FDMO: Feature Assisted Direct Monocular Odometry

Georges Younes[1,2], Daniel Asmar[1] and John Zelek[2]
[1]*Mechanical Engineering Department, American Univsersity of Beirut, Beirut, Lebanon*
[2]*Department of Systems Design, University of Waterloo, Waterloo, Canada*

Keywords: Feature-based, Direct, Odometry, Localization, Monocular.

Abstract: Visual Odometry (VO) can be categorized as being either direct (*e.g.* DSO) or feature-based (*e.g.* ORB-SLAM). When the system is calibrated photometrically, and images are captured at high rates, direct methods have been shown to outperform feature-based ones in terms of accuracy and processing time; they are also more robust to failure in feature-deprived environments. On the downside, direct methods rely on heuristic motion models to seed an estimate of camera motion between frames; in the event that these models are violated (e.g., erratic motion), direct methods easily fail. This paper proposes FDMO (Feature assisted Direct Monocular Odometry), a system designed to complement the advantages of both direct and featured based techniques to achieve sub-pixel accuracy, robustness in feature deprived environments, resilience to erratic and large inter-frame motions, all while maintaining a low computational cost at frame-rate. Efficiencies are also introduced to decrease the computational complexity of the feature-based mapping part. FDMO shows an average of 10% reduction in alignment drift, and 12% reduction in rotation drift when compared to the best of both ORB-SLAM and DSO, while achieving significant drift (alignment, rotation & scale) reductions (51%, 61%, 7% respectively) going over the same sequences for a second loop. FDMO is further evaluated on the EuroC dataset and was found to inherit the resilience of feature-based methods to erratic motions, while maintaining the accuracy of direct methods.

1 INTRODUCTION

Visual Odometry (VO) is the process of localizing one or several cameras in an unknown environment. Two decades of extensive research have led to a multitude of VO systems that can be categorized based on the type of information they extract from an image, as direct, feature-based, or a hybrid of both (Younes et al., 2017). While the direct framework manipulates photometric measurements (pixel intensities), the feature-based framework extracts and uses visual features as an intermediate image representation. The choice of feature-based or direct has important ramifications on the performance of the entire VO system, with each type exhibiting its own challenges, advantages, and disadvantages.

One disadvantage of particular interest is the sensitivity of direct methods to their motion model. This limitation is depicted in Fig. 1 (A) and (B), where a direct VO system is subjected to a motion that violates its presumed motion model, and causes it to erroneously expand the map as shown in Fig. 1 (C) and (D). Inspired by the invariance of feature-based methods across relatively large motions (as shown in Fig. 1 (E) and (F)), this paper proposes to address the shortcomings of direct methods, by detecting failure in their frame to frame odometry component, and accordingly invoking an efficient feature-based strategy to cope with large inter-frame motions, hereafter referred to as large baselines. We call our approach Feature assisted Direct Monocular Odometry, or FDMO for short. We show that by effectively exploiting information available from both direct and feature-based frameworks, FDMO considerably improves the robustness of monocular VO by succesfully achieving simultaneously the following properties:

1. Sub-pixel accuracy for the odometry system.
2. Robustness in feature-deprived environments.
3. Low computational cost at frame-rate, and a reduced computational cost for feature-based map optimization.
4. Resilience to erratic and large inter-frame motions.

Younes, G., Asmar, D. and Zelek, J.
FDMO: Feature Assisted Direct Monocular Odometry.
DOI: 10.5220/0007524807370747
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 737-747
ISBN: 978-989-758-354-4

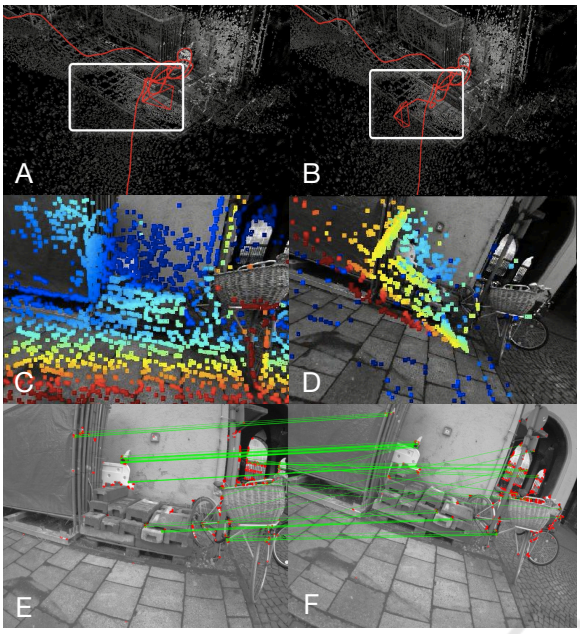

Figure 1: Direct methods failure under large baseline motion. (A) and (B) show the trajectory estimated from a direct odometry system, before and after going through a relatively large baseline between two consecutive frames (shown in (C) and (D). Notice how the camera's pose in (B) derailed from the actual path to a wrong pose. (C) and (D) show the projected direct point cloud on both frames respectively after erroneously estimating their poses. Notice how the projected point cloud is no longer aligned with the scene. On the other hand, (E) and (F) show how features can be matched across the relatively large baseline, allowing feature-based methods to cope with such motions.

2 BACKGROUND

Visual odometry can be broadly categorized as being either direct or feature-based.

2.1 Direct VO

Direct methods process raw pixel intensities with the brightness constancy assumption (Baker and Matthews, 2004):

$$I_t(x) = I_{t-1}(x + g(x)), \quad (1)$$

where x is the 2-dimensional pixel coordinates $(u,v)^T$ and $g(x)$ denotes the displacement function of x between the two images I_t and I_{t-1}. Frame-to-frame tracking is then a byproduct of an image alignment optimization (Baker and Matthews, 2004) that minimizes the photometric residual (intensity difference between the two images) over the geometric transformation that relates them.

