Aerial Single-View Depth Completion with Image-Guided Uncertainty Estimation

Originally published in:
IEEE Robotics and Automation Letters 5(2), https://doi.org/10.1109/LRA.2020.2967296

Funding acknowledgement:
157585 - Collaborative vision-based perception for teams of (aerial) robots (SNF)

Aerial Single-View Depth Completion with Image-Guided Uncertainty Estimation

Lucas Teixeira[1], Martin R. Oswald[2], Marc Pollefeys[2], and Margarita Chli[1]

***Abstract*—On the pursuit of autonomous flying robots, the scientific community has been developing onboard real-time algorithms for localisation, mapping and planning. Despite recent progress, the available solutions still lack accuracy and robustness in many aspects. While mapping for autonomous cars had a substantive boost using deep-learning techniques to enhance LIDAR measurements using image-based depth completion, the large viewpoint variations experienced by aerial vehicles are still posing major challenges for learning-based mapping approaches. In this paper, we propose a depth completion and uncertainty estimation approach that better handles the challenges of aerial platforms, such as large viewpoint and depth variations, and limited computing resources. The core of our method is a novel compact network that performs both depth completion and confidence estimation using an image-guided approach. Real-time performance onboard a GPU suitable for small flying robots is achieved by sharing deep features between both tasks. Experiments demonstrate that our network outperforms the state-of-the-art in depth completion and uncertainty estimation for single-view methods on mobile GPUs. We further present a new photorealistic aerial depth completion dataset that exhibits more challenging depth completion scenarios than the established indoor and car driving datasets. The dataset includes an open-source, visual-inertial UAV simulator for photo-realistic data generation. Our results show that our network trained on this dataset can be directly deployed on real-world outdoor aerial public datasets without fine-tuning or style transfer.**

***Index Terms*—Aerial Systems: Perception and Autonomy; Deep Learning in Robotics and Automation**

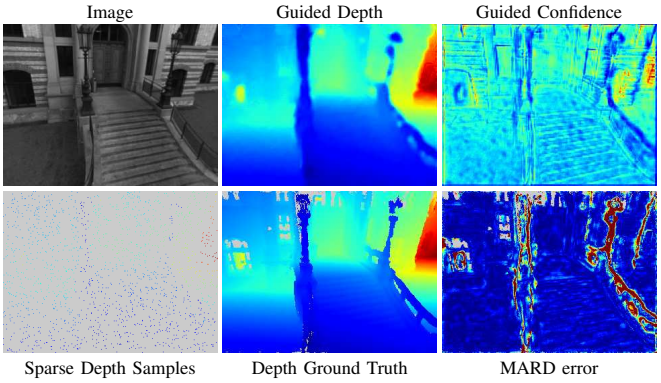

Fig. 1: Qualitative results of the proposed network trained on our new aerial dataset and tested on the public ETHZ CAB Building Aerial Dataset [3]. The input to our approach is an image and a sparse depth map. The output is the guided depth and guided confidence. This confidence indicates the probability of each pixel in the guided-depth map to be valid. The jet colour map is used in all colour-coded maps in this letter (low high; grey means unknown depth). Our results show that the confidence and the error have a high inverse correlation. As a result, the guided-depth-map pixels with low confidence can be eliminated.

I. Introduction

Spatial awareness is a crucial capability for autonomous mobile robots. The ability of a mobile robot to sense its surroundings to gain enough understanding of the environment is of fundamental importance for performing realistic autonomous tasks, such as visually inspecting a building while avoiding collisions. To this end, several approaches for Simultaneous Localisation And Mapping (SLAM) have been proposed in the literature, as robotic ego-motion estimation and map building are core competencies necessary for spatial understanding. While state-of-the-art SLAM systems provide localisation estimates that are accurate enough for controlling the motion of a wide range of robots, the sparsity of traditional SLAM maps is usually a problem for path planning and collision avoidance [1]. Although several SLAM approaches, such as infiniTAM [2], are capable of building denser maps, these methods are usually not scalable to large scenes, restricting their applicability in real-world scenarios. An alternative method is to use depth completion to build a dense 3D map of the robots workspace out of a sparse 3D map provided by either SLAM or LIDAR sensing. In this case, a depth image is obtained by projecting the sparse 3D map in the image plane of the camera. Visual SLAM sparse maps usually cover about 0.5% of the image pixels and LIDAR maps around 10%. The traditional depth-completion process estimates the depth of all other pixels in the image using both sparse depth estimates and colour image captured by a camera.

In the last years, depth completion approaches have become popular in the context of ground robots, especially in autonomous driving and indoor applications. Currently, the state of the art in depth completion makes use of Convolutional Neural Networks (CNNs). As CNNs are not invariant to rotation or scale changes, several approaches augment the training process by rotating and re-scaling the available training images. While this technique can simulate the small viewpoint variations experienced by ground vehicles, the same is not valid for aerial robots, which undergo far more dramatic

Manuscript received: September, 10, 2019; Revised December, 4, 2019; Accepted December, 28, 2019.

This paper was recommended for publication by Editor Jonathan Roberts upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Swiss National Science Foundation (SNSF, Agreement no. PP00P2_157585) and NCCR Robotics. This research was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00280. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

The authors are with the Vision for Robotics Lab[1] ({pilucas, chlim}@ethz.ch) and the Computer Vision and Geometry Group[2] ({moswald, pomarc}@inf.ethz.ch) at ETH Zurich, Switzerland. Marc Pollefeys is also with Microsoft Research.

This letter has supplementary downloadable material (code and datasets) available at www.v4rl.ethz.ch/research/datasets-code.html and visual results available at https://youtu.be/IzfFNlYCFHM .

