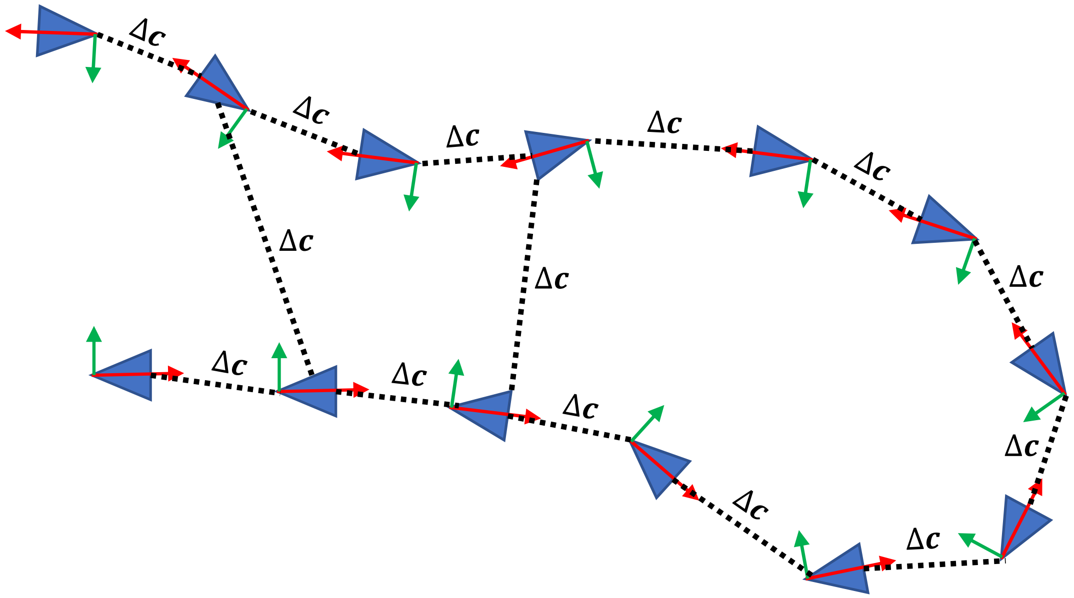
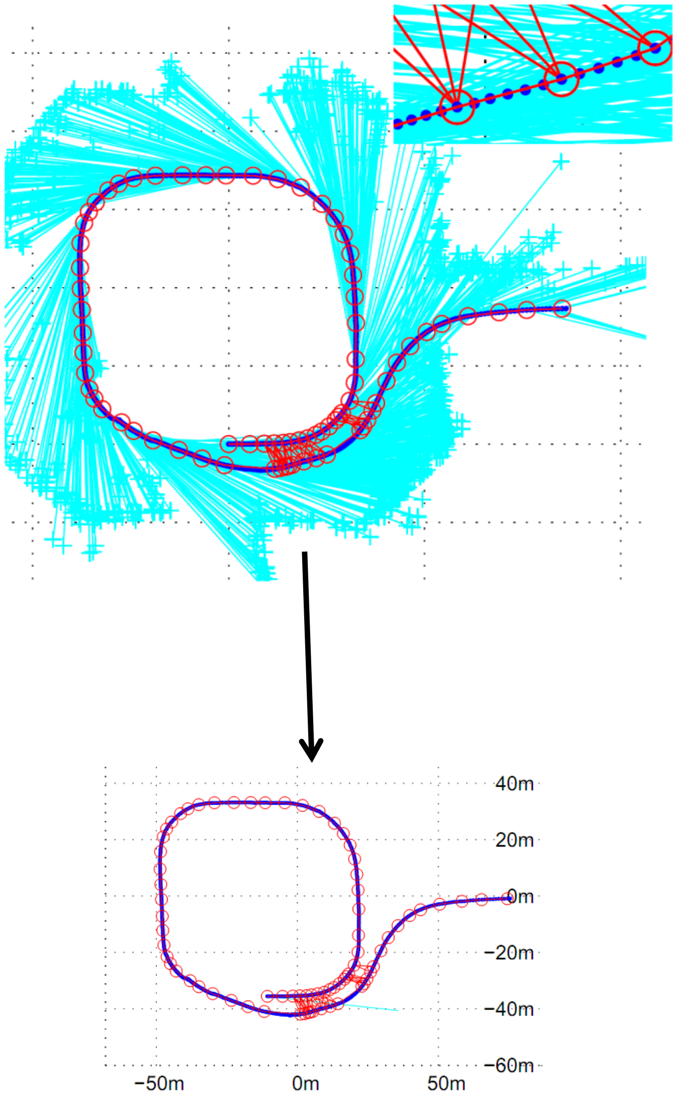
Project – Phase 6

