

Camera-based Orientation Estimation with Natural Visual Features

Mariana Martins
June 2019

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
4. Experiments
5. Results
6. Other ideas
7. Future work

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
4. Experiments
5. Results
6. Other ideas
7. Future work

Motivation



Knowing the eye's orientation is crucial

IMU presents significant drift

We want a rotational error < 0.5 deg

Camera will be useful further along

Problem and Objective



How to determine the eye's orientation as most accurately as possible?

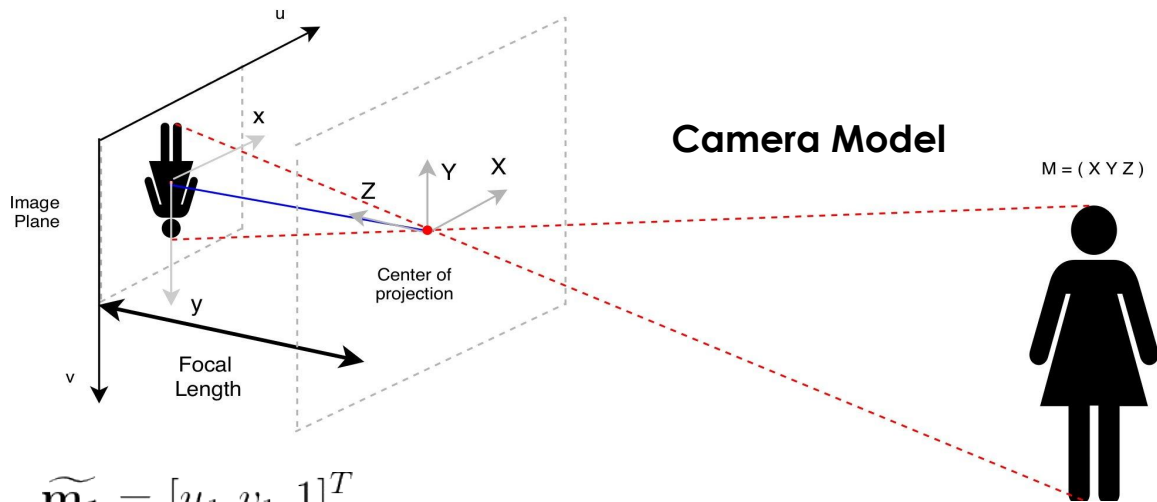
...knowing there is translation associated to the eye movement

Put to test several camera-based approaches to determine the eye's orientation

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
4. Experiments
5. Results
6. Other ideas
7. Future work

3 Potential Approaches: Background



$$\tilde{\mathbf{m}}_1 = [u_1 \ v_1 \ 1]^T$$

$$K^{-1}\tilde{\mathbf{m}}_1 = [x_1 \ y_1 \ 1]^T$$

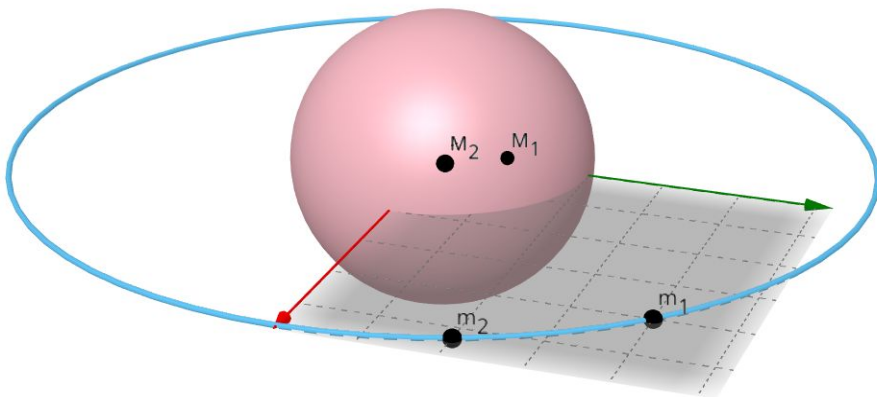
$$Z_1 K^{-1}\tilde{\mathbf{m}}_1 = [X_1 \ Y_1 \ Z_1]^T = M_1$$

$$Z_1 \tilde{\mathbf{m}}_1 = K M_1$$

3 Potential Approaches: Orthogonal Procrustes Problem

$$\min \|M_1 - RM_2\|^2$$

But... no depth information...



