Visual Inertial Slam

Kevin Shin
Department of Electrical and Computer Engineering
University of California, San Diego
San Diego, USA
d3shin@ucsd.edu

***Abstract*—This project focuses on the implementation of a Visual-Inertial Simultaneous Localization and Mapping (SLAM) system using an Extended Kalman Filter (EKF) framework. The goal is to estimate the pose of a robot equipped with an Inertial Measurement Unit (IMU) and a stereo camera, while simultaneously mapping the surrounding environment using visual landmarks. The project leverages data collected from Clearpath Jackal robots navigating MIT's campus, which includes IMU measurements (linear and angular velocities), stereo camera images, and precomputed visual feature correspondences. The intrinsic and extrinsic calibrations of the sensors are provided to facilitate accurate sensor fusion.**

The project is divided into four main tasks: (1) IMU localization via EKF prediction, where the robot's pose is estimated using SE(3) kinematics and IMU measurements; (2) optional feature detection and matching, where visual features are detected, tracked, and matched across stereo images and over time; (3) landmark mapping via EKF update, where the positions of visual landmarks are estimated using stereo-camera observations; and (4) visual-inertial SLAM, where the IMU prediction and landmark update steps are combined to achieve a complete SLAM system. The final system is tuned to balance noise and computational complexity, ensuring accurate trajectory estimation and landmark mapping.

The report includes a detailed technical approach, results with visualizations of the estimated IMU trajectory and landmark positions, and a discussion of the challenges encountered and solutions implemented. The project demonstrates the integration of visual and inertial data for robust SLAM, highlighting the importance of sensor fusion in robotics for autonomous navigation and mapping.

***Index Terms*—Visual-Inertial SLAM, Extended Kalman Filter, Sensor Fusion, Pose Estimation, Landmark Mapping, Feature Tracking, IMU Localization**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in robotics, enabling autonomous systems to navigate and map unknown environments in real time. This project focuses on Visual-Inertial SLAM, which combines data from a stereo camera and an Inertial Measurement Unit (IMU) to estimate a robot's pose and reconstruct its surroundings. Visual data provides rich information about the environment, while IMU measurements offer high-frequency motion estimates, making the fusion of these sensors a powerful approach for robust and accurate SLAM.

The goal of this project is to implement a Visual-Inertial SLAM system using an Extended Kalman Filter (EKF). The EKF framework is well-suited for this task, as it allows for the integration of noisy sensor measurements while maintaining estimates of both the robot's state and the positions of environmental landmarks. Using real-world data collected from Clearpath Jackal robots navigating MIT's campus, this project addresses key challenges such as IMU-based pose prediction, visual feature detection and tracking, and landmark mapping. By combining these components, the system achieves accurate localization and mapping, even in dynamic and uncertain environments.

Problem Formulation

The problem addressed in this project is **Visual-Inertial SLAM**, which involves estimating the pose (position and orientation) of a robot while simultaneously mapping the surrounding environment using data from a stereo camera and an IMU. The robot navigates through an unknown environment, and the goal is to fuse the high-frequency motion estimates from the IMU with the rich visual information from the stereo camera to achieve accurate localization and mapping.

Inputs:

- **IMU Measurements**: Linear velocity $v_t \in \mathbb{R}^3$ and angular velocity $\omega_t \in \mathbb{R}^3$ in the body frame of the IMU.
- **Stereo Camera Measurements**: Grayscale images and precomputed visual feature correspondences between the left and right camera frames.
- **Calibration Data**: Intrinsic camera calibration matrices K_L and K_R for the left and right cameras, and extrinsic calibration $T_C^I \in SE(3)$ representing the transformation from the camera frame to the IMU frame.

Outputs:

- **Robot Pose**: The estimated pose $T_t \in SE(3)$ of the robot at each timestep t.
- **Landmark Positions**: The estimated 3D positions $m \in \mathbb{R}^3$ of visual landmarks in the environment.

