Multiview Absolute Pose Using 3D - 2D Perspective Line Correspondences and Vertical Direction

Nora Horanyi, Zoltan Kato

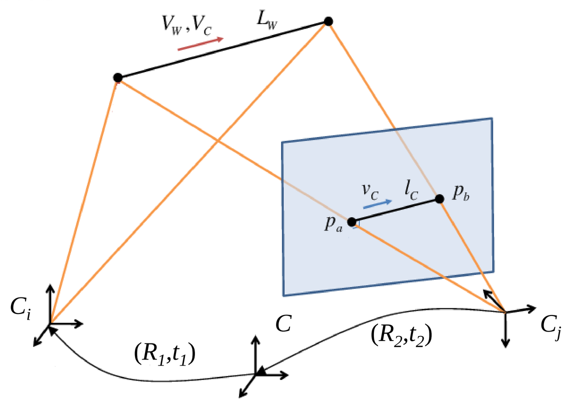
Research Group on Visual Computation, University of Szeged

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

Problem statement

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

Contribution: We propose two methods to compute absolute pose from 3D - 2D perspective line pairs. Both can be used as a minimal solver as well as least squares solver without reformulation



Assumption:

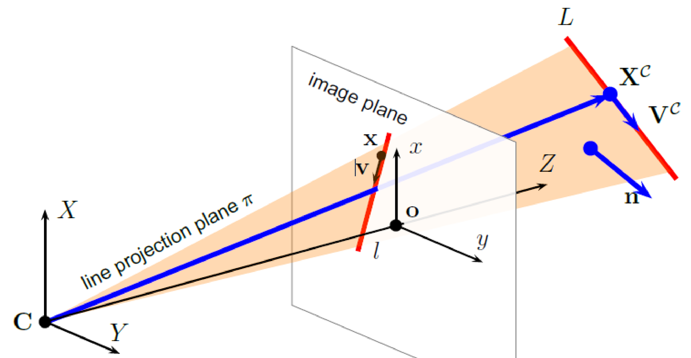
- Vertical direction is available
- 3D lines are represented as L = (V, X)
- Projection of L is given as $l = (v, x)$
- **n** is the unit normal to the projection plane

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

$$(1) \quad \mathbf{n}^{i\top}\mathbf{V}^C = \mathbf{n}^{i\top}\mathbf{R}^i\mathbf{R}\mathbf{V} = 0$$

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

$$(2) \quad \mathbf{n}^{i\top}(\mathbf{R}^i\mathbf{X}^C + \mathbf{t}^i) = \mathbf{n}_i^{\top}(\mathbf{R}^i(\mathbf{R}\mathbf{X} + \mathbf{t}) + \mathbf{t}^i) = 0$$



Solution: The first solution consists of a single linear system of equations, while the second solution yields a polynomial equation of degree three in one variable and one systems of linear equations which can be efficiently solved in closed-form.

Vertical direction: 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**

Efficient solutions

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

I. Solution: Linear solution (NPnLupL)

Let c=cos(α) and s=sin(α) be separate unknowns:

$$\mathbf{R}_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \Rightarrow \mathbf{R}_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}$$

Subtitution into (1),(2) yields a single linear system of equations

II. Solution: Cubic polynomial solution (NPnLupC)

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

Method	Advantages	Disadvantages
NPnLupL	• Linear solver has good accuracy for reasonable computing time • Provides quite stable pose estimates under moderate noise levels	• Orthonormality constraint on $\mathbf{R}_X(\alpha)$ has been ignored ($c^2 + s^2 = 1$) ➔ solution can be far from a rigid body transformation for noisy input data
NPnLupC	• Trigonometric constraint on α is explicitly taken into account • Increased robustness under noisy observations	• The estimation of α and **t** is decoupled, any error in α is directly propagated into the linear system of **t** • Computational complexity is slightly higher than the purely linear solver

