PSMNet-FusionX3: LiDAR-Guided Deep Learning Stereo Dense Matching On Aerial Images

Teng Wu, Bruno Vallet, Marc Pierrot-Deseilligny
LASTIG, Univ Gustave Eiffel, IGN-ENSG
firstname.lastname@ign.fr

Abstract

Dense image matching (DIM) and LiDAR are two complementary techniques for recovering the 3D geometry of real scenes. While DIM provides dense surfaces, they are often noisy and contaminated with outliers. Conversely, LiDAR is more accurate and robust, but less dense and more expensive compared to DIM. In this work, we investigate learning-based methods to refine surfaces produced by photogrammetry with sparse LiDAR point clouds. Unlike the current state-of-the-art approaches in the computer vision community, our focus is on aerial acquisitions typical in photogrammetry. We propose a densification pipeline that adopts a PSMNet backbone with triangulated irregular network interpolation based expansion, feature enhancement in cost volume, and conditional cost volume normalization, i.e. PSMNet-FusionX3. Our method works better on low density and is less sensitive to distribution, demonstrating its effectiveness across a range of LiDAR point cloud densities and distributions, including analyses of dataset shifts. Furthermore, we have made both our aerial (image and disparity) dataset and code available for public use. Further information can be found at https://github.com/whuwuteng/PSMNet-FusionX3.

1. Introduction

Reconstructing the 3D geometry of real scenes is a crucial task in computer vision and photogrammetry. Various methods, including feature-based, optimization-based, and machine learning-based methods, have been developed to tackle this challenge. With the advent of deep learning and autonomous driving, convolution neural networks (CNN) have been widely applied to dense image matching (DIM). On the other hand, the development of LiDAR technology made it more accessible, especially in automotive driving, as illustrated by popular datasets: KITTI [24], Robotcar [22], and more recently Radar [2]. While image pixels are usually more densely packed than LiDAR points for the same platform, LiDAR provides higher geometric accuracy and robustness. As such, it is natural to aim at combining the advantages of LiDAR and images in order to achieve both the geometric quality of LiDAR and the high resolution of images.

Dense 3D reconstruction by combining image and LiDAR data have been an important topic in the computer vision community, and several open datasets have been proposed to facilitate research in this area [33]. In aerial photogrammetry, images have much higher spatial resolution compared to sparser LiDAR, but the geometric quality of LiDAR can help to improve the DIM performance. This paper focuses on using LiDAR to guide an aerial photogrammetry DIM pipeline. While this topic has been addressed based on traditional optimization frameworks (such as based on semi-global matching) [13], it has been scarcely investigated using deep learning techniques. The
most similar work is conditional cost volume normalization (CCVNNorm) [34], whose backbone is GCNet [16]. Therefore, we refer to it as GCNet-CCVNNorm in this paper. However, GCNet-CCVNNorm performs poorly on low-density LiDAR input, and the building boundaries are often inaccurate, as shown in Figure 1.

This paper focuses on deep learning-based LiDAR-guided DIM in the aerial context. We generate training data, investigate the influence of the LiDAR density, distribution imbalance, and propose PSMNet-FusionX3, a PSMNet-based LiDAR-guided DIM pipeline. PSMNet-FusionX3 combines the advantage of PSMNet, including clean and accurate depth discontinuities and good handling of shadowed and uniform areas, with the LiDAR guidance improvement. This is illustrated in Figure 1. We also generate a high-resolution and high-density stereo dense matching aerial dataset for stereo-LiDAR fusion and evaluate our approach on it. Our contributions can be summarized as follows:

1. We propose to use a triangulated irregular network (TIN) interpolation-based expansion for the aerial image, and integrate it with the network for more accurate depth estimation.

2. We firstly combine TIN interpolation, feature enhancement in cost volume, and CCVNrom with PSMNet backbone, i.e. PSMNet-FusionX3, it is more robust to the density and distribution of the LiDAR guidance.

3. We build a high resolution and high dense stereo dense matching aerial dataset for stereo-LiDAR fusion, and exploit the DL based stereo-LiDAR methods on the aerial dataset.

