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

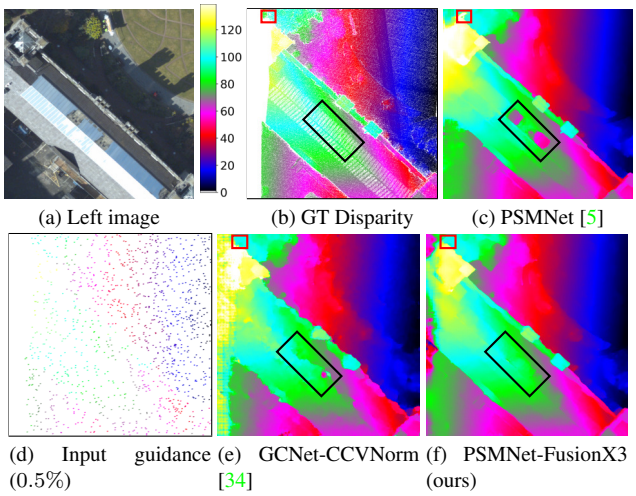
Teng Wu, Bruno Vallet, Marc Pierrot-Deseilligny
LASTIG, Univ Gustave Eiffel, IGN-ENSG
firstname.lastname@ign.fr

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



(a) Left image (b) GT Disparity (c) PSMNet [5]

(d) Input guidance (0.5%) (e) GCNet-CCVNorm [34] (f) PSMNet-FusionX3 (ours)

Figure 1. Comparison of DIM on DublinCity aerial dataset [20]: our proposed PSMNet-FusionX3 (f) is capable of combining the robustness of PSMNet [5] and LiDAR guidance when the guidance LiDAR density is low (0.5%) in (d). As a result, it outperforms the *state-of-the-art* GCNet-CCVNorm [34] (e), especially in shadow (red rectangle) and uniform (black rectangle) areas.

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 (CCVNorm) [34], whose backbone is GCNet [16]. Therefore, we refer to it as GCNet-CCVNorm in this paper. However, GCNet-CCVNorm 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 upscaling in the 2D CNN network, so the 3D cost volume can be upscaled 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 Guassian 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

build TIN using 2D Delaunay triangulator with x, y where $D(x, y) > 0$ (with dispairty)
for ΔABC in TIN **do**
  if Equation (3) is true **then**
    for pixel S (x_s, y_s) in ΔABC in Figure 2a **do**
      calculate d_s,M_s using Equation (2)
      $D(x_s, y_s) = d_s$
      $M(x_s, y_s) = M_s$
    end for
  end if
end for
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 deep 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].

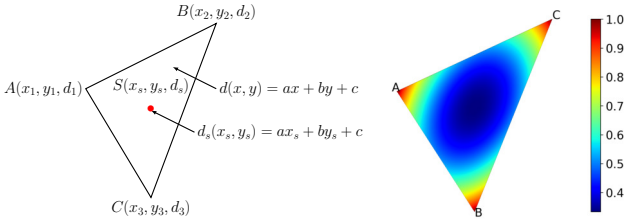
Our study, PSMNet-FusionX3, uses a simple backbone network and incorporates LiDAR guidance through 2D input and cost volume fusion in three steps.

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.

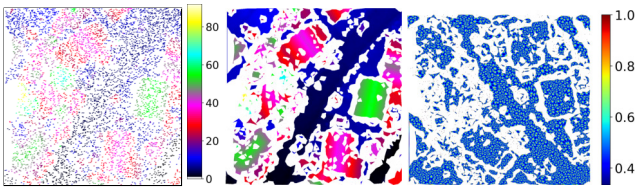
3.1. Sparse disparity interpolation

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) Triangle plane (b) Interpolation confidence

Figure 2. Illustration of the linear interpolation of disparity values that are solely known on the vertices of a given triangle, the symbols in (a) are used in Equation (1). In (b), Equation (2) is used to calculate the confidence of the interpolation which is influenced by the shape of the triangle, and the distance to the vertice of the triangle.



