Evaluating Visual Odometry & SLAM Initialization Methods for Arbitrary Multi-Camera Rigs

Mikhail Terekhov[1] mterekhov@student.ethz.ch
Mike Zhang[1] shaozhang@student.ethz.ch
Jiaqi Chen[1] jiaqchen@student.ethz.ch
Shane Kelly[1] skelly@student.ethz.ch

Abstract

Multi-camera visual SLAM systems are becoming increasingly relevant and generalized relative pose solvers have been developed to initialize them. However, in practice, heuristics-based methods are almost always used. This work investigates this discrepancy by evaluating several initialization methods. Namely, ORB-SLAM3 (stereo SLAM system), MultiCol (multi-camera adaptation of ORB-SLAM), and generalized camera solvers. Our main contribution in this paper is a benchmarking framework that is used to evaluate the multi-camera initialization schemes. We compare these methods in different environmental conditions and parameters. We extract several insights including common failure modes, and identify the current best methods for multi-camera initialization. Lastly, we present possible future work that would be needed to bring these theoretical methods to practical applications.

1. Introduction

Visual odometry (VO) and simultaneous localization and mapping (SLAM) have become increasingly used in industrial applications such as mobile robots. The demand for improved performance and robustness has necessitated the use of multiple cameras in these systems. Several recent works in multi-camera SLAM systems have been developed [20, 3, 10]. All VO/SLAM approaches need a separate procedure to initialize the map and initial camera pose. We argue that current systems could benefit from a generic multi-camera initialization scheme without any constraints on the camera system. Existing VO/SLAM approaches are either restricted to including a stereo pair [10] or use heuristic approaches that often lead to poor initializations [20].

There are theoretical developments that are supposed to solve this issue, but they are not employed in real systems. Thus, before creating a generic multi-camera initialization pipeline, we need to evaluate the available theoretical and practical approaches. This would reveal insights that allow us to address the problems in existing initialization methods.

[1]ETH Zürich, Rämistrasse 101, 8092 Zurich

2. Related Work

For monocular SLAM, there is a principled approach to system initialization: run RANSAC with a minimal solver, such as [14] and use the epipolar geometry constraint to initialize a map of 3D points for SLAM tracking. This approach can be naturally extended to a multi-camera scenario with a *generalized solver*. Such a solver estimates the relative pose of a *generalized camera* [15], a formulation where the unprojected camera rays do not share a common starting point. As with the monocular case, a generalized epipolar constraint [15] applies to multi-camera rigs treated as a single generalized camera. Based on this idea, many generalized solvers have been recently developed. In this work, we consider the Linear 17-Point solver [9], Generalized Eigensolver (GE) [8] and a minimal 6-Point solver [19]. To leverage these algorithms, we use the implementations from the OpenGV library [7].

However, these solvers are rarely used in practice. Sometimes a heuristic method based on monocular initialization is implemented [20], and sometimes visual odometry is limited to multi-camera systems that contain a stereo-pair [10]. Stereo SLAM, such as ORB-SLAM3 [2] is known to achieve much better performance than its monocular counterpart because of the availability of depths for each frame, but it cannot be used for any arbitrary multi-camera setup. A state-of-the-art system SVO [3] does employ a principled approach using the 17-point algorithm, but we were not able to make it run in practice. This work investigates why this gap between theory and practice exists.

3. Contributions

Our contributions are the following:

- Evaluated initialization of two real SLAM systems (ORB-SLAM3 and MultiCol SLAM) on two datasets
- Evaluated initialization using generalized camera solvers on three datasets
- Investigated the failure modes of each method.
- Attempted a fair comparison between all methods.
- Proposed improvement to the existing methods given our experiments.

4. Methodology

4.1. Datasets

EuRoC. For isolated evaluation of stereo initialization, we use the EuRoC [1] datasets, which provide synchronized stereo image pairs from an MAV under a wide range of aggressive maneuvers. Specifically, we focus on the V103 dataset, which is one of four EuRoC datasets labeled with the highest difficulty rating (large amounts of motion blur and extreme lighting changes), to allow for more investigation of typical stereo initialization failure modes.