Digital Object Identifier (DOI): see top of this page.

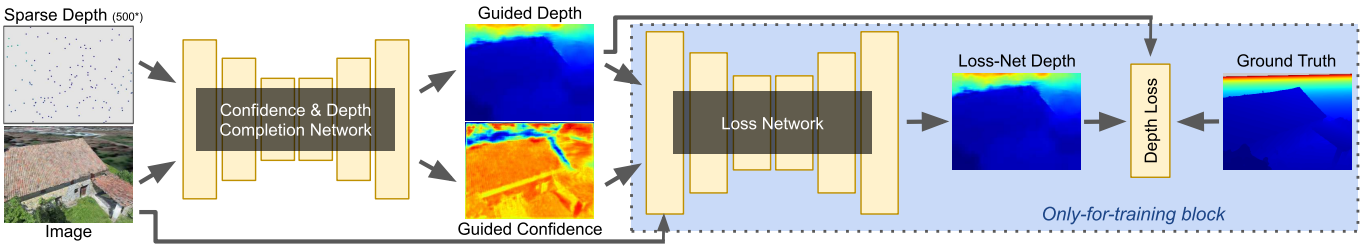

Fig. 2: Overview of our confidence training framework. Given a sparse depth map and a corresponding colour image, a depth completion network estimates a dense depth map and confidences which, later, are used to filter the estimated depth map. Grey input pixels refer to missing depth information. The sparse samples are enlarged for better visualisation.

viewpoint changes.

As datasets exhibiting both viewpoint variations and depth information of a large number of different places are very hard to obtain, current CNN-based approaches for depth completion usually rely on datasets recorded from the ground, such as KITTI dataset [4]. However, networks trained using ground images do not perform well in predicting depth from aerial images, as the self-similarity of typical road scenes builds a bias in the network. On the other hand, generic aerial footage captured from an Unmanned Aerial Vehicle (UAV) lacks ground-truth depth information for supervised training and computing it using photogrammetry [5] is very time-consuming and usually noisy. While the KITTI dataset makes use of a high-resolution large-baseline stereo camera to build ground-truth for depth, such a sensor suite is usually too wide to be carried by small aircraft.

Inspired by the challenges of depth completion for aerial images, here, we publish a new photo-realistic synthetic dataset and open-source the simulator used in this paper. With this dataset in hand, we were able to design a novel network that can perform depth completion while computing the uncertainty of the estimated depths. The uncertainty map, also called confidences, can be used to filter out unreliable depth estimates to obtain a more accurate result for the proposed depth completion approach. An overview of our approach is shown in Figure 2.

In summary, this work makes the following **contributions:**

- A novel compact network for depth and confidence estimation with real-time performance onboard a lightweight GPU suitable for small UAVs. When using the estimated confidence for filtering out erroneous depth values, our network outperforms the state of the art in single-view depth completion for small UAVs with relatively small compromise in density.
- A novel end-to-end confidence training framework. The results show that our framework can train a confidence estimation network better suited for depth filtering than the state-of-the-art methods.
- A new, publicly available, photo-realistic, large, visual-inertial dataset exhibiting a wide range of viewpoints of a UAV with depth and pose ground-truth information per image. Our experiments show that the proposed network trained using our synthetic dataset together with NYUv2 indoor dataset [6] can be successfully compute depth completion on real-world public aerial datasets.
- An open-source, visual-inertial simulator that allows the creation of photo-realistic datasets from a UAV.

II. Related Work

Mapping, depth estimation and completion from cameras have been studied for several years. Here, we focus on discussing the most related works.

Real-time aerial depth estimation and mapping: Depth estimation is well studied in the literature, especially in the context of autonomous driving. However, as aforementioned, few works focus on real-time aerial depth estimation. For many years, real-time aerial mapping was a task for an off-board computer connected to the aircraft via wireless communication. In [7] a system is proposed, where a UAV running visual-inertial SLAM onboard as odometry, transmits its poses and images to a ground station, for further optimisation (aka bundle-adjustment) and denser mapping computation. Later, Weiss et al. [8] show that it is possible to achieve better accuracy using visual-inertial SLAM systems, but the poses and landmark-map produced by SLAM still are not very accurate. However, today, there is a large range of visual-inertial SLAM methods [9] that already facilitates more exciting applications. Some of these modern SLAM systems, e.g. OKVIS [10], produce poses and landmarks that are good enough to be used in 3D reconstruction without further optimisation. The poses are not as good as needed for pure Structure from Motion [11], but in [3], it is shown that the landmarks produced by OKVIS can be filtered and meshed to build a rough representation of scenes with simple geometry. Other scene structures, such as planes, are also commonly extracted from landmarks to better understand and map the environment [12], [13]. The main problem with these methods is the strong assumptions that they make about the environment, limiting their application.

Machine learning for depth estimation: Another way to enhance sparse depth estimations is by using CNN-based depth completion. Despite being very popular in autonomous cars, it is not available for aerial robots due to lack of training data and more challenging scenarios. Neural Networks have already been demonstrated to make reasonable predictions of scene depth from a single colour image only. Even though primitive networks try to blindly learn using generic convolution blocks [14], today, the state-of-the-art uses a composition of building blocks known to perform well in specific tasks, such as in [15]. Such large networks, however, are too slow and memory-demanding for resource-constrained platforms.

