Direct Vision-Based SLAM for Environments with Onboard Illumination

Senior Thesis, BS Computer Science

Ryan Slocum
Department of Computer Science
University of Colorado Boulder

***Abstract*—Many potential applications of mobile robotics are set in environments where there is little to no ambient light, such as underground or underwater. In these settings, reliable state estimation and mapping can be critical, especially when autonomous operation is required. Current visual simultaneous localization and mapping (SLAM) methods typically do not perform well under these conditions, due to their assumption of constant ambient lighting. Despite the current limitations, vision-based systems are extremely versatile due to their small form factor, low cost and power requirements, and general ubiquity. To address the gap in the current visual SLAM landscape, we propose a novel direct visual SLAM system that utilizes a model of the onboard illumination source to improve tracking performance. This work builds on previous advances in vision-based SLAM systems and light source estimation.**

I. Introduction

Visual simultaneous localization and mapping is a powerful and widely used method for robots to both localize themselves in their environments and build a representation of their surroundings, all through the visual information provided by an RGB camera. However, these methods perform poorly in environments without ambient lighting, such as underground or underwater, where light sources are manipulated by the robotic platforms themselves for the purposes of exploration.

The poor performance is due to visual SLAM systems making an assumption of lighting constancy, assuming that the brightness of a point in an image will be the same from one frame to the next. In most environments, the assumption is valid, given the constant intensity and location of ambient lighting. However, in low-light or dynamically-lit environments, this assumption is unfounded, and leads to degraded performance.

Despite these limitations, vision-based systems are an attractive option for exploratory field robotics. Due to their ubiquity, low cost, low power requirements, small form factor, and the semantic information provided, RGB cameras remain an extremely useful sensor to use on robotic platforms.

This work presents modifications to an existing SLAM algorithm, Direct Sparse Odometry (DSO) [2], which integrates information about the onboard illumination sources in order to improve tracking and motion estimates. These modifications are tested on a synthetic cave exploration dataset, and performance is compared against the original DSO algorithm to determine the improvement in Relative Pose Error and Absolute Trajectory Error.

II. Related Work

A. Current SLAM methods

1) Lidar-based systems: Currently, state-of-the-art methods in lidar SLAM such as Cartographer [3], and Fast-LIO [15] generally perform well in tracking and localization in conditions with no ambient lighting, especially when neglecting drift in the z-axis [8]. This is primarily due to the fact that they can directly measure depth without any additional light source, making them especially practical for this type of environment.

The robustness of lidar and lidar-inertial SLAM systems, invariant of illumination conditions, is compelling for low ambient-lighting environments. However, there are also many downsides to these types of systems, as they provide less semantic information and are more power-intensive, expensive, and cumbersome than passive cameras.

2) Indirect vision-based systems: Indirect methods, typically based on key-point detectors and descriptors, are the most popular visual SLAM methods today. These methods are naturally more lighting invariant, due to the careful design of the feature descriptors they employ. [5] presents a dataset for evaluating the performance of Visual-Inertial Odometry Systems with Onboard Illumination (OIVIO). OIVIO benchmarks several indirect vision-based methods, namely the Stereo and Mono versions of OKVIS [9] and ORB-SLAM [11], against the dataset of dark environments with onboard illumination.

OKVIS performed decently in most environments and lighting conditions, with relatively little loss of tracking. ORB-SLAM Stereo and Mono each performed worse, due to lack of integration of inertial data and tracking errors from motion blur. However, because these methods rely on the commonly used ORB and BRIEF feature descriptors, their performance is far more illumination invariant than direct methods.

3) Direct vision-based systems: Direct visual SLAM systems have the most potential for improvement in environments that require onboard illumination, given the fact that they consider light intensity on a pixel-by-pixel basis, and thus

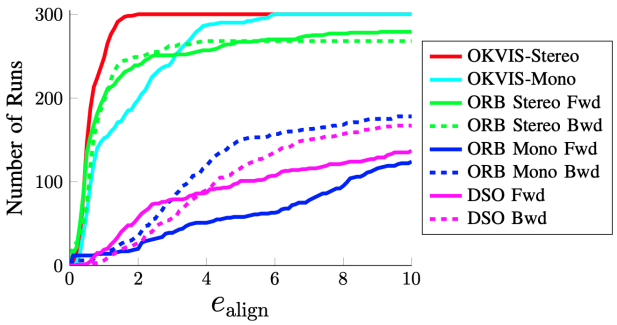


Fig. 1: Comparison of different visual methods in dark environments with onboard illumination from [5]; note the especially poor performance of DSO

would benefit from including more information about the lighting in the surrounding environment.