2. Related Work

2.1. Stereo dense matching

Stereo Dense image matching (DIM) is a widely-explored topic with proposed solutions can be categorized into local methods [32], global optimization methods such as Semi global matching (SGM) [12], traditional machine learning methods [4], and deep learning (DL) method [21]. As this paper focuses on DL methods, we will primarily review works using DL to replace one or all (end-to-end) steps of the DIM pipeline.

Single step DL. 2D CNNs have proved effective to replace the feature extraction step [38]. For instance, SGM-Net uses a CNN network to learn the penalties in SGM [30]. In the refinement step, variational networks can be used to refine the disparity [19].

End-to-end DL. In the pioneering work DispNet [23], a 2D CNN encoder-decoder structure named DispNetS was the first end-to-end network processing two images as 6 bands to obtain the disparity, while DispNetC investigates cost aggregation with a 2D CNN. Another pioneering work GC-Net proposes a 3D CNN-based network for cost aggregation in the cost volume [16]. The Pyramid Stereo Matching network uses spatial pyramid pooling to extract the feature and 3D CNN stack network to process the cost volume [5]. High-resolution stereo network structures utilize upsampling in the 2D CNN network, so the 3D cost volume can be upsampled inorder to have a better resolution [36]. For high-resolution image matching, the 3D cost volume can be pruned with a differential patch match method to handle the high memory consumption [8]. A 3D CNN named GANet integrates a Semi-Global Guided Aggregation (SGA) layer and a Local Guided Aggregation (LGA) layer [39]. More recently, reinforcement learning such as neural architecture search was applied to the stereo dense matching problem [6]. Finally, the boundary information can be exploited by a hierarchical refinement [17].

2.2. LiDAR guided stereo dense image matching

With the emergence of LiDAR and Radar technology, it has become possible to acquire images and point cloud data simultaneously. This allows for the use of LiDAR data to improve DIM. Traditionally, LiDAR 3D points are utilized as constraints in DIM [13], a Gaussian enhancement function has been proposed as a means to guide DIM [28]. This approach is applicable to both traditional global stereo methods and some cost-volume-based deep learning methods, such as PSMNet. More recently, a riverbed enhancement function instead of Gaussian function has been introduced to further improve the quality of guidance information used in DIM [41].

DIM guided by LiDAR has also been investigated in the context of autonomous driving [37], as LiDAR and Radar can be mounted on a car [1,3]. Research in this area has focused on Guided Stereo Matching and Stereo-LiDAR fusion, taking inspiration from [28]. However, as these terms are not yet clearly defined, we propose to classify these methods into three distinct categories instead:

Fusion in 2D input or output. Fusing information in 2D is an intuitive approach and can be easily implemented using LiDAR data. One common technique is to add an extra branch to the traditional DIM networks, which extracts features from a sparse LiDAR depth map and employs a 2D CNN to generate the final disparity [14,34,40]. Another strategy is to refine the disparity by combining LiDAR data with the results from traditional SGM methods, and then using the image to refine the disparity map [25].

Fusion in cost volume. A Gaussian kernel has been demonstrated to be effective in enhancing the feature vector
Algorithm 1 Algorithm for sparse disparity interpolation

Input: sparse disparity map \( D \)
Output: interpolation map \( D \), confidence map \( M \)

1: build TIN using 2D Delaunay triangulator with \( x, y \) where \( D(x, y) > 0 \) (with disparity)
2: for \( \Delta ABC \) in TIN do
3: if Equation (3) is true then
4: for pixel \( S(x_s, y_s) \) in \( \Delta ABC \) in Figure 2a do
5: calculate \( d_sM_s \) using Equation (2)
6: \( D(x_s, y_s) = d_s \)
7: \( M(x_s, y_s) = M_s \)
8: end for
9: end if
10: end for
11: return \( D, M \)

within the cost volume [34]. Inspired by this work, refine the Gaussian value by expanding the sparse disparity and integrating it with the confidence metric in [14]. Similarly, 3D line and 3D graph hint expansion techniques are used to expand the sparse hints from the LiDAR data [27].

Fusion in 3D space. A graph-based depth correction approach has been proposed to refine the depth of dense learning stereo methods in 3D space [37]. This approach has been leveraged to merge the LiDAR data in 3D space with the predicted disparity map [14]. Moreover, with the advancement of 3D convolution techniques for point cloud processing such as PointNet [29], DL-based point features can also be extracted and incorporated into the cost volume [7].

