Learned Monocular Depth Priors in Visual-Inertial Initialization

Yunwen Zhou, Abhishek Kar, Eric Turner, Adarsh Kowdle, Chao X. Guo, Ryan C. DuToit, and Konstantine Tsotsos

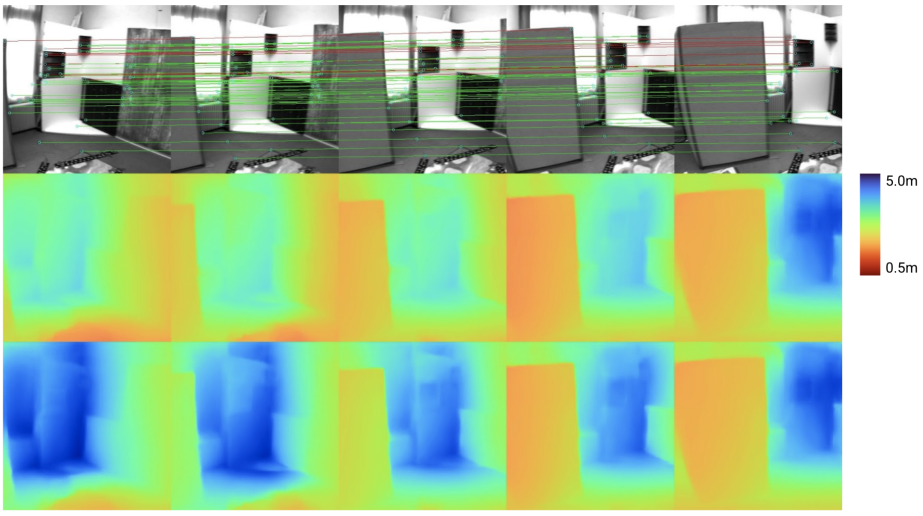
Google AR
{verse,abhiskar,elturner,adarshkowdle,chaoguo,rdutoit,ktsotsos}@google.com

Abstract. Visual-inertial odometry (VIO) is the pose estimation backbone for most AR/VR and autonomous robotic systems today, in both academia and industry. However, these systems are highly sensitive to the initialization of key parameters such as sensor biases, gravity direction, and metric scale. In practical scenarios where high-parallax or variable acceleration assumptions are rarely met (e.g. hovering aerial robot, smartphone AR user not gesticulating with phone), classical visual-inertial initialization formulations often become ill-conditioned and/or fail to meaningfully converge. In this paper we target visual-inertial initialization specifically for these low-excitation scenarios critical to in-the-wild usage. We propose to circumvent the limitations of classical visual-inertial structure-from-motion (SfM) initialization by incorporating a new learning-based measurement as a higher-level input. We leverage learned monocular depth images (mono-depth) to constrain the relative depth of features, and upgrade the mono-depth to metric scale by jointly optimizing for its scale and shift. Our experiments show a significant improvement in problem conditioning compared to a classical formulation for visual-inertial initialization, and demonstrate significant accuracy and robustness improvements relative to the state-of-the-art on public benchmarks, particularly under motion-restricted scenarios. We further extend this improvement to implementation within an existing odometry system to illustrate the impact of our improved initialization method on resulting tracking trajectories.

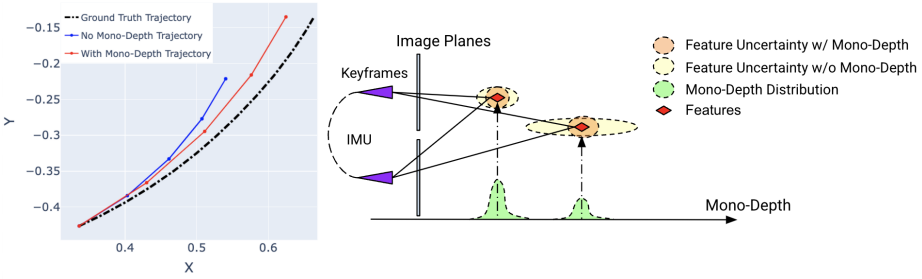
Keywords: Visual-inertial initialization, Monocular depth, Visual-inertial structure from motion

1 Introduction

Monocular visual-inertial odometry (VIO) enables accurate tracking of metric 3D position and orientation (pose) using just a monocular camera and inertial measurement unit (IMU) providing linear acceleration and rotational velocity. These techniques have unlocked an economical and near-ubiquitous solution for powering augmented or virtual reality (AR/VR) experiences on commodity platforms (e.g, ARCore on Android and ARKit on iOS [1]), alongside other robotic applications such as aerial delivery drones. A precondition of successful



(a) First Row: Intensity image inputs. **Second Row:** Mono-depth images. **Third Row:** Metric-depth images, recovered after joint motion, scale, and shift optimization. Stable metric-depth is recovered after the optimization from initial inconsistent and inaccurate mono-depth. **Green Tracks on First Row:** Inlier feature-tracks for mono depth constraints. **Red Tracks on First Row:** Outlier feature-tracks due to temporally inconsistent associated mono-depth values (see Sec. 3.2)



(b) Left: Initialization trajectory under low motion/parallax scenario in meters. Trajectory recovery is improved with tight coupling between VI-SFM and mono-depth (note incorrect scale in blue trajectory). **Right**: Mono-depth coupling improves problem conditioning, potentially reducing uncertainty of estimates and increasing accuracy.

Fig. 1: At top, demonstration of depth constraints over a keyframe initialization window. At bottom, demonstration of trajectories estimated with and without mono depth on the sequence shown at **top**, and illustration of feature position uncertainty.

operation in these scenarios is successful (and accurate) initialization of key system parameters such as scale, initial velocity, accelerometer and gyro biases, and initial gravity direction. Poor initialization typically leads to tracking divergence,

unacceptable transients, low-accuracy operation, or outright failures, especially of downstream modules (e.g. drone navigation software). Unfortunately, visual-inertial initialization routines have a very common failure mode in these realistic scenarios: insufficient motion for the system's motion and calibration states to be unambiguously resolvable [2–6]. This occurs, for example, if the user of a phone-based AR game moves with very little parallax relative to the visible scene or when a drone must initialize while hovering. These are extremely common in practice. To improve VIO initialization in these scenarios on commodity hardware we must optimize for the total (user-visible) latency to initialization and accuracy of the resulting trajectories, while not violating real-time operation. For example, a phone-based AR user may expect a responsive ($< 500ms$) startup of their game, regardless of how they moved their phone, and without taking noticeable compute resources from the primary AR application.

