Visual-Inertial Odometry Robustness to Adverse Conditions in Proximity Flight

Alexander P. Teacu
Dept. of Aerospace Engineering
University of Maryland
College Park, MD, USA
ateacu@umd.edu

Derek A. Paley
Dept. of Aerospace Engineering
and Institute for Systems Research
University of Maryland
College Park, MD, USA
dpaley@umd.edu

***Abstract*—This paper tests and evaluates the performance of visual-inertial odometry (VIO) under adverse visual factors within the context of autonomous proximity flight. VIO is widely used in UAV autonomy for micro air vehicles (MAVs) due to its cost-effectiveness and the small size of required sensors. Since VIO relies on a camera, visual factors can significantly impact its performance; however, there is a general lack of public VIO flight datasets that contain adverse visual factors. Here we describe the use of a MAV platform to collect real-world visual-inertial datasets that contain feature-rich environments, feature-sparse environments, reflections, and repeating features, along with the ground-truth pose of the MAV. However, visual factors, such as fire, dust, and smoke are challenging or hazardous to reproduce in a controlled environment. To produce datasets containing these visual factors, we augment the real-world images in a computational simulation environment. The real and augmented datasets along with ground-truth data are used to evaluate VIO robustness through a combination of practical performance metrics.**

I. INTRODUCTION

Unmanned Aerial Systems (UAS) are a uniquely useful tool in hazardous situations such as emergency response. They have gathered recent attention [1], [2] due to their potential to inform users of critical information with less risk and cost. Some situations demand close approach to obstacles and hazards to collect adequate data, often indoors, where maneuverability of a UAV is critical. Generally, a pilot is required for the UAV in this scenario, which necessitates a line of sight to the vehicle and/or a live video feed. Alternatively, increased UAS autonomy has the potential to enable non-line-of-sight operations and a lower communications bandwidth, therefore reducing the burden on human operators.

However, UAS autonomy faces several challenges in hazardous situations. UAS autonomy commonly uses a Global Navigation Satellite System (GNSS) such as the Global Positioning System (GPS) for localizing the vehicle. Unfortunately, GNSS localization is not always available in hazardous or indoors environments. To address localization without GNSS, a wide range of odometry methods using different sensing modalities have been developed in the literature [3], [4]. Many of these odometry methods require sensing modalities that are expensive, high-power, or heavy such as LiDAR, which limits commercial adoption and usage in small vehicles. In contrast, visual-inertial odometry (VIO) can be applied to a micro air vehicle (MAV) using a cheap and lightweight camera and IMU. Applications of VIO go beyond emergency response; VIO robustness studies have been conducted for underwater environments [5] and for other planets [6], [7].

Because VIO requires visual information to function, it may become impaired due to visual factors that may be present in hazardous environments. Adverse visual factors such as smoke, fog, dust, fire, glare, or a dirty lens can severely impact visibility. Visual odometry methods may also fail in other conditions that don't impact visibility, such as when the camera is looking at a featureless image, when features look identical to one another, or in a dynamic environment with moving features. Additionally, VIO is not the only autonomy process that suffers from visual failure modes; visual-inertial solutions are also widely used in simultaneous localization and mapping (SLAM) [8], which exhibits many of the same failure modes.

Some studies have attempted to investigate and address VIO failure modes. A VIO algorithm to improve robustness to variable lighting conditions was presented in [9]; in [10], lens contaminants were computationally removed, whereas [11] tested VIO algorithms on both lighting conditions and lens contamination; [12] dealt with foggy weather and similarly [13] addressed smoke; and [14] addressed dynamic features caused by motion in the environment; [15] tested visual odometry performance in rain. A similar study has investigated adverse visual effects for vision-based detection of UAVs [16]; however, that study focuses on detection of UAVs in images, not odometry given images from a UAV.

Nonetheless, there is an absence of publicly available visual-inertial datasets that contain these visual factors; such a dataset could significantly accelerate research and development of VIO and SLAM. Existing flight datasets may also be prohibitively difficult to replicate, due to the uncertain nature of flight and visual factors such as smoke and dust. Existing public visual datasets containing adverse visual factors are either exclusive to ground vehicles [17] or are not entirely reproducible [18].

Research supported by U.S. Army Grant No. W911W62120003

This paper aims to address both the dataset availability issue and reproducibility issue by providing a reproducible methodology to create such datasets. A two-pronged approach is applied: first, real-world datasets containing non-synthetic adverse visual factors are collected using a robotic setup that has high reproducibility and accurate ground truth data. Second, visual factors that have prohibitive logistical requirements or are otherwise non-reproducible (i.e. smoke) are computationally added via simulation on top of real-world flight datasets in order to create an augmented dataset.

Visual dataset augmentation has been investigated in order to address robustness in machine-learning models [19]. Rothmeier et al. [20] augmented a driving dataset using machine learning models to apply fog, rain, and snow augmentations onto images. Nikolov [21] augmented a pedestrian dataset with fog, rain, and snow using the Unity game engine. Herbert [22] augmented various images with raindrops, frost, and water present on the camera lens.

In order to accurately mimic positionally dependent visual factors, an augmentation system should accurately and predictably model the spatial configuration and the time evolution of the environment. This rules out most existing machine-learning approaches to image generation, as they currently are generally inconsistent with respect to physics and time and may introduce undue noise into the dataset. One approach that addresses these problems is the use of a 3D graphics program or 3D game engine such as Unity [21].

