Expansion of Visual Hints for Improved Generalization in Stereo Matching

Andrea Pilzer[1]* Yuxin Hou[2,3] Niki Loppi[1] Arno Solin[2] Juho Kannala[2]

[1]NVIDIA [2]Aalto University [3]Niantic

{apilzer,nloppi}@nvidia.com, {yuxin.hou, arno.solin, juho.kannala}@aalto.fi

Abstract

We introduce visual hints expansion for guiding stereo matching to improve generalization. Our work is motivated by the robustness of Visual Inertial Odometry (VIO) in computer vision and robotics, where a sparse and unevenly distributed set of feature points characterizes a scene. To improve stereo matching, we propose to elevate 2D hints to 3D points. These sparse and unevenly distributed 3D visual hints are expanded using a 3D random geometric graph, which enhances the learning and inference process. We evaluate our proposal on multiple widely adopted benchmarks and show improved performance without access to additional sensors other than the image sequence. To highlight practical applicability and symbiosis with visual odometry, we demonstrate how our methods run on embedded hardware.

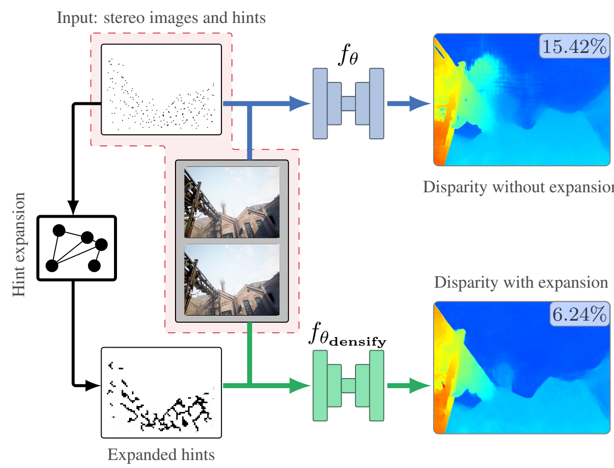

Figure 1: Visual hints expansion for deep stereo matching. (**top**) Inference with a model θ trained with sparse visual hints, and (**bottom**) with a model $\theta_{\mathbf{densify}}$ trained on expanded visual hints. 3D expanded hints lead to more accurate predictions: the overlay labels show the error rate >3.

1. Introduction

Accurate depth estimation is an important task for many 3D applications such as AR/VR and robotics navigation. Technological advances have made active depth sensors (*e.g.*, LiDaR) affordable. Yet, they have the drawback of providing only sparse depth maps. Algorithmic techniques are still more common for dense depth predictions, where deep learning approaches [11, 12, 19, 6, 14, 10, 17, 13, 18] have overtaken traditional matching techniques [5, 20, 29, 15, 4, 42], as their accuracy has kept improving with bigger annotated data sets available [22, 23, 21]. Nevertheless, techniques based on geometric computer vision are still viable options for scarce data or for ensuring good out-of-domain performance.

Purely image-based dense 3D reconstructions can be computed using state-of-the-art photogrammetry software (*e.g.*, Metashape, ReCap Pro). However, many of these photogrammetry tools do not perform robustly if the scene has numerous textureless surfaces, such as in typical indoor office spaces. They also require high-resolution images to match fine surfaces resulting in high computational costs. In practice, visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) techniques [24, 34] are typically employed for real-time camera motion estimation (*e.g.*, ARKit, ARCore, *etc.*) as well as in classical computer vision pipelines, which reconstruct sparse point cloud models based on matching of local features between multiple registered images (*e.g.*, COLMAP [31, 30]).

In this work, we propose VIO guidance for improved robustness and more accurate predictions of stereo matching pipelines on data with domain shift (see Fig. 1). We argue that by exploiting synthetic data sets, stereo matching algorithms have increased their performance but may need costly fine tuning—typically on real data. Our work hinges on the realization, that visual localization methods (*e.g.*, SLAM) can provide a valuable source of sparse 3D world information to guide our algorithm towards accurate

*Work done while at Aalto University

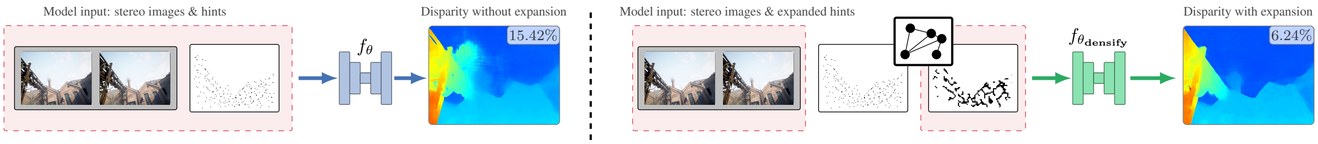

Figure 2: Guided Stereo Matching Pipeline. **Left** vanilla guided stereo matching, model inputs are stereo images and hints (from VIO or LiDaR). **Right** expanded hints for guided stereo matching, model inputs are stereo images and expanded hints.

dense predictions. Seminal work in this direction was presented by Poggi *et al.* [26], who considered sparse uniformly distributed guidance (*e.g.*, from a LiDaR), to guide feature matching at lower scale (*i.e.* to build the cost volume) Fig. 2-*left*. However, our work has two key differences. First, visual guidance is sparse and non-uniformly distributed, ranging from tens to a few hundred points. Second, visual guidance may be imprecise at some locations, which requires filtering to improve robustness. Sparse and uneven VIO hints at lower scale could be discarded due to downsampling. Therefore, an expansion is used to improve guidance and our quantitative and qualitative experiments prove its effectiveness.

