SVIO - Stereo Visual-Inertial Odometry Using an Extended Kalman Filter

Ziling Lu
ziling.lu@mail.utoronto.ca

Sam Weinberg
sam.weinberg@mail.utoronto.ca

Abstract

This paper presents a filter-based algorithm using an EKF to perform stereo visual-inertial odometry (SVIO) for autonomous vehicles. The motion model uses IMU measurements to predict the pose of the vehicle. Stereo camera image pairs are used to extract and match features at each timestep. A correction to the predicted pose is calculated by fusing the data within an EKF. The algorithm is evaluated using a raw dataset from the KITTI Vision Benchmark Suite in a residential environment. The algorithm achieves moderate results, with clear improvements demonstrated by the fusion of the two sensors. The accuracy is limited by the formulation of the EKF, where the landmarks are projected into the global frame using the previous mean estimate. Significant improvements can be attained by performing the visual odometry step using a pose-alignment technique to obtain the frame-to-frame transform, as opposed to the global context (i.e. current frame-to-inertial frame) outlined in this paper.

1 Introduction

In recent years, robotics applications have undergone drastic improvements in multiple industries including unmanned aerial vehicles (UAVs) and autonomous cars. At the heart of these exciting new technologies is the problem of localization, estimating the state of a robot within its environment based on sensor input. Common measurement devices include the global positioning system (GPS), light detection and ranging (LiDAR), cameras, and several others. GPS is widely implemented and can achieve accuracies of 1m, yet is unreliable in urban environments where satelite signal is blocked. LiDAR is extremely accurate, but the technology can be very expensive which limits the number of projects it is viable for. Depending on the environment conditions, cameras are useful in that they can achieve decent results for a reduced price[1].

Using a camera to localize is called Visual Odometry (VO), and it can be broadly separated into two categories: monocular and stereo VO. Monocular VO is a prominent topic in research for its low cost and limited space requirements. The difficulty arises in the inability discern depth using a single camera. Stereo VO is slightly more expensive, but the addition of a second camera with a known pose in relation to the first camera enables depth inference.

This section provides an introduction and motivation for the project topic. Section 2 reviews related work in academia and how it relates to our approach. Section 3 discusses the data used to evaluate the algorithm and the reference frames used in the KITTI experimental setup. Section 4 discusses the methodology behind our feature extraction, and Section 5 formulates the equations used in the EKF. Section 6 displays the experimental results, and Section 7 draws conclusions and future improvements for the project.

2 Related Work

As mentioned, visual-inertial pose estimation is an extremely active area of research due to the numerous appealing properties of the sensors. Inertial based approaches are useful since the sensors are cheap and operate at a high frequency, which enables them to capture sudden changes in movement. Propagating the position forward in time however requires integrating uncertain measurements, leading to unbound error. Visual based methods are used both independently to solve pointcloud-alignment problems and as a corrective measurement for IMU data. Additionally, a major current research topic is simultaneous localization and mapping (SLAM), which is especially

