Robust Real-Time Visual Odometry with a Single Camera and an IMU

Laurent Kneip
laurent.kneip@mavt.ethz.ch
Margarita Chli
margarita.chli@mavt.ethz.ch
Roland Siegwart
rsiegwart@ethz.ch

Autonomous Systems Lab
ETH Zurich, Switzerland

Abstract

The increasing demand for real-time high-precision Visual Odometry systems as part of navigation and localization tasks has recently been driving research towards more versatile and scalable solutions. In this paper, we present a novel framework for combining the merits of inertial and visual data from a monocular camera to accumulate estimates of local motion incrementally and reliably reconstruct the trajectory traversed. We demonstrate the robustness and efficiency of our methodology in a scenario with challenging camera dynamics, and present a comprehensive evaluation against ground-truth data.

1 Introduction

Good local estimation of egomotion forms the backbone of any modern high performance localization/navigation system. While approaches to Simultaneous Localization And Mapping (SLAM) – as for example [5, 10] – aim to build a global map of all visited locations, for autonomous local navigation it is usually enough to obtain good estimates of the local trajectory – similar to the way humans employ to navigate in their environment. Even SLAM techniques however, have a critical dependency on accurate and timely estimates of frame-to-frame motion for a successful end result [4].

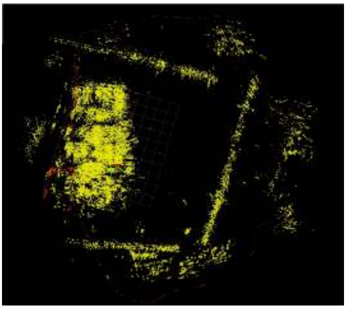


Figure 1: Map created by our VO implementation.

The term *Visual Odometry* (VO) has been introduced and investigated in both the computer vision and robotics communities for a few years now, referring to the problem of estimating the position and orientation of a camera-carrying platform by analyzing images taken from consecutive poses. Without any assumptions on prior knowledge about the camera's workspace, approaches to VO promise general applicability to incrementally reconstruct the trajectory of the camera with the only requirement of visual input data.

BMVC 2011 http://dx.doi.org/10.5244/C.25.16

Approaches to VO range from super-dense optical flow to matching sparser, salient image regions from one image to the next. Back in 1981, Lucas and Kanade [16] assumed constant flow in local pixel regions which has been applied in [3] for VO estimation in various types of terrain. More recently, the computer vision literature has seen some very powerful, yet computationally demanding methods for optical flow as for example [24]. In navigation/localization tasks however, pose estimation has to be performed on a per-frame basis hence *real-time* is a requirement, thus sparser correspondence-based methods able to run on general platforms have seen great popularity for such tasks.

The seminal work in [19] presented a framework for both monocular and stereo VO which has been subsequently used as a basis for many successful systems. In [8, 12] we have seen impressive setups for real-time VO from stereo images, while more recently, Konolige *et al*. [13] demonstrated results over long trajectories on off-road terrain, following the trend for lightweight high-precision algorithms performing in the presence of challenging camera dynamics. The inherent difficulty there lies in robustly resolving data association as feature tracks become highly jerky and mismatches are far more likely. The aforementioned algorithms apply classical Structure From Motion (SFM) principles, and thus return triangulated structure alongside the estimated camera trajectory. Figure 1 depicts an example of a point cloud which has been successively triangulated by our VO setup, recovering both the camera motion and the depth of sparse feature correspondences in parallel.

It has long been acknowledged that the use of inertial sensors together with cameras can complement each other in challenging scenarios, aiding the resolution of ambiguities in motion estimation arising when using each modality alone [23]. Here, we employ a monocular camera and an Inertial Measurement Unit (IMU) to recover relative camera motion, in a sensor setup available in practically most modern smart phones (*e.g*. iPhone, Google phones). We use a RANSAC based [6] hypothesize-and-test procedure as in [19], only here we specifically address the efficiency and robustness of monocular VO, presenting an elegant framework which exploits the additional benefits of the available rotation information.

