Simultaneous localisation and mapping

By Bohdan Hlovatskyi, Mykhailo Pasichnyk, Stefan-Yuriy Malyk

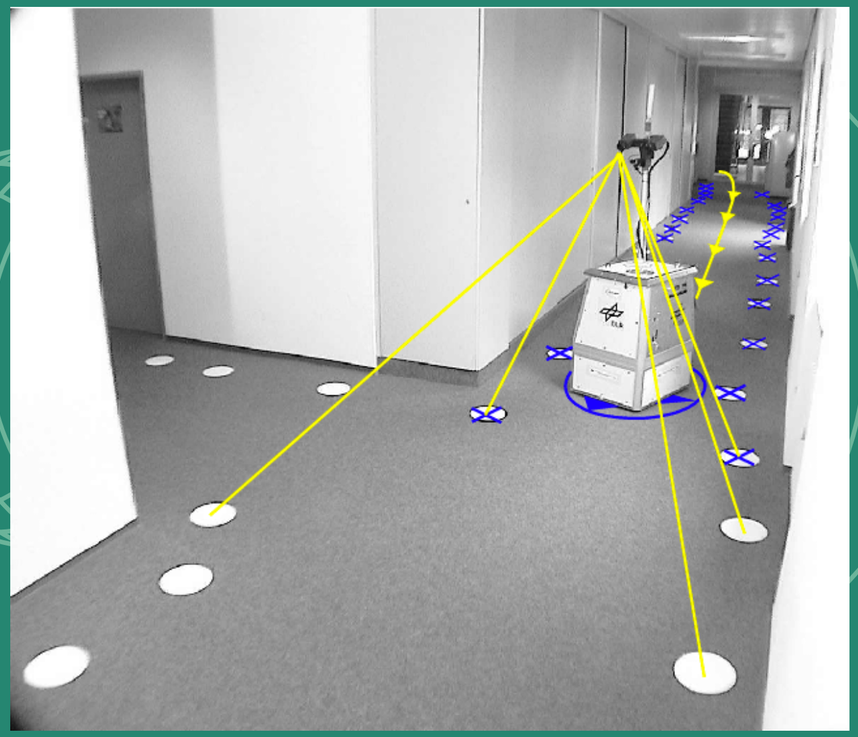
Problem

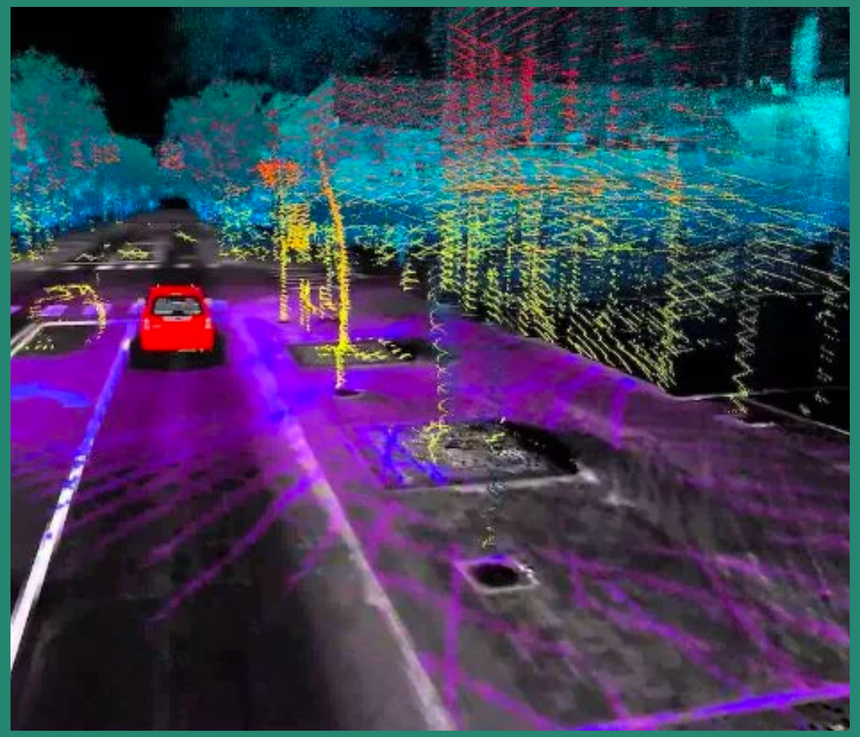
Students of UCU are creating a terminator that will join the Armed Forces of Ukraine. They constructed a bulletproof body, and powerful servo motors to guarantee fast movements. However, ally counties refused to provide students with a modern navigation system. Now they need to construct one on their own!

Problem

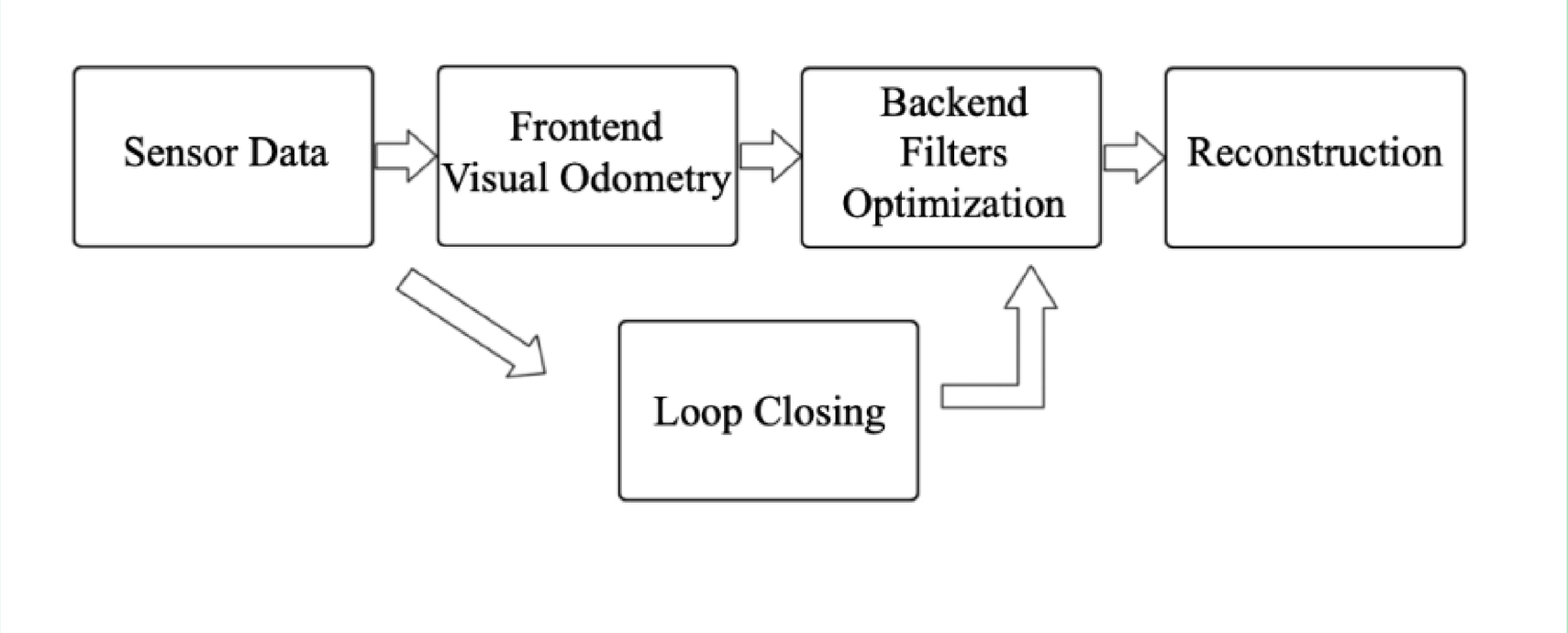
Where am I?

