Technical Report

Goal

Extract absolute pose from a cylinder object.

Motivation

State of the art related to absolute pose estimation is suitable for planar markers. However, most of the probes for CAS interventions are cylindrical. There is a necessity of pose estimation from cylindrical objects to improve the usability of the probes.

There are a few methods to extract the 3D pose from a cylinder.

PnP methods that englobe the P3P and others. This methodology relies on the absolute pose estimation of an 3D object to a 2D image. These methods are widely studied and could be a good starting point for the cylinder pose estimation [1,2,3,4].

Homography methods that estimates the cylinder pose by doing an approximation to a plane from points extracted in the cylinder surface. A homography is estimated by a 2D-2D relationship between a plane defined as the template in the cylinder represented in Cartesian coordinates and an image of the cylinder [5].

References

[1] Gao's P3P (classical P3P)

X.S. Gao, X.R. Hou, J. Tang, H.F. Cheng, Complete solution classification for the Perspective-Three-Point problem, IEEE Trans. PAMI, 25 (8) (2003), pp. 930-942

[2] Kneip's P3P (new P3P)

L. Kneip, D. Scaramuzza, R. Siegwart, A novel parametrization of the Perspective-Three-Point problem for a direct computation of absolute camera position and orientation, CVPR, 2011.

[3] Banno's P3P (new P3P)

Banno, A. (2018). A P3P problem solver representing all parameters as a linear combination. *Image and Vision Computing*, *70*, 55-62.

[4] PnP for cylinder probes

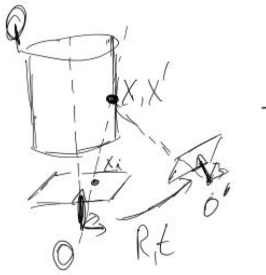
Jayarathne UL, McLeod AJ, Peters TM, Chen EC (2013) Robust intraoperative us probe tracking using a monocular endoscopic camera. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 363–370

[5] PnP + homography for cylinders (Imperial College of London)

Zhang, L., Ye, M., Chan, P. L., & Yang, G. Z. (2017). Real-time surgical tool tracking and pose estimation using a hybrid cylindrical marker. *International journal of computer assisted radiology and surgery*, *12*(6), 921-930.

Mathematical Formulation

DLT (6 points)



$\rightarrow X' = RX + t$ with X, X' being 3D coordinates expressed in O, O' references frames

$\rightarrow Q$ is the equation of cylinder in O reference frame such that

$$\begin{pmatrix} X^T & 1 \end{pmatrix} Q \begin{pmatrix} X \\ 1 \end{pmatrix} = 0 \quad \text{with } Q \simeq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -r^2 \end{pmatrix}$$

(1) Let $x_i = (x\ y\ z)^T$ be the image point where point X in Q is projected. It comes that

$X = \lambda x_i$ with λ being such that

$$\lambda^2 x^2 + \lambda^2 y^2 = r^2$$

$$\Leftrightarrow \lambda = \pm \sqrt{\frac{r^2}{x^2+y^2}} \Leftrightarrow \boxed{\lambda = \pm \frac{r}{\sqrt{x^2+y^2}}}$$

(2) Since $X' = RX + t$ and $X = \lambda x_i$ it comes that

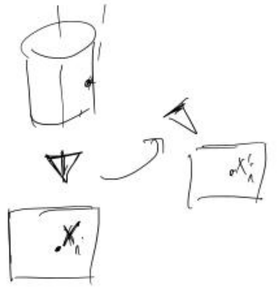
$$X' = \frac{r}{\sqrt{x^2+y^2}} R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + t \quad \text{(strict equality)}$$

Considering now equality up to scale in both sides of the equation it comes that

$$x_i' \sim R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \frac{\sqrt{x^2+y^2}}{r} t$$

$$\boxed{x_i' \sim \begin{pmatrix} \frac{1}{r} t & R \end{pmatrix} \hat{x}_i \quad \text{with } \hat{x}_i \sim \begin{pmatrix} \sqrt{x^2+y^2} \\ x \\ y \\ z \end{pmatrix}}$$