As shown in several recent works [7, 9, 14], knowledge about the vertical direction can for instance be used for reducing the minimum number of points for instantiating a hypothesis about the relative camera pose down to three or even only two in the perspective pose computation case. However, even though the vertical direction can be obtained from inertial data, it only works reasonably well in the static case. In this work, we propose an alternative tightly coupled SFM approach, that incorporates short-term full 3D relative rotation information from inertial data in order to support the geometric computation. We demonstrate the successful application of our pose estimation methodology on data obtained using a Micro Aerial Vehicle (MAV) exhibiting full 3D motion with notably much more challenging dynamics than hand-held cameras or ground vehicles. The performance of the proposed system is assessed in terms of both speed and accuracy with respect to ground truth.

2 Framework

The proposed strategy for robust, continuous pose computation of the camera operates in two modes: firstly, a local point cloud is initialized from two views exhibiting sufficient disparity among them, while for subsequent poses where the computed disparity is not big enough to triangulate a new point cloud, the method switches to a perspective pose estimation algorithm in order to derive the relative camera pose (following the paradigm of [20]). We follow a keyframe-based methodology such that whenever the median-filtered disparity exceeds a

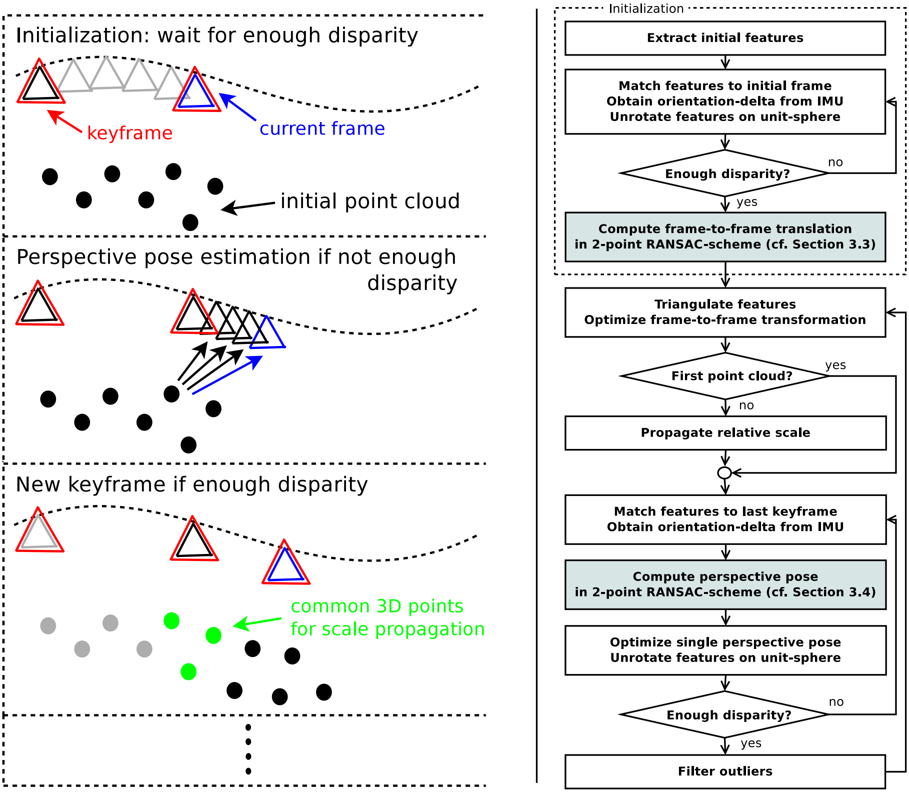

Initialization: wait for enough disparity
keyframe
current frame
initial point cloud
Perspective pose estimation if not enough disparity
New keyframe if enough disparity
common 3D points for scale propagation
Initialization
Extract initial features
Match features to initial frame
Obtain orientation-delta from IMU
Unrotate features on unit-sphere
Enough disparity?
no
yes
Compute frame-to-frame translation in 2-point RANSAC-scheme (cf. Section 3.3)
Triangulate features
Optimize frame-to-frame transformation
First point cloud?
yes
no
Propagate relative scale
Match features to last keyframe
Obtain orientation-delta from IMU
Compute perspective pose in 2-point RANSAC-scheme (cf. Section 3.4)
Optimize single perspective pose
Unrotate features on unit-sphere
Enough disparity?
no
yes
Filter outliers