(a) Randomly sample (b) Randomly w/ TIN (c) confidence of TIN

Figure 3. An example of sparse disparity interpolation from the Toulouse2020 dataset, where (a) represents the disparity randomly subsampled from the dense disparity map using a 2.5% ratio, (b) displays the linear TIN interpolation outcome, and (c) presents the corresponding interpolation confidence.

$$
\begin{aligned}
a &= \frac{(y_2 - y_3)d_1 + (y_3 - y_1)d_1 + (y_1 - y_2)d_3}{x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_1y_3 - x_2y_1} \\
&= a_1d_1 + a_2d_2 + a_3d_3 \\
b &= \frac{(x_3 - x_2)d_1 + (x_1 - x_3)d_1 + (x_2 - x_1)d_3}{x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_1y_3 - x_2y_1} \\
&= b_1d_1 + b_2d_2 + b_3d_3 \\
c &= \frac{(x_2y_3 - x_3y_2)d_1 + (x_3y_1 - x_1y_3)d_1 + (x_1y_2 - x_2y_1)d_3}{x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_1y_3 - x_2y_1} \\
&= c_1d_1 + c_2d_2 + c_3d_3
\end{aligned}
\tag{1}
$$

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.

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

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.

$$\begin{cases} |d_1 - d_2| < \Delta d \\ |d_1 - d_3| < \Delta d \\ |d_2 - d_3| < \Delta d \end{cases} \tag{3}$$

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 map, 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.

$$L(d, \hat{d}) = \frac{1}{N}\sum_{i=1}^{N} smooth_{L_1}(d_i - \hat{d}_i) * W_{confidence} \tag{4}$$

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

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

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

$$W_{confidence} = \begin{cases} M * 5.0, & ifM > 0 \\ 1, & otherwise \end{cases}$$

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_i 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).

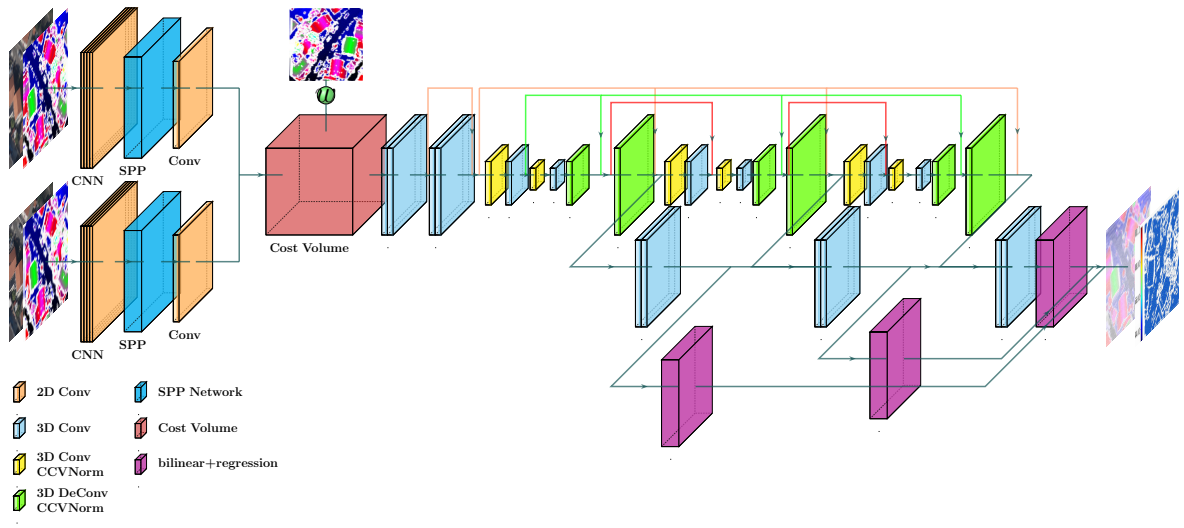


Figure 4. The PSMNet-FusionX3 nework. The stereo images and their corresponding interpolated disparity maps serve as inputs for the two weight-sharing pipelines that consist of a 2D CNN and a Spatial Pyramid Pooling (SPP) Module. Following 2D CNN processing, utilizing the learned image feature, a 4D 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.

