# Camera-based Orientation Estimation with Natural Visual Features

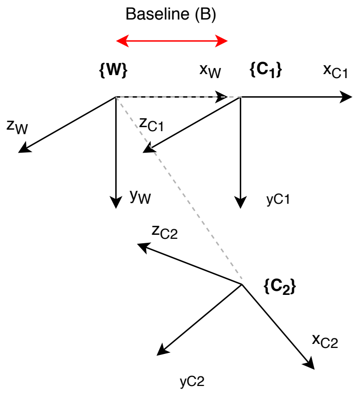
A very lazy and drafty summary

Mariana Martins

July 2019

## 1 System information

There is a dependency relationship between the translation and rotation of our system's camera.



The transformation from the World to the Camera View 1 is

$$ {}^{W}_{C_1}T = \begin{bmatrix} I & -\mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \tag{1}$$

and the transformation from the World to the Camera View 2 is

$$ {}^{W}_{C_2}T = \begin{bmatrix} R & -\mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2}$$

Hence, the transformation from Camera View 1 to Camera View 2 is

$$ {}^{C_1}_{C_2}T = {}^{W}_{C_2}T\,{}^{C_1}_{W}T = {}^{W}_{C_2}T\,{}^{W}_{C_1}T^{-1} = \begin{bmatrix} R & -\mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} I & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} R & R\mathbf{b} - \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3}$$

Thus, the translation is a function of the rotation and the baseline on the following way

$$ \mathbf{t}(R, \mathbf{b}) = R\mathbf{b} - \mathbf{b}. \tag{4}$$

Besides that, the camera has no depth information.

# 2 Potential Methods

The objective is to find the method that gives us the most accurate pose and completes that task faster. So the ones found most promising are put to test.

## 2.1 Pose Estimation

### 2.1.1 Orthogonal Procrustes Problem (OPPr)

This method works by finding the best fitting for our two point clouds, $M_1$ and $M_2$, through a rotation matrix, R, by minimizing

$$\|M_1 - RM_2\|^2 . \tag{5}$$

To obtain the point clouds $M_1$ and $M_2$, the corresponding image points, $m_1$ and $m_2$, are projected into a sphere.
Through the Frobenius norm, (5) can be expanded to

$$\|M_1 - RM_2\|^2 = \operatorname{trace}(M_1^T M_1 + M_2^T M_2) - 2\operatorname{trace}(M_2^T M_1 R). \tag{6}$$

So minimizing (5) with respect to $R$ is equivalent to maximizing the second term of (6). By applying a Singular Value decomposition (SVD) to $M_2^T M_1$, the latter term can be further simplified into

$$\operatorname{trace}(M_2^T M_1 R) = \operatorname{trace}(U\Sigma V^T R) = \operatorname{trace}(\Sigma V^T R U) = \operatorname{trace}(\Sigma H) = \sum_{i=1}^{N} \sigma_i h_{ii}. \tag{7}$$

The singular values of $\sigma_i$ are all non-negative, and so the expression becomes maximum when $h_{ii} = 1$ for $i = 1, 2, ..., N$, since $H$, a product of orthogonal matrices, is an orthogonal matrix itself, thus having its maximal value when $H = I$. This results in $I = V^T RU$, and so $R = VU^T$.

Here, the translation is not taken into consideration.

### 2.1.2 Minimization of the Back Projection Error (MBPE)

This method is a bundle adjustment of the previous one. It uses the rotation matrix obtained with OPPr and it tries to tune it to obtain a rotation and a translation dependent on the former, through

$$\min_{R, Z_{e11}, \ldots, Z_{e1N}} \sum_{i=1}^{N} [(u_{e1i} - u_{1i})^2 + (u_{e2i} - u_{2i})^2 + (v_{e1i} - v_{1i})^2 + (v_{e2i} - v_{2i})^2]$$

$$\text{with } Z_{e1init} = \frac{1}{\sqrt{u_{1i}^2 + v_{1i}^2 + 1}} \text{ and } R_{init} = R_{oppr},$$

where $u_{1i}$ and $v_{1i}$ are the image points of the Camera View 1, $u_{2i}$ and $v_{2i}$ are the image points of the Camera View 2, $u_{e1i}$, $v_{e1i}$, $u_{e2i}$ and $v_{e2i}$ are the corresponding image points

estimations and $Z_{e1i}$ is the depth of the Camera View 1.
The image point estimations are obtained the following way

$$\mathbf{m_{e1}} = \frac{KR^T(Z_{e2}K^{-1}\mathbf{m_2}) - R^T t)}{Z_{e1}}$$
$$\mathbf{m_{e2}} = \frac{KR(Z_{e1}K^{-1}\mathbf{m_1}) + t)}{Z_{e2}}.$$

The depth is initialized by projecting the image points in a sphere.

