Design of a bi-monocular Visual Odometry System for Lava Tubes exploration

César Debeunne[1], Alex Torres[2] and Damien Vivet[1]

***Abstract*— With the discovery of potential extra terrestrial Lava Tubes, their exploration has become a hot topic for space agencies. Autonomous robots are needed to answer many questions that are raised by their discovery. We propose in this paper to discuss the design of a feasible and performing Visual Odometry (VO) solution to enable a safe navigation for spatial rovers in such environment. For this, we propose a modular bi-monocular indirect approach adapted to every camera model while keeping the processing load as low as possible. This paper presents an experimental study in a self illuminated cave environment. We focus on keypoint / descriptor pairs, feature association modalities and camera models. We show results on both real and simulated scenario to enhance the comprehension of methods widely used in the robotic community and their adaptation to such environment. Results from these experiments will be used to design a visual based navigation adapted for extra terrestrial Lava Tubes exploration.**

I. INTRODUCTION

Spatial exploration is a exciting and challenging application of robotics that demands an advanced level of autonomy while using low embedded power and processing capabilities. Planetary robots are mostly Unmaned Ground Vehicles (UGV) that rely on several sensors to execute autonomous tasks. To navigate in an unknown environment, a robot needs to perform state estimation and mapping in order to plan and follow trajectories. On the second successful rover mission on Mars in 2004, the Mars Exploration Rovers (MER) [1], a pair of stereo cameras is used for VO and point cloud generation to navigate safely in hazardous terrain. However, spatial context imposes computational limitations: the motion of the rover was only estimated with two stereo pairs thanks to a Maximum-Likelihood estimator. On Mars Science Laboratory (MSL) mission in 2012, a similar but faster and more robust algorithm was implemented [2]. The observations from the MER mission were determinant to highlight the main improvements necessary for a better solution [3]. Important wheel slippage was noticed during this mission. As a result, odometers measurements are not incorporated into the navigation solution.

Recently, satellite images from Lunar and Martian volcanic areas show features that may be interpreted as lava tubes [4]. These hypothetical caves seem similar to their terrestrial analog, but subsurface exploration is needed to determine their precise morphology. Moreover, as a consequence of the small gravity, they could be large enough to host artificial bases shielded from radiation. Thus, robotic exploration in a lava tube context may be the goal of a future extra terrestrial mission and an appropriate navigation software needs to be developed. Such missions require to push further the navigation capabilities of the rovers while keeping in mind the intrinsic limitations and constraints of spatial technologies.

Cave mapping has gained a lot of interest during the Subterranean Challenge that saw many solutions reaching outstanding performances [5]. Most of them use rotating LiDARs and redundancy in the system to reach high levels of robustness. However, rotating devices like LiDARs are not ready to be send in space yet because of their power consumption, bigger mass and reliability concerns regarding their fast spinning motors. Moreover, redundancy in the system requires too much power and processing cost for spatial applications. The only mature LiDAR solution for space is the solid state LiDAR but, such device provides a very narrow Field of View (FoV) and is more dedicated to traversability analysis. In the context of cave mapping an important FoV is required in order to optimize both the mapping and the localisation capabilities of the rover. It has been shown that a Stereo Fisheye setup, which has a wider FoV, may offer larger navigability maps for rover navigation [6].

Consequently, visual navigation is the solution that needs to be studied in this context. On Earth, the robotic community has reached a very good knowledge of the Visual SLAM problem. Versatile and robust systems have emerged in the past few years in both direct [7] and indirect [8] fashions. However, these systems have also reached a great degree of complexity. Many design choices in the SLAM pipeline deserve to be discussed to improve our knowledge of these powerful tools and for a potential use of VSLAM in space application.

A major issue in the use of visual sensor in such environments is that it requires artificial light to operate. Thus, we propose to investigate a design solution for a bi-monocular VO system for lava tube navigation with onboard illumination. This paper sheds light on three design choices and is the support of a preliminary work toward a pertinent visual navigation solution for extra terrestrial lava tube exploration:

- Keypoint and Descriptor choice for feature extraction and association,
- Method for associating features frame to frame,
- Comparison between standard and wide angle camera setup.