dataset	GSD_i (cm)	D_l (pt/m^2)	R_i (px/pt)
DublinCity	3.4	250-348	2.3
Toulouse2020	5	≈ 50	3.0

Table 1. Image Ground Sampling Distance (GSD_i), LiDAR density D_l, and image guide density R_i which means R_i 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 $\frac{N_{selected}}{N_{valid}}$, where $N_{selected}$ is the number of selected pixels and N_{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.

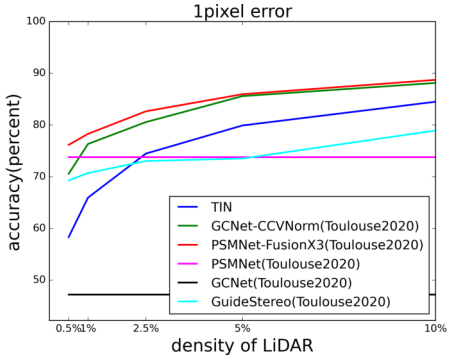


Figure 5. Influence of LiDAR density on 1-pixel error on Toulouse2020. For image-only DIM, PSMNet is much better than GCNet. GuideStereo does not improve over image only below 5% ratio. Above, its performance is slightly better than PSMNet, but worse than the simple TIN interpolation. GCNet-CCVNorm is better than GuideStereo and below image only DIM only for very low density (0.5%). Our PSMNet-FusionX3 is always slightly better than GCNet-CCVNorm and is the only guided method that outperforms PSMNet at the lowest density of 0.5%

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-

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

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.

method	batch size	<1-pixel	<2-pixel	<3-pixel	<5-pixel	<9-pixel
PSMNet-FusionX3	8	85.94	91.82	94.45	97.08	99.15
PSMNet-FusionX3(Lightning)	12	85.78	91.70	94.34	96.97	99.09

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.

dataset	0.5%	0.1%	2.5%	5%	10%
DublinCity(D_l)	1.5	3	7.5	15	30
DublinCity(R_i)	462.0	231.0	92.4	46.2	23.1
Toulouse2020(D_l)	0.25	0.5	1.25	2.5	5
Toulouse2020(R_i)	605.9	302.9	121.1	60.6	30.3

Table 4. D_l is the $\frac{N_{selected}}{N_{valid}}$, equivalent density ($pt/m^2$) of input LiDAR in experiment, R_i is image guide density (px/pt).

sion performance. 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.

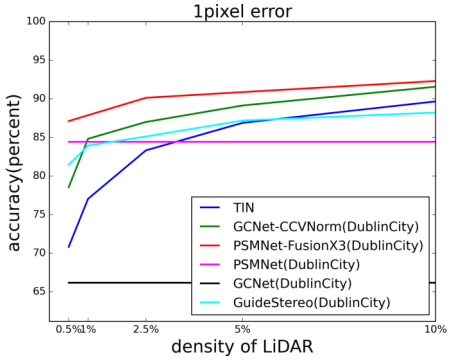
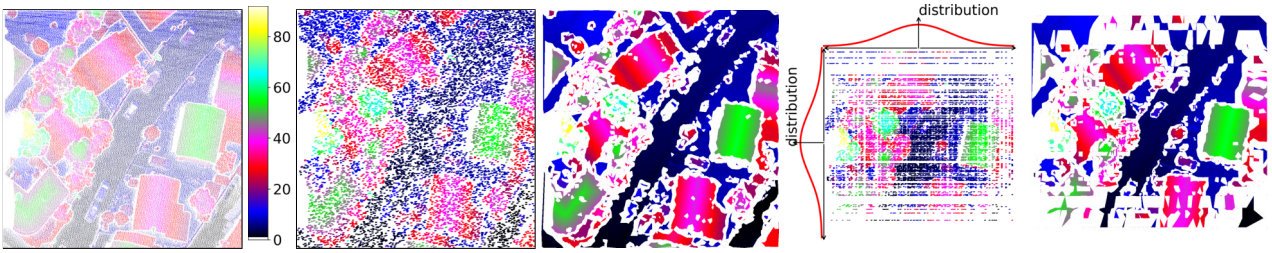