Direct methods operate by comparing pixel intensity across consecutive frames, generating an estimate of how the camera's pose has changed between frames. Due to the direct measurement of pixel intensity, direct methods typically function with a brightness constancy constraint [4], meaning they are especially susceptible to losing tracking in environments with dynamic lighting conditions. The direct comparison of intensity also leads to lower computational expense when compared with indirect methods.

A prominent example of monocular direct SLAM is Direct Sparse Odometry (DSO) [2], and its derivative Direct Sparse Visual-Inertial Odometry (VI-DSO) [14]. DSO was also tested in [5], and performed poorly, losing tracking frequently. One can imagine that the inertial version might perform better, but would struggle from the same issues of changing pixel intensity and photometric inconsistency. See Figure 1 for a comparison of DSO and the other aforementioned visual methods across the OIVIO dataset.

B. Other works in dynamic illumination conditions

There are other interesting and relevant contributions in the domain of state estimation and mapping in environments with no ambient lighting. Two examples focus on using external lighting for vehicular operation in the dark. [12] attempts to use vision to localize a vehicle as it drives through a dark environment by creating a map of artificial light sources it senses along the way. [10] evaluates the performance of a visual odometry system on a vehicle navigating by headlights in the dark. However, neither of these contributions make an effort towards light source modeling to compensate for the difficulties encountered in maintaining reliable odometry.

Additionally, [7] uses a near-infrared light source and imager, as well as a depth camera, to capture an image without light in the visual spectrum. This approach contributes an albedo-consistency bundle adjustment system that may be helpful in the case of pure visual SLAM as well. While this contribution is promising in terms of methodology, it's clear that using a normal RGB camera and visual light source would be more practical for most robotics applications, due to their ubiquity, cost, and the useful visual information that would be captured.

C. Light source estimation

Light source estimation is in many ways the inverse of the visual SLAM problem with onboard illumination. Instead of estimating the 3D geometry and camera pose given the lighting model, a light source estimation algorithm is given the 3D geometry and albedos of a scene and needs to estimate the position and intensity of the light source.

[6] presents a method in light source estimation that utilizes knowledge of the transfer of light to estimate the direction and intensity of the source of illumination in the scene. Light source estimation and visual SLAM with onboard lighting are closely related, and methods presented in this paper can be directly applied to the visual SLAM problem.

III. Methodology

The presented model is a visual SLAM system that builds on the Direct Sparse Odometry paper [2] with the incorporation of a simple lighting model. This system includes prior data about the onboard light source, including extrinsic (pose relative to camera) and intrinsic (light intensity) parameters. Using these parameters, we modify the energy functional calculation in DSO to account for the amount of additional light reflected from the onboard light source. The original per-pixel energy functional for the DSO algorithm is shown below in equation 3

$$E_{\boldsymbol{p}j} := \sum_{\boldsymbol{p}\in\mathcal{N}_{\boldsymbol{p}}} w_{\boldsymbol{p}} \left\| (I_j[\boldsymbol{p}'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[\boldsymbol{p}] - b_i) \right\|_{\gamma} \quad (1)$$

Where $\boldsymbol{p}$ is a pixel in a pattern of pixels $\mathcal{N}_{\boldsymbol{p}}$, I_j and I_i are the intensity of the target and source images, a and b are photometric factors associated with their respective images, t is the the exposure time of the respective image, w is a gradient dependent weighting, and $\|\cdot\|_{\gamma}$ is a Huber norm.

The original DSO algorithm sums these energy residuals across all pixels in the key-frames to generate a photometric error value, which is then minimized through Gauss-Newton optimization.