### 2.1.3 Epipolar Geometry approaches

The rotation and translation may be obtained through the essential matrix, $E$,

$$E = [\mathbf{t}]_\times R \tag{8}$$

which itself is given by the fundamental matrix, $F$,

$$E = K^T F K. \tag{9}$$

There is a relationship between the images points of Camera View 1 and 2 given by

$$\widetilde{\mathbf{m_2}}^T F \widetilde{\mathbf{m_1}} = 0. \tag{10}$$

This previous equation can be written as

$$\mathbf{u}_i^T \mathbf{f} = 0, \tag{11}$$

where

$$\mathbf{u}_i^T = [u_i u_i', v_i u_i', u_i', u_i v_i', v_i v_i', v_i' u_i, v_i.1]^T$$
$$\mathbf{f} = [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]^T.$$

For n point matches,

$$\mathbf{U}_n \mathbf{f} = 0$$
$$\mathbf{U}_n = [\mathbf{u}_1, ..., \mathbf{u}_n].$$

The fundamental matrix has a rank-2 constraint, $det(F) = 0$. Using this constraint on the computation of $F$ reduces noise.

Using $[\mathbf{t}]_\times R$ when building the fundamental matrix enforces this rank-2 constraint.

A solution for finding the fundamental matrix can be obtained using 7 point matches due to the rank-2, but considering the previous methods can use less point matches to obtain an estimation, there is no use in testing that as it doesn't seem to present better results than the other methods.

Normalizing input data and putting the centroid at the origin seems to improve the results.

### Linear Solution or Norm-8 point (the fastest)

$$\min_F \sum_i (\widetilde{\mathbf{m_i}}'^T F \widetilde{\mathbf{m_i}})^2 \tag{12}$$

$$\min_f \|\mathbf{U_n f}\|^2 \tag{13}$$

So that the solution for this is not $\mathbf{f} = \mathbf{0}$, we apply a constraint, $\|\mathbf{f}\| = 1$. Transforming it into an unconstrained minimization problem through Lagrange multipliers, we obtain

$$\min_{\mathbf{f}} \mathcal{F}(\mathbf{f}, \lambda), \tag{14}$$

where

$$\mathcal{F}(\mathbf{f}, \lambda) = \|\mathbf{U}_n\mathbf{f}\|^2 + \lambda(1 - \|\mathbf{f}\|^2) \tag{15}$$

and $\lambda$ is a Lagrange multiplier. Requiring the first derivative of $\mathcal{F}(\mathbf{f}, \lambda)$ with respect to f to be zero, we have

$$\mathbf{U}_n^T\mathbf{U}_n\mathbf{f} = \lambda\mathbf{f} \tag{16}$$

To minimize $\mathcal{F}(\mathbf{f}, \lambda)$ the solution is the unit eigenvector of $\mathbf{U}_n^T\mathbf{U}_n$ associated with the smallest value of $\lambda_i$, $\lambda_9$.

This still has a lot of noise because it doesn't impose rank-2 constraint. Hence, using the calculated $F$ matrix we can minimize the Frobenius norm of $F - \hat{F}$ by having $\hat{F} = U\hat{S}V^T$ with $\hat{S} = diag(\sigma_1, \sigma_2, 0)$.

### Geometric Interpretation (most physically meaningful)

With the linear criterion the quantity we are minimizing is not physically meaningful. Physically meaningful quantity should be something measured in the image plane, because the available information (2D points) are extracted from images. One such quantity is the distance from a point $\mathbf{m}'_i$ to its corresponding epipolar line $\mathbf{l}'_i = F\widetilde{\mathbf{m}}_i = [l'_1, l'_2, l'_3]^T$ given by

$$d(\mathbf{m}'_i, \mathbf{l}'_i) = \frac{\widetilde{\mathbf{m}}_i^{T\prime}\mathbf{l}'_i}{\sqrt{l_1^{2\prime} + l_2^{2\prime}}}. \tag{17}$$

Hence, one possibility is,

$$\min_R \sum_i^N d^2(\mathbf{m}'_i, F\widetilde{\mathbf{m}}_i) + d^2(\mathbf{m}_i, F^T\widetilde{\mathbf{m}}'_i) \tag{18}$$

which also forces the rank-2 constraint by doing $[\mathbf{t}]_\times R$. Normalizing the points and bringing their centroid to the center might yield better results.

