Real-time, Dense UAV Mapping by Leveraging Monocular Depth Prediction with Monocular-Inertial SLAM

Yassine Habib, Panagiotis Papadakis, Cédric Le Barz, Antoine Fagette, Tiago Gonçalves, Cédric Buche

Real-time, Dense UAV Mapping by Leveraging Monocular Depth Prediction with Monocular-Inertial SLAM

Yassine Habib[a], Panagiotis Papadakis[a], Cédric Le Barz[c], Antoine Fagette[b], Tiago Gonçalves[c], Cédric Buche[d]

[a]IMT Atlantique Lab-STICC, UMR 6285, F-29238, team RAMBO, Brest France
[b]Thales Digital Solutions, Montreal, Canada
[c]Thales, Palaiseau, France
[d]ENIB, Brest, France

ABSTRACT
We present a dense and metric 3D mapping pipeline designed for embedded operation on-board UAVs, by loosely coupling deep neural networks trained to infer dense depth single images with a SLAM system that restores metric scale from sparse depth. In contrast to computationally restrictive approaches that leverage multiple views, we propose a highly efficient, single-view approach without sacrificing 3D mapping performance. This enables real-time construction of a global 3D voxel map by iterative fusion of the rescaled dense depth maps obtained via ray-casting from the estimated camera poses. Quantitative and qualitative experimentations of our framework in challenging environmental conditions show comparable or superior performance with respect to state-of-the-art approaches via a better effectiveness-efficiency trade-off.

KEYWORDS
3D metric mapping, depth prediction, deep learning, embedded systems

1. Introduction

The use of robots in recent years is becoming of strategic importance in safety, security, and rescue operations allowing first responders to more easily reach inaccessible areas, handle objects with precision, and collect information [25]. In the particular case of Urban Search And Rescue (USAR), autonomous navigation and 3D mapping help to reduce the cognitive load of pilots, allowing them to focus on analyzing images, and facilitating victim localization, rather than avoiding obstacles and planning optimal routes.Towards this goal, localization and environment mapping are critical functionalities that allow obstacle avoidance on the basis of a dense and metric 3D map that should be constructed in real-time. A monocular camera coupled with an Inertial Measurement Unit (IMU) is an ideal sensor set for embedding such tasks onto UAVs thanks to their reduced Size, Weight, Power, and Cost (SWaP-C) properties [2]. However, the lack of direct depth sensing brings about the additional problem of estimating depth.

In this context, while Simultaneous Localization And Mapping (SLAM) may provide accurate maps in real-time, most state-of-the-art methods build and maintain a sparse map to reduce computational complexity [7]. While their robustness has been demonstrated on different benchmarks [6,36], dealing with challenging drone flying conditions (aggressive motion, high illumination changes) is relatively understudied. In recent years, certain works have addressed monocular SLAM densification by leveraging Deep Neural Network (DNN) capabilities to directly infer a dense depth map from a monocular image. These works achieved promising results in obtaining precise dense 3D reconstruction. Yet, they typically do not address indoor navigation tasks that require the map to be metric and to be constructed in real-time on an embedded system.

Email: panagiotis.papadakis@imt-atlantique.fr, ORCID: https://orcid.org/0000-0002-2193-8087

This work builds upon and extends our previous developments [21],[22] where we presented the first results of a loosely-coupled framework for monocular SLAM densification tailored for use on embedded systems, as well as a lightweight scale recovery leveraging Monocular Depth Estimation (MDE). In particular, we go beyond our previous work by employing a significantly more accurate monocular depth prediction method, namely, ZeroDepth [19]. This leads in recovering both the metric scale as well as more precise depth/3D estimates, hence, allowing to generate a 3D voxel map that is exploitable for drone navigation. Furthermore, comparing the results of our entire pipeline using two alternative monocular depth estimation methods (PackNet-SfM [18] and ZeroDepth) highlights that the overall approach can be customized and that the scale recovery is general purpose. In summary, the contributions of this paper are as follows :

(i) A comprehensive description of the entire pipeline which includes the chosen multi-view volumetric mapping and fusion approach.
(ii) A broader quantitative as well as qualitative evaluation of the complete framework.

The remainder of this paper begins by reviewing related works in monocular SLAM densification and MDE in Section 2, highlighting their limitations for the scenario of indoor drone mapping. Subsequently, in Section 3, we present our proposed framework for densifying monocular SLAM, focusing on the metric scale recovery procedure and the voxel mapping solution. We evaluate our single-view method in Section 4 against state-of-the-art approaches of different mapping paradigms, namely, against sparse and multi-view mapping approaches, showing comparable mapping performance yet without the need to rely on ground-truth scale while running at commendable frame rate. Finally, we conclude and discuss future perspectives in Section 5.

2. Background

Current advances in monocular SLAM enable robust localization and accurate sparse mapping, leading to a resurgence of efforts in dense 3D reconstruction. Section 2.1 therefore reviews recent developments in densifying monocular SLAM using DNNs while section 2.2 reports on the advances in the closely related field of MDE.

2.1. Monocular SLAM Densification

In recent years, the advent of deep learning has revived interest in monocular SLAM densification, enabling the prediction of dense depth maps. These depth maps, when combined using estimated camera poses, facilitate high-quality 3D reconstruction. A selection of notable developments in this field is presented in Table 1.

Method	Year	Metric	Sensors	Localization evaluation	Mapping evaluation	Computing resources	Code available
NERF-SLAM [34]	2023	No	Mono	No	Depth map	RTX 2080 Ti	Yes
Rosinol et al. [35]	2023	No	Mono	No	Point cloud	RTX 2080 Ti	No
CodeMapping [27]	2021	Yes	Mono-IMU	No	Depth map	RTX 3080	No
DROID-SLAM [38]	2021	No	Mono	Yes	N/A	2x RTX 3090	Yes
TANDEM [24]	2021	No	Mono	Yes	Depth map	RTX 2080	Yes
CodeVIO [45]	2021	Yes	Mono-IMU	Yes	Depth map	GTX 1080 Ti	No
DeepRelativeFusion [26]	2021	No	Mono	Yes	Depth map	GTX 1070	No
DeepFactors [11]	2020	No	Mono	Yes	Depth map	GTX 1080	Yes
CodeSLAM [5]	2018	No	Mono	No	N/A	N/A	Yes
CNN-SLAM [37]	2017	No	Mono	Yes	Depth map	Quadro K5200	Yes

Table 1.: Comparative overview of the main monocular SLAM densification methods.

Early efforts fused conventional monocular SLAM with deep learning, in particular with Con-

volutional Neural Networks (CNNs) [26,37], to enhance dense depth estimation. In later works, the use of Conditional Variational Auto-Encoders (CVAE) to learn a compact and dense depth representation was introduced by CodeSLAM [5], a technique further developed by DeepFactors [11] for dense multi-view refinement within a Bundle Adjustment (BA) framework. Building on a monocular-inertial SLAM baseline, CodeMapping [27] and CodeVIO [45] additionally incorporated sparse depth into their CVAE models to improve depth accuracy and perform metric 3D mesh reconstruction. Despite achieving the best accuracy, the implementation details of CodeVIO and CodeMapping have not been released. Meanwhile, DROID-SLAM [38] adopted a Recurrent Neural Network for dense optical flow prediction that was integrated into a dense BA layer. Its mapping process was subsequently enhanced by probabilistic volumetric fusion [35] or with Neural Radiance Fields (NeRF) [34].