2.1.1 Traits of Direct Methods

Since direct methods rely on the entire image for localization, they are less susceptible to failure in feature-deprived environments, and do not require a time-consuming feature extraction and matching step. More importantly, since the alignment takes place at the pixel intensity level, the photometric residuals can be interpolated over the image domain ΩI, resulting in an image alignment with sub-pixel accuracy, and relatively less drift than feature-based odometry methods (Irani and Anandan, 2000). However, the objective function to minimize is highly non-convex; its convergence basin is very small, and will lock to an erroneous configuration if the optimization is not accurately initialized. Most direct methods cope with this limitation by adopting a pyramidal implementation, by assuming small inter-frame motions, and by relying on relatively high frame rate cameras; however, even with a pyramidal implementation that slightly increases the convergence basin, all parameters involved in the optimization should be initialized such that x and $g(x)$ are within 1-2 pixel radii from each other.

2.1.2 State of the Art in Direct Methods

Direct Sparse Odometry (DSO) (Engel et al., 2017) is currently considered the state of the art in direct methods. It is a keyframe-based VO that exploits the small inter-frame motions nature of a video feed to perform a pyramidal implementation of the forward additive image alignment (Baker and Matthews, 2004). DSO's image alignment optimizes a variant of the brightness constancy assumption over the incremental geometric transformation between the current frame and a reference keyframe. The aligned patches are then used to update the depth estimates for each point of interest as described in (Engel et al., 2013).

2.2 Feature-based VO

Feature-based methods process 2D images to extract locations that are salient in an image. Let $x = (u,v)^T$ represent a feature's pixel coordinates in the 2-dimensional image domain ΩI. Associated with each feature is an n-dimensional vector $Q^n(x)$, known as a *descriptor*. The set $\Phi I\{x, Q(x)\}$ is an intermediate image representation after which the image itself becomes obsolete and is discarded.

2.2.1 Traits of Feature-based Methods

On the positive side, features with their associated descriptors are somewhat invariant to viewpoint and

illumination changes, such that a feature $x \in \Phi I_1$ in one image can be identified as $x' \in \Phi I_2$ in another, across *relatively large illumination and motion baselines*. However, the robustness of the data association relies on the distinctiveness of each feature from the other, a condition that becomes more difficult to satisfy the higher the number of features extracted in each scene; thereby favouring sparse, versus dense, feature representations. On the downside, and as a result of their discretized image representation space, feature-based solutions offer inferior accuracy when compared to *direct* methods, as the image domain cannot be interpolated for sub-pixel accuracy.

2.2.2 State of the Art in Feature-based Methods

ORB-SLAM (Mur-Artal et al., 2015), currently considered the state of the art in feature-based methods, associates FAST corners (Rosten and Drummond, 2006) with ORB descriptors (Rublee et al., 2011) as an intermediate image representation. Regular frames are localized by minimizing the traditional geometric re-projection error; the 3D points are triangulated using Epipolar geometry (Hartley and Zisserman, 2003), from multiple observations of the feature $\{x_i, Q(x_i)\}$ in two or more keyframes. The consistency of the map is maintained through a local bundle adjustment minimization. Both, localization and mapping optimizations are resilient to relatively large inter-frame baseline motions and have a relatively large convergence radius. To further increase its performance and cut down processing time, ORB-SLAM resorts to various methods for data association such as the covisibility graph (Strasdat et al., 2011) and bag of visual words. (Galvez-López and Tardos, 2012).

3 RELATED WORK

When the corresponding pros and cons of both feature-based and direct frameworks are placed side by side, a pattern of complementary traits emerges (Table 1). An ideal framework would exploit both direct and feature-based advantages to benefit from the direct formulation accuracy and robustness to feature-deprived scenes, while making use of feature-based methods for large baseline motions.

In an attempt to achieve the aforementioned properties, hybrid direct-feature-based systems were previously proposed in (Forster et al., 2014), (Krombach et al., 2016) and (Ait-Jellal and Zell, 2017); however, (Forster et al., 2014) did not extract feature descriptors, it relied on the direct image alignment to perform data association between the features. While this led to significant speed-ups in the processing required for data association, it could not handle large baseline motions; as a result, their work was limited to high frame rate cameras, which ensured frame-to-frame motion is small. On the other hand, both (Krombach et al., 2016) and (Ait-Jellal and Zell, 2017) adopted a feature-based approach as a front-end to their system, and subsequently optimized the measurements with a direct image alignment; as such, both systems suffer from the limitations of the feature-based framework, *i.e.* they are subject to failure in feature-deprived environments and therefore not able to simultaneously meet all of the desired traits of Table 1. To address this issue, both systems resorted to stereo cameras.

Table 1: Comparison between the feature-based and direct methods. The more of the symbol +, the higher the attribute.

Trait	Feature-based	Direct
Large baseline	+++	+
Robust to Feature Deprivation	+	+++
Recovered scene point density	+	+++
Accuracy	+	+++
Optimization Non-Convexity	+	++

In contrast to these systems, FDMO can operate using a monocular camera, and simultaneously achieve all of the desired traits. FDMO can also be adapted for stereo and RGBD cameras as well. FDMO's source code will be made publicly available on this URL upon the acceptance of this work.

4 PROPOSED SYSTEM

To capitalize on the advantages of both feature-based and direct frameworks, our proposed approach consists of a local direct visual odometry, assisted with a feature-based map, such that it may resort to feature-based odometry only when necessary. Therefore, FDMO does not need to perform a computationally expensive feature extraction and matching step at every frame. During its feature-based map expansion, FDMO exploits the localized keyframes with sub-pixel accuracy from the direct framework, to efficiently establish feature matches in feature-deprived environments using restricted epipolar search lines.

