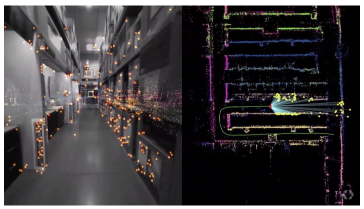
An Efficient Algebraic Solution to the Perspective-Three-Point Problem

Tong Ke and Stergios I. Roumeliotis
Multiple Autonomous Robotic Systems (MARS) Lab - University of Minnesota

IEEE 2017 Conference on Computer Vision and Pattern Recognition

Introduction

- Camera position and orientation (pose) estimation based on known landmarks is used in numerous applications (e.g., VR/AR)



- Perspective-3-Point (P3P) Problem
  - Estimate the 6 dof of camera pose from 3 3D-to-2D point correspondences
- Previous work
  - Solving for the distances first:
    – Grunert (1841), Haralick et al. (1991), Gao et al. (2003)
  - Solving for the camera's pose directly:
    – Kneip et al. (2011), Masselli and Zell (2014)

References

[1] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2969–2976, Colorado Springs, CO, June 21–25 2011.

[2] A. Masselli and A. Zell. A new geometric approach for faster solving the perspective-three-point problem. In Proc. of the IEEE International Conference on Pattern Recognition, pages 2119–2124, Stockholm, Sweden, Aug. 24–28 2014.

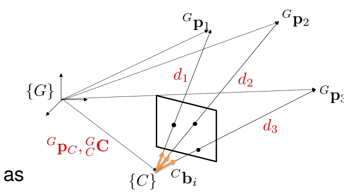
[3] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(8):930–943, 2003.

[4] J. A. Grunert. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. Grunerts archiv für mathematik und physik, 1:238–248, 1841.

Acknowledgements

This work was supported by the National Science Foundation (IIS-1328722).

Proposed P3P Approach



$${}^G\mathbf{p}_i = {}^G\mathbf{p}_C + d_i {}^G_C\mathbf{C}\,{}^C\mathbf{b}_i,\ i = 1,2,3$$

- Step 1: Eliminate position
  $\mathbf{p}_i - \mathbf{p}_j = {}^G_C\mathbf{C}(d_i\mathbf{b}_i - d_j\mathbf{b}_j)$
- Step 2: Eliminate distances
  $(\mathbf{p}_i - \mathbf{p}_j)^T {}^G_C\mathbf{C}(\mathbf{b}_i \times \mathbf{b}_j) = 0$
- Step 3: Describe the rotation matrix as
  ${}^G_C\mathbf{C} = \mathbf{C}(\mathbf{k}_1, \theta_1)\mathbf{C}(\mathbf{k}_2, \theta_2)\mathbf{C}(\mathbf{k}_3, \theta_3)$
  $$\mathbf{k}_1 \triangleq \frac{\mathbf{p}_1 - \mathbf{p}_2}{\|\mathbf{p}_1 - \mathbf{p}_2\|},\ \mathbf{k}_3 \triangleq \frac{\mathbf{b}_1 \times \mathbf{b}_2}{\|\mathbf{b}_1 \times \mathbf{b}_2\|},\ \mathbf{k}_2 \triangleq \frac{\mathbf{k}_1 \times \mathbf{k}_3}{\|\mathbf{k}_1 \times \mathbf{k}_3\|}$$
- Step 4: Determine 1 dof of rotation
  $\mathbf{k}_1^T {}^G_C\mathbf{C}\mathbf{k}_3 = 0$

$$\Rightarrow \mathbf{k}_1^T\mathbf{C}(\mathbf{k}_1, \theta_1)\mathbf{C}(\mathbf{k}_2, \theta_2)\mathbf{C}(\mathbf{k}_3, \theta_3)\mathbf{k}_3 = 0$$
$$\Rightarrow \mathbf{k}_1^T\mathbf{C}(\mathbf{k}_2, \theta_2)\mathbf{k}_3 = 0 \Rightarrow \theta_2 = \arccos(\mathbf{k}_1^T\mathbf{k}_3) - \frac{\pi}{2}$$

- Step 5: Substitute θ_2 back to the other 2 equations

$$\begin{bmatrix}\cos\theta_1\\ \sin\theta_1\end{bmatrix}^T \begin{bmatrix}\mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor^2\lfloor\mathbf{k}'_3\rfloor^2\mathbf{v}'_i & \mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor^2\lfloor\mathbf{k}'_3\rfloor\mathbf{v}'_i\\ \mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor\lfloor\mathbf{k}'_3\rfloor^2\mathbf{v}'_i & \mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor\lfloor\mathbf{k}'_3\rfloor\mathbf{v}'_i\end{bmatrix} \begin{bmatrix}\cos\theta_3\\ \sin\theta_3\end{bmatrix} + (\mathbf{k}_1^T\mathbf{u}_i)\begin{bmatrix}-\mathbf{k}_1^T\lfloor\mathbf{k}'_3\rfloor^2\mathbf{v}'_i & -\mathbf{k}_1^T\lfloor\mathbf{k}'_3\rfloor\mathbf{v}'_i\end{bmatrix}\begin{bmatrix}\cos\theta_3\\ \sin\theta_3\end{bmatrix} =$$
$$(\mathbf{k}'^T_3\mathbf{v}'_i)\begin{bmatrix}\mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor\lfloor\mathbf{k}'_3\rfloor\mathbf{k}_1 & \mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor\mathbf{k}'_3\end{bmatrix}\begin{bmatrix}\cos\theta_1\\ \sin\theta_1\end{bmatrix} \quad (1)$$

