2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Towards Noise Resilient SLAM

Anirud Thyagharajan[1], Om Ji Omer[1], Dipan Mandal[1], Sreenivas Subramoney[1]

***Abstract*— Sparse-indirect SLAM systems have been dominantly popular due to their computational efficiency and photometric invariance properties. Depth sensors are critical to SLAM frameworks for providing scale information to the 3D world, yet known to be plagued by a wide variety of noise sources, possessing lateral and axial components. In this work, we demonstrate the detrimental impact of these depth noise components on the performance of the state-of-the-art sparse-indirect SLAM system (ORB-SLAM2). We propose (i) Map-Point Consensus based Outlier Rejection (MC-OR) to counter lateral noise, and (ii) Adaptive Virtual Camera (AVC) to combat axial noise accurately. MC-OR utilizes consensus information between multiple sightings of the same landmark to disambiguate noisy depth and filter it out before pose optimization. In AVC, we introduce an error vector as an accurate representation of the axial depth error. We additionally propose an adaptive algorithm to find the virtual camera location for projecting the error used in the objective function of the pose optimization. Our techniques work equally well for stereo image pairs and RGB-D input directly used by sparse-indirect SLAM systems. Our methods were tested on the TUM (RGB-D) and EuRoC (stereo) datasets and we show that they outperform existing state-of-the-art ORB-SLAM2 by 2-3x, especially in sequences critically affected by depth noise.**

I. Introduction

SLAM is a fundamental building block in several mobile autonomous systems [9], [1], [3]. SLAM methods are classified [3], [13] as either (i) sparse/dense, indicating whether a select set of points or all the pixels in the image are used for processing and (ii) direct/indirect, indicating whether the system works directly on the measured quantities or not. Beyond being computationally economical than their dense counterparts [27], sparse methods have evolved over the years to be more robust and accurate as well. In addition, dense implementations have inherent bias due to geometric priors and assumptions, leading to reduced long-term accuracy [13]. Eventhough direct approaches do not need feature extraction and can be useful for denser reconstructions as well, they suffer from artifacts created by assuming a surface reflectance model. These are further impacted by rolling shutter, auto gain and auto-exposure artifacts if not modeled properly [27], [13]. Also, [44] highlighted that feature-based methods demonstrate a wide range of photometric and geometric invariance as compared to direct methods. Due to these reasons, sparse-indirect SLAM implementations have been adopted widely [18], [8], [21], [28].

While information arising from multiple diverse sensor modalities can be fused together [26], [24], [32], [25] to provide the required estimates, there is a rising trend to utilize only cameras. Cameras prove to be cost-effective for the rich source of information that they provide, with RGB-D cameras providing depth estimates as well. While there are multiple commodity RGB-D sensors available, they are impacted by various noise issues caused by the measurement setup and the surface properties [20], [31]. Though generic outlier removal based SLAM systems have been proposed [46], [4], [17], [34], [33], [41], [42], they either require significant compute or are not robust enough in dealing with the noise.

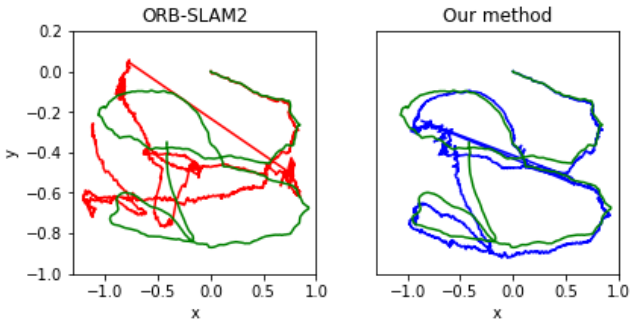

Fig. 1: A comparison of trajectory plots between state-of-the-art ORB-SLAM2 [28] (in red), our method (in blue) and ground truth trajectory (in green) for the TUM fr2_coke sequence, which is found to be critically affected by lateral depth noise in Figure 2.

Though monocular cameras can perform visual odometry, they suffer from scale, drift and initialization issues [28], [35] as they do not have a direct perception of depth. Active depth sensors (eg. Kinect V2) have been preferred over passive ones for a wide variety of use cases and have been studied extensively [20], [38], [31], [37], [47]. These works also focussed on modelling the uncertainty of depth sensors; [20] modelled depth measurement from disparity through mathematical formulation, while Nguyen *et al.* [31] empirically derived a noise model for the Kinect and applied it to the KinectFusion system [29]. Utilizing such fundamental models of sensor characteristics, there have been only a few extensions to apply them to tackle depth noise problems in stereo/RGB-D SLAM; using Gaussian mixture models with Kalman filters [11], probabilistic line fitting solutions [36], along with weighting schemes to denoise depth using residuals [46]. There have been various works in monocular SLAM using inverse depth parameterization [5], [30], [14]. Though direct noise models are not employed, volumetric depth fusion models also exist [19], [29], [48], [49], [7].

Estimating depth error is crucial for SLAM performance. Strasdat *et al.* [39] proposed a virtual right stereo camera onto which the depth error is projected, which was also adopted by ORB-SLAM2 [28]. It uses a Bundle Adjustment

[1]All authors are with Processor Architecture Research Lab, Intel Labs. {anirud.thyagharajan, om.j.omer, dipan.mandal, sreenivas.subramoney}@intel.com

(BA) based optimization framework for minimizing the 3D geometric error between the estimated landmark in 3D coordinates (Map-Point [27]) and 3D feature (obtained by back-projecting 2D feature of the landmark from a camera viewpoint using estimated depth), helping in providing noise resistant poses to an extent. However, the optimization process is unlikely to detect scenarios where noisy depth affects feature correspondences, which could severely impact SLAM performance. [36], [15], [12], [43] have explored denoising methods to mitigate a few limitations, but incorporating noise resilience inherently into the SLAM system is always beneficial, especially since heavy pre-processing is expensive.