Figure 2: Overall scheme and execution flowchart of the proposed VO. The blue boxes highlight the key-contributions of this paper.

certain threshold in the number of pixels, we triangulate a new point cloud and the two frames used for this process serve as *keyframes*. In subsequent frames, the current pose is estimated with respect to the scene model constructed from the most recent keyframe-pair until a new keyframe is added to the system. Figure 2 illustrates all the steps of the overall scheme and the execution flowchart of the algorithm. The algorithm is based on a static scene model, meaning that it is able to cope with moving parts in the structure up to a limited extend only.

Both the initial relative frame-to-frame transformation and the perspective pose computation are performed within robust RANSAC outlier-rejection schemes. Each perspective pose as well as the initialization of a new point cloud get refined by an iterative non-linear optimization step on the inlier subset. Note that the initialization of a new point cloud is preceded by a simple outlier rejection based on the reprojection error. Finally, relative scale propagation is performed by considering the subset of features present in all three views: the current keyframe and the two keyframes from the previous point cloud. The relative scale is preserved by imposing the constraint that the newly triangulated features occur at the same depth they occur in the previous point cloud. Even though the relative scale is in principle preserved via initializing the relative transformation in between the keyframes from the perspective pose computation, the final scale propagation step is still required since this is not necessarily preserved during the 6D non-linear refinement of the relative transformation (the optimization is invariant against variations of the norm of the translation vector).

The scheme has some similarities with the state-of-the-art monocular scheme presented in [19]. However, the novel methodology presented here has been specifically designed to increase both the efficiency and the robustness of monocular VO estimation by exploiting priors about the relative rotations from an additional IMU. Therefore, the main difference here is that in our framework, except for the scale propagation step, we only need to consider two-frame matches. The second major difference and also the key-contribution of this paper, is that this allows for a reduction in the number of points used in the RANSAC-hypotheses down to a minimum of two, for both the 2D-to-2D correspondence estimation during initialization and the 3D-to-2D correspondence problem upon perspective pose computation. It is important to note that these two-point algorithms do not suffer from any geometric degeneracies and always return a unique solution.

Knowledge of the relative camera rotation also provides great benefit during initialization of the algorithm since disparities due to rotation can be compensated for, and as a result, the method can guarantee that there is enough translation between the first two keyframes (boosting robustness of triangulation) despite that there is no prior information about the structure of the scene that the camera is exploring. Finally, it is worth noting that all per-feature error measures are realized on the unit sphere, avoiding frequent projections back and forth between the camera frame and the image plane.

3 Geometric Pose Computation

After a brief introduction to notation, here we detail the individual modules of our framework that elegantly combine visual and inertial information to provide VO estimates.

3.1 Conventions

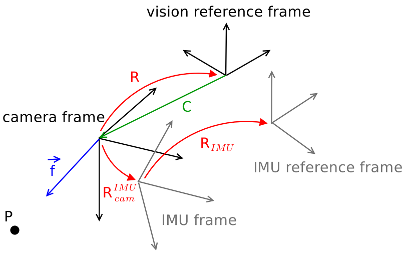


Figure 3: Notation.

Figure 3 indicates the notation used throughout this paper. The intrinsic parameters of the camera are assumed to be known. Here, we use the omnidirectional camera model presented in [22] which provides the generality to allow this framework to operate on any type of optical system (*e.g.* catadioptric, dioptric cameras). The rotation from the camera frame to the IMU frame is also assumed to be known and given by R^{IMU}_{cam}. This can be obtained by off-the-shelf toolboxes as for instance [15].

The position of the camera with respect to the vision reference frame is given by C. The rotation from the camera to the reference frame is expressed with R, while the rotation from the IMU frame to the inertial reference frame is given by R_{IMU}. Unit feature vectors pointing from C to a certain world point P are expressed with $\vec{f}$ and can always be obtained using the camera calibration parameters to project features from the image plane onto the unit sphere.

