Monocular Visual Odometry on Blimp Platform

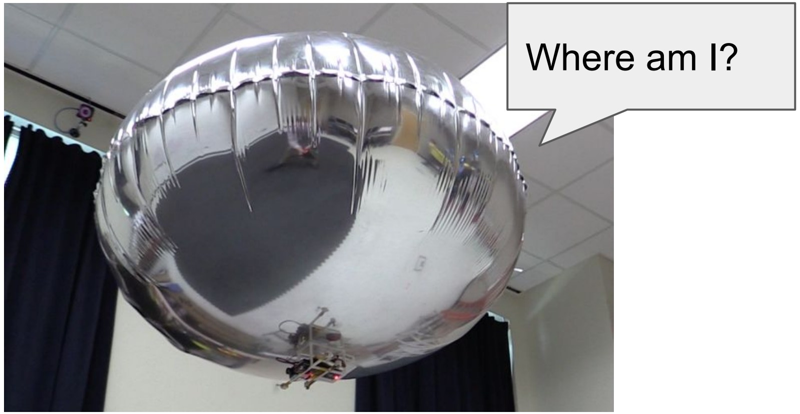
with Noisy Image Stream

Senior Design Team XXLs

Real-life Problems and Motivations

1. The current blimp platform does not have a localization capability without opti-track system
2. The dynamics of the blimp platform is unique and ruled out easy vo implementation
3. Carrying capability of the blimp limits the camera that provides high quality images

- We would like to extend the blimp localization capability outside the lab given the blimp dynamics and hardware limitations



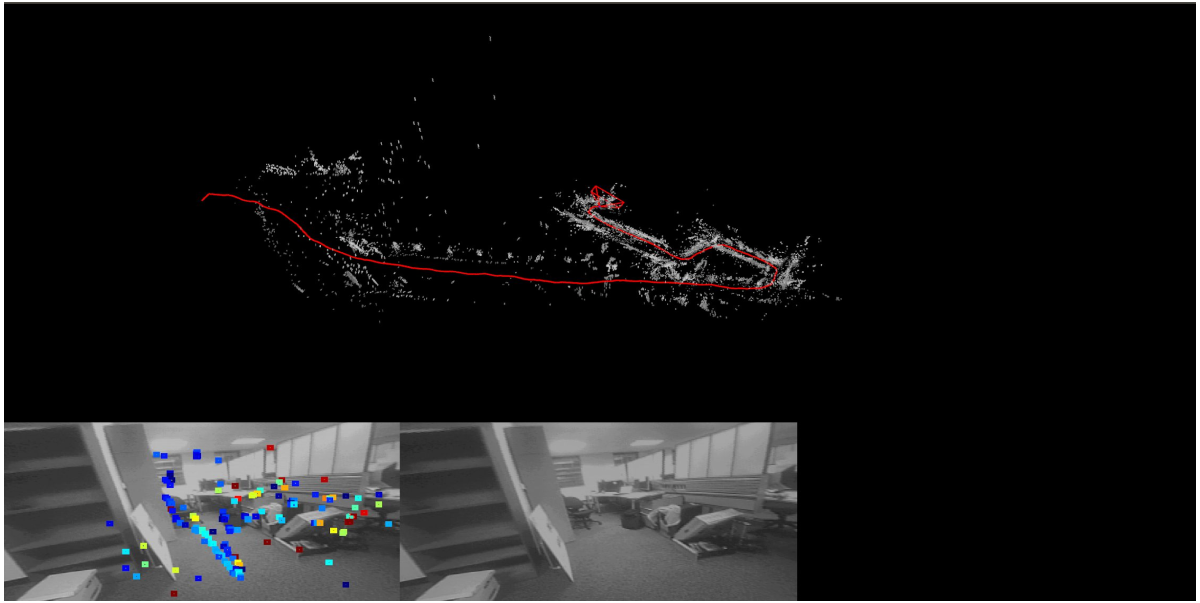
Real-life Problems and Motivations Cont'd

Possible Applications with a VO pipeline:

1. Automatic Navigations
   a. Blimp Tour Guide
   b. Warehouse Surveillance
2. Human/Robot-Robot Interaction
   a. Swarm Blimps
   b. Tool Delivering
3. With the current situation, we could let blimp guide people at COVID-19 testing locations.

Goal

- Deploy and achieve accurate (robust) visual odometry on Blimp, given the constraints of camera image quality and the platform dynamics characteristics.



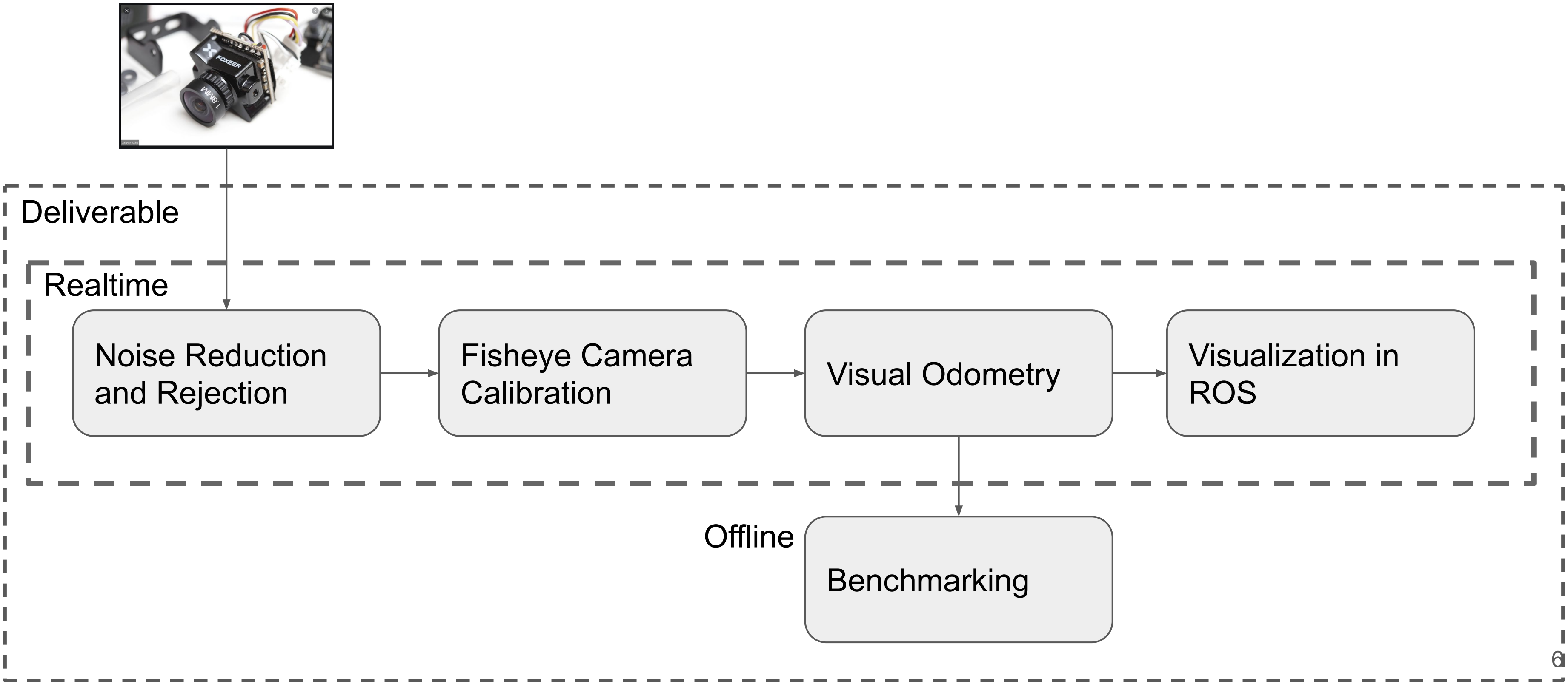
Project Deliverables

A ROS pipeline for blimp to perform visual odometry on the blimp, including:

1. Noise reduction of camera stream and rejection of unrepairable image frames
2. Calibration of Fisheye Camera
3. Visual Odometry and Studies on Different Methods
4. Visualization of trajectory
5. Benchmarking

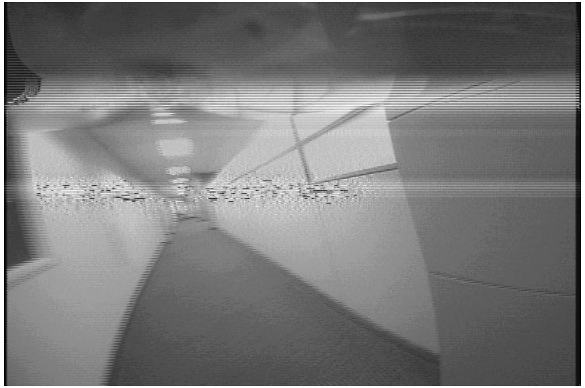
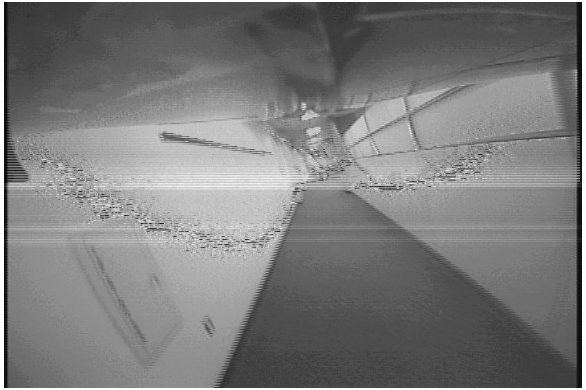
Also, we would like to share lessons learned from the project