Our study, PSMNet-FusionX3, uses a simple backbone network and integrates LiDAR guidance throughout the cost volume fusion in three steps.

3. PSMNet-FusionX3

Prior to introducing the pipeline, we provide a comprehensive description of the datasets utilized for training and evaluation, which were generated using aerial data generation [35]. The input LiDAR data is highly dense, resulting in a disparity map that is sufficiently dense for the experiment. Under these conditions, we can sample the dense disparity map using various ratios and strategies, followed by the exploitation of densification on the sparse disparity.

3.1. Sparse disparity interpolation

Our aerial image/LiDAR dataset encompasses an urban area where disparity remains relatively regular due to the presence of man-made objects. As the input disparity guidance is sparse in epipolar image geometry, we interpolate it using a 2D Delaunay triangulation [31] to generate a triangulated irregular network (TIN) from the disparity map. Although interpolation may introduce errors, an confidence map can be beneficial in assessing interpolation quality. There is a similar approach [15] used in [14], where the expansion is based on pixel values. If the neighbor pixel has the same intensity or a small difference as the known pixel, the disparity should be the same; a threshold of 2 was used in their experiment. Nevertheless, selecting an appropriate threshold for remote sensing images is challenging. To overcome this issue, we define confidence using a location-dependent parameter \( M \) [10]. We produce an confidence map for interpolation and leverage it as a weight in the loss function.

\[
a = \frac{(y_2 - y_3)d_1 + (y_3 - y_1)d_2 + (y_1 - y_2)d_3}{x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_1y_3 - x_2y_1} = a_1d_1 + a_2d_2 + a_3d_3
\]

\[
b = \frac{(x_3 - x_2)d_1 + (x_1 - x_3)d_2 + (x_2 - x_1)d_3}{x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_1y_3 - x_2y_1} = b_1d_1 + b_2d_2 + b_3d_3
\]

\[
c = \frac{(x_2y_3 - x_3y_2)d_1 + (x_3y_1 - x_1y_3)d_2 + (x_1y_2 - x_2y_1)d_3}{x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_1y_3 - x_2y_1} = c_1d_1 + c_2d_2 + c_3d_3
\]  

(1)
The process of triangle linear interpolation is described in great detail in Equation (1), and the specifics of the variables $a_1, b_1, c_1$ etc. can be found in the supplementary document. Additionally, the confidence parameter $M$ is defined in Equation (2), and both are visually represented in Figure 2. The complete outcome of these calculations is demonstrated in Figure 3.

$$d_s = ax_s + by_s + c$$
$$= (a_1d_1 + a_2d_2 + a_3d_3)x_s + (b_1d_1 + b_2d_2 + b_3d_3)y_s$$
$$+ (c_1d_1 + c_2d_2 + c_3d_3)$$
$$= (a_1x_s + b_1y_s + c_1)d_1 + (a_2x_s + b_2y_s + c_2)d_2$$
$$+ (a_3x_s + b_3y_s + c_3)d_3$$
$$M = (a_1x_s + b_1y_s + c_1)^2 + (a_2x_s + b_2y_s + c_2)^2$$
$$+ (a_3x_s + b_3y_s + c_3)^2$$

(2)

During our experiments, we discovered that interpolation can introduce significant errors during training, which in turn can degrade performance. As we know, the majority of errors arise during interpolation in areas of disparity discontinuity, where the disparity difference between vertices is considerable. To mitigate these errors, we enforced a limitation on the disparity difference at triangle vertices (cf. Equation (3)), setting it to $\Delta d = 3$ for our experiments. The corresponding pseudo-code is presented Algorithm 1.

$$\begin{align*}
|d_1 - d_2| &< \Delta d \\
|d_1 - d_3| &< \Delta d \\
|d_2 - d_3| &< \Delta d
\end{align*}$$

(3)

3.2. Network

The primary contribution of this study is the incorporation of CCVNorm into the PSMNet backbone model (PSMNet-FusionX3) that is inspired by GCNet-based CCVNorm (GCNet-CCVNorm) [34], and with improved performance, given that PSMNet outperforms GCNet. Our proposed network, PSMNet-FusionX3, integrates the LiDAR guidance in the PSMNet framework [5], as depicted in Figure 4. The guidance is added in three phases:

1. in the 2D CNN processing step, the disparity map is used as the 4th band.
2. in the cost volume, incorporate the left guidance map adding the weight into the cost volume [28].
3. in the 3D CNN step, i.e. CCVNorm integrates the guidance.