Many of these studies rely on pure monocular SLAM, requiring the use of ground-truth depth measurements for scale adjustment. Furthermore, they rely on different depth estimation metrics to evaluate their accuracy since no universal standard for mapping evaluation has been established. Finally, these methods often require significant computational resources and face challenges in generalization due to their over-reliance on supervised learning techniques.

2.2. Monocular Depth Estimation

Conventional approaches rely on supervised learning, which requires a large amount of data with accurate depth ground truth, which may be difficult to collect. Eigen et al. [12] pioneered DNN-based depth estimation, enhancing resolution with a multi-scale CNN approach, while DORN [14] addressed depth estimation as an ordinal regression problem to increase prediction stability and accuracy. MiDaS [32] improved generalizability across diverse datasets through a scale and shift-invariant loss function. Recently, methods such as DPT [31], AdaBins [13], or ZoeDepth [4] have integrated Vision Transformers to maintain high resolution throughout the DNN and predict more detailed depth maps. Building on Transformers, ZeroDepth [19] introduced a zero-shot approach for scale-aware depth estimation that uses variational inference to account for uncertainty and integrates camera intrinsics for improved scale accuracy, positioning it as a leading method.

Self-supervised depth estimation methods, which were popularized by Garg et al. [15], use epipolar geometry constraints for image reconstruction, eliminating the need for ground truth depth data. Monodepth [17] improves upon this approach by utilizing a fully differentiable loss and ensuring depth consistency across stereo images. SfMLearner [43] and Monodepth2 [16] further extend this paradigm to monocular video sequences, resulting in improved occlusion handling and the introduction of multi-scale supervision. PackNet-SfM [18] introduces a 3D convolutional architecture and velocity supervision for scale-aware depth estimation. Recent advancements like DIFFNet [42] and MonoFormer [3] integrate semantic segmentation and hybrid CNN-Transformer architectures, enhancing generalization and performance, whereas MonoViT [41] achieves superior accuracy with significantly reduced complexity.

Research in MDE has been largely centered around autonomous vehicle applications, demonstrating effectiveness on benchmarks built in outdoor urban environments. Although self-supervised methods tend to exhibit enhanced generalization capabilities, the recent integration of Transformers into supervised methods has significantly increased their learning capacity, thereby improving generalization. This aspect remains challenging for MDE, in addition to achieving accurate absolute scale estimation.

To overcome these constraints, we opt for a loosely coupled approach designed for real-time drone operation in indoor environments. We alleviate the computational load of the densification process by decoupling it from SLAM using the GPU, and focusing primarily on voxel map reconstruction. Our solution builds on a sparse monocular-inertial SLAM baseline that provides metric scale information, which is then combined with state-of-the-art MDE.

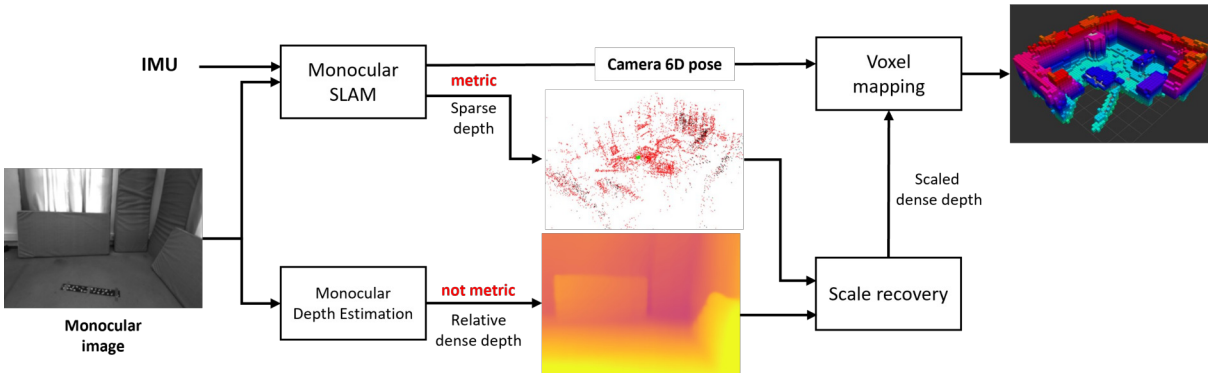


Figure 1.: Proposed loosely coupled pipeline which recovers the scale of the predicted dense depth from the sparse depth estimated by SLAM. The final voxel map is built and maintained by multi-view fusion from estimated camera poses and the scaled dense depth maps.

3. Proposed framework

Figure 1 illustrates our proposed pipeline for densifying monocular SLAM that employs a sparse SLAM baseline alongside a baseline, DNN-based, dense depth map prediction for each keyframe. Depth map prediction DNNs solutions such as Packnet-SfM [18] or ZeroDepth [19] which we tested claim to be scale-aware and exhibit reasonable generalization capabilities. However, due to the significant domain shift in our target environments which include large indoor spaces and variable lighting conditions, we take a slightly different, more realistic standpoint. In particular, we assume that depth values are at least scale-consistent. i.e. they are predicted up to a global, relative scale factor. This leads us to propose a scale recovery step, which we detail in Section 3.1, leveraging the SLAM sparse depth to adjust the dense depth maps and ensure metric scale. Finally, we detail the voxel mapping module in Section 3.2, which utilizes the resulting dense, metric-scaled depth map and the corresponding camera pose estimated by SLAM to incrementally construct a voxel map by raycasting, thereby achieving multi-view volumetric refinement.

3.1. Loosely coupled dense and metric depth estimation

Our scale recovery procedure relies on the assumption of scale consistency. For any pixel p of a frame I_k, the scale factor α_k relates the ground truth depth Z_p^k to the predicted depth $\hat{D}_p^k$ according to the equation:

$$\exists \alpha_k \in \mathbb{R}, \forall p \in I_k \rightarrow Z_p^k = \alpha_k \hat{D}_p^k \tag{1}$$

This is in line with related works that typically estimate dense depth maps up to a global scale factor, which is then recovered in post-processing using ground truth depth measurements collected from a LiDAR. The scale is typically calculated by dividing the median values of the estimated depths by the ground truth depths, as defined by the following equation:

$$\alpha_k = \frac{med(\{Z_p^k, p \in I_k\})}{med(\{\hat{D}_p^k, p \in I_k\})} \tag{2}$$

The use of medians to estimate scale is less prone to outliers than the use of means, but it is still less reliable for smaller datasets where the variance can be high and outliers can have a greater impact.