Figure 6. Influence of LiDAR density on 1-pixel error on DublinCity. The behavior is the same as Toulouse2020, even though the density is different.

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



(a) Ground truth (b) Random subsampling (c) Random subsampling w/ TIN interpolation (d) Gaussian subsamping (e) Gaussian subsamping w/ TIN interpolation

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%.

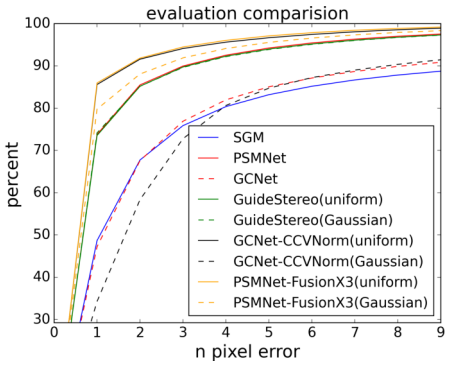


Figure 8. The influence of the distribution of LiDAR on Toulouse2020. GCNet-CCVNorm is highly influenced by the distribution of LiDAR. GuideStereo and PSMNet-FusionX3 are less sensitive, and the imbalanced result is much worse than the randomly sampled result.

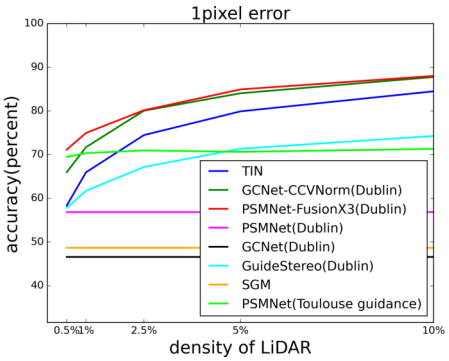


Figure 10. Training on DublinCity and testing on Toulouse2020. Learning-based methods except for GCNet are still better than SGM, PSMNet learning on sparse ground truth work well, for low density guidance(0.5%), GCNet-CCVNorm is worse than PSMNet training on guidance, but PSMNet-FusionX3 is better.

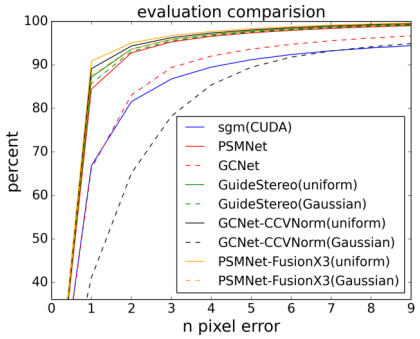


Figure 9. The influence of the distribution of LiDAR on DublinCity. The behavior is the same as Toulouse2020, GCNet-CCVNorm is highly influenced by the distribution, and PSMNet-FusionX3 is less sensitive compared to Toulouse2020.

