VOLDOR⁺SLAM:

For the times when feature-based or direct methods are not good enough

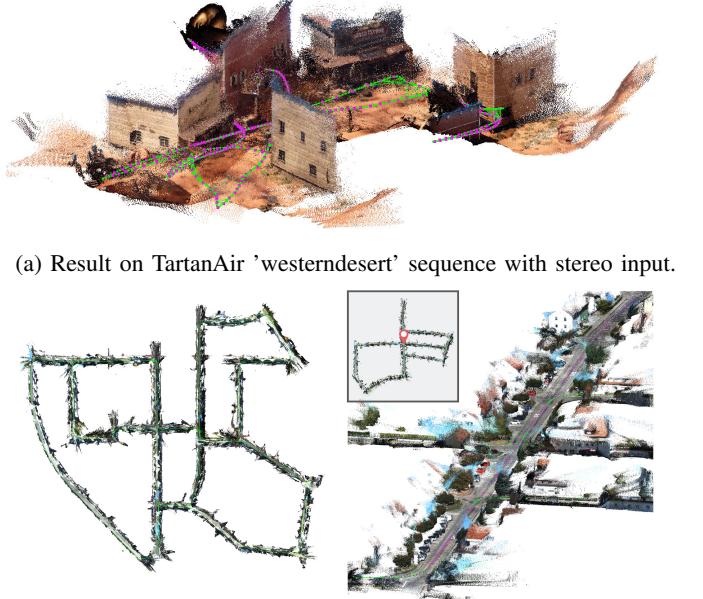Zhixiang Min[1] and Enrique Dunn[1]

Abstract—We present a dense-indirect SLAM system using external dense optical flows as input. We extend the recent probabilistic visual odometry model VOLDOR [1], by incorporating the use of geometric priors to 1) robustly bootstrap estimation from monocular capture, while 2) seamlessly supporting stereo and/or RGB-D input imagery. Our customized back-end tightly couples our intermediate geometric estimates with an adaptive priority scheme managing the connectivity of an incremental pose graph. We leverage recent advances in dense optical flow methods to achieve accurate and robust camera pose estimates, while constructing fine-grain globally-consistent dense environmental maps. Our open source implementation [https://github.com/htkseason/VOLDOR] operates online at around 15 FPS on a single GTX1080Ti GPU.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) addresses the incremental construction and instantaneous establishment of a global geometric reference using measurements from a dynamic observer as input. Applications benefiting from such capabilities include robotics, augmented/virtual reality, and autonomous driving. For visual SLAM, the choice of geometric representation and estimation framework must balance requirements for online operation (e.g. latency and computational burden) and achievement of task-relevant performance metrics (e.g. robustness, accuracy, and completeness). Recent dense-indirect methods challenge the standard dichotomy among feature-based and direct SLAM methods.

Feature-based methods determine multi-view relationships among input video frames through the geometric analysis of sparse key-point correspondences. Popular feature-based SLAM systems [2], [3] have a visual odometry (VO) front-end solving feature correspondence/triangulation and PnP instances to initialize camera pose estimates, which are subsequently refined (e.g. bundle-adjusted) by back-end and mapping modules. Given that sparse feature analysis can operate on real-time, the trade-offs between online operation and accuracy/robustness achieved by these methods is mainly determined by the scope of the analysis performed by the back-end. However, the accuracy of front-end modules may be affected by lack of texture content, repetitive structures, and/or degenerate geometric configurations; compromising the efficacy and efficiency of back-end modules.

Direct methods jointly estimate photometrically consistent scene structure and camera poses from image content, obviating key-point correspondence. Dense direct methods [4] are typically sensitive to illumination changes, and need a sufficient number of images to triangulate structure at poorly

[1]Stevens Institute of Technology {zmin1,edunn}@stevens.edu

(a) Result on TartanAir 'westerndesert' sequence with stereo input.

(b) Result on KITTI 00, 05 sequence with stereo input.

Fig. 1: Reconstructed scene models. 3D point clouds are aggregated from keyframe depth maps.

textured regions. Sparse or semi-dense direct methods [5], [6] are less sensitive to illumination changes, but may still require accurate photometric calibration for full performance.

Recent dense-indirect (DI) formulations for VO [1] are a promising alternative to both their sparse and direct counterparts. The DI approach conditions local 3D geometry and camera motion estimates on their consistency w.r.t. (observed) dense optical flow (OF) estimates. Such frameworks actively leverage ongoing improvements in accuracy and robustness of learning-based OF estimators, while yielding high-quality dense geometry estimates as a byproduct.

To extend the recent DI inference model VOLDOR [1] into a SLAM pipeline, we develop integrative modules customized to the intermediate geometric estimates produced by its inference process. We address the efficient estimation, management and refinement of global-scope 3D data associations within the context of DI geometric estimates. Our contributions are: 1) extending VOLDOR's inference framework to both generate and input explicit geometric priors for improved estimation accuracy across monocular, stereo and RGB-D image sources, 2) a robust point-to-plane inverse-depth alignment formulation for source-agnostic frame registration, and 3) a priority linking framework for incremental real-time pose graph management.