Oxford RobotCar. The Robotcar datasets include a forward stereo pair, three surround view cameras [12], and have recently been equipped with RTK globally optimized ground truth poses [**?**]. We evaluate all initialization methods using Oxford RobotCar, namely three different segments to account for different environments: `2015-10-30-11-56-36` overcast conditions, `2014-12-16-18-44-24` nighttime, and `2015-02-17-14-42-12` direct sunlight.

Extension of Multi-FoV. We also use an extension of the synthetic Multi-FoV dataset [21]. Originally, the dataset was created for one wide-angle camera, but the authors generously provided the Blender scene as well as the patch for enabling wide-angle cameras. This allowed us to extend the set up to four cameras to achieve a nearly 360 degree FOV. We call the resulting dataset *MultiCam*. It was used to evaluate all methods except ORB-SLAM3.

Autovision Samples. The last dataset that we considered was samples of data from the AutoVision project [4]. We received five short driving sequences, each with images from five cameras. We were given the intrinsic calibrations parameters for the Unified Camera Model [13]. This dataset was used to evaluate the generalized solver methods.

4.2. Heuristic-Based Multi-Camera Initialization

MultiCol SLAM [20] is a generic multi-camera SLAM pipeline that uses a heuristic method for initialization. It first tries to estimate the motion of each camera independently via RANSAC and the 5-point minimal solver [18]. The camera with the most inliers is then selected. Afterwards, the points from the selected camera are reprojected onto the other cameras to get a scale estimate and observations from other cameras. To evaluate the quality of this approach, we run the whole system on a chunk of 50 frames until it initializes. We let it process 20 additional frames (still within the chunk) to see if it can recover from a bad initialization and continue tracking.

4.3. Stereo Initialization

Unlike arbitrary multi-camera rigs, stereo camera rigs are guaranteed to have significant FOV overlap that allows for direct scale initialization of map points via triangulation without any camera motion, often on the first observation. However, the relatively limited FOVs of some stereo cameras, compared to multi-camera rigs that can have nearly 360 degree FOVs, limits the observability of available visual texture and may lead to delayed or poor initializations.

We include ORB-SLAM3 [2], a state-of-the-art visual SLAM system, in our evaluation to serve as a reliable performance baseline and to gain insights from scenarios where stereo initialization succeeds while multi-camera initialization methods fail, and vice-versa.

4.4. Generalized Camera Initialization

To the best of our knowledge, the only state-of-the-art method that uses a generalized solver for multi-camera initialization is SVO [3]. However, we were not able to run SVO on multi-camera data as we only had access to compiled binaries which had issues in our environments.

We implement our own pipeline for evaluating generalized camera initialization using the solvers from OpenGV [7]. It only operates on pairs of frame bundles at two different timestamps. Correspondences are found using feature detectors (SIFT [11] and ORB [16]) from the OpenCV library [5]. The matches are refined using the cross match and ratio tests [11]. Matches were only found in individual cameras. We did not implement finding cross camera matches due to time constraints. The solver uses the correspondences to estimate the relative pose between the frame bundles which is then used to triangulate the 3D map.

OpenGV solvers require the correspondences as bearings pointing into the scene, which we obtain by back-projecting the matched keypoints. We implemented the Scaramuzza Camera Model [17] which accurately approximates projection for wide-angle cameras. For AutoVision, we use the Unified Camera Model [13] of the provided calibration since conversion to the Scaramuzza Model is non-trivial.

The generalized solvers from OpenGV are used inside of RANSAC as minimal solvers. We used the `MultiAdapter` data handling from OpenGV to *evenly* sample correspondences from each camera, which was critical. We used RANSAC with adaptive termination which was ran three times for an averaged estimate.

Our pipeline was evaluated on the MultiCam, Oxford RobotCar, and AutoVision dataset. We did not evaluate on EuRoC as the camera rig only has two cameras, which falls into a degenerate case for some of the generalized solvers.