In contrast, our approach uses sparse depth points triangulated by SLAM, which are less

numerous than ground truth LiDAR points but are available in real time. Typically, LiDAR points account for about 5% of the image density [23]. In Figure 2, we report the number of points triangulated per keyframe as measured during our experiments. Their number tends to be high during initialization and may drop at the end due to texture-less images for instance during landing. Here, the sparse depth retrieved from SLAM points of keyframes represented around 0.1% of the image density on EuRoC/*V101* and only 0.02% on HILTI/*Basement_1*.

Therefore, as empirically suggested in [22], we propose to recover the global scale factor α_k by minimizing the square relative error between SLAM-estimated points of depth D_p^k and those predicted by the DNN and denoted as $\hat{D}_p^k$, as described by the following equation:

$$\hat{\alpha}_k = \min_{\alpha} \frac{1}{N} \sum_{p \in \Omega_k} \frac{\left\| \alpha \hat{D}_p^k - D_p^k \right\|^2}{D_p^k} \tag{3}$$

Here, $\Omega_k \subset I_k$ represents the subset of pixels that could be triangulated by SLAM to determine depth. This strategy is efficient and takes into account the sparse nature of SLAM points and their inherent variability, by a higher penalization of large errors at closer ranges. As we will show in the experiments section, this allows our system to recover the scale at an accuracy that is equivalent to using the ground-truth scale.

Our motivation in using sparse SLAM points as a source for recovering scale can be based on the evolution of the reprojection error per keyframe, as shown in Figure 2. This raises the question of which SLAM algorithm would be better suited for the task of sparse but highly precise point estimation, rather than denser and more erroneous 3D point estimation, as a means for firstly accurately recovering metric scale and secondly densely estimating depth by loosely coupling with monocular depth prediction. In addition, increasing the number of initial 3D points would

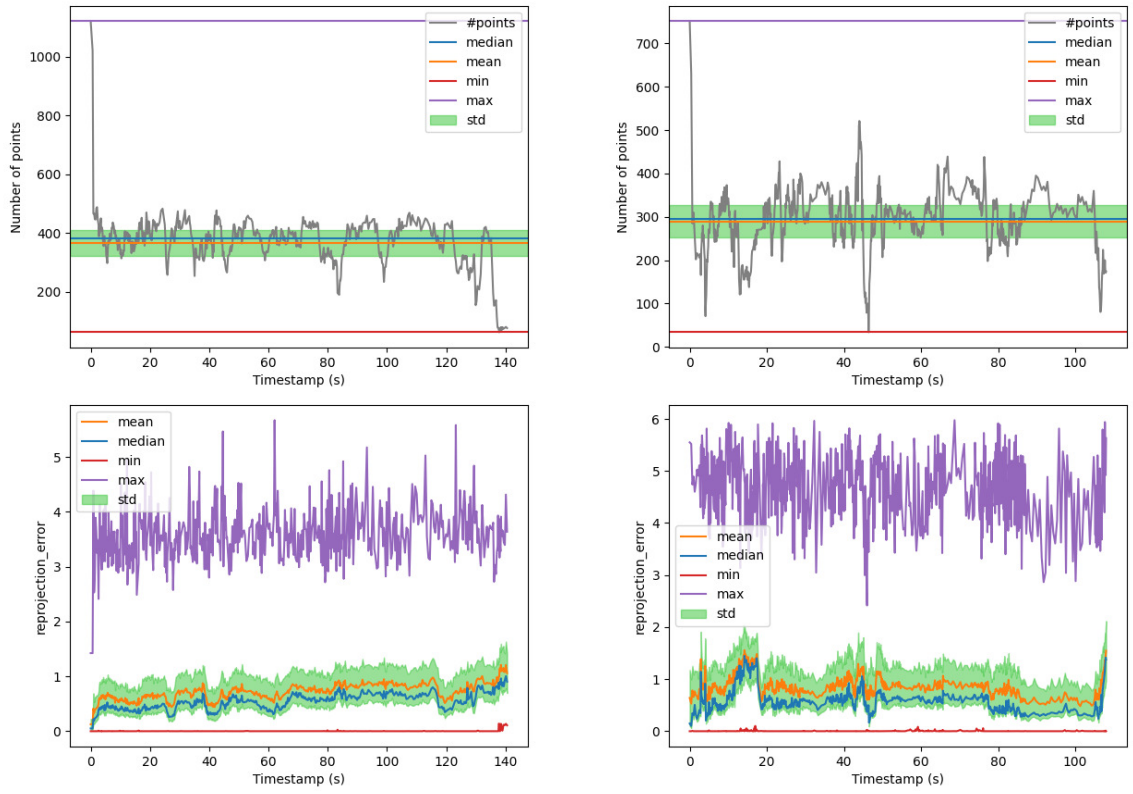


Figure 2.: Evolution of the landmarks density (top) and reprojection error (bottom) per keyframe on EuRoC *V101* (left) and HILTI *Basement_1* (right) sequences.

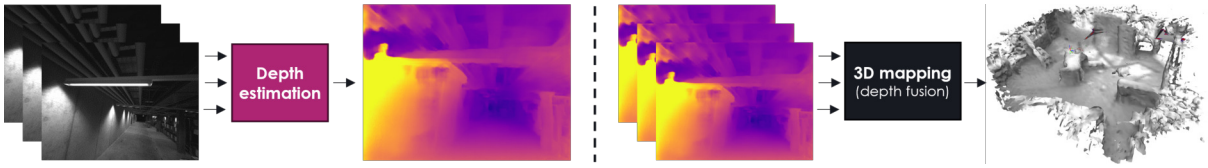


Figure 3.: Diagram of multi-view approaches for depth estimation (left) and 3D mapping (right), where the latter requires depth maps with corresponding camera poses.

increase the computational overhead of the overall pipeline and impact real-time performance.

Following extensive experimentation (see [20] and [21]) of the state-of-the-art monocular approaches Kimera [33], ORB-SLAM 3 [8] and Basalt [39] in different environments in terms of trajectory error and scene coverage, we have opted for using ORBSLAM 3 as it attained an overall superior performance. In particular, ORB-SLAM 3 employs Bundle Adjustment to triangulate points by minimizing the reprojection error, making this metric a good indicator of the confidence level in landmark estimation. As can be seen, an average reprojection error below 1 pixel was measured, with only occasional spikes reaching up to 5 pixels which reinforces the notion that the sparse points estimated by SLAM are indeed a reliable source for scale recovery.

3.2. Voxel map construction

In the proposed pipeline, we rely on a single keyframe for dense depth estimation. However, unlike MDE, multi-view approaches enforce both temporal and geometric consistency, which are essential in Visual SLAM. These two paradigms are depicted in Figure 3.

One common approach applied to monocular SLAM densification is to leverage multiple views to improve depth estimation. This can be achieved through various methods such as integrating keyframe depth codes into a factor graph [11,27,45], implementing multi-view stereo networks [24], or employing recurrent neural networks [38]. These methods improve accuracy but tend to be computationally expensive due to the joint optimization of a larger set of variables, which may become intractable as the map grows in size.