System Structure and Project Deliverables



Noisy Images

- ~25% of the image from Analog camera is noisy (on a 4000 images dataset)
- Will break feature tracking
- Need rejection on highly noisy image and filtering on all other images

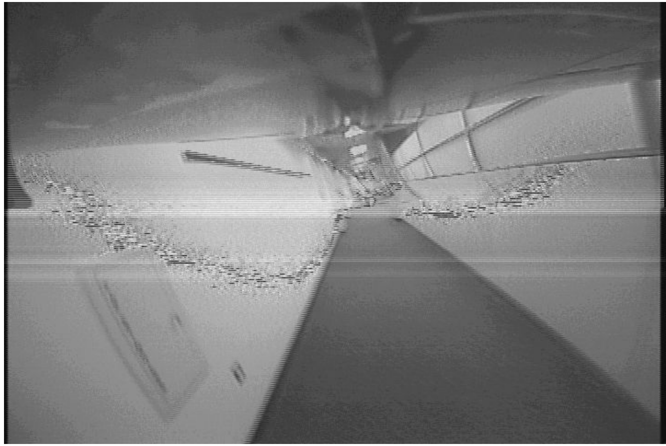




Typical Bad Features on Noisy Images

1. Noise Induced
2. Exposure / Lights Induced
3. Aliasing Induced

Dealing with Noisy Images

- Image Reconstruction
  - Apply Fourier Transform
- Noise Detection and Rejection
  - Segment image horizontally
  - Variance Estimation with vertical Gaussian kernel ([[2], [-4], [2]])



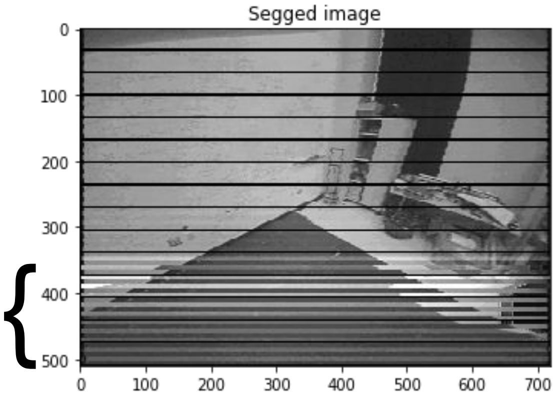
- Methods tried but poor performed
  - Erosion
  - SVM with different kernels (MSE, Entropy)
  - Deep Learning Image classifier

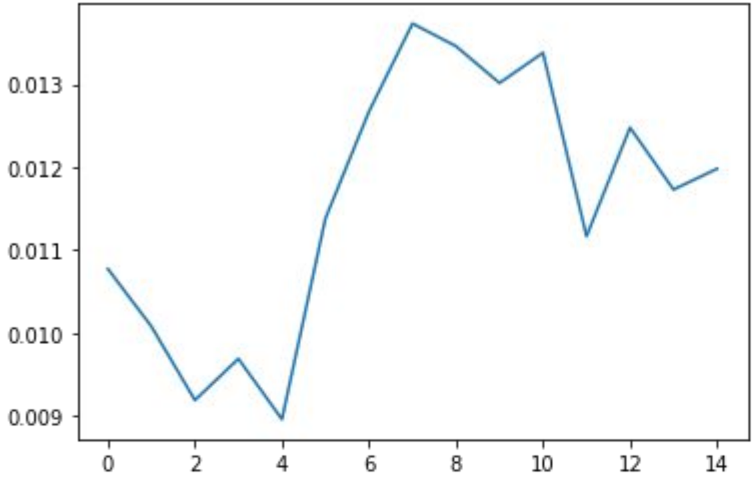
Noise is appearing with little pattern!

Noisy Images - Image Rejection

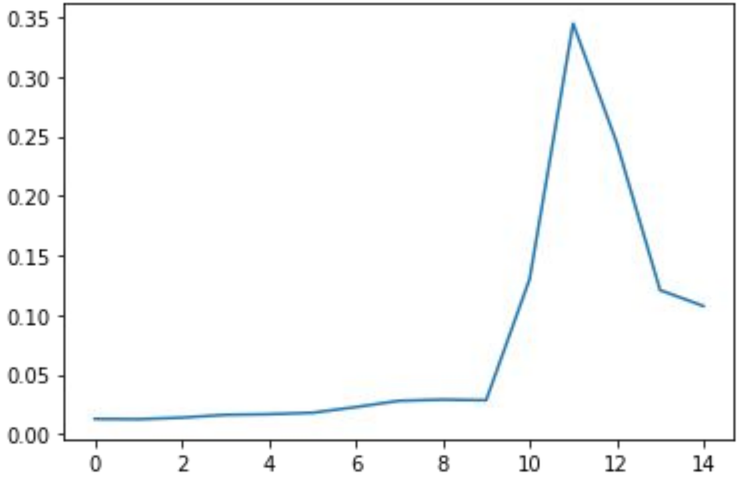
- Vertical Gaussian kernel outperforms 2D Gaussian kernel in detecting horizontal noises

Last 5 "rows"





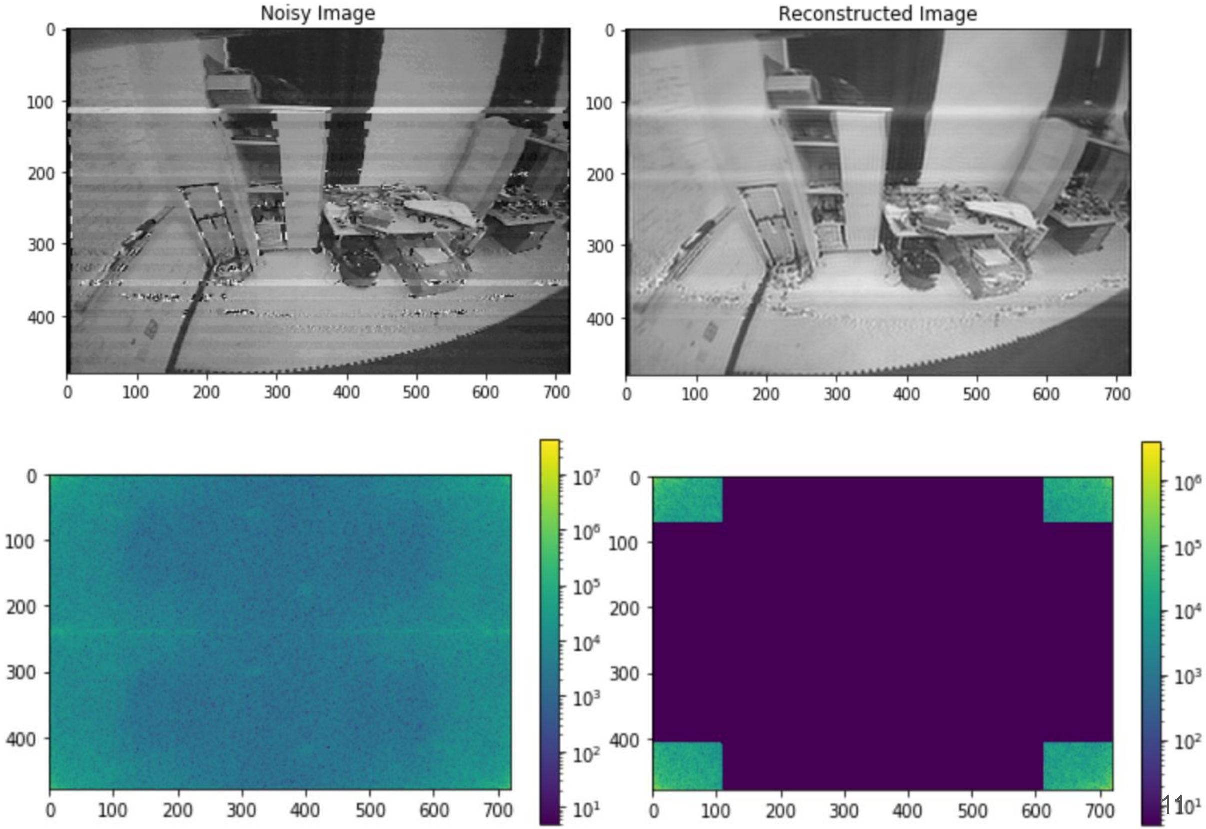
2D Gaussian



1D Gaussian (in the vertical direction)