5. Results

5.1. Stereo Initialization

The ORB-SLAM3 stereo initialization was evaluated on the difficult EuRoC V103 dataset, split into 20-frame chunks. We processed each chunk 20 times to capture a distribution of initialization results. Despite the difficulty, most estimates align well with the ground-truth.

Initialization is defined as successful if the maximum rotation error and the maximum translation error are both within thresholds. We set the success thresholds to 10 degrees of rotation error and 0.25 meters of translation error. Figure 1 shows that the ORB-SLAM3 initialization is quite robust, achieving a 100% success rate for every chunk ex-

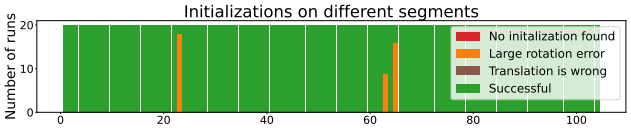

Figure 1: ORB-SLAM3 initialization results on the EuRoC V103 dataset split into 20 frame chunks. Each chunk was processed 20 times and distributions of initialization success is shown.

cept for three.

The failure modes of the ORB-SLAM3 stereo initialization were determined by inspecting the system during each of the three chunks with failures. In all three chunks there is a combination of fast motion and sudden change in image exposure. Figure 2 shows a visualization of the front-end of the system during the chunk with the most failed initializations. The under-exposed images led to sparse texture and few tracked features.

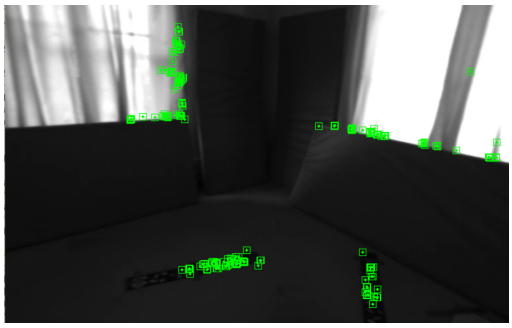

Figure 2: A visualization of the front-end of the ORB-SLAM3 system during initialization in a common failure mode which all failed chunks in this dataset have in common: a combination of fast motion and extreme lighting conditions.

5.2. Heuristics-Based Multi-Camera Initialization

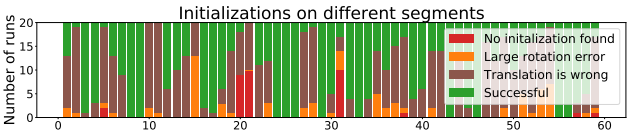

Figure 3: MultiCol initialization results on the modified Multi-FoV dataset split into 60 chunks. Each chunk was processed 20 times and the distributions of initialization success are shown.

We have evaluated MultiCol SLAM initialization on two datasets: modified Multi-FoV and Oxford RobotCar. Most of the results are provided in the general comparison later. To analyze the performance in more detail, we have created plots like Figure 3. These plots reveal interesting failure modes for the heuristics-based initialization approach, like the one around segment $\#20$ on the plot.

Main reasons for failures of the heuristic initialization were found to be:

1. The chunk has no motion in it.
2. Surrounding region has poor/repetitive visual texture.
3. Only a few points are successfully reprojected from the leading camera onto the rest.

While the first two reasons can be considered more or less fundamental, the third could be avoided by a principled initialization approach.

5.3. Generalized Camera Initialization

Our generalized camera solver pipeline was evaluated by running on frame pairs. Note that we do not consider frame pairs that are static, as they cannot triangulate the 3D map and would therefore fail to initialize for tracking regardless of the accuracy of the relative pose estimate.

