2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Monocular Visual-Inertial Depth Estimation

Diana Wofk[1], René Ranftl[1], Matthias Müller[1], and Vladlen Koltun[1,2]

***Abstract*—We present a visual-inertial depth estimation pipeline that integrates monocular depth estimation and visual-inertial odometry to produce dense depth estimates with metric scale. Our approach performs global scale and shift alignment against sparse metric depth, followed by learning-based dense alignment. We evaluate on the TartanAir and VOID datasets, observing up to 30% reduction in inverse RMSE with dense scale alignment relative to performing just global alignment alone. Our approach is especially competitive at low density; with just 150 sparse metric depth points, our dense-to-dense depth alignment method achieves over 50% lower iRMSE over sparse-to-dense depth completion by KBNet, currently the state of the art on VOID. We demonstrate successful zero-shot transfer from synthetic TartanAir to real-world VOID data and perform generalization tests on NYUv2 and VCU-RVI. Our approach is modular and is compatible with a variety of monocular depth estimation models.**

I. Introduction

Depth perception is fundamental to visual navigation, where correctly estimating distances can help plan motion and avoid obstacles. Accurate depth estimation can also aid scene reconstruction, mapping, and object manipulation. Some applications of estimated depth benefit when it is *metrically accurate*—when every depth value is provided in absolute metric units and represents physical distance.

Algorithms for dense depth estimation can be broadly grouped into several categories. Stereo-based approaches rely on two or more cameras that capture different views. Structure-from-motion (SfM) tries to estimate scene geometry from a sequence of images taken by a moving camera, but it is difficult to recover depth with absolute scale since the relative pose of the camera across images is not known. Monocular approaches require just one camera and try to estimate depth from a single image. Such approaches are appealing since simple RGB cameras are compact and ubiquitous. However, monocular approaches that rely solely on visual data still exhibit scale ambiguity.

Incorporating inertial data can help resolve scale ambiguity, and most mobile devices already contain inertial measurement units (IMUs). Simultaneous localization and mapping (SLAM) systems [1]–[3] use visual or visual-inertial data to track scene landmarks under camera motion, compute the camera trajectory, and map the traversed environment. However, SLAM systems typically only track on the order of hundreds to thousands of sparse feature points, resulting in metric depth measurements that are only semi-dense at best. Our work explores how to use inertial data in conjunction with monocular visual data to produce fully-dense metrically accurate depth predictions as in Figure 1.

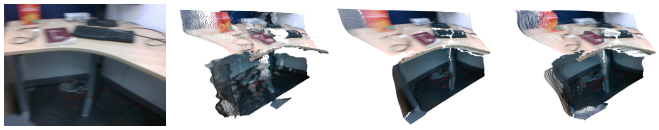

Fig. 1. We integrate visual-inertial odometry and monocular depth estimation to produce dense depth with metric scale. Global alignment (GA) determines appropriate global scale, while dense scale alignment (SML) operates locally and pushes or pulls regions towards correct metric depth. Here, with GA+SML, objects are aligned more accurately, the center desk leg is straightened, and the top of the desk is pulled forward.

Recent advances in supervised learning-based monocular depth estimation [4], [5] provide high generality but do not resolve absolute metric scale and only predict relative depth. Works that use inertial data to inform metric scale typically perform depth completion given a set of known sparse metric depth points and tend to be self-supervised in nature due to a lack of visual-inertial datasets [6], [7]. We seek to bridge these approaches by leveraging monocular depth estimation models trained on diverse datasets and recovering metric scale for individual depth estimates.

Our approach performs least-squares fitting of monocular depth estimates against sparse metric depth, followed by learned local per-pixel adjustment. This combination of global and dense (local) depth alignment successfully rectifies metric scale, with dense alignment consistently outperforming a purely global alignment baseline. Alignment succeeds with just 150 metric depth anchors and is robust to zero-shot cross-dataset transfer. Our pipeline is modular and is agnostic to the monocular depth estimation model and VIO system being used; it should thus benefit from continual improvement in these modules.

II. Related Work

Monocular depth estimation is inherently an ill-posed problem facing challenges like scale ambiguity. A common approach to handling this in supervised training has been to limit training data to particular datasets with desired environments, e.g., indoor or outdoor scenes. This encourages the supervised network to memorize a metric scale that may be globally inconsistent, results in overfitting to specific depth ranges, and hurts generalizability across environments. Recent work on dataset mixing and training loss construction [4] has enabled robust affine-invariant monocular depth estimation across a variety of datasets. However, recovering absolute metric scale in these depth estimates remains a challenge.

[1]This work was done at Intel Labs.
[2]Currently affiliated with Apple.