How environment around me looks like?



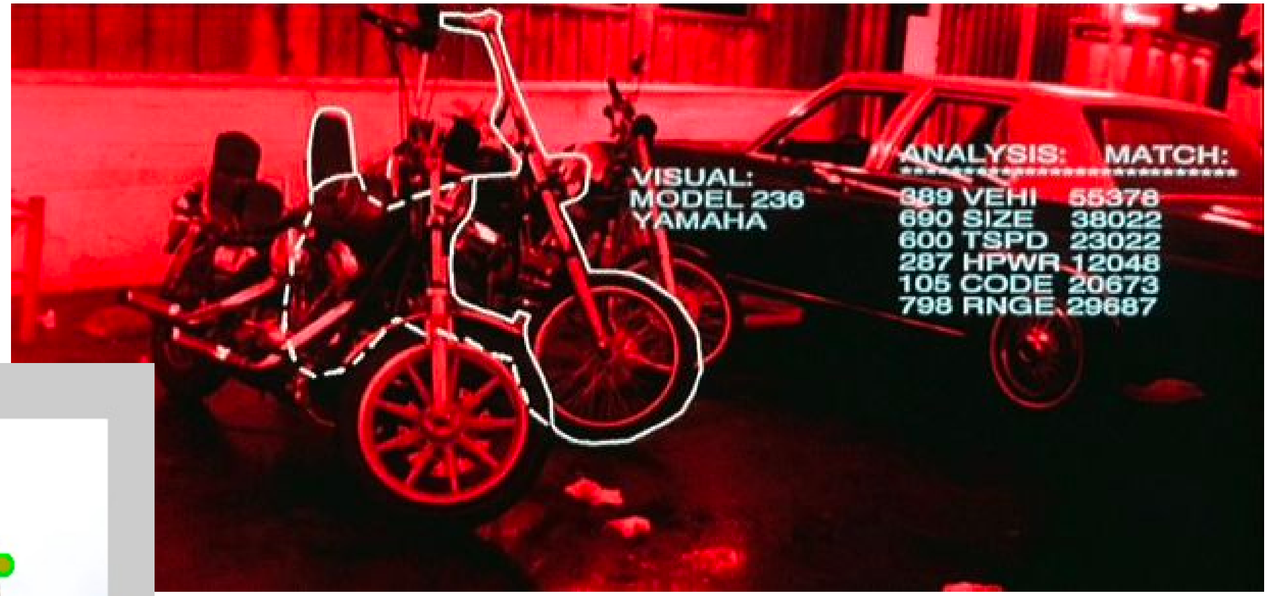


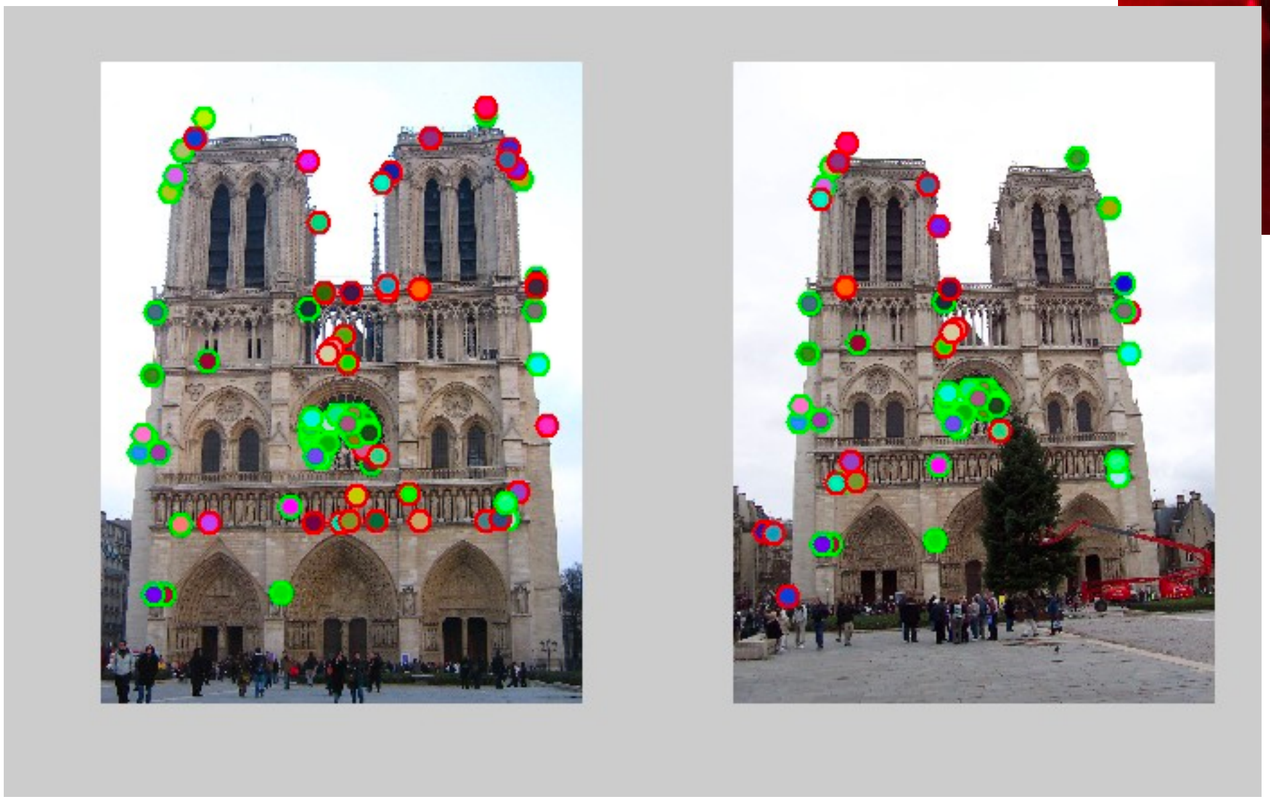
How it works


Sensor Data
Frontend
Visual Odometry
Backend
Filters
Optimization
Reconstruction
Loop Closing