Due to its impact, many recent works have focused on formulating fast and accurate initialization algorithms for robust monocular VIO [5, 7–11]. These works rely on sparse visual feature tracks to constrain relative pose (up to scale) in the visual-inertial structure-from-motion (VI-SFM) problem. Under low parallax initialization scenarios, any classical depth estimation approach for these features in the VI-SFM problem will be susceptible to large uncertainty, such as in the sequence in Fig. 1a. This uncertainty (illustrated in Fig. 1b) makes the overall system ill-conditioned, often resulting in poor or failed initializations. This ambiguity is exacerbated if the inertial measurements lack enough variation to reliably recover metric scale [5].

Inspired by the robustness achievements of depth-enabled visual SLAM systems [12–15] and recent advances in generalized learning-based monocular depth (mono-depth) [16,17], we propose a novel formulation of monocular VIO initialization. We incorporate depth measurements from a mono-depth model directly into a classical VI-SFM framework as measurements. Our proposed method operates in real-time on a mobile phone and is able to accurately initialize in traditionally challenging low parallax or limited acceleration scenarios, *without* requiring an additional dedicated sensor for estimating depth (e.g. LiDAR, Time-of-Flight). Our primary contributions are:

- We apply learned monocular depth priors for VIO initialization. To the best of our knowledge, we are the first to leverage the power of learned depth for this problem through coupling with classical methods.
- We propose a novel residual function which tightly couples scale and shift invariant monocular depth measurements within a traditional VI-SFM formulation.
- We propose a gradient-based residual weighting function and an outlier rejection module to effectively deal with noisy depth predictions.
- We demonstrate robust and accurate initialization relative to the state-of-the-art on public benchmarks, particularly under motion-restricted scenarios and when embedded within an existing tracking system. We achieve all of the above while maintaining real-time performance on 10Hz image streams on resource constrained devices.

2 Related Work

Visual-inertial odometry [18, 19] is a well-studied problem in both the computer vision and robotics communities and many works [20–28] have focused specifically on accurate initial estimation of states required by the inertial sensor. These works can be roughly classified into two categories - 1) jointly solving a visual-inertial SFM problem directly in closed form or as a bundle adjustment problem [5, 10, 29] and 2) cascaded approaches which solve a pure visual SFM for up to scale pose followed by metric scale recovery using inertial observations [7–9, 30]. Both approaches typically use a visual-inertial bundle adjustment (VI-BA) step to further refine their solution.

Feature-based visual odometry (VO) plays a key role in VIO initialization but often exhibits large uncertainty in low parallax and motion scenarios. Additionally, the VO prior requires enough non-zero inertial measurements for observing metric scale [5] to initialize VIO. A recent state-of-the-art method [7] (used as the initialization routine for the popular ORBSLAM3 system [23]) still requires around 2 seconds (at 10Hz) to initialize and only succeeds with reasonable motion excitation. Our proposed method aims to initialize with lower (user-visible) latency (i.e. less data collection time) even in challenging low-motion scenarios. Some prior works have explored using higher order visual information such as lines [30] for increased system observability in monocular VIO. Additionally, RGB-D SLAM systems [12–14] have been tremendously successful in a number of domains (AR/VR, self driving cars, etc.) and can inherently initialize faster given direct depth observations. For example, [31] demonstrated that the inclusion of a depth sensor significantly reduces the required number of feature observations.

With the advent of deep learning, there has been significant interest in end-to-end learning for VIO [32–37]. However, the proposed methods often lack the explainability and modular nature of traditional VIO systems, have alternative end-goals (e.g. self supervised depth/optical flow/camera pose estimation), or are too expensive to operate on commodity hardware without custom accelerators. Moreover, end-to-end methods don't explicitly consider in-motion initialization and often benchmark on datasets with the trajectory starting at stationary point [38, 39]. Prior works have also explored learning methods in purely inertial [40–42] or visual systems [43–45]. CodeVIO [46] demonstrated that incorporating a differentiable depth decoder into an existing VIO system (OpenVINS) [47] can improve tracking odometry accuracy. Note that CodeVIO does not tackle the VIO initialization problem and relies on tracking landmarks from already-initialized VIO. It uses the OpenVINS initialization solution which only initializes after observing enough IMU excitation following a static period. However, CodeVIO does demonstrate an effective and modular integration of learned priors within VIO and inspires us to deliver similar improvements to VIO initialization, while operating under realtime performance constraints.

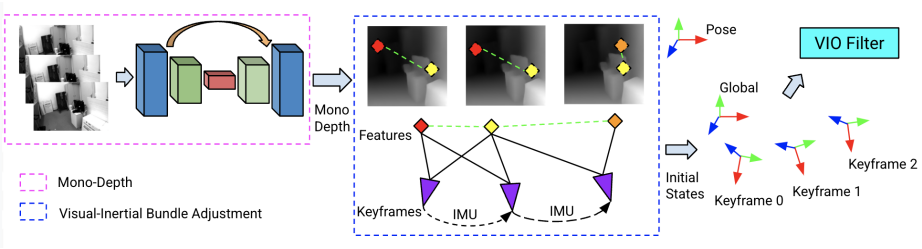


Fig. 2: Overall initialization diagram composed of monocular depth inference module running on each keyframe, and the visual-inertial bundle adjustment module. Initialized states are then fed into our VIO for tracking.

3 Methodology

Our proposed system is composed of two modules as shown in Fig. 2: 1) monocular depth inference which infers (relative) depth from each RGB keyframe, and 2) a VIO initialization module which forms a visual-inertial structure-from-motion (VI-SFM) problem, with the relative depth constraints from the inferred monocular depth. This VI-SFM problem aims to estimate keyframe poses, velocity, and calibration states, which are then used as the initial condition for a full VIO system.