In our work, we pick cues from Nguyen *et al.* [31] to formulate the components of depth noise (lateral and axial). We propose two novel techniques addressing both categories of depth noise to architect noise-resilient SLAM systems. We show that these techniques, individually and cumulatively, improve SLAM performance on all sequences, obtaining upto **2-3x** improvement over state-of-the-art on those critically affected by depth noise.

1. We highlight limitations with state-of-the-art sparse indirect SLAM solutions in handling depth-noise.

2. To the best of our knowledge, we are the first to present a notion of dynamic centroid of 3D features and propose **Map-point Consensus based Outlier Rejection** to detect and filter out noise affected landmarks and 3D features through a novel hierarchical decision flow.

3. We also propose a novel technique, **Adaptive Virtual Camera**, which dynamically identifies the virtual camera location per feature, and propose a depth error term to accurately estimate the depth noise. We extend this error term in both x and y directions, as opposed to only in the baseline direction [28].

II. Motivation

Monocular vision based SLAM suffers from scale, drift and initialization issues especially when the camera is subjected to pure rotation [35] or low camera motion along the projection plane. Therefore, depth cues are essential for good SLAM performance. However, all commodity depth sensors suffer from noisy measurements. Nguyen *et al.* [31] cited two major components of depth noise (i) axial noise: error in depth along the depth axis, (ii) lateral noise: depth of the current pixel being influenced by neighboring pixels. We highlight key issues with state-of-the-art solutions in dealing with noise which severely impacts SLAM performance:

Outlier Detection & Rejection: ORB-SLAM2 [28] classifies features into inliers and outliers during BA based on the error between the map-point and the 3D feature. It gauges the pose estimation fidelity using the inlier-outlier ratio. If the ratio is poor, the process is reiterated with additional cues (like matching more features), or tracking is considered as failed for that camera viewpoint. Similarly, [13] identifies outliers based on photometric error during the joint optimization process and rejects them in the next iterations. In such solutions where outlier detection occurs either as part of or at the end of the optimization routine, the estimated

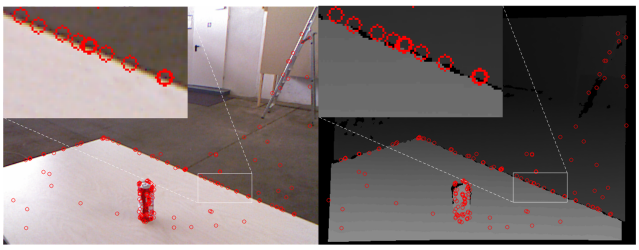

Fig. 2: RGB-D images showing outliers for the TUM fr2_coke sequence. Most of the outliers (in red) occur along the depth boundaries, typically showing the impact of lateral depth noise.

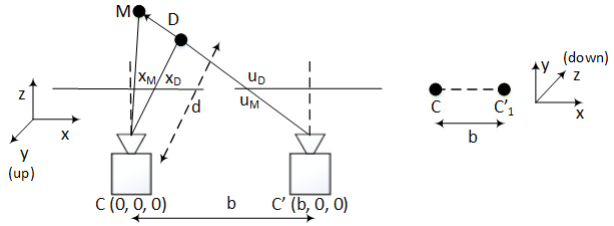

Fig. 3: Failure case for the static virtual camera

pose gets affected by outliers. Since outlier detection itself depends on the estimated camera pose, in several scenarios bad features may get annotated as inliers and good features could be discarded as outliers, badly impacting the reliability of the estimated pose. Some solutions use RANSAC [22], [10] to iteratively pick random sets of features, track number of outliers in each set and choose the set with the lowest number of outliers, but such methods are computationally demanding and non-deterministic.

Lack of Consensus: ORB-SLAM2 [28] detects outlier features on a per-frame basis assuming the feature in the anchor frame (camera viewpoint from where the landmark is first perceived) to be pristine. However, it is possible that the feature in the anchor frame (and possibly in few subsequent ones as well) is observed incorrectly, while in most of the viewpoints later it is observed accurately. Due to the lack of a consensus based approach, a potential good feature could be discarded as an outlier. Also, in some scenarios, even though a bad feature is identified as an outlier in the majority of frames, it could be treated as an inlier in a few frames, thus affecting the estimated pose for those frames.

Correspondence Issues at Boundaries: Since the depth estimate of a feature-point is influenced by neighborhood region due to lateral noise, points on the depth boundaries are generally very deviant from their true locations. As shown in Figure 2, a landmark in a foreground object might be perceived as if it is located in the background and vice versa. For tracking a feature across viewpoints, a motion-model [8] is generally employed to predict the search region. In the presence of lateral depth noise, such predictions are plausible to fall short of their true locations, leading to either tracking failure or latching onto wrong correspondences. SLAM methods such as [28], [13], which do not have outlier rejection to prune out such features prior to the optimization process, are likely to be adversely affected.