II. RELATED WORK

Indirect SLAM. Classic approaches like MonoSLAM [7] rely on EKF-based frameworks for fusing multiple observations to recover scene geometry. PTAM [8] simultaneously runs a front-end for VO and a back-end refining the estimation through bundle adjustment. ORB-SLAM [9], [3], introduced a versatile SLAM system with a more powerful back-end with global re-localization and loop closing, allowing large environment applications. More recently, ORB-SLAM3 [10] further built a multi-map system to survive from long periods of poor visual information.

Direct SLAM. DTAM [4] first introduced a GPU-based real-time dense modelling and tracking approach for small workspaces. It estimates a dense geometry model through a photometric term combined with a regularizer while estimating camera motion by finding a warping of the model that minimizes the photometric error w.r.t. the video frames. LSD-SLAM [5] switched to keyframe based semi-dense model that allows large scale CPU real-time application. DSO [6] builds sparse models and combines a probabilistic model that jointly optimizes for all parameters as well as further integrates a full photometric calibration.

Direct RGB-D SLAM. RGB-D SLAM [11] forgoes geometry estimation by using RGB-D input, while estimates camera poses by minimizing both photometric and geometric error, and managing keyframes within a pose graph. ElasticFusion [12] uses dense frame-to-model camera tracking, windowed surfel-based fusion, and frequent model refinement through non-rigid surface deformation. BundleFusion [13] tracks camera poses using sparse-to-dense optimization and a local-to-global strategy for global mapping consistency. Recently, BAD-SLAM [14] proposed a fast direct bundle-adjustment for poses, surfels and intrinsics.

Dense-Indirect VO. Valgaerts et al. [15] addressed robust recovery of fundamental matrix from a single dense OF field. Ranftl et al. [16] further addressed the motion segmentation and the recovery of depth maps. While these works highlighted the potential of DI modeling, they focused on the two-view geometry problem. VOLDOR [1] introduced the DI paradigm to the multi-view domain by modelling the VO problem within a probabilistic framework w.r.t. a log-logistic residual model on OF batches and developed a generalized-EM framework for joint real-time inference of camera motion, 3D structure and track reliability. While highly accurate, geometric analysis scope was strictly local.

Deep learning optical flow. Current deep learning works on OF estimation outperform traditional methods in terms of accuracy and robustness under challenging conditions such as texture-less regions, motion blur and large occlusions. FlowNet/FlowNet2 [17], [18] introduced an encoder-decoder CNN for OF estimation. PWC-Net [19] integrates spatial pyramids, warping and cost volumes into deep OF estimation, improving performance and generalization. Most recently, MaskFlowNet [20] proposed an asymmetric occlusion aware feature matching module and achieved current state-of-the-art performance.

III. METHOD

System Architecture. We build upon the recently proposed dense indirect VO method of Min et al. [1], which addressed the joint probabilistic estimation of camera motion, 3D structure and track reliability from a set of input dense OF estimates. As standard practice in SLAM literature [3], [6], our proposal VOLDOR$^+$SLAM has a VO front-end and a mapping back-end. The front-end operates over small sequential batches (i.e. temporal sliding window) of dense OF estimates. We adaptively determine keyframe selection and stride among subsequent batches based on visibility-coverage metrics designed for our dense SLAM system. To enforce consistency within a larger geometric scope, while enabling online operation, the back-end adaptively prioritizes the analysis and establishment pose constraints between keyframe pairs balancing the search for loop closure connections and the reinforcement of local keyframe connectivity. VOLDOR$^+$SLAM also implements a loop closure scheme based on an image retrieval and geometric verification module [21]. Finally, all pairwise camera pose constraints are managed within a $sim(3)$-based pose graph [22]. Module dependencies and data flow of our system are illustrated in Fig.2. The resulting dense SLAM implementation enables online operation at $\sim$15 FPS on a single GTX1080Ti GPU.

A. Front-End

VOLDOR$^+$: An extended VO model. Our front-end extends and improves upon the VOLDOR framework [1], which formulates monocular VO from observed OF input batches as a generalized EM problem where the three hidden variables are camera pose, depth, and track reliability. Originally, [1] proposed the probabilistic graphical model

$$P(\mathbb{X} \mid \mathbb{T}, \boldsymbol{\theta}, \mathbb{W}) \qquad (1)$$

where $\mathbb{X}=\{\boldsymbol{X_t} \mid t=1,\cdots,N_t\}$ denotes the observed OF batch, $\mathbb{T}=\{\boldsymbol{T_t} \mid t=1,\cdots,N_t\}$ the set of camera poses, $\boldsymbol{\theta}$ the depth map of the first frame in the batch and $\mathbb{W}=\{\boldsymbol{W_t} \mid t=1,\cdots,N_t\}$ the set of rigidness maps used to down-weight OF pixels belonging to occlusions, moving objects or outliers. The likelihood function in [1] defines residuals as the endpoint-error (EPE) between the observed optical flow vs. the rigid flow computed from the parameters. This residual is assumed to follow an empirically validated log-logistic (i.e. Fisk) distribution model. The ensuing MAP problem is solved using a generalized-EM framework alternatively updating each of the parameters. Whereas [1] bootstraps each of the hidden variables deterministically, VOLDOR$^+$ accommodates explicit priors, extending Eq.(1) to yield