The approach to visual factor dataset creation described in this paper uses the 3D graphics engine Blender in combination with ground-truth position and a 3D scan of the environment to augment existing visual-inertial dataset images with additional visual factors. Unlike Unity, Blender is not a game engine, and has features focused on photorealism such as physically-based rendering, fluid simulations, ray-tracing, and compositing [23].

Overall, this paper presents the following contributions: (1) an experimental framework for creating accurate, reproducible real-world visual-inertial datasets with adverse visual factors; (2) a simulation-based framework for augmentation of complex and dynamic visual factors to real-world visual-inertial datasets; and (3) a quantitative evaluation of VIO performance under these experimental and simulated visual factors. The significance of this work lies in its potential to advance knowledge of how adverse visual factors affect positional awareness of micro air vehicles, and is a preliminary step towards mitigating them.

Section II discusses the creation of visual-inertial datasets containing various adverse visual factors. First, the methodology of collecting a real-world dataset is discussed including an example of such a dataset; next, the system for augmenting real-world data with visual factors is discussed including the creation of an example augmented dataset leveraging the real-world EuRoC [24] flight dataset. Section III then proceeds to evaluate the performance of VIO under these visual factors through quantitative analysis. Section IV concludes the paper.

II. CREATION OF VISUAL-INERTIAL DATASETS CONTAINING ADVERSE VISUAL FACTORS

First, the collection of a real-world experimental visual-inertial dataset containing adverse visual factors is described. Second, an existing dataset, the EuRoC MAV [24] flight dataset, is augmented with a simulation of the adverse visual factor of smoke.

A. Experimental Data Collection

In order to ensure reproducibility of the flight trajectory, a MAV is mounted on a UR5 Universal Robotics robot arm that maneuvers the end effector through a pre-determined trajectory with sub-millimeter precision. At each point on this trajectory, the robot arm position is computed through the use of encoders and forward kinematics and used to determine the ground-truth location of the MAV camera.

The robot arm is surrounded with foam panels of uniform shape and size, such that the visual factors of the data collection are reproducible; see Figure 1. To modify these visual factors in a reproducible way, foam panels are switched out for equivalents of the same shape and size, but with a different surface; for example, replacing a high-feature panel with a low-feature panel. The location of the adverse effect panel, shown in red in Figure 1, can then be used to correlate errors in visual odometry with the presence of the panel.

The MAV mounted to the robot arm in this experiment is custom-built [25] using commercial off-the-shelf parts. The onboard camera used to create this dataset is the IMX412 high-resolution low-light sensor from ModalAI, mounted at the front of the MAV, shown at the origin of the reference frame in Figure 2.

The constraints of this relatively small experimental setup raise concerns about the scalability of this data to realistic scenarios of UAV flight, however, there are some factors which bring this setup in-line with real flight datasets like the EuRoC dataset [24]. While the position of this setup is heavily constrained, the velocity and accelerations are less so. The fundamental task of VIO is to find the difference in pose (i.e. velocity) between two points in time, given the input of current and previous images and IMU readings. Therefore, if the distance to objects, the change between images, and the acceleration remain similar to real flight, then the scale of position is unlikely to be relevant in contributing to relative error. This setup is capable of producing velocities and accelerations of similar magnitude to real flight in the EuRoC dataset, shown in Figures 4a and 4b, respectively.

The trajectory through which the UR5 robot arm moves the MAV is inspired by Lissajous curves, i.e., curves parameterized by sinusoidal functions. Several sinusoidal functions with carefully-chosen frequencies and phase shifts are chosen to parameterize the trajectory in four axes relative to the MAV camera shown in Figure 2. The frequencies and phase shifts are chosen such that the functions will not repeat a given MAV position and orientation: this ensures that as time increases, the trajectory will continue to sweep out more of the search space. Additionally, the sinusoidal motion ensures a repeatably differentiable and bounded state configuration,

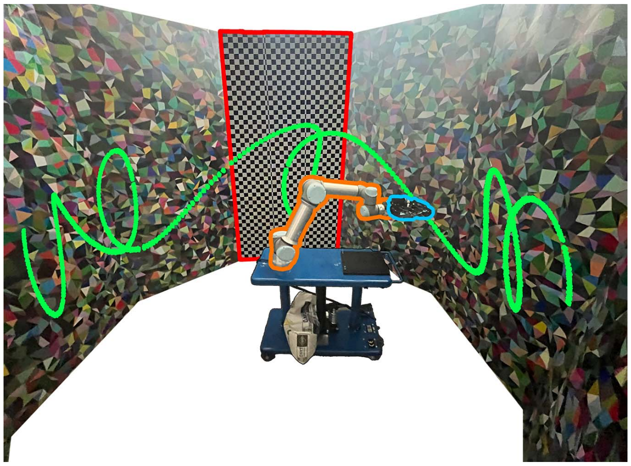


Fig. 1: Data collection setup. (blue) MAV; (orange) UR5 robot arm; (green) ground-truth trajectory; and (red) foam panel which is replaced with a different panel between dataset sequences.