As every machine learning problem, training data for supervision is a significant problem in outdoor tasks. Several current works propose self-supervised methods to avoid this problem. Some try to perform motion stereo with camera poses also calculated by the network [16], while others use stereo camera

datasets [17]. Similar to our method, Li and Snavely [18] use 3D models for learning depth prediction. Although they use crowd-sourced images from famous landmarks, such as the Eiffel Tower and the Colosseum, this dataset only provides satisfactory reconstruction accuracy close to the ground, from where the pictures were taken, which is insufficient for a more generic setting. In addition, the approach in [18] is tailored to scale-free 3D models while robotic navigation requires the estimation of metric maps. Instead, in our work, the 3D models are complete, and the datasets are metric with aligned gravity to enable aerial visual-inertial SLAM with this data.

Depth completion algorithms are more useful for aerial depth estimation because they perform better than pure depth prediction, which uses only the colour image. Since completion relies on sparse depth information available from the visual SLAM or another sensor, such as a portable LIDAR, the problem is more straightforward and can be addressed using smaller networks.

One of the top-performing depth completion algorithms was proposed by Ma and Karaman [19] and it makes use of residual networks. Later, the same authors [20] proposed an even more accurate U-shaped residual network as an improvement while also presenting a self-supervised approach.

Weerasekera et al. [21] also present a very competitive approach using conditional random fields and convolution networks, albeit it is too slow for aerial navigation. The same applies to Zhang and Funkhouser [22]: instead of using a CNN to estimate depth directly, they predict geometric features such as normals and object boundaries and fuse them within an optimisation step. Another interesting direction is presented by Chen et al. [23], in which they perform a geometric depth extrapolation in the sparse data before inputting it to the network. Therefore, the network task changes to refining the initial extrapolation.

Lastly, Eldesokey et al. [24] present an algorithm with one of the best performances to date. They proposed a network with two intertwined paths that explicitly propagates the confidences through the network. In this way, the network can know which pixels had depth information in the input and which ones were estimated by the network.

Confidence-aware deep learning: Uncertainty prediction [25] for convolution neural networks is especially salient in the context of Bayesian deep learning using dropout sampling [26], [27] or ensemble techniques [28]. The true statistical uncertainty is a measure of confidence and can be supervised using one of the previous methods. However, it is a much more complex task to be learned in the context of an already difficult scenario of aerial imagery. Another way to compute depth with associated confidence is presented by Liu et al. [29], but they use multiple views of a video to compute the depth. The confidence is the uncertainty of this multi-view process. Similar confidence estimation is not possible to be computed by our method given that it is a single-view approach.

In addition, several methods use a confidence-aware loss function such as in [24], [30], that combine confidence and depth error in the same function. The main problem with this type of method, based on a confidence-weighted sum of the depth errors, is that zero confidence in every pixel is the best way to minimise the loss. As a result, these methods require complex manual tuning of the multiple loss functions in order to prevent the zero confidence behaviour.

Some large networks, e.g. [31], [32], produce intermediary results with some correlation with a confidence measurement, but these results are not meant to be used as output. In this work, the confidence is interpreted as a classification problem in which the confidence indicates the probability of each pixel to have a correct depth estimation. In addition, our training framework can be trained end-to-end and does not require tuning.

Fig. 3: Aerial-scanned 3D models used for creating our novel dataset with around 84K images. The last column also depicts an extract from a manually-piloted real drone used to build some of the trajectories (shown in green), as explained in Section III-A.

III. Our Approach

This section presents our approach for depth completion and confidence estimation, details our confidence training framework, and the new aerial RGB-D dataset.

A. Aerial Depth Dataset

Inspired by the lack of aerial datasets with sufficient viewpoint variations and different scenes, we provide a new aerial dataset to enable training of neural networks with more realistic depth supervision. The dataset uses 18 3D-reconstructed models built using photogrammetry software. Figure 3 shows some examples of 3D models used in this work. With this set of 3D models, we created 26 independent camera trajectories with no visual overlap, which were used to render 83797 RGB and depth images separated, 19 trajectories for training and 7 for validation. This totals in 67435 training images and 16362 validation images in the dataset. Furthermore, the dataset includes inertial data and the result of the visual-inertial SLAM OKVIS [10], i.e. poses along of the trajectory and landmarks seen from these poses.

We used two types of trajectories: *(i)* trajectories extracted from real-drone flights in which the camera positions were computed using photogrammetry [5]. These trajectories do not have inertial data because we used an off-the-shelf UAV without access to the inertial sensor, but they reproduce precisely the dynamics of a real UAV; *(ii)* trajectories generated using sparse waypoints and executed by a carrot-following-like path-planning algorithm [33]. We used three popular types of waypoint trajectories: lawnmower pattern, circular (looking

at the centre), and manually selected waypoints for complex trajectories. The camera is mounted in front of the UAV with one degree of freedom, pitch. The trajectories use a pitch looking at $0°$(forward), $30°$, $45°$, or $90°$(downward). Upon the acceptance of this article, the dataset and the simulator developed to its creation will be made publicly available together with the source-code of our approach. Our simulator is built on top of the RotorS Framework [34] for UAV dynamics, ROS Gazebo for Physics, and Blender Render engine for rendering.

B. Confidence and Depth Completion Network

While designing an aerial depth-completion network, the main constraint for small UAVs is the limited onboard GPU. This constraint rules out most of the recent state-of-the-art network designs. Instead, we designed a compact network that can run in real-time and fits to small GPUs, such as the NVIDIA Jetson TX2. As a drawback, our network is thus not capable of outperforming larger and slower networks that are in the top of the KITTI Depth Completion challenge, e.g. [4].

However, aiming for its deployment in robotic platforms with limited payload and computational capabilities, the proposed network outputs not only the depth estimates but also their confidence values that are used to filter out the unreliable depths.

