Published in Image Processing On Line on 2022–11–14.
Submitted on 2022–10–03, accepted on 2022–10–03.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.435

Disparity Estimation Networks for Aerial and High-Resolution Satellite Images: A Review

Roger Marí, Thibaud Ehret, Gabriele Facciolo

Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, Gif-sur-Yvette, France
({roger.mari, thibaud.ehret, gabriele.facciolo}@ens-paris-saclay.fr)

Communicated by Jean-Michel Morel *Demo edited by* Roger Marí

Abstract

This study reviews the PSM and HSM deep learning architectures for disparity estimation from an input stereo pair and assesses their applicability for satellite stereo reconstruction. All methods are tested on urban landscapes unseen at training time, using pre-trained weights learned from a stereo matching benchmark for aerial imagery. The quality of the disparity maps output by each method is assessed based on the subsequent surface models, which are evaluated using a lidar reference model. The conducted experiments give insight into the robustness of each architecture (e.g. robustness to different input resolutions, color spaces or acquisition dates), as well as their generalizability across different cities. Lastly, the results obtained with the different networks are compared with those of a state-of-the-art variant of the semi-global matching algorithm, which is a well-known classic methodology for satellite dense stereo matching.

Source Code

The source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the README.md file of the archive. The original implementation of the methods is available here[2] and here[3].
This is an MLBriefs article, the source code has not been reviewed!

Keywords: stereo matching; disparity estimation; deep learning; satellite images; aerial images

[1] https://doi.org/10.5201/ipol.2022.435
[2] https://github.com/JiaRenChang/PSMNet
[3] https://github.com/gengshan-y/high-res-stereo

1 Introduction

The concept of disparity refers to the horizontal displacement d between corresponding pixels of two images that observe the same scene from different viewpoints. Estimating disparity from stereo, i.e., knowing that a pixel (x, y) in the left image corresponds to a pixel $(x - d, y)$ in the right image, is a classic and well-known problem in computer vision. The disparity values that match a stereo pair of images are valuable information because they are inversely proportional to the depth of the scene. Assuming a simple pinhole camera model, the depth z of the point that corresponds to the 3D point denoted $\mathbf{x}$ and observed by the pixels (x, y) and $(x - d, y)$ is equal to

$$z = \frac{f \cdot B}{d}, \tag{1}$$

where d is the disparity, f is the focal length of the camera and B is the baseline length, corresponding to the segment between the two camera centers. The depth z in (1) represents the depth of $\mathbf{x}$ with respect to the baseline of the system.

In this paper we review some deep learning networks for disparity estimation from an input stereo pair. We focus on the Pyramid Stereo Matching (PSM) and Hierarchical Stereo Matching (HSM) architectures [4, 33]. In particular, the objective is to evaluate the suitability of such methods for high-resolution remote sensing images. Figure 1 shows an example of the input and output data.

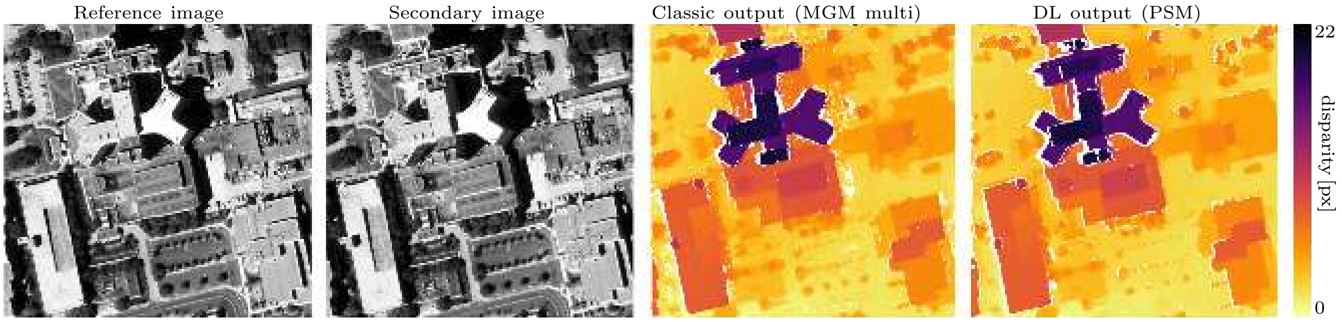


Figure 1: Example of input stereo pair [3] and the resulting disparity maps, in pixel units, obtained with a classic matching algorithm [11] and a deep learning (DL) network [4].

The majority of satellite stereo pipelines employ classic methodologies to construct disparity maps, which could be potentially replaced by a neural network. However, the lack of public benchmarks makes direct comparisons difficult [15, 32]. Most deep architectures are originally conceived and trained for synthetic or street-level scenes [14, 24, 23], which raises questions about their performance in other fields, such as satellite imagery. It seems unfair to draw conclusions using networks that have never seen anything resembling a satellite image. For this purpose, we employ pre-trained weights that have been fine-tuned using a stereo matching benchmark for aerial imagery [32]. Our choice is motivated by the fact that aerial images can be understood as fragments of very high resolution satellite images, with a great similarity from the semantic point of view. Previous work has already demonstrated the advantages of using fine-tuned weights to work with remote sensing images [15, 32].

To assess the methods, we replace the matching algorithm of the satellite stereo pipeline S2P [6] using each of them. Then, using the same rectified pairs and camera models, S2P is used to reconstruct multiple surface models, which are evaluated using a lidar-acquired ground truth. Since the only element that changes is the disparity map used to extract each surface model, the altitude errors are directly indicative of disparity accuracy.

2 Related Work

The combination of global and local information is a widely studied subject in the field of disparity estimation. Both classic and deep learning methods have proposed various strategies to address this key issue. For each branch, this section reviews the most relevant algorithms related to this work.

2.1 Classic Stereo Matching

Local methods compute a matching cost (e.g. sum of absolute differences, normalized cross correlation, census) between a window centered on a pixel of the reference image and an equivalent window centered on some pixel of the secondary image [30, 13]. Epipolar geometry is used to reduce the amount of matching candidates [16]. These methods are known to fail on ill-posed regions containing repetitive patterns, untextured regions or reflective surfaces.

Global methods overcome the limitations of local methods by approaching stereo matching as an energy minimization problem that adds a regularization term to the matching cost. The idea behind the regularization term is that neighbor pixels of the same object should have similar disparities. Most global matching energies take the generic form

$$E(d) = \sum_{\mathbf{p}\in I} C(d_{\mathbf{p}}) + \sum_{(\mathbf{p},\mathbf{q})\in\xi} V(d_{\mathbf{p}}, d_{\mathbf{q}}), \tag{2}$$

where $C(d_{\mathbf{p}})$ is the local matching cost of assigning disparity $d_{\mathbf{p}}$ to pixel $\mathbf{p}$ and $V(d_{\mathbf{p}}, d_{\mathbf{q}})$ is the regularization term enforcing that $d_{\mathbf{p}}$ should be similar to $d_{\mathbf{q}}$, where $\mathbf{q}$ is a neighbor pixel of $\mathbf{p}$. The domain I comprises all nodes (or pixel coordinates) and ξ is the edge set pointing to the neighbor pixels taken into account. Usually, the graph $G = (I, \xi)$ is 4-connected or 8-connected.

The Semi-Global Matching (SGM) algorithm [18] is a popular choice for satellite imagery [2, 6, 5]. It computes an approximate solution to the NP-hard problem (2) using a regularization term equal to

$$V(d, d') = \begin{cases} 0 & \text{if } d = d', \\ P1 & \text{if } |d - d'| = 1, \\ P2 & \text{otherwise.} \end{cases} \tag{3}$$

The regularization term (3) considers three different categories. A small penalty $P1$ is imposed for small disparity differences (up to 1 pixel), which are common on slanted surfaces. A larger penalty $P2$ (with $P2 > P1$) is given to stronger disparity discontinuities. Finally, there is no penalty if neighbor disparities d and d' are the same. Such categories are particularly suitable for terrestrial surface modeling, which consists mostly of flat or slanted terrain and roofs, with a minority of large discontinuities (e.g. cliffs or building boundaries).

The strategy adopted by SGM consists in dividing the original 2D problem into multiple 1D problems defined on scan lines, which are straight lines that run through the image in the 4 or 8 cardinal directions. Each scan line can be processed as an independent process, allowing for parallelization and high speed computation. For each direction $\mathbf{r}$, a cost volume $L_{\mathbf{r}}$ is computed recursively starting from the image borders. The cost $L_{\mathbf{r}}(\mathbf{p}, d)$ at pixel $\mathbf{p}$ along direction $\mathbf{r}$ at disparity level d is

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min \begin{pmatrix} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P1, \\ L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P1, \\ \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P2 \end{pmatrix}, \tag{4}$$

where

- $L_{\mathbf{r}}(\mathbf{p}, d)$ is the cost of assigning disparity d to pixel $\mathbf{p}$ following direction $\mathbf{r}$.
- $C(\mathbf{p}, d)$ is the matching cost of assigning disparity d to pixel $\mathbf{p}$.
- $L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d)$ is the previous cost in $\mathbf{r}$ direction at disparity d.
- $L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d-1)$ is the previous cost in $\mathbf{r}$ direction at disparity $d-1$.
- $L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d+1)$ is the previous cost in $\mathbf{r}$ direction at disparity $d+1$.
- $\min_i L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, i)$ is the previous minimum cost in $\mathbf{r}$ direction, at any disparity level.

By combining (3) and (4), the cost (4) can be summarized as

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min_{d' \in D} \left(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d') + V(d, d') \right). \tag{5}$$

The different costs (5) computed in each direction $\mathbf{r}$ are added to obtain a single cost volume,

$$C_{cost} = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d), \tag{6}$$

and the final disparity for each pixel is selected using a Winner-Takes-All (WTA) evaluation [13] of $C_{cost}(\mathbf{p}, \cdot)$, i.e. the disparity d with minimum cost is taken for each pixel $\mathbf{p}$.

Multiple variants have been proposed to further improve the performance of SGM [28, 11, 1]. In this work, the MGM variant is taken as a reference for classic stereo matching. MGM or More Global Matching [11] improves SGM by injecting information from the perpendicular direction $\mathbf{r}^{\perp}$ to each cost along the direction $\mathbf{r}$. In particular, MGM modifies (5) as

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \sum_{\mathbf{x} \in (\mathbf{r}, \mathbf{r}^{\perp})} \frac{1}{2} \min_{d' \in D} \left(L_{\mathbf{r}}(\mathbf{p}-\mathbf{x}, d') + V(d, d') \right). \tag{7}$$

Expression (7) preserves its recursive nature and requires only minor adjustments in the parallelization process [11]. The difference is that the cost $L_{\mathbf{r}}(\mathbf{p}, d)$ not only considers the preceding points in a single scan line, but also uses the points of the preceding scan line (i.e. the pixel above). Such strategy improves the predicted disparities and prevents streaking artifacts, which are characteristic of SGM due to the 1D nature of the method.

2.2 Deep Stereo Matching

Like classic methods, deep learning architectures for disparity estimation also aim to obtain a good compromise between local and global matching costs. Feature extraction and the construction of cost volumes also constitute the usual steps in this branch of methods [21].

Convolutional neural networks (CNN) with encoder-decoder architectures, which had already proven successful in aggregating coarse-to-fine features for semantic segmentation (e.g. UNet), served as inspiration for the first end-to-end models for disparity regression. For example, DispNet [23] and CRL [27] reused hierarchical information by concatenating features from the encoder layers with those from the decoder layers. In these earlier models, the features extracted from the left and right image of an input pair were fused in the first layers of the contracting path, by building a correlation volume. Given some left and right feature maps, $\mathbf{f}_L$ and $\mathbf{f}_R$, the correlation volume C_{corr} is commonly computed as the normalized inner product at each disparity level d

$$C_{corr}(x, y, d) = \frac{1}{F} \langle \mathbf{f}_L(x-d, y), \mathbf{f}_R(x, y) \rangle, \tag{8}$$

where F is the number of channels in the feature maps $\mathbf{f}_L$ and $\mathbf{f}_R$. The operation is done for each 2D position (x, y), resulting in an output volume of size $H \times W \times D$ where H and W are the height and width of the feature maps, and D is the disparity range. The resulting correlation volume can be forwarded as another feature map of D channels. The DispNet authors demonstrated that such strategy outperforms directly feeding the CNN with a stack containing both input images [23].

GC-Net [20] proposed a different strategy to merge the information from the input pair, by building a cost volume of features learned using a series of residual blocks [17]. Given the feature maps $\mathbf{f}_L$ and $\mathbf{f}_R$, the cost volume C_{cost} is built by concatenating the feature vectors at each disparity level d

$$C_{cost}(x, y, d) = concat\left(\mathbf{f}_L(x-d, y), \mathbf{f}_R(x, y)\right), \tag{9}$$

where *concat* is the concatenation operation. Following the notation in (8) this strategy produces a 4D volume with size $H \times W \times D \times 2F$. The feature dimension F is preserved in this way, allowing the network to exploit contextual information in the later stages. The GC-Net then uses a 3D convolution encoder-decoder structure to regularize the cost volume at multiple scales, followed by a differentiable soft argmin operation to predict the disparity values.

