Towards Robust Visual Odometry Systems Against Camera Lens Failures

(Article begins on next page)

Towards Robust Visual Odometry Systems Against Camera Lens Failures

Lorenzo Sarti*, Hudson Bruno†, Tommaso Puccetti*, Esther Colombini†, Andrea Ceccarelli*
*University of Florence
Email: lorenzo.sarti@unifi.it, tommaso.puccetti@unifi.it, andrea.ceccarelli@unifi.it
†University of Campinas
Email: hudson.bruno@ic.unicamp.br, esther@ic.unicamp.br

***Abstract*—A Visual Odometry system relies on one or more cameras to estimate the motion of an agent. These systems are mainly applied in the real world (e.g. UAVs, Autonomous Cars, etc.) because systems based on cameras are cheaper and easier to install and operate than other alternatives such as LiDARs, and more informative than IMUs. It is then evident that the camera is a critical component of these agents and malfunctions may lead to system failures, from out of trajectory to collisions. In this paper, we show that problems with the lenses, which are realistic in the operational environment of a camera-bearing agent, can alter the proper behavior of the system. Then, we propose a research roadmap to make the system robust to such failures.**

***Index Terms*—Visual Odometry, KITTI, Failure Injection;**

I. Introduction

A Visual Odometry (VO) system estimates the motion of an agent (e.g., automobile, UAVs, mobile phones) using only the input of single or multiple cameras attached to it [1]. Usually, this system is crucial to the agents' behavior, especially in autonomous systems. On such systems a camera malfunction may lead to the inability of the system to deliver a correct motion, possibly compromising the autonomous navigation task in unsafe ways. In this paper, we investigate the effects of common camera lens failures on a monocular feature-based VO system and propose possible mitigations.

II. System Description

For the experiments, we use a hybrid feature-based VO system. The system is classified as feature-based because it extracts and matches keypoints from the images and uses them to estimate the camera motion. We also call it a hybrid system because some traditional modules are replaced by Deep Neural Networks (DNNs). In this pipeline, we use DNNs for feature detection and matching, with traditional motion estimation from the keypoints matches. For these experiments we use the DNNs Superpoint [2] for feature detection and Superglue [3] for feature matching. The system receives a pair of images and predicts the camera's pose (position and orientation) after the changes in the features matched between the images.

We use a hybrid VO system because they can leverage the robustness of deep learning to enhance traditional VO systems [4]. Therefore, they are more robust against camera failures when compared to a traditional pipeline that uses classic feature detectors (ORB, SIFT, etc.) and feature matchers (Brute-Force, Nearest-Neighbours, etc.).

III. Failure Mode

RGB cameras are a widely known and well-established technology that has seen significant advancements over the years. As a result, finding a reliable RGB camera is relatively straightforward nowadays. However, despite these improvements, cameras can still encounter failures, particularly when subjected to external factors like harsh lighting conditions or adverse weather, such as heavy rain [5]. While some of these failures can be easily tested using common image augmentation libraries, there are other failure scenarios that are more challenging to simulate, making VO algorithms susceptible to such shortcomings.

To address this issue, the objective of our study is to test a VO system against specific camera lens failures and devise potential solutions or mitigations for these failures. To conduct our experiments, we use the KITTI dataset [6], a reference dataset consisting of diverse driving scenarios captured as sequences of images. To simulate camera lens failures, we introduce a patch into the target image sequence. Given that our target model compares images in pairs, we ensure that the same patch persists throughout the entire sequence.

To facilitate our experiment campaign and rigorously evaluate the performance of the VO system under normal and failure scenarios, we create an environment for injecting these failures, following the structure proposed in [7]. This setup allows us to compare the system's performance during nominal runs with those containing the injected failed images, providing valuable insights into the impact of lens failures on VO accuracy.

We select the following failures (visualized in Figure 1):

- *Broken/Scratched Lens*: a lens may break, for example, because of mechanical stresses due to vehicle jolts or the impact of gravel throw-up by the tires of nearby vehicles. The camera regularly outputs the image, but it will include an additional line (in case of a scratch) or more complicated patterns (Fig. 1b);
- *Condensation*: when the outside air temperature drops sharply, condensation may appear on the lenses. Condensation, or humidity, degrades the images (Fig. 1c). The image is acquired, but it may have defects due to halos on the lenses.