To address the previously mentioned challenges, we propose a 3D graph based hints expansion Fig. 2-*right*. Expansion of sparse hints has been considered before by, *e.g.*, Huang *et al.* [16] who proposed a constant expansion for close points. We argue this is a too strict condition, and a slanted linear expansion as in [2] is more suitable. Therefore, we consider the guidance points not in 2D but in 3D. Intuitively, close points on the 2D image plane may be very far in their 3D position. With this in mind, we turn hints into nodes and connect them with edges only if they are close in the 3D world. After obtaining the graph, we linearly approximate the disparity with a 3D slanted line following the edges. In our case, slanted 3D lines do not add any additional computational overhead other than assigning the corresponding disparity value to the pixels on the graph edges. Furthermore, we devise a 3D linear approximation of our graph expansion. By building upon a heuristic that searches for pixels only along vertical and horizontal lines and dividing the images in non-overlapping patches, our approach becomes efficient in connecting hints with 3D slanted lines.

We leverage our expanded hints for deep stereo matching with DeepPruner [10], where their role is to guide the differentiable patch match in a narrow range instead of the full disparity range. We demonstrate that sparse visual guidance is accurate enough to lead the model towards correct predictions. Extensive experimental results show that expanded guidance improves the performance of deep stereo matching on unseen data sets at training time. Following [26], we also demonstrate the proposed contributions on PSMNet [6]. However, expanded visual hints may contain errors, and to this end, we propose a confidence-based guidance filtering. In PSMNet, feature activation maps of both stereo images are extracted to build the 3D cost volume. We filter hints if the feature of the depth hint in the reference frame (*i.e.* left stereo view) is not similar-enough to the corresponding feature in the right stereo view.

We summarize our contributions as follows: *(i)* We propose a novel 3D graph based hints expansion which circumvents pitfalls in previous expansion methods by considering the 3D neighbourhood of the guidance. *(ii)* We devise a 3D linear approximation of our graph guidance, and show that this is an efficient heuristic. *(iii)* We leverage our expanded hints on DeepPruner [10], where their role is to guide the differentiable patch match in a narrow range instead of the full disparity range. *(iv)* Expanded visual hints may still be error prone, and we propose a confidence based guidance filtering method for improved robustness.

2. Related Work

Stereo Matching has a long history in computer vision and has well established benchmarks [28, 33]. Traditional methods are based on local [15, 4, 42] or global matching [5, 20, 29]. Local algorithms are faster and work well in textured regions, while global matching algorithms are computationally more expensive and work well also in textureless regions. Recently, deep learning based methods [19, 6, 14, 10, 17, 25] have shown superior performance over traditional methods. Deep architectures typically use convolutional neural networks (CNN) as feature extractors, U-Net style architectures [12, 11, 13], and cost volumes aggregation [17, 18, 6]. In order to improve cost volume construction, Guo *et al.* [14] proposed a feature correlation method. At the same time, Duggal *et al.* [10] revisited patch matching in a differentiable way allowing end-to-end learning of CNNs. HITNet [37] showed that slanted surfaces allow for smooth and accurate disparity prediction. We take inspiration from their work in devising a hint expansion algorithm that exploits the local slanted nature of disparity.

Guided Stereo Matching. CNNs offer accurate dense predictions but suffer of overconfidence and domain-shift with out-of-distribution (OOD) data. Poggi *et al.* [26] proposed a first attempt to address these issues through a feature enhancement method that exploits sparse LiDaR guidance. Later, [16] further extended [26] by expanding sparse guidance and learning where to confidently use it. Unlike them, we employ visual hints as a sparse and unevenly distributed guidance. This poses a greater challenge as we cannot assume neighbouring pixels to have a similar disparity. On the other hand, long term tracking of sparse feature points in VIO can provide accurate triangulation and help solving ambiguous regions in two-view stereo. VIO algorithms find strong features to match on edges or corners, where also disparity quickly transitions. Quick transitions are harder to model. Bleyer *et al.* [2] proposed slanted planes as a locally linear approximation.

Beyond stereo matching, Sinha *et al.* [35] proposed a learning-based triangulation method for dense multi-view stereo depth. Wong *et al.* [40] use *scaffolding*, based on convex hulls, as part of monocular depth completion. However, we do not use convex hulls but propose our own 3D graph approximation. For 3D object detection, [41] developed a LiDaR to pseudo-LiDaR depth refinement based on graph neural networks. Nonetheless, our algorithm is orthogonal to theirs as 3D hint expansion are not learned.

We leverage existing VIO algorithms. We focus on how to effectively exploit sparse visual hint guidance for deep stereo matching. VIO [3, 34, 36, 27] and SLAM [1, 27, 38, 32] are core for navigation. The CV community has pushed to make these methods fast, robust, and general. Thus, the sparse hints are agnostic to use case and generally their errors are uncorrelated to learned depth methods.

3. Methods

Expanding hints is not trivial since VIO algorithms return hints at image keypoints, *e.g.* edges or corner points. These areas are difficult to expand, because of the geometry around the feature point. While a wrong value could be filtered (as in [16]), it is important to note that it will also not have positive impact. Hints expansion is further motivated by the structure of deep CNNs for stereo matching. Feature representations of the input images have smaller spatial size compared to the image itself, leading to a difficult, and in some case impossible, alignment of the sparse hints with the downsampled image grid. This can be avoided by expanding the hints over a larger area.

In our case, in addition to the stereo image pair, sparse unevenly distributed hints are available as guidance for our model. They are encoded in sparse matrices H_n, each corresponding to a stereo image pair $(I_n^{\mathrm{R}}, I_n^{\mathrm{L}})$. We aim at thoroughly exploiting the rich information regarding the 3D world encoded in the hints by lifting them to 3D points and not simply modelling them as disparity values on a 2D matrix. The 3D hypothesis prompts us to more clearly discriminate hints that are close to each other.