GC-Net inspired the PSM [4] and HSM [33] networks that this article reviews in detail. Such models brought significant accuracy improvements by explicitly employing multi-scale features to construct the cost volume C_{cost}. Subsequent deep stereo matching models have also addressed other interesting issues. For instance, DeepPruner [9] does not require a predefined disparity range to search for matches. The architecture is similar to that of PSM [4] but a differentiable PatchMatch algorithm is introduced to obtain a sparse cost volume, where the disparity search range is learned and adapted to each pixel. Pruning unlikely disparities provides a significant gain in efficiency. An image guided refinement module is also added to reduce noise and improve sharp boundaries: the disparity map predicted after cost regularization is coupled with features learned from the reference view, and used to feed a lightweight CNN that refines the disparity values.

GA-Net [35, 15] is another interesting method. It replaces some of the 3D convolutional layers used for cost regularization, which are computationally costly and memory-consuming, by using a Semi-global Guided Aggregation layer (SGA) which is a differentiable approximation of SGM, i.e. (5). The SGA layers are followed by a Local Guided Aggregation layer (LGA) to refine thin structures and object edges. The LGA filtering uses the cost values of adjacent disparity levels $(d-1, d, d+1)$ in a $K \times K$ spatial window to refine the cost of disparity d at the central point of the window.

It is common in end-to-end models for disparity and depth estimation [4, 33, 34] that the final predicted values are regressed by means of a differentiable soft argmin operation, as originally proposed in GC-Net [20]. In the last layer of such networks, the regularized costs C_{cost} are compressed from a 4D volume to a 3D volume by means of a single-channel 3D convolutional layer that reduces the feature dimension to a single value. The depth of the 3D volume C_{cost} is then equal to D, the disparity range. The predicted disparity $\hat{d}$ at a 2D point (x, y) is obtained as

$$\hat{d}(x, y) = \sum_{d=0}^{D} d \times softmax(-C_{cost}(x, y, d)). \tag{10}$$

All possible disparity levels $d \in D$ contribute to the prediction (10) according to a weight equal to the *softmax* function applied to the negative cost at position (x, y, d), to give higher weights to lower costs. Each weight represents the normalized probability of the corresponding disparity level.

3 Methodology

This work reviews Pyramid Stereo Matching (PSM) [4] and Hierarchical Stereo Matching (HSM) [33], two deep learning architectures for disparity estimation from an input rectified stereo pair of images.

Both models follow a structure that can be divided into a first part dedicated to feature extraction and a second part dedicated to the regularization of a cost volume constructed with the previously extracted features. Disparity values are then regressed based on the regularized costs as in (10).

3.1 Pyramid Stereo Matching network (PSM)

The PSM network [4] architecture is summarized in Figure 2(a). This model is inspired by GC-Net [20] but adds a major ingredient: a Spatial Pyramid Pooling (SPP) module at the end of the feature extraction path to further exploit the global information. This idea was motivated by the effectiveness of the SPP modules, originally introduced for semantic segmentation [36], to expand the receptive field and capture context information.

Feature extraction. Illustrated in Figure 2(b), this part consists of a CNN followed by a SPP module. The CNN is a contracting path consisting of 2D convolutional layers and a series of residual blocks. Downsampling takes place due to the use of a stride of 2 in certain layers. Some residual blocks use dilated convolutions to help increase the receptive field.

Given an input feature map $\mathbf{f}_i$, the SPP module of PSM applies an average pooling operation with four different kernel sizes, to explicitly generate features at different spatial scales. The multi-scale feature maps then undergo a 1×1 convolutional layer to compress the feature dimension, followed by an upsampling step employing bilinear interpolation, in such a way that their final height and width is the same as that of the input $\mathbf{f}_i$. The different levels of feature maps output by the SPP module are concatenated with $\mathbf{f}_i$ and fused using further convolutional layers (fusion block).

Cost regularization. Feature maps learned by the feature extraction part are used to build a 4D cost volume as explained in Section 2. This is done by concatenating the feature vectors of the two images at each disparity level, as in (9). The cost volume is regularized using a stacked hourglass architecture, detailed in Figure 2(c), that consists of a chain of three encoder-decoder modules of 3D convolutional and deconvolution layers. Each hourglass or encoder-decoder module generates a disparity map that contributes to the loss function, in what is referred to as intermediate supervision. The highest resolution is achieved in the last output, i.e. the third disparity map.

Stacked hourglass architectures with intermediate supervision exploit the possibility to reevaluate initial estimates [25]. The idea is to give multiple opportunities to the network to produce coherent results at both local and global contexts. For example, if the first hourglass module focuses on very local neighborhoods, subsequent modules will explore higher order spatial relationships.

Loss function. The PSM network is supervised using a *smooth* L_1 loss function, chosen for its higher robustness to outliers with respect to the L_2 loss.

$$L(d_{\mathrm{GT}}, \hat{d}) = \frac{1}{N}\sum_{i=1}^{N} smoothL_1(d_{\mathrm{GT}} - \hat{d}), \quad \text{where} \quad smoothL_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{11}$$

where N is the number of pixels and d_{GT} and $\hat{d}$ are the ground truth and predicted disparities, respectively. Following the strategy of intermediate supervision of the stacked hourglass architecture, the loss terms obtained at each of the three hourglass modules are added to compute the final cost at each training iteration (12). The weight of each term is fixed, with increasing value according to the number of hourglass modules already covered

$$L_{\mathrm{PSM}} = 0.5L_1 + 0.7L_2 + L_3, \tag{12}$$

where each L_i takes the form of (11) and the subscript $i = \{1, 2, 3\}$ refers to the index of the hourglass modules in Figure 2(c).

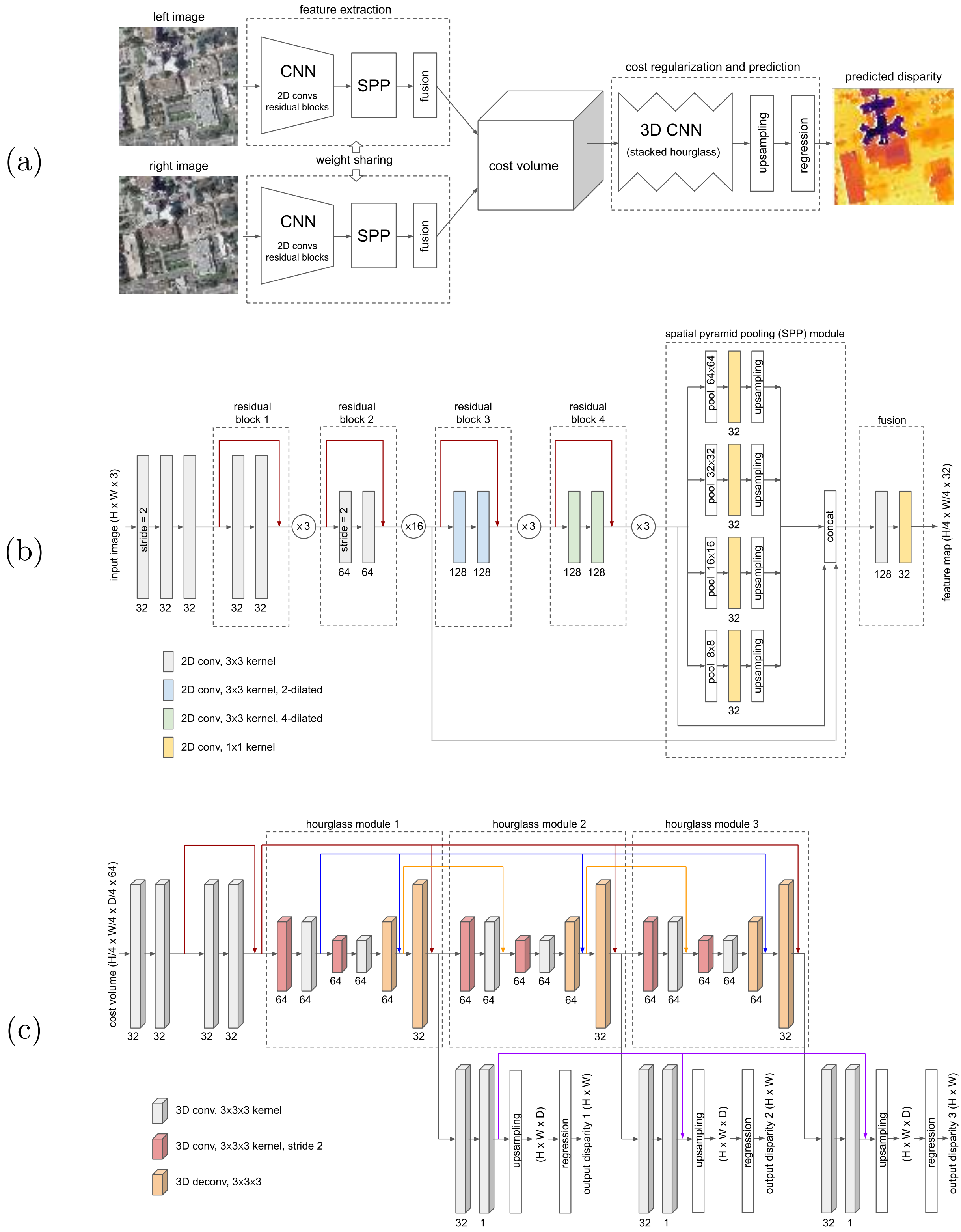


Figure 2: PSM network: (a) Overview. (b) Feature extraction path. The number of channels of each convolutional layer is shown below its rectangle. Circles indicate the number of residual blocks: e.g. residual block 1 is repeated 3 times. Bilinear interpolation is used for upsampling. The 2D conv layers are used with batch normalization, to gain stability, and ReLU activation to introduce non-linearities. Colored arrows represent additive skip connections. (c) Cost volume regularization path. This 3D CNN consists of three hourglass modules of 3D conv layers, which are used to aggregate the feature information along the disparity dimension. Trilinear interpolation is used for upsampling. Colored arrows represent additive skip connections. Each hourglass module predicts a disparity map, with increasing detail definition.

3.2 Hierarchical Stereo Matching Network (HSM)

After the breakthrough of the GC-Net and PSM architectures, HSM [33] was developed with the purpose of gaining efficiency and accuracy when handling high-resolution image pairs, with a larger input image size. The HSM architecture is summarized in Figure 3(a).

Feature extraction. Like PSM, the feature extraction part of HSM starts with a CNN followed by a SPP module. The CNN also consists of an encoder structure of 2D convolutional layers and residual blocks. However, the number of residual blocks is significantly decreased with respect to PSM and max pooling and convolutions with a stride of 2 are used to further compress the spatial scale of the feature maps. Similarly to PSM, the SPP module of HSM applies average pooling with four different kernel sizes, but resulting features are merged by addition.

Instead of directly using the SPP output, denoted $\mathbf{f}_{SPP}$, to construct the cost volume, HSM reprocesses $\mathbf{f}_{SPP}$ by means of a decoder structure. This is shown in Figure 3(b). In particular, the decoder fuses features before and after SPP by concatenation and gradually upsamples the result using up-convolutions (convolution after upsampling), to produce coarse-to-fine feature maps that reach higher spatial resolutions. Four feature maps are obtained in the end, $\{\mathbf{f}^{(k)}_{SPP}\}$, where $k = \{0, 2, 4, 8\}$ is the upsampling factor with respect to the features provided by SPP. The four final feature maps $\{\mathbf{f}^{(k)}_{SPP}\}$ are compressed to 32 or 16 channels using 1×1 convolutional layers.

Cost regularization. Instead of building a single cost volume, HSM builds a multi-scale pyramid of four cost volumes using the four multi-scale $\{\mathbf{f}^{(k)}_{SPP}\}$ feature maps produced in the feature extraction part (see Figure 3(c)). Each cost volume of the pyramid has increasing spatial and disparity resolution. To control the size of the cost volumes, these are constructed in a different way with respect to PSM and GC-Net. Instead of concatenating the left $\mathbf{f}_L$ and right $\mathbf{f}_R$ features, as in (9), each C_{cost} volume is constructed using absolute differences

$$C_{cost}(x, y, d) = |\mathbf{f}_L(x - d, y) - \mathbf{f}_R(x, y)|. \tag{13}$$

The set of multi-scale cost volumes is then regularized using a chain of 4 decoders, each devoted to one of the volumes, as shown in Figure 3(c). The decoders employ 3D convolutional layers and trilinear upsampling. Larger scale decoders take as input their corresponding cost volume concatenated with the filtered costs provided by the previous decoder. The first two decoders also contain a Volumetric Pyramid Pooling (VPP), which is the exact equivalent of the SPP module, but extended for 4D feature volumes, i.e. using 3D convolutional layers and 3D kernels for average pooling.

Four different disparity maps are regressed using the regularized costs output by the chain of decoders. The highest resolution is achieved in the last output, i.e. the fourth disparity map.