Ignoring the influence of \mathbf{t}

Camera Model

$$Z_1 \mathbf{m}_1 = K M_1$$

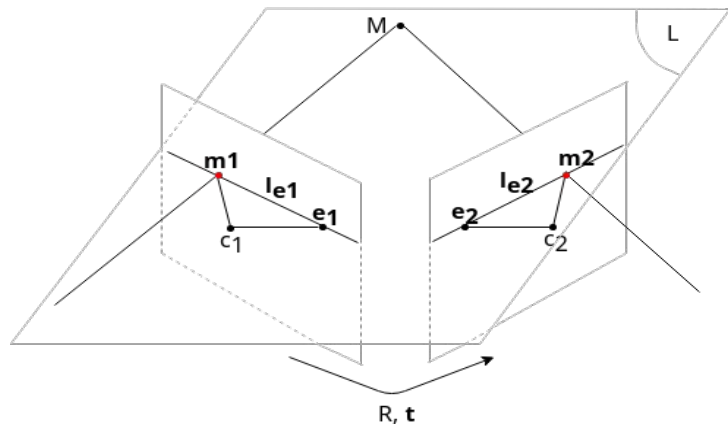
$$Z_2 \mathbf{m}_2 = K M_2$$

$$M_2 = R M_1 + \mathbf{t}$$

$$\|(Z_1 x_1, Z_1 y_1, Z_1)\| = \text{radius}$$

$$Z_1 = \frac{\text{radius}}{\sqrt{x_1^2 + y_1^2 + 1}}$$

3 Potential Approaches: Epipolar Geometry



Min 7 needed point matches

Must have translation

Camera Model

$$Z_1 \mathbf{m}_1 = K M_1$$

$$Z_2 \mathbf{m}_2 = K M_2$$

$$M_2 = R M_1 + \mathbf{t}$$

$$\tilde{\mathbf{m}}_2^T F \tilde{\mathbf{m}}_1 = 0 \longrightarrow E = K^T F K \longrightarrow E = [\mathbf{t}]_{\times} R$$

3 Potential Approaches: Minimization of the Backprojection Error

$$Z_{e2}m_{e2} = \underbrace{KR(Z_{e1}K^{-1}m_1)}_{M_{e2}} + \mathbf{t}$$

$\mathbf{t} = (R - I)\mathbf{b}$

Camera Model

$$Z_1\mathbf{m}_1 = KM_1$$

$$Z_2\mathbf{m}_2 = KM_2$$

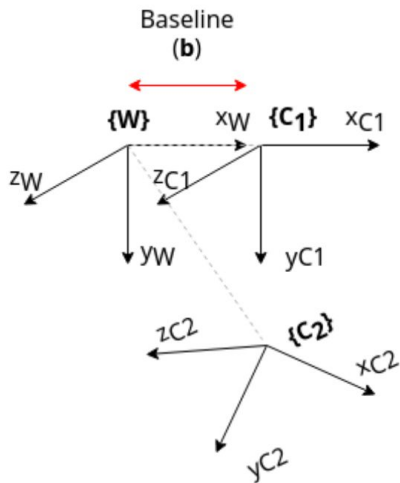
$$M_2 = RM_1 + \mathbf{t}$$

$$\min_{R, Z_{e1}, \dots, Z_{eN}} \sum_{i=1}^N [(u_{e1i} - u_{1i})^2 + (u_{e2i} - u_{2i})^2 + (v_{e1i} - v_{1i})^2 + (v_{e2i} - v_{2i})^2]$$

$$R_{init} = R_{procrustes}$$

$$Z_{e1i_{init}} = \frac{1}{\sqrt{u_{1i}^2 + v_{1i}^2 + 1}}$$

3 Potential Approaches: Minimization of the Backprojection Error



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

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

$${}^{C_1} C_2 T = {}^W C_2 T {}^W C_1 T = \begin{bmatrix} R & R\mathbf{b} - \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}$$

$${}^{C_1} C_2 T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\mathbf{t} = \mathbf{b}(R - I)$$

