VI-SLAM for Subterranean Environments

Andrew Kramer, Mike Kasper, Christoffer Heckman

Abstract Among the most challenging of environments in which an autonomous mobile robot might be required to serve is the subterranean environment. The complete lack of ambient light, unavailability of GPS, and geometric ambiguity make subterranean simultaneous localization and mapping (SLAM) exceptionally difficult. While there are many possible solutions to this problem, a visual-inertial framework has the potential to be fielded on a variety of robotic platforms which can operate in the spatially constrained and hazardous environments presented by the subterranean domain. In this work we present an evaluation of visual-inertial SLAM in the subterranean environment with onboard lighting and show that it can consistently perform quite well, with less than 4% translational drift. However, this performance is dependent on including some modifications that depart from the typical formulation of VI-SLAM, as well as careful tuning of the system's visual tracking parameters. We discuss the sometimes counter-intuitive effects of these parameters and provide insight into how they affect the system's overall performance.

1 Introduction

Mobile field robots are being deployed in ever more challenging environments. They are expected to operate in conditions where the sensors normally used for state estimation are severely limited at best and completely useless in extreme cases. As a motivating example, in the subterranean environment [1] vision is limited to what can be seen with on-board lighting and GPS is not a viable option to constrain drift.

Andrew Kramer
University of Colorado, Boulder e-mail: andrew.kramer@colorado.edu

Mike Kasper
University of Colorado, Boulder e-mail: mike.kasper@colorado.edu

Christoffer Heckman
University of Colorado, Boulider e-mail: christoffer.heckman@colorado.edu

Nevertheless, autonomous subterranean robots must be able to localize and map in these challenging conditions. Without these critical skills an autonomous robot in the subterranean environment will be unable to plan or accurately control its movements.

Vision-only SLAM is exceedingly popular [8] but this approach works well only when sensing conditions are favorable. i.e. if the environment is well-lit and trackable features are plentiful and evenly distributed. If visual tracking fails however, there may be no way to recover. This is especially true if visual tracking fails in an area that isn't previously explored, preventing the use of place recognition as a means to reinitialize. Generally these methods show a great deal of promise, but are improved through the addition of other sensors in order to operate through loss of visual tracking.

An attractive sensor combination for SLAM that balances weight, size and cost consists of visible light cameras and inertial measurement units (IMUs). The IMU in these frameworks provides a way to localize when visual tracking fails for short periods. However, current methods for visual-inertial SLAM depend heavily on several assumptions that do not hold in the subterranean context. The first and most damning of these is that the environment is well and evenly lit. The second is that a large number of trackable features are evenly distributed throughout the environment. Finally, despite the community's goal of handling aggressive motions and their effect on data from these sensors, current visual-inertial SLAM methods perform best when the sensors' motion is smooth and slow, minimizing motion blur and enabling at least several informative features to be tracked [3]. For subterranean robots it is likely that none of these conditions will be met. The only light source in the environment may be one carried by the robot. This direct lighting results in poor illumination of the environment and requires slower shutter speeds which in turn cause more motion blur. Also, depending on the environment's structure and the robot's path, the features used for tracking may not be evenly distributed in the environment. Lastly, it cannot be guaranteed that the movement of a subterranean robot will be smooth and slow. For example, a wheeled robot moving slowly over uneven, rocky ground will still experience sudden, jerky sensor motions as its wheels traverse discontinuities in the terrain.

SLAM using direct-depth sensors is also popular. 3D scanning LIDAR is a particularly attractive sensor as it does not require ambient light. It does come with its own limitations in the subterranean environment however. For example, most LIDAR SLAM methods rely on scan matching such as iterative closest point (ICP) [2]. These methods can fail to converge to the correct relative transform between reference and target point clouds if the robot's environment is geometrically ambiguous. For instance, in a long, uniform tunnel ICP will not have a unique solution for translation along the tunnel.