Similar to DSO, FDMO's local temporary map is defined by a set of seven direct-based keyframes and 2000 active direct points. Increasing these parameters was found by (Engel et al., 2017) to significantly increase the computational cost without much impro-

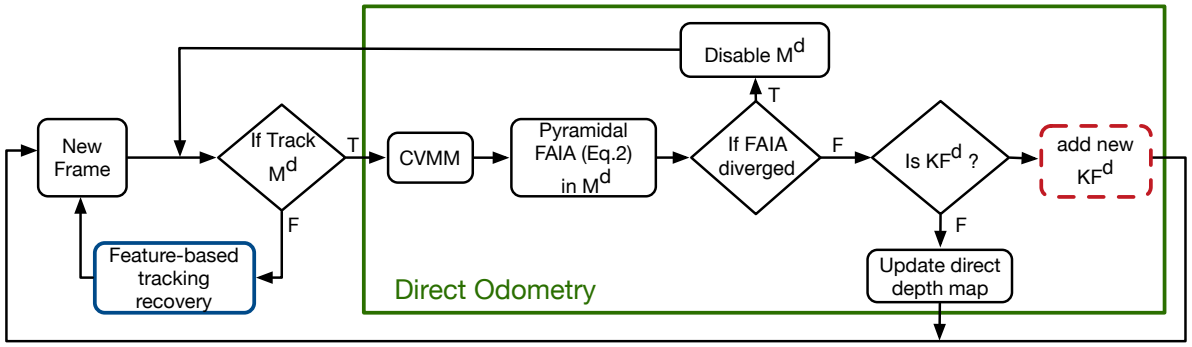

Figure 2: Front-end flowchart of FDMO. It runs on a frame by frame basis and uses a constant velocity motion model (CVMM) to seed a Forward Additive Image Alignment (FAIA) to estimate the new frame's pose and update the direct map depth values. It also decides whether to invoke the feature-based tracking or add a new keyframe into the system. The blue (solid line) and red (dashed line) boxes are further expanded in figures 3 and 4 respectively.

vement in accuracy. Direct keyframe insertion and marginalization occurs frequently according to conditions described in (Engel et al., 2017). In contrast, the feature-based map is made of an undetermined number of keyframes, each with an associated set of features and their corresponding ORB descriptors $\Phi(x, Q(x))$.

4.1 Notation

To address any ambiguities, the superscript d will be assigned to all direct-based measurements and f for all feature-based measurements; not to be confused with underscript f assigned to the word frame. Therefore, M^d refers to the temporary direct map, and M^f to the feature-based map, which is made of an unrestricted number of keyframes κ^f and a set of 3D points X^f. I_{f_i} refers to the image of frame i and T_{f_i,KF^d} is the $se(3)$ transformation relating frame i to the latest active keyframe KF in the direct map. We also make the distinction between z referring to depth measurements associated with a 2D point x and Z refering to the Z coordinate of a 3D point.

4.2 Odometry

4.2.1 Direct Image Alignment

Frame by frame operations are handled by the flowchart described in Fig. 2. Similar to (Engel et al., 2017), newly acquired frames are tracked by minimizing

$$\underset{T_{f_i,KF^d}}{\operatorname{argmin}} \sum_{x^d} \sum_{x \in N(x^d)} Obj(I_{f_i}(\omega(x, z, T_{f_i,KF^d}) - I_{KF^d}(x, z))), \quad (2)$$

where f_i is the current frame, KF^d is the latest added keyframe in M^d, $x^d \in \Omega I_f$ is the set of image locations with sufficient intensity gradient and an associated depth value d. $N(x^d)$ is the set of pixels neighbouring x^d and $w(\cdot)$ is the projection function that maps a 2D point from f_i to KF^d.

The minimization is seeded from a constant velocity motion model (CVMM). However, erratic motion or large motion baselines can easily violate the CVMM, erroneously initializing the highly-non convex optimization, and yielding unrecoverable tracking failure. We detect tracking failure by monitoring the RMSE of Eq. (2) before and after the optimization; if the ratio $\frac{RMSE_{after}}{RMSE_{before}} > 1 + \varepsilon$ we consider that the optimization has diverged and we invoke the feature-based tracking recovery, summarized in the flowchart of Fig. 3. The ε is used to restrict feature-based intervention when the original motion model used is accurate, a value of $\varepsilon = 0.1$ was found as a good trade-off between continuously invoking the feature-based tracking and not detecting failure in the optimization. To avoid extra computational cost, feature extraction and matching is not performed on a frame by frame basis, it is only invoked during feature-based tracking recovery and feature-based KF insertion.

4.2.2 Feature-based Tracking Recovery

Our proposed feature-based tracking operates in M^f. When direct tracking diverges, we consider the CVMM estimate to be invalid and seek to estimate a new motion model using the feature-based map. Our proposed feature-based tracking recovery is a variant of the global re-localization method proposed in (Mur-Artal et al., 2015); we first start by detecting $\Phi f_i = \Phi(x^f, Q(x^f))$ in the current image, which are then parsed into a vocabulary tree. Since we consider the CVMM to be invalid, we fall back on the last piece of information the system was sure of before failure: the pose of the last successfully added keyframe. We define a set κ^f of feature-based keyframes KF^f con-

nected to the last added keyframe KF_d through a covisibility graph (Strasdat et al., 2011), and their associated 3D map points X^f.

Blind feature matching is then performed between Φf_i and all keyframes in κ^f, by restricting feature matching to take place between features that exist in the same node in a vocabulary tree (Galvez-López and Tardos, 2012); this is done to reduce the computational cost of blindly matching all features.

Once data association is established between f_i and the map points, we set up an EPnP (Efficient Perspective-n-Point Camera Pose Estimation) (Lepetit et al., 2009) to solve for an initial pose T_{f_i} using 3D-2D correspondences in an non-iterative manner. The new pose is then used to define a 5×5 search window in f_i surrounding the projected locations of all 3D map points $X^f \in \kappa^f$. Finally the pose T_{f_i} is refined through the traditional feature-based optimization. To achieve sub-pixel accuracy, the recovered pose T_{f_i} is then converted into a local increment over the pose of the last active direct keyframe, and then further refined in a direct image alignment optimization Eq. (2).