Using inertial and pose information. Incorporating inertial data is being explored as a means of improving metric depth accuracy in self-supervised depth estimation approaches. Fei et al. [8] propose using global orientation from inertial measurements to regularize depth regression at training time, with an expanded loss function that penalizes planarity deviation based on gravity vectors estimated through VIO. SelfVIO [9] combines learning-based VIO and depth estimation to develop an adversarially trained architecture that jointly estimates egomotion and dense depth from input RGB and IMU readings. A number of additional works incorporate pose into supervised and unsupervised approaches [10]–[13], often as part of pose consistency and reprojection terms, or as a pose estimation task that is performed jointly with depth estimation. In the latter case, replacing pose networks with pose estimation from VIO/SLAM is known to improve performance [6], [8].

Depth completion from sparse depth. Sparse depth maps or sparse point clouds, e.g., obtained with LiDAR or through VIO tracking, commonly serve as input to metric depth completion. In VOICED [6], sparse depth from VIO is used as a depth scaffold that is refined to minimize photometric, pose, and depth consistency losses. KBNet [7] adds camera calibration and connects sparse depth and RGB encoders with backprojection layers. Other recent works also explore visual-inertial depth completion [14], [15], although they rely on depth completion networks that are trained primarily on indoor data, thus limiting generality.

Video depth estimation. In the absence of inertial data, given an ordered sequence of images, temporal correlation can be used to improve scale consistency of monocular depth estimates, though still without absolute scale. CVD [16] leverages SfM [17] to estimate camera parameters and define geometric constraints that help resolve global scale consistency across per-frame depth maps predicted from monocular video input. Since SfM may fail under challenging motion, Robust CVD [18] replaces it with pose estimation and optimization done jointly with depth scale realignment based on a bilinear spline. In both methods, absolute metric scale remains unknown.

Our work aims to resolve scale ambiguity by performing global and local depth alignment in absolute metric space, given an off-the-shelf monocular depth estimation model and VIO system. Instead of designing novel depth estimation architectures and training procedures, we build upon existing monocular depth models and realign their output depth estimates. We do not perform depth completion [6], [16], but rather align an already-dense depth map to absolute metric scale. This is a more versatile approach as it can incorporate arbitrary monocular depth estimation models.

III. Method

We develop a modular three-stage pipeline for visual-inertial depth estimation. Its structure is illustrated in Figure 2.

Monocular depth estimation. The *visual* branch of our pipeline predicts depth from a single image. This is done using a pretrained model that takes in a single RGB image and produces a dense depth map up to some scale. Monocular processing is appealing as it allows for low-complexity architectures that do not carry large computational costs.

Our approach is compatible with traditional convolutional models as well as newer architectures. We select DPT-Hybrid [5] as our depth estimator; this is a transformer-based model trained on a large meta-dataset using scale- and shift-invariant losses. While it achieves high generalizability, its output measures depth relations between pixels, and depth values do not carry metric meaning. Our alignment pipeline aims to recover metric scale for every pixel in this output depth map.

Visual-inertial odometry. The *inertial* branch of our pipeline uses IMU data together with visual data to determine metric scale. Given a sequence of RGB images with synchronized IMU data, we run VINS-Mono [19] to compute the camera trajectory and yield a set of 3D world coordinates of features tracked throughout the sequence. In a reasonably textured environment, we can expect tens of tracked features per frame. By projecting feature coordinates to image space, we obtain a sequence of sparse maps containing metric depth values. These sparse depth maps serve as inputs to later alignment tasks, thereby propagating metric scale through our pipeline.

Global scale and shift alignment (GA). Let $\mathbf{z}$ refer to unitless affine-invariant inverse depth that is output by a monocular depth estimation model such as DPT-Hybrid. To reintroduce metric scale into depth, we align monocular depth estimates to sparse metric depth obtained through VIO. This global alignment is performed in inverse depth space based on a least-squares criterion [4]. The result is a per-frame global scale s_g and global shift t_g that are applied to $\mathbf{z}$ as a linear transformation. Applying global scale can be interpreted as bringing depth values to a correct order of magnitude, while applying global shift can help undo potential bias or offset in the original prediction. The resulting globally-aligned depth estimates are $\tilde{\mathbf{z}} = s_g\mathbf{z} + t_g$.

Dense (local) scale alignment. Due to its coarse nature, global alignment will not adequately resolve metric scale in all regions of a depth map. To address this, we propose a learning-based approach for determining dense (per-pixel) scale factors that are applied to globally-aligned depth estimates. Using MiDaS-small [4], we construct a network that is trained to realign individual pixels in a depth map to improve their metric accuracy. We call this network the ScaleMapLearner (SML) and feed it an input of two concatenated data channels: the *globally-aligned depth* $\tilde{\mathbf{z}}$, and a *scaffolding for a dense scale map*, where n locations of known sparse depth values $\mathbf{v}$ from VIO define n scale anchor points $\mathbf{v}_i/\tilde{\mathbf{z}}_i$, $i \in \{1...n\}$. The region within the convex hull defined by the anchors is filled via linear interpolation of anchor values. The region outside the convex hull is filled with an identity scale value of 1.