In this work we present an evaluation of a factor graph based sparse, indirect visual-inertial SLAM system on a novel subterranean dataset. The same dataset is run using a variety of visual measurement techniques and parameters. The effects of these variations on the SLAM system's performance are discussed with an eye to generally improving the performance of visual-inertial SLAM in the subterranean

environment. We develop a novel depth-enabled framework for evaluating the triangulated position of landmarks in the visual SLAM front-end and demonstrate its effectiveness in the subterranean setting. We also show the choice of parameters can have a profound effect on the system's overall performance in terms of accumulated translational drift over the robot's trajectory. Finally, we note some counterintuitive and interesting results in optimizing our system for the subterranean environment.

2 Related Work

2.1 Visual-Inertial SLAM Techniques

Visual-inertial SLAM techniques fall into two categories: filtering and batch optimization. In the filtering approach, IMU measurements are used to propagate the robot's state while image measurements are used to correct the state. The filtering approach is exemplified by MSCKF [7]. Filtering approaches to visual-inertial SLAM are generally faster and more efficient, but their accuracy is dependent on proper filter tuning. Additionally, filtering approaches only consider the previously estimated state and current sensor measurements in estimating the current state. They do not allow for the correction of previous states given new information.

In the batch optimization approach, the system's current and previous states are estimated using camera and IMU measurements as constraints in a nonlinear optimization problem. This approach is more resource intensive but also more accurate than filtering as it uses measurements at multiple timesteps to jointly estimate the system's current and previous states. Of course, a batch optimization method that optimizes over the complete set of measurements from the beginning of the robot's run would quickly become computationally intractable. So there are several methods for limiting the size of the problem while maintaining an accurate state estimate. For example, keyframing assumes that most camera images do not carry significant additional information; the camera does not move significantly between frames. So keyframing approaches select a subset of the most informative camera measurements, referred to as keyframes, and only estimate the system's states from those. However, when using this technique the size of the problem still grows linearly in the number of keyframes. A sliding window filter [13] is a popular way to limit this growth. The sliding window filter optimizes over a fixed-size set of the most recent keyframes, marginalizing out measurements and states from older keyframes. This means the complexity of the SLAM problem stays roughly constant over time, allowing it to be used for long-term localization and mapping. Both of these techniques are used by the popular optimization-based visual-inertial SLAM systems OKVIS [6] and VINS-Fusion [11].

2.2 Landmark Depth Estimation in Sparse Visual SLAM

Several methods have been used to incorporate depth measurements from stereo or RGB-D cameras into the sparse visual SLAM problem. The most straightforward way to do this is to add the camera measurements to the optimization as a 3D constraint on the position of the keypoint in space. However, this can cause problems because the depth and projection measurements generally have different units and are subject to different kinds of errors [12]. There are a few ways to address this.

In [12] camera and depth measurements are treated separately: as a 2D measurement of a point in space projected onto the camera's sensor and a 1D measurement of that point's depth in the camera's frame. This allows for errors on these measurements to be modeled differently. It also makes depth measurements optional. A camera measurement can still be added to the optimization if its associated depth measurement can't be calculated or its uncertainty is too high.

ORB-SLAM2 takes an interesting approach to incorporate depth measurements. It extracts ORB features from the left and right stereo images. It then parameterizes camera measurements as $[u_L, v_L, u_R]$ where u_L and v_L are the horizontal and vertical coordinates of the ORB keypoint on the left camera's sensor and u_R is the horizontal coordinate of the corresponding ORB keypoint on the right camera's sensor. The depth of the keypoint follows the following linear relationship:

$$d = \frac{f_x b}{u_L - u_R} \tag{1}$$

where f_x is the camera's horizontal focal length and b is the distance between the two cameras. Because of the point's depth is a linear function of the difference in the two horizontal coordinates of the point on the camera's sensor, the horizontal coordinate of the keypoint in the right camera can be used in place of the point's depth in the optimization. This means that depth measurements in this method have the same units as projection measurements and are subject to the same kinds of errors. This method assumes, however, that the input images are stereo rectified and have the same focal length. In Section 3.1 we introduce a method that overcomes some of these limitations and explain how it is augmented into our overall framework.