Notation Consider a data set $\mathcal{D} = \{(I_n^{\mathrm{L}}, I_n^{\mathrm{R}}, H_n)\}_{n=1}^{N}$, where N is the number of data samples, $I_n^{\mathrm{L}}, I_n^{\mathrm{R}}$ are stereo image pairs, and H_n are sparse disparity hints. We learn a model with parameters θ to infer a dense disparity map $\mathbf{disp}$ such that $\mathbf{disp}_n = f_\theta(I_n^{\mathrm{L}}, I_n^{\mathrm{R}}, H_n)$. We assume H_n to be a sparse matrix of visual hints, and that an expansion function $\mathbf{densify}(H)$ exists such that the number of hints increases. The function $\mathbf{densify}$ should also lead to more accurate estimation of the disparity $\mathbf{disp}_n = f_{\theta_{\mathbf{densify}}}(I_n^{\mathrm{L}}, I_n^{\mathrm{R}}, \mathbf{densify}(H_n))$.

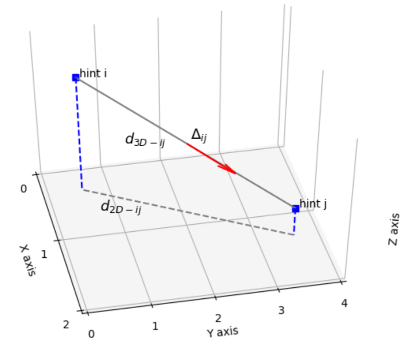

Figure 3: Hints i and j are shown as blue dots in the center of their corresponding pixels, their distance in 3D space and 2D are d_{3D-ij} (gray line) and $d_{2\mathrm{D}-ij}$ (gray dashed line), respectively. The red arrow represents the direction Δ_{ij} used to travel between the hints. Z represents the depth.

3.1. 3D Linear Hints Expansion (Lin 3D)

We propose a linear densification based on two core steps. *(i)* The first, is to connect two hints if they are found on the same horizontal or vertical axis. Hints are connected with a slanted linear interpolation as in [2]. *(ii)* Secondly, the image is split into non-overlapping square patches of fixed size. In this way, the expansion process transforms into a patch-wise line-by-line (first horizontal and then vertical) search of hints that can be connected. Detailed algorithm and implementation is available in the supplementary.

3.2. 3D Graph Hints Expansion (3D $\mathcal{G}$)

To improve over 3D linear hints expansion, we propose to build a 3-dimensional Random Geometric Graph (**RGG**, [9, 8]), where the hints are nodes and edges connect hints that are close in 3D. Connections are further constrained by color similarity between nodes (each node has a corresponding RGB value in I_n^{L}, I_n^{L} and H_n are aligned). Formally, an edge will be created if

$$E_{ij} : (d_{3\mathrm{D}-ij} < R) \wedge (RGB_i \odot RGB_j > \tau), \qquad (1)$$

where E_{ij} is the edge connecting hints i and j, $d_{3\mathrm{D}-ij}$ is their (Euclidean) distance, and R is the maximum distance. $RGB_{i,j}$ are the RGB values in the left image at the location of the hints, $\odot$ is the cosine similarity between the two color vectors and τ is a threshold of similarity ($\tau = 0.9$). Note that $d_{3\mathrm{D}-ij}$ is calculated in 3D coordinates, and we will refer to it also as volumetric distance to help distinguish it from 2D spatial distance in the 2D image plane, denoted as

$d_{2\mathrm{D}-ij}$. The process is visualized in Fig. 3, where disparity is shown along the Z-axis. Once the edges are created, they are *(i)* sorted by volumetric 3D distance and *(ii)* the adjacent ones are discarded. While *(ii)* is trivial, between two spatially (2D) adjacent pixels there are no further pixels to possibly assign an expanded disparity value. The first step *(i)* is grounded on a locally linear approximation of the real disparity, a similar hypotheses to [2], [40]. Therefore, the shorter edges are expanded first as they are more likely to accurately model the real disparity.

The expansion for each edge is performed by travelling one spatial unit distances $d_{\mathbf{1}-2\mathrm{D}}$ at a time between the two hints in the 2D image plane, where their 2D Euclidean distance is denoted $d_{2\mathrm{D}-ij}$. We define E_{ij} be the edge between hints i and j, with distances $(d_{3\mathrm{D}-ij}, d_{2\mathrm{D}-ij})$ and coordinates $\{x_i, y_i, z_i\}$, $\{x_j, y_j, z_j\}$, respectively. They are connected by a slanted 3D line with a slope Δ_{ij} (red arrow in Fig. 3)

$$\Delta_{ij} = \begin{cases} \delta_x = \cos(\arctan((y_i - y_j)/(x_i - x_j))), \\ \delta_y = \sin(\arctan((y_i - y_j)/(x_i - x_j))), \\ \delta_z = (z_i - z_j)/((y_i - y_j)/(x_i - x_j)). \end{cases} \quad (2)$$

Moving by m steps of unit distances $d_{\mathbf{1}-2\mathrm{D}}$ until $d_{\mathbf{1}-2\mathrm{D}}m < d_{2\mathrm{D}-ij}$, the corresponding z disparity value is assigned to the closest hint pixel $\mathbf{h} = \mathcal{H}[\delta_x m, \delta_y m]$ by rounding the computed coordinates to the closest integer. Formally, the loop writes

$$\{H[\mathbf{r}(\delta_x m), \mathbf{r}(\delta_y m)] = \delta_z m \mid m < d_{2\mathrm{D}-ij}, m \in \mathcal{N}\}, \quad (3)$$

where $\mathbf{r}$ is the rounding to the closest integer operation, and m is the set of unit distances. Note that a value is assigned only if the corresponding pixel $\mathbf{h}$ is empty, in order to preserve original hints and values already assigned in Eq. (3). We present the graph expansion algorithm and the implementation in the supplementary material for more clarity.

3.3. DeepPruner

Our aim is to show that expanded VIO hints are effective in enhancing model performance on unseen sequences. Given expanded sparse hints, a model can consider fewer candidate disparities around the suggested values and return more accurate predictions. To demonstrate this, we implemented a novel framework based on DeepPruner [10]. DeepPruner proposed pipeline consists of four parts: *(i)* a PatchMatch-based module first compute matching scores from a small subset of disparities. *(ii)* Then given the estimation from PatchMatch, there is a confidence range prediction network to adjust search range for each pixel. *(iii)* A sparse cost volume will be constructed and aggregated within the predicted range. *(iv)* Finally, an image guided refinement module uses low-level details to further improve the prediction.

