h^t]} \]

Last update: \[ \hat{h} = h^\infty - (J_r^T J_r)^{-1} J_r^T r(h^\infty) \]

Gradients: \[ \frac{\partial}{\partial y_i} \hat{h} \approx -(J_r^T J_r)^{-1} J_r^T \frac{\partial}{\partial y_i} r(h^\infty) \]


[Bra18] Brachmann and Rother, “Learning less is more - 6D camera localization via 3D surface regression”, CVPR’18
RANSAC

Object Detection | Object Classification | Correspondence Prediction | Pose Solver | Pose Scoring | Pose Loss

Hypothesis | Selection

RANSAC

Pose Solver
Pose Scoring
Pose Loss

Image Correspondence Prediction Hypothesis Sampling Scoring Hypothesis Selection Result

Soft Inlier Counting [Bra18]:
\[ s(h, y) = \sum_i \operatorname{sig}(\tau - \beta \| p_i - C h y_i \|) \]

argmax Selection
\[ \hat{h} = \arg\max_{h_j} s(h_j, y) \]

hard decision
non-differentiable

Probabilistic Selection [Bra17]
\[ \hat{h} = h_j, \text{ where } j \sim \frac{\exp(s(h_j, y))}{\sum_k \exp(s(h_k, y))} \]

hard decision
differentiable

[Bra17] Brachmann et al., “DSAC - Differentiable RANSAC for camera localization”, CVPR’17
[Bra18] Brachmann and Rother, “Learning less is more - 6D camera localization via 3D surface regression”, CVPR’18
Differentiable RANSAC (DSAC)

Hypothesis selection: \[ \hat{h} = h_j, \text{where } j \sim \frac{\exp(s(h_j, y))}{\sum_k \exp(s(h_k, y))} = P(j; y) \]

Learning objective: \[ \mathcal{L}(y) = \mathbb{E}_{j \sim P(j; y)}[\ell(h_j, h^*)] \]

Gradients: \[ \frac{\partial}{\partial y} \mathcal{L}(y) = \mathbb{E}_{j \sim P(j; y)} \left[ \ell(h_j, h^*) \frac{\partial}{\partial y} \log P(j; y) + \frac{\partial}{\partial y} \ell(h_j, h^*) \right] \]

derivative of selection probability derivative of task loss

[Bra17] Brachmann et al., “DSAC - Differentiable RANSAC for camera localization”, CVPR’17

C++ code for camera re-localization online. PyTorch code for DSAC line fitting also online.
Differentiable RANSAC (DSAC)

<table>
<thead>
<tr>
<th>Method</th>
<th>Error (cm)</th>
<th>Error (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoseNet</td>
<td>149</td>
<td>3.4</td>
</tr>
<tr>
<td>Active Search</td>
<td>19</td>
<td>0.5</td>
</tr>
<tr>
<td>DSAC++</td>
<td>13</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Test Video
Cambridge St Mary’s Church

Our Estimates
Trained with the 3D Model

[Posenet] “Geometric Loss Functions for Camera Pose Regression with Deep Learning” Kendall and Cipolla, CVPR ’17
[Active Search] “Efficient & effective prioritized matching for large-scale image-based localization”, Sattler et al., TPAMI’17
[DSAC] “DSAC - Differentiable RANSAC for Camera Localization”, Brachmann et al., CVPR’17
[DSAC++] “Learning Less is More – 6D Camera Localization via 3D Surface Regression”, Brachmann and Rother, CVPR’18
Correspondence Prediction

Input Image → Dense Correspondences → RANSAC / DSAC

Object Detection
Object Classification
Correspondence Prediction
RANSAC
Pose Solver
Pose Scoring
Pose Loss

Input Image

Dense Correspondences

RANSAC / DSAC

W
Neural Guided RANSAC (NG-RANSAC)

Object Detection
Object Classification
Correspondence Prediction
RANSAC
Pose Solver
Pose Scoring
Pose Loss

Input Image

Dense Correspondences

RANSAC / DSAC

Selecting a scene coordinate: $p(y) = g(I; w)$
Selecting a hypothesis: $p(h) = \prod_{i=0}^4 p(y_i)$
Selecting a hypotheses pool: $p(H) = \prod_j p(h_j)$
Learning objective: $\mathbb{E}_{H \sim p(H)} [L(w)]$

$\mathbb{E}_{H \sim p(H)} \mathbb{E}_{j \sim \mathcal{P}(j|H; w)} [\ell(h_j, h^*)]$

Neural Guidance
DSAC
Neural Guided RANSAC (NG-RANSAC)

Object Detection  Object Classification  Correspondence Prediction  RANSAC  Pose Loss