We evaluate the initialization on the estimated relative pose as ground truth 3D maps are sparse or unavailable. We define the following metrics: The translation direction error (TDE) is defined as the cosine distance between the estimate and actual translations. The rotation error (RE) is the angular magnitude of the error rotation matrix $\tilde{R} = R_{GT}^T \hat{R}$. The normalized translation error is defined as $||t_{GT} - \hat{t}||/||t_{GT}||$. When it is low, both the direction and scale are well estimated. It is large when the scale is poorly estimate regardless of the accuracy of the translation direction. Note that for very underestimated scale, it approaches 1, while for very over estimated scale, it approaches infinity.

5.3.1 AutoVision Dataset

The distribution of metrics with varying distance and rotation between frames is shown in Figure 4 for evaluated frame pairs from Autovision. For most frame pairs, the estimated rotation and direction of translation are reasonable, except when the rotation between frames is large, causing poor correspondence matching. The translation direction is estimated poorly for very small motions. This is likely due to the ground truth translation being too small.

More interesting is the NTE. It is only small when the motion has some, but not too much rotation, as seen in Figure 4 at the top-right. Moreover, looking at the top-left of Figure 4, there is a clear separation in the NTE when the scale can and cannot be estimated. When there is no rotational motion, the direction of motion may be well estimated, but not the scale.

5.3.2 MultiCam Dataset

A trend similar to that of the AutoVision dataset is observed for both the NTE and RE in Figure 6. However, compared

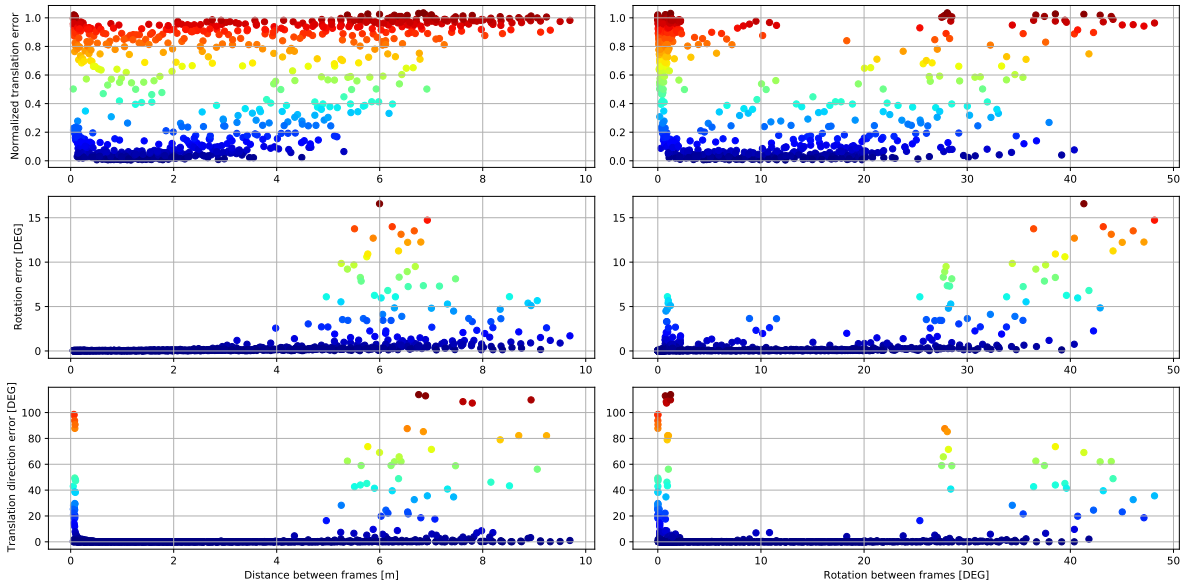

Figure 4: Relative pose estimate metrics for frame pairs of the AutoVision dataset using 17-Point solver and SIFT.