3 VI SLAM Method

Our SLAM approach is split into a frontend for visual detection and matching and a backend for nonlinear optimization. The frontend detects visual features and matches them to previously seen landmarks using the BRISK feature detector and descriptor. BRISK provides high robustness to motion blur, lighting changes, and changes in viewpoint. This makes it an ideal detector and descriptor for SLAM in subterranean environments.

Our approach's backend uses batch optimization over a sliding window [13] of recent camera frames and IMU measurements to estimate the system's state at the time of each camera frame. A camera frame consists of the 2-D measurements of landmarks projected into the the cameras' sensors and optionally the 1-D depth of the same landmarks. An IMU measurement is the set of raw IMU readings that occur between successive camera frames. These IMU readings are preintegrated as in [4]. The sliding window consists of the most recent P camera frames linked by IMU measurements (these are referred to as IMU frames) and the N most recent keyframes. The number of IMU frames and keyframes, P and N can be tuned to suit the robot's environment and the capabilities of its computing hardware.

The IMU frames are used to estimate robot poses, landmark positions, robot speeds and IMU biases. After a frame passes out of the IMU window the system decides if that frame should be a keyframe. If so, the IMU measurements are marginalized but the camera measurements continue to be used to estimate the robot's poses and the landmark positions. If not, then the camera *and* IMU measurements associated with that frame are marginalized. This marginalization strategy is pictured below in Figure 1.

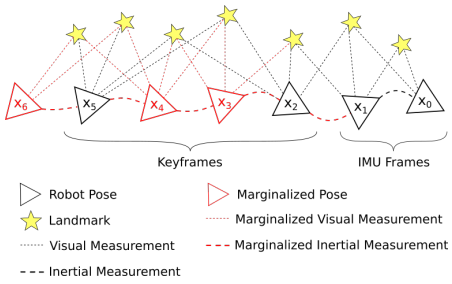


Fig. 1 Illustration of the marginalization strategy used by our SLAM method. In this case there are 2 IMU frames and 2 keyframes.

The graph structure of the problem is shown in Figure 2. This problem structure is similar to that used by OKVIS [6].

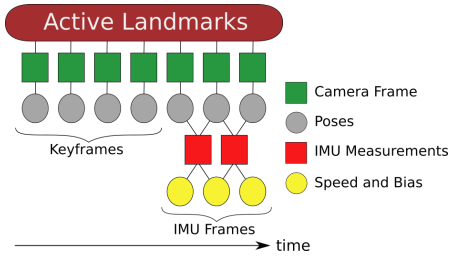


Fig. 2 Diagram of the factor graph used by our VI-SLAM system.

The cost function used by our method's backend is defined in Equation 2

$$J(x) \coloneqq \underbrace{\sum_{i=1}^{N+P} \sum_{j \in \mathcal{J}(i)} \left[e_r^{i,j^T} W_r^{i,j} e_r^{i,j} + \left(w_d^{i,j} e_d^{i,j} \right)^2 \right]}_{\text{visual term}} + \underbrace{\sum_{i=1}^{P} e_s^{i^T} W_s^i e_s^i}_{\text{inertial term}} \tag{2}$$

where N is the number of keyframes and P is the number of IMU frames in the optimization, $\mathcal{J}(i)$ is the set of landmarks observed in frame i, $e_r^{i,j}$ is the reprojection error for landmark j in frame i, $e_d^{i,j}$ is the scalar depth error for landmark j in frame i, e_s^i is the IMU error for frame i, and the W matrices contain the weights for the errors. Note that inertial terms are only present for the IMU frames, and depth terms (e_d and w_d) are optional for all frames.

Our SLAM system uses three steps to ensure robustness to incorrect data association. First, the system's frontend uses IMU measurements to propagate the camera's pose from the last optimized pose to the current frame time and predicts where matched landmarks should be observed in the new pose. Landmark matches that are greater than a certain distance from this predicted observation are discarded. Second, the frontend uses RANSAC to ensure the observed positions of the matched landmarks in the camera frame are geometrically consistent with the landmarks' estimated positions in the global frame. Lastly in the system's backend the optimizer computes the cost using robust norms.