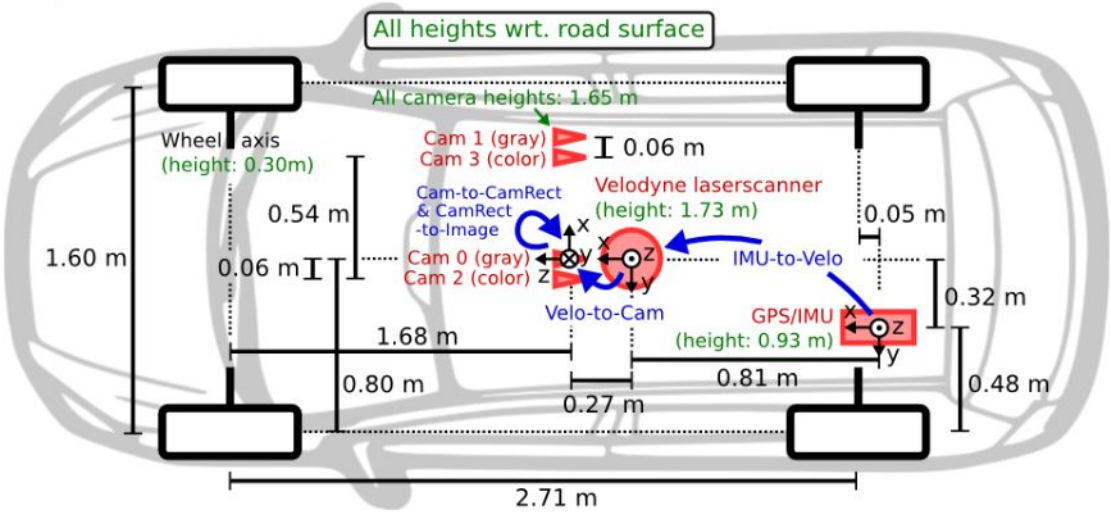

Figure 1: Overhead view of KITTI experimental setup with sensor frame transformations.

useful in unknown environments where we would like to create a map of the area. There are many uses of stereo visual inertial odometry SLAM in literature including state-of-the-art models such as ORB-SLAM[2] and DSO[3]. For this project we focus on localization, using landmarks temporarily and discarding them afterward.

The visual-inertial sensor fusion approaches can be organized according to a few different properties. In general, there are loosely-coupled and tightly-coupled solutions. This refers to whether or not the IMU and camera data are processed independently of each other. In the interest of computational cost, we chose to focus on loosely coupled solutions, sacrificing accuracy in the process. Additionally, solutions are often split into optimization and filter based approaches. Optimization approaches aim to determine a state that minimizes a joint cost function, incorporating error functions for the IMU and image models[4]. Filter-based approaches use probability theory to formulate the problem in terms of mean and covariance. The most common filtering method is the EKF, which is widely used in robotics to fuse sensor data. This paper focuses on the filter-based method for its simplicity, computational efficiency, and versatility.

The EKF is robust and can be adapted in a number of ways to fuse sensor data. Alatise and Hancke use the accelerometer data in the process model and fuse both gyroscope and vision data into the measurement model[1]. Xiong et al. use a dual stage EKF where the gyroscope is first corrected by the accelerometer, and the resulting prediction is corrected by the vision data[5]. We will be simply using the accelerometer and gyroscope data to form a prediction, and stereo measurements as the corrector.

3 Experimental Setup

The proposed algorithm will be evaluated using a variety of KITTI benchmark datasets in different environments. The car is equipped with a number of sensors, each with its own defined reference frame.

The GPS/IMU sensor frame near the back left passenger side is used as the vehicle frame, since the GPS measurements will be used as groundtruth. The camera system consist of two sets of stereo cameras: colour and grayscale. This project will use the grayscale cameras corresponding to ”Cam 0” for the left camera and ”Cam 1” for the right camera. To convert feature locations in the image plane to the vehicle frame, a series of transformations provided in the KITTI development kit are used. The dimensions relating the different sensor frame origins along with the transformations indicated by the blue arrows are shown in figure 1[7].

4 Feature Extraction

To detect the 3D feature location of the landmarks in the camera frame, we used the SURF feature detector to find features in rectified stereo camera images. By comparing feature descriptors of features extracted from the left stereo camera image in current time step k and previous time step