Since we have sparse hints H as extra inputs, we can modify the first step and skip the iterative PatchMatch; instead, we use a simpler sampling strategy to compute the sparse matching scores. For a sparse hints matrix H_n

$$\mathbf{range}_n\colon \begin{cases} d_{\mathrm{low}} = (1 - V_n)\, d_{\min} + V_n\, H_n\, (1 - \alpha) \\ d_{\mathrm{high}} = (1 - V_n)\, d_{\max} + V_n\, H_n\, (1 + \alpha) \end{cases} \quad (4)$$

where $d_{\mathrm{low}}, d_{\mathrm{high}}$ are the lower bound and the upper bound of the search range $\mathbf{range}_n$, respectively. $V = (H > 0)$ indicates if the pixels have hints and α is a hyper-parameter to control the relative range. We set $\alpha = 0.2$ to accommodate a small margin of error for our densified visual hints, and $(d_{\min}, d_{\max})$ are the minimum and maximum disparities allowed for the $\mathbf{range}$ during training and testing. $(d_{\min}, d_{\max})$ are used in case no hints is available at a given location. Then we sample disparity candidates uniformly from $[d_{\mathrm{low}}, d_{\mathrm{high}}]$ for each pixel to compute the matching scores. Even if it falls out of the scope of this paper, we believe such guidance approach is straight-forward to apply to several traditional stereo matching methods based on patch matching.

3.4. 3D Expansion for Guided Stereo Matching

Guided Stereo Matching for PSMNet Guided stereo matching (GSM) [26] proposed to guide learning with sparse supervision points, which we refer to as hints in this work. Their method is based on a scaling factor that promotes learning for the pixels of the image where the ground truth is present. Namely, in a model with cost volume, this is formulated as

$$G_n = \left(1 - V_n + V_n\, k\, \exp\left(-\frac{d - H_n}{2c}\right)\right) F_n, \quad (5)$$

where F_n are the original cost volume features, G_n are the enhanced output features, H_n are the known hints, and $V_n = (H_n > 0)$ is a binary mask specifying which pixel will be enhanced. The Gaussian modulation is parameterized by magnitude $k = 10$ and variance $c = 1$.

Confidence-guided Stereo Matching for PSMNet Although guided stereo matching proved to be effective on an uniformly sampled ground-truth, in our setting VIO would be the source of the guiding points. This leads to relevant differences to the previous setting. First, our hints will be localized in areas with keypoint presence, effectively invalidating the assumption of uniformly sampled hints. Second, VIO hints and the expanded hints generated by the proposed expansions may be imprecise. Third, the density is lower than with sparse supervision.

To address the noisy hints, we devise a confidence-based hints filtering for 3D cost volume based architectures as PSMNet [6]. In our scarce supervision scenario, it is particularly important to ensure that hints are positively contributing in the pipeline. The confidence is implemented as

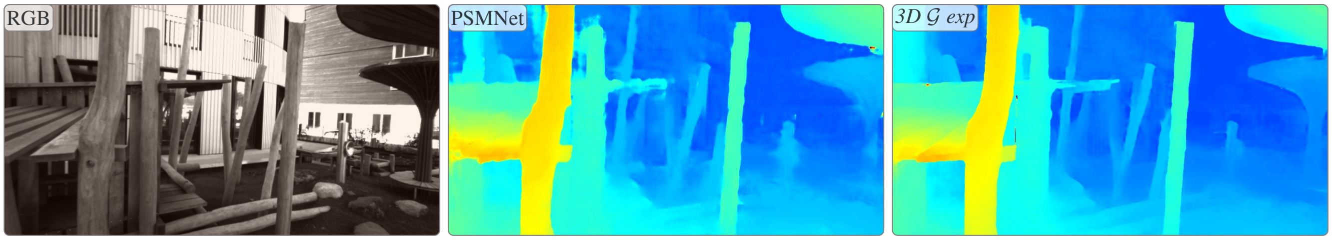

Figure 4: PSMNet qualitative results on ETH3D. Our expansion sharpens details.

DATA SET	IMG SIZE	VGD [DENSITY]	MAE	3D $\mathcal{G}$ EXP [DENSITY]	MAE	Param R	LIN 3D EXP [DENSITY]	MAE	Param W
SCENEFLOW	(256,512)$^\circ$	440 [0.33%]	1.59	8690 [6.6%]	1.89	$R = 8$	8840 [6%]	2.08	$W = (8, 16)$
TARTAN	(480,640)	13 [0.004%]	1.43	849 [0.27%]	1.57	$R = 25$	113 [0.03%]	1.63	$W = (8, 16)$
ETH3D	(544,960)	263 [0.05%]	2.06	3703 [0.7%]	2.16	$R = 8$	4627 [0.88%]	2.34	$W = (8, 16)$
KITTI15	(368,1232)	562 [0.1%]	7.02	40494 [8.9%]	7.47	$R = 20$	5271 [1.1%]	7.77	$W = (8, 16)$

Table 1: Sparsity of the visual guidance hints for each data set. Average hints per image over all the data sets, [density in brackets], MAE hints error, and expansion parameters. 'Img size' is the image size in pixels, '$\circ$' refers to training data. The proposed expansions do not introduce noticeable additional error while greatly improving model prediction performance.

an Euclidean distance between feature vectors. Given the normalized feature maps of the left stereo view f_{L} and the feature maps of the right stereo view f_{R}, we compare the left feature at hint position (x_H, y_H) with the corresponding right feature as suggested by the hint,

$$\mathrm{conf}_{H[x_H,y_H]} = 1 - \tanh(\|f_{\mathrm{L}}(x_H, y_H) - f_{\mathrm{R}}(x_H + H[x_H, y_H], y_H)\|^2), \quad (6)$$

where $H[x_H, y_H]$ is one of our hints. This process is trivially repeated for all hints. Subsequently, the final mask is obtained as follows

$$V = \begin{cases} 1, & \text{where } \mathrm{conf}_H > \tau \wedge H > 0, \\ 0, & \text{where } \mathrm{conf}_H < \tau \vee H > 0, \end{cases} \quad (7)$$

and it is directly applied to the hints for filtering noisy ones ($\tau = 0.9$). A further minor modification to PSMNet is an additional loss term. To mitigate the over-smoothing problem [7], we add a classification loss on the disparity. It is implemented as negative log-likelihood (NLL) loss between the ground truth disparity and the model prediction. Both of them are rounded to the closest integer in order to satisfy the classification objective constraints.

