Generalized Pose Estimation from Line Correspondences with Known Vertical Direction

Nora Horanyi, Zoltan Kato

Research Group on Visual Computation, University of Szeged

(http://www.inf.u-szeged.hu/rgvc/)

Problem statement

Goal: absolute pose estimation, to determine the position and orientation of a camera with respect to a 3D world coordinate frame

Contribution: We propose a method to compute absolute pose of a generalized camera based on 3D straight lines

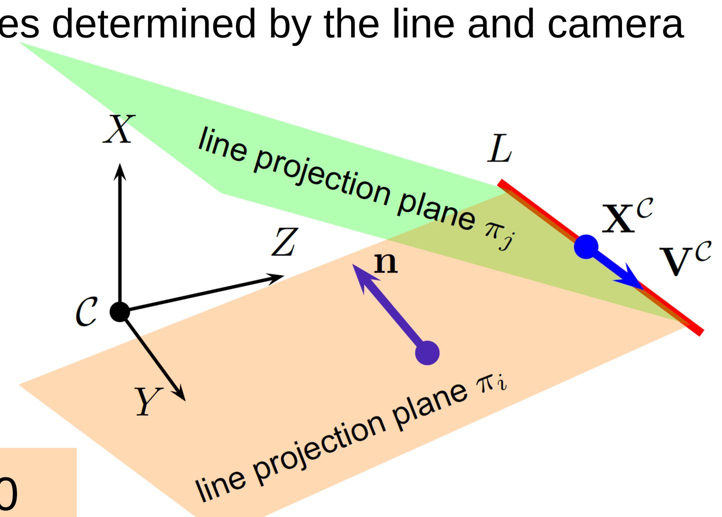
Assumption:

- Vertical direction is available
- 3D straight lines are projected via projection planes determined by the line and camera projection directions
- 3D lines are represented as L = (v, X)
- Projection planes are given as $\pi_i^L = (\mathbf{n}_i^\top, d_i)$, $\mathbf{n}_i$ is the unit normal to a projection plane
- $\mathbf{V^C}$ is on the projection plane →

(1) $\mathbf{n}_i^\top \mathbf{V^C} = \mathbf{n}_i^\top \mathbf{Rv} = 0$

- $\mathbf{X^C}$ point is on the projection plane →

(2) $(\pi_i^L)^\top(\mathbf{X^C},1) = \mathbf{n}_i^\top(\mathbf{RX} + \mathbf{t}) + d_i = 0$



Solution: We formulate the problem in terms of 4 unknowns using 3D line–projection plane correspondences which yields a closed form solution. It can be used as a minimal solver as well as least squares solver without reformulation

Efficient solution: gPnLup

- Vertical direction is known when the camera system is coupled with *e.g.* an IMU
  → rotation $\mathbf{R}_v$ around Y and Z axes are known, $\mathbf{R}_v = \mathbf{R}_z\mathbf{R}_y$
  → rotation of absolute pose: $\mathbf{R} = \mathbf{R}_v\mathbf{R}_X(\alpha)$
- We thus have 4 unknowns: the rotation angle α and the 3 translation components of **t**

How to get rid of the trigonometric functions in $\mathbf{R}_X(\alpha)$?

- Substituting $q = \tan(\alpha/2)$, gives us the following form of $\mathbf{R}_X(\alpha)$:

$$\mathbf{R}_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \Rightarrow \mathbf{R}_X(q) = \frac{1}{(1+q^2)} \begin{bmatrix} 1+q^2 & 0 & 0 \\ 0 & 1-q^2 & -2q \\ 0 & 2q & 1-q^2 \end{bmatrix}$$