An alternative approach, commonly applied for 3D reconstruction, consists in leveraging multiple views in the mapping task, instead of the depth estimation task, by fusing previously estimated dense depth maps. For instance, the integration of NeRF into SLAM [10,34,44] enables high-quality 3D mapping but is computationally complex due to the necessity of real-time NeRF model learning from SLAM data. More conventional methods often utilize volumetric fusion of depth maps derived from RGB-D sensors or stereo matching [30,33], constructing the fusion scheme based on the depth model of the sensor.

Along this line, we integrate Voxblox [30] in the mapping task similarly to the Kimera [33] pipeline. Yet, instead of relying on stereo vision, we build on the procedure presented earlier to obtain dense and metric depth maps from single views. Thus, given an estimated camera pose, Voxblox uses raycasting to update a Truncated Signed Distance Function (TSDF) iteratively from each depth map. Performing this operation for multiple viewpoints allows for a volumetric fusion. The selection of voxel map representation for navigation is beneficial due to its scalability, geometric reasoning, and inherent suitability for path planning with TSDF and Euclidean Signed Distance Function (ESDF) that can generate occupancy grids and metric distances to obstacles [9,29]. Additionally, voxel hashing may enable further optimization in map usage and storage [29]. The overall process is already relatively efficient and can be even further optimized by parallelizing it on a GPU [28].

Voxblox provides some alternative construction strategies, specifically *merged* and *fast* integrators. Unlike the canonical raycasting procedure, the *merged* integrator uses bundled raycasting, which groups together all pixels that end in the same voxel and casts a single ray, merging their values. This process is illustrated in Figure 4. Otherwise, the *fast* integrator proposes casting rays from the point cloud towards the sensor origin, optimized by early termination of rays that

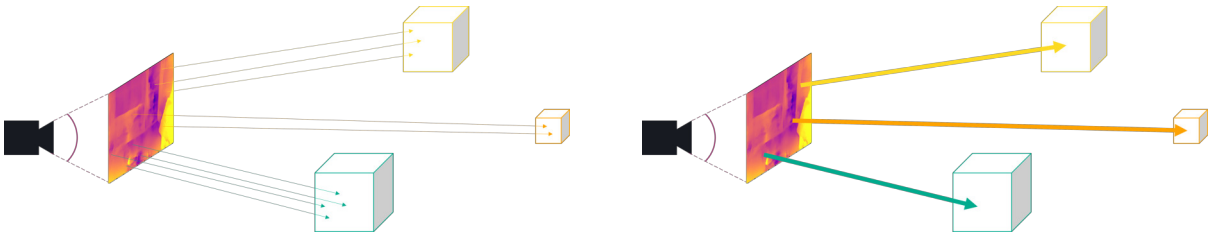


Figure 4.: Diagram of simple (left) and bundled (right) raycasting technique.

intersect a sequence of voxels that have already been updated. This process can be accelerated by limiting the number of ray-casts per voxel, resulting in up to tenfold speed improvements compared to other techniques, particularly for small voxels.

4. Experiments

This section presents the results of densifying monocular SLAM. First, we evaluate our scale recovery process and compare it to related works. Then, we present qualitative outcomes from our complete 3D mapping pipeline and discuss their implications for indoor drone navigation. For brevity, we only provide a subset of qualitative experiments within the text and refer the readers to the complete video recordings available on-line (link provided in page 10). The experiments were conducted on a laptop equipped with an Intel i7-8750H CPU, 16 GB of memory, and a NVIDIA RTX 2080 Mobile GPU.

4.1. Metric dense depth estimation

The evaluation of our scale recovery process requires a dataset with monocular-inertial data for SLAM and structural ground truth for depth evaluation. The EuRoC dataset [6] provides accurate structural ground truth. Specifically, we used the *V101* sequence, which contains a 3D LiDAR scan of the environment. Using the ground truth 6D poses, we back-projected the point cloud to generate the corresponding depth maps.

The results are summarized in Table 2, where we present the results from related works as reported in their corresponding publications. We used standard depth estimation metrics [12], including Absolute Difference (*abs_diff*), Absolute Relative Difference (*abs_rel*), Squared Relative Difference (*sqr_rel*), and Root Mean Squared Error (*rmse*) with and without log scale, to assess the accuracy of predicted depths compared to ground truth values. The Accuracy Rate δ_i is calculated as the ratio of depth predictions falling within 1.25^i of the true depth, where i ranges from 1 to 3, offering a gradual measure of prediction accuracy.

First, we report ORB-SLAM 3 [8] sparse depth evaluation. Since fewer points are considered, the measurements are more affected by high error values. However, it still achieves great performance, with a δ_1 accuracy of almost 90% of valid points, which supports our hypothesis that the sparse depth estimated by ORB-SLAM 3 is a reliable source for scale recovery.

Subsequently, we present the outcomes of some related works that rely on a multi-view approach. DeepFactors [11] and TANDEM [24] are not metric and require depth ground truth for scale alignment. Nevertheless, TANDEM gets an excellent δ_1 accuracy of 94.25% of valid points. On the other hand, CodeVIO [45] and CodeMapping [27] are metric techniques that achieve the best results, with a Root Mean Square Error (*rmse*) below 0.5 meters.

Finally, we discuss the results of our approach without multi-view processing. The original predictions of PackNet-SfM [18] show a significant scale error, with a *rmse* exceeding 7 meters. After applying scale adjustment, both the GT-scale and SR-scale methods yield very similar results, confirming the effectiveness of our approach for metric scale recovery.

Additional experimentation of this procedure with ZeroDepth [19], which inherently provides

Category	Method	abs_diff	abs_rel	sq_rel	rmse	rmse_log	δ_1	δ_2	δ_3
Sparse	ORB-SLAM 3 [8]	0.284	0.156	0.266	0.572	0.222	89.8%	94.6%	96.5%
Multi-view	DeepFactors [11] (GT-scale)	0.842			1.050				
	TANDEM [24] (GT-scale)						**94.25%**		
	CodeVIO [45]				0.468		87.0%	**95.2%**	**97.9%**
	CodeMapping [27]	**0.192**			**0.381**				
Single-view	PackNet-SfM	6.309	2.720	22.945	7.267	1.258	1.3%	4.4%	12.0%
	PackNet-SfM (GT-scale)	0.807	0.331	0.530	1.145	0.396	48.2%	76.0%	89.6%
	PackNet-SfM (SR-scale)	0.792	0.318	0.443	1.063	0.418	43.2%	72.7%	87.9%
	ZeroDepth	0.791	0.285	0.326	0.953	0.392	38.8%	70.6%	88.8%
	ZeroDepth (GT-scale)	0.386	0.167	0.152	0.545	**0.217**	81.3%	92.5%	96.6%
	ZeroDepth (SR-scale)	0.412	**0.167**	**0.145**	0.569	0.222	77.0%	92.5%	96.6%

Table 2.: Evaluation of depth estimation performance in the EuRoC *V101* scene, with measurements in meters except for the δ_i metrics. The best value for each metric is denoted in bold for dense evaluations, excluding the initial row.

