The Andrew and Erna Viterbi Faculty of ELECTRICAL & COMPUTER ENGINEERING

VISL Vision and Image Sciences Laboratory

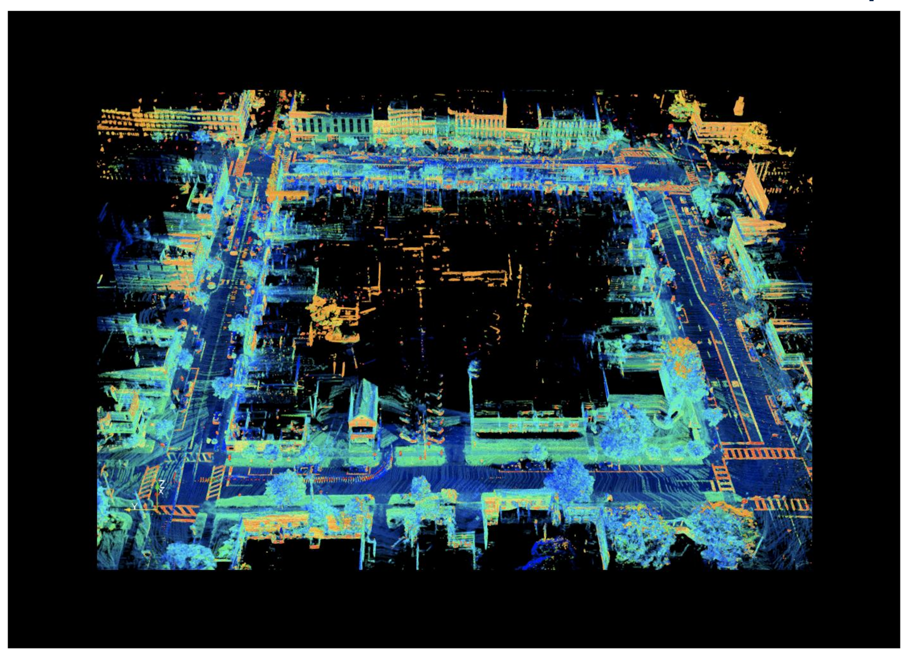
TECHNION Israel Institute of Technology

Indoor Navigation with Camera

Izar Hasson and Raz Shemesh, Supervised by Chen Katz

Introduction

- Simultaneous Localization and Mapping (SLAM) is a computational process that enables a system to map its environment while simultaneously determining its own location within that map.
- Monocular SLAM systems, such as ORB-SLAM3, are limited by their inability to determine absolute scale and true location on the map.