Static Virtual Camera: Indirect methods such as [8], [28], [18], [21] use projections of the 3D geometric error onto

the image plane as error terms for the optimization process. This favors feature points nearer to the camera to have a stronger contribution towards pose estimation as compared to the farther ones. In indirect methods, 3D error projection cannot capture error along depth axis, as all points on the depth axis would have the same projection. To mitigate this, ORB-SLAM2 [28] followed [39] to use a virtual camera at a fixed baseline distance on the x-axis from the actual camera to project the 3D error onto the virtual camera image plane as a means to reflect the depth error component. We find this method to be inaccurate as the static virtual camera cannot capture the depth error for all 3D error orientations. As shown in Figure 3, when the 3D feature D, virtual camera C' and the hitherto estimated map-point M are collinear, the 3D error projection would be zero. In such scenarios, a non-zero 3D error vector would be treated as zero depth error with a static virtual camera causing an inherent bias for certain depth error orientations, thus severely affecting performance of SLAM methods. Direct methods [13], [14] do not suffer from this phenomenon as they operate with photometric error instead of geometric error.

III. Method

A. Map-point Consensus based Outlier Rejection (MC-OR)

We propose the concept of a moving centroid $\mathbf{G}$ amongst 3D features of the landmark which is updated after every camera viewpoint. Using $\mathbf{G}$, we introduce distance metrics to draft a hierarchical outlier rejection scheme, that weeds out both individual aberrant points and entire groups of points. As seen in Figure 4, for a generic SLAM system in the world coordinate, let L_X be a landmark which is tracked across N camera views, $C = \{\mathbf{C_1}, \mathbf{C_2}, \dots \mathbf{C_N}\}$; this landmark is perceived from these camera views at 2D features $F_{2D,X} = \{\mathbf{x_1}, \mathbf{x_2} \dots \mathbf{x_N}\}$. By backprojecting these 2D projections using their depth d_i, we obtain the set of 3D features $F_{3D,X} = \{\mathbf{X_1}, \mathbf{X_2} \dots \mathbf{X_N}\}$, referred to as the cluster of landmark L_X. The SLAM system maintains an estimated representation of the landmark, the map-point $\mathbf{M}$.

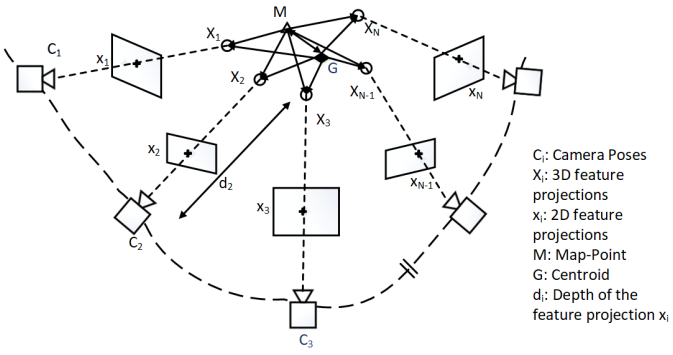

Fig. 4: Description of the MC-OR scheme in a multi-view scenario

$$\mathbf{G} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X_i} \tag{1}$$

For outlier rejection, we propose the following Map-point Consensus Error (MCE) metrics:

Feature-level metrics (FL): These metrics are used to score 3D features individually. The FL metrics portray how consistent the 3D feature is with rest of the data. The FL metrics aid in providing a specific understanding of the adherent and deviant 3D projections of the landmark and this helps to prune the bad projections individually.

$$MCE_{FL,1,i} = \|\mathbf{M} - \mathbf{X_i}\| \tag{2}$$

$$MCE_{FL,2,i} = \|\mathbf{G} - \mathbf{X_i}\| \tag{3}$$

If majority of the 3D features of a particular frame have high FL metrics, it is very likely that the camera pose is inaccurate, providing essential cues in performance indication.

Cluster-level metrics (CL): These metrics are used to score the cluster as a whole. The first two metrics (Equations 4 and 5) represent the mean values of the FL metrics (Equations 2 and 3), while Equation 6 conveys how close the map-point is to the statistical mean of the depth data.

$$MCE_{CL,1} = \frac{1}{N}\sum_{i=1}^{N} MCE_{FL,1,i} \tag{4}$$

$$MCE_{CL,2} = \frac{1}{N}\sum_{i=1}^{N} MCE_{FL,2,i} \tag{5}$$

$$MCE_{CL,3} = \|\mathbf{M} - \mathbf{G}\| \tag{6}$$

The CL metrics provide a notion of consensus; denoting how consistent multiple sightings of a landmark are and in general how well-observed the landmark is. If they are high, it means that the landmark is bad to consider as an interest point.

We group these metrics based on similarity and subject similar metrics to the same thresholds: (i) distances between $\mathbf{M}$, $F_{3D,X}$ and their average (Equations 2 and 4), (ii) distances between $\mathbf{G}$, $F_{3D,X}$ and their average (Equations 3 and 5), and (iii) distances between $\mathbf{G}$ and $\mathbf{M}$ (Equation 6). The corresponding thresholds are $\tau_{M,F}$, $\tau_{G,F}$ and $\tau_{M,G}$. For fresh frames, the camera poses are assumed to be initialized by the SLAM framework, utilizing modules like motion-

Algorithm 1 Map-point Consensus based Outlier Rejection

```
L ← Number of feature-clusters
C ← {c_1, c_2, c_3..c_L}
τ_{M,F}, τ_{G,F}, τ_{M,G} ← thresholds
for i ← 1 to L do
    c ← C[i]
    F ← c.getFeaturesInCluster()
    for f in F do
        f.computeFeatureLevelMetrics()
    end for
    c.computeClusterLevelMetrics()
end for
selected_features = []
for i ← 1 to L do
    c ← C[i]
    k_{CL,1}, k_{CL,2}, k_{CL,3} ← c.getCLMetrics()
    if k_{CL,1} > τ_{M,F} or k_{CL,2} > τ_{G,F} or k_{CL,3} > τ_{M,G} then
        continue
    end if
    F ← c.getFeaturesInCluster()
    for f in F do
        k_{FL,1}, k_{FL,2} ← f.getFLMetrics()
        if k_{FL,1} > τ_{M,F} or k_{FL,2} > τ_{G,F} then
            continue
        end if
        selected_features.append(f)
    end for
end for
```