*This work was supported by the CNES and Occitanie Region

[1]César Debeunne and Damien Vivet are with ISAE-SUPAERO, University of Toulouse, France firstname.lastname@isae-supaero.fr

[2]Alex Torres is with the CNES, France alex.torres@cnes.fr

II. Notation

In this paper, we note the pose of a camera at timestamp i in the world frame as ${}^{w}\mathbf{T}_{c_i} \in SE(3)$. We parameterize landmark j with its 3D position in the world frame $\mathbf{l}_j^w \in \mathbb{R}^3$. We can compute the coordinates of j-th landmark in the i-th camera frame with $\mathbf{l}_j^{c_i} = {}^{w}\mathbf{T}_{c_i}^{-1}\mathbf{l}_j^w = {}^{c_i}\mathbf{T}_w\mathbf{l}_j^w$. We denote the projection function of a camera $\pi : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$ that maps a 3D point in the camera frame in the 2D image $x_{i,j} = \pi({}^{c_i}\mathbf{T}_w\mathbf{l}_j^w)$. We note abusively $\pi^{-1} : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ the function that computes the bearing vector of a given pixel $\mathbf{u}_{i,j} = \pi^{-1}(x_{i,j})$. We work with a set of keyframes in a sliding window W and each keyframe K_i has a set of map point L_i. We concatenate all landmarks and keyframe poses in a state vector X.

III. System Overview

In order to evaluate the performances of a bi-monocular VO design in an underground cave exploration context, we first need to get a simple modular baseline odometry framework. This section aims to present the proposed baseline with some specificities (Figure 2).

A. Feature detection

1) Image pre-processing: A first mandatory step while navigating in dark environment with artificial illumination is to enhance the contrast of the image. Indeed, due to the lighting conditions, the appearance of the environment is changing as the robot moves. The image pair taken by the robot at each step is first pre-processed with CLAHE [9].

2) Feature extraction and association: Then the features have to be detected and associated between the different acquired images. As in classical bi-camera approaches there is two association steps: between cameras from the same frame and between successive frames on one of the two cameras. We compared and evaluated two techniques to proceed such association: tracking and matching (see Section IV-B). Finally, the map is updated with the matched features and a landmark recovery phase is performed to recover lost features due to illumination change or occlusion.

B. Estimation step and match rejection

From the matches between the currently detected features and the map landmarks, a first pose estimation is produced using P3P algorithm [10] in a RANSAC fashion to remove false feature-to-landmark associations.

A second outlier rejection phase is performed by checking the epipolar constraint due to the newly estimated camera motion. Instead of checking the classical distance to the epipolar line (or curve in the case of fisheye camera), we propose to check the angle to the epipolar plane (Figure 1). This rejection method makes use of bearing vectors and can be generalized to any camera model with less computational time. The residual δ_e can be computed as follow:

$$\delta_e = \frac{\pi}{2} - \arccos\left(\left(\hat{\mathbf{t}} \times \mathbf{u_1}\right).\mathbf{u_2}\right), \tag{1}$$

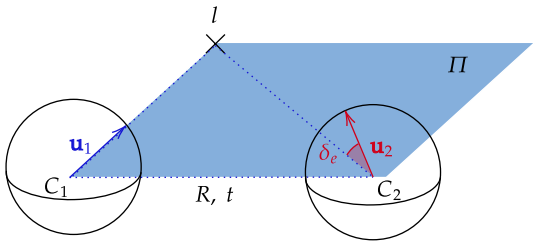

Fig. 1: Illustration of the epipolar angular error. $\prod$ is the epipolar plane, $\{C_1, C_2\}$ are the optical centers of the two views and $\{\mathbf{u_1}, \mathbf{u_2}\}$ are the two bearing vectors of landmark l observation.

with $\hat{\mathbf{t}} = \frac{\mathbf{t}}{\|\mathbf{t}\|}$ being the direction of the translation vector between the two views and $\mathbf{u_1}, \mathbf{u_2}$ the bearing vectors of the same observation in both views.

If δ_e is superior to a threshold, the new matched feature can be classified as an outlier. Finally, a pose refinement is performed with a classical single frame bundle adjustement.

