DESCRIPTOR SCORING FOR FEATURE SELECTION IN REAL-TIME VISUAL SLAM

Prashant Laddha, Om Ji Omer, Gurpreet Singh Kalsi, Dipan Kumar Mandal, Sreenivas Subramoney

Processor Architecture Research Lab, Intel Labs

Many emerging applications of Visual SLAM running on resource constrained hardware platforms impose very aggressive pose accuracy requirements and highly demanding latency constraints. To achieve the required pose accuracy under constrained compute budget, real-time SLAM implementations have to work with few but highly repeatable and invariant features. While many state-of-the-art techniques, proposed for selecting good features to track, do address some of these concerns, they are computationally complex and therefore, not suitable for power, latency and cost sensitive edge devices. On the other hand, simpler feature selection methods based on detector (corner) score, lack in identifying features with required invariance and trackability. We present a notion of feature descriptor score as a measure of invariance under distortions. We further propose feature selection method based on descriptor score requiring very minimal compute and demonstrate its performance with binary descriptors on an EKF based visual inertial odometry (VIO). Compared to detector score based methods, our method provides an improvement up to 10% in ATE (Absolute Trajectory Error) score on EuroC dataset.

Index Terms— Real time SLAM, Low latency, AR/VR

1. INTRODUCTION

Visual SLAM is fundamental to navigation, robotics, immersive experiences, automotive and many such applications. Emerging use cases like AR/VR, HMDs, pico-drones [32] running on highly constrained (power, compute, memory, thermal), low-cost embedded platforms impose challenging performance requirements for SLAM in terms of real-time latency and pose accuracy. For example, AR/VR usecase requires less than 20ms pose latency for immersive experience without motion sickness [17] and power consumption needs to be in sub milli-watt range for battery operated mode [32, 23].

Among many visual SLAM methods, sparse feature based indirect SLAM methods (Figure 1) are preferred due to their computational efficiency and robustness to photometric and geometric distortions [26, 16, 27]. These methods select key features, establish correspondence with features in previous frames and solve for joint camera pose and feature locations with solver methods like bundle adjustment (BA) or extended Kalman filter (EKF). To meet challenging performance requirements with highly constrained resources, visual SLAM methods need to consider following aspects: 1) process input sensor data in single pass 2) maintain a minimal set of high fidelity features and 3) avoid multiple iterative solver stages. First one minimizes data-transfers from system memory for lower power and latency considerations which are essential for SLAM on ultra low-cost, resource constrained systems. Second and third aim at reducing the compute complexity without compromising on SLAM accuracy. Methods like ORB-SLAM [26] can afford to ignore these considerations to achieve high SLAM accuracy, as either they cater to offline mode of operation or run on systems with massive compute and power budget. But for real-time SLAM methods maintaining the accuracy becomes quite challenging. Table 1 compares the accuracy (ATE scores) for ORB-SLAM[26] and RC-SLAM[16] representing two categories of SLAM.

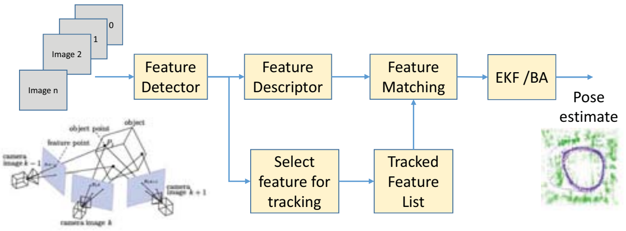


Figure 1: Sparse feature based Visual SLAM. It maintains list of features (3D landmarks) to be tracked. The matches for these features are searched in every subsequent frame

The compute complexity of feature based SLAM grows in quadratic or cubic order with the number of features used in feature matching and solver. The feature selection plays a critical role to strike a balance between compute and accuracy requirements. In absence of good features, optimization techniques like RANSAC [7], geometric consistency [26], relocalization [26] are adopted, which require iterative search and solver stages and often not suitable for real-time usages. Therefore, we argue in favor of choosing feature point wisely. Recent works such as [15, 11] have been proposed for im-