Note that the EPnP step could have been skipped in favour of using the last correctly tracked keyframe's position as a starting point; however, data association would then require a relatively larger search window, which in turn increases its computational burden in the subsequent step. Data association using a search window was also found to fail when the baseline motion was relatively large.

4.3 Mapping

FDMO's mapping process is composed of two components: direct, and feature-based as described in Fig. 4. The direct map propagation used here is the same as suggested in (Engel et al., 2017); however we expand its capabilities to propagate the feature-based map. When a new keyframe is added to M^d, we create a new feature-based keyframe KF^f that inherits its pose from KF^d. $\Phi KF^f(x^f, Q(x^f))$ is then extracted and data association takes place between the new keyframe and a set of local keyframes κ^f surrounding it via epipolar search lines. The data association is used to keep track of all map points X^f visible in the new keyframe and to triangulate new map points.

To ensure an accurate and reliable feature-based map, typical feature-based methods employ local bundle adjustment (LBA)(Mouragnon et al., 2006) to optimize for both the keyframes poses and their associated map points; however, employing an LBA may generate inconsistencies between both map representations, and is computationally expensive; instead, we make use of the fact that the new keyframe's pose is already locally optimal, to replace the typical local bundle adjustment with a computationally less demanding structure only optimization defined for each 3D point X_j^f:

$$\underset{X_j^f}{\operatorname{argmin}} \sum_{i \in \kappa^f} Obj(x_{i,j}^f - \pi(T_{KF_i^f} X_j^f)), \tag{3}$$

where X_j spans all 3D map points observed in all keyframes $\in \kappa^f$. We use ten iterations of Gauss-Newton to minimize the normal equations associated with Eq. (3) which yield the following update rule per 3D point X_j per iteration:

$$X_j^{t+1} = X_j^t - (J^T W J)^{-1} J^T W e \tag{4}$$

Where e is the stacked reprojection residuals e_i associated with a point X_j and its found match x_i in keyframe i. J is the stacked Jacobians of the reprojection error which is found by stacking:

$$J_i = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} \end{bmatrix} R_{KF_i} \tag{5}$$

and R_{KF_i} is the 3×3 orientation matrix of the keyframe observing the 3D point X_j. Similar to ORB-SLAM, W is a block diagonal weight matrix that down-weighs the effect of residuals computed from feature matches found at high pyramidal levels[1] and is computed as

$$W_{ii} = \begin{bmatrix} Sf^{2n} & 0 \\ 0 & Sf^{2n} \end{bmatrix} \tag{6}$$

where Sf is the scale factor used to generate the pyramidal representation of the keyframe (we use $Sf = 1.2$) and n is the pyramid level from which the feature was extracted ($0 < n < 8$). The Huber norm is also used to detect and remove outliers. We have limited the number of iterations in the optimization of Eq. (3) to ten, since no significant reduction in the feature-based re-projection error was recorded beyond that.

4.4 Feature-based Map Maintenance

To ensure a reliable feature-based map, the following practices are employed. For proper operation, direct methods require frequent addition of keyframes, resulting in small baselines between the keyframes, which in turn can cause degeneracies if used to triangulate feature-based points. To avoid numerical instabilities, following the suggestion of (Klein and Murray, 2007), we prevent feature triangulation between

[1] Features matched at higher pyramidal levels are less reliable.

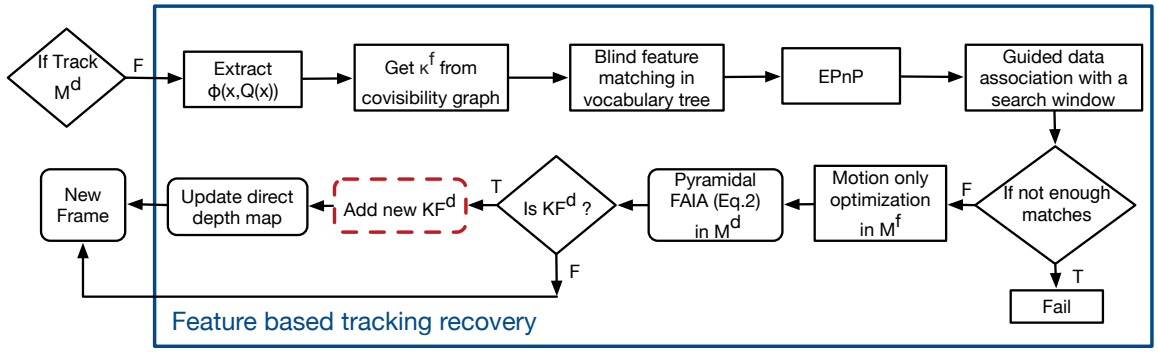

Figure 3: FDMO Tracking Recovery flowchart. Only invoked when direct image alignment fails, it takes over the front end operations of the system until the direct map is re-initialized. FDMO's tracking recovery is a variant of ORB-SLAM's global failure recovery that exploits the information available from the direct framework to constrain the recovery procedure locally. We start by extracting features from the new frame and matching them to 3D features observed in a set of keyframes κ^f connected to the last correctly added keyframe from KF^d. Efficient Perspective-n-Point (EPnP) camera pose estimation is used to estimate an initial guess which is then refined by a guided data association between the local map and the frame. The refined pose is then used to seed a Forward additive image alignment step to achieve sub-pixel accuracy.