Noisy Images - Image Reconstruction

- Fourier Transform
    - Reflect Intensity change
    - Reconstruct noisy image



Fisheye Camera Calibration - Why

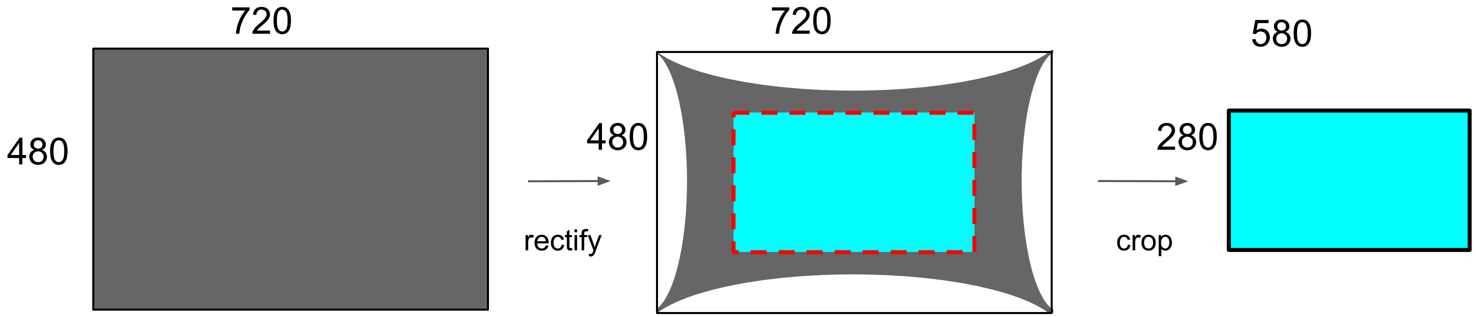
1. Why we need camera calibration
    a. Fisheye Camera imposes huge lens distortion on the image stream
    b. To restore the original features, we need to perform camera calibration
2. Camera calibration is after dealing noisy images
    a. Noise usually appear within large horizontal blocks
    b. Camera rectification may warped the horizontal lines in to “parabola”

120° FOV

Fisheye Camera Calibration

OpenCV / ROS cannot perform Fisheye Calibration Well

We use OcamCalib, but need to manually select calibration points

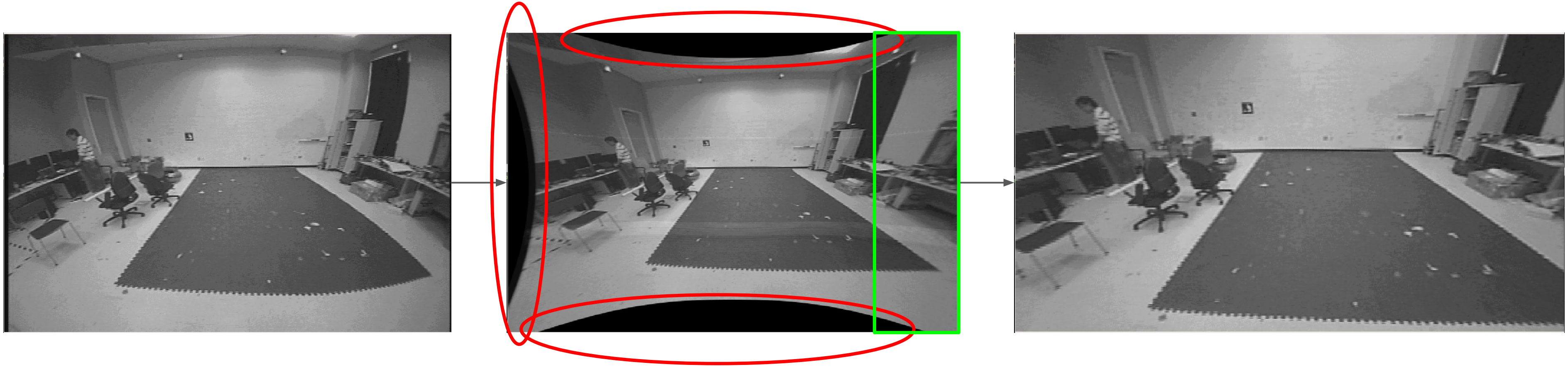


Fisheye Camera Calibration

Original

After Rectification

After Cropping



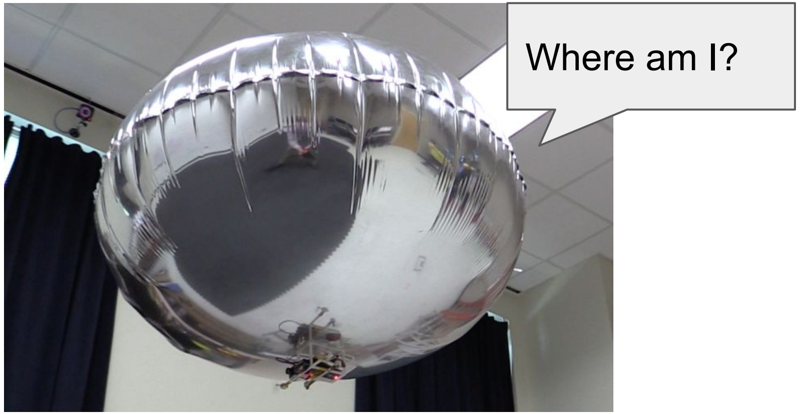
720 x 480

720 x 480

580 x 280

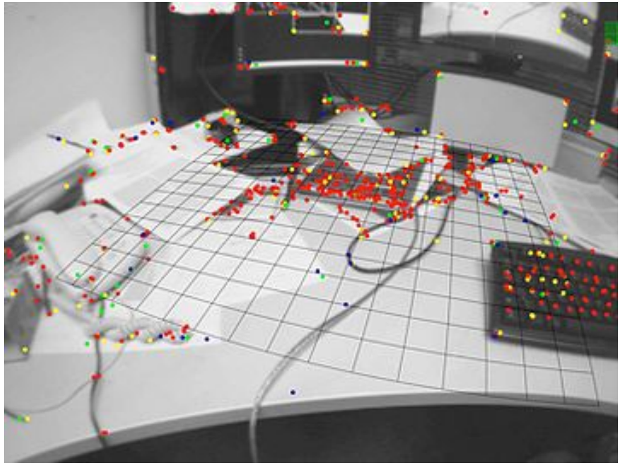
Visual Odometries

To get the localize the blimp, we use the preprocessed data to perform visual odometry



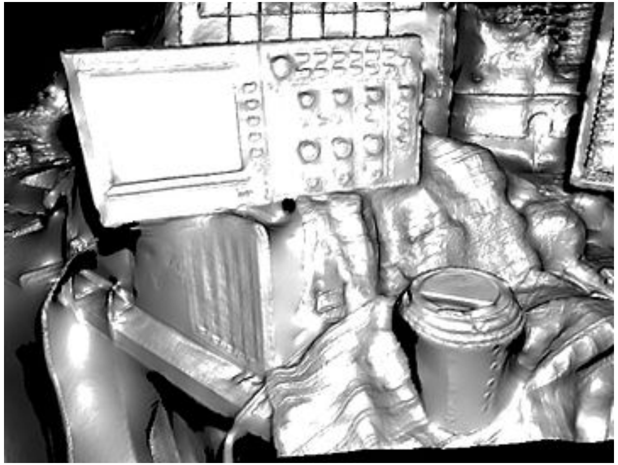
Sparse vs Dense

Sparse: Using only small selected pixels[1]



Fast

Dense: Using all pixels[2]



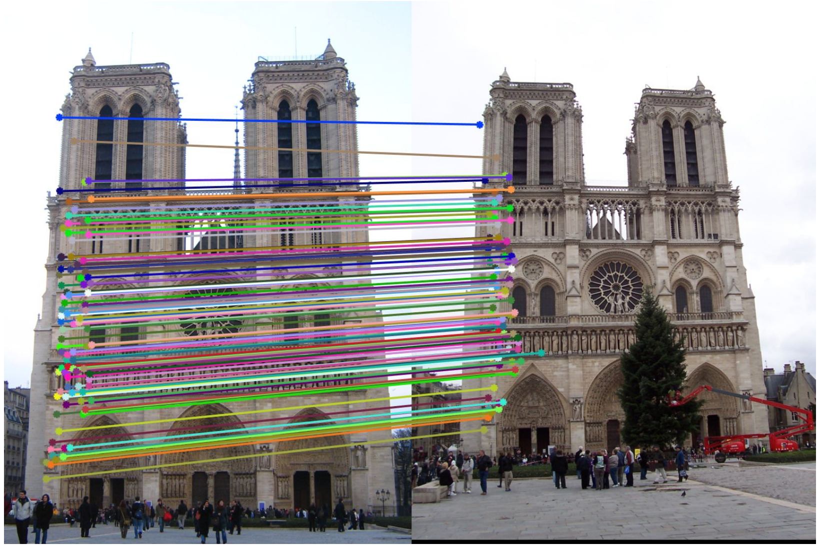
Slow

[1]G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, 2007, pp. 225-234.

[2] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2320-2327.