A. Modifications

The first step in integrating illumination information into this algorithm is to define the onboard illumination source parameters. In the case of our simple point-light model, there are two variables we define: the fixed position of the light source in the camera's coordinate frame $\boldsymbol{x}_{ill} \in \mathbb{R}^3$, and the intensity of the light source in lumens I_{ill}.

1) Lighting model: In order to integrate this information into the energy residual calculation, we need to calculate the intensity of the light reflected from a point in the image back to the camera. For this task, we use a point-light model to represent the radiation of illumination from the light source. Because there is no surface normal information available in the

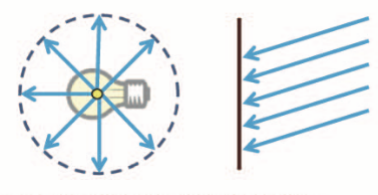


Fig. 2: Point light model illustration from [1]. Note that in the 3D case, illumination intensity at some point distance r from the light is proportional to $\frac{1}{r^2}$

DSO algorithm due to the sparsity of the tracked 3D geometry, we model only diffuse illumination on the surface. With these assumptions, we have the estimated reflected light intensity $\phi_{\boldsymbol{p}}$ for pixel $\boldsymbol{p}$:

$$\phi_{\boldsymbol{p}} = \frac{\lambda I_{ill}}{\|\Pi_c^{-1}(\boldsymbol{p}, d_{\boldsymbol{p}}) - \boldsymbol{x}_{ill}\|^2} \tag{2}$$

Where λ represents an arbitrary scaling parameter that can be adjusted for performance, and $\Pi_c^{-1} : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^3$ represents the projection function used in DSO that produces a 3-dimensional point given a pixel location and depth measurement.

2) Residual calculation: The above-resulting quantity $\phi_{\boldsymbol{p}}$ can be approximated as the amount of light reflected by the pixel $\boldsymbol{p}$ to the camera, from the onboard light source. We can then utilize this estimate in a modified residual calculation to take into account the changing lighting conditions frame-to-frame:

$$E_{\boldsymbol{p}j} := \sum_{\boldsymbol{p} \in \mathcal{N}_{\boldsymbol{p}}} w_{\boldsymbol{p}} \left\| (I_j[\boldsymbol{p}'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[\boldsymbol{p}] - b_i) - \phi_{\boldsymbol{p}} \right\|_\gamma \tag{3}$$

With this change to the energy functional, DSO has more information on the changes in pixel intensity between frames in low-ambient lighting environments. Because it takes into account the amount of light there is due to the light source, it can more accurately gauge the difference in pixel intensity.

After this point in the algorithm, back-end optimization for the summed residuals occurs and the camera poses and 3D geometry of the scene are calculated according to the original DSO algorithm.

B. Experiments

The above changes to the residual formulation were implemented in a fork of the DSO codebase. This version takes as command-line arguments the illumination intensity and the relative pose of the light source, allowing users to specify the parameters used for their onboard illumination setup.

1) Simulation data: To test these modifications in an environment with reliable ground-truth data and a consistent conditions, we utilized a cave simulation program called Spelunk, written by Mike Kasper of the Autonomous Robotics and Perception Group at CU Boulder. Spelunk procedurally generates cave-like environments for an agent with onboard lighting to explore. There are several advantages to using this program for testing, including the lack of photometric distortion (as the camera model used in simulation is a pinhole camera), the complete certainty of the ground truth data collected, and the light source that can be accurately modeled as a point-light.

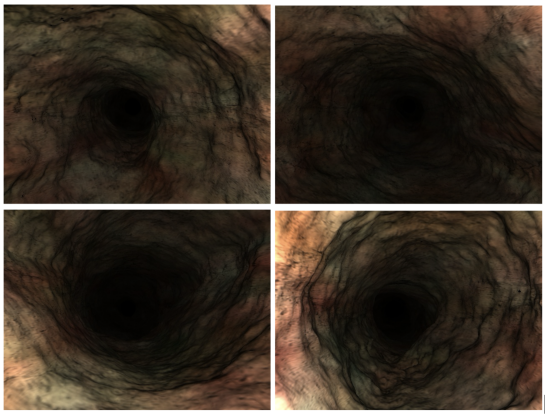


Fig. 3: Example frames from Spelunk simulation of procedurally generated cave

Name	Duration (sec)
backwards_2min	115
mostly_straight_2min	111
close_quarters_2min	104
completely_planar_2min	129
straight_fullspeed_3min	147
tricky_2min	108
frequent_stops_2min	114
turn_around_2min	117

TABLE I: Spelunk trajectories and durations

In order to test the robustness of the modified SLAM system, we created eight trajectories in the simulation, each with differing motion patterns ranging from extremely simple to difficult. Table I contains the name and duration of each trajectory used in testing.