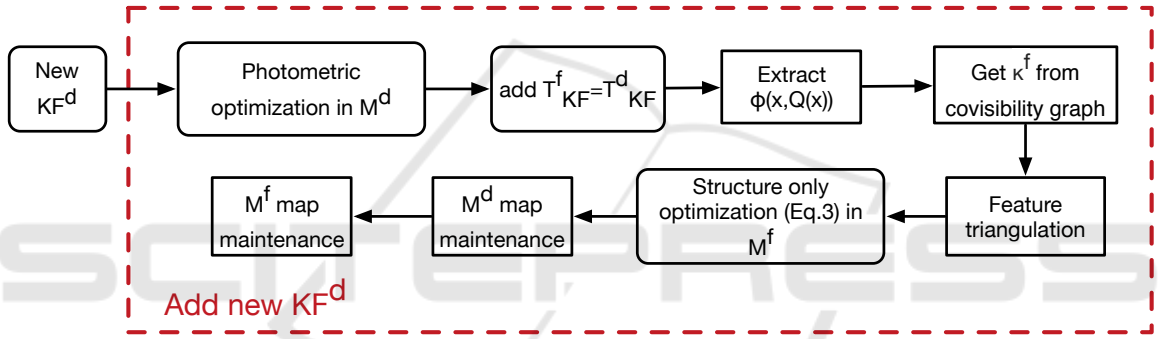

Figure 4: Our proposed mapping flowchart, is a variant of DSO's mapping backend; we augment its capabilities to expand the feature-based map with new KF^f. It operates after or parallel to the direct photometric optimization of DSO, by first establishing feature matches using restricted epipolar search lines; the 3D feature-based map is then optimized using a computationally efficient *structure-only* bundle adjustment, before map maintenance ensures the map remain outliers free.

keyframes with a $\frac{baseline}{depth}$ ratio less than 0.02 which is a trade-off between numerically unstable triangulated features and feature deprivation problems. We exploit the frequent addition of keyframes as a feature quality check. In other words, a feature has to be correctly found in at least 4 of the 7 keyframes subsequent to the keyframe it was first observed in, otherwise it is considered spurious and is subsequently removed. To ensure no feature deprivation occurs, a feature cannot be removed until at least 7 keyframes have been added since it was first observed. Finally, similar to (Mur-Artal et al., 2015) a keyframe with ninety percent of its points shared with other keyframes is removed from M^f only once marginalized from M^d.

The aforementioned practices ensure that sufficient reliable map points and features are available in the immediate surrounding of the current frame, and that only necessary map points and keyframes are kept once the camera moves on.

5 EXPERIMENTS AND RESULTS

To evaluate FDMO's tracking robustness, experiments were performed on several well-known datasets (Burri et al., 2016) and (Engel et al., 2016), and both qualitative and quantitative appraisal was conducted. To further validate FDMO's effectiveness, the experiments were also repeated on state of the art open-source systems in both direct (DSO) and feature-based (ORB-SLAM2). For fairness of comparison, we evaluate ORB-SLAM2 as an odometry system (not as a SLAM system); therefore, similar to (Engel et al., 2017) we disable its loop closure thread but we keep its global failure recovery, local, and global bundle adjustments intact. Note that we've also attempted to include results from SVO (Forster et al., 2014) but it continuously failed on most datasets, so we excluded it.

5.1 Datasets

5.1.1 TUM MONO Dataset

(Engel et al., 2016) Contains 50 sequences of a camera moving along a path that begins and ends at the same location. The dataset is photometrically calibrated: camera response function, exposure times and vignetting are all available; however, ground truth pose information is only available for two small segments at the beginning and end of each sequence; fortunately, such information is enough to compute translation, rotation, and scale drifts accumulated over the path, as described in (Engel et al., 2016).

5.1.2 EuRoC MAV Dataset

(Burri et al., 2016) Contains 11 sequences of stereo images recorded by a drone mounted camera. Ground truth pose for each frame is available from a Vicon motion capture system.

5.2 Computational Cost

The experiments were conducted on an Intel Core i7-4710HQ 2.5GHZ CPU, 16 GB memory; no GPU acceleration was used. The time required by each of the processes was recorded and summarized in Table 2. Both DSO and ORB-SLAM2 consist of two parallel components, a tracking process (at frame-rate[2]) and a mapping process (keyframe-rate[3]). On the other hand, FDMO has three main processes: a direct tracking process (frame-rate), a direct mapping process (keyframe-rate), and a feature-based mapping process (keyframe-rate). Both of FDMO's mapping processes can run either sequentially for a total computational cost of 200 ms on a single thread, or in parallel on two threads. As Table 2 shows, the mean tracking time for FDMO remains almost the same that of DSO: we don't extract features at frame-rate; feature-based tracking in FDMO is only performed when the direct tracking diverges; the extra time is reflected in the slightly increased standard deviation of the computational time with respect to DSO. Nevertheless, it is considerably less than ORB-SLAM2's 23 ms. The highest computational cost during FDMO tracking occurs when the recovery method is invoked, with a highest recorded processing time during our experiments of 35 ms. As for FDMO's mapping processes, its direct part remains the same as DSO, whereas the feature-based part takes 153 ms which is also significantly less than ORB-SLAM2's feature-based mapping process that requires 236 ms.

[2]occur at every frame.

[3]occur at new keyframes only.

Table 2: Computational time (ms) for processes in DSO, FDMO and ORB-SLAM2. (Empty means the process does not exist).

Process	DSO	FDMO	ORB-SLAM2
Tracking (frame-rate)	12.35± 9.62	13.54± 14.19	23.04± 4.11
Direct mapping (Keyframe-rate)	46.94± 51.62	46.89± 65.21	—
Feature-based mapping (Keyframe-rate)	—	153.8± 58.08	236.47± 101.8

5.3 Quantitative Results

We assess FDMO, ORB-SLAM2 and DSO using the following experiments.