Challenges:

- **Noisy Sensor Data**: Both IMU and camera measurements are subject to noise, requiring robust filtering techniques.
- **Computational Complexity**: The large number of landmarks and measurements necessitates efficient algorithms to manage computational resources.

Department of Electrical and Computer Engineering, University of California, San Diego

- **Sensor Fusion**: Combining asynchronous and heterogeneous data from the IMU and stereo camera into a unified framework.

High-Level Overview

The system is implemented using an **Extended Kalman Filter (EKF)** framework, which is well-suited for fusing noisy sensor data and estimating both the robot's pose and landmark positions. The process is divided into two main steps:

1) **Prediction Step**: The robot's pose is predicted using IMU measurements and SE(3) kinematics. This step propagates the state estimate forward in time based on the motion model.
2) **Update Step**: The predicted pose is corrected using visual observations from the stereo camera. Landmark positions are estimated by triangulating features from the stereo images and updating the state using the EKF update equations.

The system is designed to handle the challenges of noisy data and computational complexity by:

- Using sparse matrices for efficient landmark covariance management.
- Filtering out unreliable features to improve the observation model.
- Tuning noise parameters to balance accuracy and computational efficiency.

Preliminaries

This section provides the mathematical background necessary to understand the Visual-Inertial SLAM problem and the proposed solution. Key concepts include SE(3) kinematics, the Extended Kalman Filter (EKF), and stereo vision triangulation.

1. SE(3) Kinematics

The Special Euclidean Group $SE(3)$ represents the set of rigid body transformations in 3D space, which include both rotation and translation. A transformation $T \in SE(3)$ can be expressed as:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix},$$

where $R \in SO(3)$ is a rotation matrix and $t \in \mathbb{R}^3$ is a translation vector. The Lie algebra $\mathfrak{se}(3)$ associated with $SE(3)$ is used to represent twists, which describe motions:

$$\xi = \begin{bmatrix} \omega \\ v \end{bmatrix},$$

where $\omega \in \mathbb{R}^3$ is the angular velocity and $v \in \mathbb{R}^3$ is the linear velocity. The exponential map $\exp : \mathfrak{se}(3) \to SE(3)$ converts a twist to a transformation:

$$\exp(\xi^\wedge) = \begin{bmatrix} \exp([\omega]_\times) & A(\omega)v \\ 0 & 1 \end{bmatrix},$$

where $[\omega]_\times$ is the skew-symmetric matrix of ω, and $A(\omega)$ is a matrix function that depends on ω.

2. Extended Kalman Filter (EKF)

The EKF is a recursive state estimation algorithm used to estimate the state of a nonlinear system from noisy measurements. The state x_t and its covariance P_t are updated in two steps:

- **Prediction Step**:

$$\hat{x}_{t|t-1} = f(x_{t-1}, u_t),$$

$$P_{t|t-1} = F_t P_{t-1} F_t^T + Q_t,$$

where $f(\cdot)$ is the motion model, u_t is the control input, F_t is the Jacobian of $f(\cdot)$, and Q_t is the process noise covariance.

- **Update Step**:

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1},$$

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(z_t - h(\hat{x}_{t|t-1})),$$

$$P_t = (I - K_t H_t) P_{t|t-1},$$

where $h(\cdot)$ is the observation model, H_t is the Jacobian of $h(\cdot)$, z_t is the measurement, and R_t is the observation noise covariance.

3. Stereo Vision Triangulation

Stereo vision uses two cameras to estimate the 3D position of a point by triangulating its projections in the left and right images. Given the pixel coordinates (u_L, v_L) and (u_R, v_R) of a point in the left and right images, respectively, the 3D position $m = (X, Y, Z)$ can be computed as:

$$Z = \frac{f \cdot b}{d},$$

$$X = \frac{Z \cdot (u_L - c_x)}{f},$$

$$Y = \frac{Z \cdot (v_L - c_y)}{f},$$

where f is the focal length, b is the baseline (distance between the cameras), $d = u_L - u_R$ is the disparity, and (c_x, c_y) is the principal point of the camera.