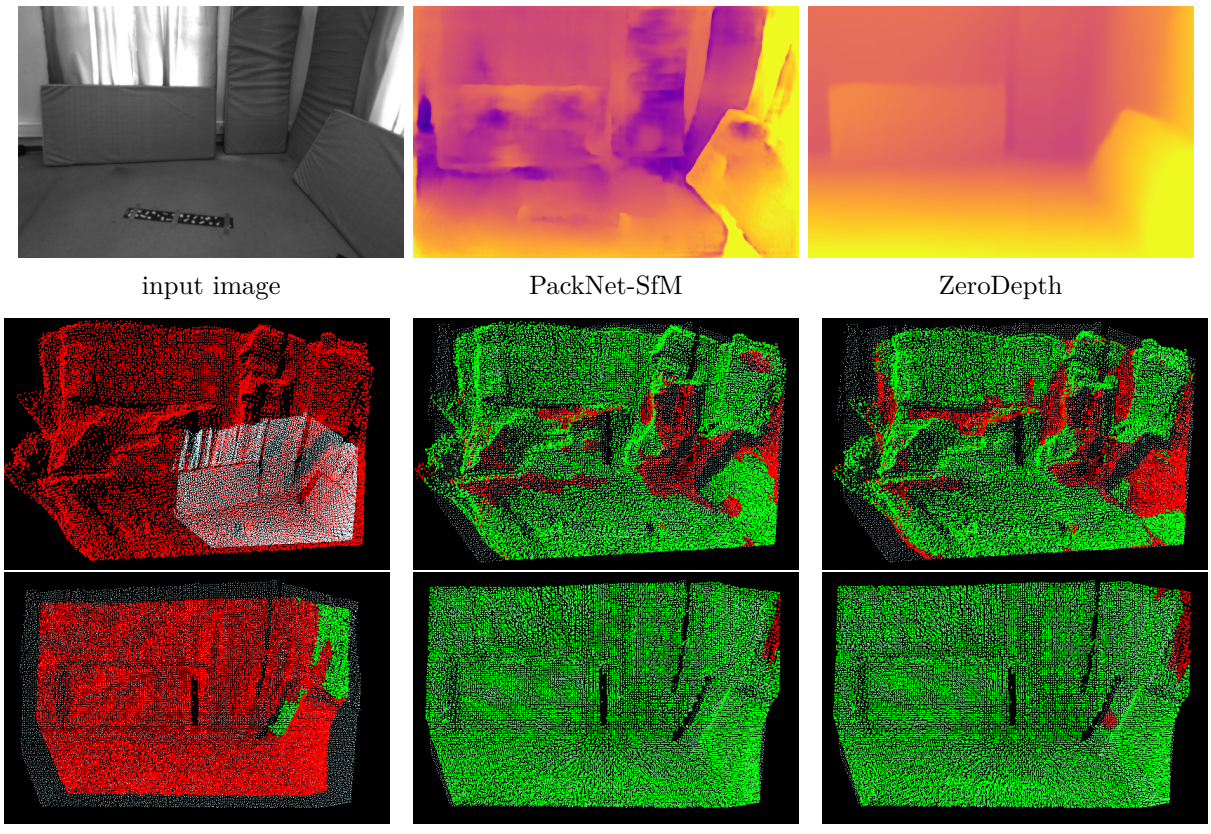


Figure 5.: Monocular depth estimation results on the *V101* scene from EuRoC. The first row shows the predicted dense depth maps. Subsequent rows show 3D visualizations comparing PackNet-SfM (middle) and ZeroDepth (bottom) predictions to the ground truth using the δ_1 metric. Depth maps, with and without scaling, are projected onto a 3D point cloud. Ground truth points are shown in white, correct predictions in green, and incorrect ones in red.

more accurate predictions, attains a *rmse* below 1 meter before scale correction. Our scale recovery method demonstrates effective scale recovery compared to the ground truth, despite relying on significantly fewer points. Although our solution has a larger error than CodeMapping and CodeVIO, it has lower computational requirements by operating on single frames.

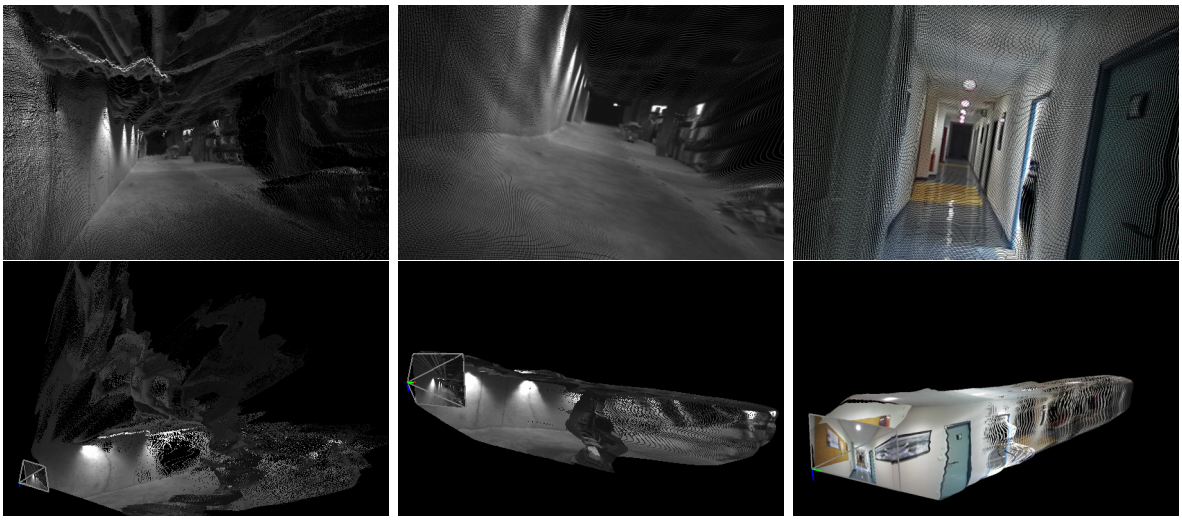
On the other hand, ZeroDepth uses a variational approach to sample depth maps from a latent space. The authors generate 10 samples to derive the final depth map from the mean and

	PackNet-SfM	ZD01	ZD02	ZD03	ZD04	ZD05	ZD06	ZD07	ZD08	ZD09	ZD10
Inference (ms)	9.2	13.9	19.7	22.5	24.8	29.1	144.6	456.0	804.9	1152.9	1501.1

Table 3.: Average inference time in milliseconds for PackNet-SfM and ZeroDepth (ZD) on the *V101* sequence of EuRoC, measured using 1 to 10 samples.

its uncertainty from the standard deviation. Predictions with low confidence can then be excluded using the uncertainty. As shown in Table 3, increasing the number of samples significantly affects the inference time. However, our experiments on the EuRoC dataset revealed only negligible differences in uncertainty and accuracy when we varied the number of samples. This suggests that it is possible to use fewer samples to improve speed without compromising performance.