3.2 Rotation Priors from the IMU

The relative rotation priors from the IMU are obtained by a fast integration of the high-frequency gyroscopic measurements. Typically, common low-cost IMUs already accomplish this integration internally using a complementary filter along with the gravity direction obtained from the acceleration signals. Experience has shown that the resulting orientation of the IMU is only affected by a slowly changing drift term and that short-term relative orientation of the system can hence be recovered safely, directly from consecutive orientation information delivered by the IMU. The relative rotation of the current camera frame with respect to a keyframe is given by

$$R^{key}_{current} = R^{IMU^T}_{cam} \cdot R^{T}_{IMU_{key}} \cdot R_{IMU_{current}} \cdot R^{IMU}_{cam} \,. \tag{1}$$

The necessary condition for obtaining good relative orientation information is that the system should be in motion so that the temporal difference between different keyframes can be bounded. This way, the angular errors of the priors about the relative rotation remain small and can be easily compensated for in the non-linear refinement steps. If the system enters an approximately static phase, there still exists the option of switching back to vision-only based RANSAC-algorithms without affecting the overall scheme or the computational efficiency of the algorithm (the major part of the processing resources is used for other tasks than the RANSAC-iterations).

3.3 Frame-to-Frame Initialization

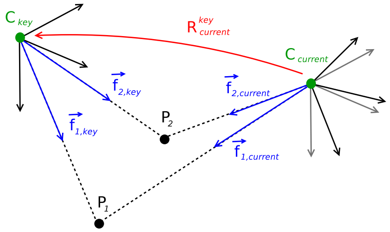


Figure 4: Two point correspondences between consecutive frames.

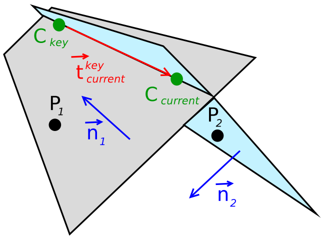


Figure 5: Translation as an intersection of the resulting epipolar planes.

The derivation of the frame-to-frame translation based on a relative rotation information obtained from the IMU follows a relatively easy strategy. The relative orientation between the current frame and the previous keyframe is given by (1). As indicated in Figure 4, the unit feature vectors of the current frame can then be unrotated on the unit sphere using $\vec{f}_{i,current\ unrotated} = R^{key}_{current} \cdot \vec{f}_{i,current}$. The feature vectors expressed in frames with identical orientation, the normal vector of the epipolar plane of each feature correspondence $\vec{f}_{i,key} \leftrightarrow \vec{f}_{i,current}$ is then simply given by

$$\vec{n}_i = \vec{f}_{i,key} \times (R^{key}_{current} \cdot \vec{f}_{i,current}). \tag{2}$$

The magnitude of the normal vector $\vec{n}_i$ resulting from the cross-product can be used as a degeneracy measure of the determined epipolar plane normal. As shown in Figure 5, two feature correspondences which result in distinct epipolar planes can then be used for intersecting the direction of the translation vector. This is given by

$$\vec{d}^{\,current}_{key} = \vec{n}_1 \times \vec{n}_2 \,. \tag{3}$$

If the plane normal vectors are normalized to one before the cross-product computation, the magnitude of the result can again be used in order to determine the degeneracy of the result. Since the scale of the problem is anyway not determinable, the final 3D translation vector is found to be $\vec{t}\,_{key}^{current} = \pm \frac{\vec{d}\,_{key}^{current}}{||\vec{d}\,_{key}^{current}||}$. In order to determine the sign, we need to impose that $(\vec{f}_{i,key} - R_{current}^{key} \cdot \vec{f}_{i,current}) \cdot \vec{t}\,_{key}^{current} > 0$.

This two-point algorithm returns a unique solution and is executed in a robust RANSAC-scheme. During each iteration, samples with too small cross-product magnitudes are rejected. The error function simply consists of the reprojection error of the triangulated features. However, as mentioned previously, the points are not transformed back into image space. For the sake of computational efficiency, the error function simply uses dot-products between normalized feature vectors on the unit sphere.