The inputs for our network are a stereo image, along with the interpolated guidance maps and confidence maps. To begin, features are extracted by a 2D CNN along a Spatial Pyramid Pooling (SPP) module. Next, a cost volume is constructed, with feature enhancement added, and finally, CCVNorm is integrated into the 3D CNN like in [34], as illustrated in Figure 4.

3.3. Loss

The PSMNet model employs the $L_1$ loss function during training, which we also adopt here. As the guidance pixels form part of the ground truth disparity, they are likely to be more accurate than non-guidance pixels. To account for this, we generate a weight map for each pixel based on the confidence outlined in Section 3.1. These weight maps are used to assign a pixel-based weight in the loss function.

$$L(d, \hat{d}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L_1}(d_i - \hat{d}_i) \cdot W_{confidence}$$

(4)

where $N$ is the number of the pixels with ground truth disparity, $d$ is the ground-truth disparity, and $\hat{d}$ is the predicted disparity, and

$$smooth_{L_1} = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$$

we use the scale to define $W_{confidence}$ from $M$ defined in Section 3.1:

$$W_{confidence} = \begin{cases} M \cdot 5.0, & \text{if } M > 0 \\ 1, & \text{otherwise} \end{cases}$$

The Adam optimization algorithm [18] is used in the experiment.

4. Experiment

Our experiments were conducted using the DublinCity [20] and Toulouse2020 dataset, which are described in Table 1, more detail can be found in the supplementary document. The parameter $R_t$ is defined as $\frac{w \times h}{N_{valid}}$, where $N_{valid}$ represents the number of valid pixels with ground truth disparity, and $w$ and $h$ are the width and height of the cropped image. We utilized a GPU cluster equipped with Tesla V100 cards with 32GB of memory. To handle the memory-intensive cost volume processing in PSMNet-FusionX3, we set the batch size to 8. Further details on the training set configuration and training times are provided in Table 2. During the testing phase, there were 822 pairs for DublinCity and 411 pairs for Toulouse2020.

Our proposed method, PSMNet-FusionX3, was compared against several baseline methods, including SGM [11], PSMNet [5], GuideStereo [28], GCNet and GCNet-CCVNorm [34] (GCNet lacks official code, code from GCNet-CCVNorm).
Figure 4. The PSMNet-FusionX3 network. The stereo images and their corresponding interpolated disparity maps serve as inputs for the two weight-sharing pipelines that consist of a 2D CNN and a Spatial Pyramid Pooling (SPP) Module. Following 2D CNN processing, utilizing the learned image feature, a 4D cost volume is constructed. At this stage, the green circle indicates that the left guidance map can also be utilized to add weight to the cost volume. Subsequently, the cost volume is processed by a Stacked Hourglass network that comprises a 3D CNN, a 3D Conv CVVNorm, and a 3D DeConv CVVNorm. Ultimately, the predicted output disparity map is generated through bilinear interpolation followed by regression, and the input confidence map is leveraged to weigh the loss.

<table>
<thead>
<tr>
<th>dataset</th>
<th>GSDi (cm)</th>
<th>Di (pt/m^2)</th>
<th>Ri (px/pt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DublinCity</td>
<td>3.4</td>
<td>250-348</td>
<td>2.3</td>
</tr>
<tr>
<td>Toulouse2020</td>
<td>5</td>
<td>≈50</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 1. Image Ground Sampling Distance (GSDi), LiDAR density Di, and image guide density Ri, which means Ri pixels with 1 LiDAR point.

4.1. Batchsize in PSMNet-FusionX3

As shown in Table 2, the batch size for PSMNet-FusionX3 is 8, while it is 12 for PSMNet due to CCVNorm with 3D-CNN is high memory requirements. To investigate the impact of batch size on PSMNet-FusionX3, we employed PyTorch Lightning [9] to train on 4 GPU nodes. Interestingly, from Table 3, we observed that while the result remained unchanged with a batch size of 8, the training time per epoch was reduced. This indicates that using larger batch sizes for training on larger datasets can save time.