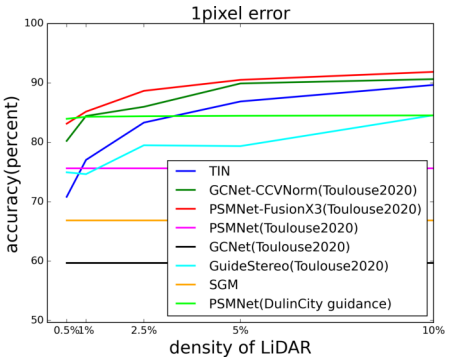
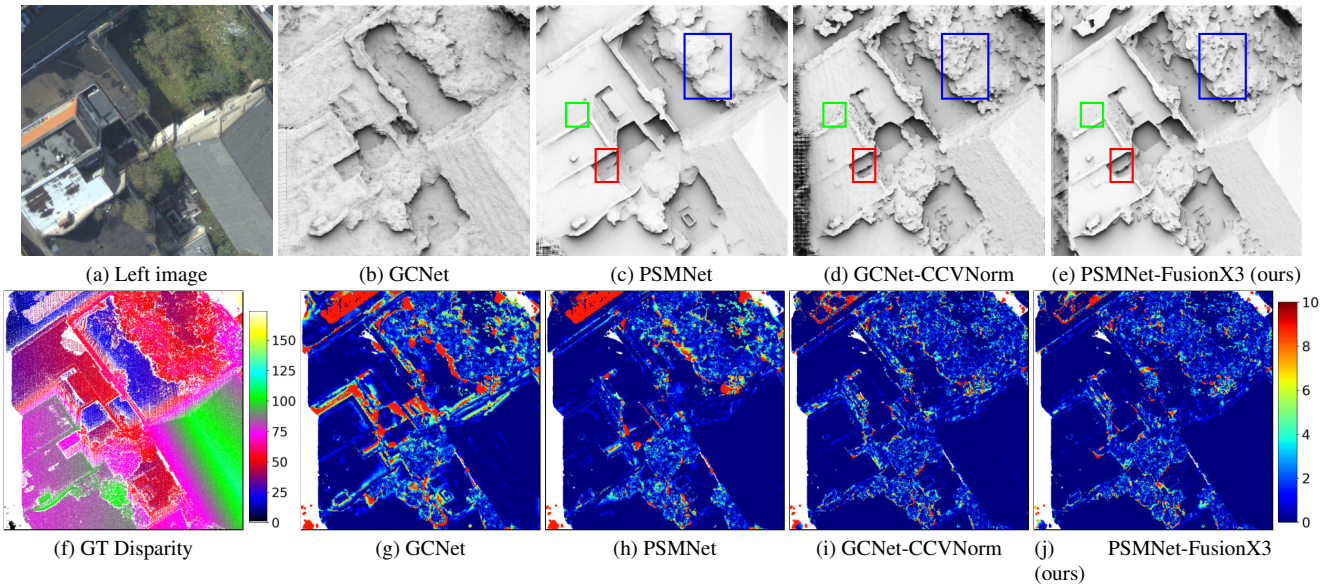


Figure 11. Training on Toulouse2020 and testing on DublinCity. The behavior is the same as Toulouse2020, when the ratio is 0.5%, PSMNet-FusionX3 is slightly worse than PSMNet training on the LiDAR guidance.

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



(a) Left image (b) GCNet (c) PSMNet (d) GCNet-CCVNorm (e) PSMNet-FusionX3 (ours)

(f) GT Disparity (g) GCNet (h) PSMNet (i) GCNet-CCVNorm (j) PSMNet-FusionX3 (ours)

Figure 12. Shaded depth map (first row) and error map (second row) in pixel on DublinCity. The shading is done using the GrShade tool from MicMac [26]. Blue rectangle: high vegetation. Red rectangle: depth discontinuity. Green rectangle: uniform area.

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

[1] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. *arXiv preprint arXiv: 1909.01300*, 2019.

[2] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020.

[3] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris, 2020.

[4] Konstantinos Batsos, Changjiang Cai, and Philippos Mordohai. Cbmv: A coalesced bidirectional matching volume for disparity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2060–2069, 2018.

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

[6] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33:22158–22169, 2020.

[7] Jaesung Choe, Kyungdon Joo, Tooba Imtiaz, and In So Kweon. Volumetric propagation network: Stereo-lidar fusion for long-range depth estimation. *IEEE Robotics and Automation Letters*, 6(3):4672–4679, 2021.