Camera Model

$$Z_1 \mathbf{m}_1 = K M_1$$

$$Z_2 \mathbf{m}_2 = K M_2$$

$$M_2 = R M_1 + \mathbf{t}$$

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
- 3. System Setup**
4. Experiments
5. Results
6. Other ideas
7. Future work

System Setup: Simulation

M random angles in a degree range are generated

For each axis

For each angle,

N random point matches within the image plane frame are generated

The camera is rotated around the given angle

The 3 approaches estimate the orientation

The results are compared with the ground truth

$$error = \|eul_{ground\ truth} - eul_{estimated}\|$$

System Setup: Real data



Take pictures before and after rotation using an uEye LE camera. The rotations were done **per each axis** by hand.



Positional error < 0.3mm
Rotational error < 0.05°

Meanwhile a motion capture system (Optitrack) tracks the camera's position and orientation

System Setup: Real data

From the Optitrack data

Ground truth (in euler angles) for each rotation made is obtained

From the pictures,

Features and matches obtained

Inlier matches selected with RANSAC

The 3 approaches estimate the orientation



The results are compared with the ground truth

$$error = \|eul_{ground\ truth} - eul_{estimated}\|$$

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
- 4. Experiments**
5. Results
6. Other ideas
7. Future work

Experiments: Simulation 1 & 2 / Real data 1 & 2

Maximum depth: 24 cm
 Point matches: 20
 Image Dimension: [2056 1542]
 No noise
 Real camera intrinsics matrix

100 rotations per each axis
 ... between 45° and -45°

Maximum depth: 5.77 m
 Point matches: 20
 Image Dimension: [2056 1542]
 No noise
 Real camera intrinsics matrix

100 rotations per each axis
 ... between 45° and -45°



Maximum depth: 24 cm
 Point matches: 20
 Image Dimension: [2056 1542]

91 rotations on x axis
 131 rotations on y axis
 91 rotations on z axis
 ... between 45° and -45°