3.4 Perspective Pose Computation

In order to derive an IMU-supported pose computation of the camera with respect to a 3D point-cloud, we use as basis the classical P3P-problem (Perspective-3-Point-problem). The P3P-problem consists of finding a minimal solution for the position of the camera inside the world reference frame under the knowledge of three 3D-to-2D point correspondences. We draw inspiration from the novel parametrization presented in [11], which introduces a direct computation of the absolute camera position and orientation. As opposed to classical solutions, it allows to compute the position of the camera in a single stage, without the intermediate derivation of the considered points for the hypothesis inside the camera frame or the subsequent point alignment step. Since the method presented here is largely based on this parametrization, we give a brief overview below.

The goal here is to find the exact position $C_{current}$ and orientation matrix $R_{current}$ of a camera with respect to the world frame (O,X,Y,Z), under the condition that the absolute spatial coordinates of two observed feature points P_1 and P_2 are given. Since the intrinsic camera parameters are known, we can assume that the unit vectors $\vec{f_1}$ and $\vec{f_2}$ pointing towards the two feature points considered in the camera frame are given. Furthermore, we assume that the position C_{key} and orientation R_{key} of the most recent keyframe are known. The relative orientation between the two camera frames is given by (1). Hence, by incorporating the known absolute orientation of the keyframe R_{key}, the prior for the absolute orientation of the current frame is obtained by

$$R_{current} = R_{key} \cdot R_{current}^{key} = R_{key} \cdot R_{cam}^{IMU^T} \cdot R_{IMU_{key}}^{T} \cdot R_{IMU_{current}} \cdot R_{cam}^{IMU}. \tag{4}$$

Again, since the short-term integral of gyroscopic signals are only affected by very low drift terms, this enables a reasonable prior on the absolute orientation of the current camera frame. Let us denote the current camera frame with ν. We define a new, intermediate camera frame τ from the feature vectors $\vec{f_1}$ and $\vec{f_2}$ inside ν. As shown in Figure 6, the new camera frame is defined as $\tau = (C, \vec{t_x}, \vec{t_y}, \vec{t_z})$, where

$$\vec{t_x} = \vec{f_1}, \qquad \vec{t_z} = \frac{\vec{f_1} \times \vec{f_2}}{||\vec{f_1} \times \vec{f_2}||}, \qquad \text{and} \qquad \vec{t_y} = \vec{t_z} \times \vec{t_x}. \tag{5}$$

$T = [\vec{t_x}, \vec{t_y}, \vec{t_z}]^T$ then represents the rotation matrix from ν into τ, and feature vectors can be transformed by $\vec{f}_i^{\tau} = T\vec{f}_i^{\nu}$. The next step involves the definition of a new world frame

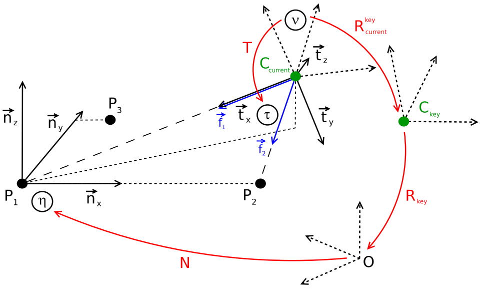


Figure 6: Illustration of the intermediate camera frame $\tau = (C, \vec{t}_x, \vec{t}_y, \vec{t}_z)$ and the intermediate world frame $\eta = (P_1, \vec{n}_x, \vec{n}_y, \vec{n}_z)$.

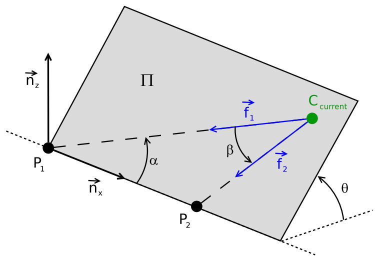


Figure 7: Novel parametrization of the P3P-problem as presented in [11].