Loss function. The HSM network is supervised in a very similar way to the PSM architecture, i.e. using a sum of smooth L_1 losses between the predicted and ground truth disparities. The weights assigned to the contribution of each disparity map increase exponentially according to the spatial resolution of the corresponding input cost volume

$$L_{\text{HSM}} = \frac{1}{2^6} L_1 + \frac{1}{2^4} L_2 + \frac{1}{2^2} L_3 + L_4, \tag{14}$$

where each L_i takes the form of (11) and the subscript $i = \{1, 2, 3, 4\}$ refers to the index of the decoder modules in Figure 3(c).

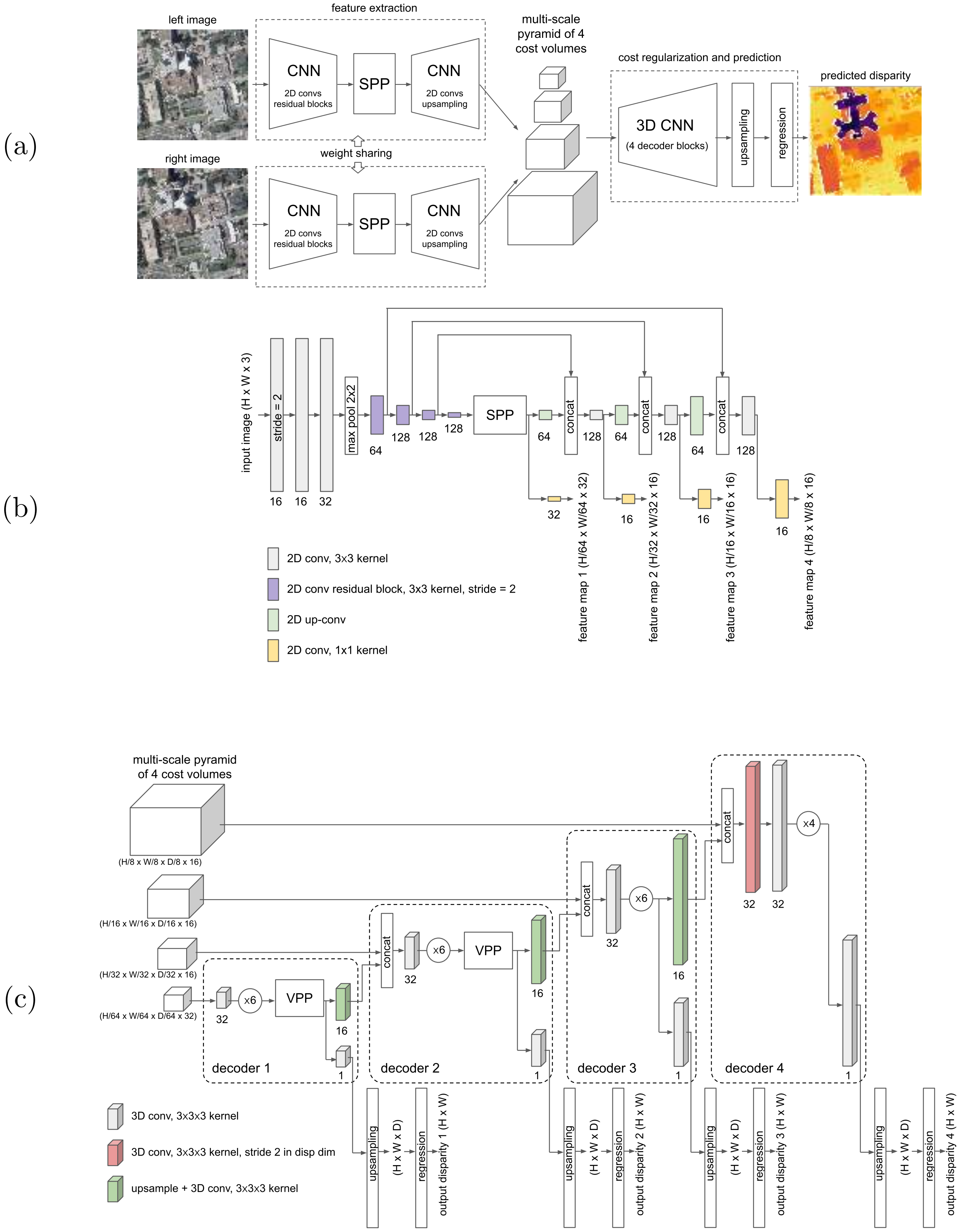


Figure 3: HSM network: (a) Overview. (b) Feature extraction path. The number of channels of each convolutional layer is shown below its rectangle. Bilinear interpolation is used for upsampling. The 2D conv layers are used with batch normalization, to gain stability, and ReLU activation to introduce non-linearities. (c) Cost volume regularization path. A chain of decoders, consisting of 3D convs and Volumetric Pyramid Pooling (VPP) blocks, are used to regularize a multi-scale pyramid of cost volumes. Trilinear interpolation is used for upsampling. Circles indicate how many times the preceding layer is repeated.

4 Experiments

The objective of this work is to evaluate the performance of PSM and HSM in the context of satellite imagery. We focus on urban areas, observed at viewing angles not far from nadir. Instead of training the deep learning models from scratch, we use pre-trained weights provided by the 2021 open-source stereo matching benchmark for aerial imagery [32]. This ensures that the networks are already familiar with the usual elements of Earth observation images: building roofs, tilted facades, roads, trees, etc. Our experiments aim to assess the generalizability of the models and establish which architecture offers higher robustness to different input specificities.

4.1 Aerial Stereo Matching Benchmark

The aerial stereo matching benchmark introduced in [32] was built using the ISPRS Vaihingen dataset for urban classification and 3D building reconstruction [12, 29].

The ISPRS Vaihingen dataset consists of 20 large-scale aerial images taken with an Intergraph/ZI digital mapping camera DMC [29]. The aerial images are 16 bit pansharpened color-infrared[4] (CIR) images with a resolution of about 8 cm per pixel. A lidar point cloud of the Vaihingen area (Germany) is also available, with a point density varying between 4 and 7 points/m^2.

The 2021 aerial stereo matching benchmark [32] provides 1092 pairs of 1024×1024 pixels, cropped from the CIR images and rectified using the MicMac library [28]. Examples are shown in Figure 4. The corresponding ground truth disparity maps were generated from the lidar point cloud and are stored on 16 bits with the disparity value scaled by 256. Since the lidar data is very sparse, a density-based filter was used to account for occlusions. The authors of the benchmark used 585 training pairs to fine tune the PSM and HSM networks, originally trained on the KITTI dataset [14, 24].

4.2 Altitude-based Evaluation using Satellite Images

For our experiments, we selected four areas of interest (AOIs) of 256×256 m from the *2019 IEEE GRSS Data Fusion Contest* (DFC2019) [3], shown in Figure 12. The DFC2019 dataset provides, among others, 26 WorldView-3 images, with a resolution of about 30 cm per pixel, acquired between 2014 and 2016 over the city of Jacksonville (Florida, US). We take image crops of varying size, around 800×800 pixels, covering each target AOI. The resulting images are used to form stereo pairs.

We select suitable stereo pairs according to the criterion of [10], with the objective of maximizing the accuracy of output disparities. The selection criterion prioritizes pairs with an angle between views from 5 to 45 degrees and a maximum incidence angle of 40 degrees for each view. From this set, we take the 30 pairs with closest acquisition dates and use them as input. The lists of pairs selected for each AOI, as well as some example views, are included in the Appendix A.

As a substitute for ground truth disparities, our assessment is based on a digital surface model (DSM) of each AOI, derived from lidar and part of the DFC2019 data. The resolution of the lidar DSMs is 0.5 m per pixel. To convert disparity values to altitude, we take the satellite stereo reconstruction pipeline S2P [6] and replace the matching algorithm with each deep learning model. The S2P tools are also used to rectify each input pair of images.

The altitude errors resulting from the disparity maps provided by the deep learning methods are compared with those achieved using the S2P baseline classic matching algorithms, i.e. MGM and an improved multi-scale version of MGM, both using the census transform [11, 10, 13].

[4] The blue channel of each image in the dataset has been replaced by the response of an infrared camera.

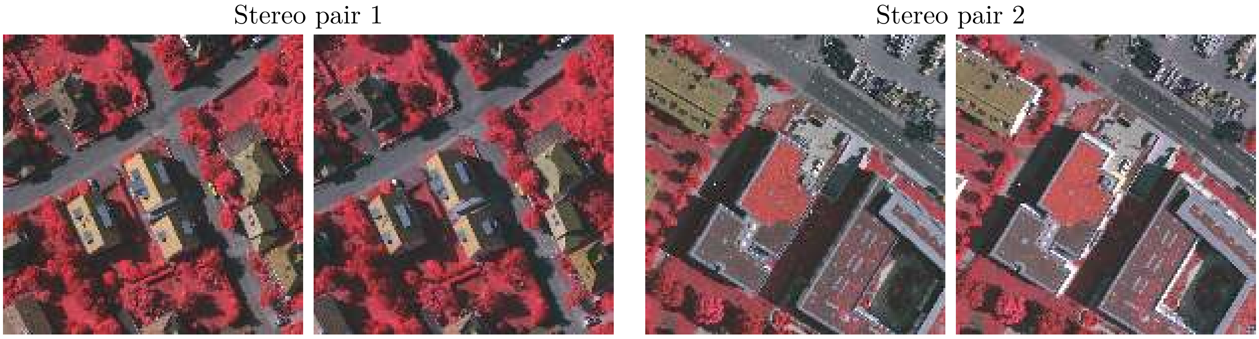


Figure 4: Example stereo pairs from the 2021 aerial stereo benchmark [32], used to fine tune the PSM and HSM networks.

4.3 Evaluation Metrics

The evaluation metrics are the following:

- **MAE.** Mean Absolute Error, in meters of altitude, between a photogrammetric DSM and the ground truth lidar DSM.
- **Completeness.** Percentage of non-water points in a photogrammetric DSM where error is less than 1 meter with respect to the lidar DSM, with undefined values counted as larger errors.
- **NaNs.** Percentage of undefined values (Not a Number) in a photogrammetric DSM.
- **Successful pairs.** Number of stereo pairs, out of N originally given as input to a matching algorithm, that resulted in DSMs with less than 50% of undefined values.

4.4 Results

Using S2P, we compute (1) single-pair DSMs, i.e. the set of DSMs that result from each independent stereo pair of the 30 selected; and (2) multi-pair DSMs, which are denser and result from fusing all the successful single-pair DSMs using a median filter as in [22]. The camera models given to S2P have been previously bundle adjusted [22]. We distinguish between two types of experiments, according to the color space of the input images. Other subsections are devoted to complementary experiments in which we study the performance of each method as a function of the distance between the acquisition dates and the baseline of the input pairs.

4.4.1 Panchromatic Inputs

In these experiments, we use the panchromatic version of the DFC2019 images as input. Panchromatic images have a single channel with a wide range of intensity values, which allows them to be highly textured. This makes them well suited for classic matching algorithms. The numerical results are reported in Table 1. Qualitative results are shown in Figure 5.

Table 1 introduces some custom parameters, s and lr, which we observe to significantly affect the accuracy of the disparity maps generated with PSM and HSM (based on the subsequent altitude values). Otherwise, the default parameters of S2P are used, with the exception of the SIFT matching threshold, which is set to 0.5 to aim for very reliable matches for the rectification step.

The following paragraphs discuss some of the main ideas reflected in Table 1.

On the left-right consistency check. The left-right consistency check is a good practice to refine disparity maps: it filters disparity values that are not consistent when the left and right images of the stereo pair are exchanged [18]. This step uses a consistency distance threshold lr, which strongly affects the percentage of NaN values in the disparity map and the subsequent DSM. Larger lr produces more complete DSMs in exchange of small inaccuracies.