Synthetic data

Various benchmark datasets of 3D–2D line pairs

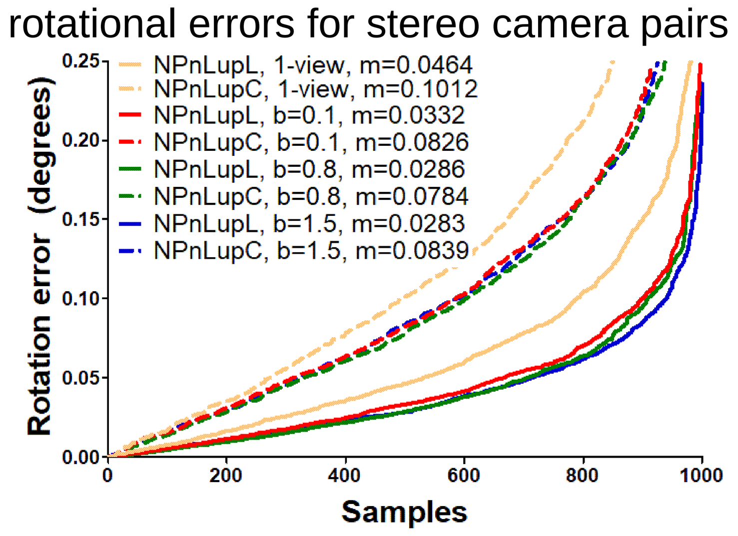
For roboustness tests we add random noise to these datasets in the following way:

- 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 methods as **least squares solver** as well as **minimal solver**
- We need 3 line pairs in the minimal case
- Implementation of the methods in MATLAB

n lines	NPnLupL	NPnLupC	RPnL	
Run time (ms)	0.9	1.3	8.8	
3 lines	**NPnLupL**	**NPnLupC**	UP3P	NP3L
Run time (ms)	0.4	0. 5	6.3	29.8

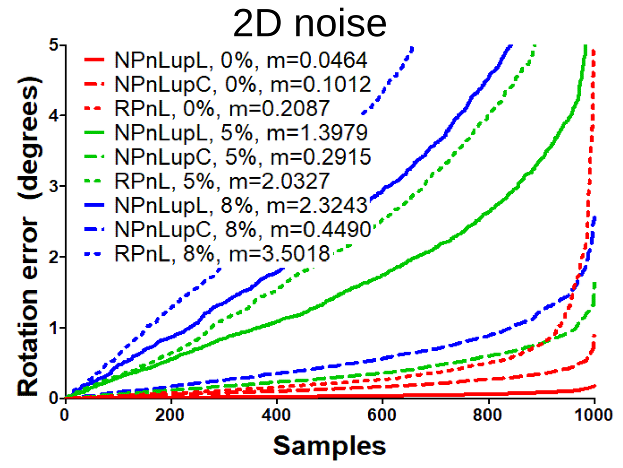
Quantitative evaluation

Comparison of the rotational errors for stereo camera pairs w.r.t. the baseline



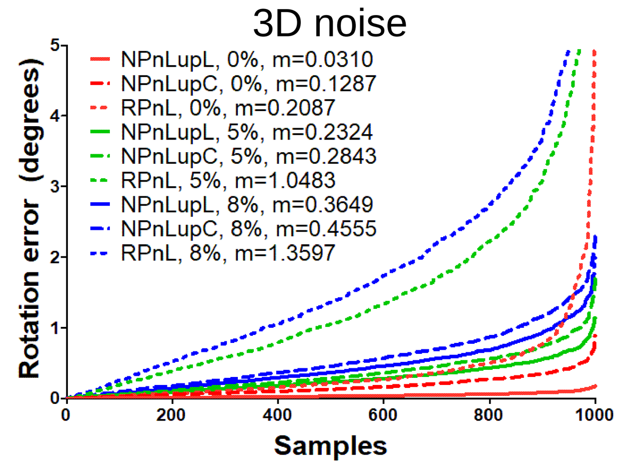
Comparison of our algorithms with RPnL of Zhang *et al.* [ACCV 2012] with 3 cameras