4. Experiments

We evaluate the proposed expansion methods in Sec. 4.1 and employ it as guidance for DeepPruner in Sec. 4.2 and PSMNet in Sec. 4.2. We also deploy it on embedded devices in Sec. 4.4 and compare against guided methods in Sec. 4.3.

Data sets We use SCENEFLOW [21], ETH3D [33], [32], TARTAN [39], KITTI [22], [23] data sets for our experiments. Detailed information about each data set is available in the supplementary material. As in [6], [26], [16] models are evaluated with MAE and threshold error rate. They are computed for every pixel where ground-truth is available, with thresholds $t \in \{2, 3, 4, 5\}$.

Implementation We implement our method with PyTorch. Training is performed on a single NVIDIA A100 GPU. With SCENEFLOW [21] training takes about 25 hours for PSMNet [6] and 125 hours for DeepPruner [10]. We follow training guidelines from the respective papers. We did not notice any training time difference with sparse hints or expanded hints. Testing is performed on the same device and on a NVIDIA Jetson AGX Xavier device. The capability of the method to run on embedded devices is of particular interest for future practical uses.

Model Details *PSMNet* [6] and *DeepPruner* [10] are the baseline models. *Vgd-test* and *Lgd-test* refer to visual guidance and LiDaR guidance at test time applied to the original model. *Vgd* refers to visual hints guided training and testing as introduced in [26]. Note that a crucial difference lies in the hints used, we have sparse visual hints and not the LiDaR hints of [26]. *3D $\mathcal{G}$ exp* and *Lin 3D exp* are the two expansion algorithms proposed in our work. PSMNet models that use expansion also use our confidence filtering.

4.1. Expansion Details

Before looking into actual model performance, we analyze the impact of the proposed expansions. VIO does not provide a uniformly spaced grid of sparse hints as a LiDaR would. Nevertheless, thanks to our densification we are able to achieve a much higher hints density. In Table 1 (upper half) we denote the original VIO hints as *Vgd*, our proposed 3D Graph expansion as *3D $\mathcal{G}$ exp* and linear expansion as *Lin 3D exp*. Generally a $10\times$ increase in hints can be observed when our proposed densification algorithms are ap-

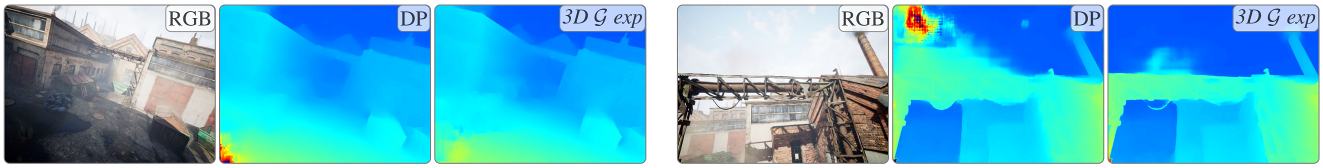

Figure 5: DeepPruner (DP) qualitative results on the TARTAN data set. Above, *3D $\mathcal{G}$ exp* sharpens some building edges, below, removes domain-shift artifacts.

plied. Particularly interesting is the case of the TARTAN data set, where the average hints over all the images is only 13 hints, but after expansion it grows to 113 and 849 hints, for *Lin 3D exp* and *3D $\mathcal{G}$ exp*, respectively. In square brackets we report the density of hints over the total number of pixels in the images. Density better conveys the sparsity or to some extent 'rare' nature of visual hints that our method expands. For example, even after expansion both TARTAN and ETH3D do not pass the 1 % threshold, while SCENEFLOW and KITTI do. In Table 1 (bottom half), we detail expansion parameters. Both *3D $\mathcal{G}$ exp* and *Lin 3D exp* algorithms are implemented as functions that take in input hints of one image and output expanded hints. *3D $\mathcal{G}$ exp* has one parameter, the maximum 3D radius R, which specifies the search range for nodes to connect. *Lin 3D exp* has one parameter, which specifies the size of the patches W to use for potential hints expansion. We assign these two parameters to obtain a good trade-off between final density, proximity of the hints and the chance of even finding a hint due to the low guidance density. *3D $\mathcal{G}$ exp* is applied once, whereas *Lin 3D exp* is applied in two iterations the first with $W_1 = 8$ and the second with $W_2 = 16$.

In Table 1 we report the proposed expansions MAE errors, it is interesting to note that the expansions introduce a minor increase in MAE errors but not significantly higher than the initial VIO hints noise. KITTI15 is the only data sets that stands out with a higher MAE error. This is due to the short sequences that compose the data sets where VIO hits are more challenging to extract and align compared to the longer sequences of the other data sets.