SML regresses a dense scale residual map $\mathbf{r}$ where values are allowed to be negative. We compute the resulting scale map as $\text{ReLU}(1 + \mathbf{r})$ and apply it to the input depth $\tilde{\mathbf{z}}$ to produce the output depth $\hat{\mathbf{z}} = \text{ReLU}(1 + \mathbf{r})\tilde{\mathbf{z}}$.

Loss function. During training, the SML network is supervised on metric ground truth $\mathbf{z}^*$ in inverse depth space. Let

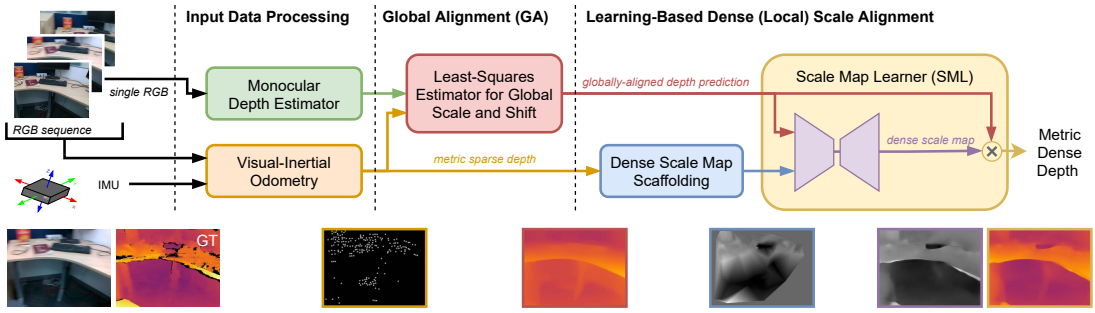

Fig. 2. Our visual-inertial depth estimation pipeline. There are three stages: (1) input processing, where RGB and IMU data feed into monocular depth estimation alongside visual-inertial odometry, (2) global scale and shift alignment, where monocular depth estimates are fitted to sparse depth from VIO in a least-squares manner, and (3) learning-based dense scale alignment, where globally-aligned depth is locally realigned using a dense scale map regressed by the ScaleMapLearner (SML). The row of images at the bottom illustrate a VOID [6] sample being processed through the pipeline; from left to right: the input RGB, ground truth depth, sparse depth from VIO, globally-aligned depth, scale map scaffolding, dense scale map regressed by SML, final depth output.

M define the number of pixels with valid ground truth. Our loss function comprises two terms: an L1 loss on depth,

$$\mathcal{L}_{depth}(\hat{\mathbf{z}}, \mathbf{z}^*) = \frac{1}{M}\sum_{i=1}^{M} |\mathbf{z}_i^* - \hat{\mathbf{z}}_i| \tag{1}$$

and a multiscale gradient matching term [20] that biases discontinuities to coincide with discontinuities in ground truth,

$$\mathcal{L}_{grad}(\hat{\mathbf{z}}, \mathbf{z}^*) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\sum_{i=1}^{M}(|\nabla_x R_i^k| + |\nabla_y R_i^k|) \tag{2}$$

where $R_i = \mathbf{z}_i^* - \hat{\mathbf{z}}_i$ and R^k denotes error at different resolutions. We use $K = 3$ levels, halving the spatial resolution at each level. Our final loss is $\mathcal{L} = \mathcal{L}_{depth} + 0.5\mathcal{L}_{grad}$.

Decoupling visual and inertial data. Our pipeline runs monocular depth estimation and VIO in parallel and independently of each other. The intermediate outputs from these steps are then fused together to generate inputs to SML. This design choice is made to better leverage ongoing advances in monocular depth and VIO systems; newly developed modules can be easily integrated within our pipeline, and SML can be quickly retrained to benefit from the improved performance of those modules. We contrast this with designing a single unified network that learns metric depth directly from a joint RGB-IMU input. A sufficiently large corpus of RGB-D datasets containing IMU data to train such a network and have it generalize well does not exist. We still face a data challenge when training SML; however, by decoupling RGB-to-depth and VIO at the input, we provide SML with an intermediate data representation that simplifies what it needs to learn to perform metric depth alignment. In this setting, a smaller amount of training data is sufficient.

IV. Datasets and Experiments

A key challenge in acquiring training data for the SML network is the lack of RGB-D+IMU datasets. In our pipeline, IMU data is needed to run VIO to generate sparse metric depth. While simulators allow recording synchronized RGB-D and IMU data [21], manually gathering sufficient training data is difficult. We select TartanAir [22] for its large size and variety of outdoor and indoor sequences. IMU data is not provided in this dataset. To proxy sparse depth map generation, we run the VINS-Mono feature tracker front-end [23] to obtain sparse feature locations and then sample ground truth depth at those locations. We use a 70%-30% train-test split for TartanAir, with 172K training and 73K test samples taken from both easy and hard sequences.