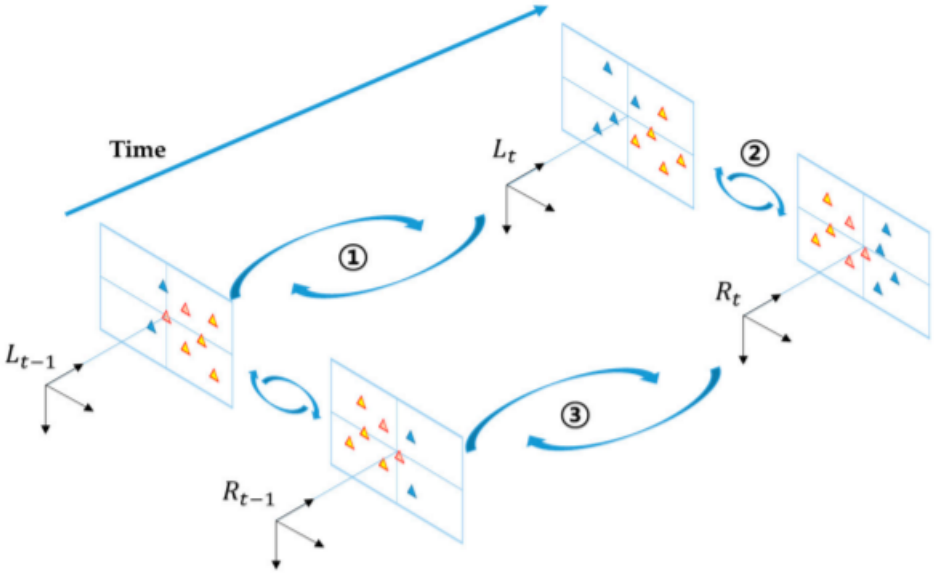

Figure 2: Feature matching mechanism between four images: stereo pair at current timestep and stereo pair at previous timestep. This methodology and graphic are from Yoon and Kim[6]

.

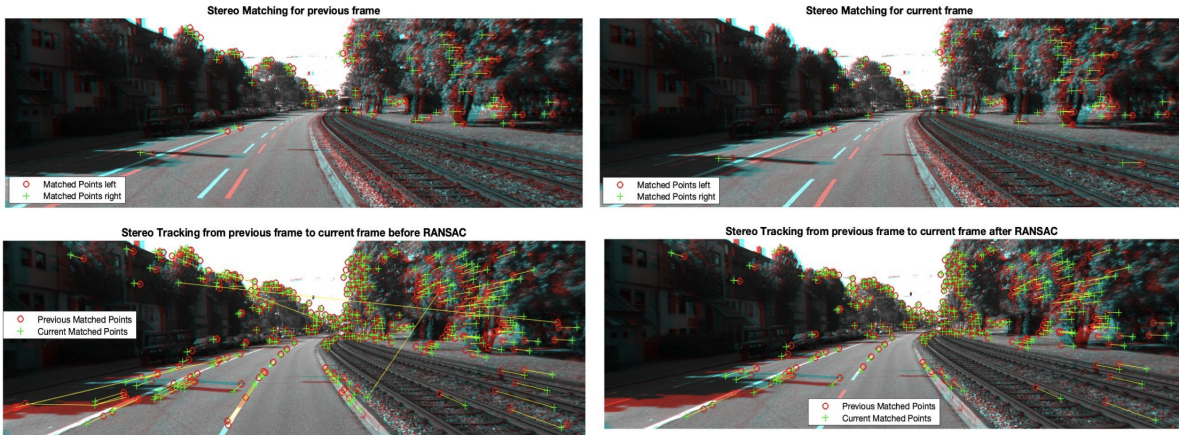

Figure 3: Feature matching results using SURF detectors. Top-Left) Feature detection in previous frame. Top-Right) Feature detection in current frame. Bottom-Left) Feature matching between previous and current frame. Bottom-Right) Feature matching between previous and current frame using RANSAC to eliminate mismatches.

$k-1$, we detect the corresponding features that move from the previous time step $k-1$ to current time step k. However, since the feature descriptors contain the intensity change around feature location, it may lead to a mismatch. Therefore, a RANSAC loop is necessary to filter out outliers after the matching of descriptors in the different time frames. Figure 2 demonstrates this feature matching between the stereo pairs at the current and previous timestamp[6].

Since we chose to use the sychronized + rectified images provided by KITTI, it is simpler to find the corresponding features in the right stereo image after we found the corresponding features in both left images. The fastest method is to select a small patch around the known feature location in the left camera image, and then slide the patch along the y-axis to find the matched patch with the smallest sum of absolute difference (SAD). However, this match is not accurate enough for features in complex environments such as trees because we need to vary the patch size for different time steps. A slower but better method is still using the SURF feature detector to find the features in the right stereo camera and then find the corresponding features between the left and right recti-

fied stereo images. Figure 3 shows the results of the feature detection algorithm in the current and previous timestamps, and demonstrates the impact of applying RANSAC to filter out mismatches. The results were obtained using the KITTI dataset "2011_09_26_drive_0001".