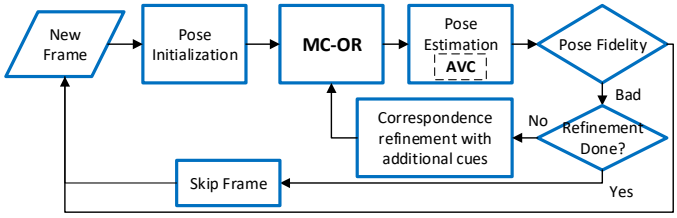

Fig. 5: Placement of MC-OR and AVC in a generic SLAM system.

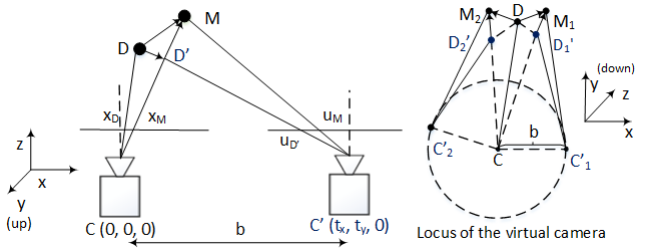

Fig. 6: Adaptive Virtual Camera Setup

models, PnP models, ICP etc. as viewed in ORB-SLAM2 and DSO. SLAM methods like ORB-SLAM2 conduct multiple pose-refinements iteratively, based on certain optimization performance indicators. As indicated in Figure 5, our outlier rejection method is designed to act before every run of the pose optimization block. The outlier rejection process is detailed in Algorithm 1. After computing the FL and CL metrics, a hierarchical approach is adopted wherein the aberrant clusters are first culled using the CL metrics and for the remnant clusters, the individual FL observations are culled using the FL metrics. Only those features that pass both the tests enter the pose optimization framework.

The running centroid serves as an estimate of the map-point prior to the optimization, thus helping in pre-filtering observations before the optimization. Even between multiple runs of the optimization for a frame, MC-OR ensures the centroid to be dynamically updated. Constraining the distance between $\mathbf{M}$ and $\mathbf{G}$ ensures the map-point to be close to the statistical mean of the 3D features with relatively accurate depth. Where $\mathbf{M}$ is updated based on cumulative projective errors across other features as well, $\mathbf{G}$ is affected only by the depth of that particular feature and thus provides a pinpointed qualitative assessment of the feature-cluster.

B. Adaptive Virtual Camera

1) Prior Art: Consider the camera located at $\mathbf{C}(0,0,0)$, viewing a landmark L_X, projected onto the image plane as the 2D feature $\mathbf{x_{2D}}$, having a depth d, as shown in Figure 6. The 3D feature of the landmark is found by back-projecting $\mathbf{x_{2D}}$ using d as $\mathbf{D}(X_D, Y_D, Z_D)$. The estimated location of the landmark, namely the map-point is $\mathbf{M}(X_M, Y_M, Z_M)$.

Let $\mathbf{D}$ and $\mathbf{M}$ be projected onto the image plane as (x_D, y_D) and (x_M, y_M) respectively using camera parameters (f_x, f_y, c_x, c_y). In previous works [28], [39], they use a static virtual camera located at a baseline b along the x-axis to the right of the original camera, such that it is rectified with respect to the original camera. The error vector $\overrightarrow{DM}$ is projected onto the image plane of the virtual camera for optimization. Their error vector is represented by:

$$e_{ORB} = \begin{bmatrix} x_M - x_D \\ y_M - y_D \\ (x_M - \frac{f_x b}{Z_M}) - (x_D - \frac{f_x b}{Z_D}) \end{bmatrix} \tag{7}$$

Since depth noise grows with depth, normalizing the error with z helps in preventing disproportionate contributions from near/far observations, hence the 3D errors are projected. The projective errors across all the features are accumulated and minimized in a least-squares BA formulation [28], [7], [45]. In ORB-SLAM2, outliers are filtered post-optimization and are removed from consequent optimizations.

2) Our Proposal: As discussed in Section II (Figure 3), the static virtual camera has biases for certain error orientations. To circumvent this, we propose the following: (i) the position of the virtual camera should be adjusted adaptively per landmark on the camera plane (both x & y direction) at constant baseline from camera instead of always positioning to right of the camera, (ii) instead of projecting 3D geometric error onto virtual camera's image plane, only the component along depth axis should be projected (iii) since virtual camera can be located anywhere on camera plane, projection error term should be capturing projection error in both x and y directions. We first identify 3D error component $\widehat{\boldsymbol{D'M}}$ along depth axis $\widehat{\boldsymbol{CM}}$ using Equations 8, 9, as shown in Figure 6.

$$\overrightarrow{\boldsymbol{CD'}} = \begin{bmatrix} X_{D'} & Y_{D'} & Z_{D'} \end{bmatrix}^T = \|\overrightarrow{\boldsymbol{CD}}\|\widehat{\boldsymbol{CM}} \tag{8}$$

$$\overrightarrow{\boldsymbol{D'M}} = \overrightarrow{\boldsymbol{CM}} - \overrightarrow{\boldsymbol{CD'}} \tag{9}$$