Like most VIO initialization algorithms [7,8,29], our VIO initialization consists of a closed-form solver, whose solution is then refined with visual-inertial bundle adjustment (VI-BA). In this section, we first briefly describe our mono-depth model. Then, we detail our contribution on employing mono-depth constraints in VI-BA refinement.

3.1 Light-weight Monocular Depth Model

Our key contribution in this work is to incorporate prior-driven monocular depth constraints within a classical VIO initialization framework for better tracking initialization. For the final system to be practical, we require the mono-depth model to generalize to a wide variety of scenes and operate under a small compute budget. We follow recent state-of-the-art monocular depth estimation models [16] and train a lightweight mono-depth network. Specifically, we use the robust scale-shift invariant loss [16] alongside various edge-sensitive depth losses [16, 48] and train a small UNet model on a variety of datasets including ScanNet [49], MannequinChallenge [48] as well as pseudo-ground truth disparity maps generated on the OpenImages [50] dataset using large pretrained publicly available models [16]. For datasets with metric depth ground truth (e.g. ScanNet), we also add a loose metric depth loss term (Charbonnier loss [51] between prediction and inverse metric depth) to inform the scale and shift priors in Eq. (5). We trained our model on gravity-aligned (or "upright") images to avoid having it learn depth maps for "sideways" images and better use its limited model

capacity. Our final model is fast (Tab. 3), light-weight ($\sim 600K$ parameters) and predicts relative (inverse) depth maps as shown in Fig. 1a.

Given the scale-shift invariant nature of our training losses, the metric inverse depth, z, can be expressed as a scaled and shifted version of the model prediction, d, as $z = ad + b$, where a and b are the scale and shift parameters respectively. Moreover, as our model is trained on gravity aligned ("upright") images, we rotate the input image in 90-degree increments before inferring depth. Since only 45-degree accuracy is required to get the best rotation, for simplicity we use accelerometer measurements rotated through pre-calibrated IMU-camera extrinsics as an estimate of gravity in the camera frame.

3.2 VI-BA with Monocular Depth Constraints

We aim to solve for the following state parameters, $\boldsymbol{\mathcal{X}}$, in our VI-BA problem

$$\boldsymbol{\mathcal{X}} = [\boldsymbol{X_0}; \ldots; \boldsymbol{X_{N-1}}; {}^{\boldsymbol{C_j}}\boldsymbol{f_0}; \ldots; {}^{\boldsymbol{C_j}}\boldsymbol{f_{M-1}}; \boldsymbol{S_0}; \ldots; \boldsymbol{S_{N-1}}] \tag{1}$$

where

- $\boldsymbol{X_k}$ represents the k^{th} IMU keyframe state among N keyframes in total, which is $[\boldsymbol{q_k}; \boldsymbol{p_k}; \boldsymbol{v_k}; \boldsymbol{b_k^a}; \boldsymbol{b_k^\omega}]$. $\boldsymbol{q_k}$ and $\boldsymbol{p_k}$ are the k^{th} IMU keyframe pose parameterized as quarternion and translation w.r.t the global frame $\{G\}$ in which we assume the direction of gravity is known. $\boldsymbol{v_k}$ is the velocity in $\{G\}$ and $\boldsymbol{b_k^a}$, $\boldsymbol{b_k^\omega}$ are the accelerometer and gyro biases at the k^{th} keyframes.
- ${}^{\boldsymbol{C_j}}\boldsymbol{f_i}$ represents the i^{th} feature point parameterized in local inverse depth $[u_{ij}, v_{ij}, w_{ij}]^T$ with respect to the j^{th} keyframe's camera coordinates. u_{ij} and v_{ij} lie on normalized image XY plane and w_{ij} is the inverse depth [52].
- $\boldsymbol{S_k} = [a_k; b_k]$ following Sec. 3.1, which are scale and shift for recovering metric depth from the raw mono-depth at the k^{th} keyframe.
- The IMU-camera extrinsics ($\boldsymbol{q_C}$, $\boldsymbol{p_C}$) and 3D-2D projection parameters $Proj(\cdot)$ are not estimated due to lack of information in such a small initialization window. We adopt pre-calibrated values as is customary.

We initialize the state $\boldsymbol{\mathcal{X}}$ using a standard closed-form solver [10] for a VI-SFM problem formulated with reprojection error, the formulation and derivation are in the supplemental material. Given keyframes $\mathcal{K}$, with up to scale and shift mono inverse depth, feature points $\mathcal{F}$, and $\mathcal{L}(\subset \mathcal{F})$ feature points with mono inverse depth measurements, the VI-BA minimizes the following objective function

$$\begin{aligned}\hat{\boldsymbol{\mathcal{X}}} = \underset{\boldsymbol{\mathcal{X}}}{\operatorname{argmin}} & \underbrace{\sum_{(i,j)\in\mathcal{K}} \|\boldsymbol{r_{\mathcal{I}_{ij}}}\|^2_{\Sigma_{ij}}}_{\text{Inertial Constraints}} + \underbrace{\sum_{i\in\mathcal{F}}\sum_{k\in\mathcal{K}} \rho(\|\boldsymbol{r_{\mathcal{F}_{ik}}}\|^2_{\Sigma_\mathcal{F}})}_{\text{Visual Constraints}} \\ + & \underbrace{\sum_{i\in\mathcal{L}}\sum_{k\in\mathcal{K}} \lambda_{ik}\rho(\|r_{\mathcal{L}_{ik}}\|^2)}_{\textbf{Mono-Depth Constraints}} + \underbrace{\|\boldsymbol{r_0}\|^2_{\Sigma_0} + \sum_{i\in\mathcal{K}} \|\boldsymbol{r_{\mathcal{S}_i}}\|^2_{\Sigma_\mathcal{S}}}_{\text{Prior Constraints}}\end{aligned} \tag{2}$$