3.1 Direct Depth Measurement

In subterranean environments with poor lighting there are often very few trackable features available. To make matters worse the features are often poorly distributed. For example, if the robot is carrying its own light, nearby objects will be well illuminated but distant objects will often be very poorly illuminated. So the few landmarks that can be tracked by the SLAM system will be very close to the camera. This can lead to ambiguities in monocular visual tracking because nearby points are less informative for estimating the robot's orientation. The inverse is also true, when the tracked landmarks are all distant from the camera they are less informative for estimating the robot's translation. Finally, scale is only introduced through the optimization of IMU factors, which makes it critically dependent on the accuracy of the IMU.

One solution to these problems is to use multi-camera systems. If the cameras have significant overlap in their fields of view they can provide multiple measurements of the same landmark at the same timestep but from different viewpoints. This can help to constrain the estimated distance to that landmark. However, this method treats measurements from multiple cameras the same as monocular measurements: the measurements are simply the 2D coordinates of a landmark when projected onto

each camera's sensor. This strategy does not take advantage of the fact that a direct depth measurement can be obtained for any landmark that is visible in both cameras.

Our method does take advantage of this fact. It first uses brute-force matching to match keypoints between the left and right stereo images. It then uses stereo triangulation to calculate the distance to the point in space corresponding to the keypoint. Two measurements are then added to the optimization problem: a 2D camera measurement of the point in camera zero's frame; and a 1D measurement of the point's distance from the left camera's sensor as in [12].

We keep the depth and 2D camera measurements separate for two reasons. First, it allows us to weight errors and handle outliers for each measurement type differently. Second, treating projection and depth as separate measurements means that depth measurements can be used but they are not required. So in our system keypoints for which depth cannot be calculated are still added to the system as 2D measurements without depth. This allows our system to use depth measurements when they're available but tracking does not fail when depth measurements aren't available. Instead it degrades gracefully back to using only 2D measurements.

4 Experiments

4.1 Sensor Setup

To obtain our dataset we used the infrared stereo cameras on an Intel Realsense D435 and a Lord Microstrain 3DM-GX5-15 IMU as sensors. Images were captured at 640×480 resolution and IMU messages were read at 100Hz. The performance of a visual-inertial SLAM system is highly dependent on knowing the camera-to-IMU transform (referred to as the extrinsics) with great accuracy. To obtain the extrinsics as well as the intrinsics (focal length and distortion parameters) for both cameras we used Vicalib, a calibration library based on ceres solver. Vicalib first finds intrinsic camera parameters (focal length, central point, and distortion parameters) by tracking conics on a calibration target and solving an overdetermined perspective-n-point ransac problem through gradient descent. Vicalib then determines extrinsic parameters (camera-to-IMU rotation and translation) between each IR camera and the IMU using gradient descent with an IMU residual as presented in [9]. In this optimization intrinsic parameters are fixed and only the extrinsic parameters are varied. Because the data was taken in complete darkness in a subterranean environment, the sensor rig was also equipped a forward-facing 9000 lumen soft-white LED headlamp.

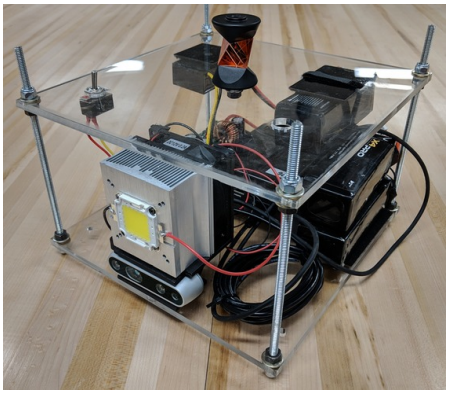


Fig. 3 The sensor rig used to capture our subterranean dataset. Includes an Intel Realsense D435 stereo camera, a Lord Microstrain 3DM-GX5-15 IMU, and a 9000 lumen LED headlamp.