Panchromatic inputs

		Single pair MAE [m] / NaNs [%] / Successful pairs			
Area index		004	068	214	260
MGM	$s = 1, lr = 1$	1.863 / 32.01 / 24	0.919 / 23.63 / 29	1.457 / 31.23 / 24	1.668 / 33.30 / 24
MGM multi	$s = 1, lr = 1$	1.531 / 34.63 / 18	0.886 / 25.74 / 28	1.288 / 31.95 / 23	1.663 / 34.48 / 21
PSM	$s = 1, lr = 1$	0.933 / 33.17 / 12	0.645 / 25.00 / 28	0.985 / 34.23 / 23	1.279 / 33.94 / 16
HSM	$s = 1, lr = 1$	1.545 / 34.64 / 20	0.838 / 28.90 / 28	1.229 / 34.89 / 23	1.580 / 31.72 / 24
PSM	$s = 2, lr = 2$	0.886 / 38.61 / 12	0.511 / 28.34 / 28	0.809 / 36.88 / 23	1.002 / 38.33 / 14
HSM	$s = 2, lr = 2$	1.110 / 37.63 / 16	0.605 / 28.09 / 28	0.911 / 35.40 / 23	1.189 / 32.98 / 19
PSM	$s = 3, lr = 3$	0.620 / 37.87 / 12	0.755 / 30.33 / 26	1.051 / 36.89 / 18	1.006 / 38.37 / 11
HSM	$s = 3, lr = 3$	0.885 / 37.15 / 12	0.635 / 28.88 / 28	0.972 / 36.14 / 23	1.163 / 34.49 / 18

		Multi-pair MAE [m] / NaNs [%]			
Area index		004	068	214	260
MGM	$s = 1, lr = 1$	1.806 / 0.65	1.030 / 0.62	1.627 / 0.39	1.474 / 0.69
MGM multi	$s = 1, lr = 1$	1.512 / 0.90	1.026 / 0.68	1.585 / 0.55	1.472 / 0.78
PSM	$s = 1, lr = 1$	1.327 / 2.59	0.813 / 0.06	1.453 / 0.28	1.297 / 1.08
HSM	$s = 1, lr = 1$	1.941 / 1.38	1.072 / 0.11	1.838 / 0.34	1.708 / 1.04
PSM	$s = 2, lr = 2$	1.401 / 6.13	0.780 / 0.25	1.347 / 1.10	1.154 / 3.39
HSM	$s = 2, lr = 2$	1.405 / 1.52	0.862 / 0.13	1.579 / 0.48	1.293 / 1.16
PSM	$s = 3, lr = 3$	1.115 / 7.08	0.838 / 0.22	1.677 / 1.26	1.155 / 4.72
HSM	$s = 3, lr = 3$	1.314 / 3.13	0.893 / 0.48	1.653 / 0.65	1.285 / 0.77

Table 1: Quantitative results using panchromatic images as input. The accuracy of the single pair (top) and multi-pair DSMs (bottom) is directly related to the accuracy of the disparity maps. For single-pair DSMs, all metrics are averaged across the successful pairs. Customized parameters: s, upsampling factor applied to the rectified images input to the matching algorithm; lr, left-right consistency check threshold in pixel units. The best altitude MAE values are highlighted in yellow.

On the scaling factor. The upsampling factor s is used to bring objects closer to the resolution of the training set, since aerial images have approximately four times the resolution of satellite images (8 cm vs. 30 cm per pixel). When $s > 1$, the lr threshold should be at least equal to s since the increase of disparity values is proportional to the size of the images. In general, we observe that $s = 2$ works best across the different AOIs, as shown in Figure 6. Using $s = 3$ only provides better results for both PSM and HSM in the AOI 004, a landscape of small houses. The rest of AOIs contain tall buildings, and $s > 2$ causes certain disparities related to skyscrapers to be larger than the upper limit of the disparity range set at training time (192 pixels). The latter downgrades the accuracy for some of the input stereo pairs.

In addition to the resolution of the training set, the architecture of the networks is another factor that could explain the improvement in performance after upsampling. Figure 7 shows that the disparity map obtained with vanilla weights (trained on KITTI2015) also improves using $s = 2$. Both PSM and HSM use non-learned upsampling operations before disparity regression, to reach the original input size. Thus, the actual costs used to predict the disparity map have lower spatial resolution, and upsampling the input pair could be seen as a bruteforce manner to increase it.

On the sensitivity to the rectification step. Deep learning methods are very sensitive to the way in which input pairs are rectified. The rectification step must satisfy several conditions to obtain

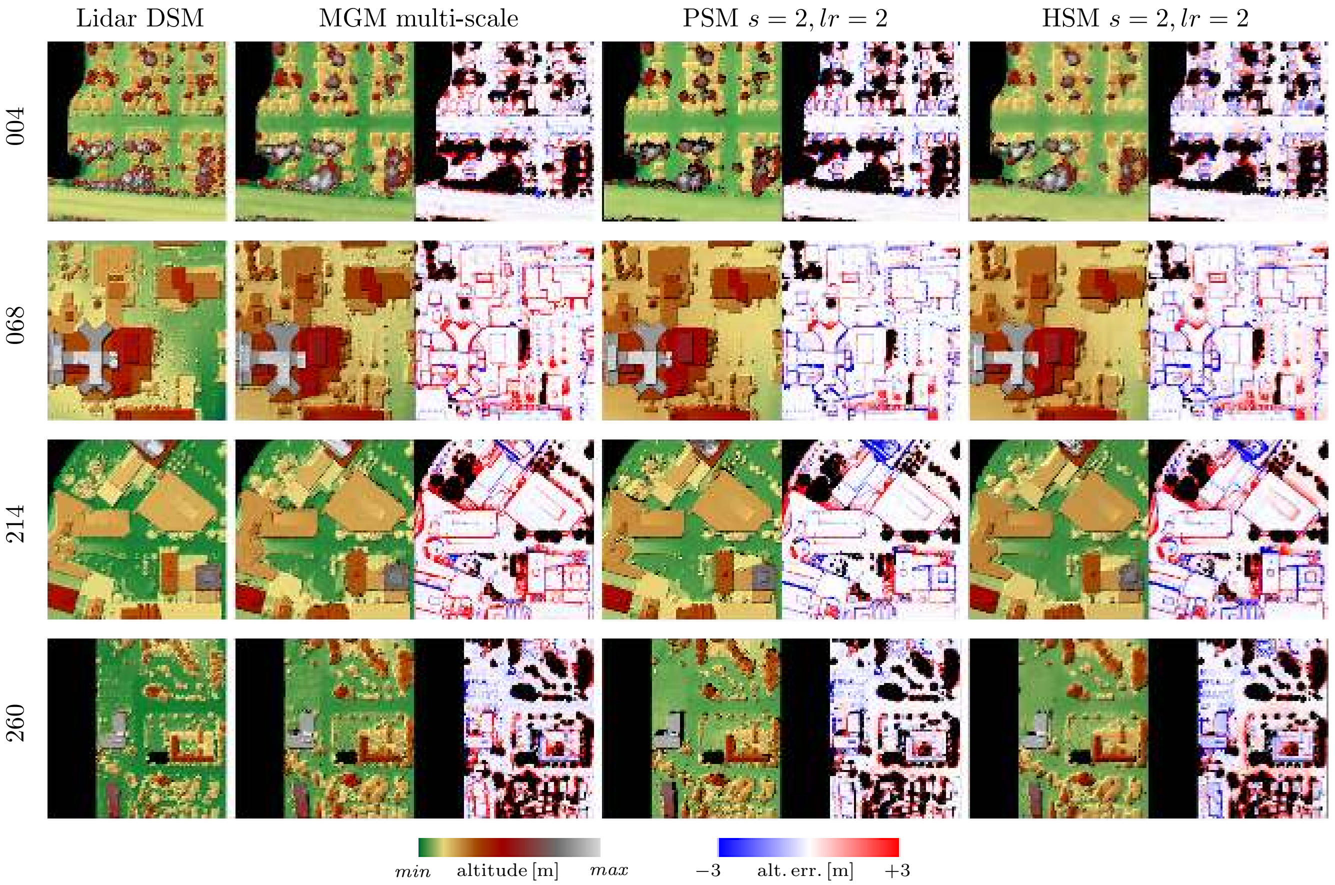


Figure 5: Visualization of the lidar DSMs, and the multi-pair photogrammetric DSMs obtained by merging the disparity maps of multiple stereo pairs obtained with MGM multi-scale, PSM and HSM. Panchromatic images used as input. Undefined values and water bodies are in black. Points corresponding to trees/vegetation in the lidar DSM are also masked in black in the error images.

optimal disparity maps: (1) all disparity values have to fall within the disparity range defined at training time; (2) all disparities from the reference image to the auxiliary image must point towards the left, following a negative displacement along each epipolar line; (3) disparity values must be proportional to the altitude level. The networks expect the ground/background to exhibit small disparity, and the opposite for objects in the foreground or tall buildings.

All pairs considered in Table 1 were rectified by means of Algorithm 1, to ensure that conditions (2) and (3) were satisfied. If one of the conditions is not met in a given region, the networks fill it with NaN values or the accuracy is degraded (Figure 8). In contrast, classic methods such as SGM or MGM do not depend on any priors and can adapt the search range of disparities for each pair.

On the trade-off between accuracy and efficiency. Both PSM and HSM consistently achieve lower MAE than the standard MGM or its multi-scale version. Only HSM with $s = 1$ seems to perform clearly worse, probably because it is more specific for large and very high resolution inputs. The behavior is the same if we take completeness as the main metric (see Appendix B, Table 7). As shown in Figure 5, the superior performance of deep learning methods can be explained by their ability to produce better and sharper contours. Between the two networks, PSM achieves better MAE and completeness than HSM, but the difference narrows as the scaling factor s increases. Using $s \geq 2$, HSM might be the better choice for certain applications, as it requires less memory and is much faster for large inputs. For example, with $s = 2$, on a CPU, most stereo pairs are processed within 5 seconds using HSM, while they take longer than 1 minute with PSM.

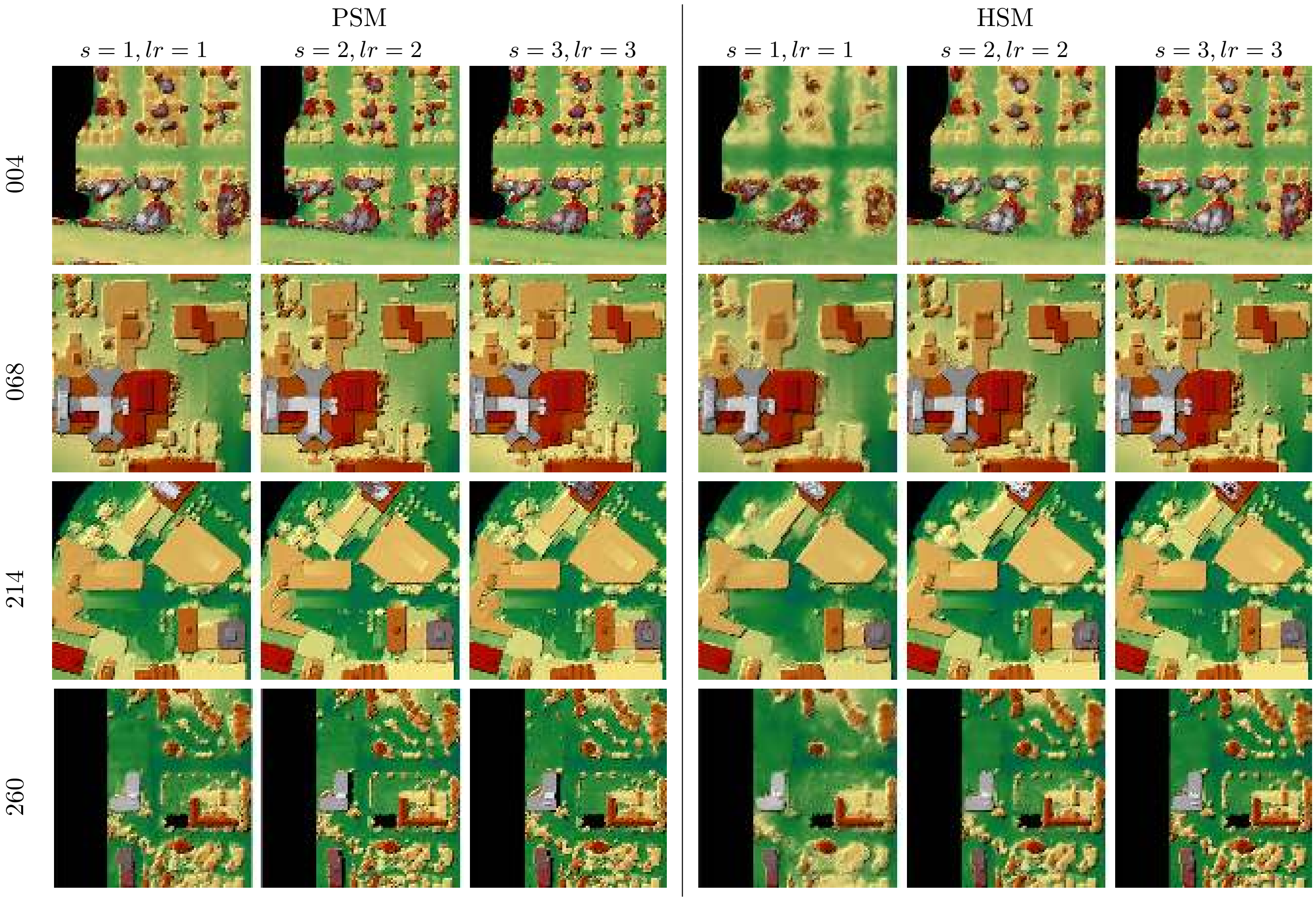


Figure 6: The resolution of the input images plays an important role in deep learning methods. By upsampling the input pair by a factor s, we simulate the resolution of the satellite images to be higher and closer to that of the aerial image training set. We find that $s = 2$ works best, as $s = 1$ produces blurred edges and $s = 3$ introduces artifacts in tall buildings. HSM is more sensitive to the resolution than PSM, as the difference in detail sharpness is stronger for different s values.