C. Keyframe selection

As we have limited power and memory on embedded devices, the number of frames to keep in memory has to be reduced while retaining the maximum information. We use the parallax information as a keyframe indicator as we initially proposed in [11]. Parallax can be seen as an increase of information over each landmark: a bigger parallax value enables a more precise landmark triangulation. If the average parallax increase of all the landmarks in the map since the last keyframe goes over a certain threshold, the new frame has to be kept in memory. This metric does not take into account landmark loss due to illmination change for instance, so we also select a keyframe if the number of tracked landmark since the last keyframe goes below a certain threshold.

D. Landmark initialization

In order to reduce computation, we choose to initialize landmarks only when a keyframe is voted. At this time, all the landmarks linked with a feature that was associated with the previous keyframe are initialized. We prioritized features that were associated on both left and right cameras, then features that were associated on left cameras only and finally, if the number of features is still too low, landmark are initialized from left / right associations of the current frame.

In most cases, landmarks are initialized from more than two views. Thus, we implemented a generalization of the mid-point algorithm to n-points for triangulation of landmarks. However, the 3D triangulation doesn't return the optimal landmark position in term of reprojection error as the residual is in the 3D space and not in the 2D observation space. This rough initialization is then followed by the minimization of the reprojection error

$$L_i^* = \underset{L_i}{\arg\min} \sum_{\mathbf{l}_j^w \in L_i} ||x_{i,j} - \pi({}^{c_i}\mathbf{T}_w\mathbf{l}_j^w)||^{\rho}. \tag{2}$$

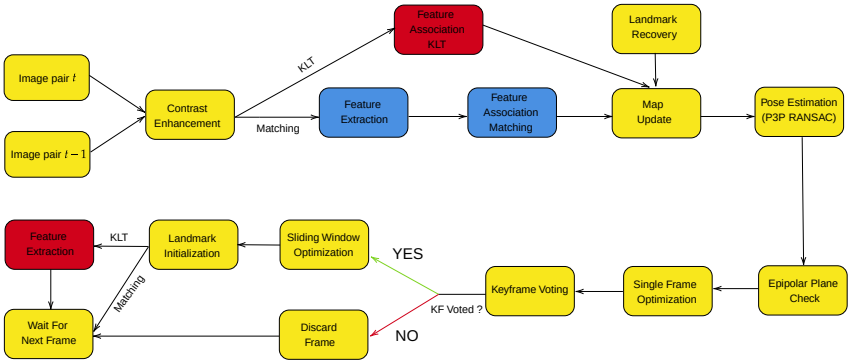

Fig. 2: Pipeline of both matching based and tracking based Visual Odometry. Steps relative to tracking design are in red, whereas boxes referring to matching design are in blue.

To mitigate the influence of outliers, we apply the robust Huber norm ρ to our reprojection error in the first iterations. Based on the mean reprojection error, an outlier rejection is finally performed to filter out wrong landmarks.

E. Local Map optimisation

Finally, a Bundle Adjustement (BA) step is performed on a sliding window of keyframes by minimizing the reprojection error of every landmark $\mathbf{l}_j^w$ observed in our window W

$$X^* = \arg\min_X \left(\sum_{K_i \in W} \sum_{\mathbf{l}_j^w \in L_i} ||x_{i,j} - \pi({}^{c_i}\mathbf{T}_w \mathbf{l}_j^w)||^2 \right). \quad (3)$$

We solve this problem with the Levenberg-Marquardt algorithm from the library Ceres. To reduce the processing time and because of the multiple outlier rejection mechanism, the Euclidean norm is now used in the global BA step.

IV. Design choice and validations

From the previously introduced modular VO framework, we investigate the impact of different designs on both the performances and processing time. We focus on three main points :

- the feature choices,
- the association step,
- the camera model.

The impacts of such design choices have been analysed on both simulated and real data. Real data were provided by the OIVIO dataset [12], that provides video sequences in dark scenes with self illumination. Simulated data were made on Gazebo, in a cave world inspired by the Subterranean challenge [13]. The software has been developed in C++, and all experiments has been performed by a desktop station equipped with an Intel Core i7, 3.2 GHz clock rate. Even for SuperPoint extraction, we have not used a GPU.