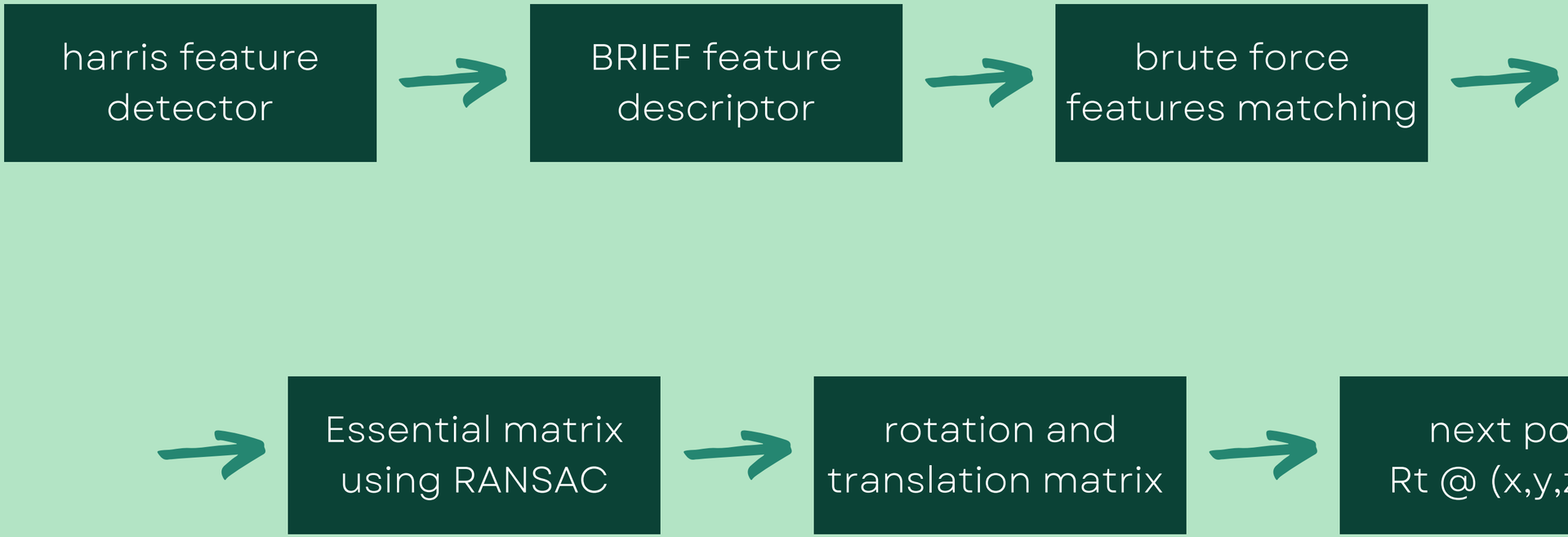
Approaches

Detect, detect, detect!





How it works? 2.0 (frontend)

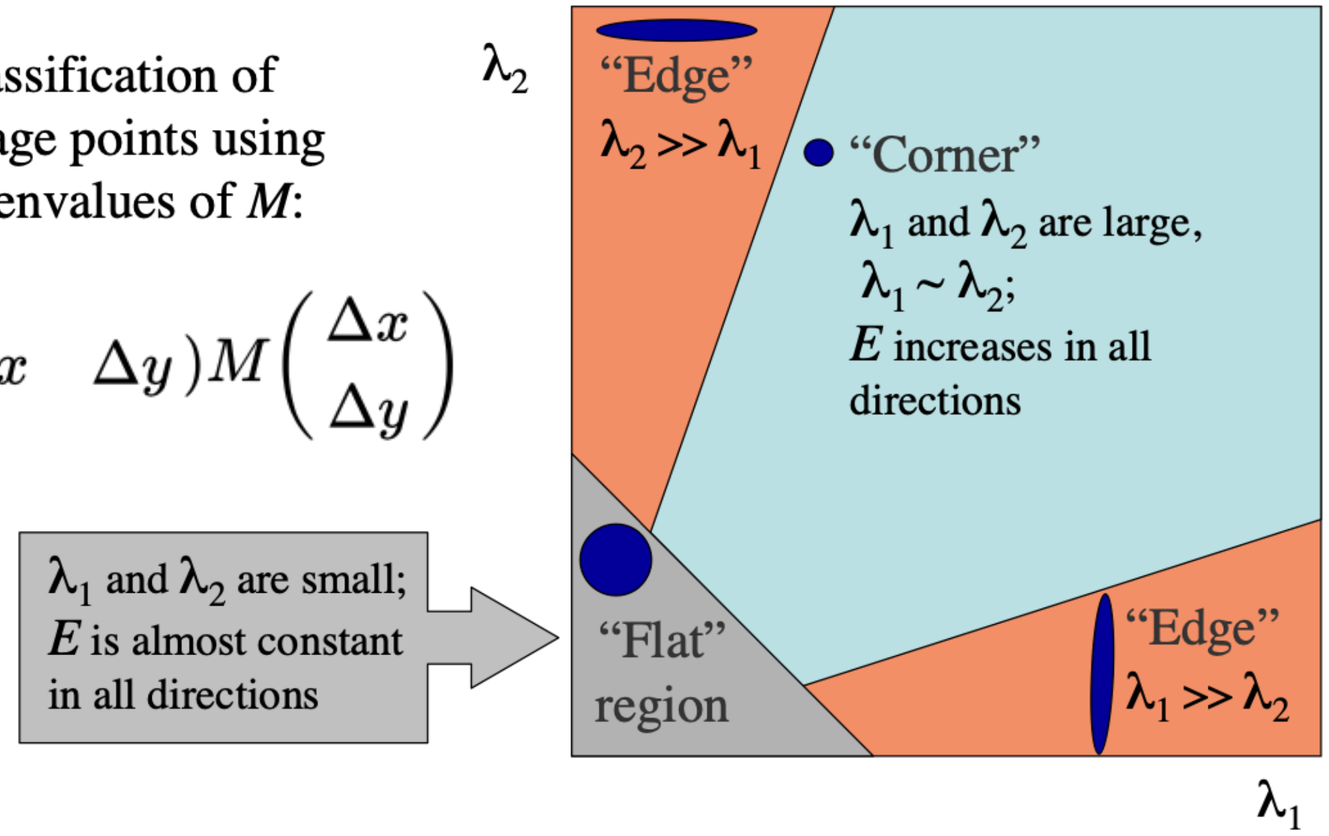


Harris corner detector

$$f(\Delta x, \Delta y) = \sum_{(x_k, y_k) \in W} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

Classification of image points using eigenvalues of M:

$$f(\Delta x, \Delta y) \approx (\Delta x \quad \Delta y) M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$