In addition to the synthetic TartanAir dataset, we benchmark on VOID [6], which offers real-world data collected using an Intel RealSense D435i camera and the VIO system XIVO [24]. This dataset is smaller than TartanAir, with only 47K training and 800 test samples. We use the published train-test split.

Setup. We use MiDaS-small [4] to construct our SML network. The encoder backbone is initialized with pretrained ImageNet [25] weights, while other layers are initialized randomly. We use AdamW [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\lambda = 0.001$. We set an initial learning rate of 5×10^{-4} when training on TartanAir and 3×10^{-4} on VOID. We additionally use a step-based scheduler that halves the learning rate after 5 epochs on TartanAir and after 8 epochs on VOID. We train for 20 epochs on a node with 8 GeForce RTX 2080 Ti GPUs, with a batch size of 256, and with mixed-precision training enabled. Input data is resized and cropped to a training resolution of 384×384. Training takes up to 4 hours.

Metrics. We mainly evaluate in inverse depth space $\mathbf{z} = 1/\mathbf{d}$ (in km^{-1}), as doing so penalizes error at closer depth ranges more significantly. We compute inverse mean absolute error $\text{iMAE} = \frac{1}{M}\sum_{i=1}^{M}|\mathbf{z}_i^* - \hat{\mathbf{z}}_i|$, inverse root mean squared error $\text{iRMSE} = [\frac{1}{M}\sum_{i=1}^{M}|\mathbf{z}_i^* - \hat{\mathbf{z}}_i|^2]^{\frac{1}{2}}$, and inverse absolute relative error $\text{iAbsRel} = \frac{1}{M}\sum_{i=1}^{M}|\mathbf{z}_i^* - \hat{\mathbf{z}}_i|/\mathbf{z}_i^*$. On VOID, we also compute MAE and RMSE in regular depth space $\mathbf{d}$ (in mm).

We follow the VOID evaluation protocol of Wong et al. [6], [7] and consider ground truth depth to be valid between 0.2 and 5.0 meters. The minimum and maximum depth prediction values in these works are set to 0.1 and 8.0 meters, respectively. We clamp depth predictions, both after global alignment and after applying regressed dense scale maps, to this range. In contrast to the mostly-indoor scenes in VOID, outdoor scenes in TartanAir exhibit larger depth ranges. For TartanAir, we define ground truth depth to be valid between 0.2 and 50 meters and clamp predictions between 0.1 and 80 meters.

TABLE I
EVALUATION ON TARTANAIR. LOWER IS BETTER FOR ALL METRICS.

Method	Depth Model	iMAE	iRMSE	iAbsRel
GA only	DPT-Hybrid	22.94	35.49	0.126
GA+SML		**16.11**	**29.48**	**0.093**
GA only	MiDaS v2.0	58.11	79.34	0.299
GA+SML		**28.79**	**46.67**	**0.156**

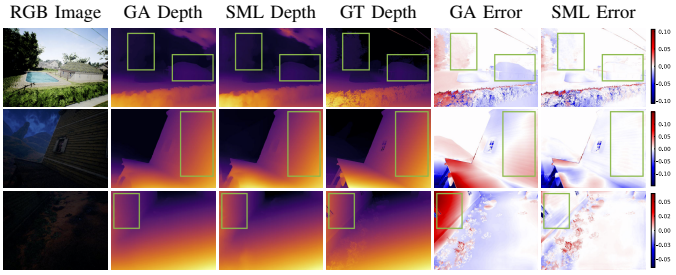

Fig. 3. Our method tested on TartanAir samples. In depth maps, brighter is closer and darker is farther. In error maps, red is positive inverse depth error (farther than ground truth GT) and blue is negative inverse depth error (closer than GT). Whiter regions in error indicate improved metric depth accuracy.

A. Evaluation on TartanAir

We first evaluate on *synthetic* samples from the TartanAir dataset, where inertial data is unknown. To proxy sparse depth generation from VIO, we preprocess TartanAir data with a sparsifier that samples depth from the ground truth at locations determined via the VINS-Mono-based feature tracker implemented in [23]. We run monocular depth estimation with DPT-Hybrid, perform global alignment against metric sparse depth, and generate a scale map scaffolding for every sample prior to SML training. We define our baseline as global alignment only and show that performing dense scale alignment with SML improves metric depth accuracy. Table I shows that SML achieves 30%, 17%, and 26% reduction in iMAE, iRMSE, and iAbsRel, respectively. Metrics are aggregated across a set of 690 samples taken from our TartanAir test split.