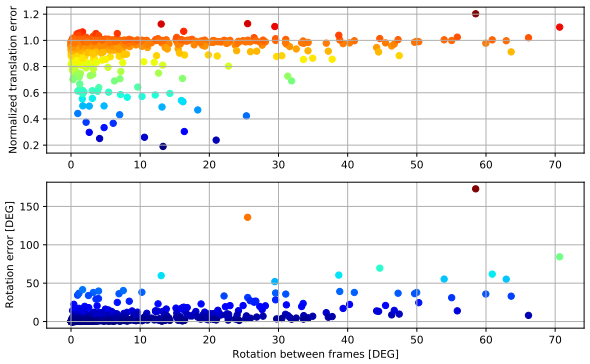

Figure 5: NTE and RE for frame pairs from overcast Oxford RobotCar sequence using 17-Point solver and SIFT.

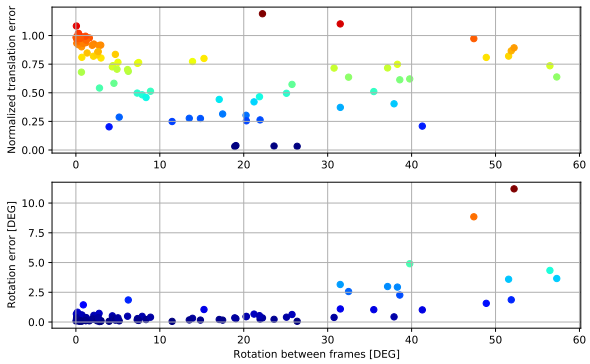

Figure 6: NTE and RE for frame pairs of the MultiCam dataset using 17-Point solver and SIFT.

to on the AutoVision dataset, the pipeline here poorly estimates the translation. Even though MultiCam is a synthetic, the distortion is more significant than AutoVision.

5.3.3 Oxford RobotCar Dataset

The pipeline was run over the three selected sequences of Oxford RobotCar. We show the results on the overcast sequence in Figure 5. Again, for brevity, we only show the NTE and RE. A similar trend in the NTE is also seen here, reinforcing the observation that the scale cannot be estimated when the motion is purely translational. The pipeline performed worse compared to Autovision we believe because of the fast motions and poor calibrations in RobotCar.

5.3.4 Generalized Solver Comparison

We compared the Linear 17-Point, 6-Point, and GE solvers on their run times and initialization quality. We track a running average of the time it takes to complete a solve with RANSAC. The timing results are shown in Table 1. The 6-Point Solver is extremely slow in comparison. It returns 64 possible rotations, and each needs to be checked by estimating the translation and triangulating the inliers. For all solvers, the mean solve times were greater than 1 second which would render them impractical for real-time use.

Table 2 compares the solver performance using the averaged metrics over the MultiCam and RobotCar dataset. Results were not included for the 6-Point solver on RobotCar as it takes too long. GE in general performs better than 17-Point, and is able to estimate the scale of the translation for a larger range of rotations between frames (results omitted

	17-Point	6-Point	GE
RANSAC sample size	17	6	8
Mean solve time [s]	1.22	44.91	1.57

Table 1: Mean solve times over the MCAM dataset.

	MultiCam			RobotCar	
Metric	17-pt	6-pt	GE	17-pt	GE
NTE	0.72	8e5	0.60	0.95	1.79
RE [DEG]	6.34	41.90	9.44	27.34	13.20
TDE [DEG]	0.86	0.59	0.32	4.41	2.66

Table 2: Averages of the metrics with 17-Point solver and GE on the MultiCam and Oxford RobotCar datasets.

Initialization	Overcast	Direct Sunlight	Night
ORB-SLAM	91.9	90.5	85.4
MultiCol	56.3	43.1	32.0
SIFT + 17pt	23.6	18.9	15.0
SIFT + GE	27.9	21.1	18.6

Table 3: Rate of successful initializations for different methods with respect to environment conditions on three Oxford RobotCar sequences. (static sequences removed)

for brevity). Note that the 17-Point solver tends to underestimate while GE sometimes drastically overestimates the scale, causing a higher average NTE with GE on the RobotCar datatset.