$$P(\mathbb{X}, \hat{\boldsymbol{\Theta}} \mid \mathbb{T}, \boldsymbol{\theta}, \mathbb{W}, \hat{\mathbb{W}} ; \hat{\mathbb{T}}), \qquad (2)$$

where the set of explicit depth map priors is denoted by $\hat{\boldsymbol{\Theta}}=\{\hat{\boldsymbol{\theta}_k} \mid k=1,\cdots,N_k\}$; their associated (*fixed*) relative camera poses by $\hat{\mathbb{T}}=\{\hat{\boldsymbol{T}_k} \mid k=1,\cdots,N_k\}$ and their pixel-level rigidness maps by $\hat{\mathbb{W}}=\{\hat{\boldsymbol{W}_k} \mid k=1,\cdots,N_k\}$. Henceforth, unless otherwise stated, our VOLDOR$^+$ extensions strictly follow the framework described in [1] and we refer the reader to that publication for further details.
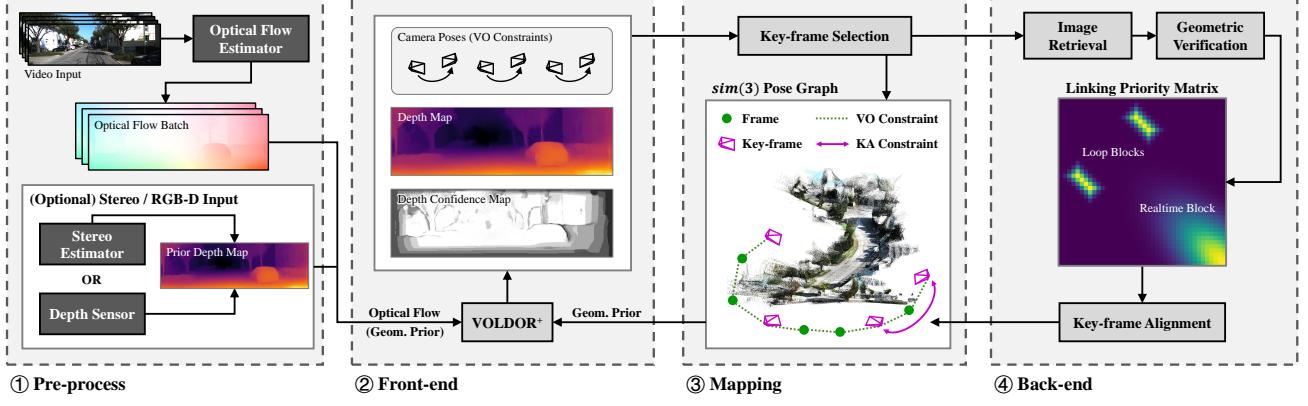
Fig. 2: VOLDOR$^+$SLAM architecture. 1) Input optical flow is externally computed from video, along with optional geometric priors. 2) Dense-indirect VO front-end estimates scene structure and local camera poses over a sliding window. 3) A pose graph enforces global consistency among all pairwise pose estimates. 4) The set of edges to include in our pose graph is prioritized based on keyframe geometric analysis aimed at both identifying loop closures and reinforce local connectivity.

Geometric Priors. While [1] estimates depth maps exclusively from monocular OF fields, VOLDOR$^+$ extends the probabilistic model to optionally account for input depth map priors. Limiting our assumptions (for now) on the source of these priors, to knowing their first and second moments, we model a generic likelihood function of the depth value at pixel j as a Gaussian-Uniform mixture

$$P(\hat{\theta}_k^\ell \mid \theta^j, \hat{W}_k^j; \hat{T}_k) = \begin{cases} \mathcal{N}_{\sigma\left(1/\hat{\theta}_k^\ell\right)}^0 \left(\frac{1}{\hat{\theta}_k^\ell} - \frac{1}{\phi_z(\theta^j, \hat{T}_k)}\right) & \text{if } \hat{W}_k^j = 1 \\ \mathcal{U}\left(\frac{1}{\hat{\theta}_k^\ell}\right) & \text{if } \hat{W}_k^j = 0 \end{cases}$$
(3)