Associated with each of these generated trajectories is a series of monocular RGB images captured by the simulated camera at about 20 frames per second, as well as ground truth pose information over time. The images and timestamps are fed to the original and modified versions of DSO at runtime, and the resulting estimated trajectories are compared with the ground truth data to determine whether improvements were made.

For each simulated trajectory, both the original and the modified DSO algorithms were run 5 times, allowing for a more detailed comparison between the performance of the two methods.

C. Evaluation

To determine whether measurable improvements were made in pose tracking with the modified SLAM system, we use two metrics: Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). According to [13], ATE is evaluated by

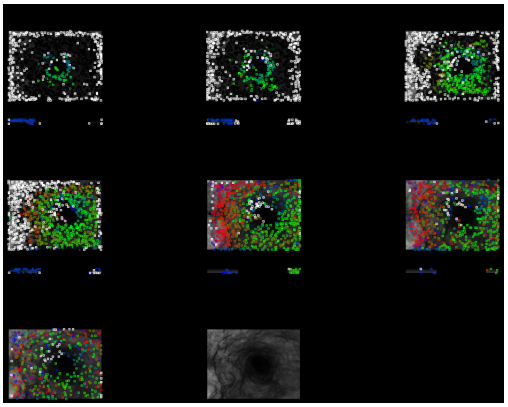


Fig. 4: Candidate points displayed on key-frames from modified DSO run

Trajectory Name	$\lambda = 1$		$\lambda = 2$	
	Modified	DSO	Modified	DSO
backwards_2min	3.84	**3.75**	**3.18**	3.75
mostly_straight_2min	**14.29**	14.90	**12.15**	14.90
close_quarters_2min	10.98	**10.71**	**8.83**	10.71
completely_planar_2min	**16.41**	18.97	**14.90**	18.97
straight_fullspeed_3min	**15.54**	24.51	**21.24**	24.51
tricky_2min	**5.18**	6.32	**5.86**	6.32
frequent_stops_2min	**7.01**	8.10	**7.65**	8.10
turn_around_2min	**9.67**	11.81	**6.88**	11.81

TABLE II: Average ATE for DSO and modified algorithm with various scaling factors. Better performances are bolded.

comparing the absolute differences between ground truth and estimated trajectories, and RPE is evaluated by comparing the difference between the ground truth and estimated trajectories in smaller time intervals, making it useful for visual odometry systems. To generate these measurements, we utilized evaluation scripts from the Computer Vision Group at the Technical University of Munich.

If a substantial reduction is made in either the ATE or RPE metrics, it stands to reason that our modification to DSO could be considered successful.

IV. Results

This section will present the results of the modified SLAM system, compared with those of original DSO. We will first look into the performance of the modified system according to the ATE and RPE metrics, with different scaling factors λ. Then we will briefly examine visual representations of the different trajectories generated.

A. Absolute Trajectory Error

Absolute Trajectory Error is commonly used to evaluate the level of drift in visual SLAM systems. In table II, we can see the average ATE values generated as a result of the five runs of both DSO and the modified system. Two different λ scaling factors are used, so that a comparison can be made between the weighting of the illumination residual.