[8] 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 International Conference on Computer Vision*, pages 4384–4393, 2019.

[9] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.

[10] Lei Fan, Joel Smethurst, Peter Atkinson, and William Powrie. Propagation of vertical and horizontal source data errors into a tin with linear interpolation. *International journal of geographical information science*, 28(7):1378–1400, 2014.

[11] Daniel Hernandez-Juarez, Alejandro Chacón, Antonio Espinosa, David Vázquez, Juan Carlos Moure, and Antonio M López. Embedded real-time stereo estimation via semi-global matching on the gpu. *Procedia Computer Science*, 80:143–153, 2016.

[12] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005.

[13] Xu Huang, Rongjun Qin, Changlin Xiao, and Xiaohu Lu. Super resolution of laser range data based on image-guided fusion and dense matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:105–118, 2018.

[14] 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*, pages 16706–16716, 2021.

[15] Yu-Kai Huang, Yueh-Cheng Liu, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. Expanding sparse guidance for stereo matching. *arXiv preprint arXiv:2005.02123*, 2020.

[16] 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 International Conference on Computer Vision*, pages 66–75, 2017.

[17] 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.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Patrick Knöbelreiter and Thomas Pock. Learned collaborative stereo refinement. *International Journal of Computer Vision*, pages 1–18, 2021.

[20] Debra F Laefer, Saleh Abuwarda, Anh-Vu Vo, Linh Truong-Hong, and Hamid Gharibi. 2015 aerial laser and photogrammetry survey of dublin city collection record. 2017.

[21] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *arXiv preprint arXiv:2006.02535*, 2020.

[22] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[23] 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 conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[24] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[25] Kihong Park, Seungryong Kim, and Kwanghoon Sohn. High-precision depth estimation with the 3d lidar and stereo fusion. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2156–2163. IEEE, 2018.

[26] Marc Pierrot-Deseilligny, D Jouin, J Belvaux, G Maillet, L Girod, E Rupnik, J Muller, M Daakir, G Choqueux, and M Deveau. Micmac, apero, pastis and other beverages in a nutshell. *Institut Géographique National*, 2014.

[27] Andrea Pilzer, Yuxin Hou, Niki Loppi, Arno Solin, and Juho Kannala. Expansion of visual hints for improved generalization in stereo matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5840–5849, 2023.

[28] 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*, pages 979–988, 2019.

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[30] Akihito Seki and Marc Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 231–240, 2017.

[31] Jonathan Richard Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Applied Computational Geometry Towards Geometric Engineering: FCRC'96 Workshop, WACG'96 Philadelphia, PA, May 27–28, 1996 Selected Papers*, pages 203–222. Springer, 2005.

[32] Federico Tombari, Stefano Mattoccia, Luigi Di Stefano, and Elisa Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[33] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

[34] Tsun-Hsuan Wang, Hou-Ning Hu, Chieh Hubert Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 3d lidar and stereo fusion using stereo matching network with conditional cost volume normalization. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5895–5902. IEEE, 2019.

[35] T Wu, B Vallet, M Pierrot-Deseilligny, and E Rupnik. A new stereo dense matching benchmark dataset for deep learning. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:405–412, 2021.

[36] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019.

[37] 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. *arXiv preprint arXiv:1906.06310*, 2019.

[38] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *The journal of machine learning research*, 17(1):2287–2318, 2016.

[39] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019.

[40] Junming Zhang, Manikandasriram Srinivasan Ramanagopal, Ram Vasudevan, and Matthew Johnson-Roberson. Listereo: Generate dense depth maps from lidar and stereo imagery. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7829–7836. IEEE, 2020.

[41] Yongjun Zhang, Siyuan Zou, Xinyi Liu, Xu Huang, Yi Wan, and Yongxiang Yao. Lidar-guided stereo matching with a spatial consistency constraint. *ISPRS Journal of Photogrammetry and Remote Sensing*, 183:164–177, 2022.