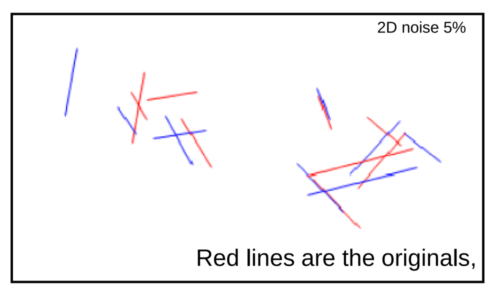
1. Backsubstitute $\mathbf{R}_X(\alpha)$ into (1) and solve it in the least squares sense:
   Squared error: $\sum_{i=1}^{n}(a_i^2 q^4 + 2a_ib_iq^3 + (2a_ic_i + b_i^2)q^2 + 2b_ic_iq + c_i^2)$
   Its derivative should vanish: $\sum_{i=1}^{n}(4a_i^2 q^3 + 6a_ib_iq^2 + (4a_ic_i + 2b_i^2)q + 2b_ic_i)$
   → the 3 roots are the possible solutions for q
2. Backsubstitute each real q into (2) → linear system of equations in **t**
3. Select the final solution with the minimal reprojection error

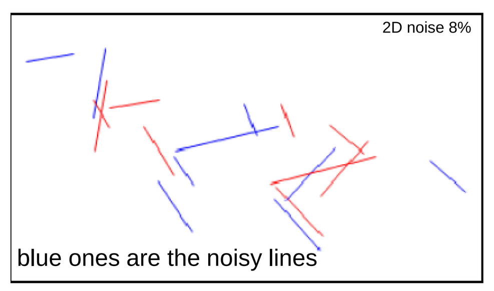
Synthetic data

Various benchmark datasets of 3D–2D line pairs

For robustness tests we add random noise to these datasets in the following way:

- 2D lines are corrupted with additive random noise on one endpoint of the line and the direction vector of the line (5% and 8%)
- This corresponds to a quite high noise rate: [-20, +20] pixels for the 5% case and [-30, +30] pixels for the 8% case
- We evaluate our method as a **least squares solver** as well as a **minimal solver**
- We need 3 line pairs in the minimal case
- Implementation in MATLAB, typical runtime of our method was 9.8ms