Based on table II, we can see that the modified system resulted in a range of results, ranging from little to no change

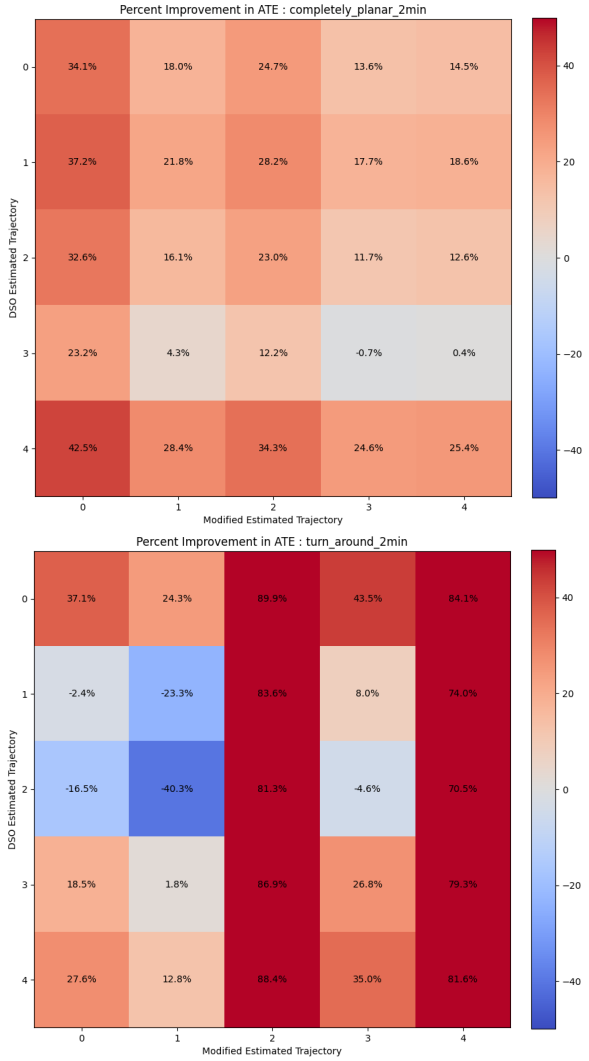


Fig. 5: ATE comparison matrices for two illustrative trajectories

in ATE, to a significant reductions in the range of 10 to 20 percent. It's also clear that on the whole, a λ value of 2 performs better in this scenario, showing that the scaling of the intensity makes a difference in the performance.

In figure 5, we can see the percent improvement in ATE for two different trajectories, across the 5 results for each of the algorithms (with $\lambda = 2$). In the first matrix, for the completely planar trajectory, there is a substantial across-the-board reduction in ATE, ranging from over 40 percent reduction to approximately no change. Several of the trajectories had results similar to these, with few results having an increase in ATE, and averaging somewhere around 20 percent reduction in ATE.

The second matrix in figure 5 displays the other typical behavior of the improved algorithm, with some runs of the modified system resulting in massive reductions in ATE (nearly 90 percent in some cases), and others resulting in increases in error. The turn around trajectory is characteristic

Trajectory Name	$\lambda = 1$		$\lambda = 2$	
	Modified	DSO	Modified	DSO
backwards_2min	4.36	**4.29**	**3.58**	4.29
mostly_straight_2min	**16.01**	16.32	**12.56**	16.32
close_quarters_2min	11.93	**11.72**	**9.04**	11.72
completely_planar_2min	**17.22**	20.68	**14.94**	20.68
straight_fullspeed_3min	**17.20**	27.89	**20.95**	27.89
tricky_2min	**5.51**	6.74	**6.07**	6.74
frequent_stops_2min	**7.88**	9.05	**8.47**	9.05
turn_around_2min	**9.31**	11.13	**6.88**	11.13

TABLE III: Average RPE for DSO and modified algorithm with various scaling factors. Better performances are bolded.

of the type of results seen in a few of the other trajectories, showing that the modified algorithm has the potential to significantly reduced the trajectory error in these types of trajectories, but it may be fickle at times.

B. Relative Pose Error

Relative Pose Error differs from ATE in the fact that it is calculated in shorter time periods, and it is more commonly used to evaluate the level of drift in visual odometry systems. Much like the ATE results above, in table III, we can see the average RPE values generated as a result of the five runs of both DSO and the modified system.

Based on the average RPE results, we can see that the reduction in ATE is mirrored in the RPE reduction, with, in some cases, even greater reduction. This is unsurprising, because RPE is a more lenient metric, and positive modifications made to DSO are likely to make greater improvements given the poor initial performance of DSO.