Streams	V1_01	V2_01	V2_02	MH_01	MH_02	MH_02
ORB SLAM (Mono)	0.015	0.015	0.017	0.070	0.066	0.071
RC-SLAM (Visual Inertial Mono)	0.032	fail	fail	0.32	0.59	1.38

Table 1: ATE for ORB-SLAM mono [25] and RC-SLAM mono. The tracking part of ORB SLAM mono takes 30-34 ms/frame whereas RC-SLAM takes 6-7 ms/frame on Intel i7@3.7 Ghz.

978-1-7281-6395-6/20/$31.00 ©2020 IEEE

proved feature selection based on machine learning methods. They promise high fidelity features but require significantly high compute resources. Also, the training with controlled environment datasets can restrict their usecases.

To improve accuracy of real-time visual SLAM, we propose simple feature selection criterion on low-complexity feature detection description methods such as ORB [29]. In this paper we make following contributions:

1. We show, through empirical data, that the track-length of a feature (defined as number of frames for which a feature can be tracked) directly impacts pose accuracy.
2. We propose descriptor score based method for selecting features with higher invariance and robustness.
3. We implement descriptor score on ORB [29] in a real-time visual SLAM application RC-SLAM [16] and show that our method helps curing many failing scenarios while improves ATE scores by 10% for other scenarios in EuroC dataset [4].

2. BACKGROUND AND MOTIVATION

2.1. Feature Selection and SLAM Accuracy

Feature detection and tracking are important to visual SLAM. Over the years, many feature detectors and descriptor methods have been explored for improving the feature repeatability, invariance (to rotation, scale and affine transformations), discriminability and computing efficiency [14, 21, 28, 29, 21, 2, 5, 20]. While higher repeatability assists in locating same features in multiple frames captured from different camera view points, higher invariance is required to find correct feature matches. Both these attributes are necessary for correct data association as required in SLAM. When features are no longer trackable, new features needs to be detected and localized before they could be used for pose estimation. This process may take some time and during that period if number of tracked features are not sufficient, the stability of SLAM system could be compromised affecting the overall pose accuracy.

To illustrate the correlation between SLAM accuracy and track-length, we take two sequences from EuroC datasets, one (V1_01) where RC-SLAM is able to estimate trajectory accurately and other (V2_02) where estimated trajectory deviates from ground truth by a big margin as shown in Figure 2. Figure 3 and 4 show frame-wise percentage of correctly tracked features and histogram of *track-lengths* in these sequences. The fraction of correctly tracked features in sequence V2_02 are consistently low compared to sequence V1_01. Consequently, track-lengths for majority of features in the sequence V2_02 are very short (4-5 frames only) compared to the sequence V1_01 with accurate trajectory estimates. This shows a strong correlation between feature track-lengths and accuracy of trajectory estimates. In visual SLAM, each feature serves as measurement for corresponding landmark location in 3D world coordinate. Tracking a feature over more number of frames is equivalent to gathering more measurements of same state variable. Since SLAM is a iterative Bayesian estimation, more number of measurements of same state variable help to improve the accuracy of estimate. Therefore, the features which can be tracked longer contributes to improved SLAM accuracy.

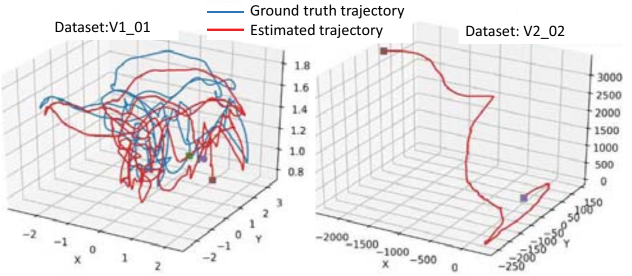


Figure 2: Accurate (left) vs failed (right) trajectory estimates. In right plot, ground truth is not even visible because estimated trajectory is way too off the mark compared to ground truth