η from the world points P_1, P_2, and an additional virtual control point P_3. The new spatial frame is defined as $\eta = (P_1, \vec{n}_x, \vec{n}_y, \vec{n}_z)$, where

$$\vec{n}_x = \frac{\overrightarrow{P_1P_2}}{||\ \overrightarrow{P_1P_2}\ ||}, \qquad \vec{n}_z = \frac{\vec{n}_x \times \overrightarrow{P_1P_3}}{||\vec{n}_x \times \overrightarrow{P_1P_3}\ ||}, \qquad \text{and} \qquad \vec{n}_y = \vec{n}_z \times \vec{n}_x. \tag{6}$$

Via the transformation matrix $N = [\vec{n}_x, \vec{n}_y, \vec{n}_z]^T$, world points can finally be transformed into η using $P_i^\eta = N \cdot (P_i - P_1)$. The resulting situation is illustrated in Figure 6. The condition of existence of η is that P_1, P_2, and P_3 are not colinear. This can be easily avoided by choosing P_3 such that $\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}$ is not zero.

In the following, we focus on the transformation between η and τ. We define the semi-plane Π that contains points P_1, P_2, and C, and hence also the unit vectors $\vec{n}_x$, $\vec{t}_x$, $\vec{t}_y$, $\vec{f}_1$, and $\vec{f}_2$. Points P_1, P_2, and C form a triangle of which two parameters are known, namely the distance d_{12} between P_1 and P_2, and the angle β between $\vec{f}_1$ and $\vec{f}_2$. As shown in detail in [11], the transformation between η and τ is depending only on the two angular parameters α and θ indicated in Figure 7. The camera center C inside η and the transformation matrix Q from η to τ are given by

$$C^\eta(\alpha,\theta) = \begin{pmatrix} d_{12}\cos\alpha(\sin\alpha\cdot\cot\beta+\cos\alpha) \\ d_{12}\sin\alpha\cos\theta(\sin\alpha\cdot\cot\beta+\cos\alpha) \\ d_{12}\sin\alpha\sin\theta(\sin\alpha\cdot\cot\beta+\cos\alpha) \end{pmatrix}, \tag{7}$$

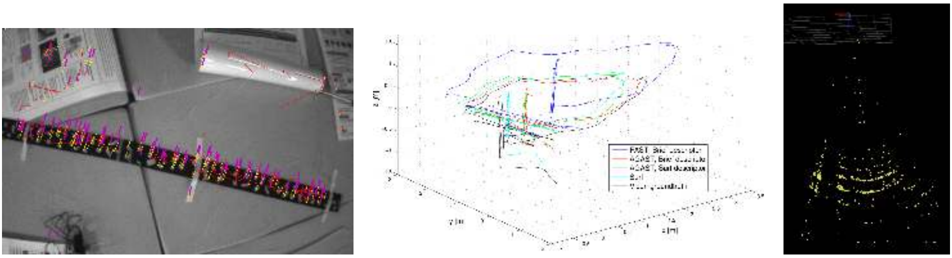
$$Q(\alpha,\theta) = \begin{pmatrix} -\cos\alpha & -\sin\alpha\cos\theta & -\sin\alpha\sin\theta \\ \sin\alpha & -\cos\alpha\cos\theta & -\cos\alpha\sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}, \tag{8}$$

Following the chain of rotations and substituting with (4), we obtain

$$\begin{aligned} R_{current} &= N^T \cdot Q^T \cdot T \\ \Rightarrow Q &= T \cdot R^T_{current} \cdot N^T \\ \Rightarrow Q &= T \cdot R^{IMU^T}_{cam} \cdot R^T_{IMU_{current}} \cdot R_{IMU_{key}} \cdot R^{IMU}_{cam} \cdot R^T_{key} \cdot N^T \end{aligned} \tag{9}$$

Hence, via comparing the obtained result with the symbolic notation in (8), we can easily obtain the values for α and θ. By replacing them in (7), we obtain the current camera position inside η. Using $C_{current} = P_1 + N^T \cdot C^\eta$, we finally obtain the absolute position of the camera.

The two-point algorithm presented delivers a unique solution and is executed in a robust RANSAC-scheme within the VO framework. The cost-function simply consists of the reprojection-error of all matched 3D points from the local cloud. As in Section 3.3, the error function simply uses dot-products between normalized feature vectors on the unit sphere.