Inspired by Eldesokey et al. [24], Figure 4 shows our novel network. In the first stage, a combination of normalised convolutions is used together with confidence-aware max-pooling and up-sampling. This first part ignores the colour image and computes the unguided depth and the unguided confidence based only in the sparse depth and the mask created by the step function. As shown in [24], the unguided confidence is very similar to the geometrically-computed distance transform in [23]. Following the first stage, both the unguided depth and the unguided confidence computed are fed together with the colour image into a UNet-like encoder-decoder architecture that computes the guided depth.

Different from previous works, we also propose a guided-confidence estimation using as input the guided depth and deep features collected across the network. Our guided-confidence estimation does not compute a statistical uncertainty, instead, it computes the probability of a point in the estimated depth map being valid or not. This is done using our classification network, *conf-net*.

Our *conf-net* can observe, for example, that a pixel coordinate is over an edge in the colour image and also in an area of rapid depth variation in the guided depth. Then, the classifier could potentially infer that the depth in this coordinate is probably poorly estimated because it is in an area where the interpolation of neighbours is prone to error. In fact, this behaviour is observed in the results.

We tried several combinations of shared features and classification networks for the *conf-net*. We observed that deeper classification networks did not impact the accuracy of the final result. The shared feature selection was a very time-consuming decision given that some options severely impact the depth estimation accuracy while others impact the performance. We selected this design because it has the best accuracy with real-time performance.

C. Confidence Training Framework

Our training framework has two parts, a loss network and a depth loss. Our loss network is inspired by methods that compute multiple depths using different approaches in the same network and then combine the depth results using a weighted sum guided by the also estimated relative-confidence or attention maps, e.g. [31], [32]. A high confidence area in one of the maps means that the correspondent approach is likely to compute a better depth estimation inside this area than the other approaches of the network. Similarly, given that our network is interested in confidences for filtering out the defectively estimated depth points, the confidence should be comparatively higher in points of the depth map with well-estimated values. In order to achieve this type of confidence, we opted for using normalized convolutions [35], as their formulation is also a weighted sum guided by the confidence, as shown in Equation 1. However, this equation is the sum of neighbouring pixels in the convolution instead of multiple depth maps like in [31], [32].

$$\mathcal{Z}^{out}_{i,j} = \frac{\sum_{m,n} \mathcal{Z}^{in}_{i+m,j+n} \mathcal{C}^{in}_{i+m,j+n} \Gamma(\mathcal{W}_{m,n})}{\sum_{m,n} \mathcal{C}^{in}_{i+m,j+n} \Gamma(\mathcal{W}_{m,n}) + \epsilon} + b \qquad (1)$$

Assuming that neighbours in the depth map are locally similar, the normalized convolution presented in Equation 1 will have a more accurate depth estimation, in general, when the confidences of the better-estimated depths in the neighbourhood of a pixel are higher than the confidence of the bad estimated ones. In our network, we used the SoftPlus function as Γ. The kernel weights, $\mathcal{W}$, and the bias, b, are learnable parameters. ϵ is a small number to avoid division by zero, $\mathcal{Z}^{out}$ is depth estimation, and $\mathcal{Z}^{in}$ is the depth input of the convolution and $\mathcal{C}^{in}$.

In fact, we use as loss network another instance of the same network used for depth completion without the *conf-net* and the step function. The loss-network's input is guided depth, guided confidence and the colour image. This network has a multi-scale sequence of normalised convolutions in addition to confidence-aware max-pooling which also helps the confidence learning. The confidence-aware max-pooling guarantees that only measurements with higher confidence survive during down-sampling. Thus, improperly estimated confidences create an even worse estimation after down-sampling.

Our depth loss is given by Equation 2. When training the depth-completion and the loss network together, α is 0.5. Manually tuning α during the training leads to slightly faster convergence, but it does not worth the effort. When training only the depth-completion network, α is zero. Ψ is the L1-norm using the ground-truth depth as reference.

$$\mathcal{L} = \Psi(Depth_{guided}) + \alpha * \Psi(Depth_{lossnet}) \qquad (2)$$

All networks in Figure 2 can be trained in an end-to-end fashion. However, training only the depth completion network first and then using the weights to initialise the depth completion network and loss network for jointly training leads to a similar or better result, as well as faster convergence. The *conf-net* is always randomly initialised.

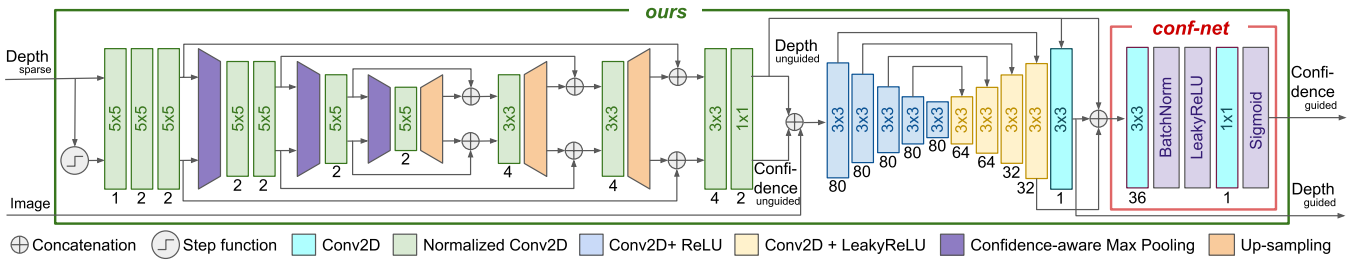

Fig. 4: This is our novel proposed network. Given as input a sparse depth map, it computes a binary mask using a step function that serves as the input confidence in a normalized convolution. Later on this network also uses the colour image to compute both a dense image-guided depth map and its associated confidences.