SLAM used on city roads

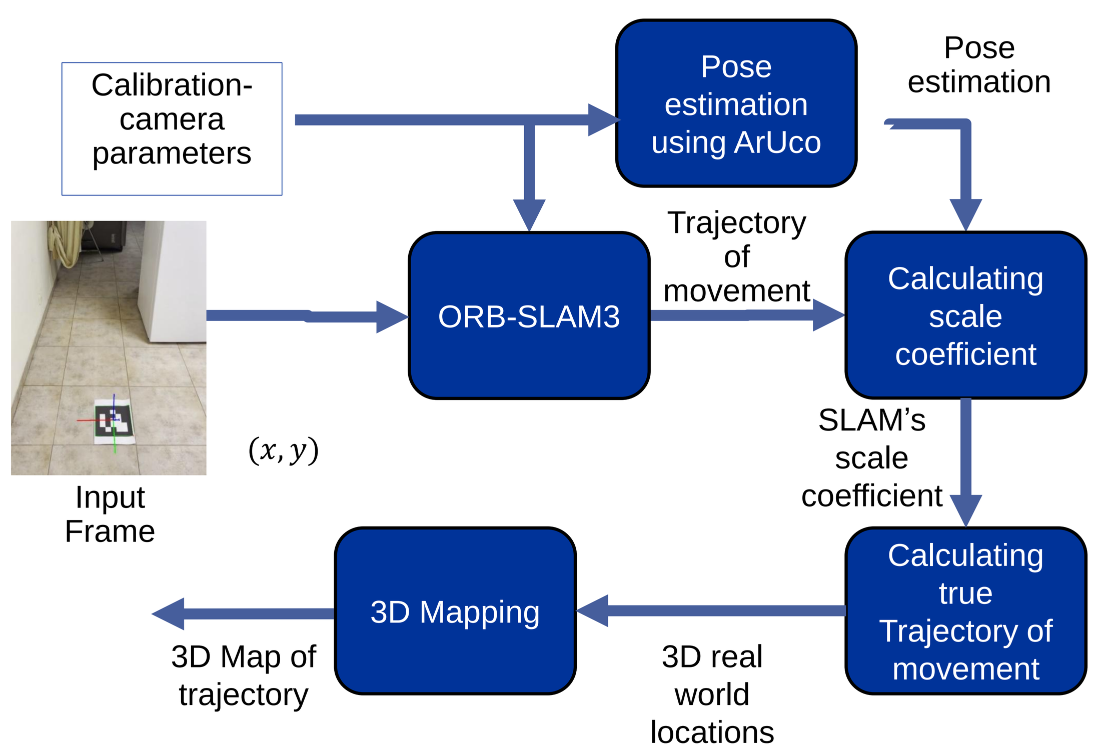
Goals

- Estimate and 3D map the true location of the camera based on a video
  - Monocular visual camera provides input
  - Robust treatment for varying conditions
  - Computationally efficient algorithm

Challenges

- Monocular ORB-SLAM3 inability to determine absolute scale
- Monocular ORB-SLAM3 inability to determine true location on the map
- Camera movement
- Environment with low amount of unique features

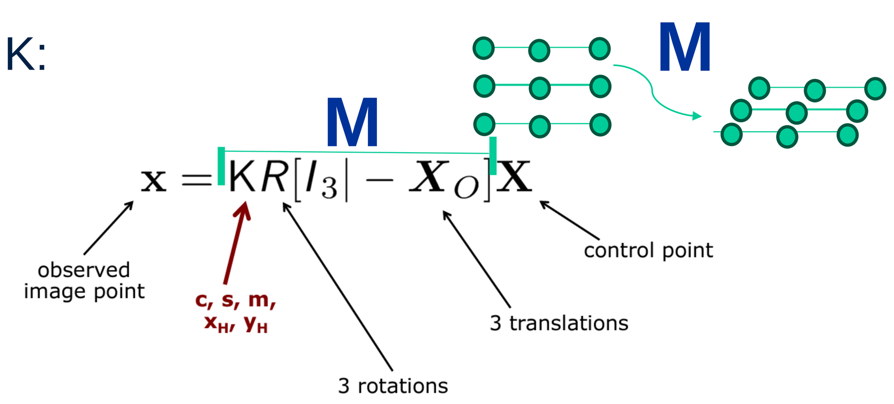
3D Mapping the trajectory



- The inputs are:
  - camera parameters from calibration
  - The exact locations of the ArUco markers, one at start and one at end of track
  - Video of moving in an environment

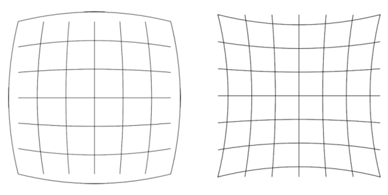
Calibration

- Purpose: Camera calibration estimates the intrinsic and extrinsic parameters of a specific camera to correct image distortions and accurately map 3D world points to 2D image points.
- Find K:



- Assumption: we know the shape and size a chess board
  - We will identify the corners of the chessboard by a simple corner detection algorithm
  - Set corner of the board to be world center
  - Distance of each corner from the other is known, that's how we get the real points
  - With enough pairs, solve using SVD
- will not be able to correct non-linear phenomena such as radial distortion
  - That's why we will also use numerical methods of minimizing the reprojection error