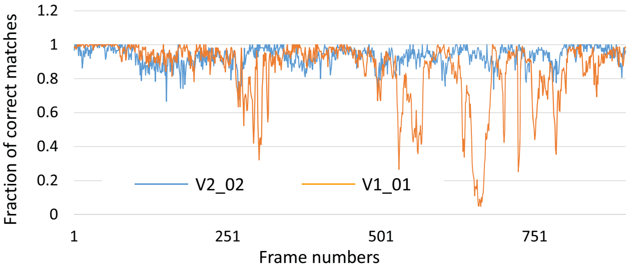


Figure 3: Framewise feature match percentage for datasets with accurate and inaccurate pose estimates

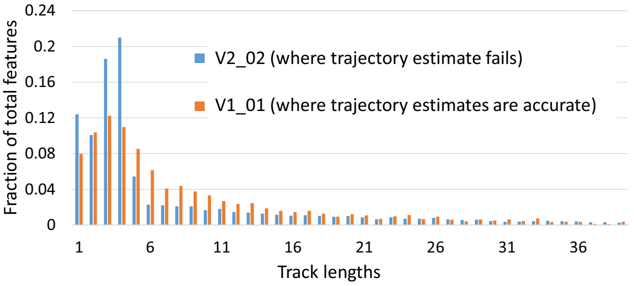


Figure 4: Track lengths (number of frame for which feature is tracked) for datasets with accurate and inaccurate pose estimates

Based on various studies on feature detection and description methods, we observe that every detected feature is not repeatable and invariant [24, 22, 33, 1]. Therefore, to track features correctly over more number of frames, features which repeatable, discriminative and invariant to various geometric, photometric distortions need to selected.

2.2. Related Works

Feature selection techniques are generally applied at either of the two stages in the SLAM processing:

Selection before feature matching: Detector scores like Harris score [14], Shi-Tomasi [31], FAST score [28], which quantify the corner strength are the most commonly used methods to select top few features for SLAM. Selecting features based on spatial distribution have shown better results. These algorithms often use either a uniform spatial grid, oct-tree based dynamic grid[29], or adaptive non-maximal suppression methods [3], [12]. Learning based methods like [15] use classifiers to select descriptors having higher chance of finding a match to reduce number of search points.

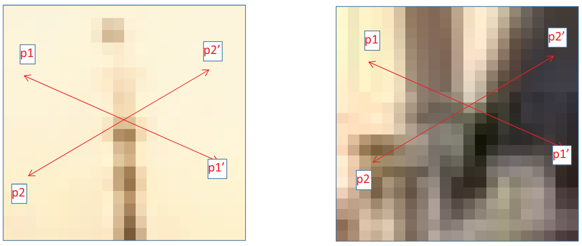
Selection after feature matching: Outlier rejection using RANSAC [7] has been used many SLAM algorithms [18, 26, 10] to filter out erroneous, unreliable feature matches from participating in EKF/BA. Geometric and Structural consistency checks like depths, view points have been used to reject unreliable feature-matches in [29]. The [11] proposes a stability classifier based on reprojection error to predict stable feature points for CNN based feature tracking front-end. Information theoretic approaches for feature selection using information gain [9], entropy [30], trace [19], covariance ratio [6] or observability [35, 34] have also been proposed to reduce uncertainties in estimation.

Majority of the SLAM implementations select features based on detector scores [14, 31, 28] and spatial distribution. Other techniques selecting features after matching, though help in pruning out bad matches, but they do not save on the computation required in feature detection, description generation or matching. Though detector scores help to select features with good repeatability, they are often not sufficient to identify features with higher invariance to various distortions.