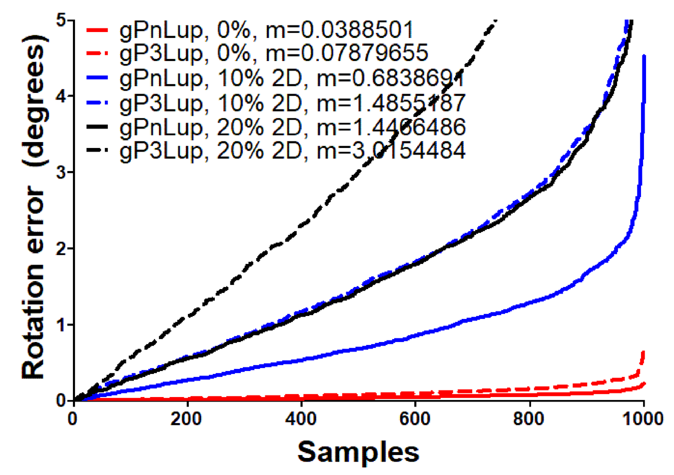




Quantitative evaluation

Noise sensitivity with a 1-omni-2-perspective camera system

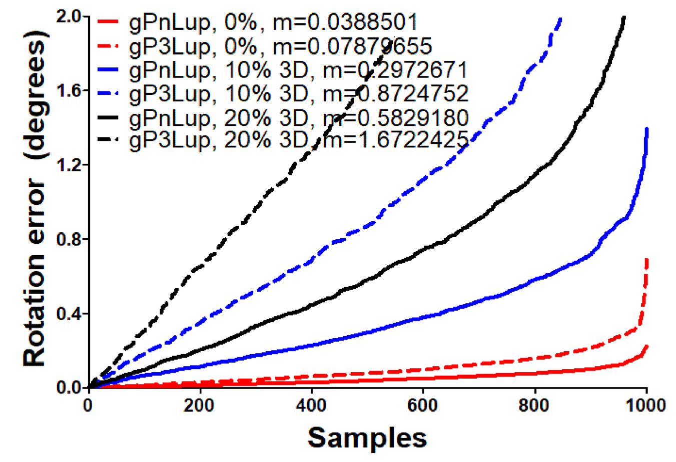
2D noise



Robust up to 10%

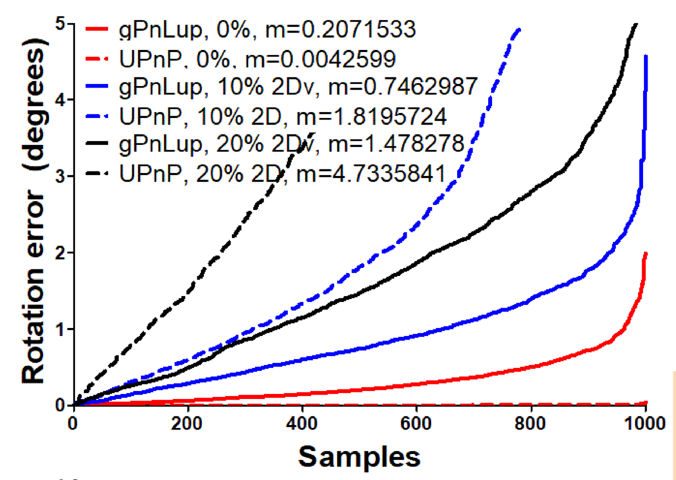
Robust up to 20%