Figure 3 provides a visualization of our approach on several TartanAir samples. Performance is qualitatively evaluated by comparing metric depth error for globally-aligned depth (GA error) versus densely-scaled depth (SML error). A whiter region in the error map indicates that SML improved metric depth accuracy there. The first sample depicts a neighborhood scene where the building towards center-right is pushed further back under dense scale alignment; this is confirmed by a reduction in negative (blue) error in inverse depth. The tree behind the pool is brought closer, as shown by the reduction in positive (red) error. The latter two samples depict significantly more challenging scenes due to low light as well as proximity to walls and the ground. In both, the SML still aligns surfaces towards the correct metric depth.

We note that DPT-Hybrid was trained on a large mixed dataset containing TartanAir. To remove any potential bias this contributes to SML evaluation on TartanAir, we swap in MiDaS v2.0 [4] that has not seen any TartanAir data during training. Table I shows that MiDaS v2.0 still yields the same trends as DPT-Hybrid, with SML improving all metrics.

TABLE II
EVALUATION ON VOID. ALL METHODS USE DPT-H AS THE DEPTH MODEL AND 150 SPARSE DEPTH POINTS. LOWER IS BETTER.

Method	Training Set	MAE	RMSE	iMAE	iRMSE	iAbsRel
GA only		165.33	243.11	75.74	106.37	0.103
GA+SML	VOID	97.03	167.82	46.62	74.67	0.063
GA+SML	TA (zero-shot)	98.49	175.04	45.55	74.28	0.062
GA+SML	TA + VOID	**82.65**	**153.51**	**38.56**	**66.23**	**0.051**

B. Evaluation on VOID

We additionally evaluate on real-world data from the VOID dataset. We preprocess VOID data in the same manner as the TartanAir data, but using the sparse depth provided in the published dataset [6]. The first two rows of Table II summarize our results when training SML directly on VOID. SML again improves over global alignment, with a 38%, 30%, and 39% reduction in iMAE, iRMSE, and iAbsRel, respectively.

TartanAir-to-VOID transfer. We investigate the performance of SML when trained on TartanAir and evaluated on VOID without any finetuning (i.e., zero-shot cross-dataset transfer). This can be interpreted as a sim-to-real transfer experiment, since TartanAir consists solely of synthetic data and VOID contains real-world data samples. We observe that zero-shot testing on VOID achieves very similar error as when training directly on VOID. If evaluating in inverse depth space, zero-shot transfer even slightly outperforms direct training on VOID. This is particularly notable since it demonstrates that training on a large quantity of diverse synthetic data can indeed translate to improved real-world performance. It also shows the generalizability of our pipeline. DPT-Hybrid is already known to generalize well after having been trained on a massive mixed dataset with scale- and shift-invariant loss functions. The SML network is trained using metric loss terms; however, some metric information is provided to SML via the globally-aligned depth and scale map scaffolding inputs. Since SML only needs to learn to refine this scaffolding, it is less likely to memorize or overfit to a specific metric scale.

Pretraining. Pretraining on TartanAir and fine-tuning on VOID yields the lowest error across all metrics. We use this combination to produce the results visualized for samples in Figure 4. The first sample suffers from blurriness in the RGB input and depicts a cluttered scene. With global alignment only, depth predictions appear flattened: the table is aligned to be farther than ground truth (red error), while background surfaces such as walls and the floor are aligned to be closer than ground truth (blue error). Dense scale alignment with SML helps to rectify this, with noticeable reduction (whiter regions) throughout the error map. The second sample shows a staircase; in addition to reducing depth error on the steps, SML is able to correctly realign the handrail on the left. This is impressive as pixels near the image boundary fall outside the convex hull of known sparse depth points, and the scale map scaffolding that we input to SML signals no information at pixels outside the convex hull. The last sample shows a challenging viewpoint of the floor leading to a staircase in the top right corner. Global alignment alone misjudges the

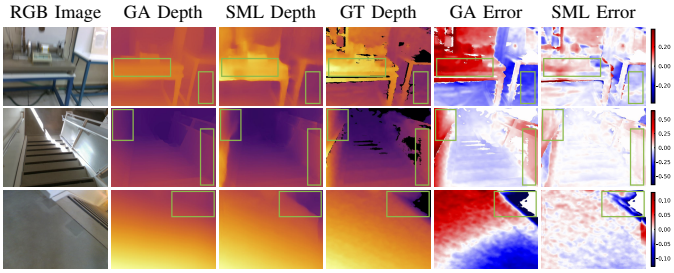

Fig. 4. Our method tested on VOID samples. SML is pretrained on TartanAir and fine-tuned on VOID. Color coding is the same as in Figure 3.

TABLE III
COMPARISON ON VOID. LOWER IS BETTER FOR ALL METRICS.