The effectiveness of our scale recovery procedure is showcased in Figure 5 through 3D visualizations, highlighting the improvement over initial deep learning model predictions in relation to the δ_1 metric. This example also demonstrates the superiority of ZeroDepth over PackNet-SfM as its predictions are more accurate and much smoother. As showcased in Figure 5, PackNet-SfM can struggle to accurately infer depth for certain structures, resulting in noisy predictions along edges. However, it generally performs well in distinguishing surfaces. Originally trained on outdoor datasets, its predictions for upper areas in indoor scenes, as shown in Figure 6, tend to be inaccurate due to its correlation with the sky during training. In contrast, ZeroDepth produces smoother and less noisy predictions overall, although in grayscale or poorly illuminated images, the performance of the algorithm becomes inconsistent, resulting in flat depth maps and inaccurate capture of planar structures such as floors and walls. Finally, when employed to a scene explored in a similar environment, PackNet-SfM yielded unsatisfactory results while ZeroDepth demonstrated improved performance, accurately estimating planar structures and providing more consistent predictions. It is important to note that this particular scene is better illuminated and that the most distant points were often flattened.



(a) (b) (c)

Figure 6.: (a)-(b) Visualization of depth prediction from *Basement_1* scene of HILTI dataset and (c) from a sequence collected for an office corridor in IMT Atlantique University. The first and second rows depict the obtained dense point clouds from different viewpoints. Column (a) was predicted by PackNet-SfM, (b) and (c) by ZeroDepth.

4.2. Voxel mapping

Despite the existence of different voxel map evaluation methods [1,29], to our knowledge, these have not been employed in the context of SLAM or MDE performance assessment. Therefore, we present qualitative results from our complete dense and metric 3D mapping pipeline on the *V101* sequence of EuRoC [6]. The map is constructed by raycasting the depth maps scaled using our approach for each keyframe from its corresponding camera pose estimated by SLAM. Using the 3D visualization tool RViz, we present in Figure 7 a sample of the resulting voxel map. All the results can be visualized through multiple videos on a dedicated web-page[1].

We employ two distinct integration techniques, namely *merged* (bundled raycasting) and *fast*, as introduced in Section 3.2. To facilitate qualitative comparison, the ground truth 3D LiDAR scan of the room is shown in white. Ideally, all ground truth points should fall within the surface voxels. However, due to an average Absolute Trajectory Error (ATE) of 49 mm on this sequence, a slight misalignment can be observed. For comparison, we ran a Voxblox example specifically tuned for this sequence, where the map was constructed from the camera's ground truth poses and depth maps estimated using stereo vision. The stereo-based approach processed 1,151 frames, while we used only 529 keyframes.

In our initial experiments, using a voxel size of 100 mm, we were able to generate a dense 3D map of the room at the metric scale without having to rely on the ground truth, as shown in Figure 7. Compared to the stereo method, our approach captured a wider area and tends to perform better on regions with uniform textures. However, the reconstructed map is coarse. It can be noticed that some details were not correctly mapped, especially around edges and thin objects. Nevertheless, since our method relies on fewer images, some places tend to be less covered. Mapping errors can also be seen in the upper part of the map (ceiling), resulting from initial predictions that were not subsequently captured by the camera and thus remain uncorrected. These phenomena can be seen in the videos available on the provided website.

Limitations

The example presented in Figure 8 shows the negative impact that erroneous DNN depth prediction may have on our approach. The noisy depth output around the edges of the ladder, where there are important depth discontinuities, results in inaccurate mapping, particularly when using the *merged* integrator, which only partially reconstructs the object. The limited coverage of this area in only a few keyframes restricted further refinement through multi-view methods. On the other hand, we believe these limitations to be less intrinsic to the SLAM system and more related to the drone trajectory and 3D scene layout that may accentuate or attenuate such effects.

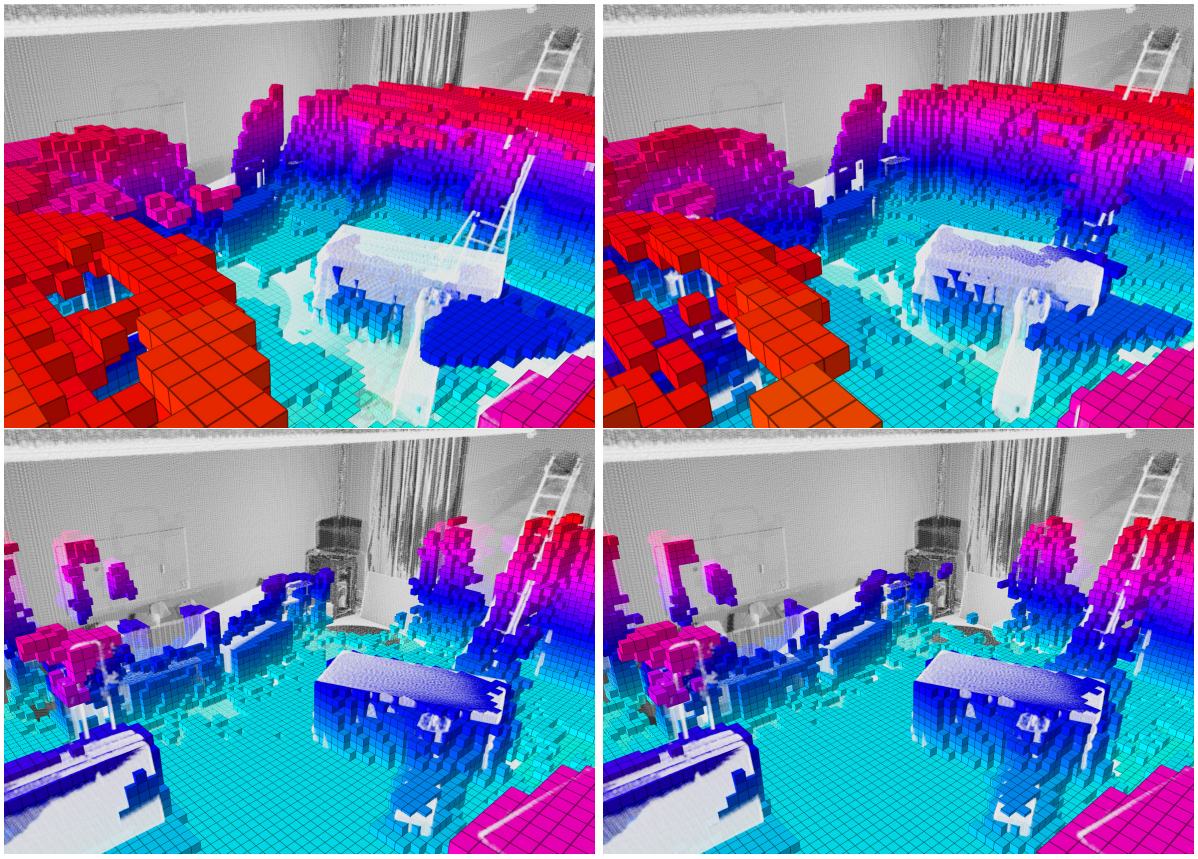
In contrast, the *fast* integrator yields finer results compared to the *merged* one. It preserves the original estimates while limiting voxel updates during raycasting whereas when using bundled raycasting, groups of points are actually grouped together, producing a coarser result. The runtime analysis provided in Table 4 confirms the efficiency of the *fast* approach, which is on average 3.65 times faster than the *merged* one.