5.4. Overall Method Robustness Comparison

Table 3 compares the rate of successful initializations over different Oxford RobotCar sequences. For the existing SLAM methods (MultiCol and ORB-SLAM3), we calculated the percentage of chunks on which all translational errors were below $T_{min} = 2\,\mathrm{m/s}$ and all rotational errors were below $R_{min} = 10\,^\circ/\mathrm{s}$. For generalized solver based methods, we only required 20% of pose estimates to be within the same thresholds to simulate best frame selection that would happen within a real SLAM pipeline. The thresholds were selected rather loosely, because we expect later bundle adjustment to correct small misalignments, and it is just important that the estimate is not grossly wrong.

We see that the success rate decreases significantly when it is sunny or night time, resulting in fewer matched correspondences due to over or under exposure. Moreover, the higher exposure at night causes motion blur which also reduces correspondence accuracy for faster motions.

A qualitative comparison for all initialization methods sequences on the Oxford RobotCar overcast sequence was performed. We have observed that ORB-SLAM3 initializes successfully on almost all sequences. MultiCol initialization succeeds consistently for over 50% of the cases regardless of the amount of translation or rotation, while generalized solver methods fail for all large rotations on this dataset.

6. Discussion

Our work confirms that stereo initialization performs better then general multi-camera approaches. Among the other methods, MultiCol initialization outperforms the generalized solver methods, but it uses SLAM bundle adjustment to improve its robustness. Still, we can observe a failure mode when MultiCol initializes with only a few points on all cameras except one. This could be avoided by using a principled initialization approach.

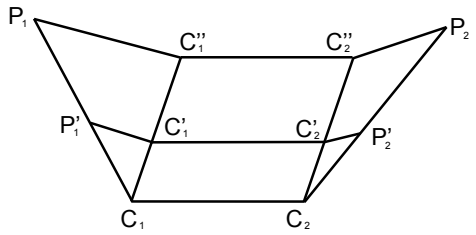

Figure 7: Theoretical justification for poor scale estimation with purely translational motion and no cross-camera correspondences: If we have a two-camera rig C_1C_2 and observe two points, P_1 and P_2, then motions to $C_1'C_2'$ or to $C_1''C_2''$ will both have the same consistent bearing vectors, because the triangles $C_1C_1'P_1' \sim C_1C_1''P_1$ and $C_2C_2'P_2' \sim C_2C_2''P_2$ are similar with the same scaling factor. This holds for an arbitrary number of cameras and points.

6.1. Generalized Camera Initialization

Among OpenGV-based methods, the Generalized Eigensolver performed the best. All solvers exhibited similar behavior to the trend observed in Figure 4. The estimated direction of translation is wrong only for large rotations. There are two modes on the normalized translation error plots: scale is either estimated correctly or not. Incorrect scale estimation almost always happens for purely translational motion. Both translation and rotation estimates become worse for larger rotations and translations. On the normalized translation vs rotation between frames plot (top right, Figure 4), we observe a narrow region of rotations, around $[10^\circ, 25^\circ]$, where the translation direction and scale can be reliably estimated. This inability to recover scale from purely translational motion is actually inherent to all multi-camera solvers that do not exploit cross-camera correspondences and is theoretically justified in Figure 7.

In addition to the case of pure translation, we also observed that the 17-Point and 6-Point solvers fail when there are only correspondences from two or fewer cameras when tested in purely synthetic OpenGV experiments. This matches the analytical results in [9, 6], where the 17-Point solver is shown to be degenerate for two or fewer rigidly connected central projection cameras. This may account for the reduced performance of the generalized camera initialization in the sunny and night sequences of Oxford RobotCar where over or under exposure of any one of the three

cameras can push the system into these degenerate cases. The GE solver was not observed to have this degeneracy.