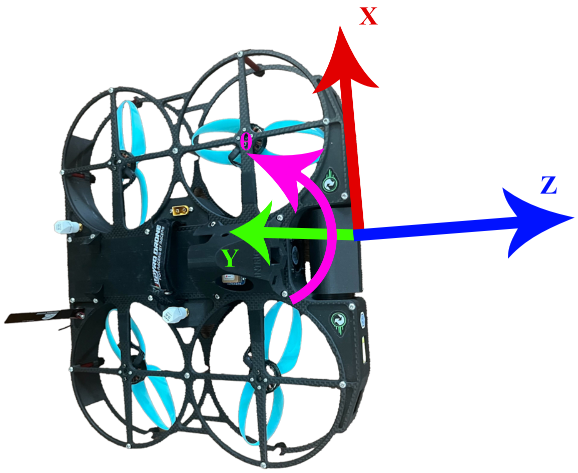


Fig. 2: MAV camera reference frame, including X, Y, and Z axes. θ, yaw, is defined as counterclockwise along the positive Y-axis. The origin of the reference frame is placed at the lens of the IMX412 camera on the front of Intrigue. During data collection, the MAV is upright with the Y-axis vertical, as shown in Figure 1.

resulting in a physically possible trajectory with constraints on the maximum acceleration and velocity of the camera. Figure 5 depicts the Lissajous-like parameterization process along the four MAV camera axes.

Six sequences were collected using the setup in Figure 1 for eight trajectories each, giving a total of 48 sub-sequences. First, a feature-rich sequence was collected serving as the baseline for an environment with no adverse visual factors. Then, five other sequences were collected with adverse visual factors: one with few features (feature-sparse), one with identical repeating features in the form of a checkerboard pattern glued to the foam panel, one with mirrored features provided by applying a layer of mylar to the foam panel, one with direct glare from a lamp, and one with a dirty lens. All of these visual factors are depicted in Figure 6.

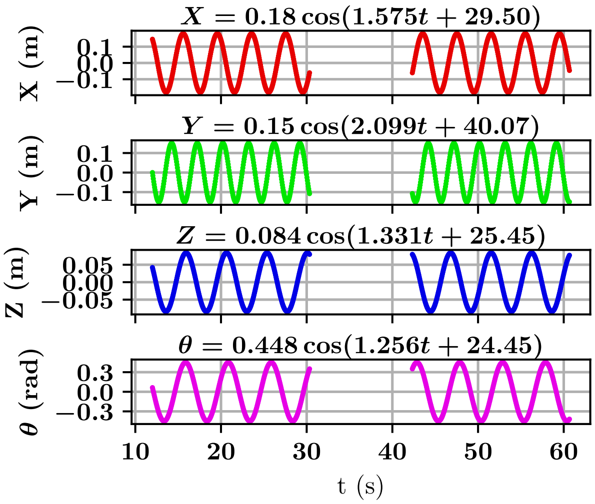
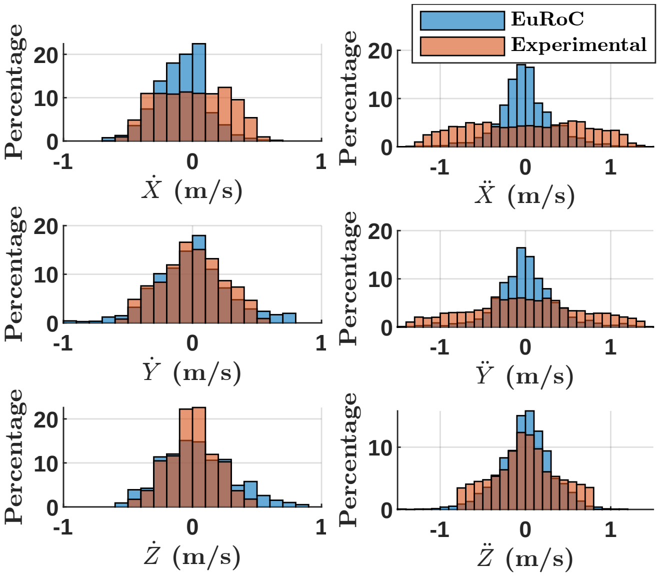


Fig. 3: The four state variables relative to the camera frame on the MAV during two sequences of dataset collection, with a pause to separate the sequences.

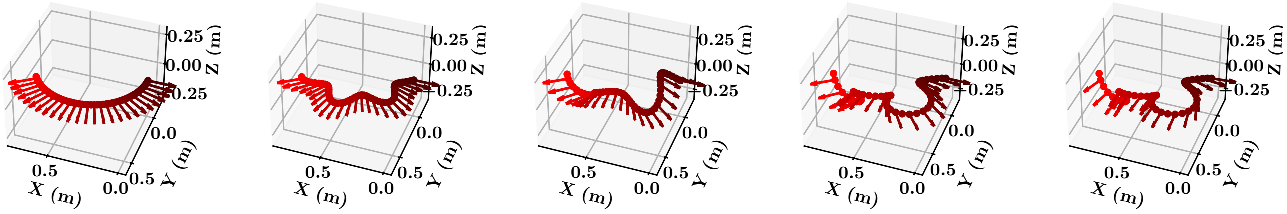


(a) Velocity

(b) Acceleration

Fig. 4: Comparison of a velocity and acceleration distribution created with the setup shown in Figure 1 compared to the EuRoC V1_01_easy dataset, relative to the MAV-fixed reference frame defined in Figure 2.