BRIEF: Binary Robust Independent Elementary Features

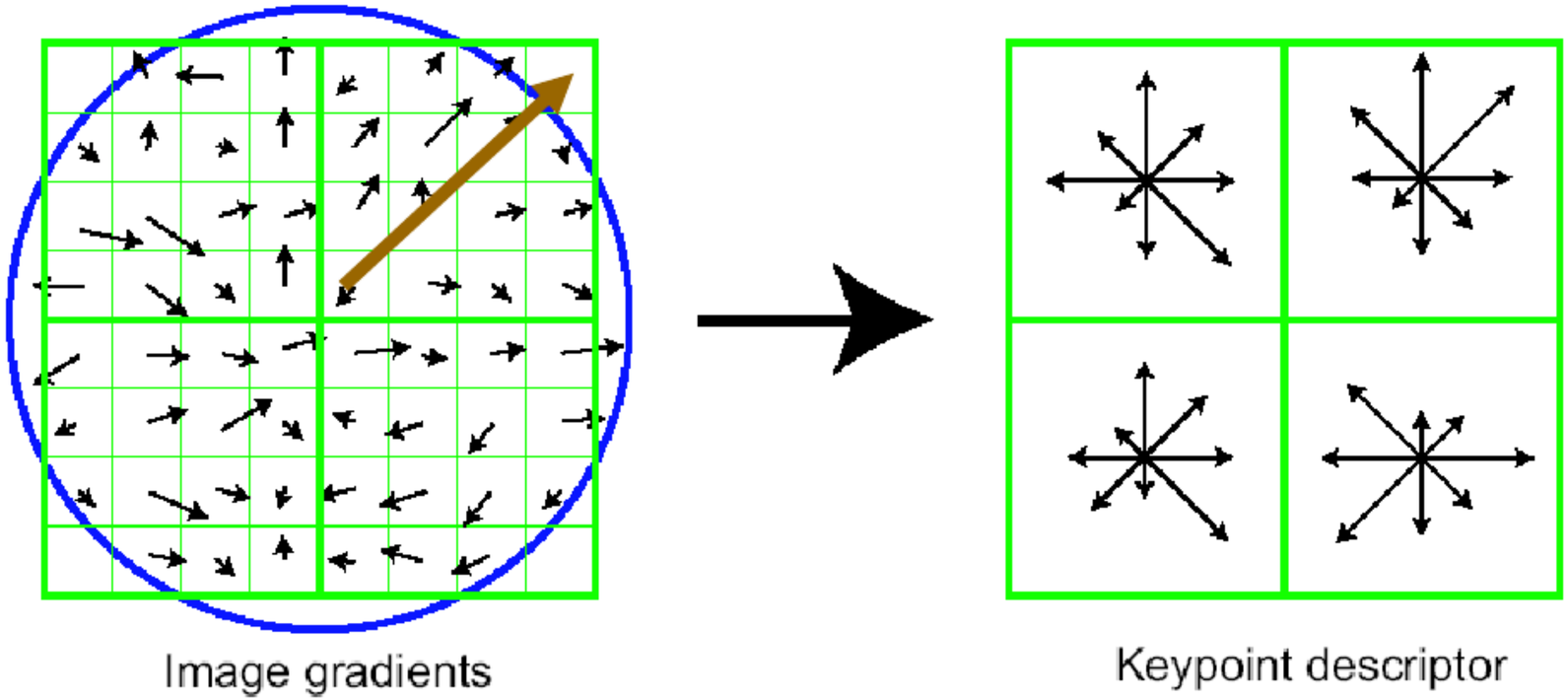


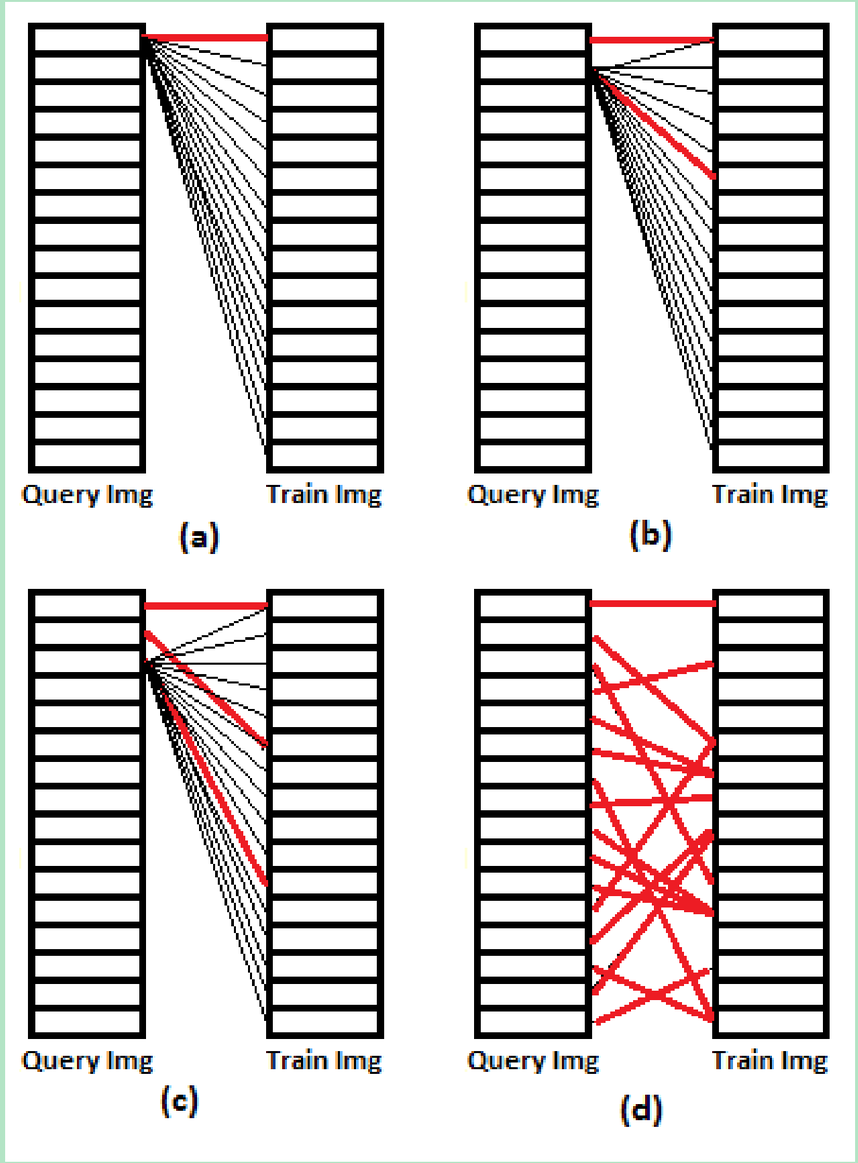
Image gradients

Keypoint descriptor

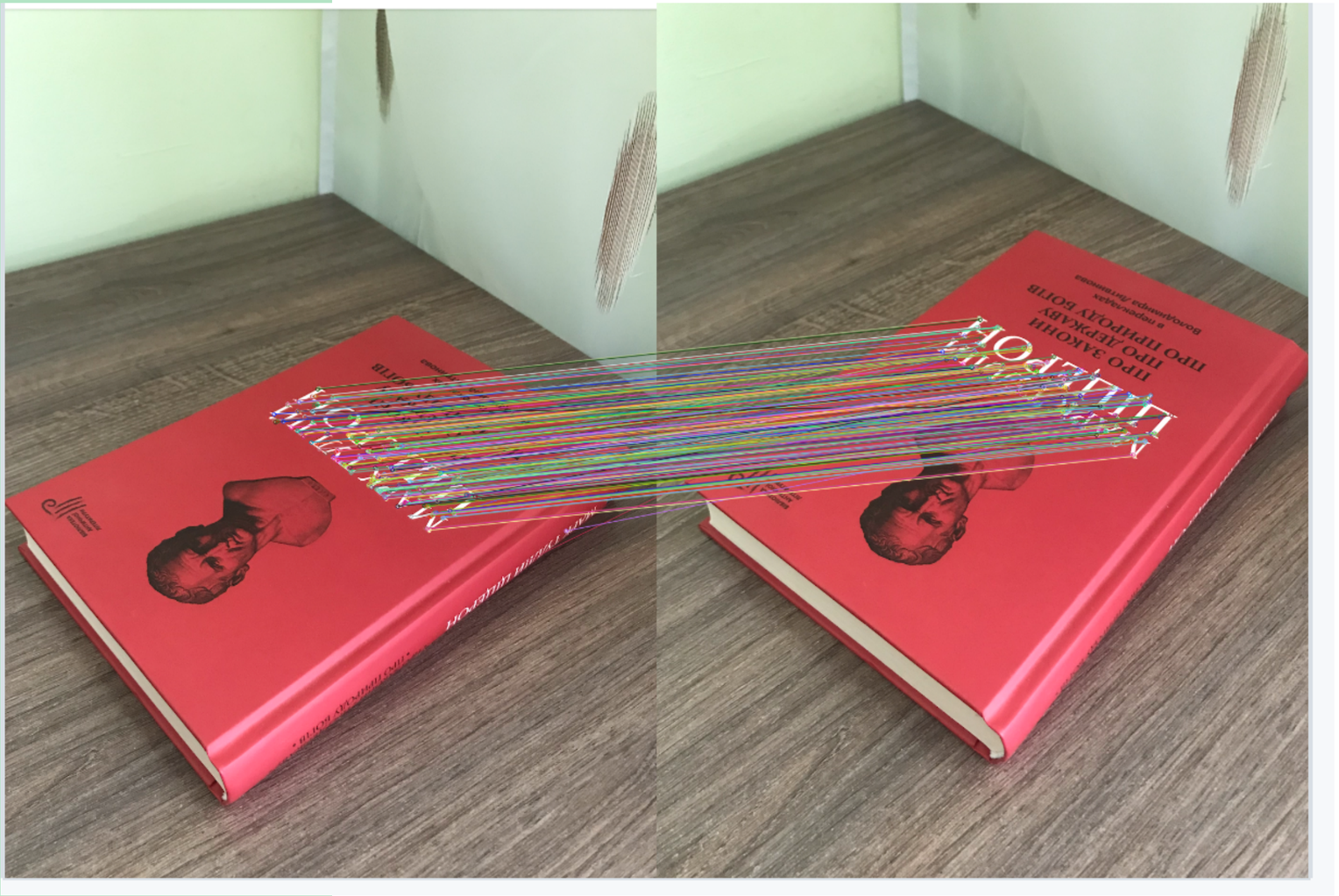
As this algorithm is binary, therefore it's very efficient and could be used in the real-time SLAM.

Brute force features matching

Nothing interesting. Just check all possible combinations and return the best matches using the Hemming distance of BRIEF. A more efficient way is KNN, but in case we have about 200-500 features and where fast binary check it enough for real-time