where $\phi_z(\theta^j, \hat{T}_k)$ denotes the Z-component of the camera-space 3D coordinates of a depth value transferred across frames having relative motion T_k, $\ell = \pi(\theta_2^j, \hat{T}_k)$ denotes the pixel of the reprojection π of a depth estimate across camera frames, while $\sigma(\cdot)$ and $\mathcal{U}(\cdot)$ are functions adjusting the variance of a Gaussian distribution and the density of a uniform distribution w.r.t. the observed depth prior, which are set in proportion to inverse depth value. As in [1], we apply the likelihood function to a maximum inlier estimation framework to define the energy function for prior depths

$$E_{\text{geom-prior}} = -\sum_k q(\hat{W}_k^j) \, log \frac{P(\hat{\theta}_k^\ell \mid \theta^j, \hat{W}_k^j = 1)}{\sum_{\hat{W}_k^j} P(\hat{\theta}_k^\ell \mid \theta^j, \hat{W}_k^j)}, \quad (4)$$

where $q(\hat{W}_k^j)$ is the probability density of $\hat{W}_k^j$. Finally, aggregating with the original energy function conditioned on input OF described in [1], optimal depth values are selected based on the criterion

$$\theta_{\text{opt}}^j = \underset{\theta^{j*}}{argmin}(E_{\text{optical-flow}} + E_{\text{geom-prior}}) \quad (5)$$

In practice each VO batch will typically have two prior depth maps: One generated during the analysis of the previous batch and another from the most recent available keyframe. Whenever both instances point to the same depth map, we will use that single depth prior. When stereo capture (or RGB-D imagery) is available, we add the corresponding depth map as a prior. However, in such case (i.e. depth priors come from known external source), VOLDOR$^+$ replaces Eq.(3) by an empirical residual model as done in [1].

Depth Map Confidence Estimation. For each depth map θ generated through the VOLDOR$^+$ framework, we associate a confidence map C, which we define as the pixel-wise average of previously estimated rigidness of optical flows and depth priors used for its estimation:

$$C^j = \frac{\sum_t W_t^j + \sum_k \hat{W}_k^j}{N_t + N_k} \quad (6)$$

When estimated depth maps are subsequently used as depth priors, the value C^j is used as a prior for $\hat{W}_k^j$ in Eq.(3), such that a new weight $C^j \hat{W}_k^j$ will replace $\hat{W}_k^j$ in the depth update step. Similarly, C^j is also used in the back-end keyframe alignment as will be described in Eq.(12).

Uncertainty-aware Pose Estimation. The probabilistic camera pose inference in [1] assumes other parameters fixed and approximates the camera pose posterior distribution by Monte-Carlo sampling of P3P instances

$$P(\mathbb{T} \mid \mathbb{X}; \theta, \mathbb{W}) \approx \prod_t \left[\frac{1}{S} \sum_g^S q(T_t^g) \right] \quad (7)$$

where t denotes the time stamp, S the total number of samples, g the sample index, and $q(T_t^g)$ is a variational distribution approximating the intractable sample posterior. Further, let $q(T_t^g)$ be a normal distribution $\mathcal{N}(\hat{T}_t^g, \Sigma)$, where $\hat{T}_t^g$ is solved using a AP3P solver [23], while the covariance matrix Σ is a fixed hyper-parameter. The camera pose defined by the mode of the approximated posterior, is found using meanshift with a Gaussian kernel in Lie algebra. To accommodate subsequent back-end modules, VOLDOR$^+$ estimates each camera pose uncertainty Σ_t by using the meanshift result as initialization and iteratively fitting a Gaussian distribution to the samples. We discard samples outside 3-sigma for robustness and force all eigenvalues of Σ_t to be larger than an epsilon to ensure numerical stability.
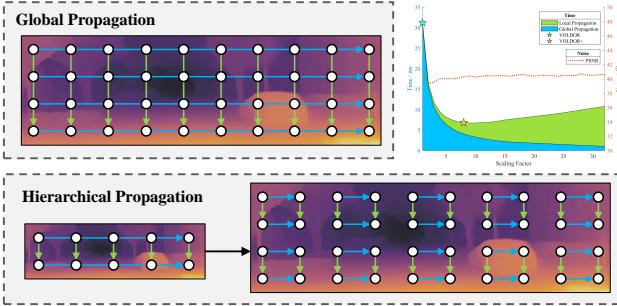
Fig. 3: Depth propagation workflow. Top right shows runtime (ms) vs depth map quality (PSNR) trade-offs of our hierarchical propagation scheme across varying scaling factors.

Hierarchical Propagation Scheme. For estimating depth and rigidness maps, [1] adopts a sampling and propagation scheme [24] based on a generalized-EM framework. The image 2D field is broken into alternatively directed 1D chains. In the M-step, depth values are randomly sampled and propagated through each chain. In the E-step, hidden Markov chain smoothing is imposed on the chains of rigidness maps using the forward-backward algorithm. Such sequential global depth propagation scheme imposes a performance bottleneck for GPU implementations, as it is not massively parallelizable. Hence, VOLDOR$^+$ proposes a hierarchical propagation scheme, where a global propagation is done on a depth map of reduced scale to ensure convergence rate, and local propagations done in local windows at full resolution to retain fine details. Per Fig.3, VOLDOR$^+$ achieves 3x speedup w/o loss in depth map quality w.r.t. [1].