4. Skew-Symmetric Matrix

The skew-symmetric matrix $[\omega]_\times$ of a vector $\omega = [\omega_x, \omega_y, \omega_z]^T$ is defined as:

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

This matrix is used to represent cross products in matrix form, e.g., $\omega \times v = [\omega]_\times v$.

5. Projection and Observation Model

The projection of a 3D point $m = (X, Y, Z)$ onto the image plane is given by:

$$u = \frac{f_x X + c_x Z}{Z},$$

$$v = \frac{f_y Y + c_y Z}{Z},$$

where (f_x, f_y) are the focal lengths and (c_x, c_y) are the principal points. The observation model $h(\cdot)$ maps the 3D landmark positions and robot pose to the observed pixel coordinates.

6. Sparse Matrices

To handle the large number of landmarks efficiently, sparse matrix representations are used for the landmark covariance matrix. This reduces memory usage and computational complexity, especially during matrix inversions and multiplications.

Technical Approach

1. IMU Localization via EKF Prediction

The IMU provides high-frequency measurements of linear and angular velocity, which are used to predict the robot's motion over time. This prediction step is crucial because it allows the system to estimate the robot's pose between camera updates, which occur at a lower frequency. Without this step, the robot's pose estimate would rely solely on visual data, leading to poor performance in fast-moving or visually degraded environments.

Formulation: The motion model is defined as:

$$T_{t+1} = T_t \cdot \exp\left(\begin{bmatrix} [\omega_t]_\times & v_t \\ 0 & 0 \end{bmatrix} \Delta t\right),$$

where $[\omega_t]_\times$ is the skew-symmetric matrix of the angular velocity, and Δt is the time step. The covariance P_t is updated using the process noise model:

$$P_{t+1} = F_t P_t F_t^T + Q_t,$$

where F_t is the Jacobian of the motion model, and Q_t is the process noise covariance.

2. Landmark Mapping via EKF Update

The stereo camera provides visual observations of the environment, which are used to correct the robot's pose estimate and map the positions of landmarks. This update step is essential because it reduces the drift in the robot's pose estimate caused by the accumulation of errors in the IMU prediction step. By fusing visual data with IMU data, the system achieves more accurate and robust localization and mapping.

Formulation: The observation model maps the 3D landmark positions m and robot pose T_t to the observed pixel coordinates z_t:

$$z_t = h(T_t, m) + \eta_t,$$

where $h(\cdot)$ is the projection function, and η_t is the observation noise. The EKF update equations are:

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1},$$

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(z_t - h(\hat{x}_{t|t-1})),$$

$$P_t = (I - K_t H_t) P_{t|t-1},$$

where H_t is the Jacobian of the observation model, and R_t is the observation noise covariance.

3. Stereo Vision Triangulation

Stereo vision triangulation is used to estimate the 3D positions of landmarks from their projections in the left and right camera images. This step is critical because it provides the geometric information needed to map the environment and correct the robot's pose estimate. Without accurate triangulation, the system would be unable to build a consistent map or localize the robot effectively.

Formulation: Given the pixel coordinates (u_L, v_L) and (u_R, v_R) of a point in the left and right images, respectively, the 3D position $m = (X, Y, Z)$ is computed as:

$$Z = \frac{f \cdot b}{d},$$

$$X = \frac{Z \cdot (u_L - c_x)}{f},$$

$$Y = \frac{Z \cdot (v_L - c_y)}{f},$$

where f is the focal length, b is the baseline, and $d = u_L - u_R$ is the disparity.

4. Sparse Matrix Representation

The landmark covariance matrix is typically large and sparse, as most landmarks are not observed in every frame. Using sparse matrix representations reduces memory usage and computational complexity, making the system more efficient. This is especially important for real-time applications, where computational resources are limited.