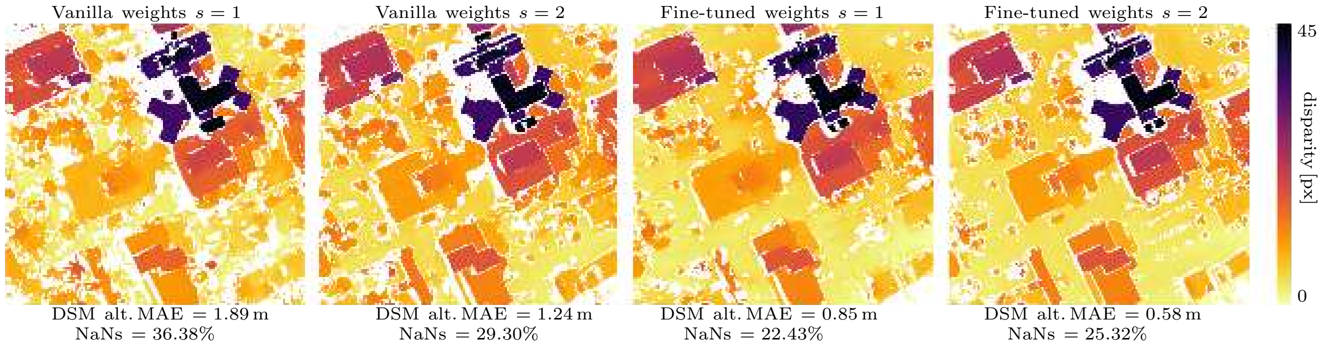


Figure 7: PSM vanilla vs. fine-tuned weights, same input pair. Vanilla weights were trained on KITTI2015 [24], while fine-tuned weights were refined using the 2021 aerial stereo matching benchmark [32]. Fine-tuned weights systematically produce better disparities and fewer undefined (NaN) values after the left-right consistency check. However, upsampling the input pair by a factor $s = 2$ improves details and accuracy in both cases. This suggests that, in addition to the resolution of the training data, the architecture of the method may also be related to this behavior.

4.4.2 RGB Inputs

In these experiments, we use the RGB version of the DFC2019 images as input. RGB images have three channels corresponding to red, green and blue values, which are compressed as integer values in $[0, 255]$. The compressed dynamics leads to a loss of texture and appearance of saturated areas.

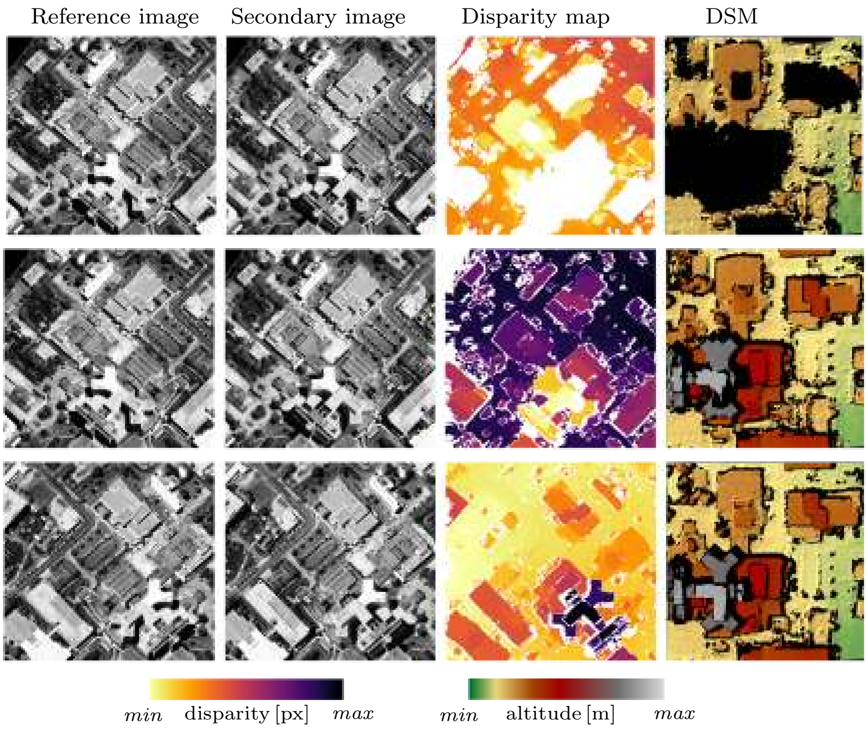
Algorithm 1: Rectify satellite stereo pair for disparity estimation network

```
input  : two satellite images $I_1$ and $I_2$ and their respective camera models
output : rectified stereo pair $I_1^R$ and $I_2^R$
1. Compute a set of $pairwise\_matches$ between $I_1$ and $I_2$
2. Use $pairwise\_matches$ to find the $3 \times 3$ rectifying homographies $H_1$ and $H_2$ (Comment 1)
3. Impose negative disparities by modifying $H_2$ into $H_2'$
     $H_2' = \begin{pmatrix} 1 & 0 & -t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} H_2,$   where $t \geq$ max disparity observed in $pairwise\_matches$
4. Check that disparity values are proportional to altitude. To do so, localize a point from $I_1^R$ at two
   different altitudes: $alt_1$ and $alt_2$, such that $alt_1 < alt_2$. Then project the resulting locations onto $I_2^R$
   (Comment 2). The reprojection localized at $alt_1$ must have smaller disparity because $alt_1 < alt_2$.
if step 4 is not satisfied then
    5. Repeat step 3 imposing positive disparities, i.e. $t \leq$ min disparity in $pairwise\_matches$
    6. Apply a horizontal flip to both rectifying homographies, e.g.
         $H_{flipped} = \begin{pmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} H,$   where $w$ is the image width
7. Rectify $I_1$ using $H_1$ and $I_2$ using $H_2'$.
```

Comment 1: Steps 1 and 2 can be covered using the stereo-rectification method for pushbroom images described in [7].

Comment 2: Let the rectifying homographies be denoted $\{H_1, H_2'\}$ and the rectified images be denoted $\{I_1^R, I_2^R\}$. Given the RPC camera models of the non-rectified satellite images $\{I_1, I_2\}$, which are characterized by a localization $\mathcal{L}$ and a projection function $\mathcal{P}$ [22], a point (x, y) in I_1^R can be localized at altitude alt and reprojected to I_2^R using (15), resulting in a disparity $d = x' - x$.

$$(x', y) = H_2' \mathcal{P}_2(\mathcal{L}_1(H_1(x, y), alt)) \tag{15}$$



Failure example 1: The input pair is rectified allowing disparities to follow a negative or positive direction, indifferently. As a result, the network only works in areas with negative displacement, making the disparity map and the derived DSM largely incomplete.

Failure example 2: The input pair is rectified forcing all disparities to follow a negative direction, but allowing the background to have a higher disparity than the foreground. As a result, the accuracy of the disparity map and the derived DSM decreases, particularly in foreground objects (e.g. tall buildings).

Success example: The input pair is rectified as expected by the network. All disparities follow a negative direction, the background has lower disparity than the foreground, and all disparities fall within the expected range (e.g. up to 192 pixels). As a result, the accuracy of the disparity map and the derived DSM are maximized.

Figure 8: Deep learning methods are very sensitive to the way in which input pairs are rectified. Using the same pair and the PSM model, the top rows show results obtained with suboptimal rectification. The bottom row shows the result obtained when the rectification follows the learned format. Undefined values are white in the disparity maps and black in the DSM.

RGB inputs

		Single pair MAE (error increase) [m] / NaNs [%]			
Area index		004	068	214	260
MGM multi	$s = 1, lr = 1$	2.020 (+0.489) / 11.19	1.196 (+0.309) / 14.11	2.388 (+1.100) / 20.04	2.064 (+0.401) / 15.85
PSM	$s = 2, lr = 2$	1.358 (+0.472) / 27.69	0.601 (+0.089) / 26.25	1.128 (+0.318) / 36.12	1.297 (+0.294) / 30.81
HSM	$s = 2, lr = 2$	1.314 (+0.203) / 31.66	0.607 (+0.002) / 28.42	0.983 (+0.071) / 35.67	1.280 (+0.091) / 32.10

		Multi-pair MAE (error increase) [m] / NaNs [%]			
Area index		004	068	214	260
MGM multi	$s = 1, lr = 1$	2.132 (+0.620) / 0.68	1.297 (+0.270) / 0.58	2.445 (+0.860) / 0.21	1.728 (+0.256) / 0.59
PSM	$s = 2, lr = 2$	1.764 (+0.363) / 1.36	0.881 (+0.101) / 0.33	1.810 (+0.463) / 1.22	1.426 (+0.272) / 1.76
HSM	$s = 2, lr = 2$	1.606 (+0.201) / 0.71	0.857 (−0.005) / 0.11	1.631 (+0.052) / 0.62	1.440 (+0.147) / 0.88

Table 2: Quantitative results for single pair (top) and multi-pair DSMs (bottom), using RGB input images. The error increase refers to the increase in altitude MAE with respect to the MAE obtained with panchromatic images (Table 1). Customized parameters s and lr are the same as in Table 1.

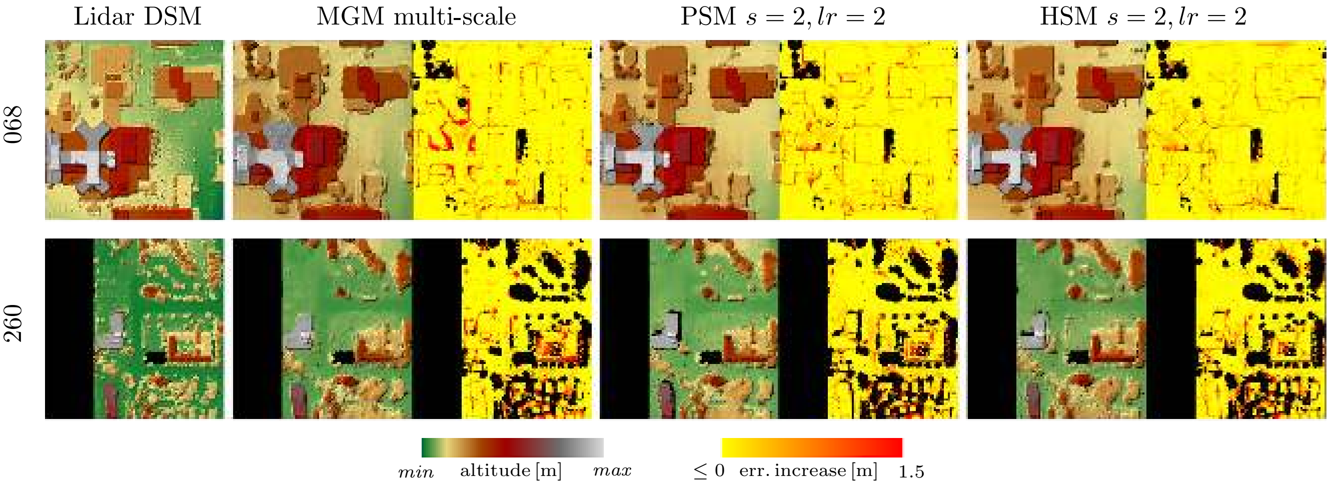


Figure 9: Examples of multi-pair photogrammetric DSMs obtained using RGB images as input instead of panchromatic as in Figure 5. The yellow-red images show the increase in altitude error with respect to the equivalent result obtained with panchromatic images. Masked points in black represent undefined points and water bodies.

The compressed dynamics makes RGB images more challenging than the panchromatic equivalent for matching purposes. The numerical results obtained with RGB images are reported in Table 2. For each row in Table 2, we used the successful pairs from the equivalent row in Table 1 as input, with the same rectifying homographies. This ensures that any difference in the resulting disparity maps and DSMs is exclusively due to the change of color space. Example results are shown in Figure 9.

As expected, the accuracy of the disparity maps decreases across all methods, inducing larger altitude MAE in the subsequent DSMs. However, the increase in error is much larger for the classic algorithm (multi-scale MGM), as it cannot compensate for the loss of information with contextual semantic cues in the same way as deep learning networks do.

Differently from the panchromatic scenario, MAE and completeness behave in a different way for RGB inputs. MGM multi-scale provided better completeness in the single pair disparity maps (Appendix B, Table 8), but higher inaccuracy too (higher MAE). RGB compression also eliminates fine-scale noise, which reduces the amount of NaN values provided by MGM in some of the pairs.

For panchromatic inputs and $s = 2$, PSM was the best performing model both in terms of MAE (Table 1) and completeness (Table 7). Using RGB inputs and $s = 2$, the performance of HSM and PSM seems to be more or less equal, with HSM being slightly more accurate in terms of MAE (Table 2) but also providing slightly lower completeness (Table 8).