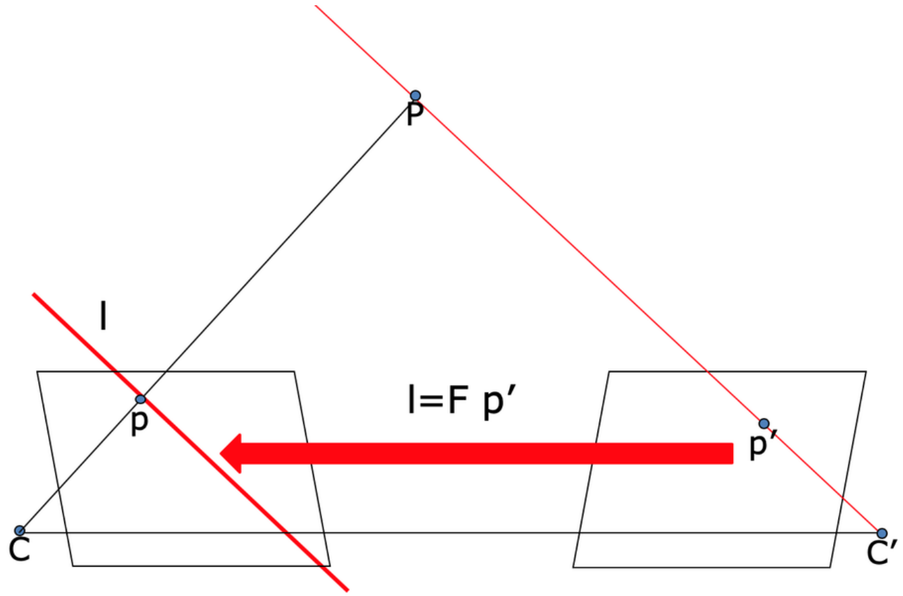


That a match, what's next?



Essential matrix estimating

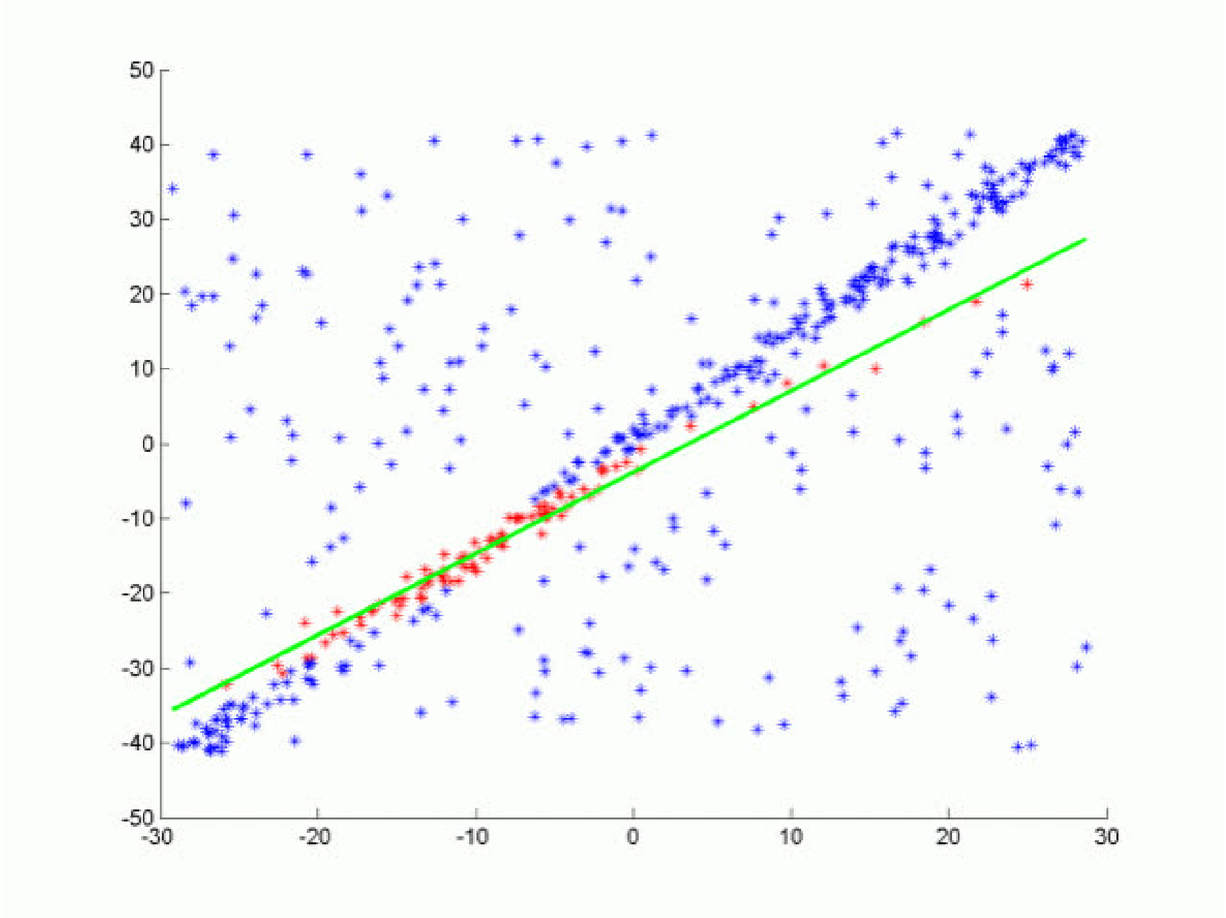
The Essential matrix is used to estimate the mapping of points in one images to lines (that correspond to the depth of the original image) in another image, which are normalizing using intrinsic of camera . Let we have normalized point x in C and x' in C'. Because x'@F gives us the epipolar line, for where the same point could appear in the image C depending on depth. Therefore (x'@E)*x = 0, as the point should be on the calculated resulting line.