(a) (b) (c)

Figure 8: In (a) is a typical image as captured by the MAV. The red (outliers) and yellow (inliers) lines denote matches between the current frame and the last keyframe. The magenta lines represent inlier matches between the last two keyframes. In (b) is the result of the VO for different combinations of feature detector/descriptor, and (c) depicts an intermediate, local point cloud obtained using FAST corners [21] triangulated during VO estimation.

4 Experimental Results

In an attempt to demonstrate the robustness of our method on a challenging scenario, we validated our approach on a dataset captured using a quadrotor MAV (Micro Aerial Vehicle) exhibiting full 6DOF motion with high dynamics. The MAV is typically equipped with an IMU, so we installed an additional downward-looking camera with a field of view of 100°, capturing images at a resolution of 752×480. Compared to handheld camera motion sequences or images taken from a ground or fixed wing aerial vehicle, this setup provides very challenging datasets since the horizontal acceleration of the vehicle can directly be translated into roll and pitch rotations only. Since the methodology presented obtains relative rotation priors from the IMU, our framework is able to robustly cope with such critical motion sequences in contrast to classical vision-only based solutions. In fact, a quantitative comparison between our and a vision-only approach cannot be done due to the fact that the vision-only approach is not able to successfully process the entire dataset. This is not only due to the motion characteristics, but also to structural degeneracies (mostly planar). If a vision-only iteration succeeds, it is in the best case as good as our approach and converging to the same local minimum through the non-linear refinement.

The entire framework has been integrated into ROS (Robotic Operating System) and the interest point detectors/descriptors used for the image processing can seamlessly be configured to any combination. Here, we present results using state-of-the-art methods, namely we use SURF [1], AGAST [18] with subpixel refinement and FAST [21] for the detection in conjunction with the SURF or BRIEF [2] descriptors. In the case where the perspective pose computation fails to find enough inliers, the algorithm is able to switch back automatically to the reinitialization case, avoiding the introduction of inconsistencies/errors in the VO estimation. The experiments have been carried out on a standard 2.8 GHz core.

The dataset we analyze has been captured in a large room ($10\times10\times10\text{m}^3$) equipped with a Vicon motion capture system for ground truth data. We enriched the scene with sparse natural features since the textureless environment is unsuitable for any vision algorithm. Figure 8(a) shows a typical image captured by the flying MAV, demonstrating that the distribution of the extracted features in the image can be very inhomogeneous. However, since the visual

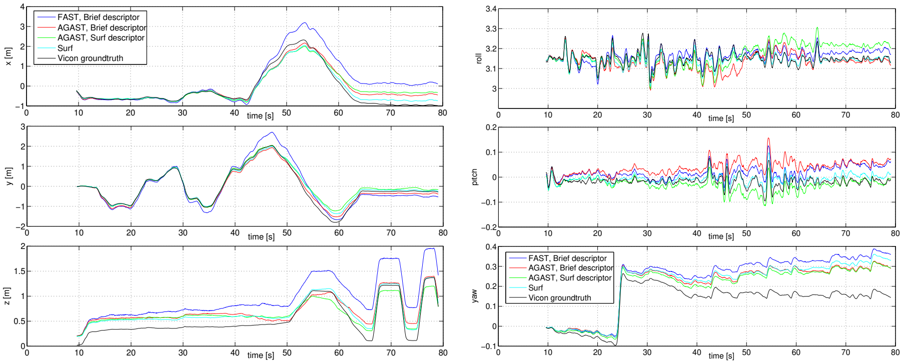


Figure 9: Results for the indoor trajectory.

information is mainly used for estimating the translation, the algorithm is able to robustly work even with such uneven distributions. Figure 8(b) shows the spatial result over the entire dataset for different combinations of feature extraction and description.