Indirect vs Direct

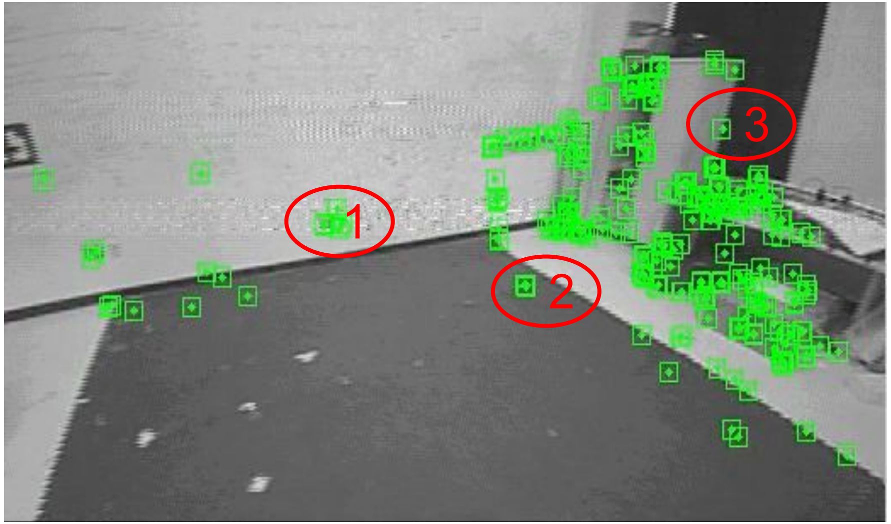
Indirect (Feature-Based): Extract features first, then solve epipolar geometry



Direct: Using photometric reprojection error to deduct epipolar geometry

Recap: Typical Bad Features on Noisy Images

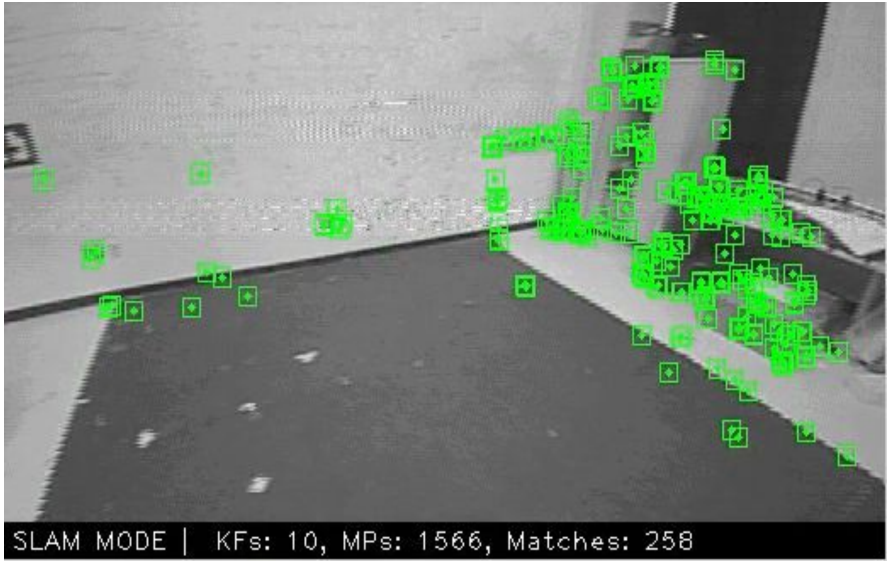
1. Noise Induced
2. Exposure / Lights Induced
3. Aliasing Induced



Feature-Based VO’s Challenges

Failure Stories of Indirect Methods

1. Vanilla KL-VO (SIFT)
2. SVO (FAST)
3. Orb-slam (ORB)

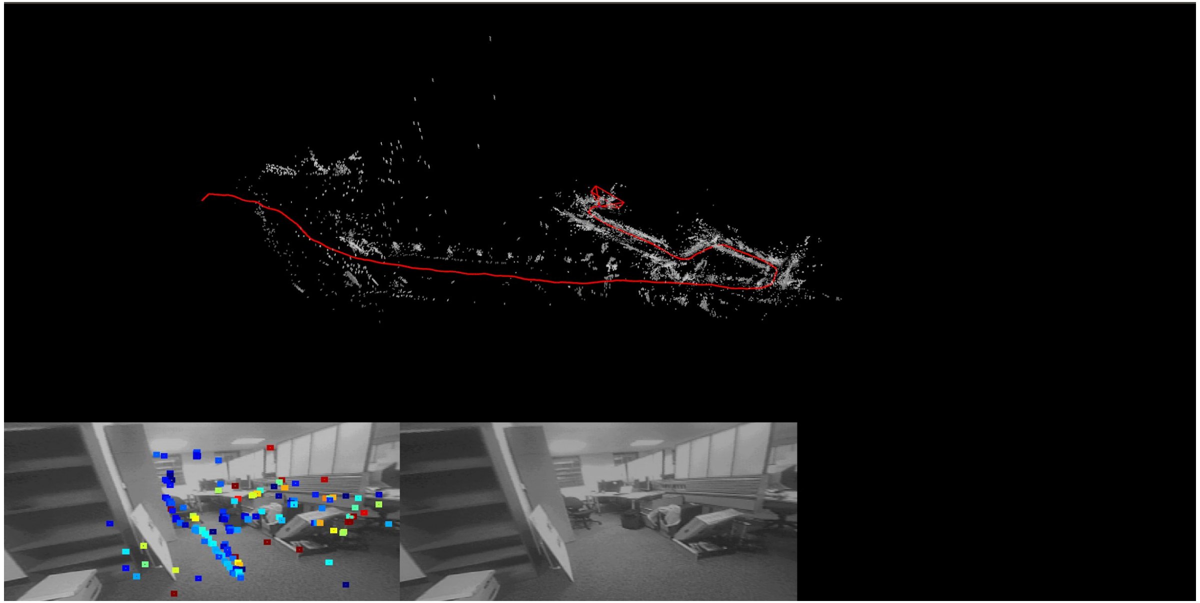


S.Lucey, “Direct Visual SLAM”, url:http://16623.courses.cs.cmu.edu/slides/Lecture_19.pdf

DSO - Direct Sparse Odometry[1]

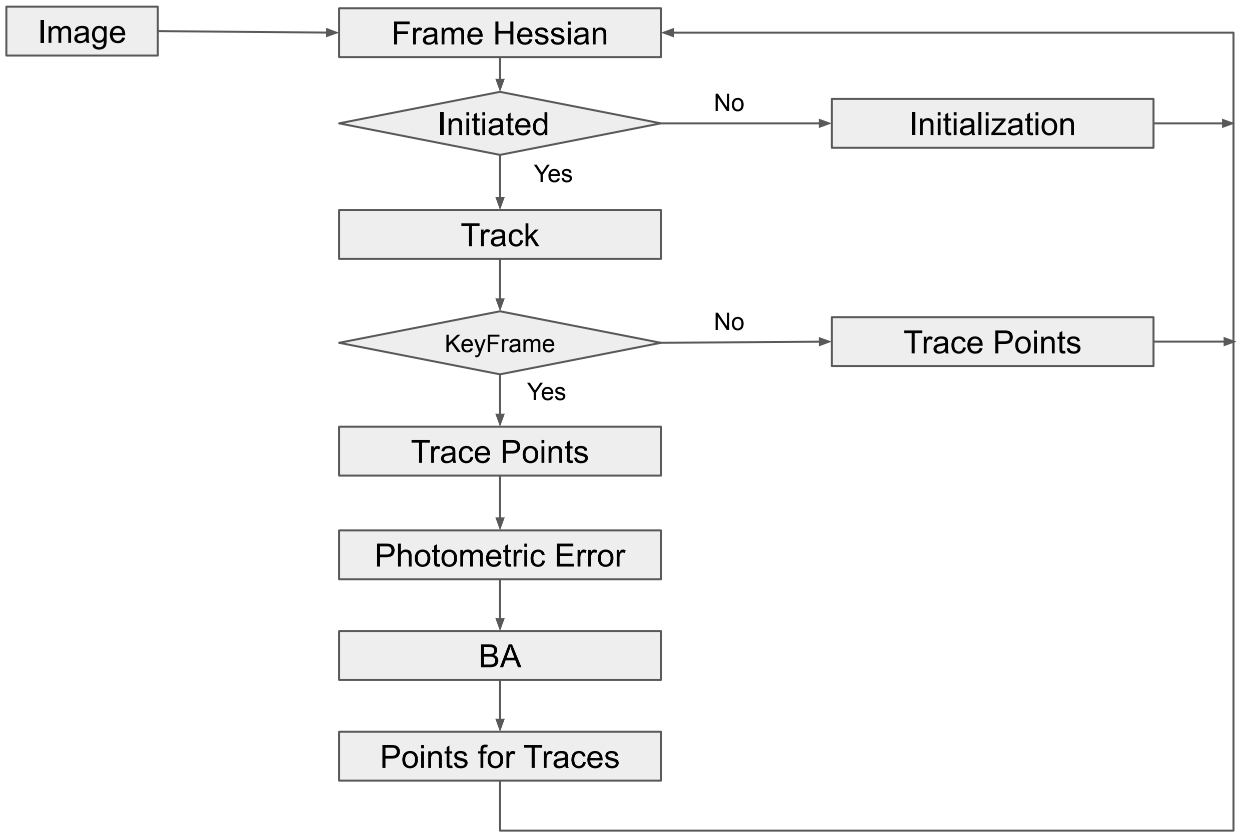
Direct: Minimizing photometric reprojection error

Sparse: Evenly sampling, selecting and using points with high image gradients

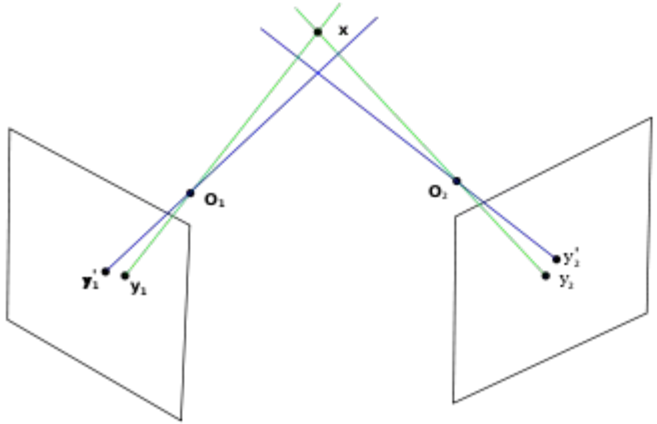


[1] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 611-625, 1 March 2018.

