A NEW STEREO DENSE MATCHING BENCHMARK DATASET FOR DEEP LEARNING

Teng Wu[1,*], Bruno Vallet[1], Marc Pierrot-Deseilligny[1], Ewelina Rupnik[1]

[1] LASTIG, Univ Gustave Eiffel, ENSG, IGN, F-94160 Saint-Mandé, France - firstname.lastname@ign.fr

Commission II, WG II/4, II/6

KEY WORDS: Stereo dense matching, deep learning, benchmarking, LiDAR processing, 3D reconstruction

ABSTRACT:

Stereo dense matching is a fundamental task for 3D scene reconstruction. Recently, deep learning based methods have proven effective on some benchmark datasets, for example Middlebury and KITTI stereo. However, it is not easy to find a training dataset for aerial photogrammetry. Generating ground truth data for real scenes is a challenging task. In the photogrammetry community, many evaluation methods use digital surface models (DSM) to generate the ground truth disparity for the stereo pairs, but in this case interpolation may bring errors in the estimated disparity. In this paper, we publish a stereo dense matching dataset based on ISPRS Vaihingen dataset, and use it to evaluate some traditional and deep learning based methods. The evaluation shows that learning-based methods outperform traditional methods significantly when the fine tuning is done on a similar landscape. The benchmark also investigates the impact of the base to height ratio on the performance of the evaluated methods. The dataset can be found in https://github.com/whuwuteng/benchmark_ISPRS2021.

1. INTRODUCTION

Dense matching is a traditional topic in 3D reconstruction, which can be performed in stereo (with only two views) (Scharstein, Szeliski, 2002) or multi-view stereo (MVS) (Jensen et al., 2014). In this paper, we focus on stereo dense matching in the specific case of epipolar stereo pairs (where expected correspondences are on the same lines of the two images) as most of the recent deep learning approaches are limited to this simple configuration. The recent successes of deep learning based dense matching methods in the computer vision community (Laga et al., 2020) raise the question of their applicability in the geospatial context. This paper will investigate this question by comparing traditional and machine learning especially deep learning dense matching techniques on geospatial data.

1.1 Traditional methods

Traditional dense matching methods (Hirschmuller, 2005) are usually decomposed into four steps: hand-crafted features computation, feature matching across images (i.e., the cost volume), cost aggregation and disparity refinement. They can be divided into local and global methods. Local methods mainly take into consideration the local features (hand-crafted feature) (Hirschmuller, Scharstein, 2007) or a local region (Tombari et al., 2008). The global methods mainly add an optimization based cost aggregation step, based on dynamic programming (Van Meerbergen et al., 2002), belief propagation (Sun et al., 2003) or Graphcut optimization (Boykov, Kolmogorov, 2004). Semi global matching (SGM) is a reference method combining mutual information and dynamic programming optimization on several directions (Hirschmuller, 2005). A GPU variant of SGM (Hernandez-Juarez et al., 2016) on the full image scale is also evaluated in this paper, as well as a Graphcut based method using plane constraints (Taniai et al., 2017). Graphcut based methods are slower than SGM which is often considered to offer the best balance between efficiency and accuracy (Bethmann, Luhmann, 2015).

1.2 Learning based methods

Traditional machine learning base methods have been proposed to address the problem of dense matching. Support vector machine can be used to learn a linear discriminant function (Li, Huttenlocher, 2008). Because features have their pros and cons, a random forest (RF) can be used to fuse several feature types, e.g. census, normalized cross-correlation (NCC), zero-mean sum of Absolute Differences (SAD), SAD of Sobel feature (Batsos et al., 2018). After feature fusion, a traditional optimization is used to obtain the final result.

With development of deep learning, some steps of the traditional methods can be replaced by deep learning counterparts (Laga et al., 2020). For instance, 2D convolutional neural networks (CNN) prove effective in feature extraction (Žbontar, LeCun, 2016). In order to make CNN efficient, feature similarity calculation can be treated as a multi-class classification (Luo et al., 2016). After feature similarity calculation, traditional optimization is used to obtain the final result. SGM-Net uses a CNN network to provide learned penalties for SGM (Seki, Pollefeys, 2017). GC-Net uses a 3D CNN based network as cost aggregation (Kendall et al., 2017). Pyramid Stereo Matching network uses spatial pyramid pooling and 3D CNN (Chang, Chen, 2018). High resolution stereo network uses upscaling in the 2D CNN network, so that the 3D cost volume does not need to be down-scaled (Yang et al., 2019a). To address the high memory consumption for high-resolution image matching, a recent method (Duggal et al., 2019) proposes to prune the 3D cost volume with a differential patch match method. CNN and conditional random fields (CRF) can be combined into a hybrid CNN-CRF model which is more effective than fully-connected CRFs (Kendall et al., 2017). While these methods were developed by the computer vision community on indoor or outdoor dataset, we tested them on the ISPRS Vaihingen dataset (Knöbelreiter et al., 2018), and we used Lidar data to generate ground truth disparity maps for training and evaluation.