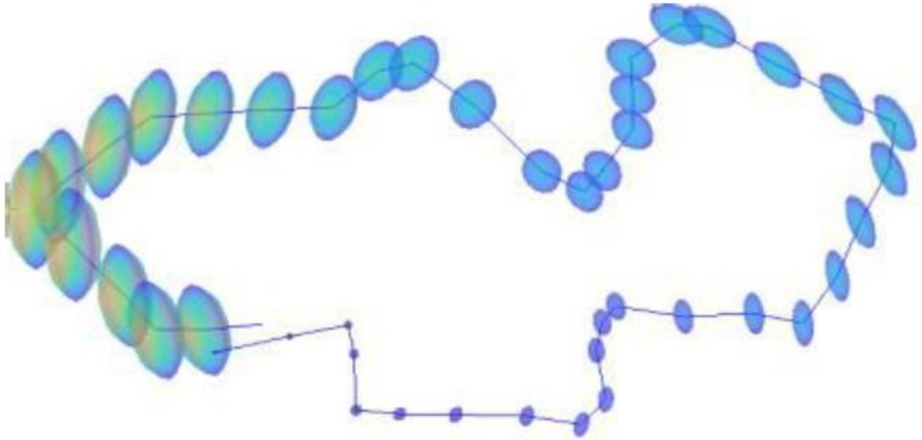
- Pose Graph - Large scale motion estimation