VO Stride and Keyframe Selection. Once VOLDOR$^+$ runs on an input batch and estimates local 3D geometries, we determine the stride to the next batch and select keyframes using visibility-coverage (VC) metrics. The VC metrics are defined over a pair of frames I_1, I_2 with known poses, where only I_1 may have a depth map. We define visibility score as the proportion of depth pixels in I_1 whose projection in I_2 is inside I_2's image frame. Conversely, we define the coverage score as the proportion of the image area in I_2 being covered by the depth pixels' projection from I_1. We define the VC score as the harmonic mean of visibility and coverage

$$VC = \frac{2}{Visibility^{-1} + Coverage^{-1}} \quad (8)$$

The VC score is estimated between the reference frame and all other frames within the batch. The first frame with VC score below τ_{stride} is selected as the next batch's reference. Finally, if the VC score between the current batch's reference frame and the latest keyframe is below $\tau_{keyframe}$, we register the current batch's reference frame as a keyframe.

B. Back-End

The front-end VO only establishes local pair-wise constraints between successive frames I_t and I_{t+1} within an input batch. Conversely, the back-end manages spatial relationships of larger scope by establishing spatial constraints between keyframes and detecting loop closure instances.

Keyframe Alignment. We establish keyframe alignment (KA) constraints by 3D registration of their depthmaps. That is, for two depth maps θ_1 and θ_2, we estimate both their relative motion $\boldsymbol{T}$ and scale factor s. Similar to [25], we formulate the depth image alignment objective in terms of the difference of the inverse depth values

$$E_{geo} = \sum_j C^j \, \psi \left(\frac{1}{s\theta_1^\ell} - \frac{1}{\phi_z(\theta_2^j, \boldsymbol{T})} \right) \quad (9)$$

where $\psi(\cdot)$ is the Cauchy kernel function, C^j is the confidence associated with pixel j defined in Eq.(6). While effective, the convergence rate of Eq.(9) impeded online operation. Towards this end, we generalize the point-to-plane error to account for an inverse depth parameterization:

$$E_{geo} = \sum_j C^j \, \psi \left(\frac{\langle \mathbf{n}\left(\theta_1^\ell\right), \, \phi(\theta_1^\ell, \boldsymbol{I}) - \phi(\theta_2^j, \boldsymbol{T}) \rangle}{s\theta_1^\ell \phi_z(\theta_2^j, \boldsymbol{T})} \right) \quad (10)$$

where $\langle \cdot, \cdot \rangle$ denotes the vector inner product, while $\mathbf{n}(\theta_i^j)$ denotes the normal vector at pixel j of depth map $\boldsymbol{\theta}_i$. Such objective implicitly enforces spatial regularization given that $\mathbf{n}(\theta_i^j)$ is computed from the local depth estimates and significantly increases convergence speed and quality. Geometrically, scaling pair-wise point-to-plane distances among depth estimates θ_1^ℓ and θ_2^j inversely proportional to their depths, prioritizes geometry nearby to the reference frame. Conversely, weighting each kernel output by our confidence measure, mitigates depth outliers. Optionally, our keyframe alignment process can benefit from the input intensity images by adding the energy function for photometric consistency:

$$E_{photo} = \sum_j C^j \, \psi \left((I_1^\ell - b_1) - \frac{e^{a_1}}{e^{a_2}}(I_2^j - b_2) \right) \quad (11)$$

where a_1, a_2, b_1, b_2 are the pixel brightness affine transformation parameters to be estimated, while I_i^j yields the intensity value of pixel j in image I_i. The photometric term is optional since we favor our system to be an indirect method, and rely on external OF modules to handle common intensity-based imaging aberrations such as exposure and/or white-balance variations, non-Lambertian reflections, etc. Also, VOLDOR$^+$ estimates depth maps from small batches of successive imagery, while temporally distant observations are subject to arbitrary changes in appearance (i.e. global illumination, shadows, specularities) even if their underlying geometry is consistent. Moreover, our depth inference framework targets the estimation of static geometry observed throughout a multi-frame input batch. Empirically, we observe it consistently "looks through" dynamic or small foreground structures observed in the batch (first) reference frame and estimates the background's depth. For such cases, the photometric term may reduce the overall accuracy. Finally, the criteria for keyframe alignment is

$$\{\boldsymbol{T}, s\} = \underset{\boldsymbol{T}^*, s^*}{argmin}(E_{geo} + \lambda \cdot E_{photo}) \quad (12)$$

where λ balances geometry and photometric errors, which are omitted when intensity images are absent. The relative

depth scaling factor s is set to 1 for depth input with absolute scale (e.g. stereo or RGB-D). Once Eq.(12) is optimized using Levenberg-Marquardt, the covariance of $\boldsymbol{T}$ is linearly approximated through its Jacobian matrix as $\boldsymbol{\Sigma} = (\boldsymbol{J}^\top \boldsymbol{J})^{-1}$.