The covariance matrix P_t is stored using the `scipy.sparse.lil_matrix` class, which is efficient for incremental updates. The Kalman gain computation and covariance update are performed using sparse matrix operations.

5. Feature Filtering

Not all visual features are reliable for mapping. Some features may be noisy, occluded, or too far away to provide useful information. Filtering out unreliable features improves the accuracy of the observation model and reduces computational complexity. This step ensures that only high-quality features are used for mapping and localization.

Features are filtered based on depth and distance from the robot. Landmarks with depth Z outside the range $[0.1, 10.0]$ are discarded, and landmarks far from the robot's trajectory are ignored.

6. Noise Tuning

The performance of the EKF depends heavily on the choice of process and observation noise parameters. Tuning these parameters is essential to achieve a balance between accuracy and computational efficiency. Poorly tuned noise parameters can lead to overconfident or underconfident estimates, degrading the system's performance.

The noise parameters are tuned empirically to achieve the best results. The process noise covariance Q_t and observation noise covariance R_t are adjusted based on the characteristics of the IMU and camera data.

II. Results

A. EKF Motion Model

The results show the estimated trajectory of the robot over time for each dataset.

The EKF motion model successfully estimates the robot's trajectory for all three datasets. However, some drift is observed over time due to the accumulation of errors in the IMU measurements. This drift is more pronounced in Dataset 3, where the robot undergoes more complex motions. The results highlight the importance of the update step, which corrects the pose estimate using visual observations.

B. Landmark Visualization

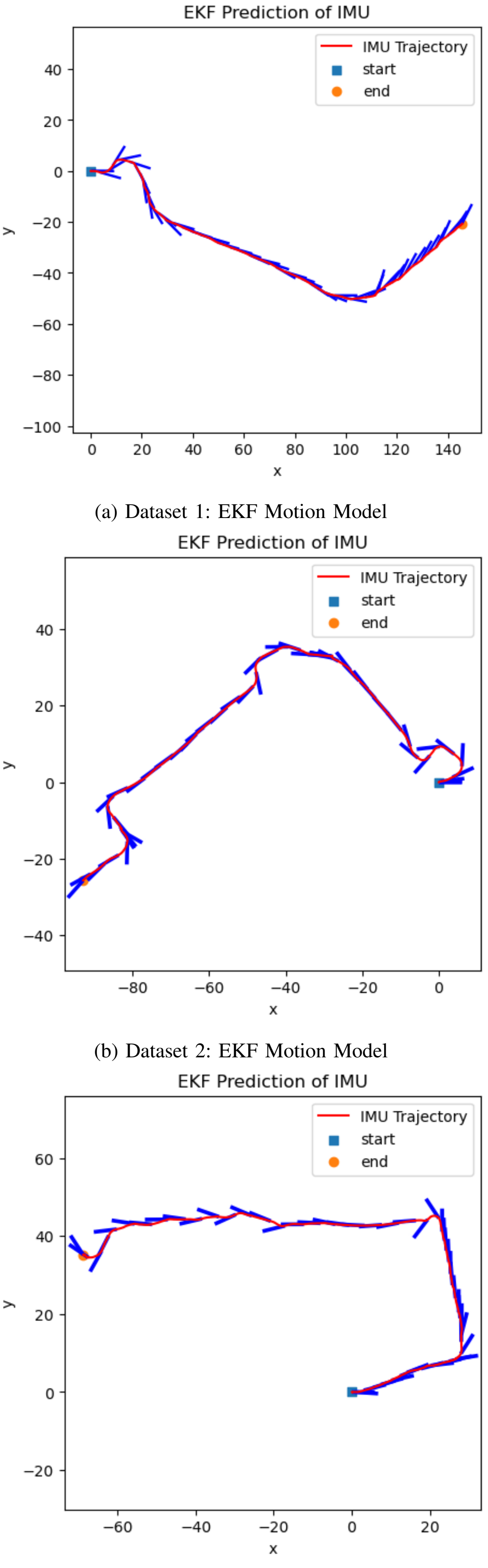
The landmark mapping component estimates the 3D positions of visual landmarks using stereo camera observations. The results show the estimated landmark positions and their relationship to the robot's trajectory for each dataset.