In the error matrices in figure 6, using the same trajectories as in the ATE comparison as examples, we see that the same trends extend to RPE. In the completely planar runs, there is an even greater improvement in RPE, averaging closer to 25 percent reduction. In the turn around trajectory, there the results seem to be slightly worse, with some significant increases in RPE, but with many of the same massive decreases as well.

Overall, the RPE and ATE trends are similar: a reduction in error with the modified system, but with some definite volatility that should be investigated further.

C. Estimated Trajectory Generation

From the evaluation of the ATE and RPE metrics, it is clear that there was some quantitative improvement in the modifications made to DSO, with the improvements being pronounced on certain types of trajectories. It's worth showing here the differences between the types of trajectories produced with DSO and the modified system by showing top-down plots of the estimated trajectories and ground truth. For these plots, we will use the trajectories produced with $\lambda = 2$, as this is where the modified algorithm seemed to perform best. In the below plots, X represents the left-and-right motion of the camera, and Z represents the forward motion of the camera (out of the image plane).

In figure 7, we see that some of the estimated trajectories are significantly improved by the introduction of illumination

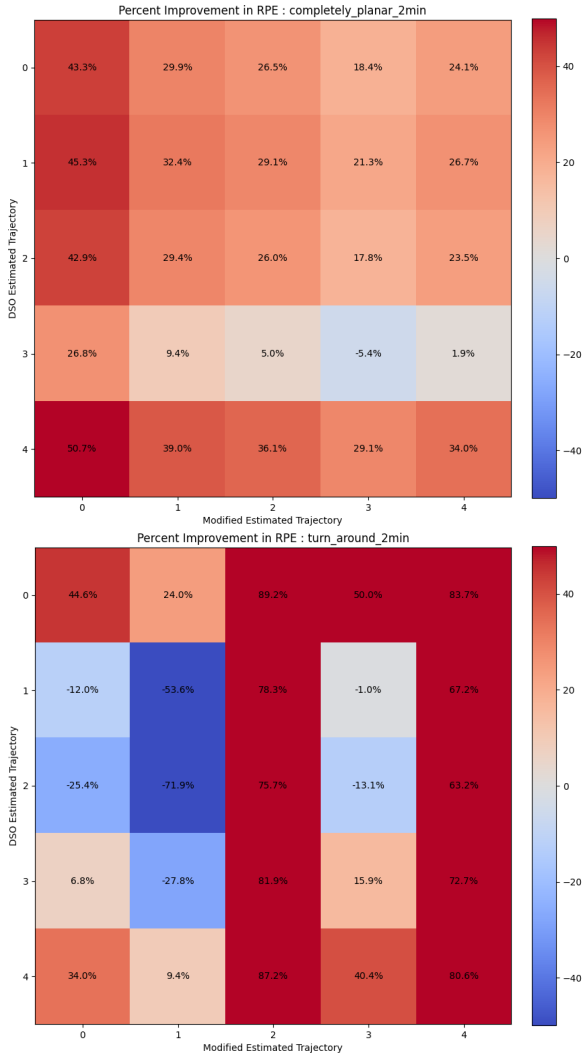
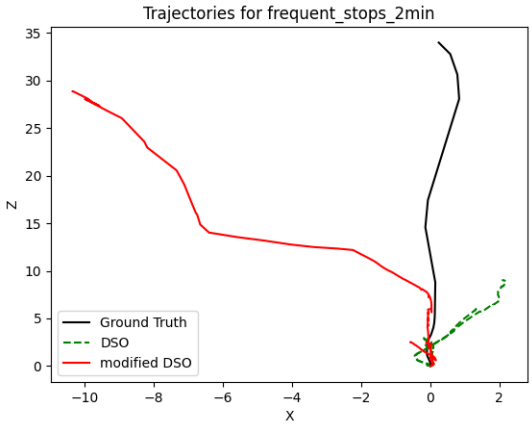


Fig. 6: RPE comparison matrices for two illustrative trajectories

information. The tracking better keeps the scale of the environment when compared with the DSO results. There is certainly still a significant amount of rotation error, as the estimated and ground truth trajectories are not well aligned.