* Corresponding author

This paper evaluates the *state-of-the-art* deep learning based dense matching approaches for which a public implementation is available.

1.3 Benchmark data

Table 1 lists some benchmarks for stereo dense matching on real scenes proposed by the robotics and computer vision communities. Some are popular and widely used: the indoor Middlebury dataset (Scharstein et al., 2014); the KITTI datasets in two versions, KITTI 2012 (Geiger et al., 2012) and KITTI 2015 (Mayer et al., 2016); or the ETH3D dataset containing stereo pairs (Schops et al., 2017). Using advanced computer graphics, some virtual datasets have also been generated (Yang et al., 2019a; Mayer et al., 2016). The strong ongoing research activity on autonomous driving has also resulted in several dedicated datasets such as Drivingstereo or AppolloScape (Yang et al., 2019b; Huang et al., 2019).

In aerial photogrammetry, benchmark datasets focus mainly on MVS dense matching (Cavegn et al., 2014). For satellite imagery, IARPA dataset is widely used for image dense matching evaluation (Bosch et al., 2016), the traditional evaluation method depends on DSM (Bosch et al., 2019), but the pipeline contains other steps, for example, point cloud fusion and DSM generation (Cournet et al., 2020). These steps can influence the accuracy. A recent satellite image stereo benchmark from LiDAR DSM (Patil et al., 2019) uses IARPA dataset. At the same time, for high density LiDAR, when generating the DSM, points on the walls are ignored.

Generating ground truth data from real scenes is challenging. To avoid errors introduced by grid interpolation or point cloud fusion and keep the points on walls, we propose a method to generate sparse disparity ground truth for training deep learning architectures. Then, we evaluate both traditional and learning based methods on the proposed benchmark.

1.4 Overview

The paper is structured as follows: Benchmark data generation is described in Section 2. Evaluations are provided in Section 3. Finally, conclusions are drawn and the perspectives are proposed in Section 4.

2. GROUND TRUTH DATA GENERATION

In this paper, we use the Vaihingen dataset from the ISPRS 3D reconstruction benchmark which provides a good registration of oriented images and LiDAR point clouds. The dataset is composed of 20 images with a depth of 11 bits and the ground sample distance (GSD) of 8 *cm*. The median LiDAR point density is 6.7 $points/m^2$ (Rottensteiner et al., 2012). From this data, epipolar stereo image pairs and the corresponding disparity maps are generated in four steps: data preprocessing, epipolar image generation, point cloud projection, and benchmark data production.

2.1 Data preprocessing

The Vaihingen dataset contains the images and their orientation parameters, plus a LiDAR point cloud. First, we convert the orientation files into the open source MicMac format (Pierrot-Deseilligny et al., 2014) on which our workflow is based. As the images can only capture the apparent surface we keep only the first LiDAR echo. Moreover, we remove isolated points based on the point cloud density using a grid filtering approach (Cho et al., 2004).

2.2 Epipolar image generation

The first step of our process is to create epipolar image pairs from input image pairs with sufficient footprint overlap. Coarse image footprints are obtained using an approximate height of the scene and the Computational Geometry Algorithms Library (CGAL) (Flötotto, 2020) is used to compute their intersections. Only image pairs with an intersection area above half the image footprint are considered for epipolar rectification, which was also done using the MicMac library. The corresponding orientation parameter files are generated in order to allow the direct projection of 3D points into the rectified images.

2.3 Point cloud projection

The ground truth disparity maps used for both training and evaluation are computed from the LiDAR point cloud. The perspective projection model is used to project the 3D point cloud into image plane. The disparity d is defined in Equation (1).

$$d = x_l - x_r \tag{1}$$

where x_l and x_r are the projection coordinate on x axis in the image plane after projecting the 3D point cloud to the left and right images.

Because the point cloud is sparse, it is difficult to predict the occlusions which can be important (Bevilacqua et al., 2017). An example of occlusion (in the left image) is shown in Figure 1. Some points from the ground area should be removed as they are occluded by the roof.

According to the epipolar geometry, the disparity is related to the depth of object (Jain et al., 1995), in Equation (2), z is the depth, b is base line length, f is the focal length, d is the disparity. The disparity is inversely proportional to the depth.

$$z = \frac{b \cdot f}{d} \tag{2}$$

For the aerial images, the disparity is related to the elevation, such that ground points have a larger disparity than points on the roof. Similarly as (Biasutti et al., 2019), we use a filtering based on the density to remove occluded points: if the difference between the disparity and its median is larger than a threshold, then the point is removed. This filter removes the occluded points quite effectively, as shown in Figure 2.