5 Extended Kalman Filter

The equations used follow the derivations and notation in Barfoot[8] chapter 8.2. We define the state as the set of inertial to vehicle frame transformation matrices $\boldsymbol{T}_{v_k i}$ from timestep 1 to K:

$$\{\boldsymbol{T}_{v_1 i}, ..., \boldsymbol{T}_{v_K i}\} = \{\boldsymbol{T}_1, ..., \boldsymbol{T}_K\} \tag{1}$$

where we have used short-hand notation by dropping the reference frames. In the experiment, we take the first vehicle frame as the inertial frame. Therefore, the states correspond to the transformation from the current pose to the initial pose.

5.1 Motion Model

The IMU indirectly provides translational velocities $\boldsymbol{v}_{v_k}^{iv_k}$ and rotational velocities $\boldsymbol{\omega}_{v_k}^{iv_k}$ which are considered as inputs to the motion model. These can be stacked to create a 6x1 input vector $\boldsymbol{\varpi}_k$ as follows

$$\varpi_k = \begin{bmatrix} v_{v_k}^{iv_k} \\ \omega_{v_k}^{iv_k} \end{bmatrix}, k = 1, ..., K \tag{2}$$

The pose and inputs are perturbed by process noise, resulting in the decomposition of nominal and perturbation kinematics for our motion model. Our discrete time SE(3) motion model is as follows:

$$\bar{\boldsymbol{T}}_k = exp(\Delta t_k \bar{\boldsymbol{\varpi}}_k^{\wedge}) \bar{\boldsymbol{T}}_{k-1} \tag{3}$$

$$\delta\boldsymbol{\xi}_k = exp(\Delta t_k \bar{\boldsymbol{\varpi}}_k^{\curlywedge}) \delta\boldsymbol{\xi}_{k-1} + \boldsymbol{w}_k \tag{4}$$

where we have $\Delta t_k = t(k) - t(k-1)$ as the sampling period between inputs, the process noise is modelled by $\boldsymbol{w}_k = \mathcal{N}(0, \boldsymbol{Q}_k)$, and we define $\boldsymbol{\Xi}_k = exp(\Delta t_k \bar{\boldsymbol{\varpi}}_k^{\wedge})$. This allows to perform the prediction step of the EKF by propagating the pose forward in time using the most recent IMU measurements. The predicted pose is calculated by:

$$\check{\boldsymbol{T}}_k = \boldsymbol{\Xi}_k \hat{\boldsymbol{T}}_{k-1} \tag{5}$$

and the predicted covariance is given by:

$$\check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1} \hat{\boldsymbol{P}}_{k-1} \boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_k \tag{6}$$

where we have defined $\boldsymbol{F}_{k-1} = Ad(\boldsymbol{\Xi}_k)$.

5.2 Observation Model

The stereo camera system produces 3x1 key feature location measurements in the vehicle frame. At each timestamp k, we can have up to M measurements, where the jth measurement is denoted as $\boldsymbol{y}_{jk}$. Our 3x1 measurement model projects a 4x1 stationary point in the inertial frame $\boldsymbol{p}_j$ expressed in homogeneous coordinates into the vehicle frame as follows:

$$\boldsymbol{y}_{jk} = \boldsymbol{D}^T \boldsymbol{T}_k \boldsymbol{p}_j + \boldsymbol{n}_{jk} \tag{7}$$

where $\boldsymbol{n}_{jk}$ is the measurement noise and $\boldsymbol{D}$ is a dilation matrix which is used to pick off the first three coordinates of the homogeneous representation:

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

We can invoke our perturbation scheme to arrive at a nominal solution and perturbed solution for the measurement error:

$$\boldsymbol{y}_{jk} = \boldsymbol{D}^T \bar{\boldsymbol{T}}_k \boldsymbol{p}_j \tag{9}$$

$$\delta\boldsymbol{y}_{jk} \approx \boldsymbol{D}^T (\bar{\boldsymbol{T}}_k \boldsymbol{p}_j)^{\odot} \delta\boldsymbol{\xi}_k + \boldsymbol{n}_{jk} \tag{10}$$