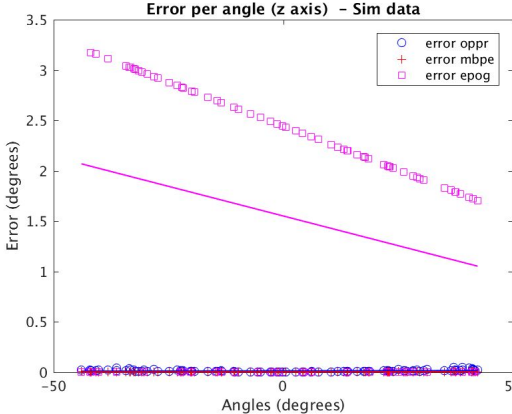
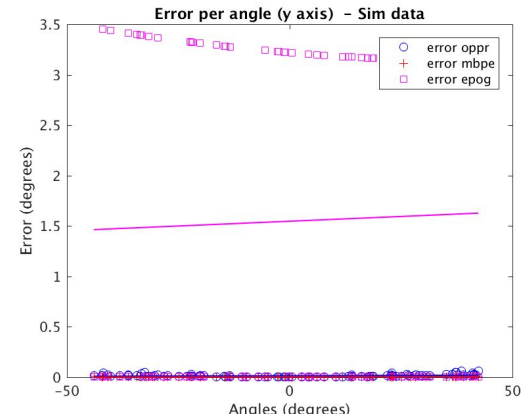
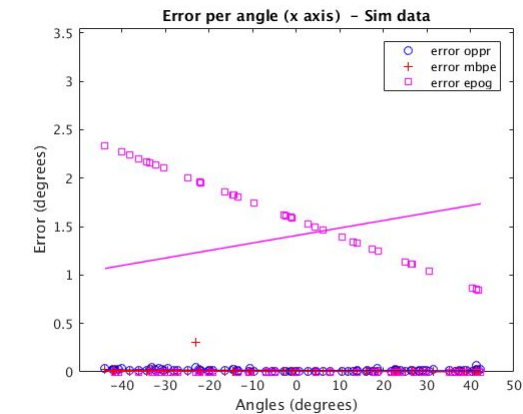
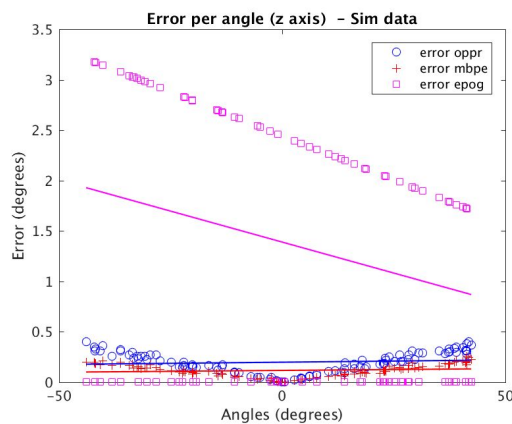
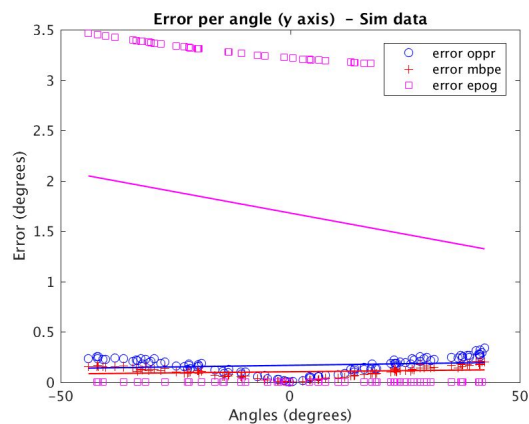
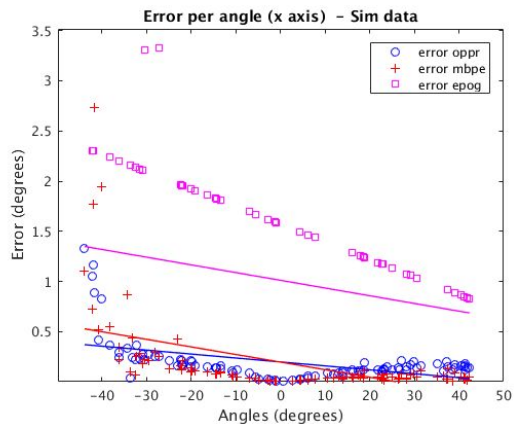
Maximum depth: 5.77 m
 Point matches: 20
 Image Dimension: [2056 1542]

49 rotations on x axis
 182 rotations on y axis
 82 rotations on z axis
 ... between 45° and -45°

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
4. Experiments
- 5. Results**
6. Other ideas
7. Future work