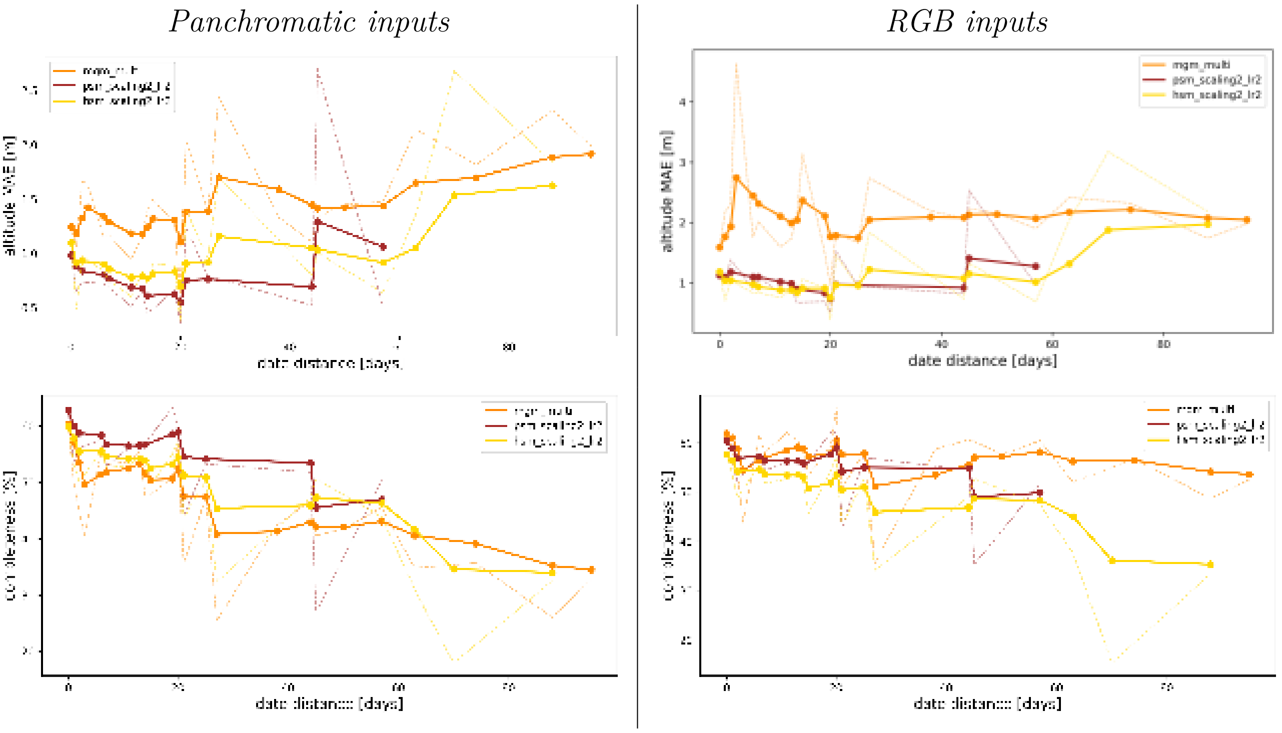


Figure 10: Average evaluation metrics as a function of the distance between the acquisition dates of each pair of images. Only successful pairs taken into account, i.e. those resulting in less than 50% of undefined altitude values. For better visualization, the opaque lines represent a smoothed version of the real dashed functions. Each smoothed value x_i' is obtained as $x_i' = 0.7x_{i-1} + 0.3x_i$.

4.4.3 Multi-date Inputs

Figure 10 shows the altitude MAE values reported in Table 1 and Table 2 plotted as a function of the distance between the acquisition dates of each pair of images. The same is also shown using the completeness values (Tables 7 and 8). The more distant the dates are, the less radiometric correlation is expected (e.g. due to seasonal changes) and the more difficult it is to find correspondences.

For all methods, the disparity maps lose accuracy as the time distance increases, causing an increase in altitude error and a decrease in completeness. Using panchromatic inputs, the decline in performance seems to be slightly more pronounced for deep learning methods, as the gap between MGM multi-scale and the networks narrows towards the end of the plots. For RGB, the decline in performance of deep learning methods is more evident, especially in terms of completeness. In particular, PSM did not produce any successful pairs for time distances over 60 days.

To better understand Figure 10, note that the number of pairs with a given distance between acquisition dates is not uniformly distributed. In 90% of cases the distance is less than 30 days.

4.4.4 Small Baseline Inputs

Figure 11 shows multi-scale MGM, PSM and HSM tested on two additional panchromatic input pairs. The *Basilique Saint-Sernin* and *Prison Saint-Michel* (both in Toulouse, France) are not part of the DFC2019 data. We selected these input pairs because of the small baseline between the two cameras, which is a challenging factor for stereo vision [8, 19][5]. Small baseline inputs are a common source of errors, because small changes in disparity can induce a large variation in the resulting depth (Equation (1)). At the same time, excessively large baselines reduce the amount of correspondences

[5] The two pairs exhibit a B/H factor between 0.05 and 0.10, where B is the baseline and H is the distance between the scene and the camera system.

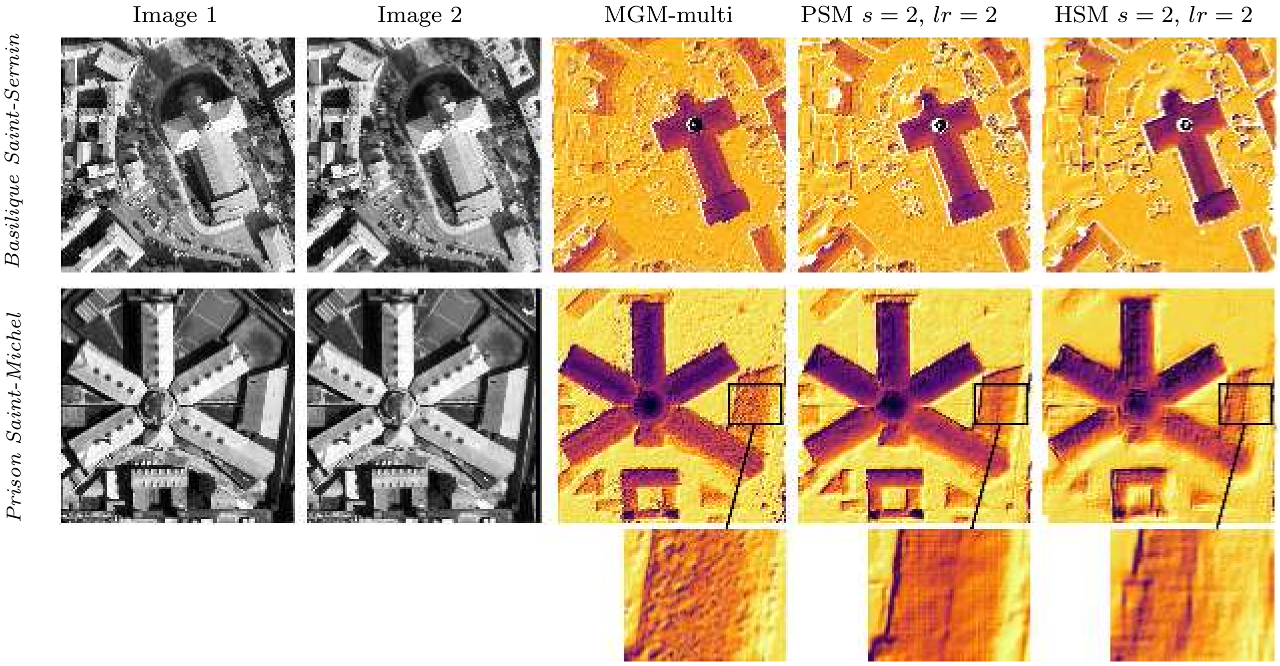


Figure 11: Small baseline experiments. Left to right: input pairs and output disparity. Moderate shading is applied to the disparity maps for better visualization. Undefined values are in white.

between the images, due to occlusions, and objects that appear and disappear from the scene. An adequate baseline must result from a compromise between these factors.

As shown in Figure 11, the disparity maps produced by multi-scale MGM are more complete. The PSM output appears less noisy, but some areas are affected by fine-scale checkerboard artifacts (see the detail in Figure 11). These fine-scale checkerboard artifacts are probably explained by the use of deconvolution layers in PSM [26, 31] and have no connection with small baseline inputs. The HSM result is significantly worse with respect to the two previous methods: much less detailed, with large-scale artifacts and visibly smoothed contours, especially in *Prison Saint-Michel*.

5 Conclusion

This paper reviewed the PSM and HSM architectures for disparity estimation from an input stereo pair and investigated their applicability for satellite stereo reconstruction. The two methods were compared with a variant of the SGM algorithm, which is a classic matching strategy widely used for satellite images. We used pre-trained weights, fine-tuned using an aerial stereo matching benchmark. The quality of the disparity maps output by each method is assessed based on the subsequent surface models, which are evaluated using a lidar reference model.

The conducted experiments show that the deep learning methods provide higher accuracy than classic concurrent algorithms, and should therefore be preferred for satellite 3D stereo reconstruction under ideal conditions. However, these networks require additional effort to adjust the format of the input pairs and may produce more incomplete results in difficult/unusual scenarios (e.g. very distant acquisition dates or small baselines). It is critical that the rectified images emulate the training conditions. For optimal results, it is also best to adapt the size of the input images.

PSM provides remarkable robustness to image resolution and accuracy, especially for highly textured inputs like panchromatic images, but becomes impractical as the input size increases. Alternatively, HSM is much faster but loses detail sharpness depending on the input resolution. One network or the other may be more convenient depending on the application and type of input.

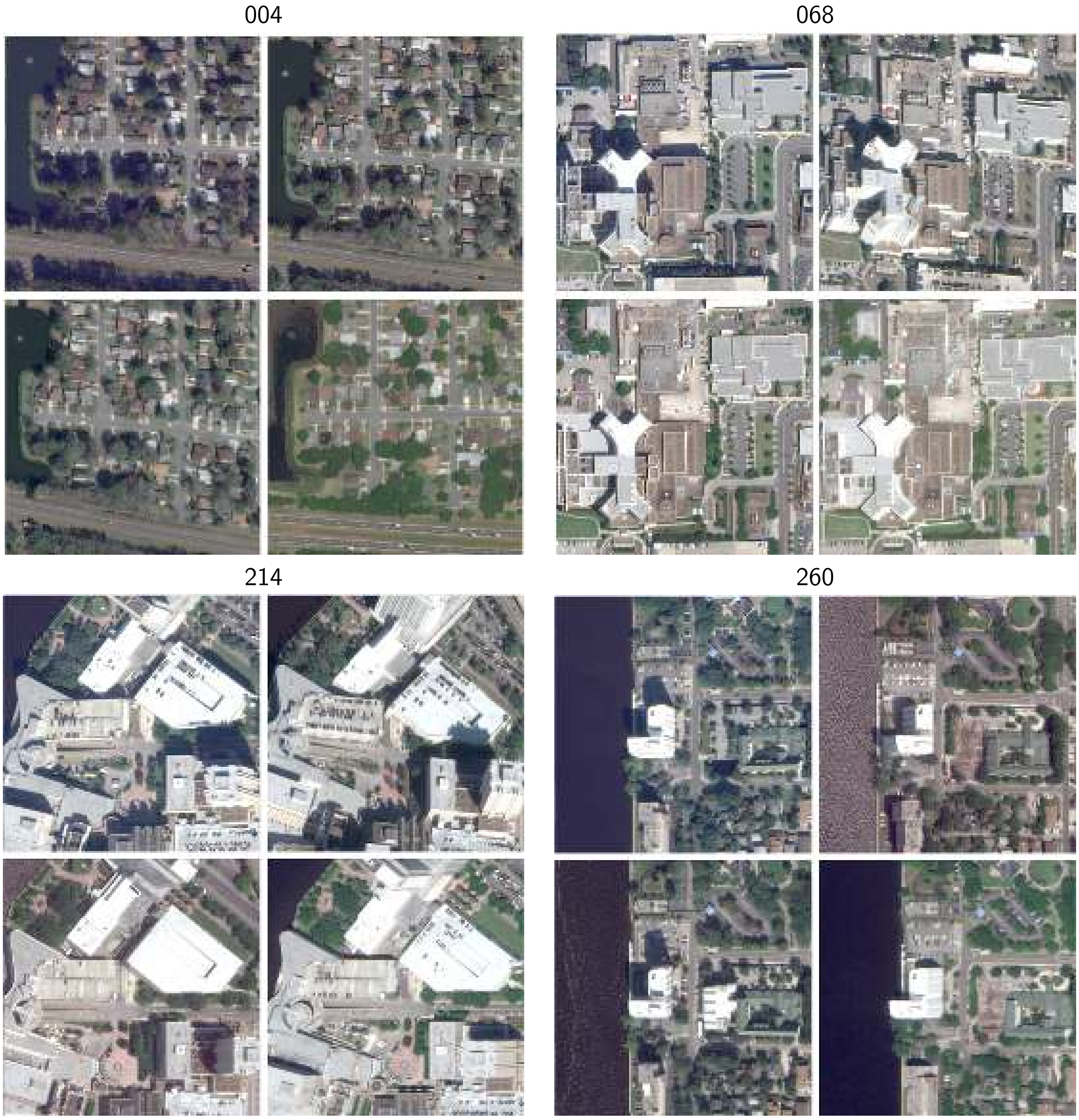


Figure 12: DFC2019 dataset, Jacksonville areas 004, 068, 214 and 260. Example RGB views.

A List of DFC2019 Selected Stereo Pairs

Tables 3 to 6 list the image pairs of the DFC2019 dataset [3] used to test the matching algorithms compared in this work. All distances between acquisition dates are expressed in days modulo one year. Figure 12 shows different RGB views of the four areas of interest that were used in the experiments in Section 4.