Note that $\overrightarrow{\boldsymbol{DD'}}$ and 3D error vector $\overrightarrow{\boldsymbol{DM}}$ will have the same projection onto image plane of $\mathbf{C}$. To find the location of the virtual camera $\mathbf{C}'$, we take component of $\overrightarrow{\boldsymbol{DD'}}$ parallel to the image plane of the camera $\mathbf{C}$ as per Equation 10. We place the virtual camera $\mathbf{C}'$ at a baseline distance b from the camera $\mathbf{C}$ using Equation 11 to maximize the error projection. Here $\hat{\boldsymbol{n_c}}$ represents the normal to image plane. Note that the locus of $\mathbf{C}'$ is a circle of radius b centred at $\mathbf{C}$.

$$\overrightarrow{\boldsymbol{DD'}}_{proj,\hat{\boldsymbol{n_c}}} = \overrightarrow{\boldsymbol{DD'}} - (\overrightarrow{\boldsymbol{DD'}}.\hat{\boldsymbol{n_c}})\hat{\boldsymbol{n_c}} \tag{10}$$

$$\mathbf{C'} = \begin{bmatrix} t_x & t_y & 0 \end{bmatrix}^T = b\widehat{\boldsymbol{DD'}}_{proj,\hat{\boldsymbol{n_c}}} \tag{11}$$

As per our proposal, the error $\overrightarrow{\boldsymbol{D'M}}$ is projected onto the virtual camera's image plane. Since $\overrightarrow{\boldsymbol{D'M}}$ is collinear with $\mathbf{C}$, all the viewpoints from which the projection of this vector would be zero, will be on the line joining $\mathbf{C}$, $\mathbf{D}'$ and $\mathbf{M}$. Since $\mathbf{C}'$ is located b distance away from $\mathbf{C}$, this provides a guarantee that a non-zero $\|\overrightarrow{\boldsymbol{D'M}}\|$ would always have a non-zero projection onto the image plane of $\mathbf{C}'$. Additionally, we also choose an optimal position of $\mathbf{C}'$ as observed in Equation 11. We construct the error equation using Equations 7, 9 and 11, including error projections on both x and y directions on the virtual camera's image plane. We minimize a least-squares error formulation to solve for the camera poses and map-point locations during the pose estimation phase (Figure 5).

$$e_{AVC} = \begin{bmatrix} x_M - x_D \\ y_M - y_D \\ (x_M - \frac{f_x t_x}{Z_M}) - (x_M - \frac{f_x t_x}{Z_{D'}}) \\ (y_M - \frac{f_y t_y}{Z_M}) - (y_M - \frac{f_y t_y}{Z_{D'}}) \end{bmatrix} = \begin{bmatrix} x_M - x_D \\ y_M - y_D \\ (\frac{f_x t_x}{Z_{D'}} - \frac{f_x t_x}{Z_M}) \\ (\frac{f_y t_y}{Z_{D'}} - \frac{f_y t_y}{Z_M}) \end{bmatrix} \tag{12}$$

IV. Evaluation

A. Experimental Setup

Datasets: We evaluate our algorithms on both stereo and RGB-D data; we utilize the TUM RGB-D dataset [40] and the EuRoC stereo dataset [2], consisting of 6DoF ground truth pose and sequences with varying difficulty. To highlight noise affected sequences and benefits with our methods, we group the TUM and EuRoC sequences into two categories, based on ATE RMSE (m) scores with ORB-SLAM2: (i) *easy*, having scores < 0.10 m, (ii) *hard*, otherwise.

Implementation Details: While we implement both the proposed techniques on ORB-SLAM2 [28], a state-of-the-art SLAM implementation, we find both techniques can generalize well to other SLAM implementations. We compare both techniques (individually and cumulatively) with baseline ORB-SLAM2, DVO [19], BundleFusion [7] and ElasticFusion [49], DSO [13], OKVIS [23] and SVO [16].

Reducing indeterminism: We find ORB-SLAM2 to exhibit high variations across runs [6]. Depending upon whether keyframe-BA in the local mapping has finished or not, the tracking thread may not obtain the most optimized keyframe pose, causing indeterminism. To mitigate these issues, we synchronize all ORB-SLAM2 threads to complete before processing the next frame. Secondly, the algorithm uses multiple RANSACs within solving PnP models for relocalization/pose initialization, causing high variations in SLAM performance. To tackle this, we run multiple ($n = 10$) runs of the same configuration to squash variability, and report mean ATE RMSE [40] for pose evaluation.

B. Map-point Consensus based Outlier Rejection

The thresholds $\tau_{M,F}$, $\tau_{G,F}$ and $\tau_{M,G}$ were varied over a discrete parameter space $\{0.1, 0.3, 0.5, 0.7, 0.9\}$; while 0.9 was lenient (allowing noisy points), a threshold of 0.1 proved to be too strict for highly noisy sequences like fr3_walking_rpy. Through exhaustive search, while different parameter optima were observed across datasets, we observed optimal SLAM performance for $\{\tau_{M,F}, \tau_{G,F}, \tau_{M,G}\} = \{0.7, 0.7, 0.5\}$ across sequences, and used this for generating results.

Pose Evaluation on MC-OR: As shown in Table III (Column 'w/ MC-OR'), MC-OR significantly improves over ORB-SLAM2 for *hard* and performs on par or marginally better for *easy* sequences. We find significant improvement for *hard* sequences due to elimination of bad features, alongwith retaining good features, which aids in maintaining the performance for *easy* sequences. Figure 7 demonstrates instances along depth boundaries where MC-OR detects inconsistencies between the RGB and depth images on account of lateral noise as described in Section II.