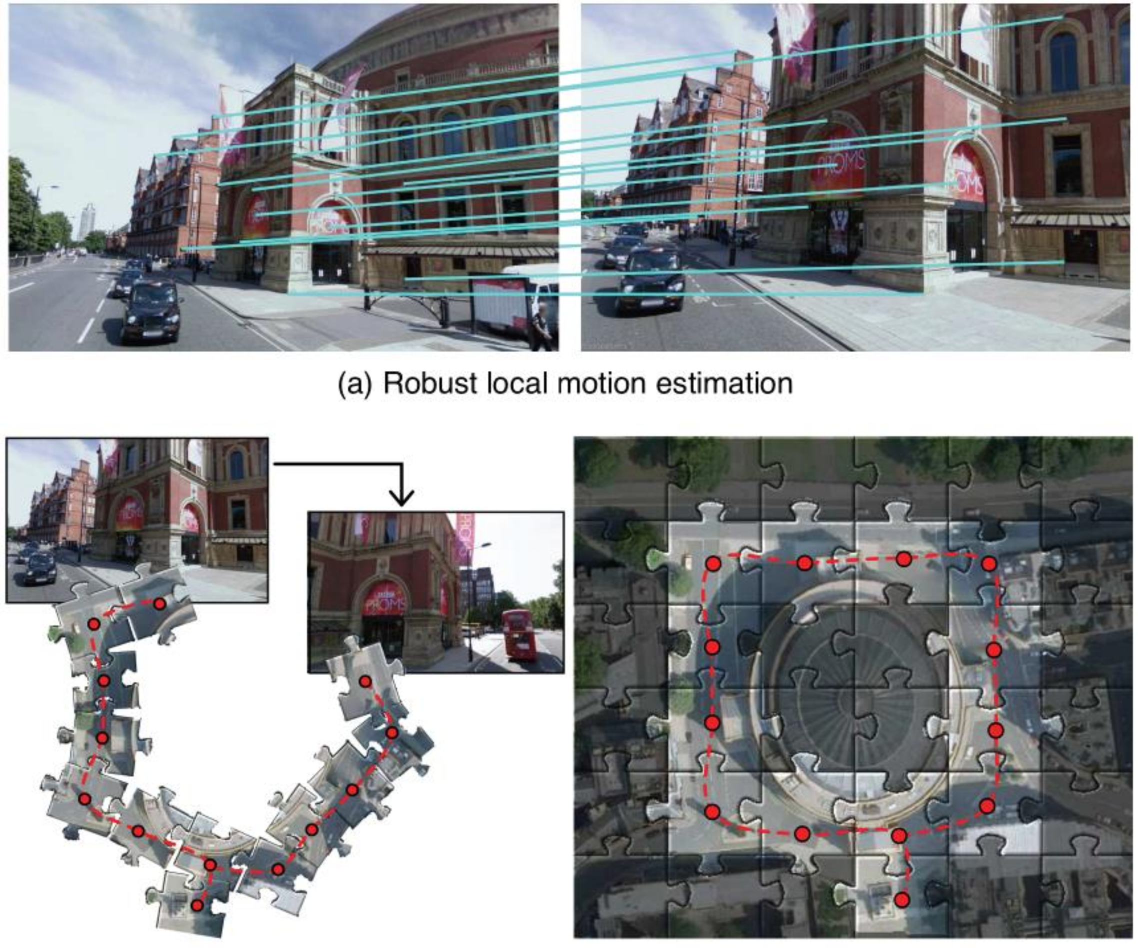




Project – Phase 7

- Loop Closure





(a) Robust local motion estimation

(b) Mapping and loop-closure detection

(c) Global optimisation

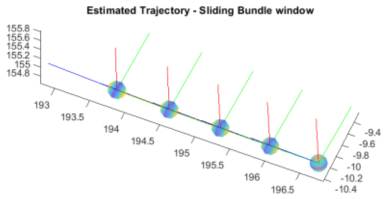
Applying Information Theory to Efficient SLAM – M.Chli

Frame Slam

- RANSAC $p = 0.99$
  - Min / max iteration
- Loop Closure
  - How many are needed?
  - Harder match
  - Can affect keyframes decision

FrameSlam - Visual Odometry

- *Sliding Bundle window* – A local window of the last 12 frames
  - New frame replaces oldest frame
  - Bundle is solved for motion estimation → VO
  - At any time contains at least one keyframe
  - When a new keyframe is introduced, marginalization is performed for a pose graph update



$$\langle c_1, \bar{c}_2, \cdots, c_{11}, c_{12} \rangle \leftarrow \boxed{\bar{c}_{13}}$$

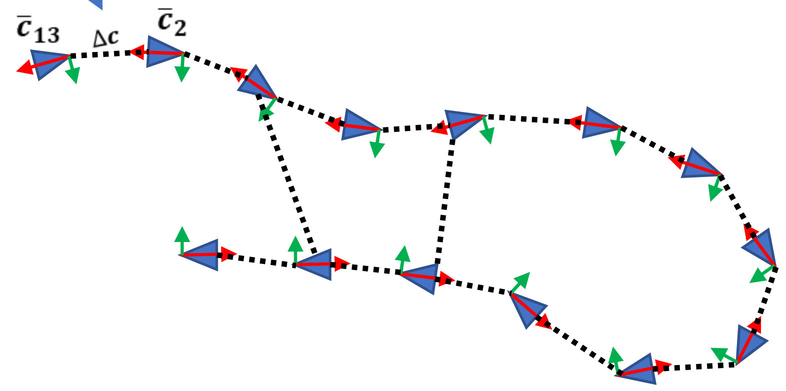
$$\Downarrow$$

$$\boxed{c_1} \leftarrow \langle \bar{c}_2, c_3, \cdots, c_{12}, \bar{c}_{13} \rangle \longrightarrow \Delta c(\bar{c}_2, \bar{c}_{13})$$

$$\Downarrow$$

$$\langle \bar{c}_2, c_3, \cdots, c_{12}, \bar{c}_{13} \rangle \leftarrow \boxed{c_{14}}$$