(3) In other words if x_i is the projection of a point in the cylinder in first view and x'_i in the second then the following holds



$$x'_i \sim \left(\tfrac{1}{r} t \; R\right) \hat{x}_i$$

with $\hat{x}_i \sim \begin{pmatrix} \sqrt{x^2+y^2} \\ x \\ y \\ z \end{pmatrix}$ being the lifted projective coordinates of x_i

(4) Properties of $\hat{H}_{3\times4}$

- since r is known, $\hat{H}$ has 6 DOFs (3 for R + 3 for t)
- the matrix $\hat{H} = (h_1\ h_2\ h_3\ h_4)$ can be estimated from a minimum of 3 point correspondences $x_i \leftrightarrow x'_i$ under the constraints that

$$\begin{cases} h_2^T h_3 = 0 \\ h_3^T h_4 = 0 \\ h_2^T h_4 = 0 \\ h_2^T h_2 - h_3^T h_3 = 0 \\ h_3^T h_3 - h_4^T h_4 = 0 \end{cases}$$

11 parameters
↳ 6 defined by correspondences
+
5 second order constraints

5. ~~Dry~~T estimation

- Use 6 correspondences and determine $\hat{H}$ in a DLT like manner (attention to normalization that can be tricky)
- Factorize by:
  - → Projecting $(l_2\ l_3\ l_4)$ in SO(3) to obtain R
  - → Compute scale factor λ such that $\lambda(l_2\ l_3\ l_4) - R = 0$
  - → Determine $\boxed{t = r \cdot \lambda \cdot l_1}$
- Note: I fear this might be very sensitive to noise.

Solver of 4 points designed by:

- Rotations restrictions.
- Kukolova's work.

Solver of 3 points designed by:

- Rotations restrictions.
- Francisco's work

Normalization scheme:

Normalization of the Points are made in 2 phases:

- Z coordinate:

  For Translation we subtract the mean from Z coordinate from points and subtract it to the Z coordinate of the points:

  $P(x,y,z) \quad = \quad Pm(x,y,z-\mathrm{mean}(z1, z2, z3, ..., zn))$

- X, Y coordinates:

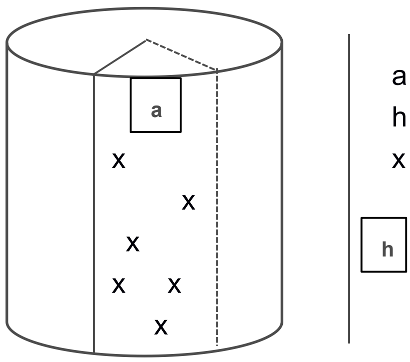
  Hartley's normalization is applied [6].

Synthetic Experiments

<u>1st Experiment: DLT 6points VS Solver 4 points VS Solver 3 points</u>

<u>Point Disposition</u>

Points distributed randomly accordingly to the following scheme:



a - point variation angle: 45º<a<90º
h - point variation height: 20mm<h<50mm
x - possible points

For simulation simplicity,6 points are generated. From these 6 points the first three and four are used for the 3-point Solver and 4-point Solver, respectively.

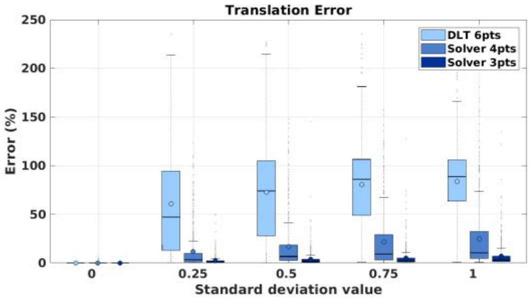
Camera Movement

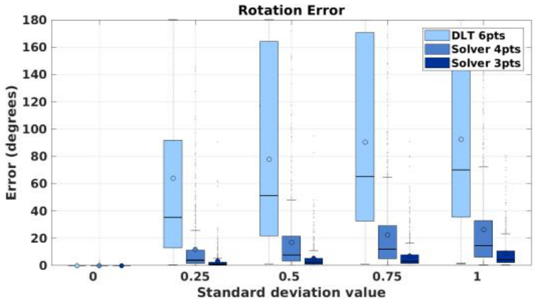
Angle Pitch: [-75º:75º]
Angle Roll: [-60º:60º]
Angle Yaw: [-90º:90º]
Translation Z: [50mm:200mm]