We construct the measurement Jabobian matrix for measurement j as follows:

$$\boldsymbol{G}_{jk} = \boldsymbol{D}^T (\bar{\boldsymbol{T}}_k \boldsymbol{p}_j)^{\odot} \tag{11}$$

At each timestamp k, we can stack these measurements and G matrices for up to M measurements:

$$\boldsymbol{y}_k = \begin{bmatrix} y_{1k} \\ . \\ . \\ . \\ y_{Mk} \end{bmatrix}, \boldsymbol{G}_k = \begin{bmatrix} G_{1k} \\ . \\ . \\ . \\ G_{Mk} \end{bmatrix} \tag{12}$$

and the measurement noise covariance matrices $\boldsymbol{R}_{jk}$:

$$\boldsymbol{R}_k = diag(\boldsymbol{R}_{1k}, ..., \boldsymbol{R}_{Mk}) \tag{13}$$

This allows us to calculate the Kalman gain as follows:

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_k \boldsymbol{G}_k^T (\boldsymbol{G}_k \check{\boldsymbol{P}}_k \boldsymbol{G}_k^T + \boldsymbol{R}_k)^{-1} \tag{14}$$

We can now apply the correction step to update our state covariance using the following equation:

$$\hat{\boldsymbol{P}}_{\boldsymbol{k}} = (\mathbf{1} - \boldsymbol{K}_k \boldsymbol{G}_k)\check{\boldsymbol{P}}_k \tag{15}$$

For the mean update, we define the mean correction as $\boldsymbol{\epsilon}_k$:

$$\boldsymbol{\epsilon}_k = \boldsymbol{K}_k(\boldsymbol{y}_k - \check{\boldsymbol{y}}_k) \tag{16}$$

where $\check{\boldsymbol{y}}_k$ is the stacked predicted measurements for up to M measurements per timestep.

$$\check{\boldsymbol{y}}_k = \begin{bmatrix} \check{\boldsymbol{y}}_{1k} \\ . \\ . \\ . \\ \check{\boldsymbol{y}}_{Mk} \end{bmatrix}, \check{\boldsymbol{y}}_{jk} = \boldsymbol{D}^T \check{\boldsymbol{T}}_k \boldsymbol{p}_j \tag{17}$$

We then apply the mean correction using the exponential map:

$$\hat{\boldsymbol{T}}_k = exp(\boldsymbol{\epsilon}_k^{\wedge})\check{\boldsymbol{T}}_k \tag{18}$$

This formulation requires known stationary points $\boldsymbol{p}_j$ in the inertial frame. As an approximation for known points, we use features matched between two timestamps and convert the measurement from the image plane to the vehicle frame using the previous timestamp's pose estimate $\hat{\boldsymbol{T}}_{k-1}$.

5.2.1 Stereo Camera Triangulation: 2D to 3D

In order to find the 3D location of the landmark with the corresponding feature locations in the left and right stereo images, the simplest method is through an inverse stereo camera model, where the stereo camera matrix is given by $\boldsymbol{M}$. Here we use the left-camera model for since it aligns well with the experimental setup.

$$\boldsymbol{M} = \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ f_u & 0 & c_u & -f_u b \\ 0 & f_v & c_v & 0 \end{bmatrix} \tag{19}$$

where $\boldsymbol{M}$ represents the projection matrix after rectification and all the calibration parameters are given by the KITTI development kit. Note that this matrix is not invertible, but the inverse can be obtained by manipulating the resultant equations. We denote the transformation from the camera frame to the image plane as $\boldsymbol{M}_{inv}$. To obtain our landmark positions in the inertial frame $p_j(x, y, z, 1)$, we project stereo features $x(u_l, v_l, u_r, v_r)$ in the image plane from the previous timestep:

$$\boldsymbol{p_j} = \hat{\boldsymbol{T}}_{k-1}\boldsymbol{T}_{i,v}\boldsymbol{T}_{v,c}\boldsymbol{M}_{inv}\boldsymbol{x} \tag{20}$$

where $T_{v,c}$ is the camera to velodyne frame transform and $T_{i,v}$ is the velodyne to IMU (vehicle) frame transform, both of which are provided by KITTI and shown in figure 1.