4.2 Dataset Description

Our dataset was recorded in the Hidee gold mine in Central City, Colorado. It consists of camera images and IMU messages recorded by our handheld sensor rig as it was carried from the start to the end of the mine and back. The total distance covered was roughly 340 m at an average speed of 1.4 m/s. The environment consisted of tunnels roughly 2 m in height and 1.5 m wide with bare rock walls. An example image of the environment can be seen in Figure 4.

Fig. 4 Typical camera image from the mine dataset with the headlamp at 15% intensity. Note the scene is well lit near the camera, and almost completely dark just a few meters from the camera. This is typical for the entire dataset.

There was no ambient light in the tunnels. The only available light in the environment came from the sensor rig's onboard LED headlamp. Two different runs were done over the same course, one with the headlamp set to 50% intensity and one with the headlamp at 15% intensity. Apart from the headlamp setting the two runs are nearly identical. The limited, direct lighting from the headlamp had two major effects. First, the cameras needed to use long exposure times to adequately expose their images. This means the images from the two IR cameras often have severe motion blur as pictured in Figure 5.

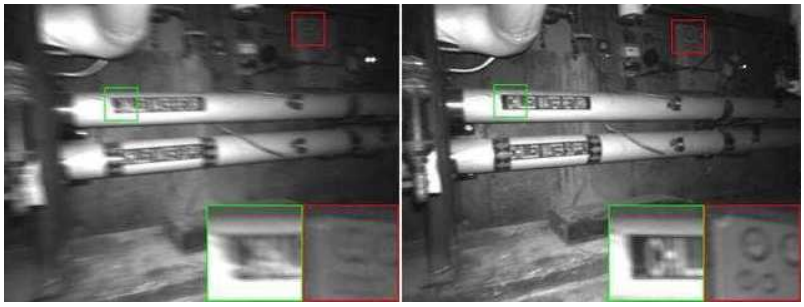


Fig. 5: Camera image comparing motion blur with headlamp at 15% intensity (left image) to 50% intensity (right image). Average image intensity for both images is similar but the left image has significantly more motion blur.

Second, light intensity in the scene generally decreases with the square of distance from the headlamp. This means parts of the scene that are very close to the camera are well exposed, but parts that are even a moderate distance from the camera are very underexposed. This gives the camera an artificially narrow depth of field. The result is that the SLAM system was unable to track keypoints more than a few meters from the cameras. This is illustrated in Figure 6

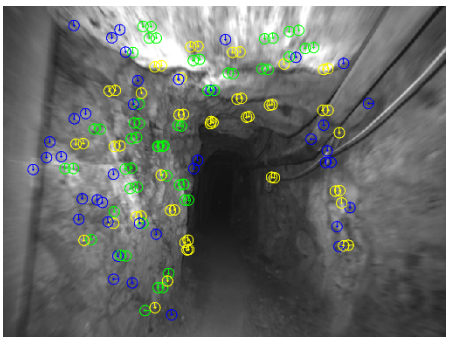


Fig. 6 Example stereo image demonstrating the limited range of the sensor rig's headlamp and the consequently shallow depth range in which keypoints are reliably trackable.

4.3 Test Description

Each dataset was run in our visual-inertial SLAM system with a different set of measurement parameters, referred to as a setting. Specifically, the keypoint detection threshold, keypoint matching threshold, and stereo measurement techniques were varied. The detection threshold is the minimum strength a keypoint must have before it is added as a measurement to the SLAM frontend [5]. The matching thresh-

old is the maximum hamming distance between two keypoint descriptors that can be considered a match. Lastly, stereo measurements can be added to the SLAM system in one of two ways. The conventional method is to add measurements from multiple cameras as multiple, independent constraints on 2D reprojection error as is done in OKVIS [6]. The alternative is to use the method described in Section 3.1, which matches features between the two cameras and explicitly calculates depth measurements through stereo triangulation. The parameters used in each setting we tested are given in Table 1.

Table 1: Parameters used for each setting on which the SLAM system was tested.