$$\Downarrow$$



Optimization

- Least Squares
  - Matlab:
    lsqnonlin(f, x0)
  - GTSAM / g2o

Accuracy

- Output Covariance vs. Input Covariance
- Covariance vs. projection error

Project Submission

Project Submission

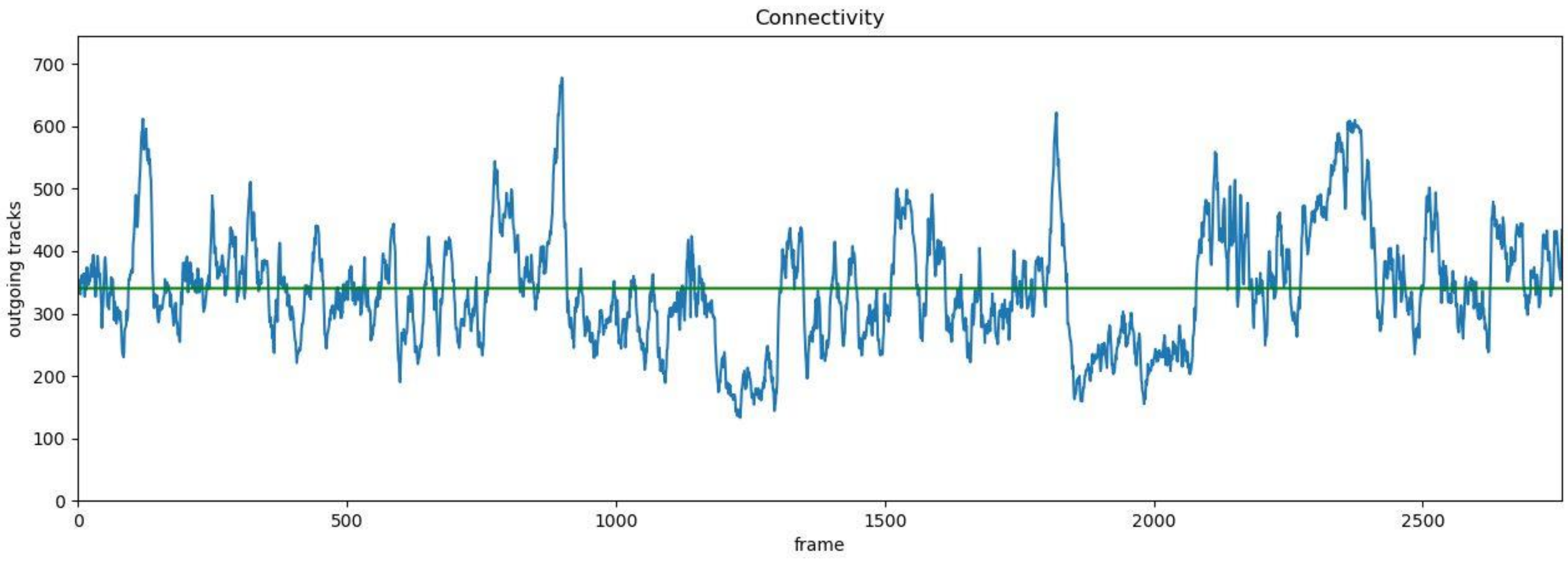
- Overview
- System Stages
  - Describe
  - Reference code
  - Document
- Criticize
- Graphs
  - Use Zoom effectively
  - Overlay when appropriate
  - Analyze for each graph
    - Explain what we see, interesting features, significance
    - Why is it there?

Project Submission

- Total number of tracks
- Number of frames
- Mean track length
- Mean number of frame links