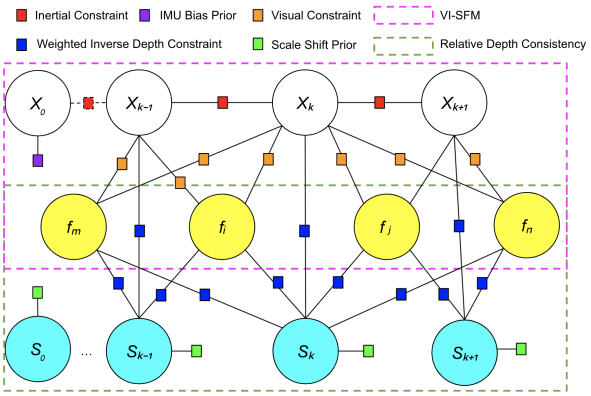


Fig. 3: A factor graph illustration of the VI-SFM depth refinement problem Eq. (2). Circled nodes represent $\boldsymbol{\mathcal{X}}$ in Eq. (1) to be estimated. They are connected by constraints illustrated in the graph. The **pink dashed box** is the traditional VI-SFM problem. The **green dashed box** represents the new proposed constraints to maintain relative feature depth consistency across keyframes. Feature points and poses are constrained through the scale-shift parameters $\boldsymbol{S}$.

where $\boldsymbol{r}_{\mathcal{I}_{ij}}$ is the IMU preintegration residual error [53] corresponding to IMU measurements between two consecutive keyframes, $\boldsymbol{r}_{\mathcal{F}_{ik}}$ is the standard visual reprojection residual resulting from subtracting a feature-point's pixel measurement from the projection of f_i into the k^{th} keyframe [54], $\boldsymbol{r}_{\mathcal{L}_{ik}}$ is an inverse depth temporal consistency residual for incorporating mono-depth, and $\boldsymbol{r}_{\mathcal{S}_i}$ is a residual relative to a prior for scale and shift (Sec. 3.2). $\boldsymbol{r}_0$ is a prior for the bias estimates of the $0th$ keyframe and Σ_0, Σ_{ij}, $\Sigma_{\mathcal{F}}$, $\Sigma_{\mathcal{S}}$ are the corresponding measurement covariance matrices. λ_{ik} is a scalar weight for each depth residual and $\rho(.)$ refers the huber-loss function [55].

The factor graph resulting from (2) is illustrated in Fig. 3. ($\boldsymbol{r}_{\mathcal{I}_{ij}}$, $\boldsymbol{r}_{\mathcal{F}_{ik}}, \boldsymbol{r}_0$) forms the traditional VI-SFM problem as highlighted in the pink dashed box. The following sections detail the proposed depth constraints ($r_{\mathcal{L}_{ik}}, \boldsymbol{r}_{\mathcal{S}_i}$) which are grouped by green dashed box.

3.3 Weighted Mono-Depth Constraints

As illustrated in Fig. 3, depth constraints relate observed feature-point depth with that keyframe's scale-shift parameters, $\boldsymbol{S}_k$. Hence only 2 additional parameters are needed to model the hundreds of mono-depth residual equations for each keyframe-landmark pair. As demonstrated in Sec. 4, this improves the system conditioning under motion restricted scenarios.

The depth constraints comprise three major components - the **residual function**, the **weight** for each residual and the **outlier rejection** module to reject inconsistent mono-depth measurements across keyframes.

Inverse Depth Residual Function. Inspired by the loss functions employed in monocular deep depth estimation [56], our proposed depth residual for keyframe k and feature point i takes the form of the *log* of the ratio between the measured depth scaled/shifted by $\boldsymbol{S_k}$ and the feature point's estimated depth:

$$r_{\mathcal{L}_{ik}} = \log\left((a_k d_{ik} + b_k) \cdot \Omega(^{\boldsymbol{C_j}}\boldsymbol{f_i}, \boldsymbol{q_j}, \boldsymbol{p_j}, \boldsymbol{q_k}, \boldsymbol{p_k})\right) \tag{3}$$

Where $\Omega(\cdot)$ is the depth of the feature point i (which is parameterized with respect to keyframe j) in keyframe k. If $k = j$ then $\Omega(\cdot)$ can be simplified to w_{ij}^{-1}. This is how we tie mono-depth parameters to multiple features and poses to better constrain the problem.

It is well known that this residual can lead to a degenerate solution of scale going to zero or a negative value [57]. To avoid this, we adopt the common technique of defining the scale parameter a_k as

$$a_k = \varepsilon + log(e^{s_k} + 1) \tag{4}$$

where $\varepsilon = 10^{-5}$, which prevents a_k from being either negative or zero, allowing us to optimize s_k freely.

Scale-shift Prior. Reiterating Sec. 3.1, the ML model is trained on certain metric depth datasets with a loss where the scale is supposed to be 1 and shift is 0. We define prior residuals for scale and shift at the i^{th} frame as

$$\boldsymbol{r}_{\mathcal{S}_i} = \begin{bmatrix} 1 - a_i & -b_i \end{bmatrix}^T \tag{5}$$

Since metric depth is not observable from the ML model, in practice we assign a *very* large covariance $\Sigma_{\mathcal{S}}$ to these scale-shift priors terms (0.3 for scale, 0.2 for shift), which keeps parameters bounded to the regime in which model training occurred, and in degenerate situations such as zero-acceleration, allows us to converge to a sensible scale.

Fig. 1a shows the effectiveness of the depth constraints and scale-shift priors. With them, we are able to upgrade the learned depth to metric level. The better-conditioned problem then yields a more accurate trajectory, illustrated in Fig. 1b.

Edge Awareness Weight. The ML model doesn't explicitly yield prediction uncertainty, however, we empirically observe the uncertainty is larger near depth edges and propose a loss weight, λ_{ik}, which modulates the residual with gradients of image I_k and depth D_k as follows

$$\lambda_{ik} = e^{-(\alpha|\nabla^2\Phi(I_k(u_{ik},v_{ik}))|+|\nabla^2\Phi(D_k(u_{ik},v_{ik}))|)} \tag{6}$$

where ∇^2 is the laplacian operator, $\Phi(\cdot)$ is a bilateral filter for sharpening image and depth edges, α is a hyperparameter for relative weighting of image/depth gradients and (u_{ik}, v_{ik}) is the pixel location of the feature point in keyframe k. This weight diminishes the effect of depth constraints on feature points near image/depth edges and favors non-edge regions where the depth and image gradients are in agreement.