Setting Number	1	2	3	4	5	6	7	8	9	10	11	12
Detection Threshold	50	50	50	50	40	40	40	40	40	40	40	40
Matching Threshold	80	80	80	80	60	60	60	60	80	80	80	80
Headlamp Setting	low	high	high	low	low	high	low	high	low	high	low	high
Depth Measurement	no	no	yes	yes	no	no	yes	yes	no	no	yes	yes

4.4 Error Evaluation

Due to the length of the path, the confined space of the mine tunnels, and the lack of line-of-sight between all points in the path we were not able to obtain accurate groundtruth over the whole path with motion capture, laser tracker, or similar system. Instead, SLAM performance is evaluated only on the accumulated translational drift over the course of the dataset. To assist in calculating this metric, we started and ended each dataset in the same place and placed an AprilTag [10, 14] at that location. This allows us to calculate the groundtruth transform from the start of the dataset to the end, $T_{\text{groundtruth}}$. We then compare this to the same quantity estimated by the SLAM system, T_{est} as follows:

$$T_{\text{err}} = T_{\text{est}} T_{\text{groundtruth}}^{-1} \tag{3}$$

The error e is defined as the magnitude of the translational component of T_{err}: $e = \|t_{\text{err}}\|_2$.

5 Results

Table 2 shows the averaged results over 5 runs for each setting. Each run was 340 m at an average speed of 1.4 m/s. 2D keypoint counts are not reported for settings in which depth measurements are used because the keypoint detection and matching

parameters are same as in other settings where depth measurements were not used. So the 2D keypoint counts for settings with depth would be redundant with the corresponding settings without depth measurements. Also note the 2D and depth point counts refer to the number mean number of points detected per camera frame.

Table 2: Averaged results for all settings.

Setting Number	1	2	3	4	5	6	7	8	9	10	11	12
Translational Error (m)	18.8	12.7	36.3	16.0	12.0	29.2	9.72	23.9	12.3	29.6	11.8	34.9
2D Points Detected	88.5	128			122	180			119	181		
2D Points Matched	43.3	58.5			53.6	70.8			56.6	84.0		
Depth Measurements			41.2	28.7			39.5	59.2			40.0	59.1
Depth Matches			20.6	14.9			18.4	24.7			21.1	28.0

Figure 7 shows the whole range of translational errors obtained for each setting. Settings that use the brighter headlamp are plotted in blue while low-light settings are plotted in red. It is interesting to note that low-light settings tend to have lower translational error and are more consistent than the high-light settings. Figure 8 shows the same data grouped into settings with high and low light and with and without depth measurement. Curiously, depth measurements are not helpful when used in high-light settings. However, in low-light settings depth measurements do improve both the best-case translational drift and the variance in drift. Also, the low-light settings in general are better and more consistent than high-light settings.

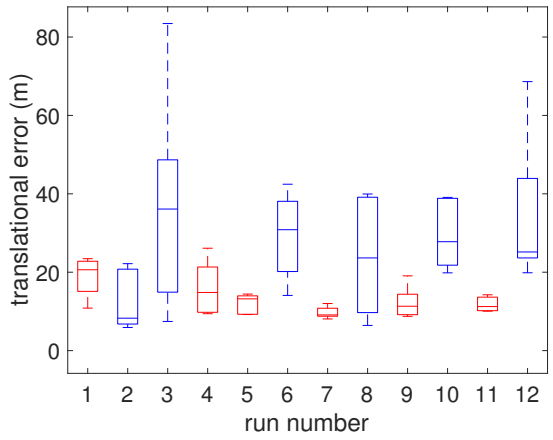


Fig. 7: Accumulated translational error for all settings. Red plots represent settings with low light. Blue plots represent settings with high light.