We need to calculate E (3*3 mat) up to scale, so need 8 equations => at least 8 points to calculate E.

RANSAC

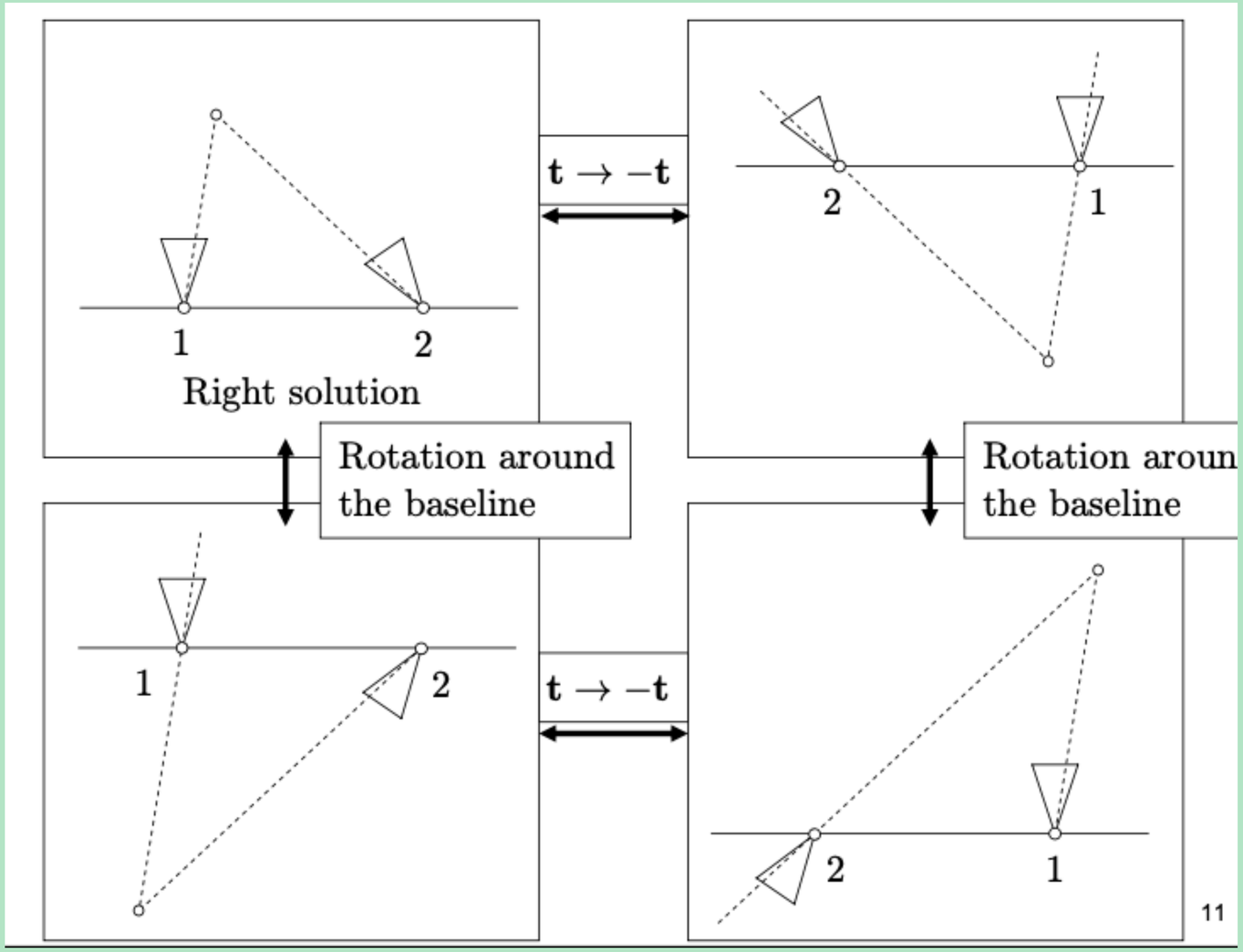
In this plot we choose randomly sample of 2 points and build linear regression and score by num of inliers each model many times. Because of P&S course we get, that our resulted model, with the best score should have the largest number of inliers. The same we do with Essential matrix, but using 8 points.