IV. Experiments

In this section, we compare our approach with several state-of-the-art network architectures on various datasets and present an ablation study. We use the standard error metrics of the KITTI depth completion challenge [4]: the Root Mean Square Error (RMSE m), the Mean Absolute Error (MAE m), the Mean Absolute Relative Difference (MARD $unitary$), and the Mean Square Error (MSE m^2), with MARD being particularly informative here as the datasets have large variations in scene depth. Given that our network uses the estimated confidence to guide the elimination of erroneous estimates, MSE is also very important because it highlights outliers.

A. Datasets and Setup

We perform experiments in three different scenarios: indoors, outdoors from an aerial platform, and from a car. As UAVs can also fly indoors, in some experiments we use both an indoor dataset and the proposed aerial datasets. This joint dataset is called Aerial+NYUv2. We use the following datasets:

- **NYUv2 dataset [6]:** this is an indoor dataset captured with a Microsoft Kinect camera. We use the same dataset split as in [19] with 48000 RGBD images for training and 654 images for validation. We also performed the same data augmentation as in [19], which includes a random application of rotation, scaling, flipping and colour jittering.
- **Aerial dataset:** this new dataset is described in Sec. III-A. We use the same image resolution, downsampling and augmentation as for the previous dataset, but with steps of 15° (instead of 5°) for rotation augmentation.
- **KITTI Depth Completion dataset [4]:** this well-established dataset has 85898 images for training and 6852 for validation. The test-set ground truth is not public and the official benchmark website does not accept confidences as input. So we are not testing on the test set. We performed the same cropping and data augmentation as in [20]. The samples have RGB image, sparse depth from LIDAR, and semi-dense ground-truth with about 30% coverage.
- **CAB dataset [3]:** this is an aerial and ground dataset recorded with a global-shutter camera. The aerial sequences were recorded by a small drone flying multiple times around a building. We used four videos at 1Hz with a total of 768 images. The ground truth was built using photogrammetry [5].
- **PVS dataset [36]:** this challenging real-world dataset is recorded from a manned aircraft over cities at much higher altitude than the CAB dataset. We use all three sequences; 180 images are available for DOWNTOWN, 240 images for CAPITOL and 226 images for BARUS&HOLLEY. Ground truth here was also built using photogrammetry [5].

Experimental Setup: All networks were implemented in PyTorch, and the original authors' code for the baseline networks was used. We adopt the Adam optimiser starting with a learning rate of 10^{-4} and reducing it by a factor of 10 every three epochs down to 10^{-6}. We let the model train for 24 hours and report the best epoch. The training was done using the NVIDIA GTX 1080 with up to 12GB of memory. We used the standard PyTorch weights for ResNet and batch size 8. All images were down-sampled to 320×240. Given the large variety of depths in the dataset, pre-scaling of the sparse depth was necessary before feeding it into the networks as demonstrated in Section IV-E. As sparse depth input, we used the given LIDAR input for KITTI, which has about 8% density (i.e. percentage of pixel with associated depth values) and random sampling over the ground truth for the other datasets. We chose 500 samples (0.65% density) to be similar to a SLAM algorithm and 10000 samples (8% density) to be similar to KITTI's LIDAR input.

B. Baseline Depth-Completion Network Architectures

As baseline methods we selected five state-of-the-art compact networks with public code from the KITTI Depth Completion challenge [4]: *(i) resnet18*, a ResNet architecture used in [19]; *(ii) u-resnet18*, a ResNet with skip connections used in [20]; *(iii) erfnet* [37] used as the core of the network described in [31]; *(iv) nconv-ed*, the normalized convolution net with an early-fusion encoder-decoder architecture from [24]; and *(v) nconv-ms*, the normalized convolution net with late-fusion multi-stream architecture from [24]. The *nconv-** methods can also compute confidences, but they are not image-guided.

C. Comparison to the State-of-the-art

Evaluation on Aerial+NYUv2 dataset - 500 samples: This is the training for later to be used with Visual-Inertial SLAM input running on aerial robots. Table II shows that our network at 100% density is nearly identical to *nconv-ed*, and both are better than all other networks at 100% density. Only our method using a confidence threshold that delivers an average of 90% density achieves significantly better results than all other methods. In particular, the MSE of the proposed method is almost 40% smaller than *nconv-ed*. This signifies a significant reduction of outliers sacrificing only 10% of the density. In Figure 6, it is visible that our confidences have low value on areas of high error, across all datasets and levels of input sparsity. The *nconv-ed* also computes confidences,

however on this highly sparse depth input, most pixels get the same low confidence. As a result, it was only possible to set a threshold in order to produce densities either 100% or smaller than 20% as observed in Figure 7.A. Figure 7.A also shows that our method has smaller MARD in densities greater than 20%.

Evaluation on Aerial+NYUv2 - 10K samples: This training is target to be later used by a UAV carrying a lightweight LIDAR sensor and a camera. Comparable behaviour is observed as before, when using 500 and 10K samples as input. However, the MSE improvement here is much greater, with *ours@90*'MSE almost **seven times smaller** than with *nconv-ed*. In Figure 6 and in the accompanying video of this work, it is clear that our method is capable of eliminating most of the mistakenly interpolated pixels around large depth discontinues. Figure 7.A also shows that our method has the smallest MARD across all densities.