Discussion: The landmark mapping results demonstrate the system's ability to reconstruct the environment using stereo camera observations. The landmarks are well-localized in Datasets 1 and 2, where the environment is relatively simple and the robot's motion is smooth. In Dataset 3, some landmarks appear less accurate due to the robot's complex motion and the presence of occlusions.

C. Conclusion

By fusing data from an IMU and a stereo camera, the system demonstrated robust and accurate performance in localizing the robot and reconstructing the surrounding environment. Key contributions and findings of the project include:

- **EKF Motion Model**: The motion model effectively integrated IMU data (linear and angular velocity) to predict

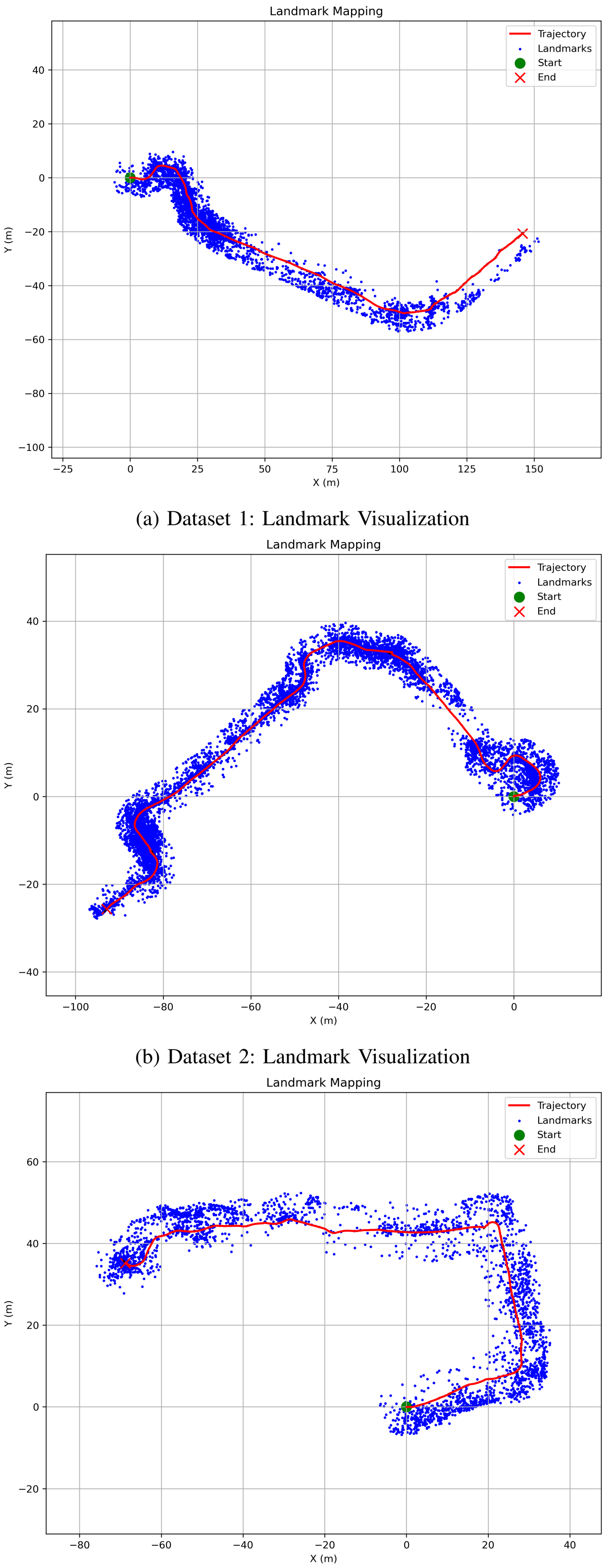


(a) Dataset 1: EKF Motion Model

(b) Dataset 2: EKF Motion Model

(c) Dataset 3: EKF Motion Model

Fig. 1: Estimated robot trajectories using the EKF motion model for Datasets 1, 2, and 3.