B Completeness Values

Table 7 and Table 8 show the completeness percentage associated with the experiments in Section 4.4.1 (panchromatic inputs) and 4.4.2 (RGB inputs), respectively. The definition of completeness is given in Section 4.3. Water bodies are not taken into account to compute the completeness percentage.

JAX 004

pair	image id 1	image id 2	intersect. angle [deg]	date dist. [days]
01	21JAN15WV031100015JAN21161253	21JAN15WV031100015JAN21161308	8.3	0.00
02	02MAY15WV031100015MAY02161943	01MAY15WV031200015MAY01160357	43.1	1.01
03	02MAY15WV031100015MAY02161943	26APR15WV031200015APR26162435	15.3	6.00
04	19APR15WV031100015APR19161439	26APR15WV031200015APR26162435	24.9	7.01
05	19APR15WV031100015APR19161439	01MAY15WV031200015MAY01160357	36.1	11.99
06	19APR15WV031100015APR19161439	02MAY15WV031100015MAY02161943	10.4	13.00
07	27DEC14WV031100014DEC27161109	14DEC14WV031100014DEC14160402	28.7	13.00
08	15FEB15WV031200015FEB15161208	21JAN15WV031100015JAN21161253	40.9	25.00
09	27DEC14WV031100014DEC27161109	21JAN15WV031100015JAN21161253	11.8	25.00
10	27DEC14WV031100014DEC27161109	21JAN15WV031100015JAN21161308	20.1	25.00
11	14DEC14WV031100014DEC14160402	21JAN15WV031100015JAN21161253	39.4	38.01
12	15JUN15WV031100015JUN15161248	02MAY15WV031100015MAY02161943	26.6	44.00
13	15JUN15WV031100015JUN15161248	01MAY15WV031200015MAY01160357	16.5	45.01
14	15JUN15WV031100015JUN15161248	26APR15WV031200015APR26162435	34.0	49.99
15	27DEC14WV031100014DEC27161109	15FEB15WV031200015FEB15161208	29.1	50.00
16	01NOV15WV031100015NOV01161954	27DEC14WV031100014DEC27161109	25.2	55.99
17	15JUN15WV031100015JUN15161248	19APR15WV031100015APR19161439	20.5	57.00
18	15FEB15WV031200015FEB15161208	19APR15WV031100015APR19161439	37.2	63.00
19	14DEC14WV031100014DEC14160402	15FEB15WV031200015FEB15161208	11.1	63.01
20	15FEB15WV031200015FEB15161208	26APR15WV031200015APR26162435	31.8	70.01
21	15FEB15WV031200015FEB15161208	01MAY15WV031200015MAY01160357	29.4	74.99
22	15FEB15WV031200015FEB15161208	02MAY15WV031100015MAY02161943	36.1	76.01
23	01NOV15WV031100015NOV01161954	21JAN15WV031100015JAN21161253	35.5	81.00
24	01NOV15WV031100015NOV01161954	21JAN15WV031100015JAN21161308	43.2	81.00
25	19APR15WV031100015APR19161439	21JAN15WV031100015JAN21161308	13.9	88.00
26	19APR15WV031100015APR19161439	21JAN15WV031100015JAN21161253	7.9	88.00
27	26APR15WV031200015APR26162435	21JAN15WV031100015JAN21161308	27.3	95.01
28	26APR15WV031200015APR26162435	21JAN15WV031100015JAN21161253	21.1	95.01
29	21JAN15WV031100015JAN21161253	01MAY15WV031200015MAY01160357	43.7	99.99
30	02MAY15WV031100015MAY02161943	21JAN15WV031100015JAN21161308	13.8	101.00

Table 3: List of 30 stereo pairs used in DFC2019 Jacksonville area 004.

JAX 068

pair	image id 1	image id 2	intersect. angle [deg]	date dist. [days]
01	05OCT14WV031100014OCT05160149	05OCT14WV031100014OCT05160138	7.2	0.00
02	21JAN15WV031100015JAN21161253	21JAN15WV031100015JAN21161308	8.3	0.00
03	02MAY15WV031100015MAY02161943	01MAY15WV031200015MAY01160357	43.1	1.01
04	01NOV15WV031100015NOV01161954	30OCT14WV031100014OCT30155732	15.3	2.02
05	02MAY15WV031100015MAY02161943	26APR15WV031200015APR26162435	15.3	6.00
06	05OCT14WV031100014OCT05160149	11OCT14WV031100014OCT11155720	27.5	6.00
07	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161308	29.6	6.00
08	11OCT14WV031100014OCT11155720	05OCT14WV031100014OCT05160138	34.0	6.00
09	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161253	21.9	6.00
10	19APR15WV031100015APR19161439	26APR15WV031200015APR26162435	24.9	7.01
11	18OCT14WV031100014OCT18160722	11OCT14WV031100014OCT11155720	25.1	7.01
12	19APR15WV031100015APR19161439	01MAY15WV031200015MAY01160357	36.1	11.99
13	18OCT14WV031100014OCT18160722	30OCT14WV031100014OCT30155732	25.3	11.99
14	19APR15WV031100015APR19161439	02MAY15WV031100015MAY02161943	10.4	13.00
15	18OCT14WV031100014OCT18160722	05OCT14WV031100014OCT05160149	18.6	13.00
16	18OCT14WV031100014OCT18160722	05OCT14WV031100014OCT05160138	24.8	13.00
17	27DEC14WV031100014DEC27161109	14DEC14WV031100014DEC14160402	28.7	13.00
18	18OCT14WV031100014OCT18160722	01NOV15WV031100015NOV01161954	19.6	14.01
19	21MAY15WV031200015MAY21161849	02MAY15WV031100015MAY02161943	15.0	19.00
20	30OCT14WV031100014OCT30155732	11OCT14WV031100014OCT11155720	8.4	19.00
21	27JAN15WV031100015JAN27160845	15FEB15WV031200015FEB15161208	22.0	19.00
22	21MAY15WV031200015MAY21161849	01MAY15WV031200015MAY01160357	33.0	20.01
23	15JUN15WV031100015JUN15161248	05JUL15WV031100015JUL05162954	37.8	20.01
24	01NOV15WV031100015NOV01161954	11OCT14WV031100014OCT11155720	21.8	21.02
25	21MAY15WV031200015MAY21161849	15JUN15WV031100015JUN15161248	17.6	25.00
26	21MAY15WV031200015MAY21161849	26APR15WV031200015APR26162435	16.6	25.00
27	05OCT14WV031100014OCT05160149	30OCT14WV031100014OCT30155732	21.8	25.00
28	30OCT14WV031100014OCT30155732	05OCT14WV031100014OCT05160138	27.4	25.00
29	15FEB15WV031200015FEB15161208	21JAN15WV031100015JAN21161253	40.9	25.00
30	27DEC14WV031100014DEC27161109	21JAN15WV031100015JAN21161253	11.8	25.00

Table 4: List of 30 stereo pairs used in DFC2019 Jacksonville area 068.

JAX 214

pair	image id 1	image id 2	intersect. angle [deg]	date dist. [days]
01	21JAN15WV031100015JAN21161243	21JAN15WV031100015JAN21161253	8.2	0.00
02	05OCT14WV031100014OCT05160149	05OCT14WV031100014OCT05160138	7.2	0.00
03	21JAN15WV031100015JAN21161253	21JAN15WV031100015JAN21161308	8.3	0.00
04	21JAN15WV031100015JAN21161243	21JAN15WV031100015JAN21161308	16.5	0.00
05	01NOV15WV031100015NOV01161954	01NOV15WV031100015NOV01162034	27.1	0.00
06	02MAY15WV031100015MAY02161943	01MAY15WV031200015MAY01160357	43.1	1.01
07	01NOV15WV031100015NOV01161954	30OCT14WV031100014OCT30155732	15.3	2.02
08	30OCT14WV031100014OCT30155732	01NOV15WV031100015NOV01162034	17.4	2.02
09	15FEB15WV031200015FEB15161208	18FEB16WV031200016FEB18164007	35.5	3.02
10	15FEB15WV031200015FEB15161208	11FEB16WV031100016FEB11163042	38.7	3.99
11	02MAY15WV031100015MAY02161943	26APR15WV031200015APR26162435	15.3	6.00
12	05OCT14WV031100014OCT05160149	11OCT14WV031100014OCT11155720	27.5	6.00
13	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161308	29.6	6.00
14	11OCT14WV031100014OCT11155720	05OCT14WV031100014OCT05160138	34.0	6.00
15	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161253	21.9	6.00
16	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161243	15.1	6.00
17	11FEB16WV031100016FEB11163042	18FEB16WV031200016FEB18164007	24.5	7.01
18	19APR15WV031100015APR19161439	26APR15WV031200015APR26162435	24.9	7.01
19	18OCT14WV031100014OCT18160722	11OCT14WV031100014OCT11155720	25.1	7.01
20	05OCT14WV031100014OCT05160149	25SEP15WV031100015SEP25163525	42.2	9.98
21	19APR15WV031100015APR19161439	01MAY15WV031200015MAY01160357	36.1	11.99
22	18OCT14WV031100014OCT18160722	30OCT14WV031100014OCT30155732	25.3	11.99
23	19APR15WV031100015APR19161439	02MAY15WV031100015MAY02161943	10.4	13.00
24	18OCT14WV031100014OCT18160722	05OCT14WV031100014OCT05160149	18.6	13.00
25	18OCT14WV031100014OCT18160722	05OCT14WV031100014OCT05160138	24.8	13.00
26	27DEC14WV031100014DEC27161109	14DEC14WV031100014DEC14160402	28.7	13.00
27	18OCT14WV031100014OCT18160722	01NOV15WV031100015NOV01161954	19.6	14.01
28	18OCT14WV031100014OCT18160722	01NOV15WV031100015NOV01162034	22.3	14.01
29	27JAN15WV031100015JAN27160845	11FEB16WV031100016FEB11163042	21.0	15.02
30	11OCT14WV031100014OCT11155720	25SEP15WV031100015SEP25163525	43.9	15.97

Table 5: List of 30 stereo pairs used in DFC2019 Jacksonville area 214.

JAX 260

pair	image id 1	image id 2	intersect. angle [deg]	date dist. [days]
01	21JAN15WV031100015JAN21161243	21JAN15WV031100015JAN21161253	8.2	0.00
02	02MAY15WV031100015MAY02161943	01MAY15WV031200015MAY01160357	43.1	1.01
03	01NOV15WV031100015NOV01161954	30OCT14WV031100014OCT30155732	15.3	2.02
04	15FEB15WV031200015FEB15161208	18FEB16WV031200016FEB18164007	35.5	3.02
05	15FEB15WV031200015FEB15161208	11FEB16WV031100016FEB11163042	38.7	3.99
06	02MAY15WV031100015MAY02161943	26APR15WV031200015APR26162435	15.3	6.00
07	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161253	21.9	6.00
08	27JAN15WV031100015JAN27160845	21JAN15WV031100015JAN21161243	15.1	6.00
09	11FEB16WV031100016FEB11163042	18FEB16WV031200016FEB18164007	24.5	7.01
10	19APR15WV031100015APR19161439	26APR15WV031200015APR26162435	24.9	7.01
11	19APR15WV031100015APR19161439	01MAY15WV031200015MAY01160357	36.1	11.99
12	18OCT14WV031100014OCT18160722	30OCT14WV031100014OCT30155732	25.3	11.99
13	19APR15WV031100015APR19161439	02MAY15WV031100015MAY02161943	10.4	13.00
14	18OCT14WV031100014OCT18160722	05OCT14WV031100014OCT05160138	24.8	13.00
15	27DEC14WV031100014DEC27161109	14DEC14WV031100014DEC14160402	28.7	13.00
16	18OCT14WV031100014OCT18160722	01NOV15WV031100015NOV01161954	19.6	14.01
17	27JAN15WV031100015JAN27160845	11FEB16WV031100016FEB11163042	21.0	15.02
18	21MAY15WV031200015MAY21161849	02MAY15WV031100015MAY02161943	15.0	19.00
19	27JAN15WV031100015JAN27160845	15FEB15WV031200015FEB15161208	22.0	19.00
20	21MAY15WV031200015MAY21161849	01MAY15WV031200015MAY01160357	33.0	20.01
21	21JAN15WV031100015JAN21161243	11FEB16WV031100016FEB11163042	6.1	21.01
22	27JAN15WV031100015JAN27160845	18FEB16WV031200016FEB18164007	32.9	22.02
23	21MAY15WV031200015MAY21161849	26APR15WV031200015APR26162435	16.6	25.00
24	30OCT14WV031100014OCT30155732	05OCT14WV031100014OCT05160138	27.4	25.00
25	15FEB15WV031200015FEB15161208	21JAN15WV031100015JAN21161253	40.9	25.00
26	21JAN15WV031100015JAN21161243	15FEB15WV031200015FEB15161208	32.7	25.00
27	27DEC14WV031100014DEC27161109	21JAN15WV031100015JAN21161253	11.8	25.00
28	01NOV15WV031100015NOV01161954	05OCT14WV031100014OCT05160138	12.3	27.01
29	21JAN15WV031100015JAN21161253	18FEB16WV031200016FEB18164007	27.6	28.02
30	21JAN15WV031100015JAN21161243	18FEB16WV031200016FEB18164007	24.1	28.02

Table 6: List of 30 stereo pairs used in DFC2019 Jacksonville area 260.