Results: Simulation 1 & 2



Simulation 1
MBPE 0.1339°
mean error

OPPR 0.1862°
mean error

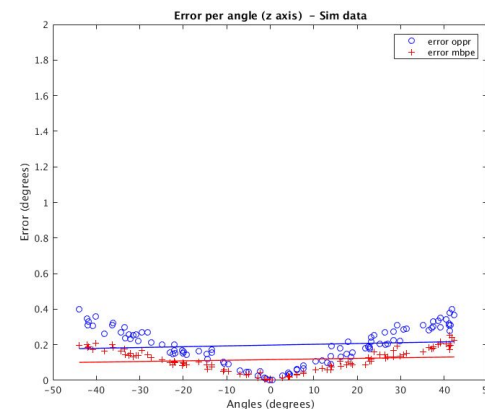
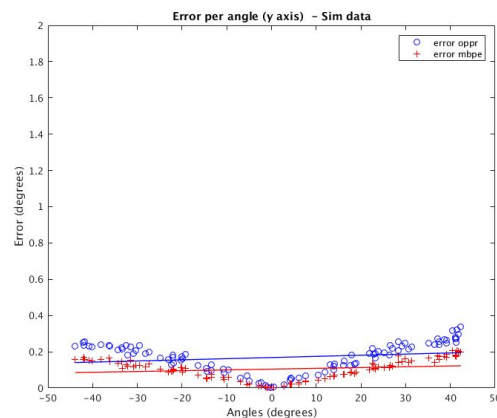
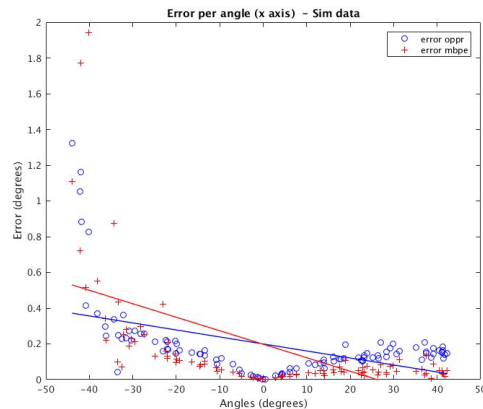
D = 0.24cm

Simulation 2
MBPE 0.0037°
mean error

OPPR 0.0143°
mean error

D = 5.77m

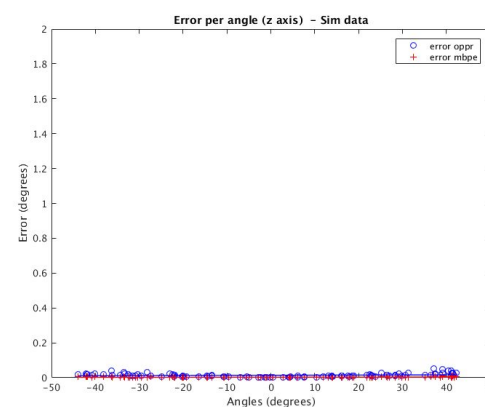
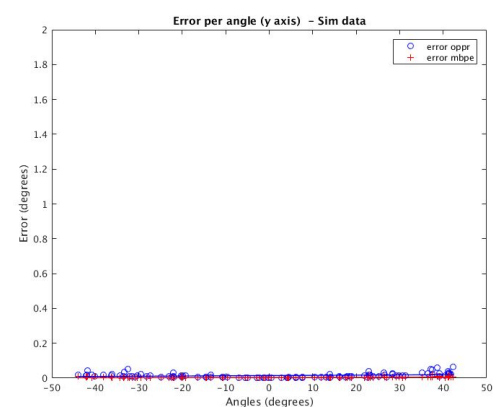
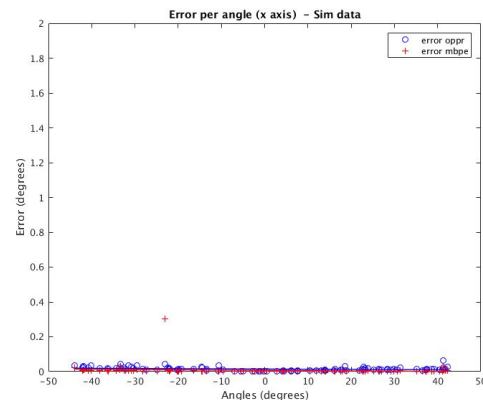
Results: Simulation 1 & 2