If the rectification is prefect, the rays of feature from the left and right camera should intersect at $\boldsymbol{p}_j$ in the 3D camera frame, however, the rectification is not perfect for every pixel in each time step, the noise, camera model uncertainty, and slights distortion might cause some shift on 3D location of the same feature for stereo camera, and the rays from left and right camera didnt intersect at same height. Therefore, the simple method is not accurate enough to predict the actual 3D location of the features, we will use the method of two light ray indication from Cheng[9]. The 3D positions of selected features can be determined by intersecting rays of feature projected from the stereo camera center toward the 3D landmark $\boldsymbol{p}_j$. Another benefit of this method is the co-variance matrix associated with each 3D landmark location after the calculation. The following derivation follows Cheng. The endpoints coordinates of the line segment of left and right camera used to find the 3D landmark position are:

$$\mathbf{P}_L = \mathbf{c}_L + \hat{\mathbf{r}}_L m_L \tag{21}$$

$$\mathbf{P}_R = \mathbf{c}_R + \hat{\mathbf{r}}_R m_R \tag{22}$$

where $\mathbf{c}_L$ and $\mathbf{c}_R$ are the 3D location of the left and right camera optical centres, $\hat{\mathbf{r}}_L$ and $\hat{\mathbf{r}}_R$ are the unit vector pointing along each ray which can obtained by the 3D feature location where:

$$\mathbf{r}_L = \mathbf{C}_{i,C_L}\mathbf{K}_L^{-1}\begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix}, \hat{\mathbf{r}}_L = \frac{\mathbf{r}_L}{|\mathbf{r}_L|} \tag{23}$$

m_L and m_R are scalar lengths that are defined as

$$m_L = \frac{(\mathbf{b}^T\hat{\mathbf{r}}_L) - (\mathbf{b}^T\hat{\mathbf{r}}_R)(\hat{\mathbf{r}}_L^T\hat{\mathbf{r}}_R)}{1 - (\hat{\mathbf{r}}_L^T\hat{\mathbf{r}}_R)^2} \tag{24}$$

$$m_R = (\hat{\mathbf{r}}_L^T\hat{\mathbf{r}}_R)m_L - (\mathbf{b}^T\hat{\mathbf{r}}_R) \tag{25}$$

where $\mathbf{b}$ is the baseline between left and right camera. After taking the partial derivative of m_L and m_R, the Jacobian matrix $P^{'} = (\hat{\mathbf{r}}_L^{'}m_L + \hat{\mathbf{r}}_L m_L^{'} +$

$\hat{\mathbf{r}}_R^{'} m_R + \hat{\mathbf{r}}_R m_R^{'})/2$ therefore, the covariance of the 3D landmark P_j is

$$\sum_P = P^i \begin{bmatrix} \sum_l & 0 \\ 0 & \sum_r \end{bmatrix} P'^T \tag{26}$$

where the $\sum_l$ and $\sum_r$ are the covariance of the corresponding features in the left and right image.

6 Results

The EKF was evaluated on a few KITTI datasets and compared to the GPS groundtruth. We present the results for the following KITTI dataset: "2011_09_26_drive_0036".

KITTI Dataset	Environment	Elapsed Time
2011_09_26_drive_0036	Residential	1min:20sec

Table 1: KITTI dataset details.