In figure 8, we see an example of a trajectory that is also improved, but suffers from obvious rotational error. Once again, when compared with the DSO results, the modified system seems to better keep the scale of the environment and maintains a straight line just like the ground truth, but there is a fairly egregious amount of error in the initialization phase, when the camera seems to rotate about ninety degrees from its starting position.

Finally, in figure 9, we see that in particularly difficult trajectories, the modifications may not represent any substantial improvement. Both DSO and the modified system perform poorly on this trajectory, with no consistent result and a lot of both translational and rotational error.



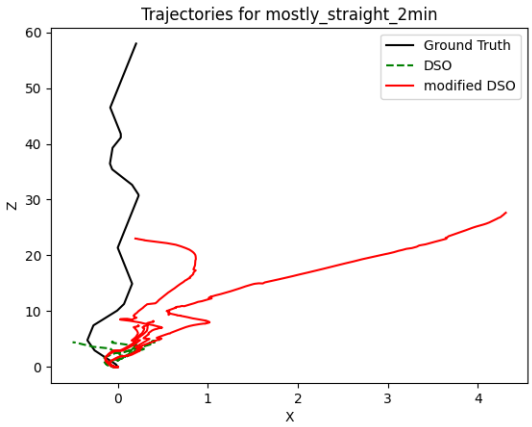


Fig. 7: Two trajectories where the modified SLAM algorithm made a notable improvement in tracking performance

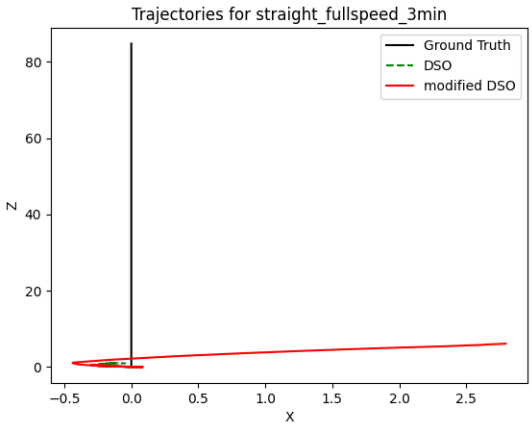


Fig. 8: The straight full-speed trajectory had better translational tracking, but still saw significant rotation error

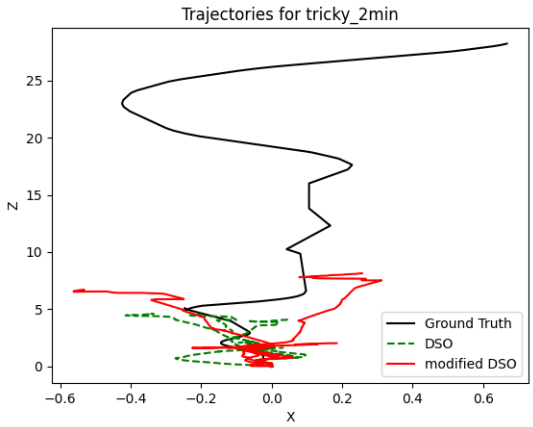


Fig. 9: The tricky trajectory saw little to no improved performance compared with the original DSO results

V. Future Work

One area of future research related to this work might be the implementation of a more sophisticated lighting model. Instead of the relatively simplistic point-light model with uniform spherical diffusion presented in this work, a more detailed model could be used. This would increase the fidelity of the residual calculation and more accurately model real-world light sources, but would also contribute to an increase in computational complexity.

Future works could also work to improve light diffusion modeling by integrating the estimated surface normal data of the 3D objects in the scene to better estimate the intensity of the light reflected to the camera.

Finally, future research in this area might aim to jointly solve for the light source parameters as a part of the function's optimization, in the same way that photometric parameters are jointly solved for in DSO. This future area might draw from the light source estimation methods presented in [6], and could increase the robustness of this algorithm to errors in calibration.

VI. Conclusion

We have presented a modified version of the Direct Sparse Odometry visual SLAM system that uses a simple lighting model to integrate onboard illumination information into its optimization procedure. This modified algorithm was then evaluated over a series of simulated data, and compared with the original algorithm with the metrics of ATE and RPE used.