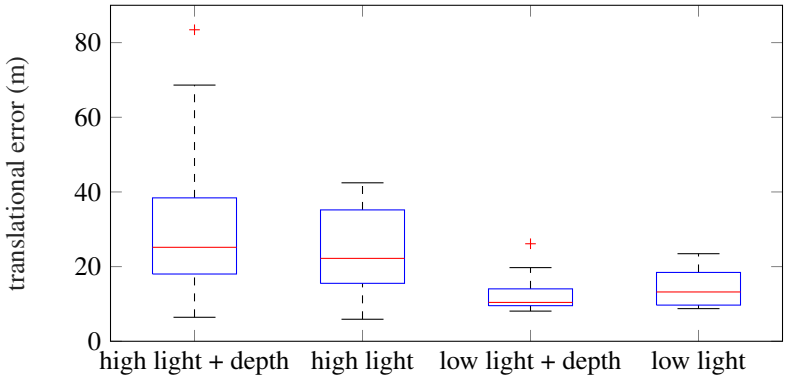
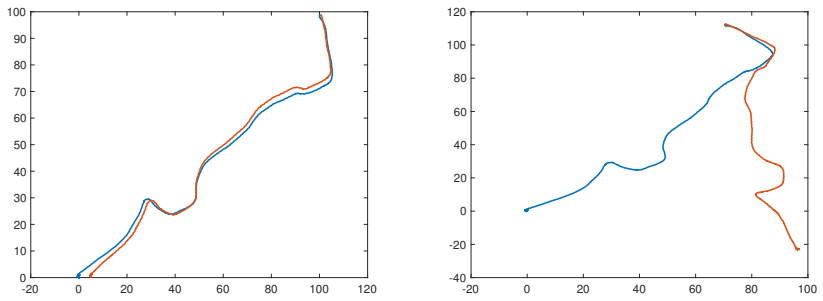


Fig. 8: Translational errors for high light and low light settings. Shows settings for which direct depth measurements were used separately from those for which depth wasn't used.

As expected, higher keypoint detection thresholds result in fewer total detected keypoints and depth measurements per frame and higher matching thresholds result in more measurements matched to existing landmarks per frame. Also more keypoints and depth measurements are able to be detected and matched when the headlamp setting is high. This increase in the number of points detected and matched does not translate to better performance in terms of translational drift, however. The settings that consistently had the best performance, settings 7 and 5, were done in low light with low detection and matching thresholds.
Lastly, Figure 9 shows plots of the estimated positions of the sensor rig throughout two example runs. Figure 9a shows the best results obtained and Figure 9b shows the worst results obtained. Only the estimated positions in the x and y dimensions are shown because the groundtruth translation and drift in the z direction for all runs is very small.

6 Conclusions

From these results it is clear that it is possible to obtain consistently good performance from a visual-inertial SLAM system in a confined subterranean environment with no ambient light. The best run on the best setting (setting number 7; i.e. low detection threshold, low matching threshold, low headlamp setting using direct depth measurements) had a translational drift of 5.90 meters, just 1.7% of the total path distance. On the worst-performing run for that setting the drift was still only 3.5% of the total path distance. However, obtaining such performance requires careful tuning of the frontend and the effects of each parameter can be counterintuitive. We will now describe our findings through our exploration.



(a) best run (b) worst run

Fig. 9: The estimated positions for two example runs plotted in the $x-y$ plane. The outgoing paths are plotted in blue and the return paths are plotted in orange. Note the outgoing and return paths in Figure 9a overlap almost perfectly, while there is nearly no overlap in 9b

Quite counterintuitively the use of a brighter headlamp is not always helpful. The reasons for this are not completely clear, but a subjective analysis of the image streams on the high and low light settings gives some clues. With brighter light, motion blur is minimized and more features are detectable. However brighter light also results in a stronger brightness dropoff from nearer to more distant objects in the scene as a result of autoexposure. While this could be mitigated through the careful tuning of autoexposure control schemes or focusing of the light source with reflectors or lenses, these strategies would likely be highly geometry and appearance dependent. In short, brighter light settings can actually result in a narrower depth of field in which features are detectable. Dimmer light settings, meanwhile result in a larger depth of field over which features can be detected and tracked. So even though a dimmer headlamp results in fewer features and more blurring, as long as the blur is not overwhelming all salient components of the image, those features are better distributed throughout the scene and are therefore more informative for the visual tracking front-end of SLAM. Furthermore, analysis of the raw depth measurements taken over two runs with bright and dim lighting supports the idea that a dimmer light setting can result in better distributed keypoints. The mean depth of detected keypoints was 2.12m for dim lighting and 2.01m for bright lighting. Additionally, the depth variance of detected keypoints was 0.68 for bright lighting and 0.72 for dim lighting. So in dim light the detected keypoints are slightly deeper and more widely distributed.