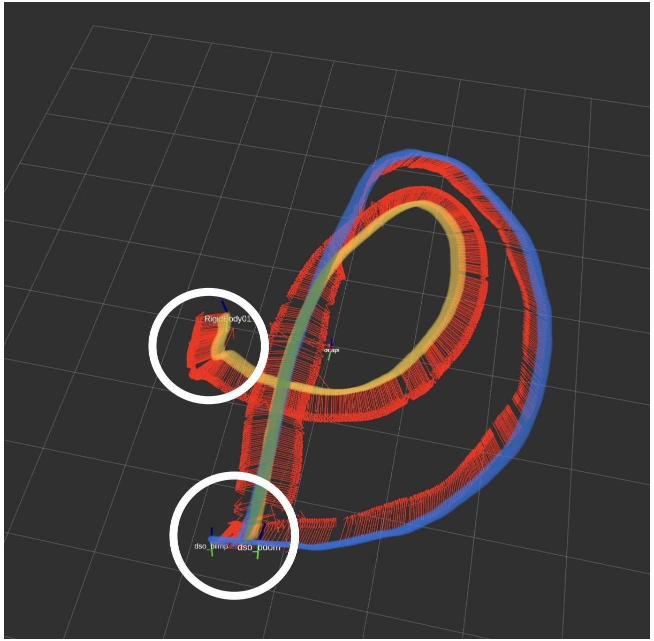
DSO - Direct Sparse Odometry



Huge Rotation - Huge Problem



Triangulation Fails Tracking

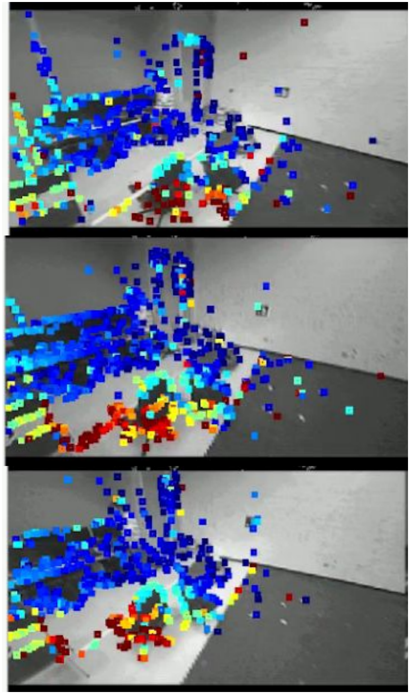


Original Method

Sample angular displacement according to previous transformation

Problems:

1. Low efficiency
2. Non-Linearity comes from rotations, failing tracking
3. Bad Tracking -> Bad Triangulations -> Failing Pipeline
4. Cluttered feature points with similar colors might yield small photometric error in the first place.



Solution

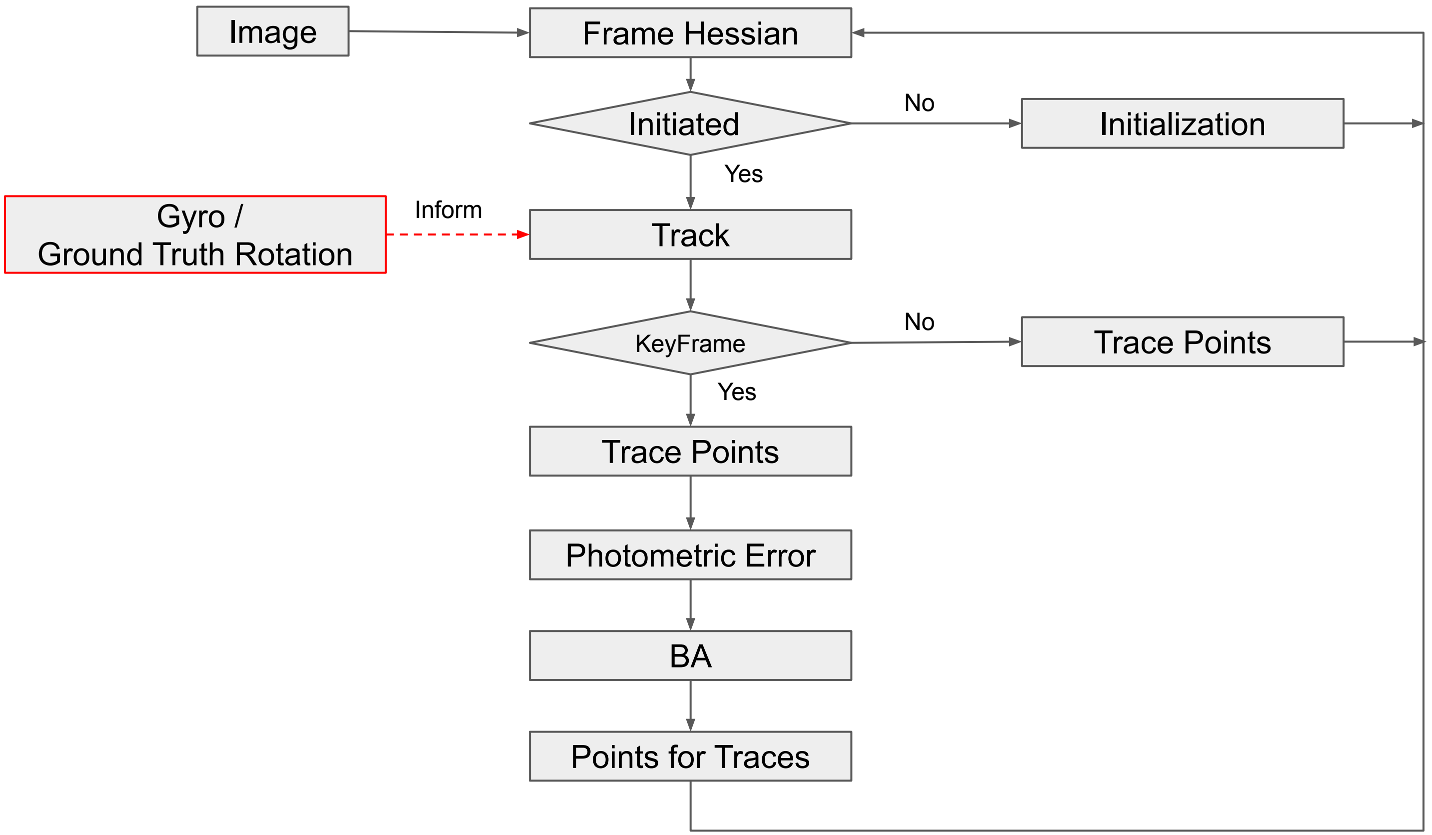
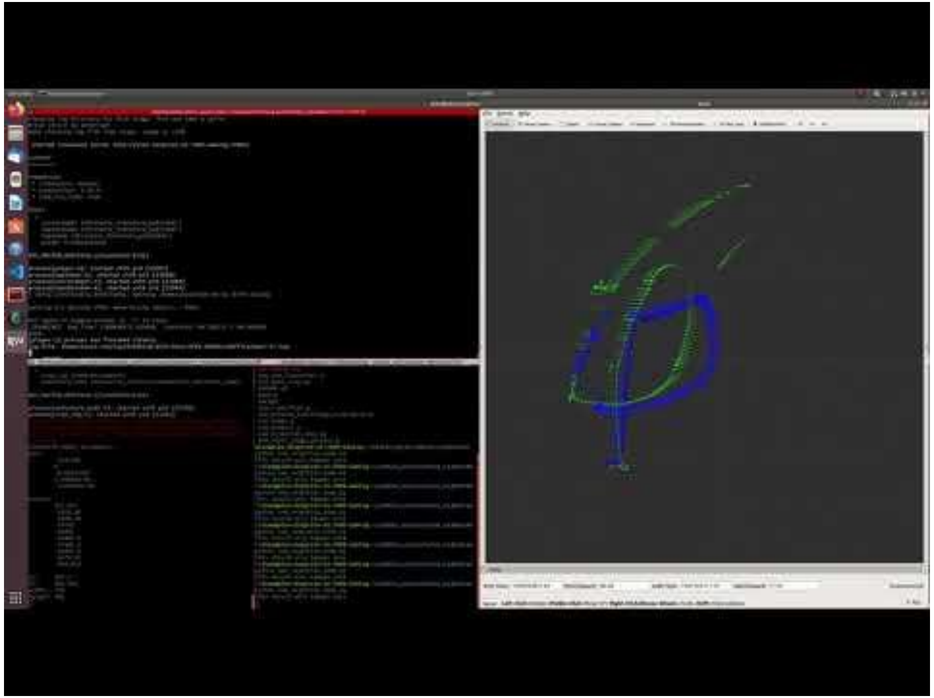