Priority Linking. The aggregation of all pairwise geometric constraints among keyframes yields a fully connected graph (i.e. quadratic complexity). To enable online operation, we avoid exhaustive exploration of these relationships by devising an adaptive prioritization mechanism aimed at: 1) Linking keyframes with sufficient overlap for robust and accurate depth map 3D alignment 2) Balancing exploration for potential loop closures vs. exploiting local connectivity. While loop closure benefits are self-evident, strengthening local connectivity fosters pose corrections to the current keyframe, which may be used as geometric priors in the VOLDOR$^+$ front-end, contributing to more accurate VO estimates. We construct a priority matrix $\mathbf{Q}$ for pairwise keyframe linkage, based on the observation that temporal proximity serves as a low-cost proxy for VC scores. That is, the distance between indexes roughly encodes co-visibility under the assumption of smooth local motion. Accordingly, we systematically update the entries in $\mathbf{Q}$ based on two linkage types (i.e. realtime linkage and loop closure linkage). First, for realtime linkages, after the current keyframe with index κ is created, we update the priority of each arbitrary keyframe pair with indices i, j, as

$$\dot{Q}_{i,j} = exp\left(-\frac{(i-j)^2}{\sigma_{spatial}^2} - \frac{(\kappa-i)(\kappa-j)}{\sigma_{temporal}^2}\right), \quad (13)$$

where the parameter $\sigma_{spatial}$ controls the priority w.r.t. co-visibility (proximal keyframes correlate to larger shared overlaps), while $\sigma_{temporal}$ controls the priority w.r.t. the timeliness of considered keyframe pair (we prioritize recent keyframes over older ones). Second, to foster the inclusion of loop closure linkages, we run an image retrieval engine based on the DBoW3 [21] library, using the current keyframe κ as query, to obtain a candidate keyframe κ'. Assume $\mathcal{L}^{th}$ pair (κ, κ') passes a geometric verification test, whenever $|i - \kappa| \leq |j - \kappa|$, we compute the linking priorities in the proximity of the detected overlap as:

$$Q_{i,j}^{\mathcal{L}} = Q_{j,i}^{\mathcal{L}} = exp\left(-\frac{(|i-\kappa| + |j-\kappa'|)^2}{\sigma_{lc}^2}\right) \quad (14)$$

where σ_{lc} controls the scope of priority propagation surrounding the loop closure pair. Given N_{lc} total detected loop closure pairs, the final linking priority matrix $\mathbf{Q}$ is

$$Q_{i,j} = max\left(\dot{Q}_{i,j}, \ max(Q_{i,j}^{\mathcal{L}} \mid \mathcal{L} = 1 \dots N_{lc})\right) \quad (15)$$

The largest unlinked element in $\mathbf{Q}$ larger than τ_{link} will be processed by the keyframe alignment module to determine an accurate relative motion among their associated depth maps. Finally, the selected pairwise motion constraint is incorporated into a $sim(3)$ pose graph framework [22], which enforces both globally consistent correction propagation (i.e. from loop closure linkages) as well as local refinements (i.e. from realtime linkages).
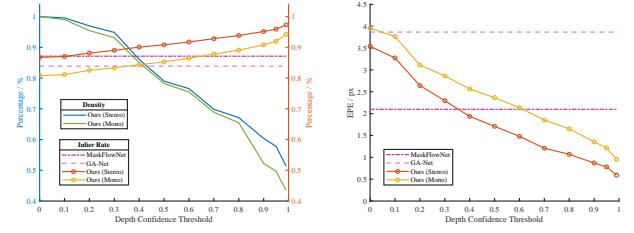

Fig. 4: Depth map quality evaluated on TartanAir dataset. We threshold over depth confidence value to evaluate pixel density, inlier rate and EPE across different confidence levels.

IV. EXPERIMENTS

Experimental Setup. We benchmarked on the photo-realistic synthetic dataset TartanAir[27], which provides ground-truth camera poses, depth maps and optical flows. The dataset features challenging environments in the presence of moving objects, changing light, various weather conditions, with diverse viewpoints and motion patterns that are difficult to obtain in real world. We tested on the "hard" track of 9 sequences covering diverse indoor and outdoor environments. Our *off-the-shelf* OF estimator MaskFlowNet [20] was not trained or fine-tuned on the TartanAir dataset. For stereo input, we use the X-component of the OF estimated from the left to the right camera as the disparity map, enabling us to use the same empirical residual model as described in [1] both for stereo and optical flow. For all sequences, we use the photometric consistency term of Eq.(11). We compared VOLDOR$^+$SLAM (full pipeline) and VOLDOR$^+$ (VO-only) vs. ORB-SLAM3 [10] (stereo and monocular) and DSO [6] (monocular).