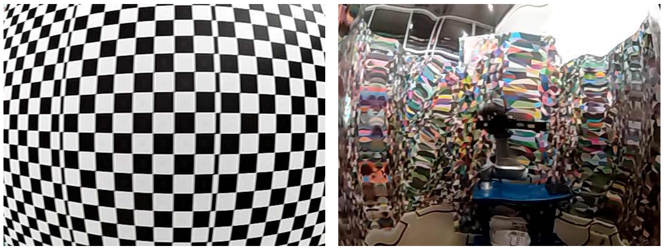
The MAV was mounted to the UR5 arm by using a custom 3D-printed mount and a network connection was initiated to the ground station [25]. Then, the UR5 arm began the planned trajectory as the MAV began to record a Rosbag containing the IMU data and images captured by the onboard IMX412 camera. Once the UR5 arm reached its end, the Rosbag was closed and downloaded for offboard analysis. In between each sequence, the board highlighted in Figure 1



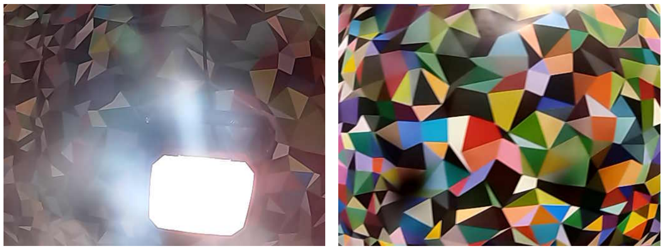
a) Baseline b) Z-parameterized c) ZY-parameterized d) ZYX-parameterized e) ZYXθ-parameterized

Fig. 5: Stages of parameterizing the trajectory. Each state variable was parameterized in the order written with respect to the MAV reference frame shown in Figure 2. For example, ZYX-parameterized refers to first displacing the trajectory along the Z-axis, then the Y axis, and then the X axis, in sequence. Arrow vectors are used to signify the forward direction of the camera at that point along the trajectory, i.e., the Z-axis in Figure 2.

(a) Feature-rich (baseline) (b) Feature-sparse



(c) Identical features (d) Mirrored features



(e) Direct glare (f) Dirty lens

Fig. 6: Samples of each experimental sequence containing a unique visual factor panel from a real-world dataset shown in Figure 1.

was replaced with a different board depending on the visual factor being investigated, shown in Figure 6.

B. Dataset Augmentation Using Blender

Certain visual factors are prohibitively expensive, non-reproducible, or problematic to collect large amounts of visual-inertial data on, such as smoke. Here we present simulation of smoke using Blender and subsequent augmentation of the smoke to the EuRoC dataset [24].

Various computational methods exist to photo-realistically simulate visual factors in Blender, classified into fluid and material simulations. Fluids such as fog, smoke, etc. are simulated using smoothed-particle hydrodynamics (SPH), achieving physically-accurate results by approximating fluid dynamics solutions with a set of particles [26]. Materials are simulated through two methods: principled volume shaders and bidirectional scattering distribution function (BSDF) shaders. Volumetric effects such as smoke and fog leverage the principled volume shader, which includes models for scattering, absorption, and blackbody emission. Other visual factors such as glare and a dynamic environment require solid surfaces and are simulated using the BSDF shader, which utilizes physically-based rendering (PBR) surface shading techniques to produce photorealistic results.

The 3D environment of the original dataset is re-created in Blender by using a 3D scan provided in the EuRoC dataset. The visual factor is then simulated within this virtual 3D environment. Each image in the original dataset is re-created by moving the virtual camera to the ground-truth location of image capture within the 3D environment, but with the simulated visual factor present. The 3D environment itself is removed from the image and made invisible by using a holdout shader, leaving only the visual factor itself, photo-realistically affected by the invisible 3D environment. This visual-factor-only image is then overlaid on top of the original real-world image using compositing, resulting in a realistically-augmented image in Figure 7.

The EuRoC V1_01_easy sequence was selected for smoke augmentation. The parameters of the smoke simulation are chosen to provide a variety of situations in the same dataset at different time intervals, depicted in Table I: there are small spots of concentrated smoke, at $t = 0$; spots of hazy, partially visible smoke at $t = 30s$; and spots that completely block out the camera at $t = 14s$. Two sequences in this dataset are produced: the unmodified original images, labeled as the feature-rich (baseline) sequence, and the smoke-augmented images, depicted side-by-side in Table I.

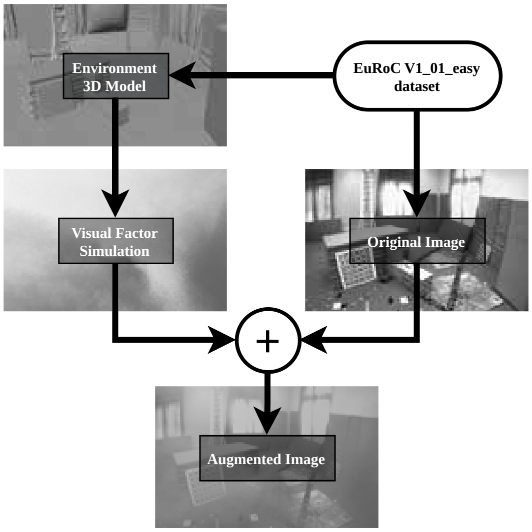
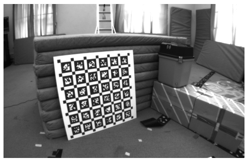


Fig. 7: Concept of dataset augmentation using Blender. The environment 3D model is imported into Blender and used to simulate the visual factor, i.e. smoke in this case. Then, the visual factor is combined using compositing, denoted by $+$, in order to produce a final augmented image, which contains both the original image and the new visual factor.

TABLE I: Images of smoke-augmented EuRoC V1_01_easy dataset, at select times

Time (s)	Feature-rich (baseline)	Smoke
$t = 0$		
$t = 14$		
$t = 30$		

III. VIO PERFORMANCE EVALUATION UNDER ADVERSE VISUAL FACTORS

A representative sample of visual-inertial odometry algorithms were selected for evaluation: VINS-Mono [27] and OpenVINS [28]. Two datasets were evaluated: the real-world dataset collected and presented in Section II-A with feature-rich, feature-sparse, repeating feature, and mirrored feature sequences; and the augmented EuRoC dataset presented in Section II-B with feature-rich and smoke sequences.