A. Keypoints and Descriptors

1) Comparison of keypoints in artificial illumination conditions: In any indirect SLAM methods, sparse representation of the observations are composed of keypoints. Such features must be stable over time, robust to changes in point-of-view and illumination conditions so they can be matched over frame. Moreover, we aim to have low computational cost for both keypoints extraction and descriptor calculation. We propose to follow *Ferrera et al* [14] work on submarine navigation, by implementing additional solutions and adapting their approach to subterranean images using onboard illumination.

We select and compare the most used features from the robotic community namely: SIFT [15], KAZE [16], ORB [17], BRISK [18], FAST+ORB and SuperPoint [19]. SuperPoint is a learned detector based on a Convolutionnal Neural Network architecture. We used the pre-trained model from the original publication. KAZE, SIFT and SuperPoint are obviously too computationally expensive for our application, nevertheless we think that studying their robustness in challenging lighting conditions is of interest for the community.

2) Experimental comparison: To study the performance of each detector / descriptor pair, we have run our VO on a trajectory of the OIVIO dataset (MN 050 GV1). The configuration was the same for every experiment: 500 keypoints were extracted in each frame, with a bucketing strategy for a better spatial dispersion of keypoints. The feature association is done by matching.

We have computed several indicators:

- The full extraction process time, including the non-maximum-suppression and the bucketing.
- The percentage of outlier per matching: it describes the ability to match reliable features. It has to be compared to the total number of match to have an idea of the global quality of the feature association. The outlier detection was done via essential matrix computation in a RANSAC scheme.
- The average number of matches per frame pair before filtering.
- The track length *i.e.* the number of matched features per landmarks.

The results are presented in table I. KAZE and SIFT offer obviously superior performances in terms of confident point selections but KAZE is disappointing on track length. Its robustness to light changes is limited. BRISK and ORB have similar performances in frame to frame matching, but BRISK has less repeatability. ORB descriptor is quite robust

TABLE I: Performance comparison between the selected detector / descriptor pair.

	dt (ms)	% outliers	matches / frame	track length
SIFT	71	1.0	180	9
KAZE	66	1.5	220	6
ORB	15	6.9	175	6.8
FAST/ORB	5	3.4	95	10
BRISK	25	5.9	200	4.5
SuperPoint	350	8	280	9.0

to lighting changes and the combination FAST + ORB is the most effective for associating features on the long run: less points are matched, but they are matched longer. BRISK, ORB and KAZE produce binary descriptors that are matched efficiently with the Hamming distance, SIFT produces patch descriptors that are matched with L2 norm that is more computationally expensive. SuperPoint seems also effective in our context as it can match a lot of points and track them for a great number of keyframes. However, such a solution is not suitable for real time applications on a power limited hardware.

B. Feature Association Strategies

1) Matching vs Tracking: In order to get multiple views of a given keypoint, two main techniques have emerged in the literature: descriptor matching or feature tracking.

Descriptor matching is done by computing the distance between the descriptors of keypoints. In our system, when we match points in two consecutive frames, we compare a keypoint from the first frame with all the keypoints in a rectangle centered on the same position in the second frame. The size of this rectangle is a hyperparameter that can be tricky to tune as it depends on the frame rate and the dynamics of the robot. A trade off between computation load and the level of motion allowed between frames has to be found.

Tracking is performed using Kanade-Lucas-Tomasi (KLT) algorithm. It is based on optical flow computation on different scale pyramids of the images [20]. To avoid wrong tracks, we implemented it in a "forward backward" fashion. The tracking is performed from the first frame to the second one and the tracked features are tracked back from the second frame to the first one. If the initial point is not recovered, the track is considered wrong. This method is based on the optical flow equation that is valid under the constant intensity hypothesis. Only small motions, in a short timeline can produce valid tracks. A good point is that features do not need to be detected on each frames while this is needed for matching. In our pipeline, frames are repopulated with features only for keyframes. The difference between our VO matching based and our VO tracking based can be observed on figure 2.

2) Association performances: We compared the performance of both feature association strategies on the OIVIO dataset which provides six sequences in a mine on a UGV with groundtruth. Three levels of onboard light intensity (1500, 5000 and 10000 lumens) are tested on two different trajectories. The Absolute Trajectory Error (ATE), Relative Pose Error (RTE) and the average duration of an iteration are the metrics evaluated. The couple FAST+ORB was chosen for all these experiments and the results are presented on table II. For these experiments, 1000 keypoints were detected on each frame with Matching while 250 keypoints were detected on each Keyframes with KLT.