Stereo Results. Per Table.I (left) ORB-SLAM3 excels on stable environments with enough textures such as 'office' and 'carwelding', but accuracy drops dramatically for sequences exhibiting poorly-textured regions and rapidly changing viewpoints (i.e. trees that rapidly move closer) such as 'ocean' and 'seasonsforest'. Conversely, both our variants perform stably across various environments with considerable improvement from our full pipeline, which has the best overall pose accuracy scores.

Monocular Results. For monocular input, the challenging dataset caused our baselines to frequently lose tracking, obfuscating a fair comparison over translation error, which requires scale correction due to scale ambiguity. That is, whenever a system loses tracking, a new map with arbitrary scale is generated. If we scale each sub-sequence respectively, the system losing tracking more often will benefit from getting more accurate scale correction. Thus, we replace translation error with the completeness metric, similar to the success rate proposed in [27]. The results in Table.I (right) show VOLDOR$^+$SLAM can robustly handle different environments with high completeness compared to our baselines, while achieving best overall rotation accuracy.

Depth Evaluation. Fig.4 conveys our depth map quality results. As baselines, we use GA-Net [28] and MaskFlowNet [20] (the stereo input of our stereo-based VO). The accuracy

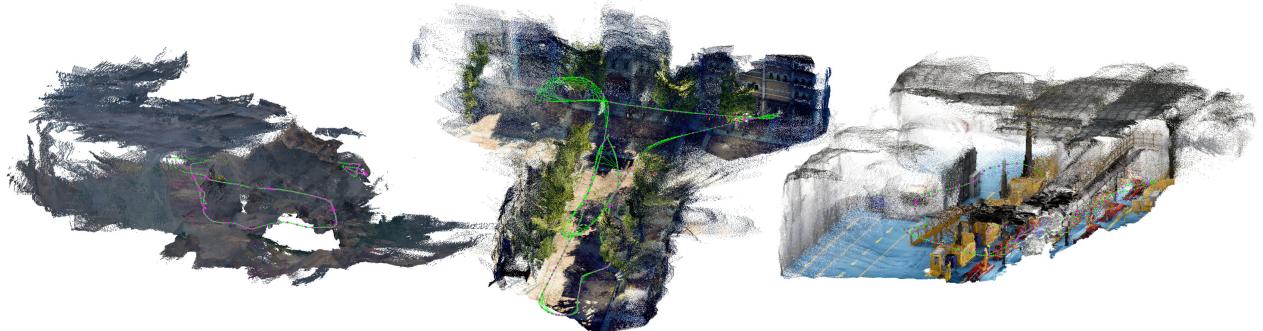| Sequence (Hard Only)	Stereo						Monocular					
	ORB-SLAM3		VOLDOR$^+$		VOLDOR$^+$SLAM		ORB-SLAM3		DSO		VOLDOR$^+$SLAM	
	Trans. (%)	Rot. (deg/m)	Trans. (%)	Rot. (deg/m)	Trans. (%)	Rot. (deg/m)	Comp. (%)	Rot. (deg/m)	Comp. (%)	Rot. (deg/m)	Comp. (%)	Rot. (deg/m)
abandonedfactory	0.0404	**0.0914**	0.0345	0.1095	**0.0269**	0.0924	0.969	**0.2204**	0.970	0.2624	**0.983**	0.3210
abandonedfactory_night	0.1633	0.0765	0.0227	0.0800	**0.0196**	**0.0702**	0.932	0.4907	0.916	1.4233	**0.964**	**0.2229**
amusement	0.0129	0.0654	0.0125	0.0315	**0.0118**	**0.0265**	0.640	-	0.505	-	**1.000**	**0.1188**
carwelding	**0.0060**	**0.0269**	0.0124	0.0518	0.0131	0.0535	0.995	**0.0850**	0.435	-	**0.997**	0.3203
ocean	0.0664	0.4379	0.0376	0.1340	**0.0335**	**0.1311**	0.908	0.7564	0.898	1.7069	**0.958**	0.4775
office	**0.0035**	**0.0323**	0.0088	0.0740	0.0071	0.0624	0.907	0.2872	**0.950**	4.4242	0.853	0.1715
japanesealley	0.0193	0.0773	0.0105	**0.0369**	**0.0102**	0.0413	0.964	0.1819	0.959	0.1291	**1.000**	**0.1150**
seasonsforest	0.1160	0.3102	0.0293	0.1094	**0.0196**	**0.0733**	0.330	-	0.534	-	**1.000**	**0.1525**
westerndesert	0.0906	0.3311	0.0177	0.0898	**0.0149**	**0.0786**	0.918	0.3933	0.855	1.4918	**0.957**	**0.2145**
Avg.	0.0576	0.1610	0.0207	0.0797	**0.0174**	**0.0699**	0.840	0.3449	0.780	1.5729	**0.968**	**0.2348**

TABLE I: Camera pose accuracy evaluated on TartanAir dataset. Translation and rotation error metrics are based on normalized relative pose error as in KITTI [26], but averaged over all possible sub-sequences of length $(5, 10, 15, \dots, 40)$ meters. The completeness metric measures the number of successfully registered frames, which reveals a method's robustness under each environment. If the completeness of a certain sequence falls bellow 0.8, we do not estimate rotation error.