The augmented smoke dataset was evaluated on OpenVINS, using the OpenVINS default parameters for the EuRoC dataset. The real-world dataset was evaluated on VINS-Mono, also using the default parameters for the EuRoC dataset, but with camera calibration done for the IMX412 via Kalibr [29].

A. *Error Metrics*

VIO estimates a vehicle's pose with respect to time, which typically includes translational Cartesian position $P = (x, y, z)$ and rotation/orientation quaternion $R = (w_Q, x_Q, y_Q, z_Q)$.

The error metric ϵ_{P_i} for position is absolute difference of the VIO estimate $\vec{P}_{VIO}$ to the ground-truth position $\vec{P}_{GT}$, where

$$\epsilon_{P_i} = |\vec{P}_{VIO} - \vec{P}_{GT}| \tag{1}$$

The error metric for orientation is computed as the angle between the estimated orientation quaternion and its ground truth counterpart. Let q_{VIO_i} denote the orientation quaternion corresponding to the VIO ouput for the image i at time t in the sequence, and let q_{GT_i} similarly represent the ground truth orientation quaternion. The difference in these quaternions is then given by

$$\epsilon_{q_i} = q_{VIO_i} \cdot q_{GT_i}^{-1} \tag{2}$$

Let the scalar component of ϵ_{q_i} be denoted by $w_{\epsilon_{q_i}}$. Then, the angular error in rotation is

$$\epsilon_{R_i} = \arccos\left(w_{\epsilon_{q_i}}\right) \tag{3}$$

where the domain of $\arccos(\theta)$ is accounted for by limiting the range of ϵ_{R_i} to $[0, 2\pi]$.

To combine the error metrics at each timestep and arrive at unified position and orientation error metrics for the entire data sequence, we use the mean absolute error (MAE), defined below, with E_P representing translational error and E_R representing rotational error, i.e.,

$$E_P = \frac{1}{N}\sum_{i=1}^{N} |\epsilon_{P_i}| \tag{4}$$

$$E_R = \frac{1}{N}\sum_{i=1}^{N} |\epsilon_{R_i}| \tag{5}$$

where N is the total number of images and i is the image number.

B. *Experimental Dataset Results*

The real-world experimental dataset was run on six sequences, each containing a unique visual factor as depicted in Figure 6. All of the sequences were evaluated in VINS-Mono and then transformed to begin at the origin along with the ground-truth trajectory. The error at each point in time was collected by linearly resampling the sequences to 10 Hz, and then computing the error metrics defined in Section III-A. The translational and rotational errors with respect to time

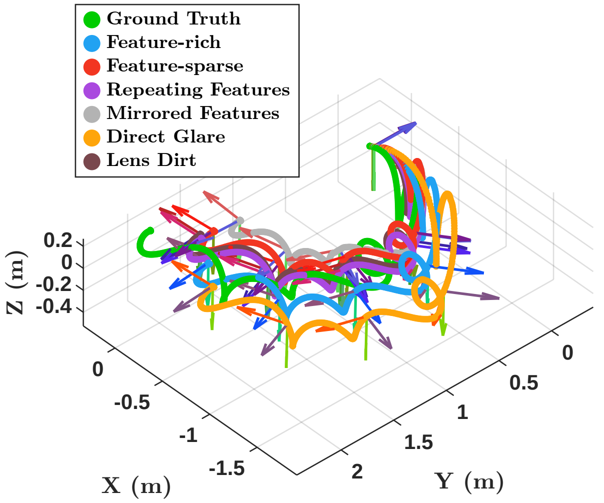


Fig. 8: Trajectories produced by VINS-Mono on first trajectory of real-world experimental dataset, for selected visual factors of feature rich, feature-sparse, repeating features, and mirrored features. All trajectories are aligned to the origin inertial reference frame, such that all errors are 0 at $t = 0$.

are shown alongside a count of features tracked in Figure 9 and are summarized using MAE in Figures 10a and 10b.

The translational error of the trajectory produced by VINS-Mono was lowest on the baseline feature-rich sequence, and highest on the direct glare sequence. The feature-rich sequence also displayed the least variation in errors, with a standard deviation of only 0.156m. Despite losing over 60% of features tracked at around $t = 40$s, the feature-sparse sequence showed little change in error. Notably, the feature-rich and direct glare sequences had higher translational error at the start of the trajectory compared to the other sequences. This suggests a limitation of the experimental setup: small differences in the sequences due to noise or different features detected may result in a different initialization of the reference frame by VIO, misaligning the trajectory from the start. This initial discrepancy in alignment is reflected in the trajectories shown in Figure 8, where trajectories near the start of the dataset at the origin are misaligned with respect to each other, despite having identical initial orientations. In an ideal world, these trajectories would all be identical (except for lens dirt) until the adverse visual factor enters the view around $t = 30$s near the middle of the trajectory.

The rotational error across all sequences but one is nearly identical, suggesting that the visual factor did not significantly impact orientation estimation. The one notable exception is the mirrored features sequence, which increased significantly in orientation error by around 6 degrees around $t = 30$s, corresponding to the first all-adverse image shown in Figure 1, with a corresponding drop in features tracked shown in Figure 9. However, the drop in features tracked for the feature-sparse sequence did not correspond to an increase in the rotational error, suggesting that the bad features introduced by the mirror uniquely contributed to rotational error.