4.2. Ablation Studies

DeepPruner Table 2 demonstrates that expanded guidance is beneficial for DeepPruner [10]. The *Vanilla* model uses a checkpoint and code from the authors official repository. Visual guidance *Vgd* improves on KITTI15 only, meaning our guided PatchMatch needs denser hints (KITTI15 has higher density among the three data sets). In fact, expanded guidance improves over all the test data sets, proving that our intuition is correct. On KITTI15 all error rate thresholds improve by roughly a factor of 4, while MAE decreases from 3.47 to 1.46 and 1.44 for *Lin 3D exp* and *3D $\mathcal{G}$ exp*, respectively. On ETH3D, *3D $\mathcal{G}$ exp* lowers MAE from 0.87 to 0.54 (38% decrease). Also error rates are all improved marginally relative to *Lin 3D exp*. Boost on

	EXPANSION	MAE	>2	>3	>4	>5
ETH3D	Vanilla ◁	0.87	5.11	3.63	2.9	2.39
	Vgd	1.01	5.41	3.82	3.01	2.52
	Lin 3D exp	0.56	3.88	2.54	1.93	1.54
	3D $\mathcal{G}$ exp	**0.54**	**3.76**	**2.43**	**1.85**	**1.47**
TARTAN	Vanilla ◁	5.63	20.63	16.70	14.50	12.98
	Vgd	8.65	20.58	17.33	15.46	14.15
	Lin 3D exp	2.18	14.78	11.36	9.47	8.15
	3D $\mathcal{G}$ exp	**2.17**	**14.77**	**11.34**	**9.45**	**8.13**
KITTI15	Vanilla ◁	3.47	34.76	23.59	17.76	14.10
	Vgd	1.94	10.98	6.37	4.68	3.79
	Lin 3D exp	1.46	9.46	5.27	3.78	3.00
	3D $\mathcal{G}$ exp	**1.44**	**9.30**	**5.01**	**3.51**	**2.75**

Table 2: Ablation study of our guidance on DeepPruner. Expanded VIO guidance improves accuracy on unseen data. *Vgd* is a sparse visual hints guided model, *3D $\mathcal{G}$ exp* is the 3D graph expansion (Sec. 3.2) and *Lin 3D exp* the linear 3D expansion (Sec. 3.1). ◁ Original DeepPruner-best SCENEFLOW model.

TARTAN is also competitive with a 60% lower MAE error. We could not observe a significant difference between *Lin 3D exp* and *3D $\mathcal{G}$ exp* densification, possibly due to the very sparse visual hints for this data set (see Table 1). Qualitative results are shown in Fig. 5, which highlights that expanded hints are able to produce sharper predictions and remove inference artifacts (more examples in the supplement).

PSMNet We test on TARTAN, ETH3D and KITTI. Results are illustrated in Table 3. Initial results confirm the findings of [26], there is a minor improvement on both ETH3D and TARTAN when using guidance *Vgd-test* at test time, and a clear improvement across all metrics with *Vgd* at training time as well. This is a clue that our sparse visual hints can positively contribute to improve generalization performance. For ETH3D, *Vgd* MAE is 1.54, with *Vgd + Lin 3D exp* MAE lowers to 1.19 (23% lower), even better with *Vgd + 3D $\mathcal{G}$ exp* down to 1.04 (30% lower). Qualitative results on challenging ETH3D scenes in Fig. 4 confirm the benefits of using the proposed guidance for sharper and more accurate disparity predictions. More examples are available in the supplementary. TARTAN is peculiar to have very sparse visual hints, which makes it a very challenging test-bed for our method. This is highlighted by the minor metric changes for *Vgd-test*. *Vgd* obtains a major improvement of 38% in MAE, increasing to 50% and 54% for *Vgd + Lin 3D exp* and *Vgd + 3D $\mathcal{G}$ exp*, respectively.

Model	ETH3D MAE	ETH3D >2	ETH3D >3	ETH3D >4	ETH3D >5	TartanAIR MAE	TartanAIR >2	TartanAIR >3	TartanAIR >4	TartanAIR >5
PSMNet [6]	5.25	16.99	7.62	5.82	5.10	5.51	21.30	14.09	11.34	9.81
PSMNet [6] *Vgd-test*	5.25	16.78	7.65	5.80	5.10	5.51	21.29	14.09	11.30	9.80
Vgd	1.54	8.59	5.47	4.17	3.41	5.53	21.03	16.62	14.14	12.45
Vgd + Lin 3D exp	1.19	8.8	5.16	3.63	2.77	2.74	17.86	13.29	10.81	9.24
Vgd + 3D $\mathcal{G}$ exp	**1.04**	**7.91**	**4.41**	**3.01**	**2.27**	**2.54**	**17.00**	**12.72**	**10.48**	**9.04**

Table 3: Ablation study of our proposed methods on PSMNet [6]. *Vgd* is a sparse visual hints guided model, *3D $\mathcal{G}$ exp* is the 3D graph expansion (Sec. 3.2) and *Lin 3D exp* the linear 3D expansion (Sec. 3.1).

KITTI We test on the 2011_09_26_0011 KITTI Velodyne sequence. Table 4 presents on the upper half reference results of PSMNet and finetuned PSMNet. The bottom half is split in LiDaR guided (Lgd) and VIO guided (Vgd) stereo matching. LiDaR expansion improves over [26] for MAE error while paying a minor loss in <2%. This is a notable achievement as the performance of [26] are strong and shows expansion can benefit LiDaR guided stereo matching. Notably, VIO expansion as well achieves a slight improvement in MAE error. To summarize, hints guide cost volume construction, leading to better overall performace (MAE) but expansion in this experiment is not yet able to improve accuracy (<2%) likely due to expansion noise.

	<2% All	<2% NoG	MAE All	MAE NoG	<2% All	<2% NoG	MAE All	MAE NoG
PSMNet [6]	38.60	38.86	2.36	2.37	–	–	–	–
PSMNet-ft	1.71	1.73	0.72	0.73	–	–	–	–
	LiDaR guidance (Lgd)				VIO guidance (Vgd)			
GD	**0.67**	**0.67**	0.47	0.47	**1.71**	**1.73**	0.72	0.73
Lin 3D exp	0.79	0.77	0.44	0.45	2.08	2.08	0.72	0.72
3D $\mathcal{G}$ exp	0.73	0.82	**0.41**	**0.44**	1.90	1.91	**0.70**	**0.71**

Table 4: Ablation study of our expansions on KITTI Velodyne. Top half, reference PSMNet results. Bottom half, expanded LiDaR and VIO guidance reduce MAE error for guided stereo matching, proving expansion is effective not only on VIO but also on LiDaR.