5.3.1 Two Loop Experiment

In this experiment, we investigate the quality of the estimated trajectory by comparing ORB-SLAM2, DSO, and FDMO. We allow all three systems to run on various sequences of the Tum_Mono dataset (Engel et al., 2016) across various conditions, both indoors and outdoors. Each system is allowed to run through every sequence for two continuous loops where each sequence begins and ends at the same location. We record the positional, rotational, and scale drifts at the end of each loop, as described in (Engel et al., 2016). The drifts recorded at the end of the first loop are indicative of the system's performance across the unmodified generic datasets, whereas the drifts recorded at the end of the second loop consist of three components: (1) the drift accumulated from the first loop, (2) an added drift accumulated over the second run, and (3) an error caused by a large baseline motion induced at the transition between the loops. The reported results are shown in Table 3 and some of the recovered trajectories are shown in Fig. 5.

5.3.2 Frame Drop Experiment

While the first experiment reports on the system's performance across large scale scenes in various conditions, this experiment investigates the effects erratic and large baseline motions have on the camera's tracking accuracy. Erratic motion can be defined as a sudden acceleration in the opposite direction of

Table 3: Measured drifts after finishing one and two loops over various sequences from the TumMono dataset. The alignment drift (meters), rotation drift (degrees) and scale($\frac{m}{m}$) drifts are computed similar to (Engel et al., 2016).

		Sequence 20		Sequence 25		Sequence 30		Sequence 35		Sequence 40		Sequence 45		Sequence 50	
		Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2	Loop 1	Loop 2
Alignment	FDMO	**0.752**	**1.434**	**0.863**	**1.762**	**0.489**	**1.045**	**0.932**	2.854	**2.216**	**4.018**	**1.344**	**2.973**	**1.504**	**2.936**
	DSO	0.847	—	0.89	3.269	0.728	5.344	0.945	—	2.266	4.251	1.402	8.702	1.813	—
	ORB SLAM	4.096	8.025	3.722	8.042	2.688	4.86	1.431	**2.846**	—	—	8.026	12.69	6.72	13.56
Rotation	FDMO	**1.4**	**1.192**	**1.154**	**2.074**	0.306	**0.317**	**1.425**	**6.246**	**3.877**	**6.524**	0.522	**5.595**	**0.448**	**1.062**
	DSO	1.607	—	1.278	7.699	**0.283**	18.9	2.22	—	4.953	19.89	**0.462**	23.17	0.594	—
	ORB SLAM	26.92	53.28	2.373	4.647	2.982	4.549	3.676	6.498	—	—	3.707	7.375	3.243	6.668
Scale	FDMO	1.079	1.161	**1.113**	**1.238**	**1.033**	**1.071**	1.072	1.211	**1.109**	**1.219**	**1.082**	1.106	**1.107**	**1.224**
	DSO	1.089	—	1.116	1.424	1.045	1.109	**1.067**	—	1.118	1.226	1.084	**1.023**	1.133	—
	ORB SLAM	**1.009**	**1.019**	1.564	2.403	1.199	1.373	1.094	**1.206**	—	—	1.867	2.574	1.7	2.675

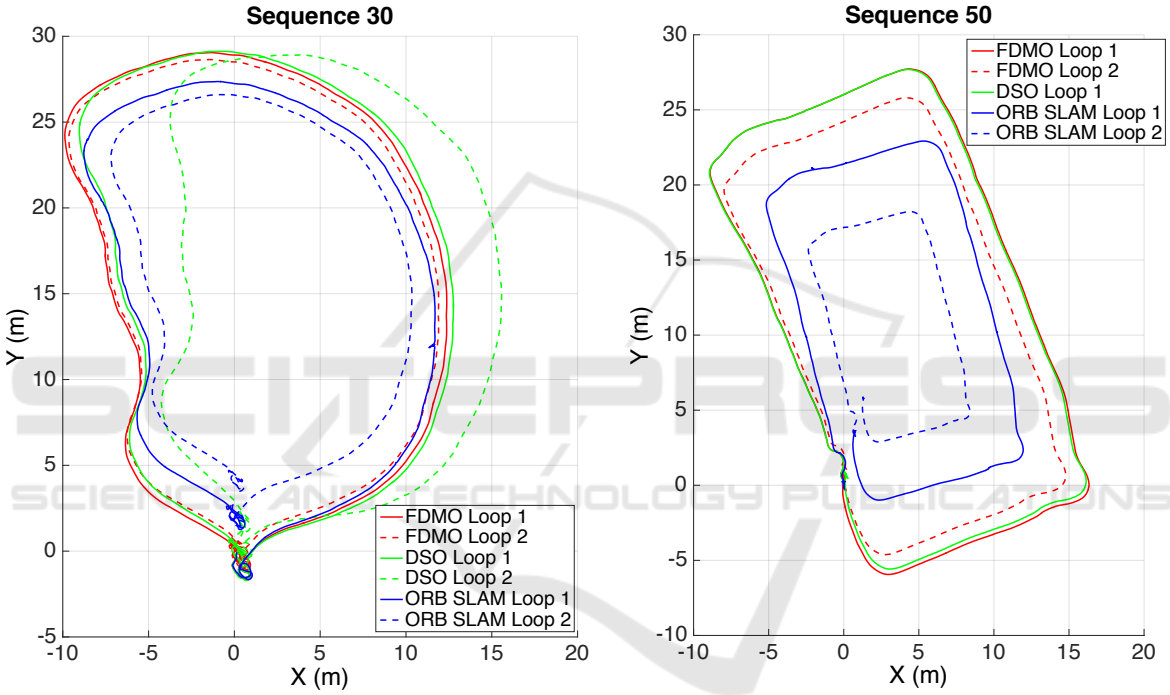

Figure 5: Sample paths estimated by the various systems on Sequences 30 and 50 of the Tum_Mono dataset. The paths are all aligned using ground truths available at the beginning and end of each loop. Each solid line corresponds to the first loop of a system while the dashed line correspond to the second loop. Ideally, all systems would start and end at the same location, while reporting the same trajectories across the two loops. Note that in Sequence 50, there is no second loop for DSO as it was not capable of dealing with the large baseline between the loops and failed.