	Method	MAE	RMSE	iMAE	iRMSE
150 points	VOICED-S [6]	174.04	253.14	87.39	126.30
	KBNet [7]	131.54	263.54	66.84	128.29
	GA+SML (DPT-BEiT-Large)	**76.95**	**142.85**	**34.25**	**57.13**
	GA+SML (DPT-Hybrid)	97.03	167.82	46.62	74.67
	GA+SML (MiDaS-small)	113.27	193.38	53.86	84.82
500 points	VOICED-S [6]	118.01	195.32	59.29	101.72
	KBNet [7]	77.70	172.49	38.87	85.59
	GA+SML (DPT-BEiT-Large)	**66.14**	**126.44**	**28.92**	**49.85**
	GA+SML (DPT-Hybrid)	81.30	146.16	37.35	60.92
	GA+SML (MiDaS-small)	94.81	164.36	43.19	69.25

depth gradient at the staircase edge. SML corrects this and also reduces depth error elsewhere on the floor surface.

Comparison to related work. Our evaluation thus far has compared the impact of SML relative to global alignment only. We now compare to related work on VOID. Table III lists VOICED [6] and state-of-the-art KBNet [7] alongside our approach (GA+SML). Figure 5 shows a qualitative comparison.

In addition to using DPT-Hybrid as the depth model in our pipeline, we try DPT-BEiT-Large for its higher accuracy and MiDaS-small for its computational efficiency. With just 150 sparse depth points, our approach GA+SML outperforms KBNet across all metrics, regardless of what depth estimator we use; improvement in iRMSE ranges from 34% to 55%. From Table II, we see that even with zero-shot transfer, our method outperforms KBNet by 42% in iRMSE. At a higher density of 500 points, our pipeline with DPT-BEiT-Large continues to outperform KBNet across all metrics.

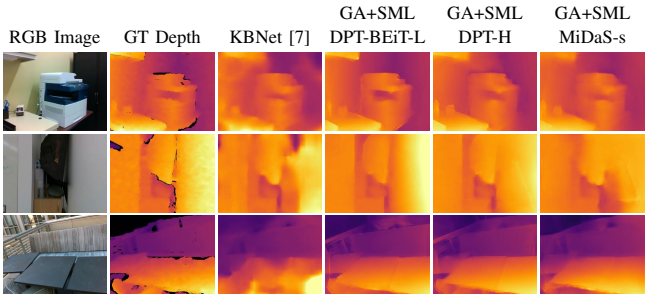

Fig. 5. Qualitative comparison of our approach against state-of-the-art KBNet on the VOID 150 dataset. SML is trained only on VOID.

C. Generalizability and Deployability

We test zero-shot generalization on NYU Depth v2 [27] and VOID, comparing against NLSPN [28] (state of the art on NYUv2) and KBNet (state of the art on VOID). These models, having been trained on a single dataset as is commonplace with depth completion tasks, underperform when run on a different dataset. Table IV shows that our approach consistently achieves better generalization performance.

TABLE IV
TESTING ZERO-SHOT GENERALIZABILITY ON NYUV2 AND VOID. DPT-HYBRID IS USED AS THE DEPTH PREDICTOR FOR GA+SML.

	NYUv2 (train) → VOID (test) Method	iMAE	iRMSE	*VOID (train) → NYUv2 (test)* Method	iMAE	iRMSE
150 pts	NLSPN [28]	143.0	238.1	KBNet [7]	35.2	67.8
	GA+SML	**55.9**	**85.2**	GA+SML	**30.2**	**48.9**
500 pts	NLSPN [28]	87.9	174.7	KBNet [7]	28.0	57.2
	GA+SML	**43.9**	**69.5**	GA+SML	**26.8**	**44.5**

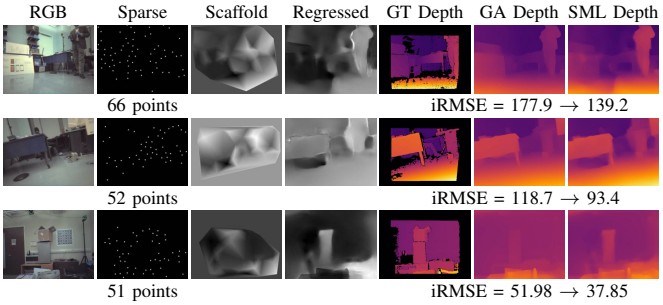

Fig. 6. Our method tested on lab and corridor samples from the VCU-RVI dataset. RGB and sparse metric depth come from published rosbag data.

We also test on rosbags from an entirely new dataset, VCU-RVI [29]. Figure 6 shows samples where available sparse metric depth is much lower in quantity than the 150+ points we have so far trained and tested with. Our pipeline still succeeds in resolving metric scale, with SML reducing depth error.