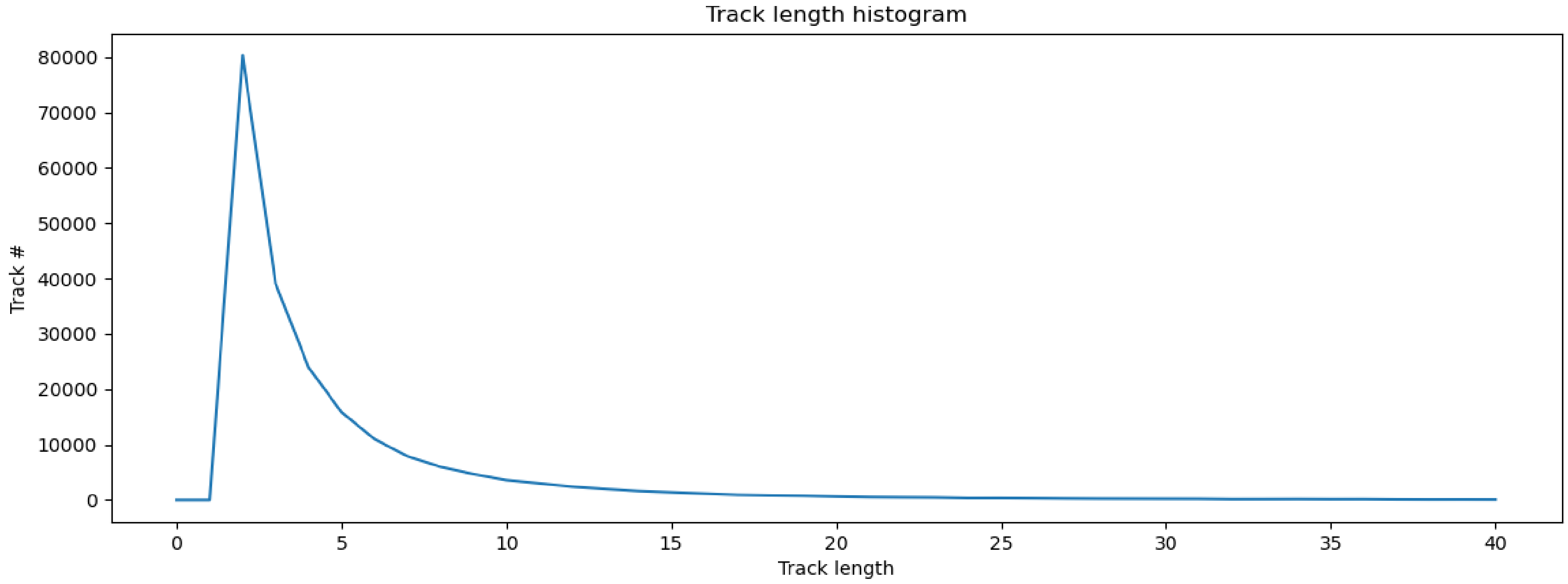
Project Submission

- Connectivity: For each frame the number of tracks **outgoing** to the next frame (the number of tracks on the frame with links also in the next frame)



Project Submission

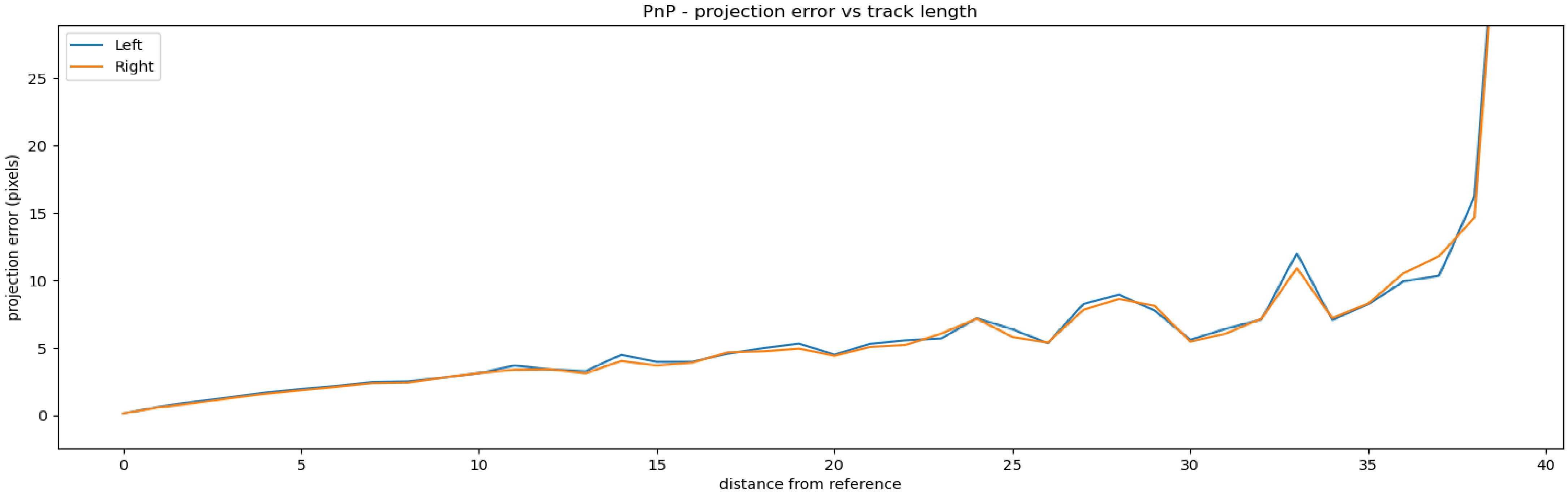
- Track length histogram



- Number of matches per frame
- Percentage of inliers per frame

Project Submission

- Median projection error of the different track links as a function of distance from reference frame
  - for PnP estimation
  - For Bundle estimation