2D noise



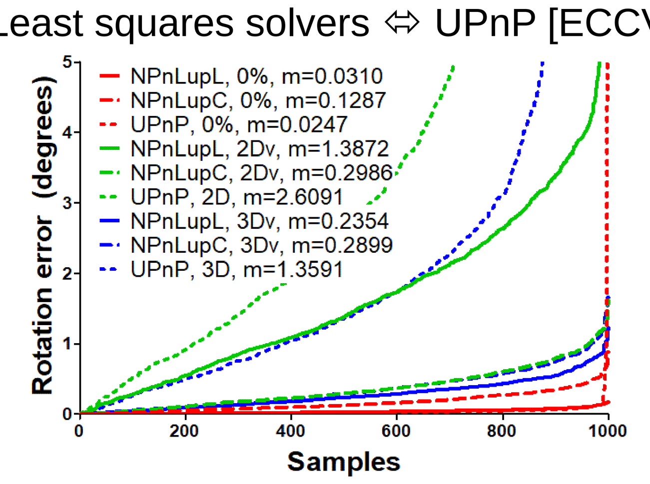
Our algorithm is more robust than RPnL

3D noise



Comparison of our algorithms with state of the art methods with 3 cameras

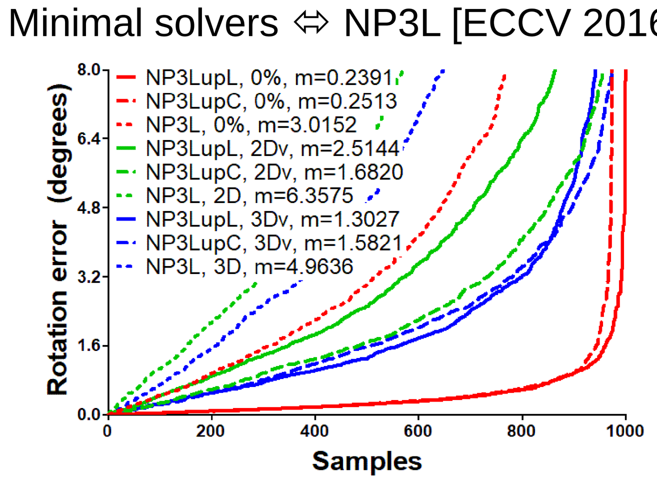
Least squares solvers ⬄ UPnP [ECCV 2014]



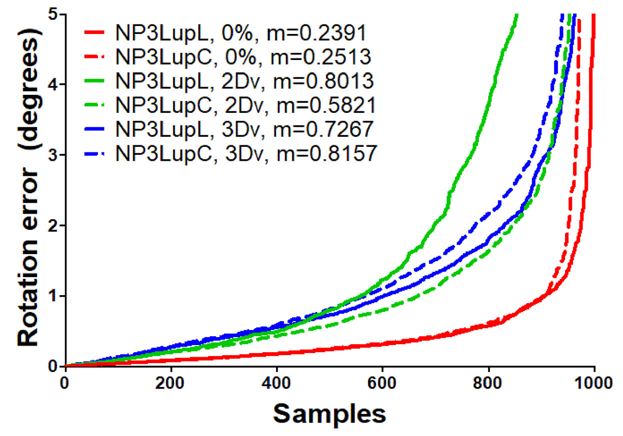
More robust than UPnP

More robust than NP3L

Minimal solvers ⬄ NP3L [ECCV 2016]



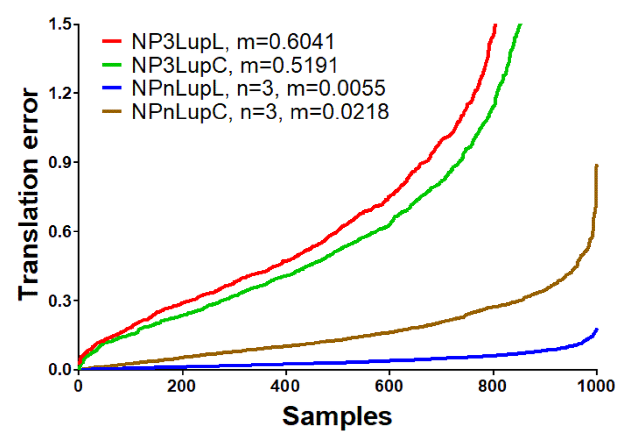
Efficiency of our minimal solutions for stereo camera pairs



2Dv: 5% 2D noise with 0.5° vertical noise

3Dv: 5% 3D noise with 0.5° vertical noise

Translation errors in case of 3 cameras for both 3 and *n* lines



Real datasets

Kolozsnema dataset	NPnLupL	NPnLupC	UPnP
Rotation error (deg)	0.0166	0.0176	0.0498
Translation error	0.0402	0.0319	0.0119

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

- Lidar laser scan
- 3-perspective multi-view camera system
- Extracted 2D lines are shown on the 2D images
- Colors of the lines are corresponding to their camera in the 3D point cloud
- Comparison with UPnP point based algorithm of Kneip *et al.* [ECCV 2014]

Conclusion

- We proposed two direct solutions which can be used as minimal solver (*e.g.* within RANSAC) as well as general least squares solver without reformulation
- Methods work for single- and multi-view perspective camera systems
- Linear solver is computationally more efficient but it is more sensitive to noise and low number of correspondences
- Cubic solver is much more robust at the price of slightly increased CPU time
- 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.

Acknowledgement

- NKFI-6 fund through project K120366
- Agence Universitaire de la Francophonie (AUF) and the Romanian Institute for Atomic Physics (IFA), through the AUF-RO project NETASSIST
- Research & Development Operational Programe 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.