### Gradient-based Technique or Sampson distance (the best)

Minimizing $(\widetilde{\mathbf{m_i}}'^T F \widetilde{\mathbf{m_i}})^2$ doesn't yield a good result because the variance is not the same. The least-squares technique produces an optimal solution if each term has the same variance. So one possibility is

$$\min_R \sum_i^N \frac{(\widetilde{\mathbf{m_i}}'^T F \widetilde{\mathbf{m_i}})^2}{\sigma_i^2}, \tag{19}$$

where $\sigma_i$ is the variance given by

$$\sigma_i^2 = \sigma[l_1^2 + l_2^2 + l_1^{2'} + l_2^{2'}]. \tag{20}$$

## 2.2 Inlier selection

### 2.2.1 RANSAC

Given that OPPr is the simplest and fastest of the methods, giving a pretty good initial estimation, we can use it for RANSAC by doing,

- Selection of a random sample (maybe inliers)
- Fit of a model to that sample
- Test the model for the rest of the points, the ones that fit are also inliers
- If (maybe inliers + also inliers) are enough points the model is accepted
- Try all this again for as many tries as desired, gather the best

### 2.2.2 LMedS

Again we can use OPPr and try to figure the 3 euler angles out.

- Selection of a random sample
- Fit model to that sample
- Do this for several samples
- Choose the one with least median

RANSAC requires an error threshold to consider a point as inlier, but it is cheaper.

If there is a large set of images of the same type of scenes to be processed, one can first apply LMedS to one pair of the images in order to find an appropriate threshold, and then apply RANSAC to the remaining images because it is cheaper.

### 2.2.3 Reject image sections

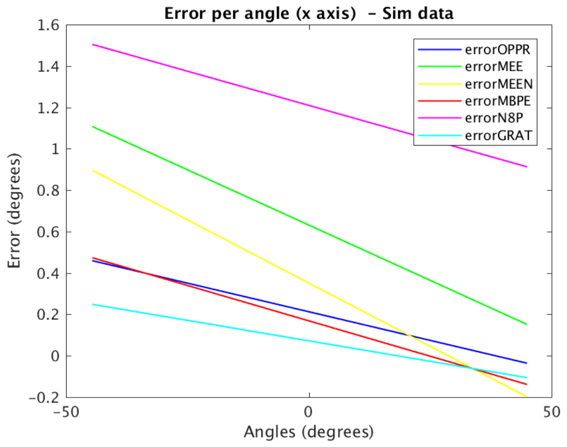
Points in the image that correspond to bigger depths suffer less translation effects. Hence, using RANSAC with OPPr will yield point sets that are located in bigger depths and thus determining the rotation through them may be easier given they are closer to a local minimum. This leaves just a translation adjustment to be done.

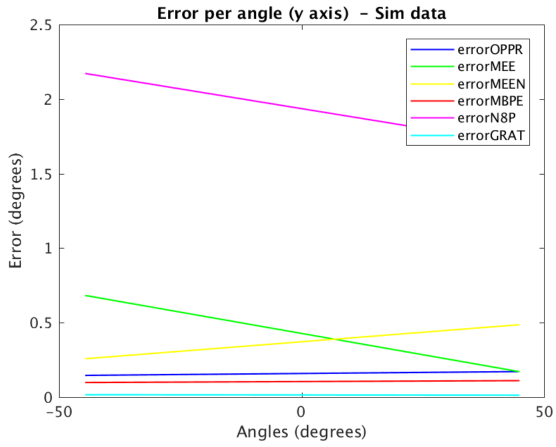
# 3 Intermidiate results

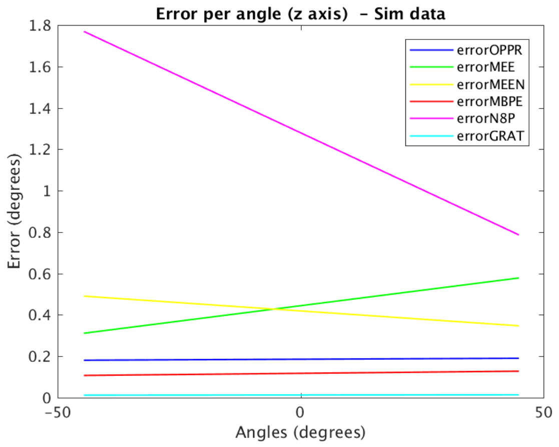
The following methods were tested,

- Orthogonal Procrustes Problem - OPPR
- Minimization of the Back Projection Error - MBPE
- Norm 8-point with Frobenius norm adjustment - N8P
- A geometric interpretation by minimizing the distance of the points to the epipolar lines without normalization - MEE
- A geometric interpretation by minimizing the distance of the points to the epipolar lines with normalization - MEEN
- A gradient based technique - GRAT.