Simulation 1
MBPE 0.1339°
mean error

OPPR 0.1862°
mean error

$D = 0.24\text{cm}$

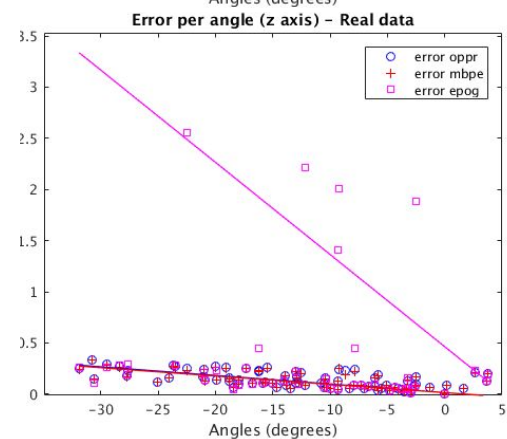
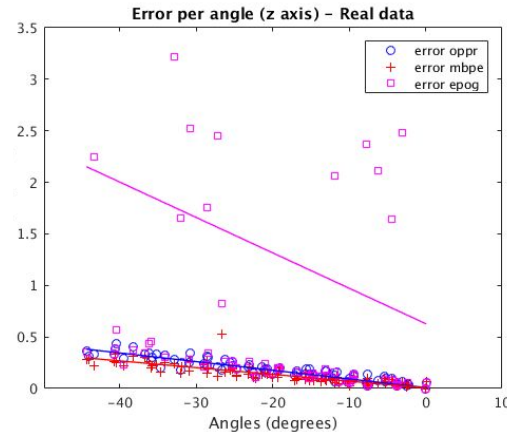
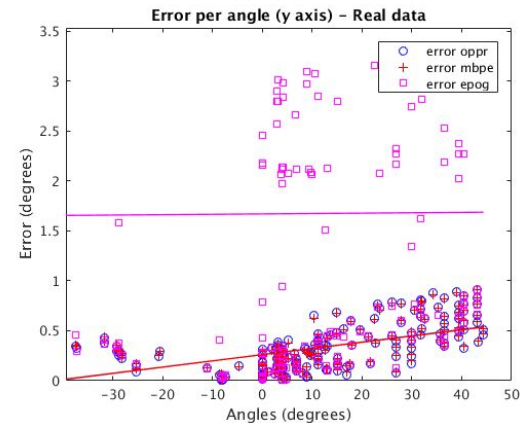
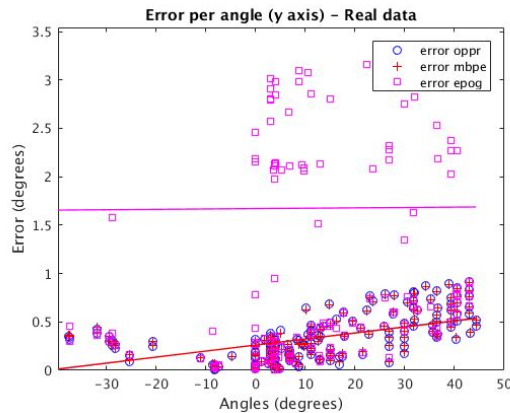
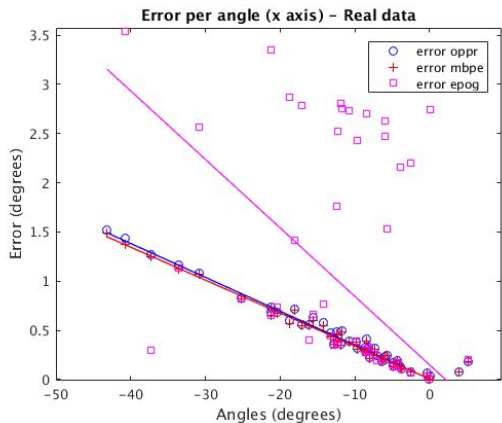
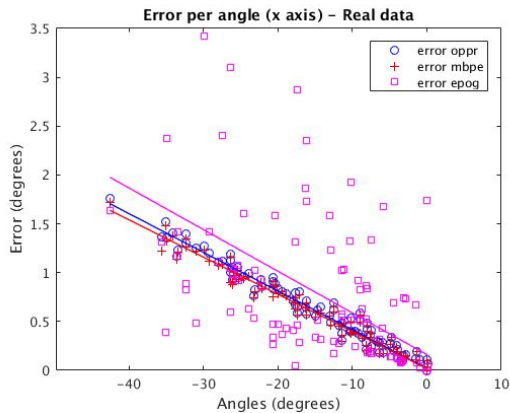