[R|t] Matrix recovery

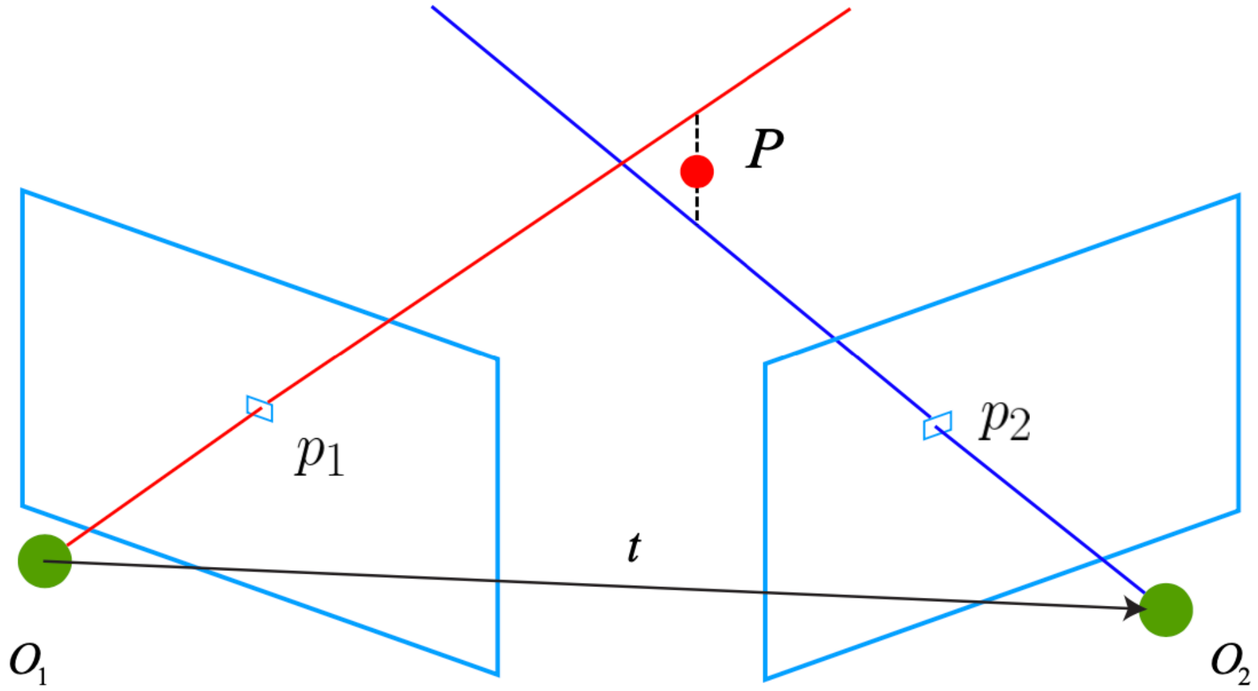
$$\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

$$\begin{aligned}
\mathbf{t}_1^\wedge &= \mathbf{U}\mathbf{R}_Z(\tfrac{\pi}{2})\mathbf{\Sigma}\mathbf{U}^T, \quad \mathbf{R}_1 = \mathbf{U}\mathbf{R}_Z^T(\tfrac{\pi}{2})\mathbf{V}^T \\
\mathbf{t}_2^\wedge &= \mathbf{U}\mathbf{R}_Z(-\tfrac{\pi}{2})\mathbf{\Sigma}\mathbf{U}^T, \quad \mathbf{R}_2 = \mathbf{U}\mathbf{R}_Z^T(-\tfrac{\pi}{2})\mathbf{V}^T.
\end{aligned}$$



Triangulation!

$$\mathbf{x}_1 = \mathbf{K}^{-1}\mathbf{p}_1, \quad \mathbf{x}_2 = \mathbf{K}^{-1}\mathbf{p}_2.$$



$$s_2\mathbf{x}_2 = s_1\mathbf{R}\mathbf{x}_1 + \mathbf{t}.$$

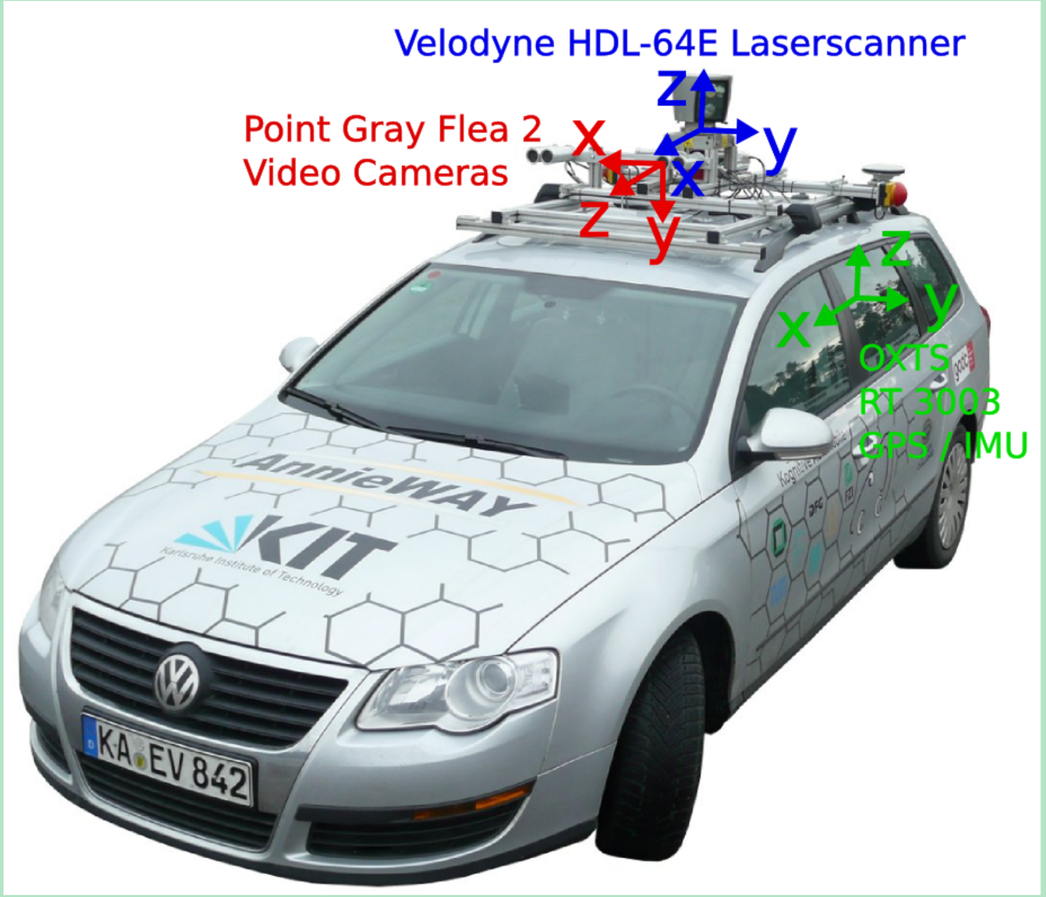
Time to test!