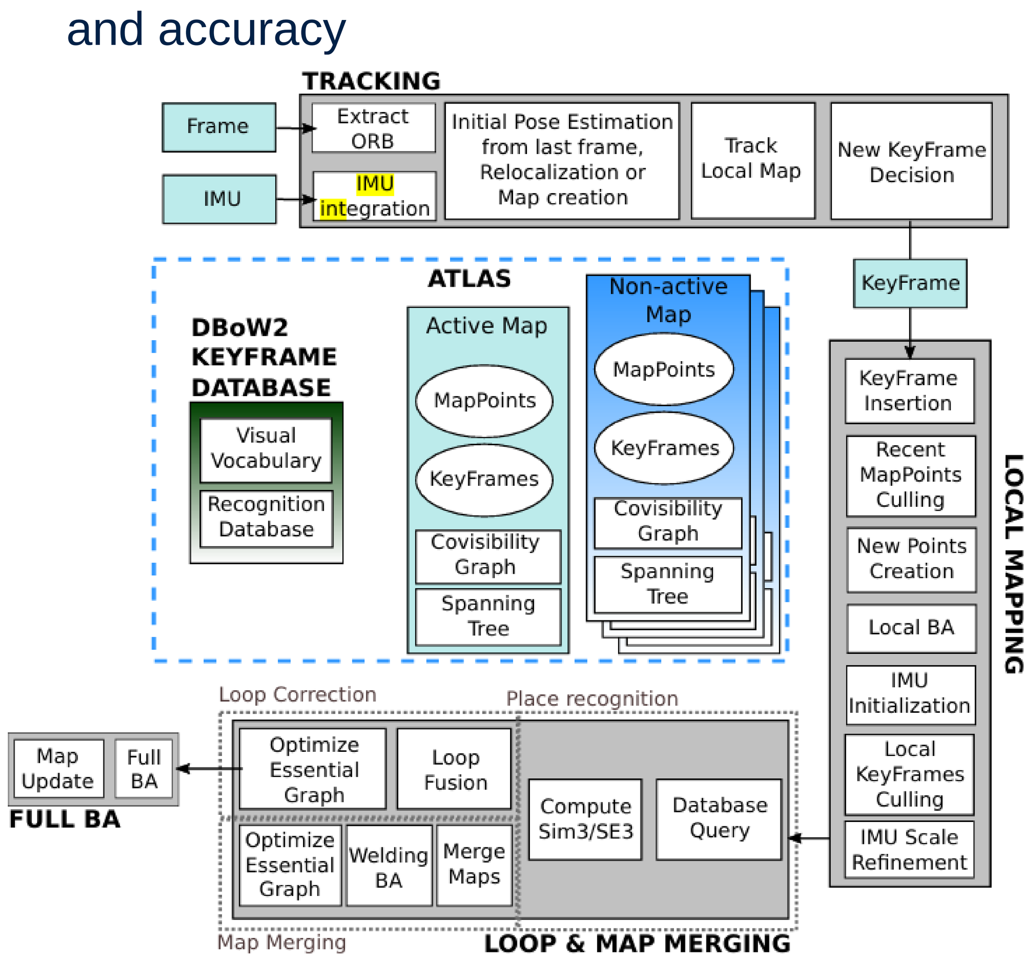
$$\min_{(K,\boldsymbol{q},R_n,\mathbf{t}_n)} \sum_n \sum_i \|\mathbf{x}_{ni} - \hat{\mathbf{x}}(K,\mathbf{q},R_n,\mathbf{t}_n,\mathbf{X}_{ni})\|^2$$

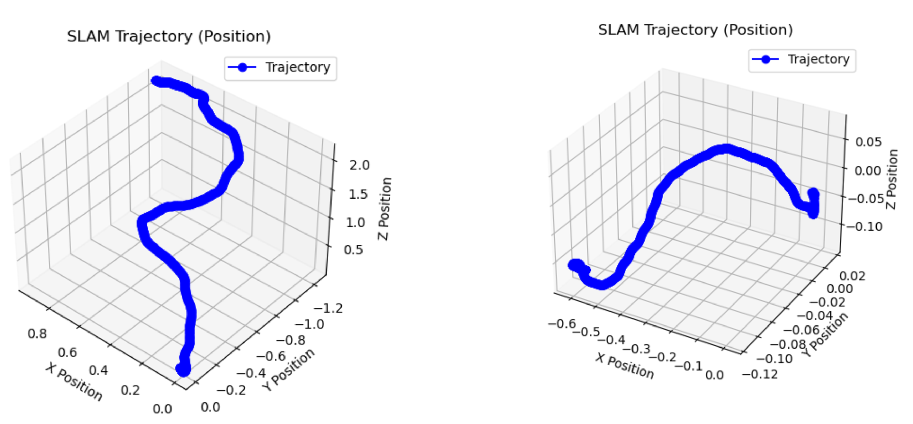


...linearize to obtain a quadratic function, compute derivative, set it to 0, solve linear system, iterate...
(solved using Levenberg-Marquardt, K by Zhang's m. as initial value)

ORB-SLAM3

- ORB features for efficient keypoint detection and description, ensuring robust tracking under varying conditions
- Constructs a sparse map using selected keyframes, balancing computational efficiency and accuracy