Inlier-Outlier Analysis: Figure 8 compares the inlier-outlier balance of the BA framework post outlier-rejection. MC-OR's outlier ratio is consistently lesser than that of ORB-SLAM2; while the number of inliers reduces by a small amount or stays the same, the number of outliers reduces considerably. Thus, MC-OR is effective in identifying depth features that have high lateral depth noise and removing them prior to pose estimation; utilizing consensus information amongst other sightings of the same landmark to cull depth observations that stray from the rest of the compact cluster.

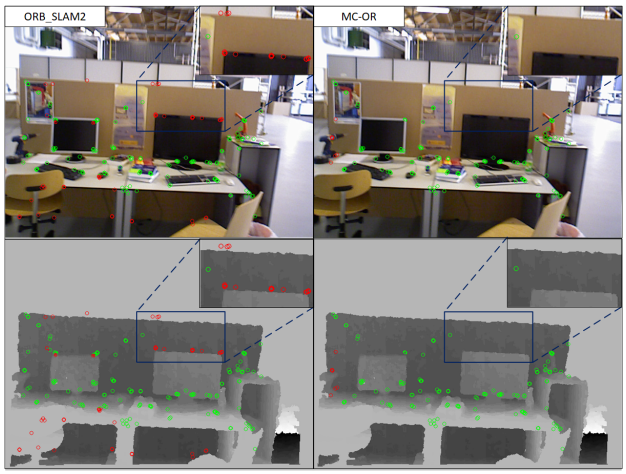

Fig. 7: Shows the outliers (in red) rejected by our scheme on frames of the *fr3_walking_rpy* dataset before pose optimization.

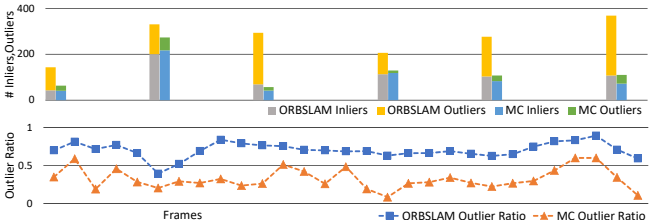

Fig. 8: Comparison for ORB-SLAM2 with and without MC-OR; lines indicate the outlier ratio comparison (lesser is better); bar charts show certain instances of inlier and outlier counts.

Invariance to multiple runs: MC-OR also aids in reducing the variation of pose-optimization over $n = 10$ runs upto 10x, as shown in Figure 9. We find that MC-OR reduces ORB-SLAM2's dependency on RANSAC while selecting a good set of inliers and consistently provides high fidelity features, thus reducing the variability.

C. Adaptive Virtual Camera (AVC)

Invariance to parameter b: We expect AVC to be resilient with baseline b, as it truly represents depth noise. Thus, we compare the performance of AVC with base ORB-SLAM2 on varying b. In ORB-SLAM2, this parameter is taken as approximately 8 cm for the Kinect Sensor. Table I shows that AVC is relatively tolerant to the choice of b. We

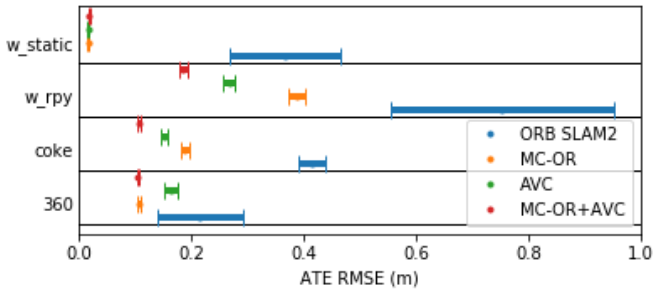

Fig. 9: Comparison of ATE variation over several runs of ORB-SLAM2 with and without MC-OR on TUM datasets

pick $b = 9$ cm for generating all subsequent results.

TABLE I: Comparison of ATE (m) for ORB-SLAM2 with and without AVC under variations in baseline, b on fr3_walking_rpy.

b (m)	0.01	0.03	0.05	0.07	0.09	0.11	0.13
w/o AVC	0.60	0.59	0.65	0.58	0.83	0.56	0.72
w/ AVC	0.30	0.42	0.37	0.30	0.27	0.34	0.26

Pose Evaluation on AVC: As observed in Table III (Column 'w/ AVC'), AVC performs significantly better than ORB-SLAM2 for *hard* sequences and does marginally better for the rest. This aids us to conclude that AVC is able to truly represent depth error and eliminate the effect of axial depth noise better than existing systems, instrumental in enhancing pose estimation performance.

Correlation Analysis: Table II demonstrates how closely the AVC error term represents depth noise as compared to that of ORB-SLAM2. $e_{ORB,z}$ indicates the projected virtual error component in ORB-SLAM2, while $e_{AVC,zx}$ and $e_{AVC,zy}$ denote the x-y virtual error components in e_{AVC} (Equation 12); $e_{BASE,z}$ denotes the z-normalized 3D error e_{BASE}, while $(\Delta x, \Delta y)$ are the 2D projective errors on the original camera's image plane. ORB-SLAM2's static

TABLE II: Correlation Scores between error terms $\{e_{ORB,z}, e_{AVC,zx}, e_{AVC,zy}\}$ and $\{e_{BASE,z}, (\Delta x, \Delta y)\}$

	$e_{ORB,z}$	$e_{AVC,zx}$	$e_{AVC,zy}$
$e_{BASE,z}$	0.25	0.72	0.76
$(\Delta x, \Delta y)$	$(0.96, 0.03)$	$(0.10, 0.12)$	$(0.12, 0.12)$

virtual camera error does not represent the z-normalized error well, rather showing a high correlation with Δx, showing redundancy in error representation. On the contrary, the AVC error terms are well correlated with $e_{BASE,z}$ and relatively independent of $(\Delta x, \Delta y)$, thus presenting new information.