metrics used are inlier-rate and EPE. A depth value is considered inlier when its corresponding disparity EPE is less than 3 pixels or 5% ground-truth disparity. Results show our subset of pixels with high confidence greatly outperforms existing baselines, where around 50% pixels have a confidence over 0.99. Our framework provides only slightly more accurate depth maps using stereo instead of monocular input. Since VOLDOR$^+$ disadvantages the estimation of observed dynamic content, we deem reported accuracy values an under-estimate for static environments.
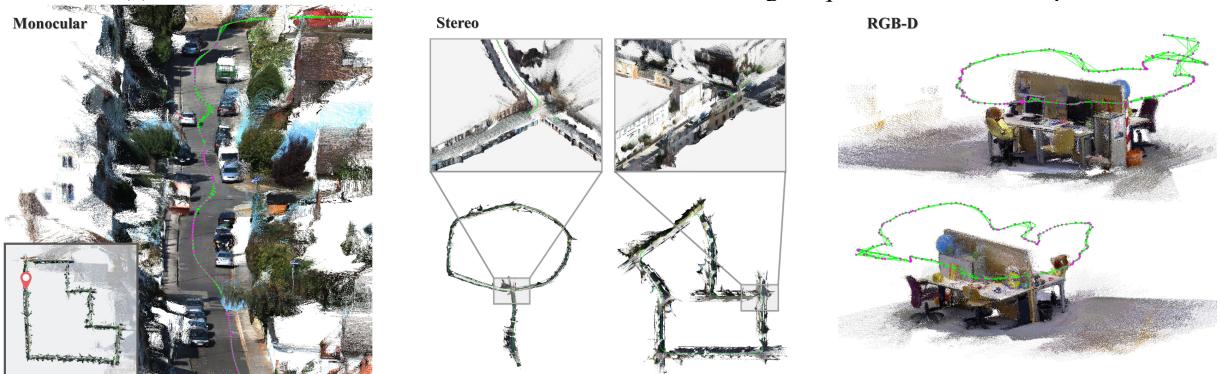
Qualitative Results. Representative reconstructions from TartanAir, KITTI and TUM datasets can be found in Fig.5.

V. CONCLUSIONS

VOLDOR$^+$SLAM demonstrates the potential of dense-indirect (DI) estimation frameworks for the geometric analysis of large-scale and unstructured environments. The modular nature of the DI approach allows for the exploration of novel formulations and ancillary tools. In particular, recently proposed learning approaches jointly estimating depth and poses [29], [30], [31], [32], and multi-way 3D registrations [33], [34], [35], [36] are highly related to our work and offer promising research paths toward tighter couplings between DI estimation and global map management modules.

(a) Results on TartanAir 'ocean', 'oldtown' and 'carwelding' sequence with stereo input.

(b) Results on KITTI and TUM RGB-D datasets with different types of input.

Fig. 5: Qualitative Results. Top: Results across diverse environments show VOLDOR$^+$SLAM's robustness to capture variability. Bottom: Zoomed-in regions and global views illustrate our 3D mapping density, consistency and level of detail.

REFERENCES

[1] Z. Min, Y. Yang, and E. Dunn, "Voldor: Visual odometry from log-logistic dense optical flow residuals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4898–4909.

[2] T. Pire, T. Fischer, J. Civera, P. De Cristóforis, and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1373–1378.

[3] 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*, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2320–2327.

[5] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

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

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

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

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam," *arXiv preprint arXiv:2007.11898*, 2020.

[11] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.

[12] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[13] 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)*, vol. 36, no. 4, p. 1, 2017.

[14] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 134–144.

[15] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert, "Dense versus sparse approaches for estimating the fundamental matrix," *International Journal of Computer Vision*, vol. 96, no. 2, pp. 212–234, 2012.

[16] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4058–4066.

[17] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[18] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[19] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[20] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu, *et al.*, "Maskflownet: Asymmetric feature matching with learnable occlusion mask," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6278–6287.

[21] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[22] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," *Robotics: Science and Systems VI*, vol. 2, no. 3, p. 7, 2010.

[23] T. Ke and S. I. Roumeliotis, "An efficient algebraic solution to the perspective-three-point problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7225–7233.

[24] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm, "Patchmatch based joint view selection and depthmap estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1510–1517.

[25] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 143–152.

[26] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *2010 ieee intelligent vehicles symposium*. IEEE, 2010, pp. 486–492.

[27] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," *arXiv preprint arXiv:2003.14338*, 2020.

[28] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[30] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[31] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5555–5564.

[32] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1281–1292.

[33] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger, "Taking a deeper look at the inverse compositional algorithm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4581–4590.

[34] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[35] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 822–838.

[36] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *arXiv preprint arXiv:1806.04807*, 2018.