Simulation 2
MBPE 0.0037°
mean error

OPPR 0.0143°
mean error

$D = 5.77\text{m}$

Results: Real data 1 & 2



Real data 1
MBPE 0.0446°
mean error

OPPR 0.0576°
mean error

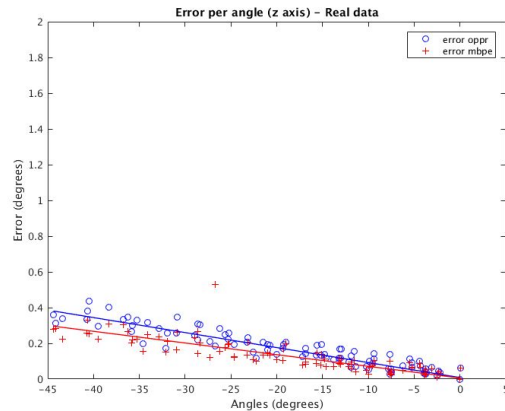
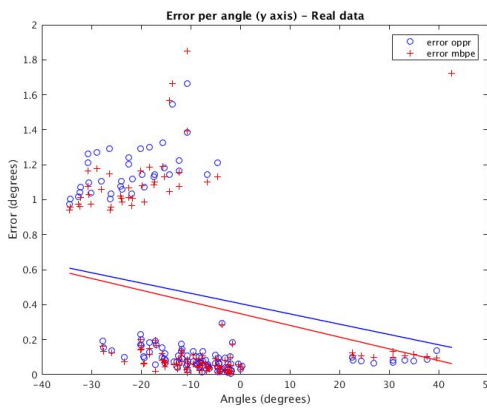
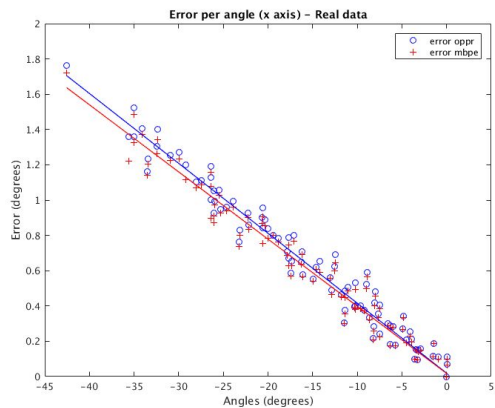
D = 24cm

Real data 2
MBPE 0.0451°
mean error

OPPR 0.0463°
mean error

D = 5.77m

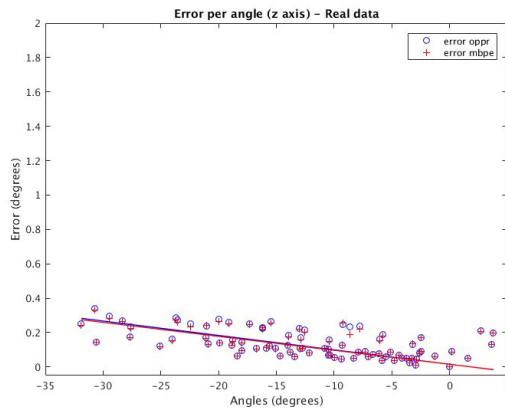
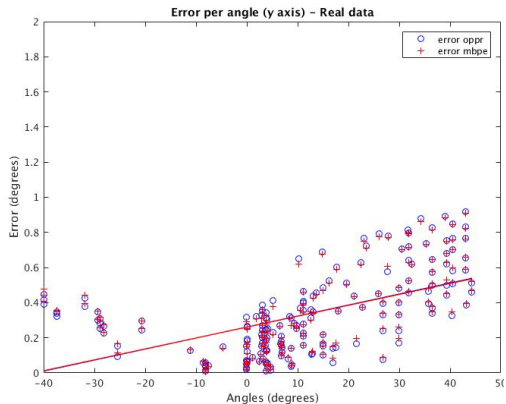
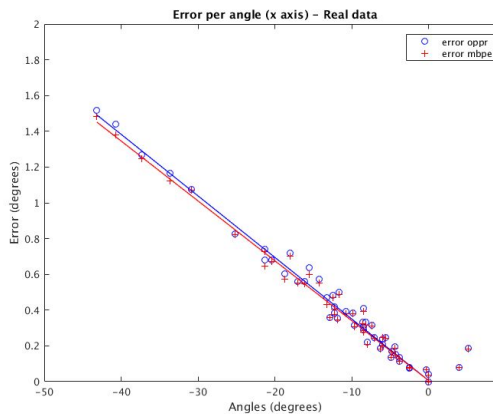
Results: Real data 1 & 2