Since the VO results are not in metric scale, they have been multiplied by an appropriate factor such that they can be compared to the metric ground truth trajectory. In order to avoid any compensation of scale-drifts – which would disturb the fairness of the comparison –, this factor is derived once, at the beginning of the trajectory. The absolute error at the end of the 22.59m long trajectory amounts to 1.28m for the combination FAST+BRIEF (5.7%), 0.57m for AGAST+BRIEF (2.5%), 0.67m for AGAST+SURF (3%), and 0.26m for SURF+SURF (1.2%). Figure 9 shows the translation and rotation of the camera. As a result, we conclude that the choice of feature detector has the biggest impact on the overall success of operation. In particular, using SURF or AGAST for detection with subpixel accuracy greatly reduces the amount of drift accumulated along the trajectory. The reason for this can also be observed in Figure 8(c), which shows the discretized appearance of the local point cloud when using the FAST keypoint detector. Comparatively, the SURF detector yields much more points than the others, yielding best results in terms of quality.

The ranking of combinations in terms of quality of results is evidently in contrast to the ranking in terms of computational efficiency. As shown in Figure 10, only the FAST or AGAST based solutions are able to run in real-time. The best trade-off between accuracy and efficiency is given by using the AGAST extractor with subpixel refinement in combination with the efficient BRIEF descriptor.

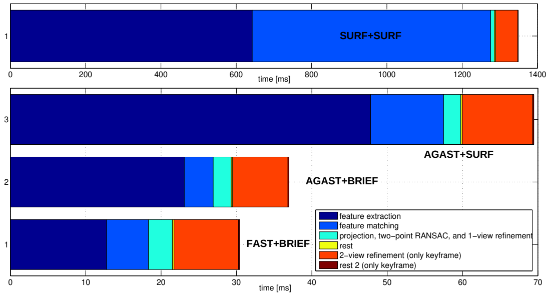


Figure 10: Timings of the different combinations of feature detectors and descriptors.

5 Conclusion

In this paper, we presented a real-time framework for robust Visual Odometry over trajectories with challenging dynamics. Using the relative orientation information from an additional

IMU, the number of points for establishing hypotheses for the relative transformation in between consecutive frames can be reduced to two in any case. The framework selects frames to serve as keyframes, used to triangulate point clouds for perspective pose computation whenever there is sufficient disparity. Disparity measures are always preceded by an unrotation of the normalized feature vectors on the unit sphere, thus avoiding triangulation in the presence of mostly rotational displacement. Our results demonstrate minimal accumulated drift in estimates, presenting a relative assessment of different state-of-the-art feature types.

6 Acknowledgments

This research was supported by the EC's 7th Framework Programe (FP7/2001-2013) under grant agreement no. 231855 (sFly), and by the Swiss National Science Foundation under grant agreement n. 200021 125017/1. We are grateful to Prof. R. D'Andrea and S. Lupashin for providing access and support for the "Flying Machine Arena" [17].

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[3] J. Campbell, R. Sukthankar, and I. Nourbakhsh. Techniques for evaluating optical flow for visual odometry in extreme terrain. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2004.

[4] M. Chli. *Applying Information Theory to Efficient SLAM*. PhD thesis, Imperial College London, 2009.

[5] A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] F. Fraundorfer, P. Tanskanen, and M. Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[8] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2008.

[9] M. Kalantari, A. Hashemi, and F. Jung J.-P. Guedon. A new solution to the relative orientation problem using only 3 points and the vertical direction. *Journal of Mathematical Imaging and Vision (JMIV)*, 39:259–268, 2011.

[10] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[11] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[12] K. Konolige, M. Agrawal, and J. Solà. Large scale visual odometry for rough terrain. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, 2007.

[13] K. Konolige, M. Agrawal, and J. Solá. Large-scale visual odometry for rough terrain. In *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 201–212. 2011.

[14] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to the minimal absolute pose problems with known vertical direction. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2010.

[15] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research (IJRR)*, 26(6):561–575, 2007.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1981.

[17] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[18] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[19] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[20] M. Pollefeys, R. Koch, M. Vergauwen, B. Deknuydt, and L. Van Gool. *Three-dimensional scene reconstruction from images*, volume 3958, pages 215–226. Proceedings of SPIE Electronic Imaging, 2000.

[21] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[22] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2006.

[23] D. Strelow and S. Singh. Motion estimation from image and inertial measurements. *International Journal of Robotics Research (IJRR)*, 23(12):1157, 2004.

[24] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2007.