Lastly, we observed that the frame-to-frame correspondences must satisfy the following necessary conditions to estimate the relative pose well. First, the correspondences must be sampled evenly across all cameras. Unbiased sampling can result in all correspondences being from two or fewer of the cameras, falling into the degenerate case. Second, the presence of matched correspondences with high frame-to-frame disparity are necessary for the translation estimate as low disparity points are effectively infinitely far away and only serve to constrain the rotation. However, high disparity correspondences are more difficult to match between frames, especially for wide-angle cameras with strong distortions or when there are large motions. A frame from Oxford RobotCar is shown in Figure 8 comparing the matched correspondences from SIFT and ORB. SIFT is able to match high disparity correspondences on the ground which ORB could not and is likely why we observed that with ORB the pipeline was almost never able to reasonably estimate translation with the correct scale. Especially for datasets with large distortions, ORB will often not match enough correspondences for the minimal RANSAC sample. For example, in the overcast RobotCar sequence, ORB failed to provided enough correspondences to RANSAC with the 17-Point solver for 56.6% of the evaluated frame pairs, whereas SIFT fails on only 4.7%.

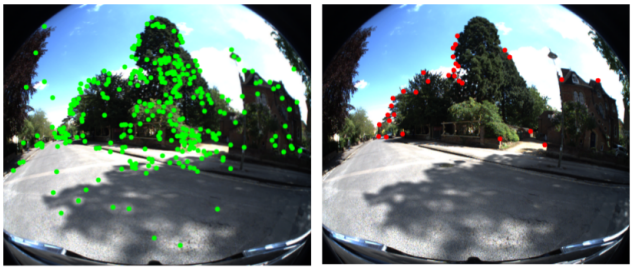

Figure 8: Correspondences with SIFT (left) and ORB (right) on a frame from the Oxford RobotCar dataset.

7. Proposals for Future Work and Conclusion

Given our experiments, we provide insights into improving generalized solvers. First, we need faster solvers. One idea could be enabling batch-processing in RANSAC. Current minimal solvers deal with small matrix computations, but we may be able to speed up the computation if we feed several hypotheses into the solver simultaneously and apply SIMD vectorization to process them all.

The scale of translation is also impossible to estimate without cross-camera correspondences, in the case of purely translational motion. It was also found that RANSAC could find a solution that only satisfies one camera's correspondences, making scale estimation hard. As such, we need a RANSAC sampling strategy that balances the correspondences between different cameras.

Feature matching is also shown to have a great effect on the quality of the result. To reliably estimate motion, we need matches with high disparity and, even better, on different cameras. This poses a problem with ORB and similar methods if they are not correctly tuned, especially in wide-angle cameras. Therefore, wide-angle camera features are also an important research direction for multi-camera SLAM initialization.

8. Conclusion

With our benchmarking, we conclude that stereo initialization is still the best method by a large margin. If it is not available, then GE solver with balanced RANSAC sampling is today's most prominent theoretical approach. However, for strict real-time requirements, one may need to fall back to heuristics. We also propose several research directions which could help theoretically grounded generalized camera based approaches become applicable in practice.

References

[1] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. *IEEE Transactions on Robotics*, pages 1–17, 2021.

[3] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016.

[4] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4695–4702, 2019.

[5] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015.

[6] J.-S. Kim and T. Kanade. Degeneracy of the linear seventeen-point algorithm for generalized essential matrix. *Journal of Mathematical Imaging and Vision*, 37(1):40–48, Feb. 2010.

[7] L. Kneip and P. Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2014.

[8] L. Kneip and H. Li. Efficient computation of relative pose for multi-camera systems. In *20014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014.

[9] H. Li, R. Hartley, and J. hak Kim. A linear approach to motion estimation using generalized camera models. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[10] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys. Towards robust visual odometry with a multi-camera system. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1154–1161. IEEE, 2018.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[13] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3945–3950, 2007.

[14] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[15] R. Pless. Using many cameras as one. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–587, 2003.

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

[17] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, 2006.

[18] H. Stewenius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.

[19] H. Stewenius, M. Oskarsson, K. Aström, and D. Nister. Solutions to minimal generalized relative pose problems. In *2004 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2004.

[20] S. Urban and S. Hinz. Multicol-slam-a modular real-time multi-camera slam system. *arXiv preprint arXiv:1610.07336*, 2016.

[21] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 801–808, 2016.