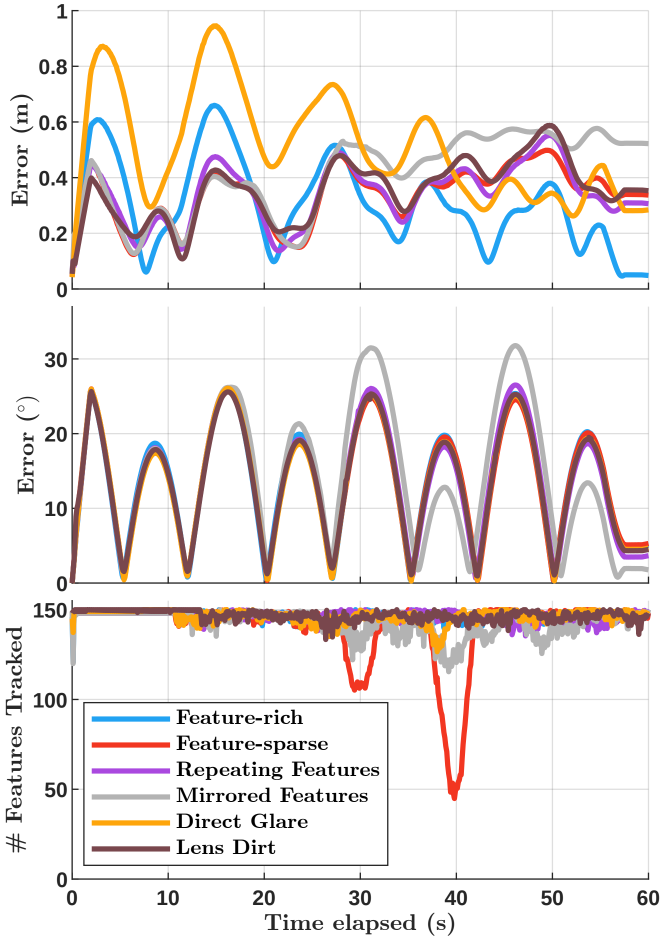
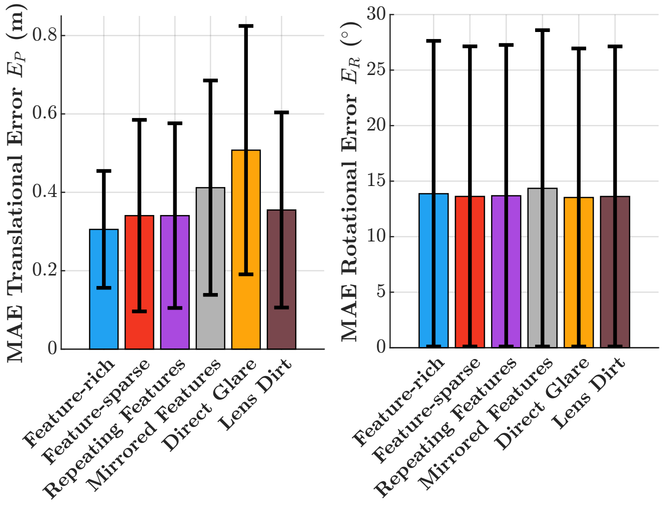


Fig. 9: Translational errors (top), rotational errors (middle), and corresponding features tracked (bottom) by VINS-Mono on the real-world experimental dataset, for the selected visual factors of feature-rich, feature-sparse, repeating features, mirrored features, direct glare, and lens dirt.



(a) Translational MAE E_P

(b) Rotational MAE E_R

Fig. 10: Translational and rotational mean absolute error (MAE) for the real-world dataset shown in Figure 8.

C. Augmented Smoke Dataset Results

The augmented dataset included two sequences: one baseline (feature-rich) sequence without smoke, and one augmented sequence with smoke. Both sequences were made from the same EuRoC V1_01_easy dataset, which provided ground truth collected from a Vicon motion capture system. Both of the sequences were evaluated in OpenVINS. The error at each point in time was collected by linearly resampling the sequences to 10 Hz, and then computing the error metrics defined in Section III-A. The translational and rotational errors with respect to time are shown in Figure 12 and are summarized using MAE in Figures 13a and 13b.

The smoke sequence had the smoke visual factor in view throughout nearly the entire dataset to various degrees. Around t=14s, as shown in Table I, the camera passes through the densest part of the smoke cloud, fully obscuring the environment. This moment coincides with a substantial jump in translational error compared to baseline, which is maintained throughout the rest of the sequence. The translational error was nearly doubled by the smoke augmentation, with an MAE of 0.99 m versus 0.59 m on the baseline dataset.

The rotational error was much lower in the augmented dataset, with both feature-rich and smoke-augmented sequences having rotational MAEs under three degrees. However, in both sequences there were several transient spikes of extremely high rotational error before OpenVINS seems to have re-localized and corrected. As expected, the rotational error on the smoke sequence is higher than the baseline, although only slightly, at less than 0.2 degrees higher than the baseline MAE.