4.2. Density analysis

The ratio between the density of LiDAR points (pt/m^2) and image pixels (px/pt) is an important factor in LiDAR guided stereo DIM. To investigate its impact, we randomly selected valid pixels (with LiDAR points) to achieve ratios of 0.5%, 1%, 2.5%, 5%, and 10%. The ratio is expressed as $\frac{N_{\text{selected}}}{N_{\text{valid}}}$, where $N_{\text{selected}}$ is the number of selected pixels and $N_{\text{valid}}$ is the number of valid pixels with ground truth disparity. $R_i$ refers to the pixel with LiDAR guidance. The equivalent density can be computed and is presented in Table 4. For the DublinCity dataset, we used an average density of 300. The 1-pixel error will be analyzed and the 3-pixel error can be found in the supplementary document.

As the disparity map is 2.5D, we employed TIN to generate a dense disparity map, which can be used as a baseline to allow us to evaluate the LiDAR and image fu-
Table 2. Training configuration and runtime for each method. The PyTorch-Lightning (PL) implementation of PSMNet-FusionX3 runs on 4 GPUs, and the other methods only run on a single GPU.

<table>
<thead>
<tr>
<th>method</th>
<th>epochs</th>
<th>crop size</th>
<th>batch size</th>
<th>train time(s/epoch)</th>
<th>pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSMNet</td>
<td>500</td>
<td>256x512</td>
<td>12</td>
<td>216</td>
<td>1200</td>
</tr>
<tr>
<td>GCNet</td>
<td>40</td>
<td>256x512</td>
<td>4</td>
<td>2851</td>
<td>1200</td>
</tr>
<tr>
<td>GCNet-CCVNorm</td>
<td>40</td>
<td>256x512</td>
<td>4</td>
<td>3457</td>
<td>1200</td>
</tr>
<tr>
<td>PSMNet-FusionX3 (ours)</td>
<td>500</td>
<td>256x512</td>
<td>8</td>
<td>335</td>
<td>1200</td>
</tr>
<tr>
<td>PSMNet-FusionX3-PL (ours)</td>
<td>500</td>
<td>256x512</td>
<td>12</td>
<td>113</td>
<td>1200</td>
</tr>
</tbody>
</table>

Table 3. Influence of the batch size of PSMNet-FusionX3. Testing on Toulouse2020 dataset with a LiDAR density ratio of 5%. The result remained unchanged even increasing the batch size.

<table>
<thead>
<tr>
<th>method</th>
<th>batch size</th>
<th>&lt;1-pixel</th>
<th>&lt;2-pixel</th>
<th>&lt;3-pixel</th>
<th>&lt;5-pixel</th>
<th>&lt;9-pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSMNet-FusionX3</td>
<td>8</td>
<td>85.94</td>
<td>91.82</td>
<td>94.45</td>
<td>97.08</td>
<td>99.15</td>
</tr>
<tr>
<td>PSMNet-FusionX3(Lightning)</td>
<td>12</td>
<td>85.78</td>
<td>91.70</td>
<td>94.34</td>
<td>96.97</td>
<td>99.09</td>
</tr>
</tbody>
</table>

Table 4. \(D_l\) is the \(\frac{N_{valid}}{N_{selected}}\), equivalent density (pt/m²) of input LiDAR in experiment, \(R_l\) is image guide density (px/pt).

The 1-pixel error on the Toulouse2020 dataset is depicted in Figure 5. Our PSMNet-FusionX3 was observed to leverage the strengths of both PSMNet and GCNet-CCVNorm, indicating that LiDAR information can be beneficial for PSMNet-FusionX3, even at very low density, without adversely impacting the performance of GCNet-CCVNorm.

The 1-pixel error on the DublinCity dataset is illustrated in Figure 6. Here, we observed similar behavior as on Toulouse2020, with the exception that GuideStereo outperformed TIN interpolation except for the highest densities. Moreover, our PSMNet-FusionX3 outperformed GCNet-CCVNorm by a larger margin.