In addition to KITTI Velodyne, we evaluate the performance on a model pre-trained on SceneFlow without any guidance, and test it on KITTI15. We report numbers in Table 5. Starting from the publicly available SceneFlow pre-trained model from PSMNet authors, we obtain MAE 4.24. With *PSMNet Vgd-test* guidance, the performance does not change except for a minor improvement on the error rates. Hints expansion is more effective, *Lin 3D exp* leads to a MAE improvement of 0.5% and a decrease on the >2 error rate, meaning that the model is more precise in some areas with small errors. *3D $\mathcal{G}$ exp* is even more effective with a 2.5% MAE decrease but shows mixed results on the errors rates. A possible reason, as happened in KITTI Velodyne, is that guided areas improve accuracy (thus lower MAE) but unguided areas do not. We speculate the reason is the feature modulation in the guided cost volume. Finally, we include GSM [26] as a sensor-based reference which gives more competitive results. Again our expansions improve the Lgd baseline by around 3% MAE and similarly all the error rates. Yet, we emphasize that our starting hint density is much lower (0.1% vs. their 5%). To summarize, in Table 5 performance gain is limited because the model was not trained to exploit guidance. However, guidance positively contributes out-of-the-box to improve accuracy.

Guidance	Vgd	Lgd	MAE	>2	>3	>4	>5
PSMNet[6]$^{\triangleleft,\oplus}$			4.24	46.54	29.61	21.26	16.41
GSM Lgd-test[26]		✓	3.90	33.38	23.12	17.59	14.01
GSM Lgd-test Lin 3D exp$^{\triangleleft}$		✓	3.82	32.83	22.45	17.09	13.19
GSM Lgd-test 3D $\mathcal{G}$ exp$^{\triangleleft}$		✓	**3.79**	**32.34**	**22.08**	**17.01**	**13.03**
PSMNet Vgd-test$^{\triangleleft}$	✓		4.24	46.45	**29.57**	**21.24**	**16.40**
Lin 3D exp$^{\triangleleft}$	✓		4.21	**46.28**	29.62	21.29	16.44
3D $\mathcal{G}$ exp$^{\triangleleft}$	✓		**4.13**	47.13	30.65	22.04	16.98

Table 5: Pre-trained PSMNet agnostic to guidance, tested on KITTI15 with hints guidance. The model was not trained to exploit guidance. However, results confirm expansion is effective on both LiDaR (*Lgd*) and VIO (*Vgd*) hints guidance out-of-the-box. $\triangleleft$ authors original checkpoint and $\oplus$ code.

4.3. Comparison with Guided Stereo Methods

We compare with state-of-the-art guided stereo matching methods [26, 16]. In the upper half of Table 6, models are trained on SceneFlow and tested on KITTI15 (as in Table 5). Our improvement over PSMNet is particularly evident in average error and for pixels with small error rate (threshold >2). The *Lin 3D exp* model suffers a minor accuracy drop on the higher error rates (thresholds >4, >5) while improving 18% on MAE. *3D $\mathcal{G}$ exp* gains

Model	Lgd	MAE	>2	>3	>4	>5
PSMNet[6]		4.24	46.54	29.61	21.26	16.41
GSM[26]	✓	**1.39**	**12.31**	**3.89**	**2.23**	**1.60**
Lin 3D exp		3.44	41.63	29.23	22.19	17.66
3D $\mathcal{G}$ exp		*3.21*	*38.27*	*26.78*	*20.35*	*16.23*
GSM-ft[26]	✓	0.763	2.73	1.82	1.51	1.33
S^3-ft[16]	✓	**0.443**	**1.65**	**0.96**	**0.71**	**0.57**
Lin 3D exp-ft		*0.95*	*6.27*	3.29	*2.35*	1.87
3D $\mathcal{G}$ exp-ft		0.98	6.28	*3.25*	*2.35*	*1.89*

Table 6: Comparison with guided stereo methods. Training on Sceneflow and testing on KITTI15 (upper half), and finetuned on KITTI12 (lower half).

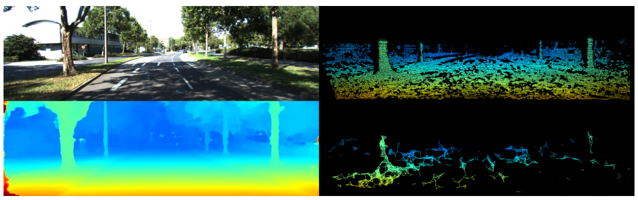

Figure 6: Jetson board results on KITTI running at ∼1 fps, 8× faster than the standard model, with further engineering it could work real-time. RGB (top left), LiDaR GT (top right), predicted disparity (bottom left) and *3D $\mathcal{G}$ exp* visual hints (bottom right).

on all the metrics, and *GSM* [26] obtains the best absolute performance. It is worth noting they exploit LiDaR guidance, which as already discussed is evenly distributed and accurate, leading to a clear performance advantage. If additional sensors are available, a LiDaR can be effective. Models finetuned (suffix '-ft') on KITTI12 are in the bottom half of Table 6. All models obtain clear gains thanks to reduced domain-shift. However, in this case *S^3-ft*[16] improved upon *GSM-ft* [26] to obtain state-of-the-art results. Finetuning reduces the gap of our methods with *GSM-ft*. Before finetuning the average error of *GSM* is 57% lower and after it is 23% lower.

4.4. Embedded Devices

To demonstrate our method on an embedded device, we perform inference on an NVIDIA Jetson AGX Xavier device with KITTI Velodyne sequence 2011_09_26_0011 at high resolution (384×1280 pixels) using PSMNet[1]. Our *3D $\mathcal{G}$ exp* has an inference time of 8.607 seconds per sample, while the MAE error on the full sequence is 0.70. Furthermore, we optimized the model to use half-precision (fp16) and target the aarch64 platform using NVIDIA TensorRT inference optimizer via TRTorch compiler. This resulted in an inference time of 1.062 seconds per sample (a speed-up factor of 8× relative to the unoptimized model), while maintaining the original accuracy MAE = 0.70. Fig. 6 illustrates a qualitative example of inference. An interesting observation is that the densification of visual hints on vertical structures like trees works particularly well. The predicted disparity is accurate except a typical stereo artifact on the right.