Algorithm 1 Outlier Depth Measurements Rejection

Input: Mono-depth residuals $r_{\mathcal{L}ik}, i \in \mathcal{L}, k \in \mathcal{K}$; thresholds $\sigma_{\min}, \sigma_{\max}$
Output: Set of inlier mono-depth residuals

$\sigma_{\mathcal{L}} \leftarrow \{\}$
for $i \in \mathcal{L}$ **do**
    Append $\sigma_i = \sqrt{\frac{\sum_k (r_{ik} - \hat{r}_i)}{N-1}}$ to $\sigma_{\mathcal{L}}$
end for
if percentile$(\sigma_{\mathcal{L}}, 25) > \sigma_{\max}$ **then**
    return $\{\}$
else if percentile$(\sigma_{\mathcal{L}}, 85) < \sigma_{\min}$ **then**
    return $\{r_{\mathcal{L}ik}, \forall i \in \mathcal{L}, \forall k \in \mathcal{K}\}$
else
    return $\{r_{\mathcal{L}ik} | \sigma_i < \text{percentile}(\sigma_{\mathcal{L}}, 85)\}$
end if

Outlier Rejection for Depth Measurements. The weighting function Eq. (6) helps mitigate effects of erroneous mono-depth measurements at a given keyframe, but cannot reconcile inconsistency in depth measurements across keyframes. For a short initialization window ($< 2s$), keyframe images tend not to vary drastically. Given this, we expect the mono-depth output to not vary significantly as well (even though they are up to an unknown scale and shift). For example, if the mono-depth model predicts a feature point to have small depth w.r.t the rest of the scene in one keyframe but large depth in another, the mono-depth residuals for this given feature are likely to be unreliable and should not be included in the final optimization.

Thus, we devise an outlier-rejection scheme detailed in Algorithm 1. This algorithm first evaluates the standard deviations of residuals involving a given feature point, $\sigma_{\mathcal{L}} = \{\sigma_i, \forall i \in \mathcal{L}\}$. Then depending on the distribution of $\sigma_{\mathcal{L}}$ we choose the inlier set. (i) If the 25^{th} percentile of $\sigma_{\mathcal{L}}$ is larger than a maximum threshold, we reject all mono-depth constraints. This scenario occurs when the ML inference is highly unstable and typically does not yeild useful constraints. (ii) When mono-depth constraints are generally self-consistent (the 85^{th} percentile of $\sigma_{\mathcal{L}}$ is smaller than a minimum threshold) we accept all mono-depth constraints. (iii) In all other cases, we reject residuals corresponding to σ_i in upper 15^{th} percentile of $\sigma_{\mathcal{L}}$, removing the least self-consistent constraints. Such a scenario is depicted in Fig. 1a, where the mono-depth residuals involving red feature tracks are rejected.

In practice, we require an up-to-scale accurate estimate of camera pose and feature position to evaluate $r_{\mathcal{L}ik}$ for input to Algorithm 1. Therefore, we first solve the VI-BA without mono-depth (i.e., the pink rectangle portion of Fig. 3). Finally after convergence of the depth-less cost-function, we add the depth constraints as detailed in this section, and solve Eq. (2).

4 Experiments

We perform two sets of experiments on the popular EuRoC dataset [39], containing visual and inertial data from a micro air vehicle (MAV) along with accurate motion ground truth. To generate reliable correspondences for visual and mono-depth constraints, our front-end uses gyro measurements as a prior for frame-to-frame rotations following 2-pt RANSAC [58]. We first exhaustively evaluate VIO initialization performance on the whole trajectory by running our initialization routine in windows sampled throughout each trajectory in the dataset, which is commonly done in a variety initialization works [7, 8, 30]. Additionally, we also evaluate the effect of initialization on tracking performance by employing our method on a baseline similar to OpenVINS [47] in $10s$ time windows distributed uniformly across datasets. In both cases, we compare against ground truth poses captured by a VICON system present in the dataset.

4.1 Exhaustive Initialization Evaluation

Table 1: Exhaustive initialization benchmark results per dataset from Inertial-only, our baseline, and our proposed method using 5 KFs with $10Hz$ image data. For each metric, lower is better.

	Scale Error (%) $\|\|\bar{a}\\|\| > 0.005G$			**Position RMSE (meters)**			**Gravity RMSE (degrees)**			log(**Condition Num**) $\|\|\bar{a}\\|\| < 0.005G$	
Dataset	**Inertial-only**	**Baseline**	**Ours**	**Inertial-only**	**Baseline**	**Ours**	**Inertial-only**	**Baseline**	**Ours**	**Baseline**	**Ours**
mh_01	41.34	43.65	**31.11**	0.047	0.035	**0.025**	**1.38**	2.43	1.82	13.97	**13.16**
mh_02	38.80	41.41	**34.98**	0.048	0.033	**0.026**	**1.33**	2.04	1.81	13.31	**12.50**
mh_03	57.44	59.09	**34.65**	0.145	0.091	**0.055**	3.09	3.73	**2.89**	13.83	**12.73**
mh_04	74.29	56.26	**48.40**	0.179	0.090	**0.075**	2.38	2.69	**2.31**	13.42	**11.27**
mh_05	70.35	54.64	**44.52**	0.145	0.078	**0.063**	**2.13**	2.77	2.30	13.66	**12.51**
v1_01	55.44	54.25	**25.59**	0.056	0.038	**0.021**	3.47	3.73	**3.36**	12.93	**11.43**
v1_02	56.86	45.12	**26.12**	0.106	0.069	**0.038**	3.77	3.86	**2.44**	13.26	**11.67**
v1_03	56.93	38.55	**20.01**	0.097	0.048	**0.025**	5.36	3.59	**2.37**	12.62	**12.03**
v2_01	42.40	40.84	**23.51**	0.035	0.026	**0.015**	1.49	1.78	**1.35**	13.45	**12.84**
v2_02	41.27	34.31	**19.33**	0.035	0.026	**0.015**	2.92	2.66	**1.96**	**12.20**	12.27
v2_03	59.64	36.42	**27.87**	0.116	0.044	**0.033**	4.10	2.81	**2.24**	13.30	**11.17**
Mean	54.07	45.87	**30.55**	0.092	0.053	**0.036**	2.86	2.92	**2.26**	13.27	**12.14**