D. Combined AVC & MC-OR

Pose Evaluation: Since our proposals deal with distinct types of depth noise, we also conducted experiments with both AVC and MC-OR enabled in Table III (Column 'w/ AVC+MC-OR') to understand if they complement each other. For *hard* sequences, our methods provide significant improvements individually and cumulatively. For *easy* sequences, these methods maintain or improve SLAM performance incrementally. Figure 1 shows a comparison of trajectories, showing that our methods are able to aid ORB-SLAM2 in localizing the camera more accurately.

Time Complexity Analysis: As shown in Table IV, both MC-OR and AVC improve computational performance along with overall pose accuracy. Rejecting noisy depth outliers through MC-OR results in heavily reducing the number of error terms for the SLAM graph-solver to process. In case of AVC, eventhough the dimensionality of e_{AVC} increases from 3 to 4, the time profile of BA is known to be highly dominated by Cholesky factorization and linear equation solver. Moreover, we observe that AVC induces the optimizer to converge quicker across frames.

TABLE III: Comparison of ATE RMSE on TUM and EuRoC datasets; '-' denotes results were not published, 'F' denotes failure cases; 'fr3_w_' and 'fr_s_' indicate the freiburg3_walking_ and freiburg3_sitting_ sequences respectively.

	TUM Sequence	ORB-SLAM2	w/ MC-OR	w/ AVC	w/ MC-OR+AVC	DVO [19]	BF [7]	EF [49]
Easy	fr1_desk	0.023	**0.014**	0.015	0.015	0.021	0.016	0.020
	fr1_plant	0.018	0.014	0.014	**0.013**	0.028	-	0.022
	fr1_rpy	0.022	0.018	0.019	**0.018**	0.021	-	0.025
	fr1_xyz	0.010	**0.009**	0.009	0.010	0.011	-	0.011
	fr1_floor	0.016	0.013	0.013	**0.012**	-	-	F
	fr1_desk2	0.031	0.023	0.023	**0.022**	0.046	-	0.048
	fr1_room	0.079	0.044	0.046	**0.043**	0.053	-	0.068
	fr1_teddy	0.055	0.032	0.033	**0.031**	0.034	-	0.083
	fr2_desk	0.011	0.009	0.009	**0.008**	0.017	-	0.009
	fr2_xyz	0.004	0.004	0.003	**0.003**	0.018	0.011	0.04
	fr3_s_sphere	0.023	0.016	0.017	**0.016**	-	-	-
	fr3_s_rpy	0.020	0.018	0.019	**0.017**	-	-	-
	fr3_s_static	0.009	0.007	0.008	**0.007**	-	-	-
	fr3_s_xyz	0.009	**0.008**	0.009	0.009	-	-	-
	fr3_office	0.010	0.009	0.009	**0.008**	0.035	0.022	0.017
	fr3_nst	0.023	0.019	0.019	0.018	0.018	**0.012**	0.016
Hard	fr1_360	0.215	0.107	0.164	**0.105**	0.083	-	0.108
	fr2_coke	0.415	0.188	0.152	**0.106**	-	-	F
	fr3_w_rpy	0.753	0.386	0.267	**0.185**	-	-	-
	fr3_w_static	0.367	**0.016**	0.016	0.018	-	-	-
	EuRoC Sequence	ORB-SLAM2	w/ MC-OR	w/ AVC	w/ MC-OR+AVC	DSO [13]	SVO [16]	OKVIS [23]
Easy	EuR/MH01	0.037	0.035	0.035	**0.035**	0.050	0.040	0.330
	EuR/MH02	0.048	0.039	0.039	**0.036**	0.050	0.050	0.370
	EuR/MH03	0.040	0.039	0.038	**0.036**	0.180	0.060	0.250
	EuR/MH05	0.053	0.043	0.043	**0.040**	0.110	0.120	0.390
	EuR/V101	0.088	0.087	0.087	0.086	0.120	**0.040**	0.094
	EuR/V102	0.065	0.061	0.061	0.061	0.110	**0.040**	0.140
	EuR/V103	0.088	0.074	0.072	**0.062**	0.660	0.070	0.210
	EuR/V201	0.066	0.062	0.060	0.057	**0.040**	0.050	0.090
	EuR/V202	0.060	0.056	0.056	**0.053**	0.190	0.090	0.170
Hard	EuR/MH04	0.124	0.070	0.060	**0.045**	2.500	0.170	0.270
	EuR/V203	0.280	0.270	0.213	**0.189**	1.160	0.790	0.230

TABLE IV: Comparison of frames processed per second (fps) for single-threaded ORB-SLAM2 on the TUM Dataset; on an Intel Xeon (R) E5-2667 3.2 GHz machine.