(a) Dataset 1: Landmark Visualization

(b) Dataset 2: Landmark Visualization

(c) Dataset 3: Landmark Visualization

Fig. 2: Estimated landmark positions and robot trajectories for Datasets 1, 2, and 3.

the robot's pose over time. Using SE(3) kinematics, the system was able to handle the robot's complex motions and provide a reliable initial estimate of the trajectory. However, some drift was observed over time due to the accumulation of errors in the IMU measurements, highlighting the importance of the update step.

- **Landmark Mapping**: The landmark mapping component leveraged stereo camera observations to estimate the 3D positions of visual landmarks. By triangulating features from the left and right camera images, the system was able to build a consistent map of the environment. Feature filtering and noise tuning were critical to improving the accuracy of the landmark estimates.
- **Sensor Fusion**: The fusion of IMU and stereo camera data enabled robust and accurate localization and mapping. The IMU provided high-frequency motion updates, while the stereo camera offered precise geometric corrections, demonstrating the complementary nature of these sensors.
- **Computational Efficiency**: The use of sparse matrix representations for the landmark covariance matrix significantly reduced memory usage and computational complexity, making the system suitable for real-time applications.
- **Results**: The system achieved accurate trajectory estimation and landmark mapping across multiple datasets. While the results were strong for simple and smooth motions, challenges remained in environments with complex motions or occlusions, underscoring the need for further improvements in feature tracking and noise tuning.

In conclusion, this project highlights the effectiveness of the EKF framework for Visual-Inertial SLAM. The integration of IMU and stereo camera data, combined with efficient computational techniques, provides a general solution for autonomous navigation and mapping in real-world environments.

Acknowledgment

I would like to thank the University of California, San Diego (UCSD) Electrical and Computer Engineering (ECE) Department and the teaching staff of **ECE 276A: Sensing & Estimation in Robotics** for their guidance and support.

Instructor: Nikolay Atanasov (natanasov@ucsd.edu)
Teaching Assistants: Yinzhuang Yi (yiyi@ucsd.edu) Jason Stanley (jtstanle@ucsd.edu) Jay Paek (jpaek@ucsd.edu)

References

[1] IEEE, “IEEE conference templates,” [Online]. Available: https://www.ieee.org/conferences_events/conferences/publishing/templates.html.

[2] N. Atanasov, “ECE 276A Course Introduction,” [Online]. Available: https://natanaso.github.io/ece276a/ref/ECE276A_1_Introduction.pdf.

[3] N. Atanasov, “Motion and Observation Models,” [Online]. Available: https://natanaso.github.io/ece276a/ref/ECE276A_4_MotionAndObservationModels.pdf.

[4] N. Atanasov, “Factor Graph SLAM,” [Online]. Available: https://natanaso.github.io/ece276a/ref/ECE276A_5_FactorGraphSLAM.pdf.

[5] N. Atanasov, “Localization and Odometry,” [Online]. Available: https://natanaso.github.io/ece276a/ref/ECE276A_6_LocalizationOdometry.pdf.

[6] M. Brett, “transforms3d: Python library for 3D transformations,” [Online]. Available: https://matthew-brett.github.io/transforms3d/..

[7] GTSAM, “Georgia Tech Smoothing and Mapping (GTSAM) Library,” [Online]. Available: https://gtsam.org/.

[8] Open3D, “Open3D: A Modern Library for 3D Data Processing,” [Online]. Available: http://www.open3d.org/.

[9] NumPy, “NumPy: Fundamental package for scientific computing with Python,” [Online]. Available: https://numpy.org/.