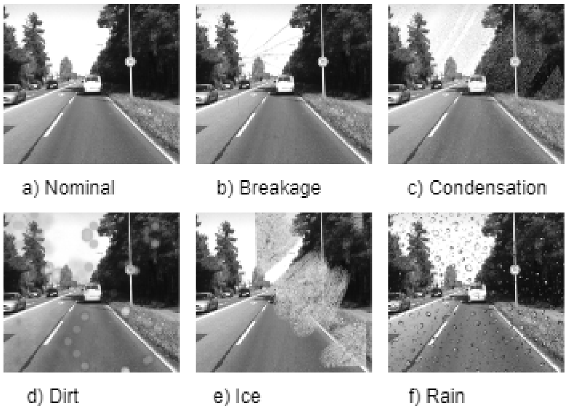

Fig. 1. Failure Types

- *Dirt*: this failure (Fig. 1d) concerns debris of various kinds and sizes (most typically, dust and dirt) which deposits on the internal or external lenses.
- *Ice*: ice can be the cause of several camera malfunctions. It can break the external materials of the camera lens and camera body. Furthermore, the external lens can be covered with a blanket of ice that prevents the acquisition of images (Fig. 1e);
- *Rain*: it refers to the case in which there are small spots on the images due to the deposit of water drops on the external lens (Fig. 1f).

IV. Experimental Campaign and Failure Effects

To perform the experimental campaign we need metrics to evaluate the model's predictions against the ground truth. We choose the following metrics [8]:

- *Absolute Trajectory Error* (**ATE**): is the average deviation from ground truth trajectory per frame.
- *Relative Pose Error* (**RPE**): measures the local accuracy of the trajectory over a fixed time interval. The RPE corresponds to the drift of the trajectory and is usually divided into translation and rotation components.

We perform the following experimental campaign. First, we execute the system without any faults and we collect the metrics. Then, we use our failure injection framework to perform a series of runs with the different types of faults. The framework works as follows:

- Generates a failure patch from an original pool of patches, making use of data augmentation to increase the diversity (mostly rotation and cropping);
- Selects an injection point;
- Applies the failure patch on all the original frames from the injection point to the end of the sequence;
- Runs the VO system on the injected sequence;
- Collects the run metrics;

We use the 11 sequences of KITTI which provide the ground truth trajectory. The sequences have different lengths (from 271 to 4661 frames, sampled at 10Hz) and provide different kinds of environments, like residential, urban, or highway, resulting in different levels of difficulty. We choose to inject the failures at approximately 30% of each sequence. For every fault type, we perform 15 runs. In total, we have 11 nominal runs and 825 runs on injected sequences.

From the results, we can do some preliminary observations: the rain and condensation create the most impact on the ATE, and this is expected since these faults are generating a lot of noise with respect to the original image. Ice and lens breakage are more localized faults and cause a significant accuracy loss only when the impacted area is bigger. The dirty lens failure has no significant detrimental impact on ATE, and also in some cases, it improves the overall trajectory accuracy. This only happens in some selected sequences that are particular either because they are very simple (just a vehicle moving forward on a straight line), or very complex (keypoints are distant from the vehicle itself).

V. Proposed Solution

To improve the robustness of the VO system, we are researching the following two strategies:

- *Data Augmentation*: the first solution we propose is to fine-tune the DNNs using data with injected failures. To avoid overfitting, this can be done using a different dataset and evaluated with failures that were not used in the training step.
- *Self-Supervised Learning*: the second option is to use a self-supervised strategy to train a DNN to learn to reconstruct images with injected failures. In this way, the model will learn weights that are robust to the noise applied. Then, we can use this same DNN to learn the VO task. We expect that the resulting model can accurately predict camera motion even in the presence of lens failures.

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry part i: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, 2011.

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2018, pp. 337–349.

[3] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4938–4947.

[4] H. M. S. Bruno and E. L. Colombini, "Lift-slam: A deep-learning feature-based monocular visual slam method," *Neurocomputing*, vol. 455, pp. 97–110, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221007803

[5] A. Ceccarelli and F. Secci, "Rgb cameras failures and their effects in autonomous driving applications," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, "Fault injection techniques and tools," *Computer*, vol. 30, no. 4, pp. 75–82, 1997.

[8] D. Prokhorov, D. Zhukov, O. Barinova, K. Anton, and A. Vorontsova, "Measuring robustness of visual slam," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6.