Sequence	# Frames	ORB-SLAM2	w/ MC-OR	w/ AVC
fr1_360	744	12.17	15.16	13.70
fr1_xyz	792	5.89	11.62	7.76
fr2_coke	2472	5.59	22.00	13.01
fr3_w_rpy	866	2.31	4.24	8.38
fr3_w_static	717	3.06	7.47	4.50

V. Conclusions

We presented novel methods MC-OR and AVC to improve fidelity of pose estimation under noisy depth scenarios. In contrast to previous methods, MC-OR tracks features over multiple frames and based on consensus prevents (i) the removal of good features with depth inaccuracies in initial frames, and (ii) latching onto bad features inconsistent with most frames, contributing to improved pose estimation. Our consensus based approach is also applicable to identify moving objects in dynamic scenes which would otherwise degrade performance of SLAM methods. AVC targets to correct the axial depth error measurement for optimization methods as used in state-of-the-art indirect techniques. Looking ahead, it would be interesting to study the impact of our work on other kinds of dense/fusion SLAM systems; expecting a positive impact due to the intuitive and generalizable reasoning behind our contributions.

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

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

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[4] L. Carlone and G. C. Calafiore. Convex relaxations for pose graph optimization with outliers. *IEEE Robotics and Automation Letters*, 3(2):1160–1167, 2018.

[5] J. Civera, A. J. Davison, and J. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE transactions on robotics*, 24(5):932–945, 2008.

[6] I. Cvisic, J. Cesic, I. Markovic, and I. Petrovic. Soft-slam: Computationally efficient stereo visual slam for autonomous uavs. *Journal of Field Robotics*, 2017.

[7] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017.

[8] 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*, (6):1052–1067, 2007.

[9] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241, 2001.

[10] P. Dong, X. Ruan, J. Huang, X. Zhu, and Y. Xiao. A rgb-d slam algorithm combining orb features and bow. In *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*, page 118. ACM, 2018.

[11] I. Dryanovski, R. G. Valenti, and J. Xiao. Fast visual odometry and mapping from rgb-d data. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2305–2310. IEEE, 2013.

[12] T. Edeler, K. Ohliger, S. Hussmann, and A. Mertins. Time-of-flight depth image denoising using prior noise information. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, pages 119–122. IEEE, 2010.

[13] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2018.

[14] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[15] K. Essmaeel, L. Gallo, E. Damiani, G. De Pietro, and A. Dipandà. Temporal denoising of kinect depth data. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 47–52. IEEE, 2012.

[16] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2017.

[17] M. Hsiao and M. Kaess. Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1274–1280. IEEE, 2019.

[18] H. Jin, P. Favaro, and S. Soatto. Real-time 3d motion and structure of point features: a front-end system for vision-based control and interaction. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 778–779. IEEE, 2000.

[19] C. Kerl et al. Dvo-slam. In *IROS*, 2013.

[20] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[21] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[22] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Rs-slam: Ransac sampling for visual fastslam. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1655–1660. IEEE, 2011.

[23] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013.

[24] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3923–3929. IEEE, 2013.

[25] D. Martins, K. van Hecke, and G. de Croon. Fusion of stereo and still monocular depth estimates in a self-supervised learning context. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 849–856, 2018.

[26] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61, 1988.

[27] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[28] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[29] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[30] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[31] C. V. Nguyen, S. Izadi, and D. Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 524–530. IEEE, 2012.

[32] K. Park, S. Kim, and K. Sohn. High-precision depth estimation with the 3d lidar and stereo fusion. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 2156–2163, 2018.

[33] M. Pfingsthorn and A. Birk. Simultaneous localization and mapping with multimodal probability distributions. *The International Journal of Robotics Research*, 32(2):143–171, 2013.

[34] M. Pfingsthorn and A. Birk. Generalized graph slam: Solving local and global ambiguities through multimodal and hyperedge constraints. *The International Journal of Robotics Research*, 35(6):601–630, 2016.

[35] C. Pirchheim, D. Schmalstieg, and G. Reitmayr. Handling pure camera rotation in keyframe-based slam. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 229–238. IEEE, 2013.

[36] P. F. Proença and Y. Gao. Probabilistic rgb-d odometry based on points, lines and planes under depth uncertainty. *Robotics and Autonomous Systems*, 104:25–39, 2018.

[37] H. Sarbolandi, D. Lefloch, and A. Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer vision and image understanding*, 139:1–20, 2015.

[38] J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. In *Consumer depth cameras for computer vision*, pages 3–25. Springer, 2013.

[39] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige. Double window optimisation for constant time visual slam. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2352–2359. IEEE, 2011.

[40] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[41] N. Sünderhauf. *Robust optimization for simultaneous localization and mapping*. PhD thesis, Technischen Universitat Chemnitz, 2012.

[42] N. Sünderhauf and P. Protzel. Switchable constraints vs. max-mixture models vs. rrr-a comparison of three approaches to robust pose graph slam. In *2013 IEEE International Conference on Robotics and Automation*, pages 5198–5203. IEEE, 2013.

[43] C. Sweeney, G. Izatt, and R. Tedrake. A supervised approach to predicting noise in depth images.

[44] P. H. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *International workshop on vision algorithms*, pages 278–294. Springer, 1999.

[45] R. Wang, M. Schwörer, and D. Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *International Conference on Computer Vision (ICCV)*, volume 42, 2017.

[46] O. Wasenmüller, M. D. Ansari, and D. Stricker. Dna-slam: Dense noise aware slam for tof rgb-d cameras. In *Asian Conference on Computer Vision*, pages 613–629. Springer, 2016.

[47] O. Wasenmüller and D. Stricker. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In *Asian Conference on Computer Vision*, pages 34–45. Springer, 2016.

[48] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially extended kinectfusion. 2012.

[49] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.