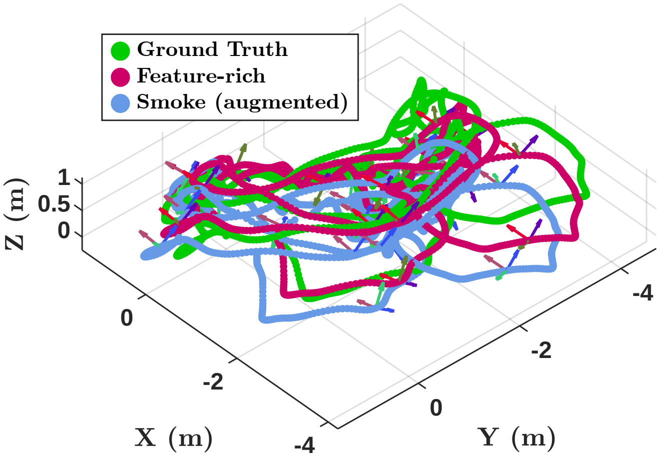


Fig. 11: State translational trajectory from OpenVINS VIO on the smoke-augmented EuRoC V1_01_easy dataset. Brightness of line indicates time elapsed, with darkest corresponding to $t = 0$s, and brightest corresponding to the end of the trajectory.

IV. CONCLUSION

Research into visual-inertial odometry robustness is vital for evaluating autonomous micro air vehicle (MAV) operations in hazardous and challenging environments. Three

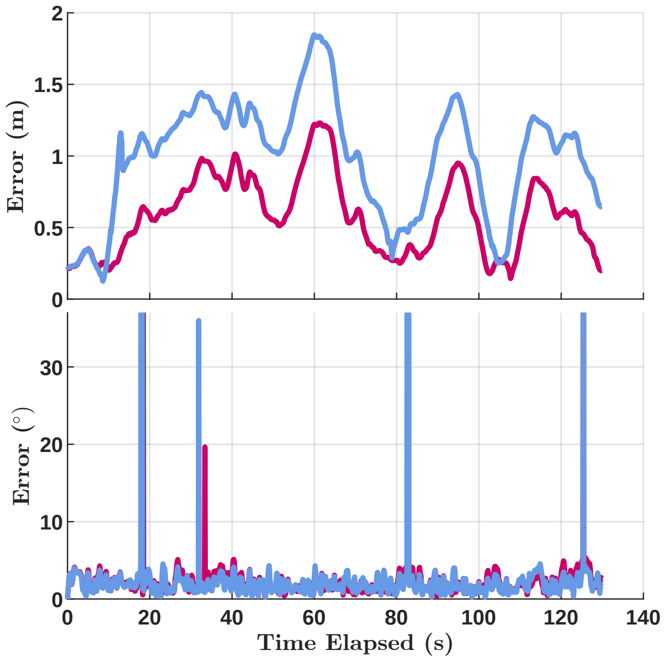
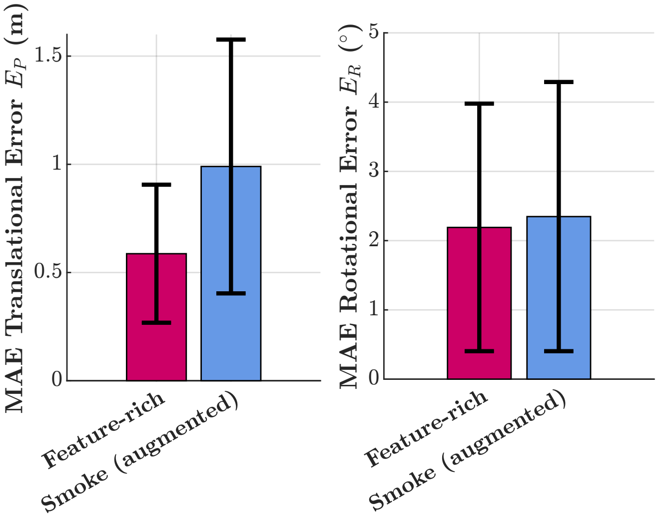


Fig. 12: Translational errors (top) and rotational errors (middle) by OpenVINS on the EuRoC V1_01_easy data with respect to time, comparing the smoke-augmented sequence to the original feature-rich non-augmented sequence.



(a) Translational MAE E_P

(b) Rotational MAE E_R

Fig. 13: Translational and rotational mean absolute error (MAE) for the smoke-augmented dataset shown in Figure 11.

major contributions are presented: an experimental setup to capture real-world datasets containing adverse visual factors; a system to augment real-world datasets with simulated adverse visual factors; and evaluation of VIO robustness to adverse visual factors present in these datasets. These contributions enable further understanding into where MAV positional awareness breaks down in challenging, hazardous environments of particular interest to MAV operations.

Ongoing work focuses on studying more recent VIO algorithms and contrasting their performances relative to each other. Further work includes increasing sample size through the collection of more datasets as described in Section II and their evaluation described in Section III along with creating an augmented dataset using the collected feature-rich experimental dataset, to allow direct comparison of simulated versus experimental visual factors.

V. ACKNOWLEDGEMENTS

The authors would like to thank Animesh Shastry for his work on the MAV and continued support in its application, Ivan Penskiy for his assistance in lab operations, and Anthony Gong for his input throughout. This work was funded by U.S. Army Grant No. W911W62120003.

REFERENCES

[1] S. Martinez-Rozas, R. Rey, D. Alejo, D. Acedo, J. A. Cobano, A. Rodriguez-Ramos, P. Campoy, L. Merino, and F. Caballero, "Skyeye team at MBZIRC 2020: A team of aerial and ground robots for GPS-denied autonomous fire extinguishing in an urban building scenario," *CoRR*, vol. abs/2104.01834, 2021.

[2] C. Chen, Z. Yao, J. Jiang, X. Pan, X. He, Z. Chen, and B. Wang, "Smokenav: Millimeter-wave-radar/inertial measurement unit integrated positioning and semantic mapping in visually degraded environments for first responders," *Advanced Intelligent Systems*, vol. 6, no. 12, p. 2400241, 2024.