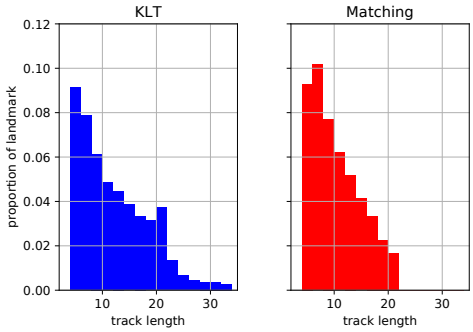

Fig. 3: Histogram that represents the lengths of tracks *i.e.* the number of keyframes on which a landmark is detected for a matching based VO and a tracking based VO.

In most cases, Matching based VO seems to be more accurate than KLT based method, especially on the RTE metric. When keypoints are tracked, as the point is not redetected at each frame, there can be a small drift over time which can explain the loss of accuracy. But for every sequence, KLT design is faster. The necessity to detect keypoints on each frame in the Matching case increases the average running time. Also, we noticed that more Keyframes were voted due to a lack of features tracked with the Matching design and Keyframes require more computations than frames. KLT enables to track features on more frames (Figure 3) so that Keyframes are more spaced in time.

In both systems, there seems to be no influence of the light intensity on the global performances. The image processing step permits to handle well low enlighten scenes. This is a good point as it shows that heavy power lights are not mandatory to perform VO in such conditions. The performances of ORBSLAM2 [8] on these sequences are also provided on table II. Its accuracy is better but it comes at a higher computational cost, mainly due to its restrictive landmark selection and complex map management.

C. Classical vs Fisheye optics

1) Camera model choice: Capturing an image flow is a fundamental piece of a VO system. Choosing the device with the most suitable Fov depends mainly on the application. For earth based applications like autonomous vehicles, high frame rate and classical optics are the standard as the FoV is mainly focused on the front of the vehicle and the positioning relies mainly on Global navigation Satellite System (GNSS). Intuitively, a wide FoV may help to maintain features in sight for a longer time to make the navigation more robust. This is why, when going in GNSS denied environment, the FoV is increased using multiple sensors or rotating LiDAR to improve the localisation and mapping capabilities. In the case

TABLE II: Performances on OIVIO Dataset

Scenario	Distance(m)	Duration(s)	KLT ATE(m)	KLT RTE(m)	KLT dt(ms)	Matching ATE(m)	Matching RTE(m)	Matching dt(ms)	ORBSLAM2 ATE(m)	ORBSLAM2 RTE(m)	ORBSLAM2 dt(ms)
MN 015 GV1			0.19	0.07	33	0.38	0.04	35	0.11	0.04	43
MN 050 GV1	80	218	0.26	0.08	25	0.43	0.05	39	0.12	0.04	47
MN 100 GV1			0.35	0.09	24	0.25	0.06	39	0.14	0.05	46
MN 015 GV2			0.13	0.04	25	0.14	0.04	38	0.11	0.03	40
MN 050 GV2	82	150	0.15	0.04	30	0.10	0.03	30	0.10	0.03	47
MN 100 GV2			0.14	0.04	26	0.10	0.04	31	0.10	0.03	40

TABLE III: Mapping functions for the normalized spherical camera model

	Mapping function	Note
Equidistant	$g(\theta) = \theta$	Maintains angular distance.
Equisolid Angle	$g(\theta) = 2\sin\left(\frac{\theta}{2}\right)$	Maintains area relations.
Orthographic	$g(\theta) = \sin(\theta)$	Maintains planar illuminance.
Stereographic	$g(\theta) = 2\tan\left(\frac{\theta}{2}\right)$	Maintains angles.
Rectilinear	$g(\theta) = \tan(\theta)$	Equivalent to the pinhole model.

TABLE IV: Experimental comparison between a VO system with a standard camera and a VO system with a fisheye camera.

	dt(ms)	n matches	track length	ATE(m)	RTE(m)
Fisheye	48	235	10.1	0.18	0.04
Classical	55	220	9.1	1.4	0.05

of spatial applications, none of these solutions is possible. The increase of the FoV is directly linked to the number of sensors, the optics of the camera or the use of a rotating head on a mast.