While the method used here is certainly not usable in the field, due to volatile tracking results and significant rotational error, there is promise in the modifications presented. Given the reduction in ATE and RPE across the board using the modified algorithms, it seems that the integration of illumination information improves the performance of visual odometry systems in low ambient-lighting environments.

VII. Acknowledgements

I would like thank to Dr. Christoffer Heckman for agreeing to serve as my advisor and for offering his expertise and mentorship throughout the course of this project. I would also like to thank Dr. Alessandro Roncone and Dr. Bradley Hayes for taking the time to be on my thesis committee. Finally, I am grateful to Jasey Chanders, Ben Peterson, Miles Mena, and Bradley and Leslie Slocum for their encouragement and support with this thesis.

References

[1] Takahito Aoto et al. "Linear Estimation of 4-D Illumination Light Field from Diffuse Reflections". In: *2013 2nd IAPR Asian Conference on Pattern Recognition*. 2013, pp. 496–500. DOI: 10.1109/ACPR.2013.141.

[2] Jakob Engel, Vladlen Koltun, and Daniel Cremers. "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (July 2016). DOI: 10.1109/TPAMI.2017.2658577.

[3] Wolfgang Hess et al. "Real-time loop closure in 2D LIDAR SLAM". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1271–1278. DOI: 10.1109/ICRA.2016.7487258.

[4] Michal Irani and P. Anandan. "All About Direct Methods". In: 1999. URL: https://api.semanticscholar.org/CorpusID:7315270.

[5] Mike Kasper, Steve McGuire, and Christoffer Heckman. "A Benchmark for Visual-Inertial Odometry Systems Employing Onboard Illumination". In: *Intelligent Robots and Systems (IROS)*. 2019.

[6] Mike Kasper et al. *Light Source Estimation with Analytical Path-tracing*. 2017. arXiv: 1701.04101 [cs.CV].

[7] Da Kong, Yu Zhang, and Weichen Dai. "Direct Near-Infrared-Depth Visual SLAM With Active Lighting". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7057–7064. DOI: 10.1109/LRA.2021.3096741.

[8] Anton Koval, Christoforos Kanellakis, and George Nikolakopoulos. "Evaluation of Lidar-based 3D SLAM algorithms in SubT environment". In: *IFAC-PapersOnLine* 55.38 (2022). 13th IFAC Symposium on Robot Control SYROCO 2022, pp. 126–131. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2023.01.144. URL: https://www.sciencedirect.com/science/article/pii/S2405896323001519.

[9] Stefan Leutenegger et al. "Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization". In: *The International Journal of Robotics Research* 34 (Feb. 2014). DOI: 10.1177/0278364914554813.

[10] Kirk MacTavish, Michael Paton, and Timothy D. Barfoot. "Night Rider: Visual Odometry Using Headlights". In: *2017 14th Conference on Computer and Robot Vision (CRV)*. 2017, pp. 314–320. DOI: 10.1109/CRV.2017.48.

[11] Raul Mur-Artal and Juan D. Tardos. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (Oct. 2017), pp. 1255–1262. DOI: 10.1109/tro.2017.2705103. URL: https://doi.org/10.1109%2Ftro.2017.2705103.

[12] Peter Nelson et al. "From dusk till dawn: Localisation at night using artificial light sources". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 5245–5252. DOI: 10.1109/ICRA.2015.7139930.

[13] David Prokhorov et al. "Measuring robustness of visual slam". In: *2019 16th International conference on machine vision applications (MVA)*. IEEE. 2019, pp. 1–6.

[14] Lukas von Stumberg, Vladyslav Usenko, and Daniel Cremers. "Direct Sparse Visual-Inertial Odometry using Dynamic Marginalization". In: *CoRR* abs/1804.05625 (2018). arXiv: 1804.05625. URL: http://arxiv.org/abs/1804.05625.

[15] Wei Xu and Fu Zhang. "FAST-LIO: A Fast, Robust LiDAR-inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter". In: *CoRR* abs/2010.08196 (2020). arXiv: 2010.08196. URL: https://arxiv.org/abs/2010.08196.