Panchromatic inputs

		Single pair completeness [%] / NaNs [%] / Successful pairs			
Area index		004	068	214	260
MGM	$s = 1, lr = 1$	43.62 / 32.01 / 24	61.64 / 23.63 / 29	53.67 / 31.23 / 24	41.03 / 33.30 / 24
MGM multi	$s = 1, lr = 1$	44.64 / 34.63 / 18	60.33 / 25.74 / 28	53.61 / 31.95 / 23	39.75 / 34.48 / 21
PSM	$s = 1, lr = 1$	51.38 / 33.17 / 12	63.41 / 25.00 / 28	53.13 / 34.23 / 23	42.95 / 33.94 / 16
HSM	$s = 1, lr = 1$	39.78 / 34.64 / 20	52.93 / 28.90 / 28	43.42 / 34.89 / 23	35.87 / 31.72 / 24
PSM	$s = 2, lr = 2$	50.51 / 38.61 / 12	64.30 / 28.34 / 28	55.01 / 36.88 / 23	45.43 / 38.33 / 14
HSM	$s = 2, lr = 2$	45.06 / 37.63 / 16	61.79 / 28.09 / 28	52.98 / 35.40 / 23	43.97 / 32.98 / 19
PSM	$s = 3, lr = 3$	54.09 / 37.87 / 12	61.10 / 30.33 / 26	54.54 / 36.89 / 18	45.14 / 38.37 / 11
HSM	$s = 3, lr = 3$	48.83 / 37.15 / 12	61.32 / 28.88 / 28	52.90 / 36.14 / 23	45.02 / 34.49 / 18

		Multi-pair completeness [%] / NaNs [%]			
Area index		004	068	214	260
MGM	$s = 1, lr = 1$	67.06 / 0.65	78.65 / 0.62	74.28 / 0.39	64.59 / 0.69
MGM multi	$s = 1, lr = 1$	67.11 / 0.90	78.47 / 0.68	74.43 / 0.55	64.23 / 0.78
PSM	$s = 1, lr = 1$	67.71 / 2.59	82.14 / 0.06	74.00 / 0.28	65.38 / 1.08
HSM	$s = 1, lr = 1$	54.39 / 1.38	73.64 / 0.11	61.56 / 0.34	50.19 / 1.04
PSM	$s = 2, lr = 2$	65.30 / 6.13	83.50 / 0.25	77.95 / 1.10	68.52 / 3.39
HSM	$s = 2, lr = 2$	67.53 / 1.52	81.89 / 0.13	73.39 / 0.48	65.53 / 1.16
PSM	$s = 3, lr = 3$	70.15 / 7.08	83.11 / 0.22	76.50 / 1.26	67.26 / 4.72
HSM	$s = 3, lr = 3$	68.36 / 3.13	81.01 / 0.48	74.62 / 0.65	68.02 / 0.77

Table 7: Quantitative results using panchromatic images as input. Equivalent to Table 1, with completeness in substitution of altitude MAE. The best completeness values are highlighted in yellow.

RGB inputs

		Single pair completeness (diff. w.r.t. PAN) [%] / NaNs [%]			
Area index		004	068	214	260
MGM multi	$s = 1, lr = 1$	58.11 (+13.47) / 11.19	68.34 (+8.01) / 14.11	56.90 (+3.29) / 20.04	48.61 (+8.86) / 15.85
PSM	$s = 2, lr = 2$	53.32 (+2.81) / 27.69	64.89 (+0.59) / 26.25	52.50 (−2.50) / 36.12	47.20 (+1.77) / 30.81
HSM	$s = 2, lr = 2$	46.48 (+1.42) / 31.66	61.50 (−0.29) / 28.42	51.38 (−1.60) / 35.67	43.23 (−0.74) / 32.10

		Multi-pair completeness (diff. w.r.t. PAN) [%] / NaNs [%]			
Area index		004	068	214	260
MGM multi	$s = 1, lr = 1$	66.27 (−0.84) / 0.68	78.71 (+0.24) / 0.58	70.25 (−4.18) / 0.21	62.67 (−1.56) / 0.59
PSM	$s = 2, lr = 2$	65.06 (−0.24) / 1.36	82.55 (−0.95) / 0.33	72.69 (−5.26) / 1.22	65.93 (−2.59) / 1.76
HSM	$s = 2, lr = 2$	66.82 (−0.71) / 0.71	81.96 (+0.07) / 0.11	71.82 (−1.57) / 0.62	62.21 (−3.32) / 0.88

Table 8: Quantitative results using RGB images as input. Equivalent to Table 2, with completeness in substitution of altitude MAE. The best completeness values are highlighted in yellow. The percentage difference with respect to the equivalent PAN experiment is shown in parentheses.

Acknowledgment

This work was supported by a grant from Région Île-de-France. It was also partly financed by Office of Naval research grant N00014-17-1-2552, MENRT, and Kayrros. This work was performed using HPC resources from GENCI–IDRIS (grants 2021-AD011012453 and 2022-AD011012453R1) and from the "Mésocentre" computing center of CentraleSupélec and ENS Paris-Saclay supported by CNRS and Région Île-de-France (http://mesocentre.centralesupelec.fr).

Image Credits

All remote sensing images used in this work were extracted from the DFC2019 public dataset [3], with the exception of the aerial images in Figure 4 which belong to the 2021 aerial stereo benchmark by Wu et al. [32], and the small baseline image pairs in Figure 11, courtesy of the French government space agency CNES (*Centre National d'Études Spatiales*).

References

[1] H. Albanwan and R. Qin, *Fine-tuning deep learning models for stereo matching using results from semi-global matching*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 5-2-2022 (2022), pp. 39–46. https://doi.org/10.5194/isprs-annals-V-2-2022-39-2022.

[2] R.A. Beyer, O. Alexandrov, and S. McMichael, *The Ames Stereo Pipeline: NASA's open source software for deriving and processing terrain data*, Earth and Space Science, 5 (2018), pp. 537–548. https://doi.org/10.1029/2018EA000409.

[3] M. Bosch, K. Foster, G. Christie, S. Wang, G.D. Hager, and M. Brown, *Semantic stereo for incidental satellite images*, in IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1524–1532. https://doi.org/10.1109/WACV.2019.00167.

[4] J-R. Chang and Y-S. Chen, *Pyramid stereo matching network*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5410–5418. https://doi.org/10.1109/CVPR.2018.00567.

[5] P. d'Angelo and P. Reinartz, *Semiglobal matching results on the ISPRS stereo matching benchmark*, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38-4/W19 (2011), pp. 79–84. https://doi.org/10.5194/isprsarchives-XXXVIII-4-W19-79-2011.

[6] C. De Franchis, E. Meinhardt-Llopis, J. Michel, J-M. Morel, and G. Facciolo, *An automatic and modular stereo pipeline for pushbroom images*, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2-3 (2014), pp. 49–56. https://doi.org/10.5194/isprsannals-II-3-49-2014.

[7] ——, *On stereo-rectification of pushbroom images*, in IEEE International Conference on Image Processing (ICIP), 2014, pp. 5447–5451. https://doi.org/10.1109/ICIP.2014.7026102.

[8] J. Delon and B. Rougé, *Small baseline stereovision*, Journal of Mathematical Imaging and Vision, 28 (2007), pp. 209–223. https://doi.org/10.1007/s10851-007-0001-1.

[9] S. Duggal, S. Wang, W-C. Ma, R. Hu, and R. Urtasun, *DeepPruner: Learning efficient stereo matching via differentiable PatchMatch*, in IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4384–4393. https://doi.org/10.1109/ICCV.2019.00448.

[10] G. Facciolo, C. De Franchis, and E. Meinhardt-Llopis, *Automatic 3D reconstruction from multi-date satellite images*, in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1542–1551. https://doi.org/10.1109/CVPRW.2017.198.

[11] G. Facciolo, C. De Franchis, and E. Meinhardt, *MGM: A significantly more global matching for stereovision*, in British Machine Vision Conference (BMVC), no. 90, 2015, pp. 1–12. https://doi.org/10.5244/C.29.90.

[12] International Society for Photogrammetry and Remote Sensing, *ISPRS test project on urban classification, 3D building reconstruction and semantic labeling*, 2012. https://www.isprs.org/education/benchmarks/UrbanSemLab/default.aspx.

[13] Y. Furukawa and C. Hernández, *Multi-view stereo: A tutorial*, Foundations and Trends in Computer Graphics and Vision, 9 (2015), pp. 1–148. https://doi.org/10.1561/0600000052.

[14] A. Geiger, P. Lenz, and R. Urtasun, *Are we ready for autonomous driving? The KITTI vision benchmark suite*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074.

[15] A. Gómez, G. Randall, G. Facciolo, and R. Grompone von Gioi, *An experimental comparison of multi-view stereo approaches on satellite images*, in IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 844–853. https://doi.org/10.1109/WACV51458.2022.00078.

[16] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, second ed., 2004. https://doi.org/10.1017/CBO9780511811685.

[17] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[18] H. Hirschmuller, *Stereo processing by semiglobal matching and mutual information*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 30 (2007), pp. 328–341. https://doi.org/10.1109/TPAMI.2007.1166.

[19] L. Igual, J. Preciozzi, L. Garrido, A. Almansa, V. Caselles, and B. Rougé, *Automatic low baseline stereo in urban areas*, Inverse Problems and Imaging, 1 (2007), pp. 319–348. https://doi.org/10.3934/ipi.2007.1.319.

[20] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, *End-to-end learning of geometry and context for deep stereo regression*, in IEEE International Conference on Computer Vision (ICCV), 2017, pp. 66–75. https://doi.org/10.1109/ICCV.2017.17.

[21] H. Laga, L. V. Jospin, F. Boussaid, and M. Bennamoun, *A survey on deep learning techniques for stereo-based depth estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2020), pp. 1738–1764. https://doi.org/10.1109/TPAMI.2020.3032602.

[22] R. Marí, C. De Franchis, E. Meinhardt-Llopis, J. Anger, and G. Facciolo, *A generic bundle adjustment methodology for indirect RPC model refinement of satellite imagery*, Image Processing On Line, 11 (2021), pp. 344–373. https://doi.org/10.5201/ipol.2021.352.

[23] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, *A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4040–4048. https://doi.org/10.1109/CVPR.2016.438.

[24] M. Menze and A. Geiger, *Object scene flow for autonomous vehicles*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3061–3070. https://doi.org/10.1109/CVPR.2015.7298925.

[25] A. Newell, K. Yang, and J. Deng, *Stacked hourglass networks for human pose estimation*, in European Conference on Computer Vision (ECCV), 2016, pp. 483–499. https://doi.org/10.1007/978-3-319-46484-8_29.

[26] A. Odena, V. Dumoulin, and C. Olah, *Deconvolution and checkerboard artifacts*, Distill, (2016). https://doi.org/10.23915/distill.00003.

[27] J. Pang, W. Sun, J.S.J. Ren, C. Yang, and Q. Yan, *Cascade residual learning: A two-stage convolutional neural network for stereo matching*, in IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 878–886. https://doi.org/10.1109/ICCVW.2017.108.

[28] M. 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. https://github.com/micmacIGN/Documentation.

[29] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, *The ISPRS benchmark on urban object classification and 3D building reconstruction*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1-3 (2012), pp. 293–298. https://doi.org/10.5194/isprsannals-I-3-293-2012.

[30] D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International Journal of Computer Vision, 47 (2002), pp. 7–42. https://vision.middlebury.edu/stereo/taxonomy-IJCV.pdf.

[31] Y. Sugawara, S. Shiota, and H. Kiya, *Super-resolution using convolutional neural networks without any checkerboard artifacts*, in IEEE International Conference on Image Processing (ICIP), 2018, pp. 66–70. https://doi.org/10.1109/ICIP.2018.8451141.

[32] T. Wu, B. Vallet, M. Pierrot-Deseilligny, and E. Rupnik, *A new stereo dense matching benchmark dataset for deep learning*, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43-B2-2021 (2021), pp. 405–412. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-405-2021.

[33] G. Yang, J. Manela, M. Happold, and D. Ramanan, *Hierarchical deep stereo matching on high-resolution images*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5515–5524. https://doi.org/10.1109/CVPR.2019.00566.

[34] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, *MVSNet: Depth inference for unstructured multi-view stereo*, in European Conference on Computer Vision (ECCV), 2018, pp. 785–801. https://doi.org/10.1007/978-3-030-01237-3_47.

[35] F. Zhang, V. Prisacariu, R. Yang, and P.H.S. Torr, *GA-Net: Guided aggregation net for end-to-end stereo matching*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 185–194. https://doi.org/10.1109/CVPR.2019.00027.

[36] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, *Pyramid scene parsing network*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230–6239. https://doi.org/10.1109/CVPR.2017.660.