[3] G. Balamurugan, J. Valarmathi, and V. P. S. Naidu, "Survey on UAV navigation in GPS denied environments," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System*, pp. 198–204, 2016.

[4] N. Gyagenda, J. V. Hatilima, H. Roth, and V. Zhmud, "A review of GNSS-independent UAV navigation techniques," *Robotics and Autonomous Systems*, vol. 152, p. 104069, 2022.

[5] B. Joshi, H. Damron, S. Rahman, and I. Rekleitis, "Sm/vio: Robust underwater state estimation switching between model-based and visual inertial odometry," in *2023 IEEE International Conference on Robotics and Automation*, pp. 5192–5199, 2023.

[6] M. Wudenka, M. G. Müller, N. Demmel, A. Wedler, R. Triebel, D. Cremers, and W. Stürzl, "Towards robust monocular visual odometry for flying robots on planetary missions," *CoRR*, vol. abs/2109.05509, 2021.

[7] J. Quell, W. Stürzl, M. G. Müller, and R. Triebel, "Bringing visual (-inertial) odometries to the limits: In search of robust and accurate state estimation for micro aerial vehicles in challenging mars-analog environments," in *2025 IEEE International Conference on Robotics and Automation*, 2025.

[8] E. P. Herrera-Granda, J. C. Torres-Cantero, and D. H. Peluffo-Ordóñez, "Monocular visual SLAM, visual odometry, and structure from motion methods applied to 3d reconstruction: A comprehensive survey," *Heliyon*, vol. 10, no. 18, p. e37356, 2024.

[9] Z. Song, X. Zhang, T. Li, S. Zhang, Y. Wang, and J. Yuan, "Ir-vio: Illumination-robust visual-inertial odometry based on adaptive weighting algorithm with two-layer confidence maximization," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 4, pp. 1920–1929, 2023.

[10] X. Li, B. Zhang, J. Liao, and P. V. Sander, "Let's see clearly: Contaminant artifact removal for moving cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2011–2020, October 2021.

[11] H. M. S. Bruno, K. M. Cabral, E. L. Colombini, and S. N. Givigi, "A comparison of deep learning-based visual odometry algorithms in challenging scenarios," in *2024 IEEE International Systems Conference*, pp. 1–8, 2024.

[12] X. Li, J. Yu, F. Li, and G. An, "A high-precision self-supervised monocular visual odometry in foggy weather based on robust cycled generative adversarial networks and multi-task learning aided depth estimation," 2022.

[13] B. Liang, Y. Tao, H. Zhu, and Y. Song, "Robust odometry for wheeled robots in smoky environments: Leveraging smoke-adaptive image features and multisensor tight coupling," *IEEE Sensors Journal*, vol. 25, no. 12, pp. 22488–22500, 2025.

[14] N. Luo, Z. Hu, Y. Ding, J. Li, H. Zhao, G. Liu, and Q. Wang, "Dff-vio: A general dynamic feature fused monocular visual-inertial odometry," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 2, pp. 1758–1773, 2025.

[15] A. Albanese, Y. Wang, D. Brunelli, and D. Boyle, "Is that rain? understanding effects on visual odometry performance for autonomous AAVs and efficient dnn-based rain classification at the edge," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 20230–20238, 2025.

[16] A. Munir, A. J. Siddiqui, S. Anwar, A. El-Maleh, A. H. Khan, and A. Rehman, "Impact of adverse weather and image distortions on vision-based uav detection: A performance evaluation of deep learning models," *Drones*, vol. 8, no. 11, 2024.

[17] S. Zhao, Y. Gao, T. Wu, D. Singh, R. Jiang, H. Sun, M. Sarawata, Y. Qiu, W. Whittaker, I. Higgins, *et al.*, "SubT-MRS dataset: Pushing SLAM towards all-weather environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22647–22657, 2024.

[18] A. Albanese, "Adverse rainy conditions for autonomous UAVs," 2024.

[19] C. Shorten and T. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 1996.

[20] T. Rothmeier, M. D. Vesco Hoger, B. T. Nassu, W. Huber, and A. C. Knoll, "Out of the box: Weather augmentation for enhanced detection in bad visibility conditions," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems*, pp. 1410–1417, 2024.

[21] I. Nikolov, "Digiweather: Synthetic rain, snow and fog dataset augmentation," in *Extended Reality* (L. T. De Paolis, P. Arpaia, and M. Sacco, eds.), (Cham), pp. 22–41, Springer Nature Switzerland, 2024.

[22] C. Herbert, "Improving robustness of image recognition through artificial image augmentation," Master's thesis, The University of Waikato, 2023.

[23] L. Soni, A. Kaur, and A. Sharma, "A review on different versions and interfaces of blender software," in *2023 7th International Conference on Trends in Electronics and Informatics)*, pp. 882–887, 2023.

[24] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, 01 2016.

[25] A. K. Shastry, W. Cui, S. Abdi, S. S. Poojari, A. Ashry, Q. Wei, A. Luterman, V. Ved, and D. A. Paley, "Indoor aerial 3D mapping and target localization with a custom-built multispectral visual-inertial sensor system," in *Autonomy and Artificial Intelligence for Aerospace Vehicle GNC I*, 2025.

[26] M. Bagheri, M. Mohammadi, and M. Riazi, "A review of smoothed particle hydrodynamics," *Computational Particle Mechanics*, vol. 11, pp. 1163–1219, 2024.

[27] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[28] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation*, pp. 4666–4672, 2020.

[29] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *IEEE International Conference on Robotics and Automation*, pp. 4304–4311, IEEE, 2016.