motion, and is quite common in hand-held devices or quad-copters. Another example of erratic motion occurs when the camera's video feed is being transmitted over a network to a ground station where computation is taking place; communication issues may cause frame drops which are seen by the odometry system as large baseline motions; therefore it is imperative for an odometry system to cope with such motions. To quantize the influence of erratic motions on an odometry system, we set up an experiment to emulate their effects, by dropping frames and measuring the recovered poses before and after dropping them. The experiment is repeated at the same location and the number of frames dropped is increased by five frames each time until each system fails. Various factors can affect the obtained results, such as the distance to the observed scene, skipping frames towards a previously observed or unobserved scene, and/or the type of camera motion (*i.e.*, sideways, forward moving, or rotational motion), to name a few. Therefore we repeat the above experiment for each system in various locations covering the above scenarios. We chose to perform the experiments on the EuroC dataset (Burri et al., 2016) whose frame to frame ground truth is

known; thus allowing us to compute the relative Euclidean distance $Translation = ||F_i - F_j||$, and the orientation difference between the recovered poses at F_i and F_j as the *geodesic metric of the normalized quaternions on the unit sphere* defined by $Rotation = cos^{-1}(2|F_i \cdot F_j|^2 - 1)$. We report on the percent error $\%Error = 100 \times \frac{|Measured - GroundTruth|}{GroundTruth}$ for the recovered Euclidean distance and relative orientation before and after the skipped frames. The obtained results for FDMO, DSO and ORB-SLAM2 are shown in Fig. 6.

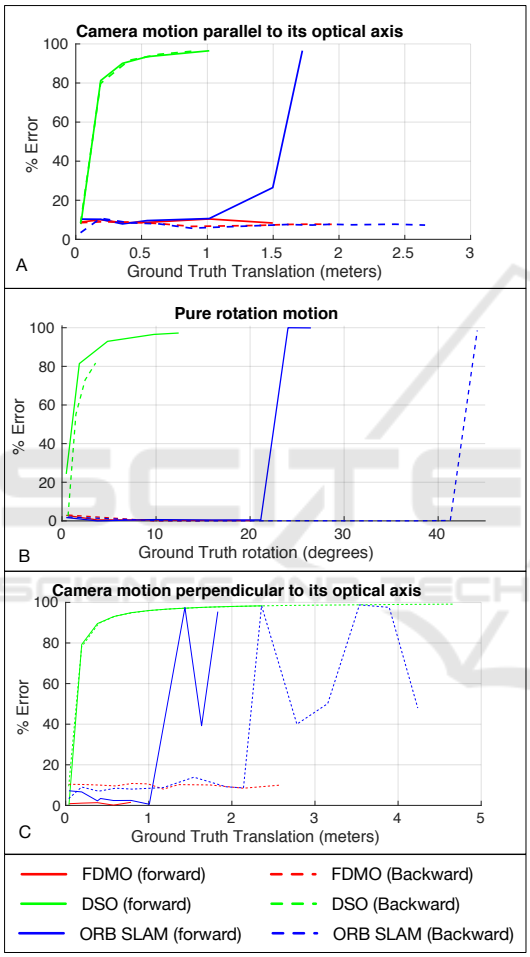

Figure 6: $\%Error$ v.s. ground truth motion measured by dropping frames and estimating the relative rotation and translation before and after the frames were dropped. After every measurement, the system is restarted and the number of dropped frames is increased/decreased by 5 frames for forward and backward jumps respectively, until failure occurs. The experiments were conducted on various sequences of the EuroC dataset (Burri et al., 2016): Experiment (A) in the sequence MH01 with *forward* starting at frame 200 and *backward* starting at 250; Experiment (B) in the sequence MH02 with *forward* starting at frame 510 and *backward* at 560. Experiment (C) in the sequence MH03 with *forward* starting at frame 950 and *backward* starting at frame 1135.

5.4 Qualitative Assessment

Fig. 7 compares the resilience of FDMO and ORB-SLAM2 to feature-deprived environments. FDMO exploits the sub-pixel accurate localized direct keyframes to propagate its feature-based map, therefore its capable of generating accurate and robust 3D landmarks that have a higher matching rate even in low textured environments. In contrast, ORB-SLAM2 fails to propagate its map causing tracking failure.

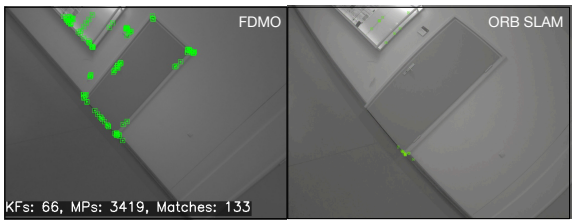

Figure 7: Features matched of FDMO (left) and ORB-SLAM2 (right) in a feature deprived environment (sequence 40 of the Tum_mono dataset).

5.5 Discussion

The results reported in the first experiment (Table. 3) demonstrate FDMO's performance in large-scale indoor and outdoor environments. The importance of the problem FDMO attempts to address is highlighted by analyzing the drifts incurred at the end of the first loop; while no artificial erratic motions nor large baselines were introduced over the first loop, i.e. unmodified dataset, FDMO was able to outperform the best of either DSO and ORB-SLAM2 in terms of positional and rotational drifts, by an average of 10% and 12% respectively on most sequences. The improved performance is due to FDMO's ability to detect and account for inaccuracies in the direct framework using its feature-based map, while benefiting from the sub-pixel accuracy of the direct framework. Furthermore, FDMO was capable of expanding both its direct and feature-based maps in feature-deprived environments (*e.g.* Sequence 40) whereas ORB-SLAM2 failed to do so. FDMO's robustness is further proven by analyzing the results obtained over the second loop. The drifts accumulated toward the end of the second loop are made of three components; mainly, the drift that occurred over the first loop, the drift that occurred over the second, and an error caused by a large baseline separating the frames at the transition between the loops. If the error caused by the large baseline is negligible, we would expect the drift at the second loop to be double that of the first. While the measured drifts for both ORB-SLAM2 and FDMO does indeed exhibit such behaviour, the drifts reported by ORB-