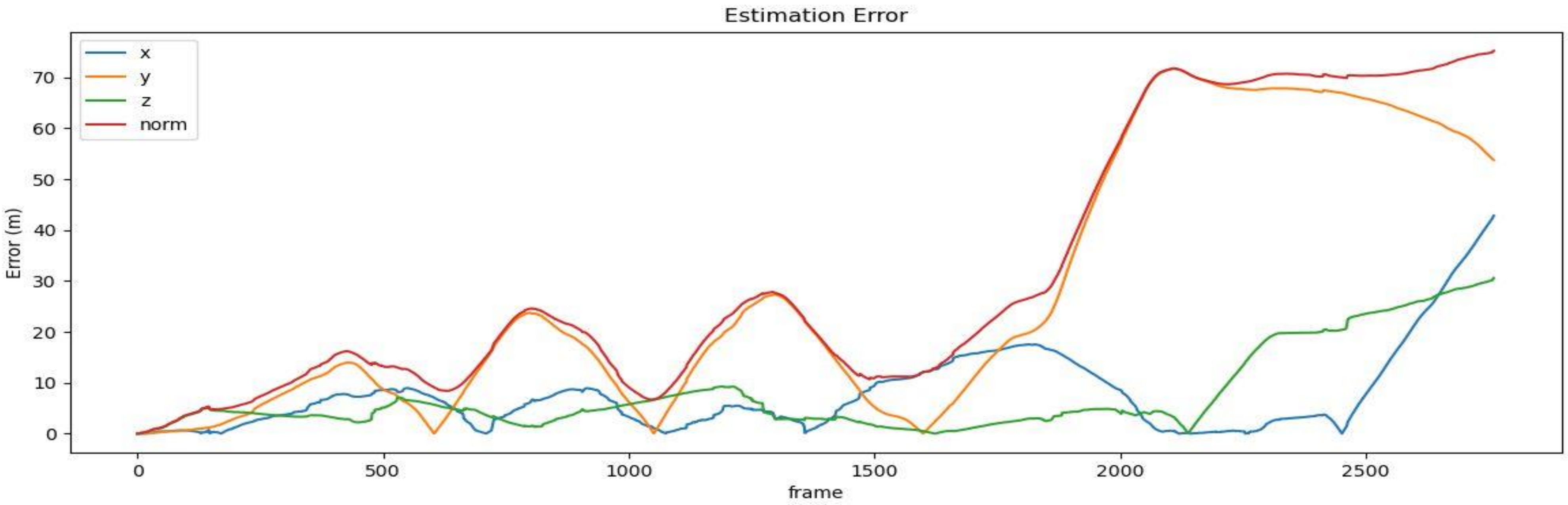


Project Submission

- Median factor error of the different track links as a function of distance from reference frame
    - for PnP estimation (initial solution)
    - for Bundle estimation (optimization result)

Project Submission

- Absolute PnP estimation error:
  - X axis error, Y axis error, Z axis error, Total error norm
  - Angle error



Project Submission

- Absolute Pose Graph (**without** loop closure) estimation error:
  - X axis error, Y axis error, Z axis error, Total error norm
  - Angle error

- Absolute Pose Graph (**with** loop closure) estimation error:
  - X axis error, Y axis error, Z axis error, Total error norm
  - Angle error

Project Submission

From all test sequences, our evaluation computes **translational and rotational errors** for all possible subsequences of length (100,...,800) meters. The evaluation table below ranks methods according to the average of those values, where errors are measured in percent (for translation) and in degrees per meter (for rotation). A more detailed comparison for different trajectory lengths and driving speeds can be found in the plots underneath. **Note:** On 03.10.2013 we have changed the evaluated sequence lengths from (5,10,50,100,...,400) to (100,200,...,800) due to the fact that the GPS/OXTS ground truth error for very small sub-sequences was large and hence biased the evaluation results.

The global leaders in inertial and GNSS since 1998

At OxTS we're passionate about inertial navigation and how we can help our customers with our technology. With over two decades of experience in combining the best of high precision GNSS receivers and world-class inertial navigation expertise, OxTS' products have become the industry standard for automotive testing and are widely used in other industries.

Our products provide position, roll, pitch, heading and other measurements of vehicles on land, sea and in the air. Our highly accurate RT3000 series is used by almost all car manufacturers in the world for vehicle dynamics testing, validating advanced driver assistance systems (ADAS) sensors or developing self-driving cars.