Integrator	*merged*		*fast*	
Voxel size (m)	0.10	0.20	0.10	0.20
Stereo	48.6	9.5	35.0	8.0
Our	86.4	25.5	69.4	22.7

Table 4.: Average integration time in milliseconds with Voxblox using *merged* and *fast* integrators, different voxel sizes, and methods on the *V101* sequence of the EuRoC dataset.

Finally, it should be noted that the assumption that dense depth maps from DNNs are scale-consistent varies according to the degree to which the encountered environments during testing

[1] https://yhabib29.github.io/monocular_slam_densification



(a) *merged* (b) *fast*

Figure 7.: Voxel mapping EuRoC *V101* sequence using voxel size of 100 mm. The first row presents our results, while the second row shows those of the stereo-based reference.

correspond to scene layouts that are similar to those encountered in training.

Characteristically, if a DNN was trained outdoors but tested indoors, then the scale consistency assumption may only be partially valid because the network will tend to estimate large depths at the top of an indoor scene since this area mostly corresponded to the sky or high structures encountered in training time, whereas in indoor environments the upper part corresponds to a ceiling. In such cases, a global scale factor may not be sufficiently representative and scale recovery may negatively impact the results. Nevertheless, in this particular situation, the global scale factor will be estimated using triangulated points on textured surfaces, thus excluding sky-like regions or the ceiling. To reinforce the scale consistency assumption, our current work employs ZeroDepth which has been trained in both indoor and outdoor environments, as opposed to PackNet-SfM which has been trained only outdoors.

5. Conclusion

This work provides a comprehensive presentation of a pipeline for monocular SLAM densification, tailored for real-time drone operation. In support of the proposed scale recovery procedure, we provide additional experiments in which we leverage geometrically-obtained sparse depth and demonstrate its effectiveness by employing state-of-the-art, deep-learning-based monocular depth estimation. In addition, we conducted a qualitative analysis of the complete mapping framework by reconstruction of dense and metric voxel maps. Despite the use of iterative multi-view volumetric refinement, the provided experiments further shed light on the influence of the quality of the underlying SLAM and MDE techniques.

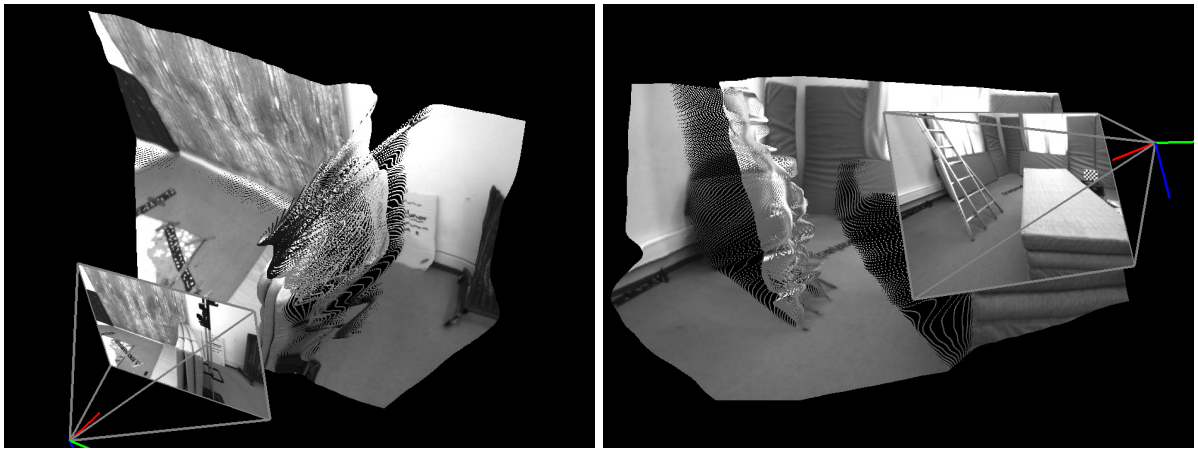


Figure 8.: Visualization of ZeroDepth prediction showing noise around edges and discontinuities.

In view of these results, certain directions for further development can be identified. Firstly, pursuing a more comprehensive quantitative evaluation would require datasets with precise pose and 3D structural ground truth. Such data could be obtained through simulations like TartanAir [40] or extracted from datasets containing laser scans using state-of-the-art LiDAR-based SLAM techniques. Furthermore, a more systematic evaluation of the voxel mapping could be improved by using standardized metrics. To improve accuracy, transitioning to a tightly coupled approach would allow leveraging DNN capabilities to infer dense and metric depth from monocular images and SLAM sparse depth simultaneously. To improve efficiency, we could opt for hardware acceleration using field programmable gate arrays (FPGA), neural processing units (NPU), and deep learning accelerators (DLA) as well as porting SLAM and/or ray casting used in voxel mapping to a GPU.

6. Acknowledgement

This work has been financed by ANRT (Association Nationale de la Recherche et de la Technologie) under the CIFRE grant 2019/1877.

References

[1] Aravecchia, Stéphanie, Marianne Clausel, and Cédric Pradalier. 2024. "Comparing metrics for evaluating 3D map quality in natural environments." *Robotics and Autonomous Systems* 173: 104617.

[2] Bachrach, Abraham, Ruijie He, and Nicholas Roy. 2009. "Autonomous flight in unknown indoor environments." *International Journal of Micro Air Vehicles* 1 (4): 217–228.

[3] Bae, Jinwoo, Sungho Moon, and Sunghoon Im. 2023. "Deep Digging into the Generalization of Self-Supervised Monocular Depth Estimation." *AAAI* .

[4] Bhat, Shariq Farooq, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. 2023. "ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth." `https://arxiv.org/abs/2302.12288`.

[5] Bloesch, Michael, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. 2018. "CodeSLAM-learning a compact, optimisable representation for dense visual SLAM." In *IEEE conference on computer vision and pattern recognition*, .

[6] Burri, Michael, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. 2016. "The EuRoC micro aerial vehicle datasets." *The International Journal of Robotics Research* 35 (10): 1157–1163.

[7] Cadena, Cesar, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. 2016. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age." *IEEE Transactions on Robotics* 32 (6): 1309–1332.

[8] Campos, Carlos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. 2021. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM." *IEEE Transactions on Robotics* 37 (6): 1874–1890.

[9] Choe, Yungeun, Inwook Shim, and Myung Jin Chung. 2011. "Geometric-featured voxel maps for 3D mapping in urban environments." In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, .

[10] Chung, Chi-Ming, Yang-Che Tseng, Ya-Ching Hsu, Xiang-Qian Shi, Yun-Hung Hua, Jia-Fong Yeh, Wen-Chin Chen, Yi-Ting Chen, and Winston H. Hsu. 2023. "Orbeez-SLAM: A Real-time Monocular Visual SLAM with ORB Features and NeRF-realized Mapping." In *IEEE International Conference on Robotics and Automation*, .