Evaluation on KITTI: The KITTI dataset is very different from the aerial datasets as it has a strong bias towards the almost constant car viewpoint. On this dataset the proposed confidence-based filtering scheme is still beneficial and it is similar or better than the results in the test set. However, the improvement is not as good as in the Aerial+NYUv2 10K samples (note that KITTI also has around 10K input samples). The main problem is the absence of valid ground truth in areas with depth discontinuities as depicted in grey in Figure 5, so large parts of our enhanced results are ignored in the evaluation.

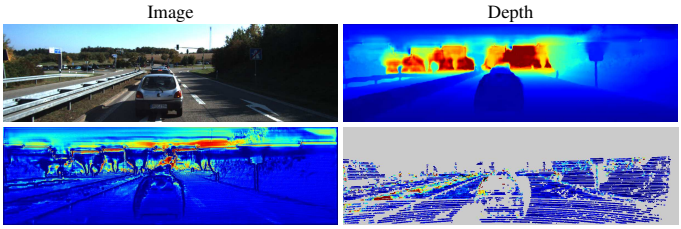

Fig. 5: Qualitative results on the KITTI dataset [4].

Dataset evaluation: Figure 7.A also shows that the depth completion and confidence networks trained with the our Aerial+NYUv2 dataset achieve better accuracy across all densities than with NYUv2 alone or Aerial alone.

D. Generalization Capability

We use the aerial outdoor real-world CAB and PVS datasets to validate the training done using both our synthetic Aerial dataset and NYUv2 with 500 and 10K samples. Figure 1 shows some example results. Figure 7.B shows that our method **can successfully perform depth completion** and confidence estimation in these real-world datasets with slightly higher MARD in all densities than in the original Aerial+NYUv2 validation set **without the need of style transfer or fine-tuning**.

E. Ablation Study

Effect of *conf-net*: The differences between *nconv-ed* and *ours@100* is the *conf-net* and the re-wiring for deep-features sharing. As shown in Table II, both methods have virtually the same results. We can conclude that feature sharing with dual purpose does not deteriorate the depth-completion results.

Effect of depth pre-scaling: We compare the per-frame scaling scheme with both no-scaling and global-scaling factors. We choose 400 meters as global factor because this is about the maximum depth value in the datasets used. The results in Table I show that the per-frame scaling has overall better results, so we used this scaling in all experiments.

Scale	per frame		1		1/400	
	MARD	RMSE	MARD	RMSE	MARD	RMSE
uresnet18 (500)	**0.051**	**2.675**	0.469	16.350	0.626	28.647
ours@100 (500)	**0.029**	**2.943**	0.032	3.008	0.031	3.002
uresnet18 (10k)	**0.026**	**1.490**	0.079	3.108	0.209	10.856
ours@100 (10k)	**0.008**	**1.151**	**0.008**	1.257	**0.008**	1.246

TABLE I: Pre-scaling factor effect on the Aerial+NYUv2 dataset. The per-frame scaling improves the results compared to other fixed scaling factors.

	MARD	MAE	RMSE	MSE
Aerial+NYUv2 - 500 samples				
uresnet18	0.051	1.486	2.675	19.958
resnet18	0.065	1.883	3.267	31.518
erfnet	0.117	3.231	4.763	42.547
nconv-ms	0.035	1.337	3.149	29.998
nconv-ed	0.029	1.179	2.922	27.882
ours@100	0.029	1.179	2.943	29.445
ours@90	**0.023**	**0.938**	**2.422**	**17.87**
Aerial+NYUv2 - 10000 samples				
uresnet18	0.026	0.730	1.490	6.524
resnet18	0.047	1.335	2.274	13.260
erfnet	0.139	3.354	4.199	21.621
nconv-ms	0.012	0.436	1.214	4.427
nconv-ed	0.008	0.319	1.162	4.736
ours@100	0.008	0.310	1.151	4.655
ours@90	**0.005**	**0.164**	**0.510**	**0.697**
Aerial - 500 samples				
uresnet18	0.263	3.533	5.236	60.821
nconv-ms	0.060	1.464	3.297	31.935
nconv-ed	0.032	1.226	3.024	28.627
ours@100	0.035	1.226	2.989	28.519
ours@90	**0.025**	**0.939**	**2.328**	**16.103**
NYUv2 - 500 samples				
nconv-ms	0.043	0.115	0.224	0.070
nconv-ed	0.043	0.116	0.226	0.073
ours@100	0.043	0.115	0.224	0.070
ours@90	**0.039**	**0.111**	**0.218**	**0.067**
KITTI Validation set				
resnet18 [19]	0.05	-	±2.2	-
nconv-ms [24]	-	0.210	0.909	-
nconv-ed [24]	-	0.237	1.008	-
nconv-ed	0.013	0.258	1.009	1.129
ours@100	0.014	0.264	1.018	1.149
ours@90	**0.010**	**0.181**	**0.597**	**0.404**
KITTI Test set				
uresnet18 [20]	-	0.250	**0.815**	-
nconv-ms [24]	-	**0.208**	0.859	-
IP-Basic [38]	-	0.303	1.288	-

TABLE II: Depth Estimation results. *ours* is the result for our network with 90% density. The results with references are taken from the respective papers, while all others are computed by us.