4.3. Density distribution analysis

In the previous experiment, we generated sparse input disparity maps of varying densities by uniformly subsampling the dense disparity map. As observed in a similar study in [14], the resulting signal distribution imbalance can have a significant impact on the outcome. In this section, we investigate how the distribution of LiDAR affects the learning process. To address this issue, we propose a Gaussian subsampling strategy that generates a 2D Gaussian distribution centered at the middle of the image, as shown in Figure 7.

The results presented in Figure 8 and Figure 9 clearly indicate that a distribution imbalance always leads to increased errors. While the distribution of LiDAR has a considerable effect on GCNet-CCVNorm, GuideStereo and PSMNet-FusionX3 are comparatively less sensitive.

4.4. Dataset shift analysis

The dataset shift is a significant factor in practical applications because it is not always possible to have a training dataset available in the production area. To address this challenge, we can use the model trained on Toulouse2020 to test on DublinCity or vice versa. The impact of dataset shift, measured in terms of 1-pixel error, is depicted in Figure 10 and Figure 11. For the LiDAR guided methods, the input LiDAR density ratio is consistent across both datasets. End-to-end training methods can also be trained on the
Figure 7. An example of random and Gaussian subsampling and the corresponding interpolations of a disparity map from Toulouse2020. For both uniform and Gaussian subsampling, the sampling ratio is 5%.

Figure 8. The influence of the distribution of LiDAR on Toulouse2020. GCNet-CCVNorm is highly influenced by the distribution of LiDAR. GuideStereo and PSMNet-FusionX3 are less sensitive, and the imbalanced result is much worse than the randomly sampled result.

Figure 9. The influence of the distribution of LiDAR on DublinCity. The behavior is the same as Toulouse2020. GCNet-CCVNorm is highly influenced by the distribution, and PSMNet-FusionX3 is less sensitive compared to Toulouse2020.

sparse ground truth disparity using LiDAR guidance. In light of GCNet’s poor performance (cf. Section 4.2), PSMNet training on the sparse LiDAR guidance is introduced. Notably, for both dataset shifts, TIN interpolation outperforms GuideStereo for almost all LiDAR densities, SGM outperforms GCNet, LiDAR input guidance enhances the performance of GCNet-CCVNorm and PSMNet-FusionX3, and PSMNet-FusionX3 outperforms GCNet-CCVNorm.

4.5. Visual assessment

To analyze the error distribution, we have provided an error map of DublinCity in Figure 12. Further analysis
on Toulouse2020 can be found in the supplementary document. When the input LiDAR density is at 5%, Figure 12c shows significant improvement in LiDAR guidance on high vegetation and large depth discontinuities. Moreover, compared to GCNet, the performance of GCNet-CCVnorm has improved significantly. In comparison to GCNet-CCVnorm, PSMNet-FusionX3 generates a smoother outcome, especially in uniform areas. Therefore, our PSMNet-FusionX3 successfully combines the quality and robustness of PSMNet while appropriately utilizing LiDAR guidance.

5. Conclusion

This paper explores the use of sparse LiDAR to guide DL-based Stereo DIM on high-resolution aerial imagery. We introduce a novel method, PSMNet-FusionX3, which combines TIN-based interpolation, feature enhancement in cost volume, and CCVNorm in PSMNet backbone. By leveraging the strengths of PSMNest and GCNet-CCVnorm, PSMNet-FusionX3 surpasses the state-of-the-art for any LiDAR density, and is less susceptible to the distribution of the LiDAR. We also demonstrate that our approach is robust to dataset shifts between two European cities. We have made available a high-resolution aerial dataset for training DP-based DIM methods.

During our experiments, we found that errors in interpolation can negatively impact the outcome, underscoring the importance of guidance LiDAR quality. The dataset is derived from an urban area where disparity discontinuity is regular, and interpolation performs well. However, it poses a challenge for small structure objects since interpolation can be inaccurate. A promising solution is to enhance the weight of image information. Future work could explore the influence of inconsistencies between images and LiDAR (especially if they were not acquired simultaneously). Furthermore, considering more confidence factors in training could be an interesting avenue to explore, for example when the image is more reliable than the LiDAR guidance.

6. Acknowledgments

We acknowledge the AI4GEO project for fully funding this work, and express our gratitude to Ewelina Rupnik for her valuable discussions and advice. Additionally, we extend our thanks to the CNES (French space agency) for providing access to the GPU cluster HAL (High Performance Computing).

References