Following prior related initialization works [7, 8, 30], we exhaustively create VIO initialization events across the whole trajectory to evaluate performance across different motion and visual scenarios. For a fair comparison, we split each dataset into segments evenly and attempt to initialize all methods on the same set of segments. We collect poses from all successful initializations for the evaluation, **though note:** not all trials are successful due to internal validation steps of the respective algorithms and success does not necessarily mean that the initialization poses are **qualified** for tracking. Accuracy may be poor (measured by scale error or RMSE), in which case tracking may diverge.

Our baseline method consists of a closed-form initialization [10] followed by VI-BA [59] with only the VI-SFM portion of residuals present (pink rectangle in

Fig. 3). We also compare against the state-of-the-art VI-initialization method Inertial-only [7], implementation of which is obtained from the open-sourced SLAM method [23]. Given N keyframes, Inertial-only uses up-to-scale visual odometry as the prior in a MAP framework to recover the metric scale, gravity vector, and IMU biases, followed by a VI-BA refinement step. Inertial-only's visual front-end performs RANSAC with PnP [60].

We configured all three methods to operate on 10Hz image streams following previous works [7–9]. We treat each image as keyframe and use either 5 or 10 keyframes (KFs) for initialization. In the 5KFs setting, we split datasets into $0.8s$ initialization windows evenly. We specifically highlight a 5KFs experiment to further exacerbate issues of insufficient baseline/motion, which are commonplace in deployment scenarios (e.g. MAVs, AR/VR). We were able to generate 1078, 1545, 1547, initialization trajectories respectively for Inertial-only, baseline, and our proposed method over all EuRoC datasets from 1680 initialization attempts. The average initialization trajectory latency for the three methods were $0.592s$, $0.399s$, and $0.399s$ respectively. For our 10KFs setting, we split datasets into $1.6s$ windows. We generated 571, 809, 815 initialization trajectories for the three methods with an average trajectory latency of 1.367, 0.897 and 0.897 from 839 initialization attempts. Since Inertial-only uses visual odometry as the prior, to better align with the resulting expectations across different methods, we rejected those trajectories with poor resulting reprojection error of each visual constraint for the baseline and our proposed method. We observed that Inertial-only had longer initialization latency and typically led to fewer successful initializations because it requires mean trajectory acceleration larger than 0.5% of gravity ($||\bar{a}|| > 0.005G$) as stated in [7].

To measure trajectory accuracy, we perform a $Sim(3)$ alignment against the ground truth trajectory to get scale error and position RMSE for each initialization. Since the global frames of the IMU sensor should be gravity-aligned, the gravity RMSE (in degrees) is computed from the global z axis angular deviation in the IMU frame. Following past work [7], we omit scale errors when the mean trajectory acceleration $||\bar{a}|| < 0.005G$, however gravity and position RMSE are still reported. Finally, we also empirically compute the condition number of the problem hessian in the most challenging of sequences (mean acceleration $||\bar{a}|| < 0.005G$) to evaluate problem conditioning with the added mono-depth constraints. We present our aggregated results for the 5KFs setting in Tab. 1. We significantly outperform state-of-the-art Inertial-only in all metrics and across datasets, achieving on average a 43% reduction in scale error, 61% reduction in position RMSE, and 21% reduction in gravity RMSE for the challenging $5KF$ setting at an initialization latency of $0.4s$. Furthermore, our formulation leads to a lower condition number compared to the baseline, indicating improved problem conditioning.

In Fig. 4, we plot the cumulative distributions for the metrics above for both the 10KFs (top) and 5KFs (bottom) settings. We can see that while we do better than the baseline and Inertial-only in the 10KFs setting, the gains are greater in the more challenging 5 KFs setting with reduced motion excitation,

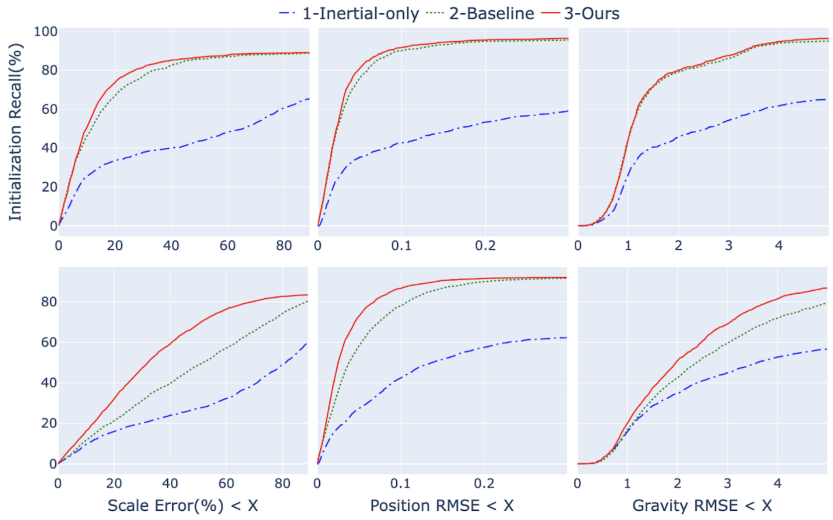


Fig. 4: Cumulative distribution plots for primary error metrics. **First row**: Results with 10 keyframes. **Second row**: Results with 5 keyframes. For each plot, the X axis denotes a threshold for error metric and the Y axis shows the fraction of initialization attempts with the respective error metric smaller than the threshold on the X axis. **Note:** 1) Improved gains in the 5KF (i.e. less motion) setting where mono-depth residuals show greater impact. 2) Recall doesn't converge to 100% due to initialization failures among attempts.

highlighting the benefit of the mono-depth residuals. In order to gain insights into where our method outperforms others, we visualize a dataset with trajectory color coded by acceleration magnitude and scale error for the various methods in Fig. 5. We outperform both Inertial-only and the baseline almost across the whole trajectory but more specifically so in low acceleration regions which are traditionally the hardest for classical VIO initialization methods. This further validates our hypothesis that the added mono-depth constraints condition the system better with direct (up to scale/shift) depth measurement priors in reduced motion scenarios, which is critical for today's practical applications of VIO.