Evaluation

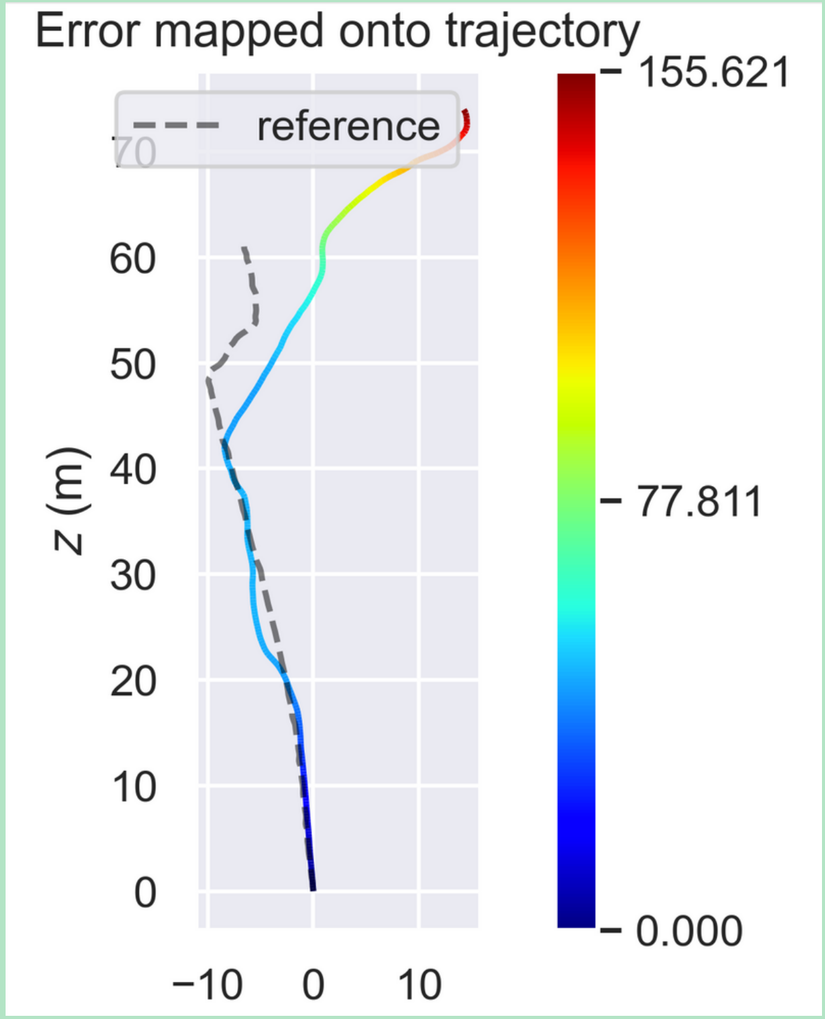
Kitti dataset - open datasets for computer vision researchers.

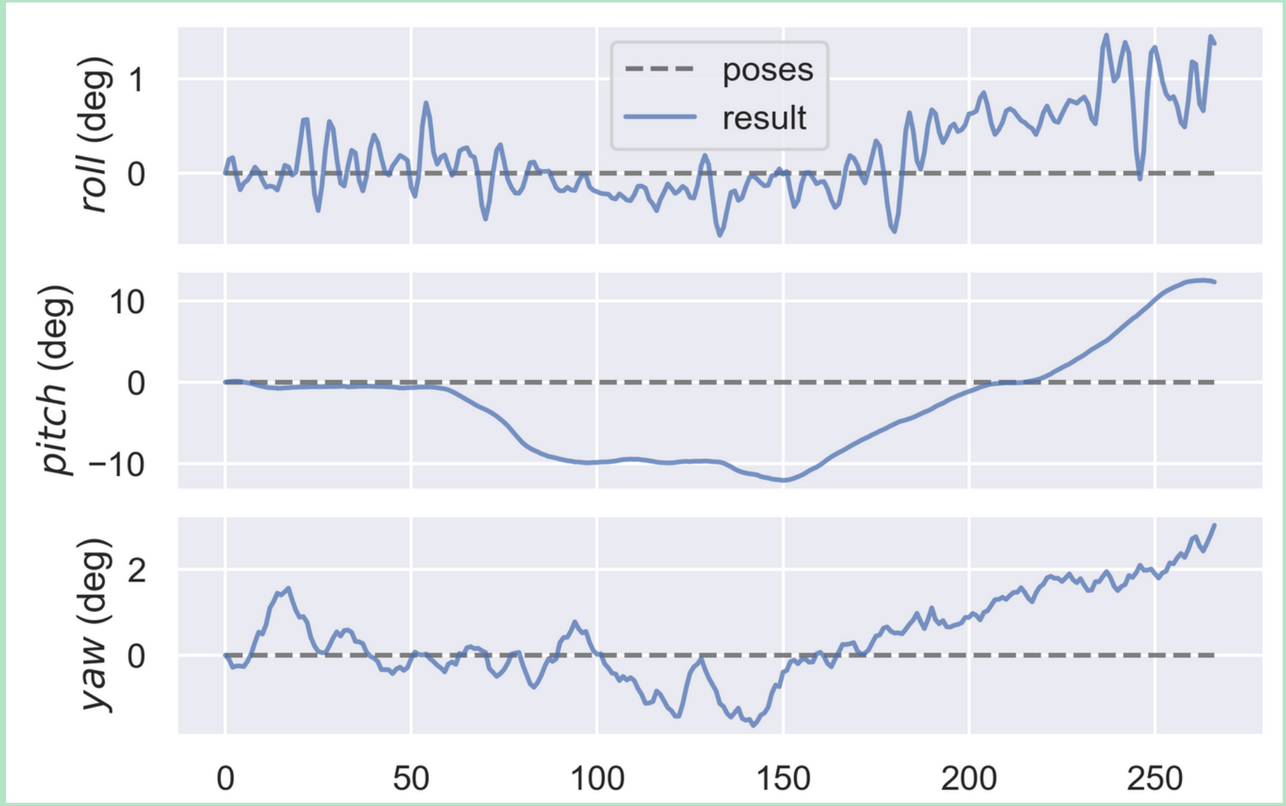
It provides data from several RGB cameras that can be used for SLAM **evaluation**.

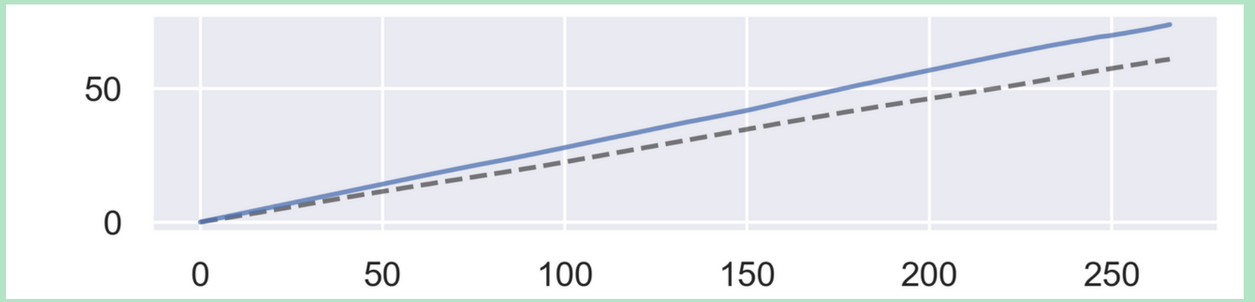
Accurate **ground truth** poses are obtained via Inertial Measurement Units (**IMU**).



Results


Error mapped onto trajectory
reference
155.621
77.811
0.000
z (m)
70
60
50
40
30
20
10
0
−10
0
10


poses
result
roll (deg)
1
0
pitch (deg)
10
0
−10
yaw (deg)
2
0
0
50
100
150
200
250


50
0
0
50
100
150
200
250

And finally

```
object = get_object_in_front()
if object.is_enemy:
  say("Hasta la vista, baby!")
  destroy(object)
else:
  say("Glory to Ukraine!🇺🇦")
```

Demonstration

https://www.youtube.com/watch?v=Gj2gREpRj60