The results displayed in figure 4 show an overlay of the estimated trajectories and the groundtruth from the GPS. The dead reckoning results are also included to demonstrate the improvements after fusing stereo camera measurements. The benefit of fusing stereo measurements

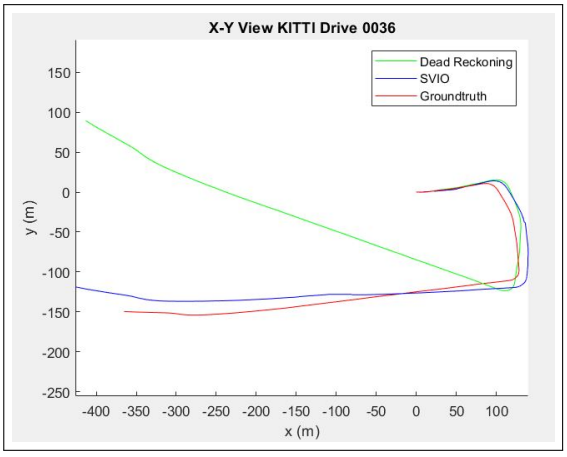

Figure 4: X-Y overhead view of results comparing the estimated path for dead reckoning and fused visual-inertial vs the ground truth from the GPS.

to the inertial odometry is exemplified in the clear improvement of the results. The IMU integration causes the estimator to deviate further from ground truth as time goes on. During the turns, the angular velocity about the z-axis increases drastically, causing the IMU to overestimate the actual rotation of the vehicle. One area of clear improvement is demonstrated in the stereo camera's ability to correct these overestimates. However, we notice that in general the estimated paths are both considerably longer than the groundtruth. Stereo cameras are relatively more accurate at determining feature locations in the horizontal and vertical direction, while features near the centre of an image have increased uncertainty in their depth. This may explain why the sensor fusion results in an error of 100m over the entire length of the drive. Over large distances, this impact is magnified.

The error plots with 3σ uncertainty envelope for the position and rotations of the vehicle are displayed in figure 5 and figure 6. The error plots demonstrate the overall innacuracy when estimating the pose over a long distance. This may indicate that SVIO is better suited for shorter distance and lower velocity applications. However, as a supplement for environments with poor GPS signal, it may be very useful over short distances. For the rotational errors, we see that they remain relatively low, except for the two spikes during the turns.

We also note that there is significant z error induced by the addition of the stereo measurements. The dead reckoning results have minimal z-direction error as the z acceleration and pitching angular velocity measurements from the IMU are relatively low. The increased error may be attributed to a poor distribution of features along the z-axis due to the complexities within the environment. This is discussed further in future work.

7 Conclusions and Future Work

An EKF based stereo visual-inertial odometry algorithm has been developed and implemented on the KITTI experimental setup. The performance of the algorithm was evaluated using KITTI benchmark datasets from a residential environment. Overall, the project has provided an excellent learning opportunity to perform stereo visual-inertial state estimation of an autonomous vehicle, utilize matrix Lie groups and algebra, implement one of the most widely used algorithms in robotics, and understand some of the complications that arise in 3D state estimation.

There are several components of the project that we would have liked to improve upon, but were unable due to time constraints. The main limitation to the accuracy of the results is the formulation of the EKF, in particular the projection of the stationary landmark positions using the previous mean estimate. This attempt to

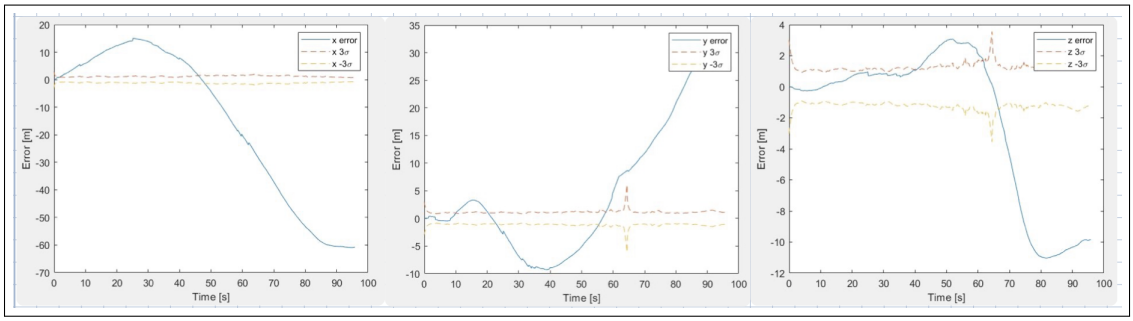

Figure 5: Position error plot for SVIO vs groundtruth. Left) Error in x direction. Middle) Error in y direction. Right) Error in z direction.