[1]We were unable to deploy DeepPruner on the device yet, as some of the operations in the model could not be optimized/converted by TRTorch

Guidance	Time (sec/sample)	MAE	DL+VIO	exp	CNN
Lin 3D exp	**0.9708**	0.72	13%	1.5%	88.5%
3D $\mathcal{G}$ exp	1.062	**0.70**	12%	10%	85.5%

Table 7: Inference on an NVIDIA Jetson AGX Xavier and KITTI Velodyne (384×1280 pixels). Breakdown of time (in %) employed for Data Loading and VIO extraction (DL+VIO), expansion (exp) and inference in CNN (CNN).

We profiled our code (after applying NVIDIA TensorRT optimization Table 7) and the execution time is divided as follows: data loading and VIO hints extraction 12%, *3D $\mathcal{G}$* expansion 10%, and deep CNN inference 78%. Most of the cost comes from the deep CNN employed. Expansion on Jetson comes at a time cost similar to data loading, but improves MAE from 2.36 to 0.70 as reported in Table 4. In the case of *Lin 3D* expansion, the execution time is slightly faster because the expansion is more lightweight. Resulting in an inference time of 0.9708 seconds per sample and $MAE = 0.72$. The detailed execution time breakdown is 13% for data loading, 1.5% for *Lin 3D* expansion, and 85.5% for deep CNN inference. To summarize, *Lin 3D* is 8.5% faster, while *3D $\mathcal{G}$* is 4% more accurate.

5. Conclusion

This work tackled a challenging scenario for stereo-matching methods. Without any additional sensors, improving their performance on unseen sequences (i.e. from different data distributions). To this end, we demonstrated the utility of VIO hints guidance for deep stereo matching. In particular, 3D visual hints expansion seamlessly works for existing pre-trained models and guidance-aware models, and across different architectures. Our technique does not need additional sensors and exploits well studied and robust odometry techniques. Nevertheless, we show that also LiDaR benefits from our expansion. Extensive experiments on distinct and challenging data sets support our findings, and practical use was also investigated by successfully deploying our algorithm on embedded devices.

Acknowledgements. We acknowledge funding from Academy of Finland (No. 339730, 324345), and the Finnish Center for Artificial Intelligence (FCAI). We acknowledge the computational resources by the Aalto Science-IT project and CSC – IT Center for Science, Finland.

References

[1] Jinqiang Bai, Junqiang Gao, Yimin Lin, Zhaoxiang Liu, Shiguo Lian, and Dijun Liu. A novel feedback mechanism-based stereo visual-inertial slam. *IEEE Access*, 7:147721–147731, 2019.

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 11, pages 1–11, 2011.

[3] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304. IEEE, 2015.

[4] Robert Bolles, Harlyn Baker, and M Hannah. The jisct stereo evaluation. In *DARPA Image Understanding Worshop*, 1993.

[5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 23(11):1222–1239, 2001.

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018.

[7] Chuangrong Chen, Xiaozhi Chen, and Hui Cheng. On the over-smoothing problem of cnn based disparity estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[8] Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

[9] Jesper Dall and Michael Christensen. Random geometric graphs. *Physical Review E*, 66(1):016121, 2002.

[10] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4384–4393, 2019.

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[12] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–756. Springer, 2016.

[13] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017.

[14] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3273–3282, 2019.

[15] Marsha Jo Hannah. Computer matching of areas in stereo images. In *PhD Thesis, Stanford University*, 1974.

[16] Yu-Kai Huang, Yueh-Cheng Liu, Tsung-Han Wu, Hung-Ting Su, Yu-Cheng Chang, Tsung-Lin Tsou, Yu-An Wang, and Winston H Hsu. S3: Learnable sparse signal superdensity for guided depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16706–16716, 2021.

[17] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 66–75, 2017.

[18] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 573–590, 2018.

[19] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2811–2820, 2018.

[20] Jose Marroquin, Sanjoy Mitter, and Tomaso Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82(397):76–89, 1987.

[21] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[22] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:427, 2015.

[23] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018.

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

[25] Andrea Pilzer, Stéphane Lathuilière, Dan Xu, Mihai Marian Puscas, Elisa Ricci, and Nicu Sebe. Progressive fusion for unsupervised binocular depth estimation using cycled networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 42(10):2380–2395, 2019.

[26] Matteo Poggi, Davide Pallotti, Fabio Tosi, and Stefano Mattoccia. Guided stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 979–988, 2019.

[27] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-

semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020.

[28] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, pages 31–42. Springer, 2014.

[29] Daniel Scharstein and Richard Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision (IJCV)*, 28(2):155–174, 1998.

[30] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[31] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–518. Springer, 2016.

[32] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019.

[33] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3260–3269, 2017.

[34] Otto Seiskari, Pekka Rantalankila, Juho Kannala, Jerry Ylilammi, Esa Rahtu, and Arno Solin. Hybvio: Pushing the limits of real-time visual-inertial odometry. *ArXiv Preprint*, 2021.

[35] Ayan Sinha, Zak Murez, James Bartolozzi, Vijay Badrinarayanan, and Andrew Rabinovich. Deltas: Depth estimation by learning triangulation and densification of sparse points. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 104–121. Springer, 2020.

[36] Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael Watterson, Sikang Liu, Yash Mulgaonkar, Camillo J Taylor, and Vijay Kumar. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters (RAL)*, 3(2):965–972, 2018.

[37] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14362–14372, 2021.

[38] Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jörg Stückler, and Daniel Cremers. Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics and Automation Letters (RAL)*, 5(2):422–429, 2019.

[39] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.

[40] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters (RAL)*, 5(2):1899–1906, 2020.

[41] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[42] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–158. Springer, 1994.