3D noise

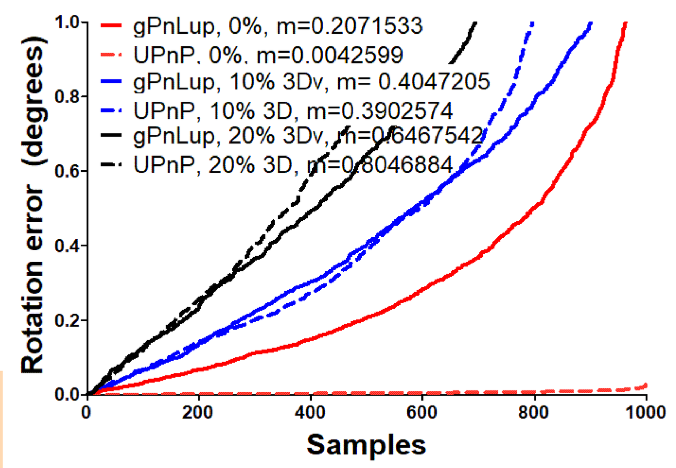


Comparison of our algorithm with UPnP

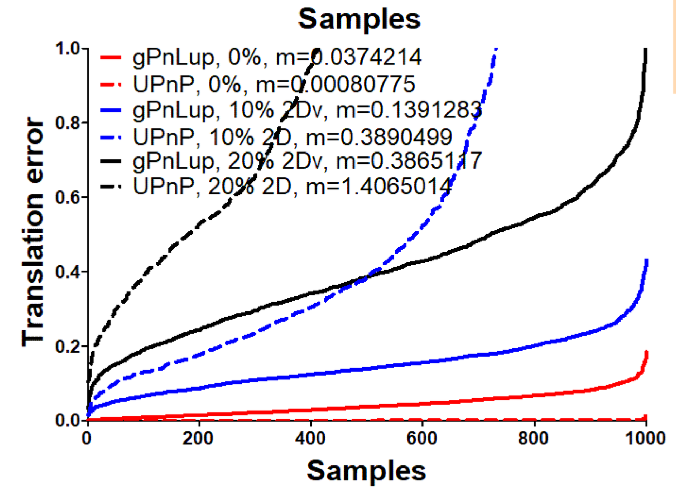
2D noise

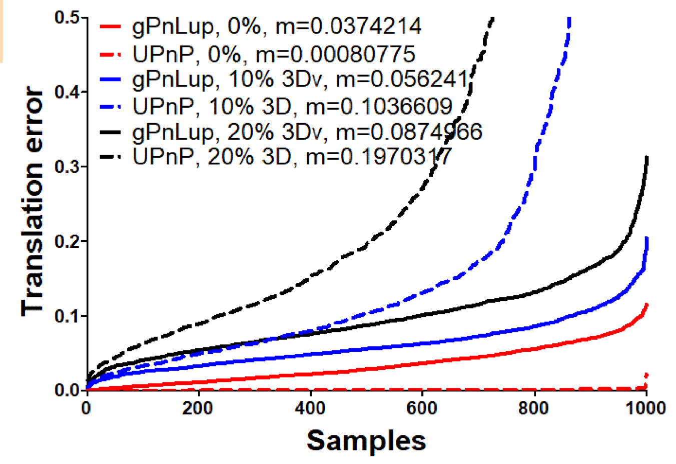


3D noise

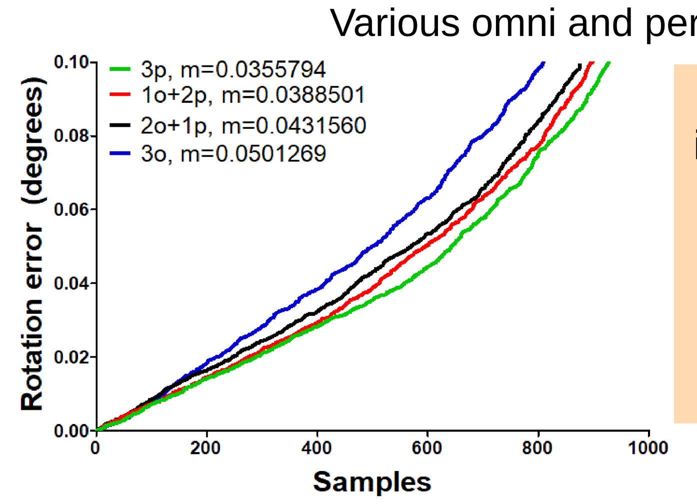