4.2 Visual-inertial Odometry Evaluation

To better illustrate our method's in-the-wild applicability, we conduct experiments quantifying the impact of our method when used in-the-loop with odometry. Considering the additional challenge of 5KFs initialization, we focus our experiments there instead of typical 10KFs [7] and evaluate the accuracy of final tracking trajectories. The evaluation is performed with a baseline similar to OpenVINS [47], which is a state-of-the-art VIO system commonly used in compute-limited use-cases (e.g, mobile AR/VR, drones). Similar to Sec. 4.1, we create initialization events periodically but evaluate the tracking trajectories instead. We split the

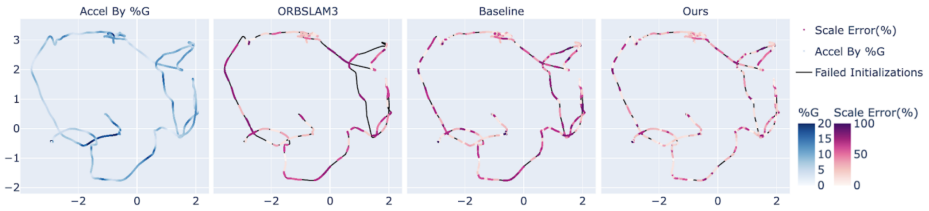


Fig. 5: Acceleration and scale error visualizations for the v2_01 dataset (best viewed in color). **Left:** Trajectory colored by acceleration magnitude as %G (lighter indicates low acceleration). **Right:** Segments of poses colored by scale error magnitude for each initialization window in the dataset (lighter is better). Segments colored black indicate failed initializations for the respective methods. We outperform other methods over the entire trajectory on scale error, especially in low acceleration regions where our method performs significantly better.

Table 2: Visual-inertial odometry benchmark results over all EuRoC datasets with and without mono-depth constraints used in initialization. VIO runs at $10Hz$ and is initialized with 5KFs.

	Position RMSE (m)			**Gravity RMSE (deg)**		
Dataset	**Baseline**	**Ours**	**Diff(%)**	**Baseline**	**Ours**	**Diff(%)**
mh_01	1.560	**0.543**	-65.19	2.21	**1.55**	-29.86
mh_02	0.604	**0.071**	-88.24	1.65	**1.31**	-20.60
mh_03	2.466	**1.299**	-47.32	2.88	**2.29**	-20.48
mh_04	0.526	**0.124**	-76.42	2.01	**1.01**	-49.75
mh_05	3.204	**0.910**	-71.59	3.44	**1.88**	-45.34
v1_01	3.438	**0.082**	-97.61	4.66	**2.69**	-42.27
v1_02	2.846	**0.097**	-96.59	3.57	**1.22**	-65.82
v1_03	2.649	**0.059**	-97.77	3.19	**1.28**	-59.87
v2_01	1.824	**0.046**	-97.47	2.19	**1.08**	-50.68
v2_02	2.615	**0.060**	-97.70	3.42	**1.25**	-63.45
v2_03	2.939	**0.567**	-80.70	3.99	**2.06**	-48.37
Mean	2.243	**0.351**	-84.35	3.02	**1.61**	-46.68

datasets evenly into $10s$ segments and initialize and perform VIO using the same $10s$ of information for both methods.

As in Sec. 4.1, our baseline is tracking initialized with VI-SFM only. We generated a total of 142 trajectories using our protocol over all EuRoC datasets for each method and report aggregated position and gravity RMSE for each dataset. The aggregated results are shown in Tab. 2 where we see an 84% improvement in position RMSE and 46% improvement in gravity RMSE over the baseline method. This suggests a significant expected improvement in downstream uses of odometry, such as rendering virtual content, depth estimation, or navigation.

Computation Cost. We ran our system on a Pixel4XL mobile phone using only CPU cores. The computation cost (in milliseconds) for different initialization modules is shown in Tab. 3. The closed-form initialization problem is solved using Eigen [61] and the subsequent VI-BA is solved with the Ceres Solver [62] using

Table 3: Computation duration of key modules in milliseconds.

Mono depth	Closed-form Initialization	VI-BA Solver (baseline)	VI-BA Solver (ours)
71.64	0.73	16.2	39.8

Levenberg–Marquardt. We run ML inference on the CPU in its own thread and hence achieve real-time performance (within $100ms$ for the $10Hz$ configuration) on a mobile phone. While we do observe that adding depth constraints increases the computational cost of the VI-SFM problem, we still improve in terms of overall initialization speed by producing a satisfactory solution with only 5KFs (**0.5s of data**) as opposed to 10KFs typically required by the baseline and Inertial-only.

5 Conclusion

In this paper, we introduced a novel VIO initialization method leveraging learned monocular depth. We integrated the learned depth estimates, with alignment parameters, into a classical VI-SFM formulation. Through the learned image priors, our method gains significant robustness to typical degenerate motion configurations for VI-SFM, such as low parallax and near-zero acceleration. This method only requires a lightweight ML model and additional residuals (with associated states) to be added to a standard pipeline and does not significantly impact runtime, enabling application to mobile devices. Our experiments demonstrated significant improvements to accuracy, problem conditioning, and robustness relative to the state-of-the-art, even when significantly reducing the number of keyframes used and exacerbating the problem of limited motion excitation. Our method could serve as a straightforward upgrade for most traditional pipelines.