F. Visual-Inertial SLAM Input

Table III reveals the degradation in performance when using the more realistically available OKVIS data (i.e. input depth values from a nominal SLAM system instead of sampling ground-truth values or using a LIDAR scanner). This degradation is expected, given the estimation errors in SLAM, but also because texture-less areas, such as the sky, cannot have their depth measured. This experiment was done using only the 10K

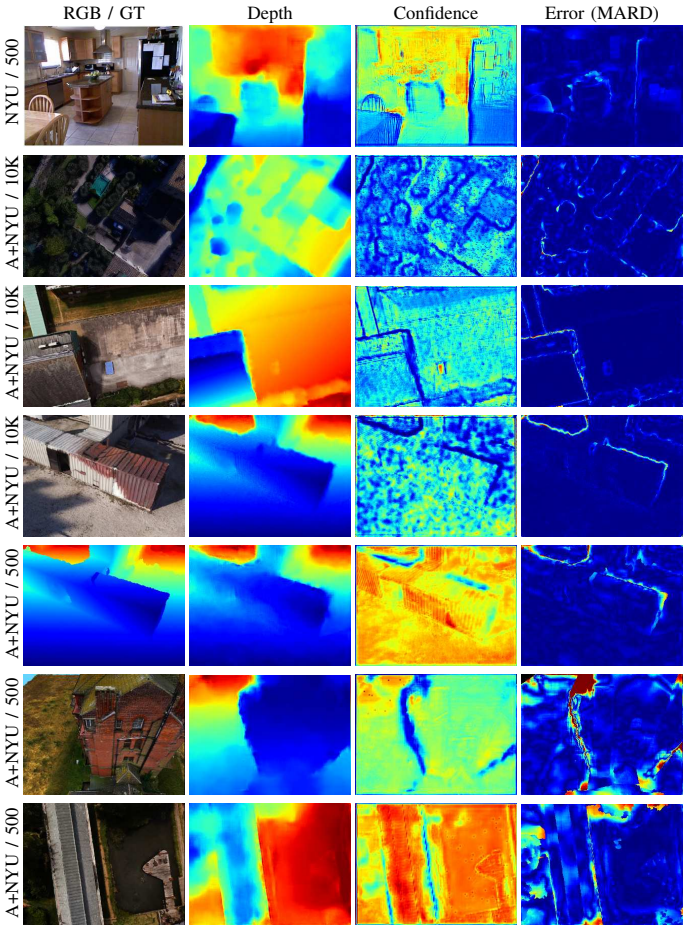

Fig. 6: Qualitative comparison on various scenes and between models trained with different sparsity. The vertical text in the first column states the dataset on which each model was trained on (e.g. A+NYU = Aerial+NYUv2 datasets), followed by the number of input samples (500 or 10K).

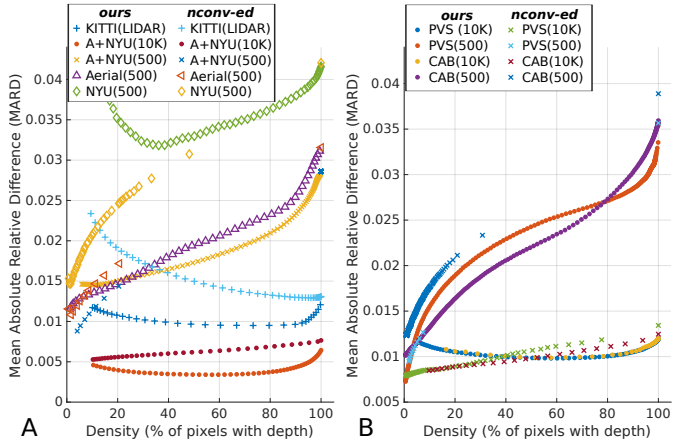

Fig. 7: Absolute relative error for various output densities resulting from a changing confidence threshold. The graph plots show the significant higher error reduction by filtering values according to the predicted confidences by our method instead of *nconv-ed*. This behaviour is consistent across different datasets and samples sizes. Even in training using synthetic data and testing in real-world datasets.

images in the Aerial validation set that has SLAM data. This experiment presents three types of input for the training of our network: (i) *Random Points* via random sampling of the ground-truth depth; (ii) *Keypoint GT* denoting ground-truth depth values available at the same pixel coordinates where the SLAM keypoints were detected; and (iii) *Keypoint SLAM*

	MARD	MAE	RMSE	MSE
Random 500 samples	0.031	1.139	1.859	4.218
Keypoint GT	0.071	2.567	4.024	19.177
Keypoint SLAM	0.076	2.900	4.104	19.866

TABLE III: Testing alternative input on a model trained with 500 depth samples.

using the depth information from the SLAM keypoints, which makes this experiment the closest to reality.

The error increase between *Random Points* and the *Keypoint GT* shows that the different distribution of points is the main source of error. In addition, the error increase between *Keypoint GT* and the *Keypoint SLAM* shows that the noise in the SLAM measurements further degrades the depth completion. The results demonstrate that the most significant error increase is due to the different point distribution and sparsity. Fig. 8 presents the results of the depth completion using both the *Random Point* input and *Keypoint SLAM* input in the same pre-trained model.

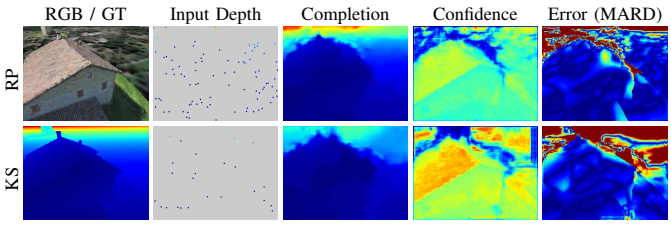

Fig. 8: Comparison between the results with 500 *Random Points* input (RP) and *Keypoint SLAM* input (KS) while using the same trained model. Top row: RGB, RP sparse depth input, RP depth completion, RP confidence, and RP depth absolute error. Bottom row: Ground truth depth in the first column followed by the KS's corresponding output in the subsequent columns.