<table>
<thead>
<tr>
<th></th>
<th>PoseNet</th>
<th>ActiveSearch</th>
<th>DSAC++</th>
<th>NG-DSAC++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Great Court</td>
<td>700cm</td>
<td>-</td>
<td>40.3cm</td>
<td>35.0cm</td>
</tr>
<tr>
<td>Kings College</td>
<td>99cm</td>
<td>42cm</td>
<td>13.0cm</td>
<td>12.6cm</td>
</tr>
<tr>
<td>Old Hospital</td>
<td>217cm</td>
<td>44cm</td>
<td>22.4cm</td>
<td>21.9cm</td>
</tr>
<tr>
<td>Shop Facade</td>
<td>107cm</td>
<td>12cm</td>
<td>5.7cm</td>
<td>5.6cm</td>
</tr>
<tr>
<td>St M. Church</td>
<td>149cm</td>
<td>19cm</td>
<td>9.9cm</td>
<td>9.8cm</td>
</tr>
</tbody>
</table>

[ActiveSearch] “Efficient & effective prioritized matching for large-scale image-based localization”, Sattler et al., TPAMI’17
[DSAC++] “Learning Less is More – 6D Camera Localization via 3D Surface Regression”, Brachmann and Rother, CVPR’18

[NG-DSAC++] “Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses”, Brachmann and Rother, ICCV19
Object Classification

Object Detection

Object Classification

Correspondence Prediction

RANSAC
  Pose Solver
  Pose Scoring

Pose Loss

Environment

Classes

Query Image

[ESAC] „Expert Sample Consensus Applied to Camera Re-Localization“, Brachmann and Rother, ICCV’19
Object Classification

7Scenes+12Scenes [ESAC]

Average Accuracy (5cm, 5°):
Classification + DSAC++: 47.5%
Oracle + DSAC++: 89.0%

[DSAC++] Brachmann and Rother, “Learning less is more - 6D camera localization via 3D surface regression”, CVPR’18
[ESAC] „Expert Sample Consensus Applied to Camera Re-Localization”, Brachmann and Rother, ICCV’19
Object Classification

7Scenes+12Scenes [ESAC]

Average Accuracy (5cm, 5°):
Classification + DSAC++: 47.5%
Oracle + DSAC++: 89.0%

[DSAC++] Brachmann and Rother, “Learning less is more - 6D camera localization via 3D surface regression”, CVPR’18
[ESAC] „Expert Sample Consensus Applied to Camera Re-Localization”, Brachmann and Rother, ICCV’19
Object Classification

Gating Network

Expert Networks

RANSAC Hypotheses \( \mathcal{H} \)

[ESAC] „Expert Sample Consensus Applied to Camera Re-Localization”, Brachmann and Rother, ICCV’19
[ESAC] „Expert Sample Consensus Applied to Camera Re-Localization”, Brachmann and Rother, ICCV’19
Expert Sample Consensus

Differentiable Objective Function:
\[ L(w) = \mathbb{E}_{\mathcal{H} \sim P(\mathcal{H})} \mathbb{E}_{j \sim P(j|\mathcal{H})} \left[ \ell(h_j) \right] \]
\[ P(\mathcal{H}) \propto g(l, w) \]

[ESAC] „Expert Sample Consensus Applied to Camera Re-Localization”, Brachmann and Rother, ICCV’19
Expert Sample Consensus

7Scenes+12Scenes [ESAC]

Average Accuracy (5cm, 5°):
Classification + DSAC++: 47.5%
Oracle + DSAC++: 89.0%
ESAC: 88.1%

[DSAC++] Brachmann and Rother, “Learning less is more - 6D camera localization via 3D surface regression”, CVPR’18
Object Detection

- Object Detection
- Object Classification
- Correspondence Prediction
- RANSAC
  - Pose Solver
  - Pose Scoring
- Pose Loss
Conclusion:

- Differentiable PnP [Bra18]
- Differentiable RANSAC → [DSAC]
- Differentiable Correspondence Selection → [NG-RANSAC]
- Differentiable Expert Selection → [ESAC]

[Bra18] Brachmann and Rother, “Learning less is more - 6D camera localization via 3D surface regression”, CVPR’18
[DSAC] Brachmann et al., “DSAC - Differentiable RANSAC for camera localization”, CVPR’17
[NG-RANSAC] Brachmann and Rother, “Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses”, ICCV19
Conclusion:

• Differentiable PnP [Bra18]
• Differentiable RANSAC $\rightarrow$ [DSAC]
• Differentiable Correspondence Selection $\rightarrow$ [NG-RANSAC]
• Differentiable Expert Selection $\rightarrow$ [ESAC]

Code of many methods online:

DSAC for camera re-localization [Lua/Torch]: https://github.com/cvlab-dresden/DSAC
DSAC for Line Fitting [PyTorch]: https://github.com/vislearn/DSACLine
DSAC*, improved DSAC++ incl. differentiable PnP and differentiable Kabsch [PyTorch]: Coming soon
ESAC, differentiable expert selection [PyTorch]: Coming soon (https://hci.iwr.uni-heidelberg.de/vislearn/research/scene-understanding/pose-estimation/#ICCV19)
NG-DSAC, differentiable correspondence selection [PyTorch]: Coming soon (https://hci.iwr.uni-heidelberg.de/vislearn/research/neural-guided-ransac/)
Thank You!