To demonstrate deployability, we benchmark performance on the NVIDIA Jetson AGX Orin platform and show a breakdown of component runtime in Table V. Measurements are averaged over 100 runs after 20 warmup runs. With acceleration via TensorRT, our depth alignment pipeline, in conjunction with a lightweight depth predictor like MiDaS-small, is viable for on-device metric depth estimation. Scale map scaffolding is one bottleneck as interpolation within the convex hull presently runs on the CPU. Data movement between the GPU and host CPU is another bottleneck that we expect can be reduced with additional engineering effort.

TABLE V
RUNTIME [MS] ON JETSON AGX ORIN IN MAX-N MODE. ALL PIPELINE VARIANTS ARE TESTED WITH 150 SPARSE METRIC DEPTH POINTS.

Depth predictor Inference resolution	DPT-BEiT-L 384×384	DPT-H 384×384	MiDaS-s 256×256	MiDaS-s-TRT 256×256
Depth inference	144.8	53.9	29.2	1.5
D2H copy depth map	12.8	18.4	0.6	5.2
Global alignment	2.6	2.5	1.3	1.3
Scale map scaffolding	12.2	12.1	6.7	6.6
H2D copy SML inputs	3.3	3.3	2.4	2.2
SML-TRT inference	2.2	2.2	1.7	1.7
Total	177.9	92.5	41.9	18.5

TABLE VI
EXPERIMENTS WITH INPUT AND REGRESSED MODALITIES IN SML. LOWER IS BETTER FOR ALL METRICS.

Input Modality Combinations							Regressing		On TartanAir			On VOID (zero-shot)		
GA Depth	Scale Scaff.	Confidence	Gradients	Grayscale	RGB		Scale	Shift	iMAE	iRMSE	iAbsRel	iMAE	iRMSE	iAbsRel
baseline (global alignment without SML)									22.94	35.49	0.126	75.74	106.37	0.103
✓							✓		22.63	35.30	0.125	111.55	159.83	0.212
✓		✓					✓		22.51	35.09	0.124	122.77	179.09	0.238
✓	✓						✓		**16.11**	**29.48**	0.093	**45.55**	**74.28**	**0.062**
✓	✓	✓					✓		16.90	30.63	0.094	63.55	94.78	0.092
✓	✓	✓	✓				✓		17.07	30.14	0.098	50.19	79.88	0.069
✓	✓	✓		✓			✓		16.87	30.15	0.096	57.25	87.83	0.083
✓	✓	✓			✓		✓		18.03	31.64	0.102	59.08	91.00	0.080
✓	✓							✓	17.03	30.12	0.096	62.34	90.91	0.086
✓	✓						✓	✓	16.28	29.53	**0.092**	50.95	80.96	0.071

D. Ablations and Analysis

We experiment with a number of input and regressed data modalities when designing the SML network.

Input data modalities. SML takes in two inputs: *globally-aligned inverse depth* $\tilde{\mathbf{z}}$ and a *scale map scaffolding*. We experiment with four additional inputs: (1) a *confidence map* derived from a binary map pinpointing known sparse depth locations, first dilated with a 7×7 circular kernel and then blurred with a $5{\times}5$ Gaussian kernel to mimic confidence spread around a fixed known point; (2) a *gradient map* computed using the Scharr operator; (3) a *grayscale* conversion of the original RGB image; (4) and the RGB image itself. Inputs are concatenated in the channel dimension and fed into SML as a single tensor. Table VI reports the impact of different input combinations on the metric accuracy of SML depth.

Globally-aligned depth alone is not sufficient for the network to learn dense scale regression well. An input scale map scaffolding is necessary. Conceptually, this acts as an initial guess at the dense scale map that the network is learning to regress. Without an accompanying scale map input, the confidence map negligibly improves SML learning; however, using both slightly underperforms compared to using only scale scaffolding. This is surprising, as the confidence map is meant to signal which regions in the input depth and scale scaffolding are more trustworthy. It may be that our representation of confidence is not being parsed well by SML, or that the scale map scaffolding encodes similar information, e.g., boundaries of the convex hull and approximate positions of interpolation anchors corresponding to known sparse metric depth. Incorporating edge representations in the form of gradient maps, grayscale, or RGB images, does not appear to be beneficial. This can be partly attributed to the high quality of depth predictions output by DPT, as those depth maps already exhibit clear edges and surfaces. RGB input actually worsens performance, implying that color cues are not particularly useful in dense metric scale regression.

Since we are also interested in cross-dataset transfer, we evaluate zero-shot performance of every input combination on VOID and report the results in Table VI. Combined depth and scale scaffolding result in noticeably lower error; we therefore select this input combination for SML.