3. FEATURE DESCRIPTOR SCORE

A feature descriptor is a statistical representation of pixel values in a region around the feature point using distribution of gradients [21, 2] or relative intensity differences [29, 5, 20]. A binary descriptor contains a vector of bits, in which each bit signifies pixel intensity comparison for a pair of sample points around the detected feature point. The invariance of descriptor to geometric, photometric distortions depends on the strength of structure among the pixels around the feature point. For example, if there are strong, prominent structures like distinct edges, high contrast regions around feature point, the descriptor is more likely to survive distortion. Inspiring from this, we introduce a notion of descriptor score as a measure of descriptor invariance. Features with higher descriptor score are more likely to find correct match.

To illustrate the relation between descriptor score and invariance of binary descriptor, we show two example scenarios with image patches around detected features Figure 5, one with a strong structure and other with a weak structure in image. The left patch gives high detector score because the corner (blob) is distinct compared to its neighbouring pixels though it would not have highly invariant descriptor. With slight distortion like noise, many descriptor bits could get flipped because there is hardly any structure around the feature point. On the other hand the patch on right has strong structure. In this case, descriptor will remain invariant because of large difference in pixel intensities for each pair of sample points. Based on these observations, we define descriptor score as follows-



Less invariant feature due to weak texture around feature point

More invariant feature due to strong texture around feature point

Figure 5: Illustration for Weak and Strong feature

$$desc\_score = \frac{1}{N} \sum_{i=0}^{N-1} |p_i - p'_i| \qquad (1)$$

where p_i and p'_i are the pixel values in i^{th} pair out of total N pairs used in descriptor generation and N is descriptor length.

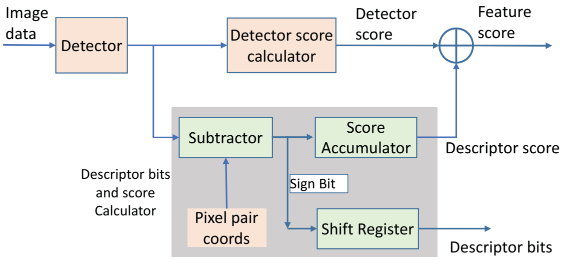


Figure 6: Key-point selection with feature descriptor score

The binary descriptor score can be calculated during the descriptor generation without any significant computational overhead. Figure 6 depicts the process of feature selection using descriptor score. For feature point detected by the detector, we calculate the descriptor score using Eq.1. The combined score for feature point is computed using detector and descriptor score to give importance to regions with high contrast and strong structure. The idea of descriptor score could be easily applied to any feature descriptor by adopting appropriate scoring technique based on the description method.

4. EVALUATION

We used RC-SLAM [16] as real time SLAM and modified it to use ORB features to create an improved baseline. Around 80 features are tracked. We replaced feature selection in the baseline with the proposed descriptor score based feature selection method. This version is referred as **Ours** in this section. Table 2 compares trajectory estimates with our proposed method against the baseline using absolute trajectory error (ATE) on EuroC dataset[8]. The descriptor score based fea-

	ATE rmse (mtrs)				
Streams	**Baseline**	**Ours**	**Streams**	**Baseline**	**Ours**
V1_01	0.381	**0.268**	**MH_01**	0.325	0.352
V1_02	0.273	0.270	**MH_02**	0.599	**0.579**
V1_03	0.347	**0.341**	**MH_03**	1.384	**1.323**
V2_01	fails	**2.784**	**MH_04**	0.989	**0.911**
V2_02	fails	**1.146**	**MH_05**	1.199	**0.877**

Table 2: ATE improvements with descriptor score based feature selection on EuroC dataset

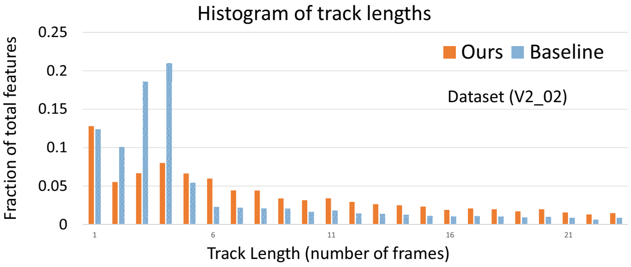


Figure 7: Improvements in track lengths with descriptor score based feature selection

ture selection improves ATE results for most of the streams. Also, by selecting more invariant features our method is able to overcome the failure cases seen in the baseline for streams V2_01 and V2_02 due to tracking failures (Section 2.1).