[11] Czarnowski, Jan, Tristan Laidlow, Ronald Clark, and Andrew J Davison. 2020. "Deepfactors: Real-time probabilistic dense monocular slam." *IEEE Robotics and Automation Letters* 5 (2): 721–728.

[12] Eigen, David, Christian Puhrsch, and Rob Fergus. 2014. "Depth map prediction from a single image using a multi-scale deep network." *Advances in neural information processing systems* 27.

[13] Farooq Bhat, Shariq, Ibraheem Alhashim, and Peter Wonka. 2021. "AdaBins: Depth Estimation Using Adaptive Bins." In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, .

[14] Fu, Huan, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. "Deep Ordinal Regression Network for Monocular Depth Estimation." In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, .

[15] Garg, Ravi, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. 2016. "Unsupervised cnn for single view depth estimation: Geometry to the rescue." In *European conference on computer vision*, .

[16] Godard, Clement, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. 2019. "Digging Into Self-Supervised Monocular Depth Estimation." In *IEEE/CVF International Conference on Computer Vision*, .

[17] Godard, Clément, Oisin Mac Aodha, and Gabriel J Brostow. 2017. "Unsupervised monocular depth estimation with left-right consistency." In *IEEE conference on computer vision and pattern recognition*, .

[18] Guizilini, Vitor, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 2020. "3d packing for self-supervised monocular depth estimation." In *IEEE Conference on Computer Vision and Pattern Recognition*, .

[19] Guizilini, Vitor, Igor Vasiljevic, Dian Chen, Rareș Ambruș, and Adrien Gaidon. 2023. "Towards Zero-Shot Scale-Aware Monocular Depth Estimation." In *IEEE/CVF International Conference on Computer Vision (ICCV)*, .

[20] Habib, Yassine. 2024. "Monocular SLAM densification for 3D mapping and autonomous drone navigation." PhD diss. 2024IMTA0390, `http://www.theses.fr/2024IMTA0390`.

[21] Habib, Yassine, Panagiotis Papadakis, Antoine Fagette, Cédric Le Barz, Tiago Gonçalves, and Cédric Buche. 2023. "From sparse SLAM to dense mapping for UAV autonomous navigation." In *Geospatial Informatics XIII*, .

[22] Habib, Yassine, Panagiotis Papadakis, Cédric Le Barz, Antoine Fagette, Tiago Gonçalves, and Cédric Buche. 2023. "Densifying SLAM for UAV Navigation by Fusion of Monocular Depth Prediction." In *IEEE International Conference on Automation, Robotics and Applications*, .

[23] Hu, Junjie, Chenyu Bao, Mete Ozay, Chenyou Fan, Qing Gao, Honghai Liu, and Tin Lun Lam. 2023. "Deep Depth Completion From Extremely Sparse Data: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (7): 8244–8264.

[24] Koestler, Lukas, Nan Yang, Niclas Zeller, and Daniel Cremers. 2022. "Tandem: Tracking and dense mapping in real-time using deep multi-view stereo." In *Conference on Robot Learning*, .

[25] Kruijff, Geert-Jan M., Fiora Pirri, Mario Gianni, Panagiotis Papadakis, Matia Pizzoli, Arnab Sinha, Viatcheslav Tretyakov, et al. 2012. "Rescue robots at earthquake-hit Mirandola, Italy: A field report." In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, .

[26] Loo, Shing Yan, Syamsiah Mashohor, Sai Hong Tang, and Hong Zhang. 2021. "Deeprelativefusion: Dense monocular slam using single-image relative depth prediction." In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, .

[27] Matsuki, Hidenobu, Raluca Scona, Jan Czarnowski, and Andrew J. Davison. 2021. "CodeMapping: Real-Time Dense Mapping for Sparse SLAM using Compact Scene Representations." *IEEE Robotics and Automation Letters* 6 (4): 7105–7112.

[28] Millane, Alexander, Helen Oleynikova, Emilie Wirbel, Remo Steiner, Vikram Ramasamy, David Tingdahl, and Roland Siegwart. 2023. "nvblox: GPU-Accelerated Incremental Signed Distance Field Mapping." `https://arxiv.org/abs/2311.00626`.

[29] Muglikar, Manasi, Zichao Zhang, and Davide Scaramuzza. 2020. "Voxel Map for Visual SLAM." In *IEEE International Conference on Robotics and Automation*, .

[30] Oleynikova, Helen, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. 2017. "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning." In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, .

[31] Ranftl, René, Alexey Bochkovskiy, and Vladlen Koltun. 2021. "Vision Transformers for Dense Prediction." In *IEEE/CVF International Conference on Computer Vision*, .

[32] Ranftl, René, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2022. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (3): 1623–1637.

[33] Rosinol, Antoni, Marcus Abate, Yun Chang, and Luca Carlone. 2020. "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping." In *IEEE International Conference on Robotics and Automation*, .

[34] Rosinol, Antoni, John J. Leonard, and Luca Carlone. 2022. "NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields." `https://arxiv.org/abs/2210.13641`.

[35] Rosinol, Antoni, John J Leonard, and Luca Carlone. 2022. "Probabilistic Volumetric Fusion for Dense Monocular SLAM." `https://arxiv.org/abs/2210.01276`.

[36] Schubert, David, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. 2018. "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry." In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, .

[37] Tateno, Keisuke, Federico Tombari, Iro Laina, and Nassir Navab. 2017. "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction." *CoRR* abs/1704.03489.

[38] Teed, Zachary, and Jia Deng. 2021. "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras." *CoRR* abs/2108.10869.

[39] Usenko, Vladyslav, Nikolaus Demmel, David Schubert, Jörg Stückler, and Daniel Cremers. 2020. "Visual-Inertial Mapping With Non-Linear Factor Recovery." *IEEE Robotics and Automation Letters* 5 (2): 422–429.

[40] Wang, Wenshan, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. 2020. "TartanAir: A Dataset to Push the Limits of Visual SLAM." In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4909–4916.

[41] Zhao, Chaoqiang, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. 2022. "MonoViT: Self-Supervised Monocular Depth Estimation with a Vision Transformer." In *International Conference on 3D Vision*, .

[42] Zhou, Hang, David Greenwood, and Sarah Taylor. 2021. "Self-Supervised Monocular Depth Estimation with Internal Feature Fusion." In *British Machine Vision Conference*, .

[43] Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G Lowe. 2017. "Unsupervised learning of depth and ego-motion from video." In *IEEE conference on computer vision and pattern recognition*, .

[44] Zhu, Zihan, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. 2022. "NICE-SLAM: Neural Implicit Scalable Encoding for SLAM." In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, .

[45] Zuo, Xingxing, Nathaniel Merrill, Wei Li, Yong Liu, Marc Pollefeys, and Guoquan Huang. 2021. "CodeVIO: Visual-Inertial Odometry with Learned Optimizable Dense Depth." In *IEEE International Conference on Robotics and Automation*, .