Real data 1
MBPE 0.0446°
mean error

OPPR 0.0576°
mean error

D = 24cm



Real data 2
MBPE 0.0451°
mean error

OPPR 0.0463°
mean error

D = 5.77m

Results: Time

- **Gather feature matches**
(feature detection + matching + matches inlier selection)
Avg time: 7.1ms
- **Orientation Estimation**
 - **Orthogonal Procrustes Problem**
Avg time: 750ns
 - **Minimization of Back Projection Error**
Avg time: 5.6ms
 - **Epipolar Geometry**
Avg time: 1.1ms

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
4. Experiments
5. Results
6. Other ideas
7. Future work

Other ideas

- Select points whose translation won't affect the estimation as much
- Use epipolar error to estimate rotation instead

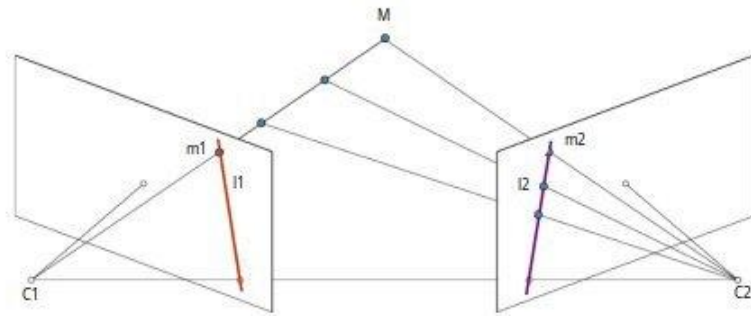
$$\tilde{\mathbf{m}}_1^T \tilde{\mathbf{l}}_1 = 0$$

$$\tilde{\mathbf{m}}_2^T \tilde{\mathbf{l}}_2 = 0$$

$$\tilde{\mathbf{m}}_2^T F \tilde{\mathbf{m}}_1 = 0$$

$$r_{2i} = \frac{\tilde{\mathbf{m}}_{2i}^T F \tilde{\mathbf{m}}_{1i}}{\|I_{2 \times 3} F \tilde{\mathbf{m}}_{1i}\|}$$

$$\min_R \sum_{i=1}^N r_{1i}^2 + r_{2i}^2$$



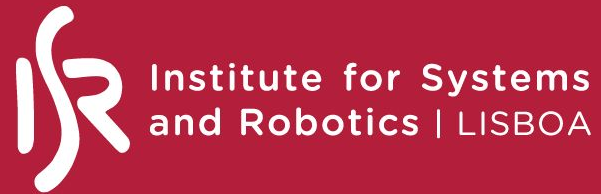
- OpenCV camera pose estimation with chessboard
- **More suggestions ..?**

Outline

1. Motivation, Problem and Objective
2. 3 Potential Approaches
3. System Setup
4. Experiments
5. Results
6. Other ideas
7. Future work

Future Work

1. Use a better experimental setup for taking the ground truth
2. Study how the IMU can complement the orientation estimation
3. Implement the final solution on the actual robotic eye



Thank you.