For a small distance to the camera **(24cm)**, the results were the following,



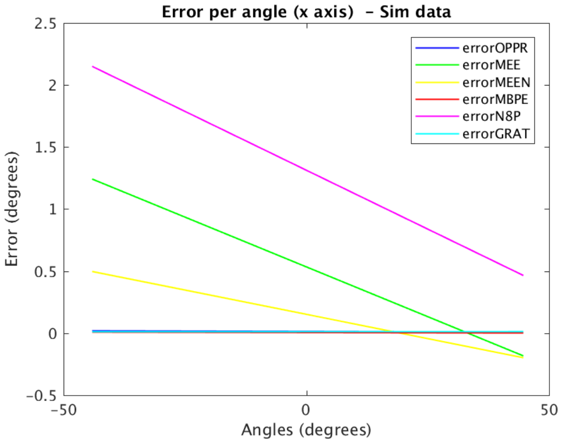


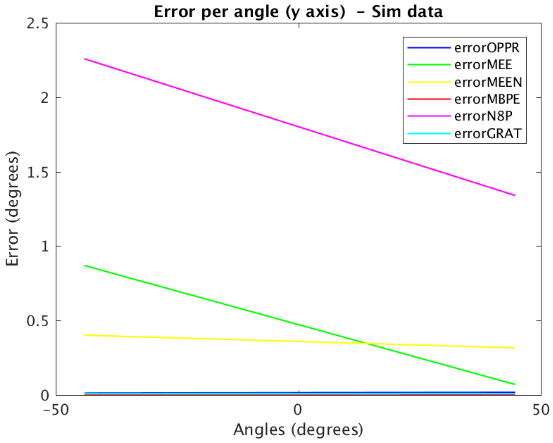


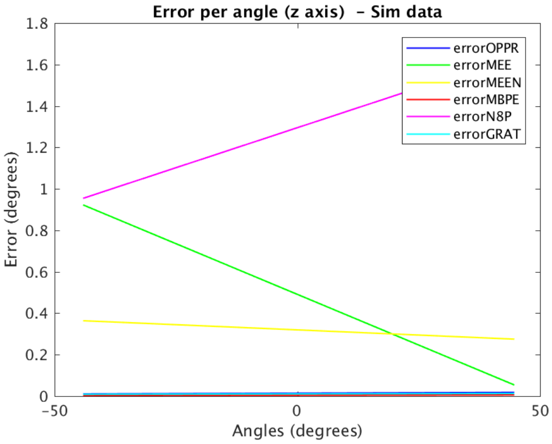
| Rotation Axis | X | | Y | | Z | |
|---|---|---|---|---|---|---|
| | Mean | Variance | Mean | Variance | Mean | Variance |
| OPPR | 0.217148 | 0.082592 | 0.158344 | 0.007315 | 0.185296 | 0.010773 |
| MEE | 0.640299 | 0.332844 | 0.432324 | 0.785632 | 0.441461 | 0.276032 |
| MEEN | 0.361375 | 0.182559 | 0.369072 | 0.214085 | 0.420247 | 0.063756 |
| MBPE | 0.174322 | 0.093475 | 0.104173 | 0.003768 | 0.117125 | 0.004677 |
| N8P | 1.215141 | 1.659817 | 1.940488 | 1.752937 | 1.289076 | 1.618723 |
| GRAT | 0.075175 | 0.060411 | 0.014437 | 0.002147 | 0.012692 | 0.001886 |

and the **best method was GRAT with 0.034102 degrees of mean error**. The error mean and variance for rotations around each axis is on the following table.

For a bigger distance **(5m)**, the results were the following,







and the **best method was MBPE with 0.004812 degrees of mean error**. The error mean and variance for rotations around each axis is on the following table.

These results show that for bigger distances (where translation is less relevant) MBPE works better than GRAT.

| Rotation Axis | X | | Y | | Z | |
|---|---|---|---|---|---|---|
| | Mean | Variance | Mean | Variance | Mean | Variance |
| OPPR | 0.014184 | 0.000133 | 0.014690 | 0.000193 | 0.013563 | 0.000250 |
| MEE | 0.572221 | 0.437065 | 0.493935 | 0.957411 | 0.513986 | 0.485687 |
| MEEN | 0.170275 | 0.063387 | 0.361200 | 0.069951 | 0.321769 | 0.043277 |
| MBPE | 0.004914 | 0.000025 | 0.005366 | 0.000028 | 0.004156 | 0.000014 |
| N8P | 1.358318 | 2.647271 | 1.826605 | 3.089146 | 1.276812 | 2.581466 |
| GRAT | 0.011956 | 0.000526 | 0.009642 | 0.000360 | 0.010321 | 0.000677 |

# 4 ToDo

1. See the effect of rejecting imagesections with RANSAC
2. Test LMeDs with RANSAC for choosing matches
3. Implement a new way of getting ground truth
4. Test methods in real world
5. Test with the sensor as initialization input
6. Implement in the eye model