Our algorithm is more robust than UPnP

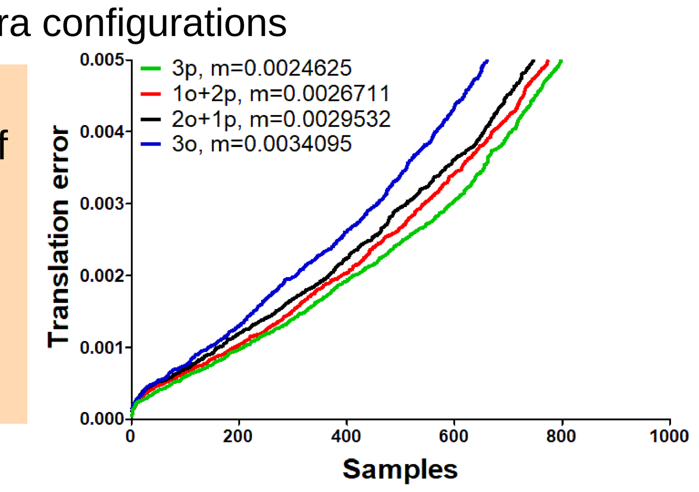




Various omni and perspective camera configurations



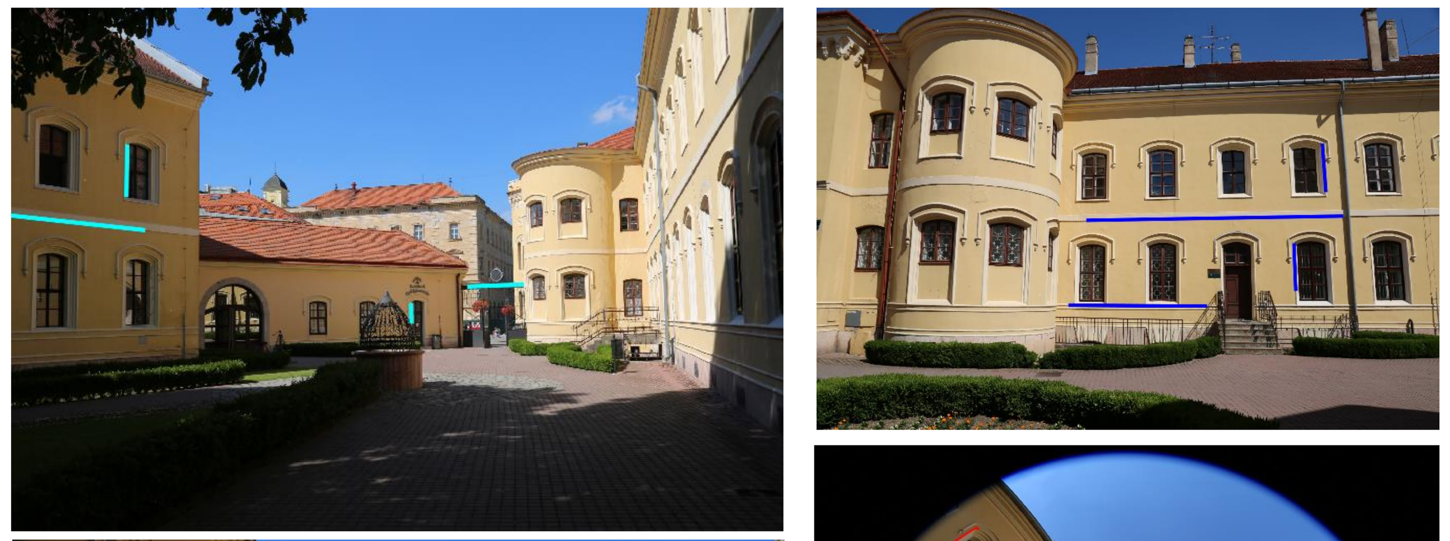
Accuracy is independent of the camera system composition (median error<0.05°)