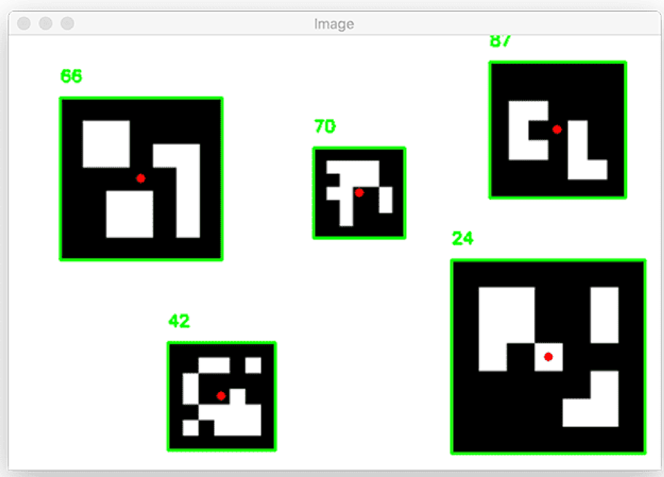




Trajectory for two different settings

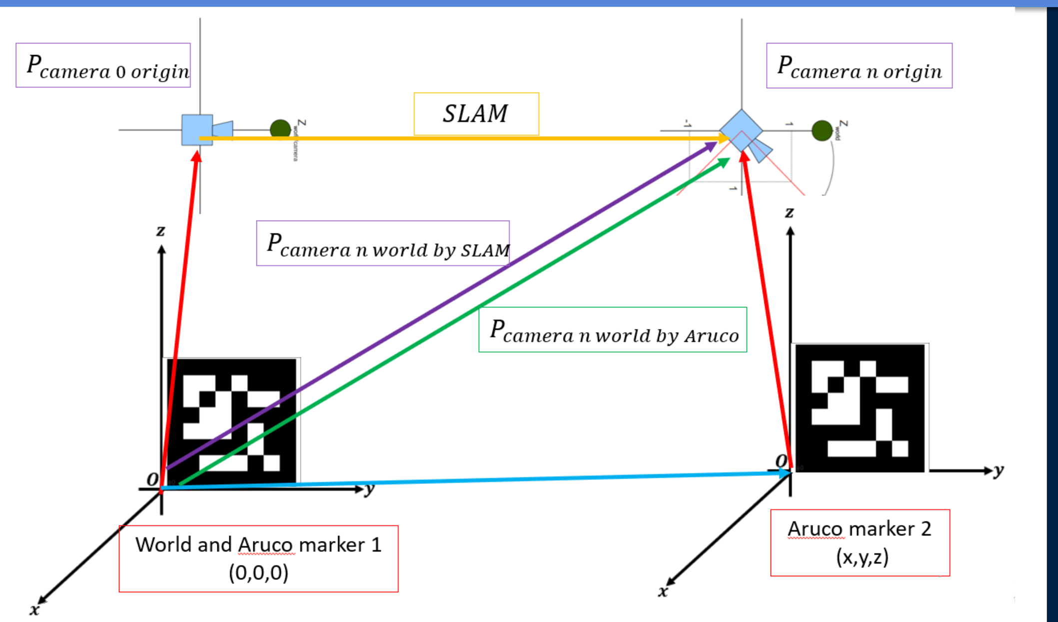
Pose Estimation by ArUco

- Detect and determine marker ID



- Assumption: we know the shape and size of the marker, like in Calibration find matrix M using PnP algorithm

Finding Scale Coefficient



- From ArUco we get the green arrow (all known):

$$P_{camera\_n\_\boldsymbol{ArUco}} = -{R_{world}^{aruco_n}}^T t_{world}^{aruco_n} + P_{marker\ 2}$$

- From ORB-SLAM3 we get the purple arrow:

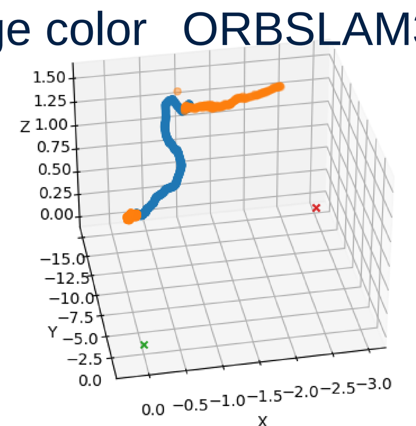
$$P_{camera\_n\_\boldsymbol{SLAM}} = R_{0\_aruco}^T \cdot \left(\alpha \cdot\ t_{camera_n}^{camera_0} - t_{0_{aruco}}\right)$$

- Equalize and find α which is the scale ambiguity

Results

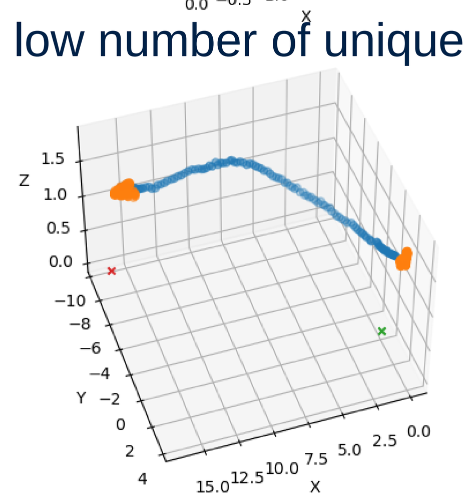
- Two experiments:
- First: high number of unique features:

ArUco: orange color ORBSLAM3:blue color



Markers marked with X

- Second: low number of unique features:



Conclusions

- By combining ArUco markers with the ORB-SLAM3 algorithm the system addresses the limitations of monocular SLAM
- achieving high accuracy in various scenarios
- This hybrid solution demonstrates the potential of integrating technologies for precise mapping and tracking in dynamic environments

January 2025