In this section, we aim at investigating the improvement of using fisheye over standard camera. A classical lens can be modeled by the well known "pinhole" model. Given a 3D point in the camera frame $\mathbf{x} \in \mathbb{R}^3$, the projection function of a pinhole camera is defined with four parameters $\{f_x, f_y, c_x, c_y\}$ as

$$\pi(\mathbf{x}) = \begin{bmatrix} f_x \frac{x}{z} \\ f_y \frac{y}{z} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}. \tag{4}$$

However this function cannot represent wide FoV cameras, as it goes to infinity for points with an angle of view close to 180°. To address this problem, we propose to model fisheye lens with the normalized spherical model

$$\begin{aligned} \pi(\mathbf{x}) &= \begin{bmatrix} f_x \frac{g(\theta)x}{r} \\ f_y \frac{g(\theta)y}{r} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \\ r &= \sqrt{x^2 + y^2}, \\ \theta &= \arctan 2(r, z), \end{aligned} \tag{5}$$

with $g(\theta)$ a function that gives the distance of the projected point from the optical center given the angle of view. This mapping function depends on the type of lens used, a summary of the different mapping functions is given on Table III. We provide bellow the comparison between these two models considering both processing time and accuracy.

2) Influence of the Camera model: On our simulated dataset, both fisheye (with an equidistant mapping function) and classical camera pairs were simultaneously mounted on the robot, at the same position and with the same

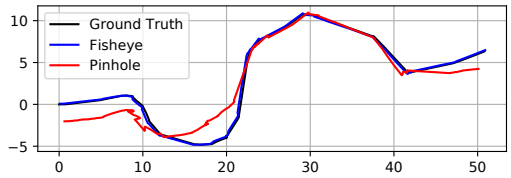

Fig. 4: Trajectories returned by our VO in a stereo fisheye setup and a standard stereo camera setup on the simulated dataset.

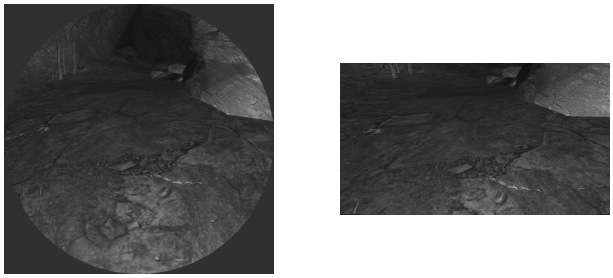

Fig. 5: On the left, a Fisheye image from the simulated sequence and on the right a standard image from the same frame.

orientation. The respective camera resolution are 848×848 and 1280×720 for the fisheye and pinhole cameras. For this experiment, the tracking design with FAST keypoints was implemented as it was retained as a good compromise from the two previous sections. The sequence lasts 200 seconds and is 70m long. Onboard illumination is simulated and the images were provided at 10 Hz. This cave environment is poorly textured and offers challenging conditions for VO (Figure 5). According to table IV, the fisheye setup outperforms the standard setup in term of accuracy, number of feature tracked and length of tracks. On figure 4, we note that with the classical camera setup, drift can be accumulated at some points of the trajectory. It is especially the case at moments when the robot is facing poorly textured surfaces, many landmark are then lost and the pose estimation is corrupted. In such cases, the wide angle camera enables to keep more landmarks in sight to keep a consistent estimate.

V. Discussion

Due to space rover constraints, current state of the art visual navigation solutions are not directly applicable as they are too complex and computational power consuming. This preliminary work discussed some specific points for a simple VO design and dig into existing works to get some insight for the development of a VO dedicated to lava tube exploration. However, some points are left for future research:

- Considering illumination changes, we applied CLAHE as an image enhancement technique, but other methods exist. For instance, a simple histogram equalization or an adaptative gamma correction should be tested [21].
- In this first design, we simply discard measurements when a frame is removed from the sliding window. This procedure leads to an information loss that can be compensated with marginalization [22]. Such an operation enables to have a prior on our problem for a better accuracy. But this comes at a certain computational cost that may be too demanding for spatial application. A solution for this can be to explore sparsification methods [23] to make it suitable for our requirements.
- Moreover, as the proposed solution is bi-monocular, the method is highly dependent on the assumption of a good rigidity of the transform between the cameras. Under the constraints of launch, cruise and planetary operation, the extrinsic cameras calibration may change. Works to include the calibration as a parameter or using constraint in the estimation [24] have to be investigated to improve the global robustness of the system.
- Our conclusions on the choice of the camera lens will be useful to build an experimental setup for real data acquisition. With real fisheye images, the question of which camera model we should choose will be raised. The presented model is not the only one available [25], an investigation for an accurate and low computational model will be needed.
- Finally, the proposed analysis of features, descriptors and association strategies from this paper will be extended for real fisheye images to validate our conclusions.

VI. Conclusion

This paper presents a modular VO system, able to handle both pinhole and fisheye cameras, and an experimental study of some of its essential components. We proposed a simple design for extra terrestrial lava tube exploration and justified our choices with comparison experiments on both real and simulated data with onboard light and in unstructured environments. The proposed fisheye based VO solution shows good performances in simulated data and limits drift in comparison with a standard setup. Our future work will focus on image processing, VO back-end and camera modelling to reduce the estimation step complexity and the global power consumption.

References

[1] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings, IEEE Aerospace Conference*, vol. 5, 2002, pp. 5–2025.

[2] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies, "Robust and efficient stereo feature tracking for visual odometry," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 39–46.

[3] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, pp. 169–186, 03 2007.

[4] F. Sauro, R. Pozzobon, M. Massironi, P. De Berardinis, T. Santagata, and J. De Waele, "Lava tubes on earth, moon and mars: A review on their size and morphology revealed by comparative planetology," *Earth-Science Reviews*, vol. 209, p. 103288, 2020.

[5] A. Reinke, M. Palieri, B. Morrell, Y. Chang, K. Ebadi, L. Carlone, and A.-A. Agha-Mohammadi, "LOCUS 2.0: Robust and computationally efficient lidar odometry for real-time 3d mapping," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.

[6] A. Torres, E. Remetean, S. Moreno, T. Germa, J.-B. Ginestet, and F. Souvannavong, "Omnidirectional stereoscopic vision systems for planetary exploration rovers," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2016.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, 2017, pp. 611–625.

[8] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[9] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," in *International Conference on Advances in Computing, Communications and Informatics*, 2014, pp. 2392–2397.

[10] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2969–2976.

[11] D. Vivet, A. Debord, and G. Pages, "Pavo: a parallax based bi-monocular vo approach for autonomous navigation in various environments," in *The International Conference on Digital Image & Signal Processing*, 2019, pp. 1–7.

[12] M. Kasper, S. McGuire, and C. Heckman, "A Benchmark for Visual-Inertial Odometry Systems Employing Onboard Illumination," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[13] A. Koval, C. Kanellakis, E. Vidmark, J. Haluska, and G. Nikolakopoulos, "A subterranean virtual cave world for gazebo based on the darpa subt challenge," *ArXiv*, vol. abs/2004.08452, 2020.

[14] M. Ferrera, J. Moras, P. Trouvé-Peloux, and V. Creuze, "Real-time monocular visual odometry for turbid and dynamic underwater environments," *Sensors*, vol. 19, no. 3, 2019.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[16] P. Fernández Alcantarilla, A. Bartoli, and A. Davison, "Kaze features," in *European Conference on Computer Vision*, 2012, pp. 214–227.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[18] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555.

[19] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 337–33 712.

[20] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel corporation*, vol. 5, no. 1-10, p. 4, 2001.

[21] M. Veluchamy and B. Subramani, "Image contrast and color enhancement using adaptive gamma correction and histogram equalization," *Optik*, vol. 183, pp. 329–337, 2019.

[22] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[23] J. Hsiung, M. Hsiao, E. Westman, R. Valencia, and M. Kaess, "Information sparsification in visual-inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 1146–1153.

[24] D. Vivet, J. Vilà-Valls, G. Pages, and E. Chaumette, "Robust filter-based visual navigation solution with miscalibrated bi-monocular or stereo cameras," *Remote Sensing*, vol. 14, no. 6, p. 1470, 2022.

[25] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *International Conference on 3D Vision*, 2018.