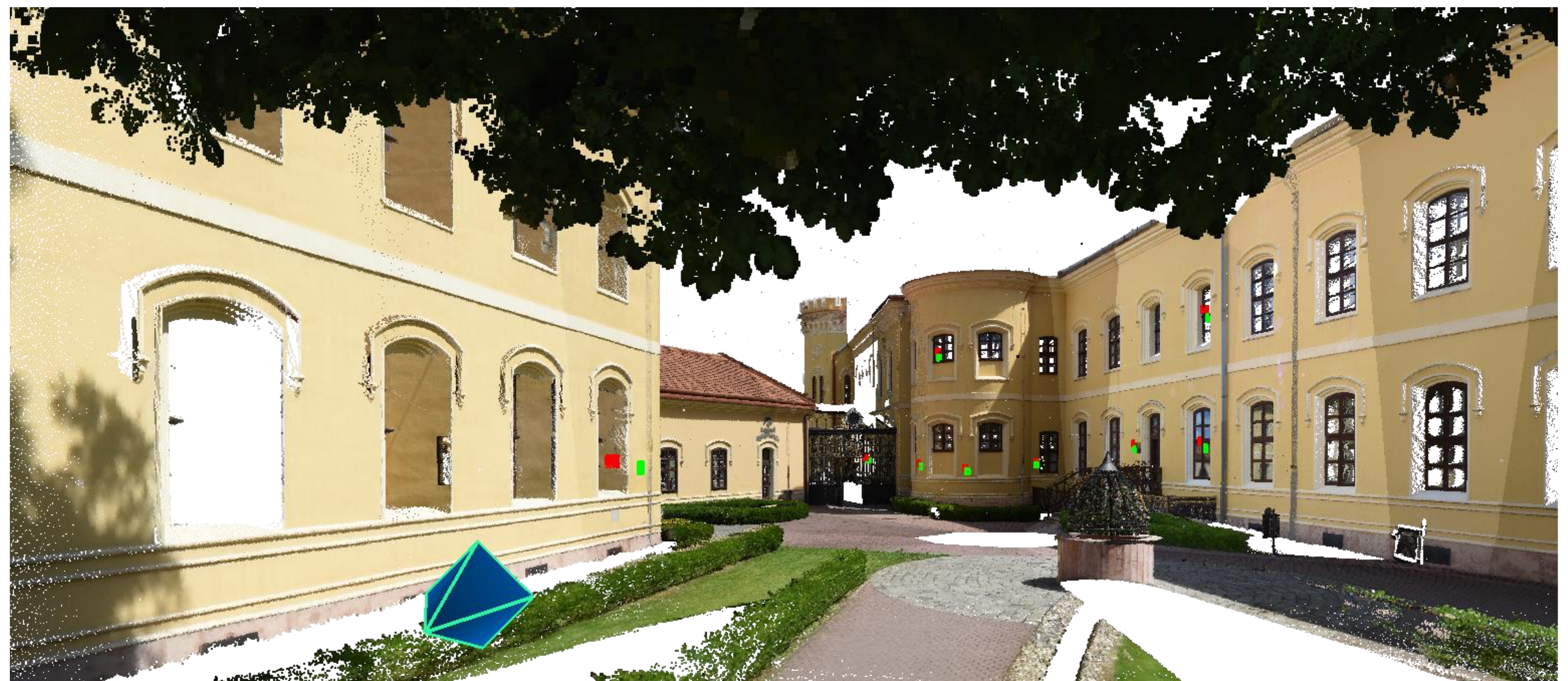


Conclusion

- We proposed a direct least squares solution which can be used as a minimal solver (*e.g.* within RANSAC) as well as a general least squares solver without reformulation
- The only assumption about our generalized camera is that 3D lines project through a set of projection planes. Typical camera setups include:
  - Stereo and multiview central camera systems composed of perspective and non-Perspective (*e.g.* omnidirectional) cameras
  - Camera (system) moving along a trajectory
  - Linear pushbroom imaging
- The proposed method have been evaluated on synthetic and real datasets. Comparative tests confirm state of the art performance both in terms of quality and computing time.

Real datasets





Comparison of the maximal rotational, translational and forward projection error of various methods on the real data.	Least squares solver gPnLup	Minimal solver gPnLup	Minimal solver NP3L
Rotation error (deg)	1.1972	1.0216	4.5029
Translation error	0.8797	0.9088	3.0037
Forward projection error(m)	0.2407	0.2904	0.5901

Least squares configuration (shown in the figure)**:**

- Lidar laser scan
- 3-perspective-1-omnidirectional multi-view camera system
- Extracted 2D lines are shown on the 2D images
- Markers in Lidar metric 3D space: red = estimated positions, green = true locations

Minimal configuration: Lidar scan + 3-perspective camera system.
Comparison with NP3L algorithm of G. H. Lee [ECCV 2016]

Acknowledgement

- NKFI-6 fund through project K120366
- the Agence Universitaire de la Francophonie (AUF) and the Romanian Institute for Atomic Physics (IFA), through the AUF-RO project NETASSIST
- "Integrated program for training new generation of scientists in the fields of computer science", no EFOP-3.6.3-VEKOP-16-2017-0002
- the Research & Development Operational Programme for the project "Modernization and Improvement of Technical Infrastructure for Research and Development of J. Selye University in the Fields of Nanotechnology and Intelligent Space", ITMS 26210120042, co-funded by the European Regional Development Fund

International Conference on 3D Vision (3DV), October 10-12, 2017, Qingdao, China