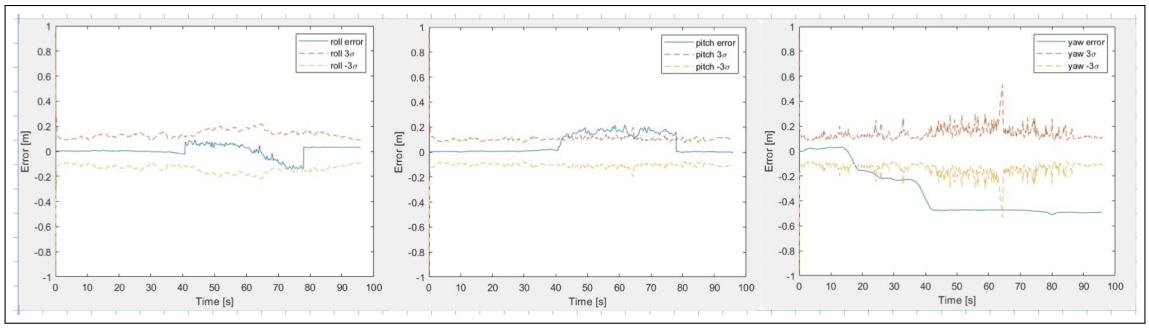

Figure 6: Rotational error plot for SVIO vs groundtruth. Left) Error in roll. Middle) Error in pitch. Right) Error in yaw.

project global landmarks has the undesirable effect of compounding the error, since the uncertain state estimate is used within the corrector step to produce the landmarks. The solution must be reformulated so that the measurement model estimates the relative pose transformation between two timestamps as opposed to the global transformation between the current timestamp and the inertial frame. This could potentially be achieved using a pointcloud-alignment technique such as formulated in Barfoot[8] chapter 8.1.

The error in the vertical direction may be directly related to a lack of uniformity in the feature detection across the entire image as suggested by Kitt et al[10]. In many environments, especially vehicles driving on the road, there are often more complexities in the upper portion of the image which causes the feature extraction to be biased. A technique called "bucketing" partitions the image into a grid and ensures that an equal number of features are selected from each rectangle. This subset of features is well distributed on the z-axis, corresponding to the roll axis, which may improve upon the inaccuracies currently present in the results. In this paper, we have not considered the dynamic environment associated with autonomous cars, including pedestrians and other moving vehicles. This may directly impact the feature matches locations, and the results of the visual odometry correction step. For instance, a moving car may provide incorrect inference to the relative pose transformation between two points within a pointcloud. Yoon and Kim [6] use photogrammetric optimization to handle this scenario.

References

[1] M. B. Alatise and G. P. Hancke, *Sensors*, vol. 17, no. 2164, 2017.

[2] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *CoRR*, vol. abs/1610.06475, 2016. [Online]. Available: http://arxiv.org/abs/1610.06475

[3] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," in *arXiv:1607.02565*, July 2016.

[4] J. S. V. Usenko, J. Engel and D. Cremers, "Direct visual-inertial odometry with stereo cameras," *International Conference on Robotics and Automation (ICRA)*, 2016.

[5] Z. L. X. Xiong, W. Chen and Q. Shen, "Ds-vio: Robust and efficient stereo visual inertial odometry based on dual stage ekf," *arXiv eprint*, vol. arXiv:1905.00684, 2019. [Online]. Available: http://arxiv.org/abs/1905.00684

[6] S.-J. Yoon and T. Kim, "Development of stereo visual odometry based on photogrammetric feature optimization," *Remote Sensing*, vol. 11, no. 67, 2019.

[7] C. S. A. Geiger, P. Lenz and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, 2013.

[8] T. Barfoot, *State Estimation for Robotics*. Cambridge, UK: Cambridge University Press, 2017.

[9] L. M. Y. Cheng, M. Maimone, "Visual odometry on the mars exploration rovers," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, 2005.

[10] A. G. B. Kitt and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," *2010 IEEE Intelligent Vehicles Symposium*, 2018.