Improved Trackability with descriptor score based selection: To evaluate the impact of our method on track-lengths, we build a histogram of track-lengths of each selected feature in stream V2_02 (Figure 7). With the baseline almost 60% of selected features could not be tracked beyond 3 or 4 frames. Frequent drops in number of tracked features lead to inaccurate pose estimates and also requires more compute for detecting and localizing new features. With our method, larger fraction of features gets tracked over longer duration. Figure 8 shows feature matches in each frame during tracking for stream V2_02. Our method provides consistently higher feature matches compared to baseline.

Distribution of descriptor distances: To validate the efficacy of the proposed method in selecting more invariant features, we look at the distribution of descriptor distances for all feature matches found during tracking. For each feature (f) selected for tracking, and its best match (fm) in each subsequent frame we calculate the normalized descriptor distance as $nd = hamming\_distance(f, f_m)/desc\_length$. Figure 9 shows the histogram of normalized descriptor distances $\{nd\}$ over complete sequence for all tracked features. With the proposed method, almost 80% of tracked features found matching descriptors within normalized distance of 0.2 which indicates that the selected features remained relatively invariant.

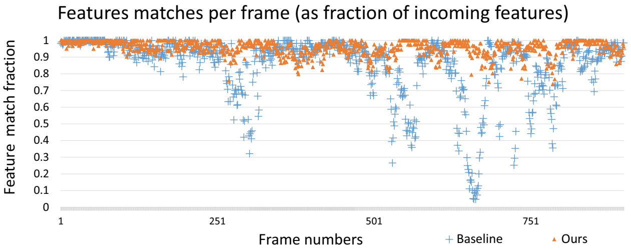


Figure 8: Comparison of frame wise feature matches for V2_02 dataset with descriptor and detector score based feature selection

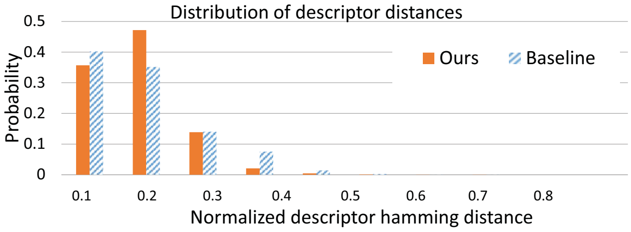


Figure 9: Distribution of feature matching distances (V2_02 dataset) with descriptor and detector score based feature selection

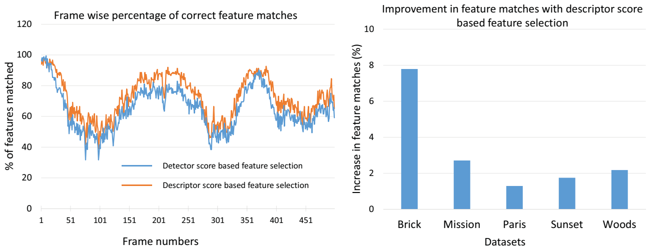


Figure 10: (a) Frame wise comparison of percentage of correctly matched features for one of the dataset (brick) (b) Average improvements in feature matches for different datasets in [13]

Improvements in feature tracking: We evaluate the impact of descriptor score based selection on standalone feature tracking. For this evaluation we use datasets from [13] where features from the first frame are visible in all subsequent frames and there are significant geometric, photo metric distortions with very high, un-constrained camera movement. In the first frame we select $\mathbf{N}$ features $(\mathbf{N} = 50)$ and try to find matches in all subsequent frames and verify correctness of match using the ground truth. With descriptor score based feature selection, we get consistent 8%-10% improvements in correctly matched features in each frame as shown in Figure 10(a) for Brick dataset. Figure 10(b) shows noticeable average improvement over entire sequence in other datasets.