Our range of combined, compact GNSS/INS systems are used for sensor position and orientation on mobile mapping vehicles or for direct georeferencing data from airborne surveying.

Project Submission

From all test sequences, our evaluation computes **translational and rotational errors** for all possible subsequences of length (100,...,800) meters. The evaluation table below ranks methods according to the average of those values, where errors are measured in percent (for translation) and in degrees per meter (for rotation). A more detailed comparison for different trajectory lengths and driving speeds can be found in the plots underneath. **Note:** On 03.10.2013 we have changed the evaluated sequence lengths from (5,10,50,100,...,400) to (100,200,...,800) due to the fact that the GPS/OXTS ground truth error for very small sub-sequences was large and hence biased the evaluation results.

- ## Relative PnP estimation error:

  The error of the relative pose estimation compared to the round truth relative pose.
  Evaluated on sequence lengths of (100, 300, 500, 800), choose closest keyframe if necessary

  - X axis, Y axis, Z axis, Total norm – Error percent
  - Angle – Error in deg/m

- ## Relative Bundle estimation error:

  - X axis, Y axis, Z axis, Total norm
  - Angle

Angles

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

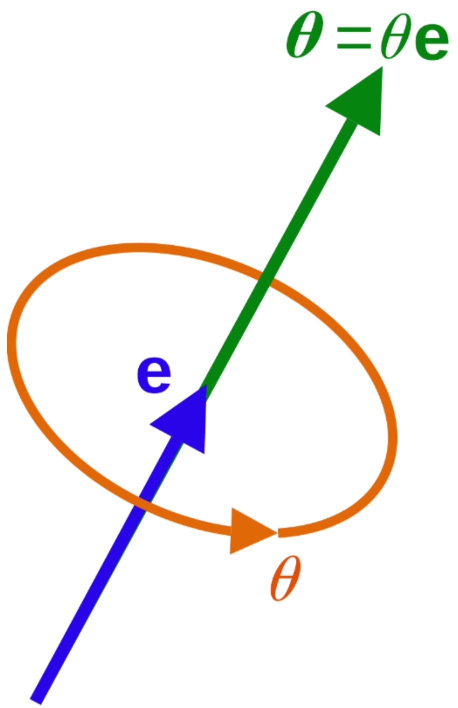
$$R = R_z(\psi)R_y(\theta)R_x(\phi)$$

```python
# Euler angles --> rotation matrix
def euler_angles_2_rotation_matrix(eul):
    Rx = np.array([[1, 0, 0],
                   [0, np.cos(eul[0]), -np.sin(eul[0])],
                   [0, np.sin(eul[0]),  np.cos(eul[0])]])
    Ry = np.array([[np.cos(eul[1]), 0,  np.sin(eul[1])],
                   [0, 1, 0],
                   [-np.sin(eul[1]), 0, np.cos(eul[1])]])
    Rz = np.array([[np.cos(eul[2]), -np.sin(eul[2]), 0],
                   [np.sin(eul[2]),  np.cos(eul[2]), 0],
                   [0, 0, 1]])

    R = np.dot(Rz, np.dot(Ry, Rx))
    return R
```

Angles

```python
# rotation matrix --> Euler angles
def rotation_matrix_2_euler_angles(R):
    sy = np.sqrt(R[0, 0]*R[0, 0] + R[1, 0]*R[1, 0])
    singular = sy < 1e-6
    if not singular:
        x = np.arctan2(R[2, 1], R[2, 2])
        y = np.arctan2(-R[2, 0], sy)
        z = np.arctan2(R[1, 0], R[0, 0])
    else:
        x = np.arctan2(-R[1, 2], R[1, 1])
        y = np.arctan2(-R[2, 0], sy)
        z = 0
    return np.array([x, y, z])
```




```python
axis_vec, _ = cv2.Rodrigues(R.transpose() @ Q)
np.linalg.norm(axis_vec) * 180 / np.pi
```

Project Submission

- Number of matches per successful loop closure frame
- Inlier percentage per successful loop closure frame

Project Submission

- Uncertainty size vs keyframe – pose graph **without** loop closure:
  - Location Uncertainty
  - Angle Uncertainty

- Uncertainty size vs keyframe – pose graph **with** loop closure:
  - Location Uncertainty
  - Angle Uncertainty

- Describe how you represent uncertainty size!