SLAM2 are significantly larger than the ones reported by FDMO as Fig. 5 also highlights. On the other hand, DSO tracking failed entirely on various occasions, and when it did not fail, it reported a significantly large increase in drifts over the second loop. As DSO went through the transition frames between the loops, its motion model estimate was violated, erroneously initializing its highly non-convex tracking optimization. The optimization got subsequently stuck in a local minimum, which led to a wrong pose estimate. The wrong pose estimate was in turn used to propagate the map, thereby causing large drifts. On the other hand, FDMO was successfully capable of handling such a scenario, reporting an average improvement of 51%, 61% and 7 % in positional, rotational, and scale drifts respectively, when compared to the best of both DSO and ORB-SLAM2, on most sequences.

The results reported in the second experiment (Fig. 6) quantify the robustness limits of each system to erratic motions. Various factors may affect the obtained results, therefore, we attempted the experiments under various types of motion and by skipping frames towards a previously observed (herein referred to as backward) and previously unobserved part of the scene (referred to as forward). The observed depth of the scene is also an important factor: far-away scenes remain for a longer time in the field of view, thus improving the systems' performance. However, we cannot model all different possibilities of depth variations; therefore, for the sake of comparison, all systems were subjected to the same frame drops at the same locations in each experiment where the observed scene's depth varied from three to eight meters. The reported results highlight DSO's brittleness to any violation of its motion model; where translations as little as thirty centimeters and rotations as small as three degrees introduced errors of over 50% in its pose estimates. On the other hand, FDMO was capable of accurately handling baselines as large as 1.5 meters and 20 degrees towards previously unobserved scene, after which failure occurred due to feature-deprivation, and two meters toward previously observed parts of the scene. ORB-SLAM2's performance was very similar to FDMO in forward jumps, however it significantly outperformed it by twice as much in the backward jumps; ORB-SLAM2 uses a global map for failure recovery whereas FDMO, being an odometry system, can only make use of its immediate surroundings. Nevertheless FDMO's current limitations in this regard are purely due to our current implementation as there are no theoretical limitations of developing FDMO into a full SLAM system. However, using a global relocalization method has its downside; the jitter in ORB-SLAM2's behaviour (shown in Fig. 6 (C)) is due to its relocalization process erroneously localizing the frame at spurious locations. Another key aspect of FDMO, visible in this experiment, is its ability to detect failure and not incorporate it into its map. In contrast, toward their failure limits, both DSO and ORB-SLAM2 incorporate spurious measurements for few frames before failing completely.

6 CONCLUSION

This paper successfully demonstrated the advantages of integrating direct and feature-based methods in VO. By relying on a feature-based map when direct tracking fails, the issue of large baselines that is characteristic of direct methods is mitigated, while maintaining the high accuracy and robustness to feature-deprived environments of direct methods in both feature-based and direct maps, at a relatively low computational cost. Both qualitative and quantitative experimental results proved the effectiveness of the collaboration between direct and feature-based methods in the localization part.

While these results are exciting, they do not make use of a global feature-based map; as such we are currently developing a more elaborate integration between both frameworks, to further improve the mapping accuracy and efficiency. Furthermore, we anticipate that the benefits to the mapping thread will also lead to added robustness and accuracy to the motion estimation within a full SLAM framework.

ACKNOWLEDGEMENTS

This work was funded by the University Research Board (UBR) at the American University of Beirut, and the Canadian National Science Research Council (NSERC).

REFERENCES

Ait-Jellal, R. and Zell, A. (2017). Outdoor obstacle avoidance based on hybrid stereo visual slam for an autonomous quadrotor mav. In *IEEE 8th European Conference on Mobile Robots (ECMR)*.

Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., and Siegwart, R. (2016). The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*.

Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1.

Engel, J., Sturm, J., and Cremers, D. (2013). Semi-dense Visual Odometry for a Monocular Camera. In *Computer Vision (ICCV), IEEE International Conference on*, pages 1449–1456. IEEE.

Engel, J., Usenko, V., and Cremers, D. (2016). A photometrically calibrated benchmark for monocular visual odometry. In *arXiv:1607.02555*.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). Svo : Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), IEEE International Conference on*.

Galvez-López, D. and Tardos, J. D. (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. *Robotics, IEEE Transactions on*, 28(5):1188–1197.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Irani, M. and Anandan, P. (2000). About direct methods. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 267–277, London, UK, UK. Springer-Verlag.

Klein, G. and Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10.

Krombach, N., Droeschel, D., and Behnke, S. (2016). Combining feature-based and direct methods for semi-dense real-time stereo visual odometry. In *International Conference on Intelligent Autonomous Systems*, pages 855–868. Springer.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155–166.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006). Real time localization and 3d reconstruction. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 363–370.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, PP(99):1–17.

Rosten, E. and Drummond, T. (2006). Machine Learning for High-speed Corner Detection. In *9th European Conference on Computer Vision - Volume Part I, Proceedings of the*, ECCV'06, pages 430–443, Berlin, Heidelberg. Springer-Verlag.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571.

Strasdat, H., Davison, A. J., Montiel, J. M. M., and Konolige, K. (2011). Double Window Optimisation for Constant Time Visual SLAM. In *International Conference on Computer Vision, Proceedings of the*, ICCV '11, pages 2352–2359, Washington, DC, USA. IEEE Computer Society.

Younes, G., Asmar, D., Shammas, E., and Zelek, J. (2017). Keyframe-based monocular slam: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:67 – 88.