There are several key limitations and directions for future work to call out:

- We do not claim any direct upgrades to VI system observability. While the use of a prior on scale and shift and the training of the mono-depth network (assuming scale and shift being 1 and 0) may provide some direct scale information, our work's primary contribution is to problem conditioning and behaviour under *reduced* motion, not *zero* motion.
- Mono-depth has generalization limitations due to biases in its training data, learning scheme, and model structure. It is crucial to note that we **did not re-train** our network for EuRoC. It was used *off the shelf* after training on general imagery which are very different from EuRoC. With a network trained specifically for the problem domain (or optimized in the loop at test time per initialization window) we expect an even greater improvement.

References

1. Santiago Cortes, Arno Solin, Esa Rahtu, and Juho Kannala. Advio: An authentic dataset for visual-inertial odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
2. Eagle Jones, Andrea Vedaldi, and Stefano Soatto. Inertial structure from motion with autocalibration. In *Workshop on Dynamical Vision*, volume 25, page 11, 2007.
3. Kejian J Wu, Chao X Guo, Georgios Georgiou, and Stergios I Roumeliotis. Vins on wheels. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5155–5162. IEEE, 2017.
4. Joshua Hernandez, Konstantine Tsotsos, and Stefano Soatto. Observability, identifiability and sensitivity of vision-aided inertial navigation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2319–2325. IEEE, 2015.
5. Agostino Martinelli. Closed-form solution of visual-inertial structure from motion. *International journal of computer vision*, 106(2):138–152, 2014.
6. Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.
7. Carlos Campos, José MM Montiel, and Juan D Tardós. Inertial-only optimization for visual-inertial initialization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 51–57. IEEE, 2020.
8. David Zuñiga-Noël, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez. An analytical solution to the imu initialization problem for visual-inertial systems. *IEEE Robotics and Automation Letters*, 6(3):6116–6122, 2021.
9. Tong Qin and Shaojie Shen. Robust initialization of monocular visual-inertial estimation on aerial robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4225–4232, 2017.
10. M. Li and A. I. Mourikis. A convex formulation for motion estimation using visual and inertial sensors. In *Proceedings of the Workshop on Multi-View Geometry, held in conjunction with RSS*, Berkeley, CA, July 2014.
11. Jacques Kaiser, Agostino Martinelli, Flavio Fontana, and Davide Scaramuzza. Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation. *IEEE Robotics and Automation Letters*, 2(1):18–25, 2017.
12. Raúl Mur-Artal and Juan 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.
13. Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE transactions on robotics*, 30(1):177–187, 2013.
14. Alejo Concha and Javier Civera. RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system. *CoRR*, abs/1703.00754, 2017.
15. Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems*, 2015.
16. René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
17. René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

18. Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics Automation Magazine*, 18(4):80–92, 2011.
19. Guoquan Huang. Visual-inertial navigation: A concise review. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9572–9582, 2019.
20. Mingyang Li and Anastasios I Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.
21. Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *CoRR*, abs/1708.03852, 2017.
22. Konstantine Tsotsos, Alessandro Chiuso, and Stefano Soatto. Robust inference for visual-inertial sensor fusion. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5203–5210. IEEE, 2015.
23. Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021.
24. Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.
25. Lukas Von Stumberg, Vladyslav Usenko, and Daniel Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2510–2517. IEEE, 2018.
26. Xiaohan Fei and Stefano Soatto. Xivo: An open-source software for visual-inertial odometry. `https://github.com/ucla-vision/xivo`, 2019.
27. Zheng Huai and Guoquan Huang. Robocentric visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6319–6326. IEEE, 2018.
28. Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.
29. Carlos Campos, J. M. M. Montiel, and Juan D. Tardós. Fast and robust initialization for visual-inertial SLAM. *CoRR*, abs/1908.10653, 2019.
30. Jinyu Li, Hujun Bao, and Guofeng Zhang. Rapid and robust monocular visual-inertial initialization with gravity estimation via vertical edges. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6230–6236, 2019.
31. Chao X. Guo and Stergios I. Roumeliotis. Imu-rgbd camera 3d pose estimation and extrinsic calibration: Observability analysis and consistency improvement. In *2013 IEEE International Conference on Robotics and Automation*, pages 2935–2942, 2013.
32. Liming Han, Yimin Lin, Guoguang Du, and Shiguo Lian. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6906–6913. IEEE, 2019.
33. Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050. IEEE, 2017.
34. Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

35. Yasin Almalioglu, Mehmet Turan, Alp Eren Sari, Muhamad Risqi U. Saputra, Pedro Porto Buarque de Gusmão, Andrew Markham, and Niki Trigoni. Selfvio: Self-supervised deep monocular visual-inertial odometry and depth estimation. *CoRR*, abs/1911.09968, 2019.
36. Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10542–10551, 2019.
37. Chunshang Li and Steven L Waslander. Towards end-to-end learning of visual inertial odometry with an ekf. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 190–197. IEEE, 2020.
38. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
39. Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.
40. Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.
41. Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, amp; new methods. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3146–3152, 2020.
42. Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
43. Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2560–2568, 2018.
44. Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment networks. In *International Conference on Learning Representations*, 2018.
45. Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
46. Xingxing Zuo, Nathaniel Merrill, Wei Li, Yong Liu, Marc Pollefeys, and Guoquan Huang. Codevio: Visual-inertial odometry with learned optimizable dense depth. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14382–14388. IEEE, 2021.
47. Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. Openvins: A research platform for visual-inertial estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4666–4672. IEEE, 2020.
48. Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

49. Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.
50. Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.
51. Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.
52. Javier Civera, Andrew J Davison, and JM Martinez Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5):932–945, 2008.
53. Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration theory for fast and accurate visual-inertial navigation. *CoRR*, abs/1512.02363, 2015.
54. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
55. Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.
56. David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014.
57. Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. *CoRR*, abs/1904.05822, 2019.
58. Chiara Troiani, Agostino Martinelli, Christian Laugier, and Davide Scaramuzza. 2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5530–5536, 2014.
59. Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.
60. Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.
61. Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.
62. Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.