Regressing scale and shift. SML learns dense (per-pixel) scale factors by which to multiply input depth estimates $\tilde{\mathbf{z}}$, such that the output depth $\hat{\mathbf{z}}$ achieves higher metric accuracy. The network is allowed to regress negative values as scale residuals $\mathbf{r}$, such that the output depth is $\hat{\mathbf{z}} = \text{ReLU}(1 + \mathbf{r})\tilde{\mathbf{z}}$. Our design choice to regress scale is motivated by scale factors having a more intuitive interpretation in projective geometry. Scaling a depth value at a pixel location can be interpreted as zooming in (pulling closer) or zooming out (pushing further) the object at that location in 3D space. It is more difficult to intuit the impact of shifting depth at individual pixels. We conduct two experiments that involve shift, listed in the bottom two rows of Table VI. We regress only dense shift $\mathbf{t}$, such that the output prediction $\hat{\mathbf{z}} = \tilde{\mathbf{z}} + \mathbf{t}$. We also regress shift $\mathbf{t}$ alongside scale residuals $\mathbf{r}$, where $\hat{\mathbf{z}} = \text{ReLU}(1 + \mathbf{r})\tilde{\mathbf{z}} + \mathbf{t}$. For the latter, we add a second output head to the SML network, while the encoder and decoder layers remain common to both regression tasks. When training with shift regression, our default learning rate of $5{\times}10^{-4}$ prohibits loss convergence and necessitates a slightly lower one of $4{\times}10^{-4}$. Overall, regressing shift does not significantly impact performance on TartanAir, and zero-shot testing on VOID indicates that regressing scale only is the most robust choice for cross-dataset transfer.

V. CONCLUSION

Combining metric accuracy and high generalizability is a key challenge in learning-based depth estimation. We propose incorporating inertial data into the visual depth estimation pipeline—not through sparse-to-dense depth completion, but rather through dense-to-dense depth alignment using estimated and learned scale factors. Inertial measurements inform and propagate metric scale through global and local depth alignment. We show improved error reduction with learning-based local alignment over least-squares global alignment alone, and demonstrate successful zero-shot cross-dataset transfer from synthetic training data to real-world test data. Our modular approach supports direct integration of existing and future monocular depth estimation and visual-inertial odometry systems. It succeeds in resolving metric scale for metrically-ambiguous monocular depth estimates, and we hope that it will assist in the deployment of robust and general monocular depth estimation models.

REFERENCES

[1] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *ECCV*, 2014.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, 2015.

[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*, 2017.

[4] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, 2022.

[5] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *ICCV*, 2021.

[6] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *IEEE Robotics and Automation Letters*, 2020.

[7] A. Wong and S. Soatto, "Unsupervised depth completion with calibrated backprojection layers," in *ICCV*, 2021.

[8] X. Fei, A. Wong, and S. Soatto, "Geo-supervised visual depth prediction," *IEEE Robotics and Automation Letters*, 2019.

[9] Y. Almalioglu, M. Turan, A. E. Sari, M. R. U. Saputra, P. P. B. de Gusmão, A. Markham, and A. Trigoni, "Selfvio: Self-supervised deep monocular visual-inertial odometry and depth estimation," *ArXiv*, 2019.

[10] V. Patil, W. V. Gansbeke, D. Dai, and L. V. Gool, "Don't forget the past: Recurrent depth estimation from monocular video," *IEEE Robotics and Automation Letters*, 2020.

[11] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," in *ICLR*, 2020.

[12] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural rgb-d sensing: Depth and uncertainty from a video camera," *CVPR*, 2019.

[13] J. Xie, C. Lei, Z. Li, L. E. Li, and Q. Chen, "Video depth estimation by fusing flow-to-depth proposals," *IROS*, 2020.

[14] K. Sartipi, T. Do, T. Ke, K. Vuong, and S. I. Roumeliotis, "Deep depth estimation from visual-inertial slam," *IROS*, 2020.

[15] N. Merrill, P. Geneva, and G. Huang, "Robust monocular visual-inertial depth completion for embedded systems," *ICRA*, 2021.

[16] X. Luo, J. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," *ACM Transactions on Graphics*, 2020.

[17] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016.

[18] J. Kopf, X. Rong, and J.-B. Huang, "Robust consistent video depth estimation," in *CVPR*, 2021.

[19] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, 2018.

[20] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *CVPR*, 2018.

[21] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, 2017.

[22] 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," *IROS*, 2020.

[23] P. Lusk and S. Sudhakar, "Anticipated vins-mono," https://github.com/plusk01/Anticipated-VINS-Mono, 2018.

[24] X. Fei and S. Soatto, "Xivo: An open-source software for visual-inertial odometry," https://github.com/ucla-vision/xivo, 2019.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[27] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[28] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-Local Spatial Propagation Network for Depth Completion," in *Proc. of European Conference on Computer Vision (ECCV)*, 2020.

[29] H. Zhang, L. Jin, and C. Ye, "The VCU-RVI Benchmark: Evaluating Visual Inertial Odometry for Indoor Navigation Applications with an RGB-D Camera," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.