5. CONCLUSION

We propose a novel method for feature selection using descriptor score which helps to select more invariant and robust features as compared to widely used detector score based selection methods. The real-time SLAM on resource constrained devices requires high quality pose estimates at low complexity and latency. The proposed method is simple to implement and improves the performance without compromising on computational efficiency as compared to other state-of-the-art techniques. We demonstrate functioning of descriptor score with ORB like binary descriptor in an end-to-end real-time visual SLAM application. As a future work, the idea of descriptor score can be extended to other types of feature descriptors and their applications in visual SLAM.

6. REFERENCES

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5173–5182.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[3] M. A. Brown, R. Szeliski, and S. A. J. Winder, "Multi-image matching using multi-scale oriented patches," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, 2005, pp. 510–517.

[4] 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*, 2016.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[6] F. A. Cheein, G. Scaglia, F. di Sciasio, and R. Carelli, "Feature selection criteria for real time ekf-slam algorithm," *International Journal of Advanced Robotic Systems*, vol. 6, no. 3, p. 21, 2009.

[7] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[8] T. dataset and Tool, "https://vision.in.tum.de/data/datasets/rgbd-dataset/tools."

[9] A. J. Davison, "Active search for real-time vision," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1052–1067, 2007.

[11] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Self-improving visual odometry," *CoRR*, vol. abs/1812.03245, 2018. [Online]. Available: http://arxiv.org/abs/1812.03245

[12] S. Gauglitz, L. Foschini, M. Turk, and T. Hllerer, "Efficiently selecting spatially distributed keypoints for visual tracking," in *2011 18th IEEE Conference on Image Processing*. IEEE, 2011, pp. 1869–1872.

[13] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International Journal of Computer Vision*, vol. 94, no. 3, pp. 335–360, 2011.

[14] C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[15] W. Hartmann, M. Havlena, and K. Schindler, "Predicting matchability," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 9–16.

[16] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[17] R. Kijima and K. Miyajima, "Measurement of head mounted display's latency in rotation and side effect caused by lag compensation by simultaneous observation an example result using oculus rift dk2," in *2016 IEEE Virtual Reality (VR)*, March 2016, pp. 203–204.

[18] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2007, pp. 1–10.

[19] R. Lerner, E. Rivlin, and I. Shimshoni, "Landmark selection for task-oriented navigation," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 494–505, 2007.

[20] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 IEEE international conference on computer vision (ICCV)*. IEEE, 2011, pp. 2548–2555.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] S. Madeo and M. Bober, "Fast, compact, and discriminative: Evaluation of binary descriptors for mobile applications," *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 221–235, 2017.

[23] D. K. Mandal, S. Jandhyala, O. J. Omer, G. S. Kalsi, B. George, G. Neela, S. K. Rethinagiri, S. Subramoney, L. Hacking, J. Radford, E. Jones, B. Kuttanna, and H. Wang, "Visual inertial odometry at the edge: A hardware-software co-design approach for ultra-low latency and power," in *Proceedings of the 2019 Design, Automation and Test in Europe Conference and Exhibition, DATE 2019*. IEEE, 2019.

[24] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 2681–2684.

[25] R. Mur-Artal and J. D. Tards, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, April 2017.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[27] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. PP, 08 2017.

[28] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[29] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[30] L. X. Sen Zhang and M. D. Adams, "Entropy based feature selection scheme for real time simultaneous localization and map building," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[31] J. Shi and C. Tomasi, "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[32] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze, "Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1106–1119, April 2019.

[33] S. Wu, A. Oerlemans, E. M. Bakker, and M. S. Lew, "A comprehensive evaluation of local detectors and descriptors," *Signal Processing: Image Communication*, vol. 59, pp. 150–167, 2017.

[34] G. Zhang and P. Vela, "Optimal observability and minimal cardinality localization and mapping," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2015.

[35] G. Zhang and P. A. Vela, "Good features to track for visual slam," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.