$$\mathbf{u}_i \triangleq \mathbf{p}_i - \mathbf{p}_3,\ \mathbf{v}_i \triangleq \mathbf{b}_i \times \mathbf{b}_3,\ \mathbf{v}'_i \triangleq \mathbf{C}(\mathbf{k}_2, \theta_2)\mathbf{v}_i,\ i = 1,2$$
$$\mathbf{k}'_3 \triangleq \mathbf{C}(\mathbf{k}_2, \theta_2),\ \mathbf{k}_3 = \mathbf{k}_2 \times \mathbf{k}_1$$

- Step 6: Change of variables
  $\theta'_1 \triangleq \theta_1 - \phi,\ \mathbf{v}''_i \triangleq \mathbf{C}(\mathbf{k}_1, \phi)\mathbf{v}'_i,\ \mathbf{k}''_3 \triangleq \mathbf{C}(\mathbf{k}_1, \phi)\mathbf{k}'_3,\ \phi = \mathrm{atan2}(\mathbf{u}_1^T\mathbf{k}'_3, \mathbf{u}_1^T\mathbf{k}_2)$
- Step 7: Rewrite (1) as

$$\begin{bmatrix}\cos\theta'_1\\ \sin\theta'_1\end{bmatrix}^T \begin{bmatrix}\mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor^2\lfloor\mathbf{k}''_3\rfloor^2\mathbf{v}''_i & \mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor^2\lfloor\mathbf{k}''_3\rfloor\mathbf{v}''_i\\ 0 & 0\end{bmatrix} \begin{bmatrix}\cos\theta_3\\ \sin\theta_3\end{bmatrix} + (\mathbf{k}_1^T\mathbf{u}_i)\begin{bmatrix}-\mathbf{k}_1^T\lfloor\mathbf{k}''_3\rfloor^2\mathbf{v}''_i & -\mathbf{k}_1^T\lfloor\mathbf{k}''_3\rfloor\mathbf{v}''_i\end{bmatrix}\begin{bmatrix}\cos\theta_3\\ \sin\theta_3\end{bmatrix} =$$
$$(\mathbf{k}''^T_3\mathbf{v}''_i)\begin{bmatrix}0 & \mathbf{u}_i^T\lfloor\mathbf{k}_1\rfloor\mathbf{k}''_3\end{bmatrix}\begin{bmatrix}\cos\theta'_1\\ \sin\theta'_1\end{bmatrix} \quad (2)$$

- Step 8: Use (3) to eliminate θ_3 in (2) to get a quadratic eq. of $\cos\theta'_1, \sin\theta'_1$
  $$\cos\theta_3^2 + \sin\theta_3^2 = 1 \quad (3)$$
- Step 9: Eliminate $\sin\theta'_1$ to get a quartic equation of $\cos\theta'_1$
- Step 10: Solve the quartic eq. and back substitute to recover ${}^G_C\mathbf{C}, {}^G\mathbf{p}_C$

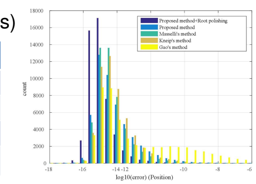
Results

- Processing cost (on a 2.0 GHz 4 Core laptop)

Kneip et al	Masselli and Zell	Proposed
1.3 µs	1.5 µs	**0.51 µs**

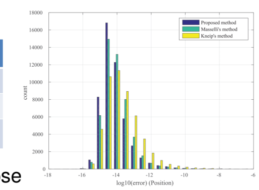
- Numerical accuracy (under nominal cond.s)

Method	Position Error
Gao et al.	6.36E-05
Kneip et al.	1.18E-05
Masselli and Zell	1.84E-08
Proposed	**1.66E-10**



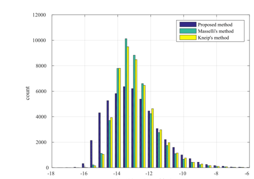
- Robustness 1: Points are almost collinear

Method	Position Error
Kneip et al.	1.42E-14
Masselli and Zell	7.24E-15
Proposed	**5.16E-15**



- Robustness 2: 2 bearing meas/nts are close

Method	Position Error
Kneip et al.	8.10E-14
Masselli and Zell	7.24E-14
Proposed	**6.73E-14**



Conclusions

- 3x faster than Kneip's et al.
- 3 orders of magnitude more accurate than Masselli and Zell under nominal conditions
- More robust than Masselli and Zell in close-to-singular conditions