The use of direct depth measurements can be helpful but this is not always the case. In the brightly-lit settings the use of depth measurements did not affect the system's performance significantly, and the best and most consistent results with bright light were obtained without using depth measurements. In low-light settings, however the use of depth measurements usually resulted in better and more consistent per-

formance. This is likely related to the brighter light's effect on feature distribution in the camera's field of view. In brighter light settings the detected features all close to the camera. In this case a small amount of translation of the camera results in a large difference in the feature's projected position on the camera's sensor and it is easy to estimate the feature's position in 3D space from 2D measurements. So the depth measurement does not add much additional information. With dimmer light on the other hand, the features may be further from the camera and, due to higher motion blur and changes in illumination, the features may not be tracked for long enough to get a good estimate of their depth from 2D measurements alone. In this case direct depth measurements can add significant additional information.
To conclude, this work illuminates some of the challenges inherent in using visual-inertial SLAM in the subterranean environment with no ambient light. These challenges are often the result of direct lighting from an onboard light source. The use of an onboard light source causes a high brightness gradient throughout the scene, forcing the operator to make tradeoffs between the number and quality of features that can be detected. This work shows it is possible to obtain good visual-inertial SLAM performance in the subterranean environment with careful tuning. The need for this environment-specific tuning could be lessened if the effects of direct lighting from an onboard light source were taken into account in the system's measurement model, however, and future work should be directed toward this goal.

References

[1] Darpa subterranean (subt) challenge. URL `https://www.darpa.mil/program/darpa-subterranean-challenge`.

[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. URL `http://dblp.uni-trier.de/db/journals/pami/pami14.html#BeslM92`.

[3] Luca Carlone and Sertac Karaman. Attention and anticipation in fast visual-inertial navigation. *IEEE Transactions on Robotics*, 35(1):1–20, 2019.

[4] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Transactions on Robotics*, 33(1):1–21, February 2017. ISSN 1552-3098, 1941-0468. doi: 10.1109/TRO.2016.2597321. URL `https://ieeexplore.ieee.org/document/7557075/`.

[5] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-1101-5. doi: 10.1109/ICCV.2011.6126542. URL `http://dx.doi.org/10.1109/ICCV.2011.6126542`.

[6] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Timothy Furgale. Keyframe-based visualinertial odometry using nonlinear optimization. *International Journal of Robotics Research (IJRR)*, 2016.

[7] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.

[8] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31: 1147–1163, 2015.

[9] Fernando Nobre, Christoffer R. Heckman, and Gabe T. Sibley. Multi-sensor slam with online self-calibration and change detection. pages 764–774, 03 2017. ISBN 978-3-319-50114-7. doi: 10.1007/978-3-319-50115-4_66.

[10] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, Shanghai, China, May 2011. IEEE. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5979561. URL `http://ieeexplore.ieee.org/document/5979561/`.

[11] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A general optimization-based framework for local odometry estimation with multiple sensors, 2019.

[12] Sebastian Scherer, Daniel Dubé, and Andreas Zell. Using depth in visual simultaneous localisation and mapping. *2012 IEEE International Conference on Robotics and Automation*, pages 5216–5221, 2012.

[13] Gabe Sibley. A sliding window filter for slam. 2006.

[14] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, Daejeon, South Korea, October 2016. IEEE. ISBN 978-1-5090-3762-9. doi: 10.1109/IROS.2016.7759617. URL `http://ieeexplore.ieee.org/document/7759617/`.