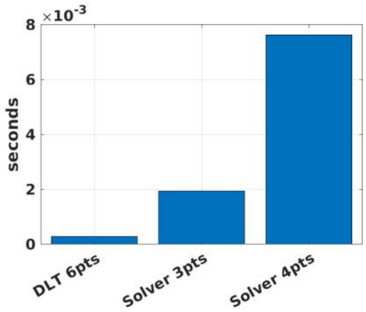
Noise applied to the image points

It is applied 5 levels of noise. These levels are the standard deviation of a normal distribution with mean = 0, i.e., 0σ, 0.25σ, 0.5σ, 0.75σ and 1σ.
Each one of the levels is multiplied with a random number generated under a normal distribution and the error is added to the pixel.

Results:



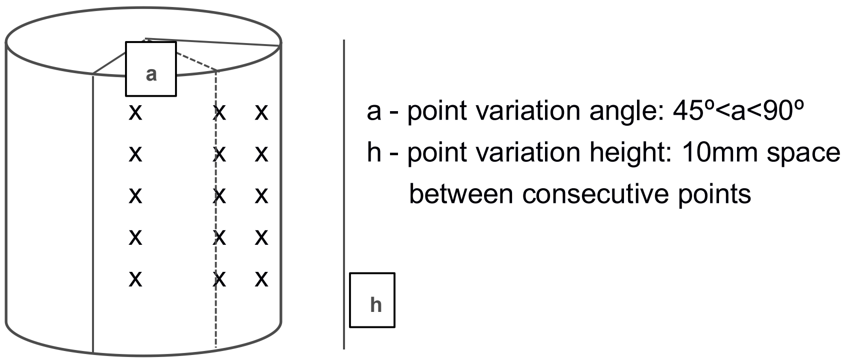




Conclusions:

- The faster solver is the DLT 6 points, however its accuracy is awful being in accordance with the theoretical propose which says that this solver is noisy.
- The best solver is the Solver 3points, wich uses 3 points. However, the correct solution is always equal to a P3P (Kneip's P3P and Gao's P3P).
- The Solver 4points that uses 4 points, is produced with an automatic code proposed by Kukolova. So its probable that if it generated as the Solver 3points (Francisco's code), the accuracy might improve being at level or better than the Solver 3points.

2nd Experiment: Solver 3points VS Kneip's P3P VS Gao's P3P VS Imperial approach (Homography + OpenCV iterative PnP Solver)



Three columns with 5 points are generated. Each column has an angle of 45º between the other. Points are distributed in a maximum of 90º.

From these 15 points set, a RANSAC is implemented with the Solver P3P, Kneip's P3P, Gao's P3P where the optimal solution is extracted. Also, for homography and for OpenCV iterative solver, all the 15 points are used.