Image
Frame Hessian
Initiated
No
Initialization
Yes
Gyro /
Ground Truth Rotation
Inform
Track
KeyFrame
No
Trace Points
Yes
Trace Points
Photometric Error
BA
Points for Traces

Rotation Informed Tracking

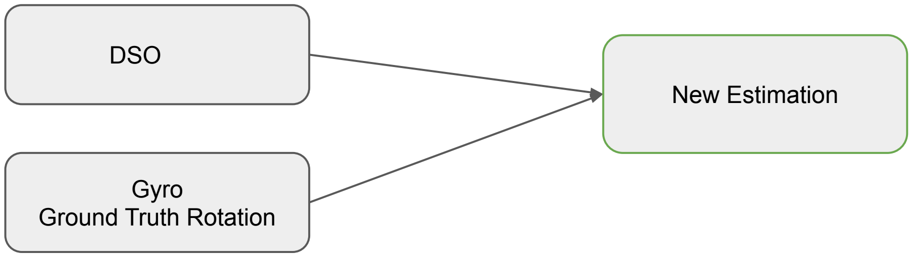
- Idea:
  - Use onboard gyroscope as a source of accurate rotation.
  - Use this value as the initialization point for the optimization problem in VO.
- Video Notation
  - Green as Optitrack Groundtruth
  - Blue as VO estimated odometry.



EKF - A last (really) little bit improvement

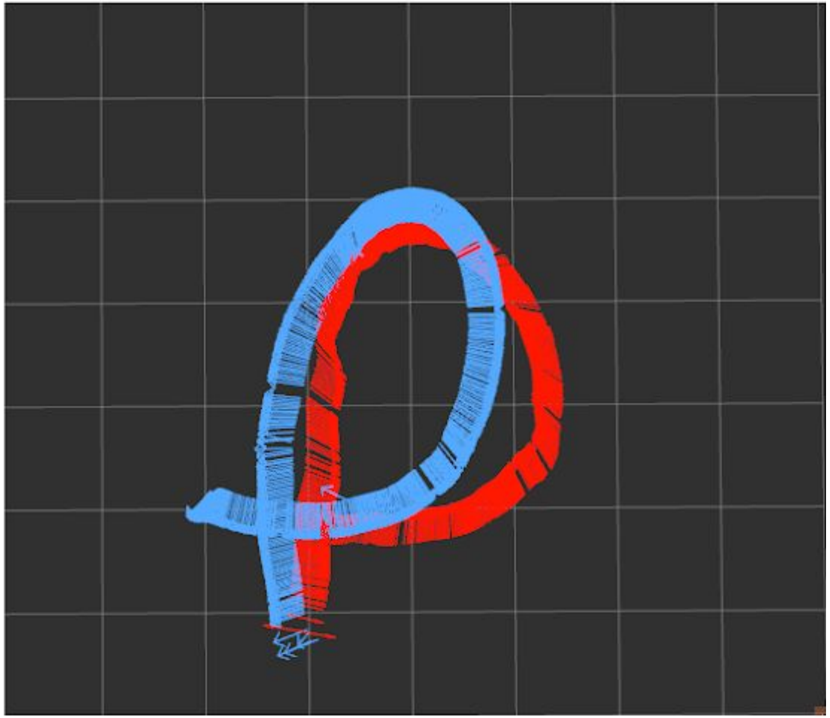
EKF on VO odometry and the rotation (Gyro or Ground Truth)

- Does not really help with the pose (trajectory plot remains same)
- Helped the orientation a little bit, not much



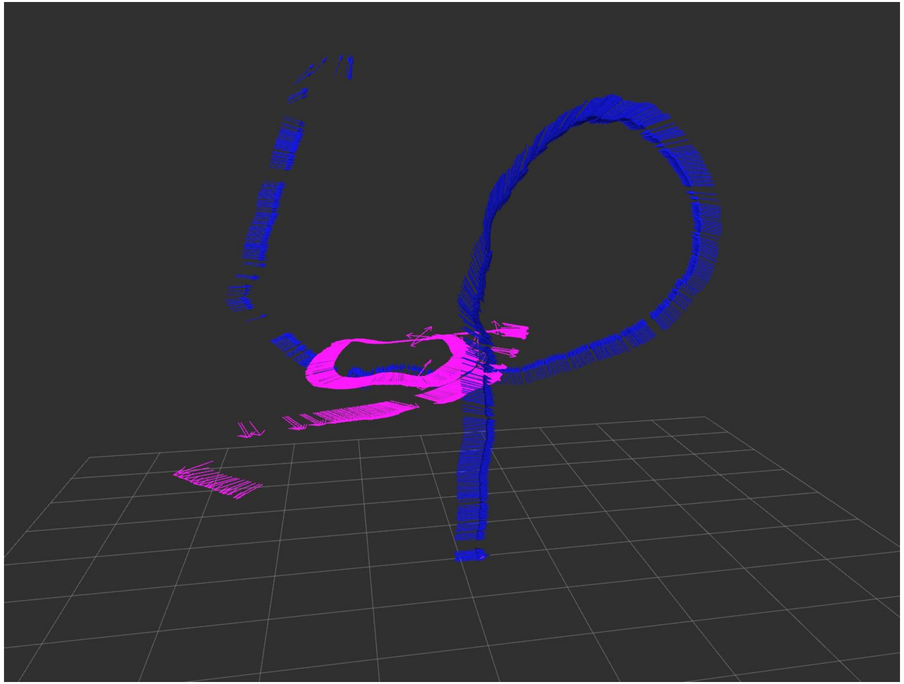
Visualization

To record, and compare the trajectory we estimated against the ground truth provided by the optitrack



Trajectory Alignment

- Why:
  - Absolute Ground Truth vs Relative VO
  - Optitrack orientation dependent on the pose of the reflective ball (may not be the direction that the blimp is facing)
  - VO initial direction might be arbitrary
  - Monocular VO run with an arbitrary scale
- Therefore we need to:
  - To align VO estimation with Ground truth Optitrack
- In real time operation
  - Use IMU to compensate for this difference



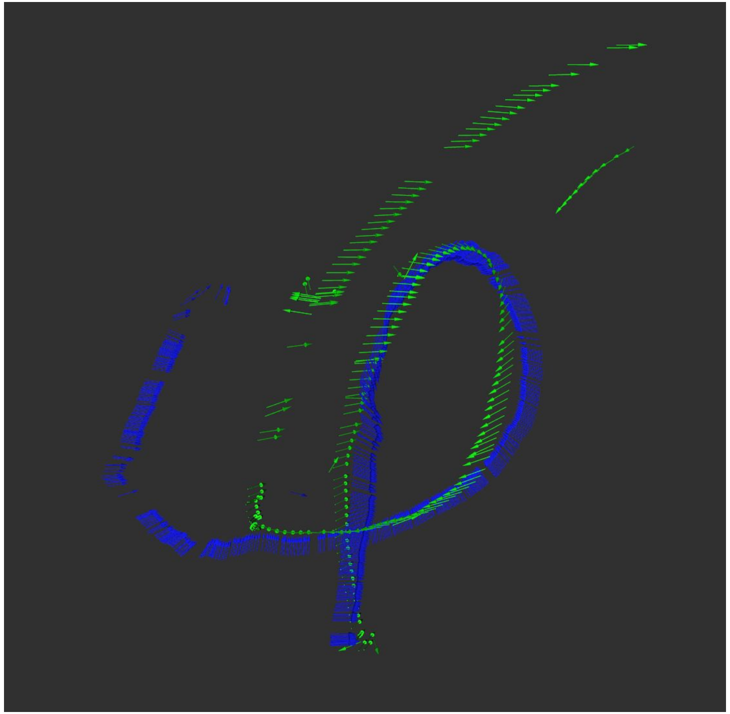
Purple: Optitrack Raw Pose; Blue: DSO Estimated

Trajectory Alignment

We know:

1. It is a SIM(3) transformation
2. It is doing great on straight lines (approximately first 100-200 points)

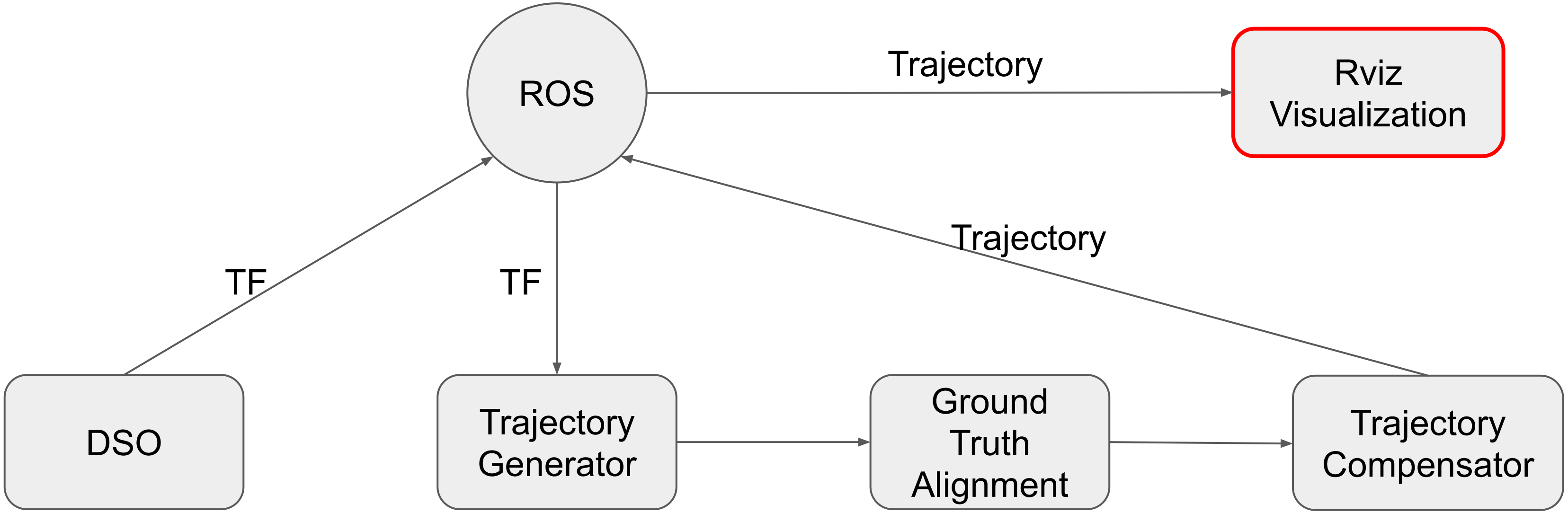
An optimization problem-
Fit first 200 points using transformation with least MSE (Euclidean Distance)



estimation

optitrack

Visualization



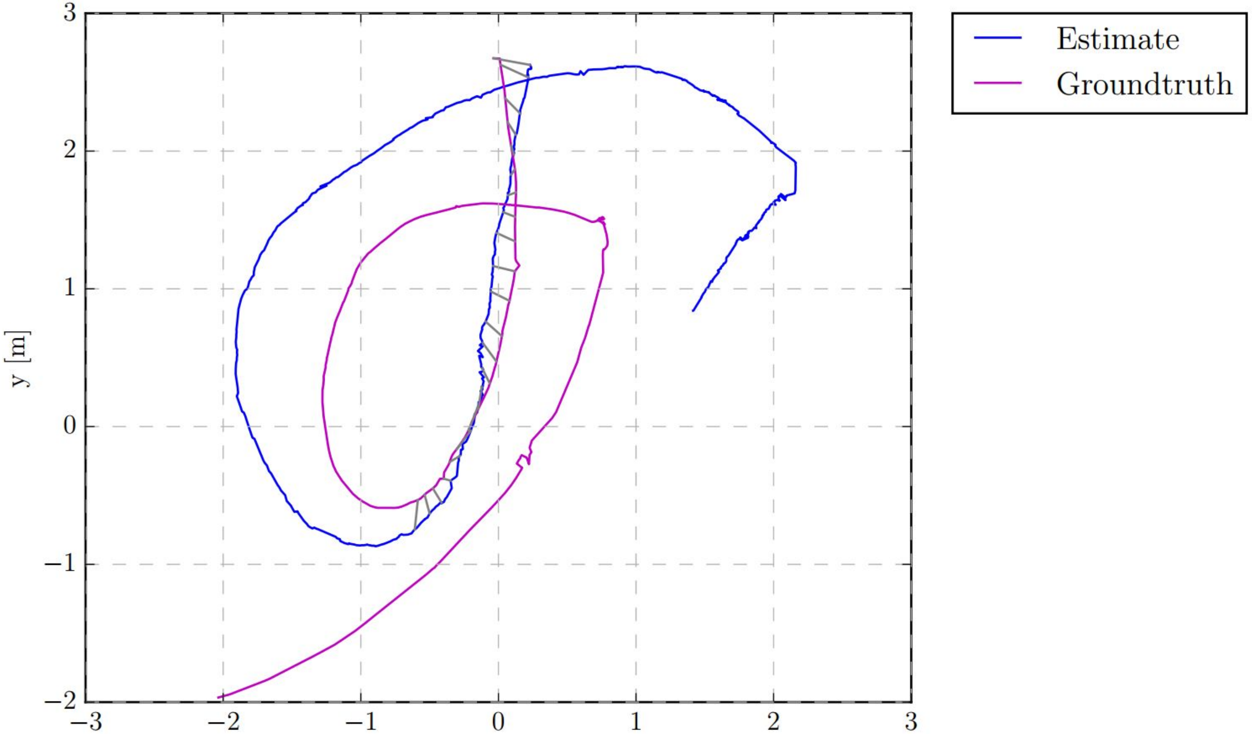
Benchmarking

To evaluate the performance of the system, and to find out which improvement could be more efficient

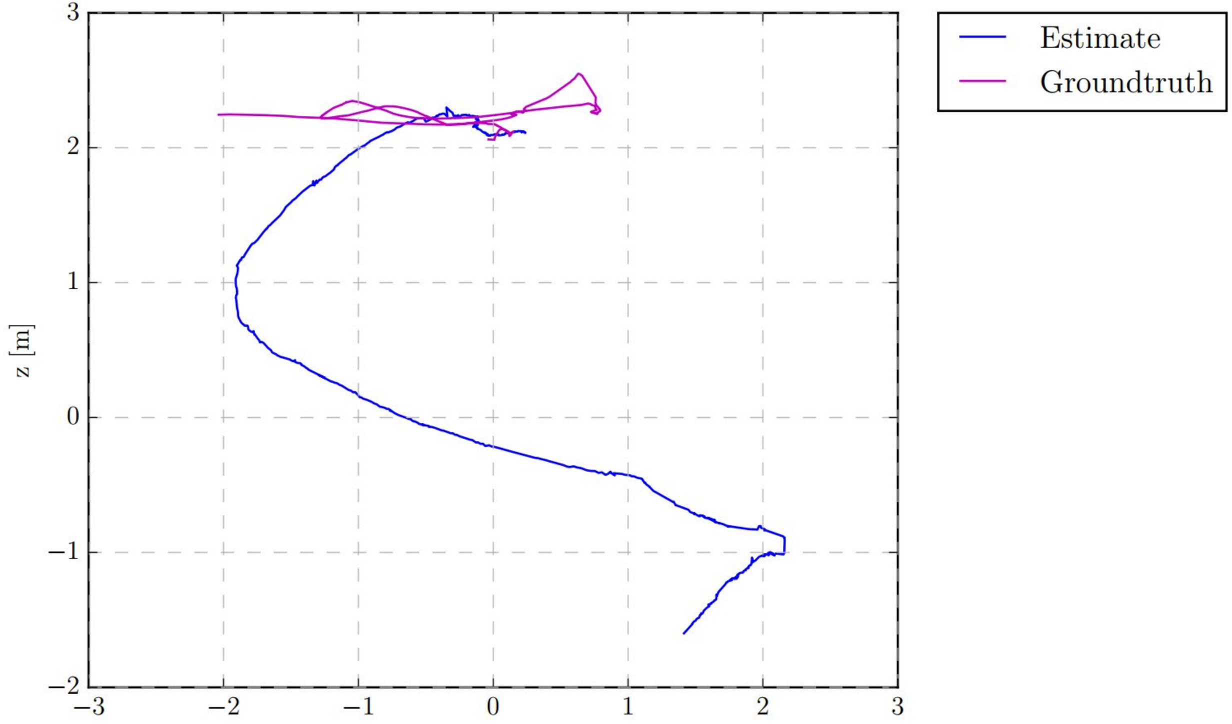
We are evaluating:

1. Drifts
2. Scale errors

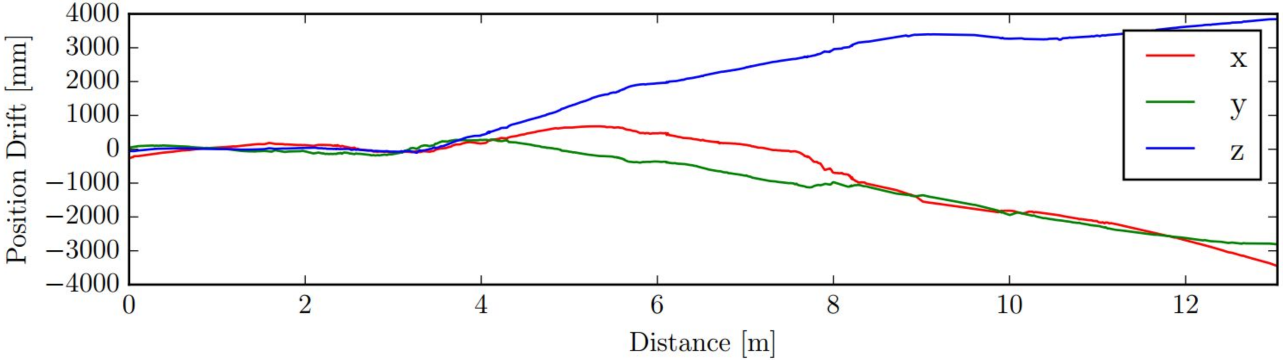
Benchmarking

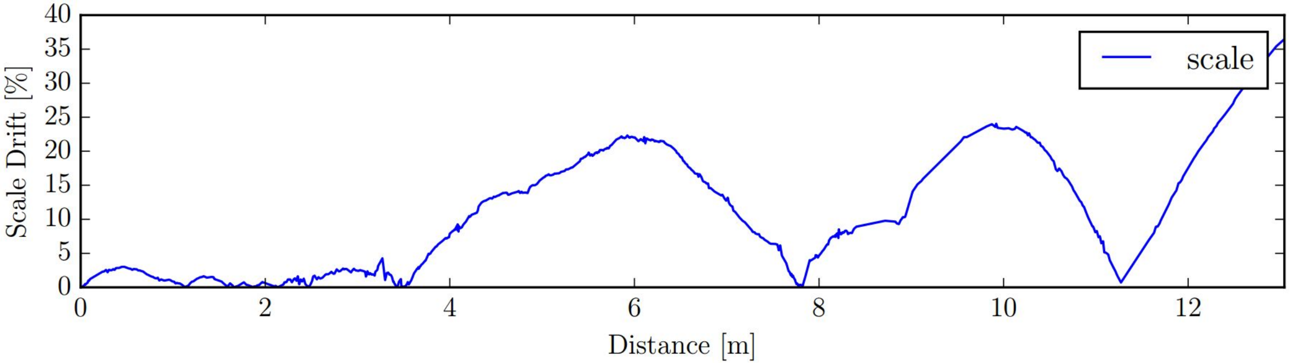


Benchmarking



Benchmarking





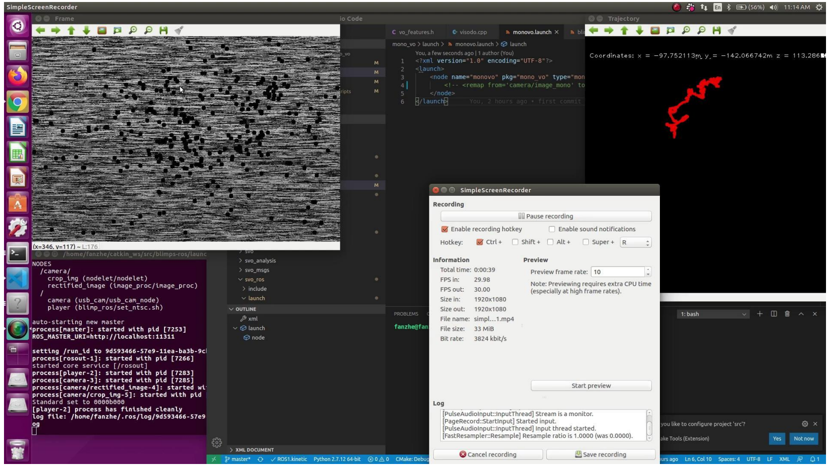
Lesson Learned

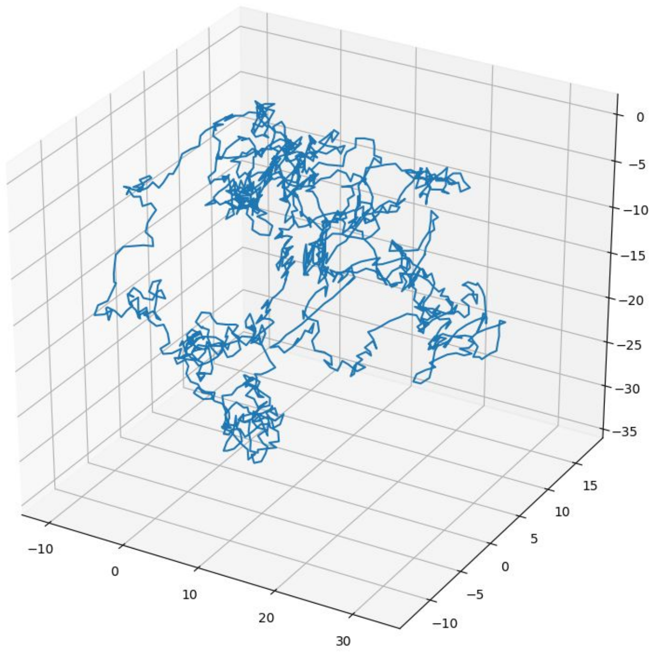
1. Images
    a. When dealing with random noises, assume gaussian is always a good place to start.
    b. Machine learning algorithms would require a good conversion of data for it to learn a result.
        i. Random noise
2. VO
    a. Indirect methods work relatively worse in noisy and low resolution image streams.
    b. In situations where the global minimum of the optimization is hard to reach
        i. Either by the limited number of features
        ii. Or by the hard triangulation problem given by large rotation + small translation
        iii. Introducing a ground truth rotation helps solve the problem.

Possible Future Work

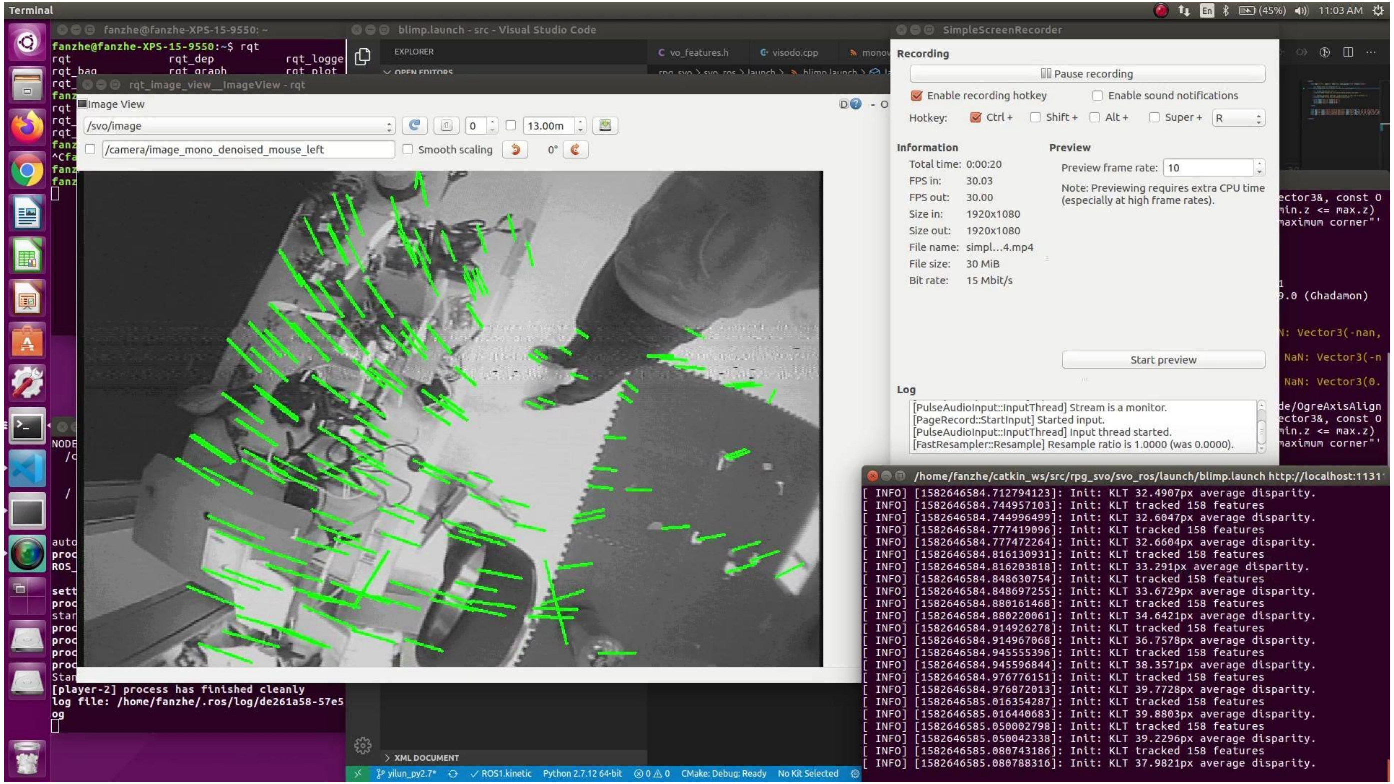
1. VI-DSO: include IMU in the BA of DSO
   a. To help adjust scale problem
   b. And the spiral downwards
2. Loop-Closure: building a Map
3. Utilize labels such as AprilTag to help compensate for DSO drift.
4. Applications:
   a. Navigations
   b. Swarms

Appendix: Failures of ORB, SVO, Vanilla VO





Appendix: Failures of ORB, SVO, Vanilla VO



Appendix: Failures of ORB, SVO, Vanilla VO