G. Runtime

Our confidence and depth completion network can run on the NVIDIA Jetson TX2 GPU at 15 Hz, i.e. 0.06s per frame. Given that the Jetson TX2 has a shared memory, the low number of parameters used by our network is beneficial for the system as a whole, as more memory is available for other tasks. Our network has only 0.5 million parameters. Similarly, *nconv-ed* has almost identical performance and memory consumption. In contrast, *uresnet18* has 16 million parameters and it is five times slower than ours, while *nconv-ms* is twice slower.

V. Conclusion

We developed a single-view depth-completion and associated confidence estimation approach capable of handling LIDAR and VI-SLAM input sparsity. Experiments on several datasets and different state-of-the-art algorithms with and without confidence estimation demonstrate that our approach for image-guided confidence estimation is able to achieve unprecedented accuracy with a very small compromise in density by removing low-confidence predictions. Our network was also shown to be fast enough to run in real-time onboard a real UAV carrying a mobile GPU. We further introduced a large synthetic visual-inertial dataset for depth completion and a simulator including sample UAV trajectories around buildings. The proposed dataset exhibits high realism and

a much wider viewpoint range when compared to existing datasets. Our network trained on this dataset was successfully applied to the well-established real-world PVS and ETHZ CAB datasets without any fine-tuning.

Limitations and Future Work. Although the proposed model already explicitly predicts confidence values for all depth estimates, which are typically low along depth discontinuities, the reconstruction of object edges remains an open problem. This could be addressed by re-balancing training data with more examples of such challenging regions or using deeper architectures to better handle these cases.

While the dataset proposed here has been specifically designed for generality, containing a large variety of scenes and viewpoints, it poses a particular challenge in learning how to predict scene depth directly from a colour image. Instead, if the goal is to achieve high-fidelity depth estimates in a particular scene (e.g. for recurring flights over it), it is worth pursuing training on data for that specific scene to improve estimates in this and similar scenes. To maintain the generality of the network, however, a promising future direction is the incorporation of the uncertainty of the SLAM estimates in the network architecture. This can be a valuable source of information in addressing challenging estimation conditions, instead of trusting completely all estimates originating from SLAM as done so far.

REFERENCES

[1] L. Teixeira, I. Alzugaray, and M. Chli, "Autonomous Aerial Inspection using Visual-Inertial Robust Localization and Mapping," in *Field and Service Robotics*. Springer, 2018.

[2] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in *Proceedings of the European Conference on Computer Vision*, 2016.

[3] L. Teixeira and M. Chli, "Real-Time Mesh-based Scene Estimation for Aerial Inspection," in *Conference on Intelligent Robots and Systems*, 2016.

[4] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

[5] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition*, 2016.

[6] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012.

[7] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof, "Dense reconstruction on-the-fly," in *Conference on Computer Vision and Pattern Recognition*, 2012.

[8] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular Vision for Long-term MAV Navigation: A Compendium," *Journal of Field Robotics (JFR)*, vol. 30, 2013.

[9] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *International Conference on Robotics and Automation*, 2018.

[10] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, and R. Siegwart, "Keyframe-based Visual-Inertial SLAM using Nonlinear Optimization," in *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[11] L. Teixeira and M. Chli, "Real-time Local 3D Reconstruction for Aerial Inspection using Superpixel Expansion," in *International Conference on Robotics and Automation*, 2017.

[12] A. Rosinol, T. Sattler, M. Pollefeys, and L. Carlone, "Incremental Visual-Inertial 3D Mesh Generation with Structural Regularities," in *International Conference on Robotics and Automation*, 2019.

[13] A. Concha and J. Civera, "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence," in *Conference on Intelligent Robots and Systems*, 2015.

[14] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014.

[15] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep Ordinal Regression Network for Monocular Depth Estimation," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[16] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[17] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[19] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *International Conference on Robotics and Automation*, 2018.

[20] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *International Conference on Robotics and Automation*, 2019.

[21] C. S. Weerasekera, T. Dharmasiri, R. Garg, T. Drummond, and I. D. Reid, "Just-in-time reconstruction: Inpainting sparse maps using single view depth predictors as priors," in *International Conference on Robotics and Automation*, 2018.

[22] Y. Zhang and T. A. Funkhouser, "Deep depth completion of a single RGB-D image," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[23] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from RGB and sparse sensing," in *Proceedings of the European Conference on Computer Vision*, 2018.

[24] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through cnns for guided sparse depth regression," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[25] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, 2018.

[26] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016.

[27] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, 2017.

[28] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017.

[29] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural rgb (r) d sensing: Depth and uncertainty from a video camera," in *Conference on Computer Vision and Pattern Recognition*, 2019.

[30] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Conference on Computer Vision and Pattern Recognition*, 2017.

[31] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *International Conference on Machine Vision Applications*, 2019.

[32] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *Conference on Computer Vision and Pattern Recognition*, 2019.

[33] P. Sujit, S. Saripalli, and J. B. Sousa, "An evaluation of uav path following algorithms," in *2013 European Control Conference*, 2013.

[34] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *RotorS—A Modular Gazebo MAV Simulator Framework*, 2016.

[35] A. Eldesokey, M. Felsberg, and F. S. Khan, "Propagating confidences through cnns for sparse data regression," in *BMVC*, 2018.

[36] M. I. Restrepo, A. O. Ulusoy, and J. L. Mundy, "Evaluation of feature-based 3-d registration of probabilistic volumetric scenes," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, 2014.

[37] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[38] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the cpu," in *Conference on Computer and Robot Vision (CRV)*, 2018.