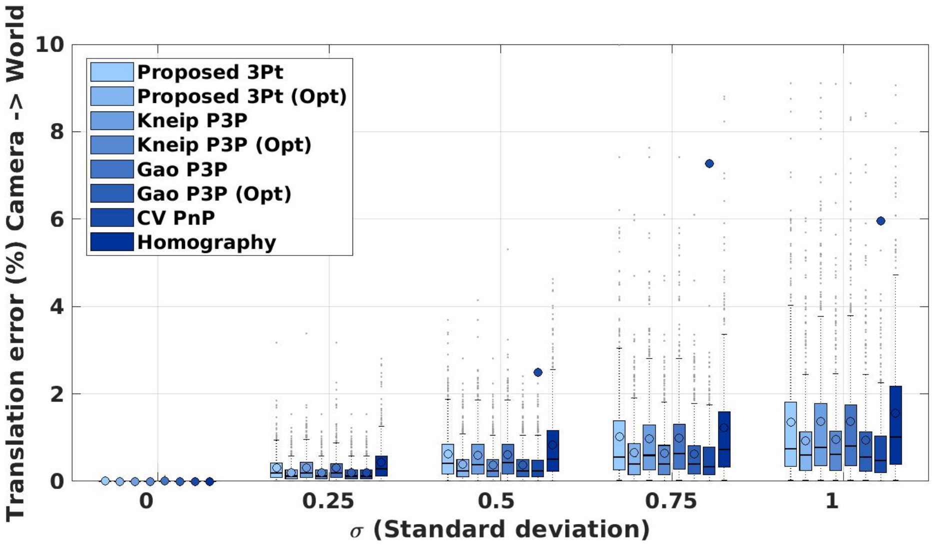
Camera Movement

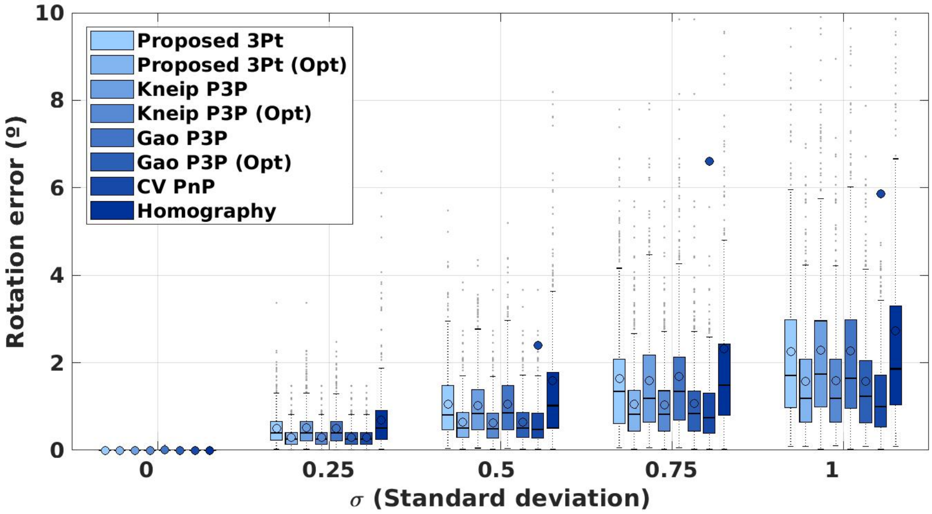
Angle Pitch: [-75º:75º]
Angle Roll: [-60º:60º]
Angle Yaw: [-90º:90º]
Translation Z: [50mm:250mm]

Noise applied to the image points

It is applied 5 levels of noise. These levels are the standard deviation of a normal distribution with mean = 0, i.e., 0**σ**, 0.25**σ**, 0.5**σ**, 0.75**σ** and 1**σ.**
Each one of the levels is multiplied with a random number generated under a normal distribution and the error is added to the pixel.

Results



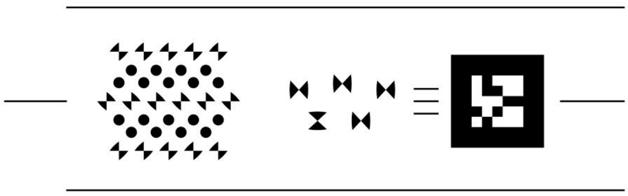


Conclusions:

- Homography has the worst accuracy, because in the Imperial paper it is not refined.
- All the initialization solutions for the 3 point solvers have the same solution, so the RANSAC extract most of the times that same solution and that fact reflects in similarities in the boxplots.
- All the optimization (Opt) solutions from the initializations are similar because all use the levenberg-marquardt algorithm (Solver P3P, Kneip's P3P, Gao's P3P and OpenCV PnP). The difference between the OpenCV and the others are in the inliers used in the optimization.

Real experiment:

The comparison between our Solver 3 points and the Imperial approach is done with the following picture, where a their marker is printed with a ground truth (innav marker) and a set of points which is used for the RANSAC-Solver 3points.



Point detection for our solver:

Those fancy triangular edges are designed by Bennet [7] in his paper: ChESS – Quick and Robust Detection of Chess-board Features where it is explained a extremely fast way to detect them.

Experiments description:

First, we glue the marker to a 3D printed cylinder which has an planar zone where the GT is placed. Those lines make it easier to place the marker in the same axis as the cylinder. A requirement valid for the Imperial marker.

Second we take a set of photos with the innav software and save them.

Lastly, we extract the Imperial pose with their code available online, we extract the GT from innav and determine the pose from Solver 3points and compare the results.

I made a few experiments but nothing was conclusive, so I will not put any result or extract any conclusion from this section.