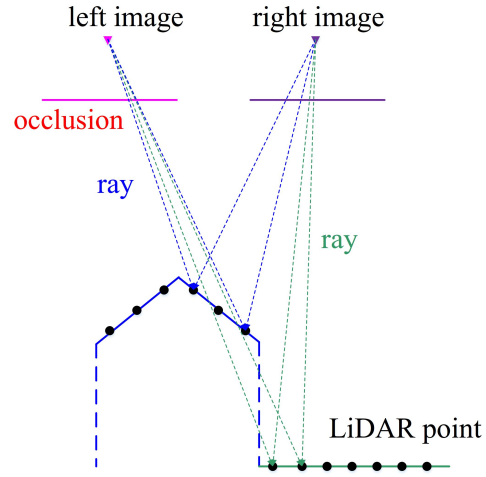
2.4 Benchmark data production

After the disparity map generation, the benchmark dataset is split into a training and testing sets. In our experiment, we split the dataset according to the LiDAR point cloud area, as shown in Figure 3. Similarly to the KITTI dataset, the final data are 1024x1024 cropped images with 8bit depth color band, and the disparity images are stored on 16bits with the disparity value scaled by 256. After cropping the left image, the offset of the right image is set to the average disparity in this area in order to avoid too large disparity values. If the footprint of a cropped image intersects the training area, it is attributed to the training dataset, otherwise to the testing dataset. An example is shown in Figure 4. Using this procedure, our training set contains 585 image pairs while the testing set has 507 image pairs. In the training dataset there are 337 stereo pairs along the flight strip (along-strip),

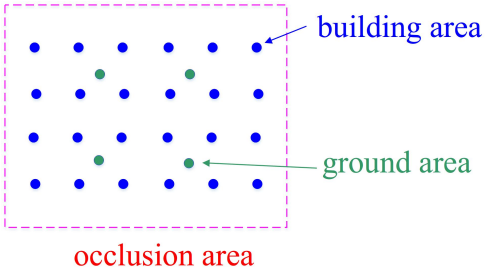
[1] The IARPA dataset was published in 2016.

Data	Year	Scene	Stereo number	Disparity density	Citation
Middlebury *V2*	2003-2006	indoor	32	dense	(Scharstein, Szeliski, 2003)
Middlebury *V3*	2014	indoor	33	dense	(Scharstein et al., 2014)
ETH3D	2017	indoor + outdoor	47	dense	(Schops et al., 2017)
KITII*2012*	2012	outdoor(driving)	389	sparse	(Geiger et al., 2012)
KITTI*2015*	2015	outdoor(driving)	1600	sparse	(Mayer et al., 2016)
Drivingstereo	2019	outdoor(driving)	182188	sparse	(Yang et al., 2019b)
ApolloScape	2019	outdoor(driving)	5165	sparse	(Huang et al., 2019)
SatStereo	2019[1]	satellite	72	dense	(Patil et al., 2019)
DFC2019	2019	satellite	8634	dense	(Bosch et al., 2019)
Ours	–	aerial	1092	sparse	–

Table 1. Benchmark datasets for disparity estimation. Stereo number is the total number of stereo pairs used for both training and testing.



(a) Lidar point occluded in an image



(b) LiDAR points projected in the image

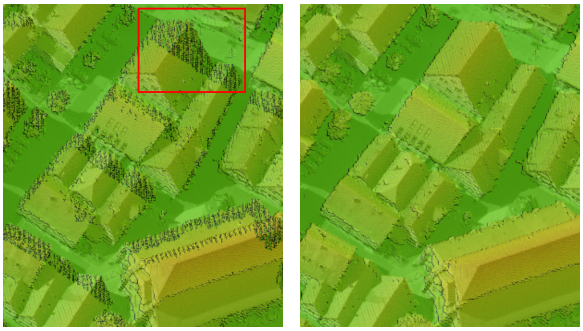
Figure 1. Occlusion in projection.

and 248 stereo pairs are across the flight strip (across-strip). In the testing dataset there are 323 stereo pairs along-strip and 184 across-strip, and they will be used to evaluate the impact of the base to height ratio (B/H) as the B/H is different in these two settings. More detailed information can be found in https://github.com/whuwuteng/benchmark_ISPRS2021.

3. EVALUATION

After generating the epipolar image pairs and the corresponding ground truth disparity images from LiDAR, we evaluate several traditional and learning based methods on this dataset:

1. *MICMAC*: A variant of SGM implemented in MicMac (Pierrot-Deseilligny, Paparoditis, 2006), using NCC as similarity feature, the code is written in C++ (micmacIGN, 2020).



(a) Disparity without occluded points filtering shown after nearest interpolation.

(b) Disparity with occluded points filtering shown after nearest interpolation

Figure 2. Occlusion removal.

2. *SGM (GPU)*: A variant of SGM based on GPU (Hernandez-Juarez et al., 2016), using census as similarity feature, the code is based on C++ and CUDA (Hernandez-Juarez, 2020).

3. *GraphCuts*: A Graphcut based method (Taniai et al., 2017), using intensity and gradient consistencies, and using plane as a constraint, the code is written in C++ (Taniai, 2020).

4. *CBMV*: A coalesced bidirectional matching volume based method (Batsos et al., 2018) using a random forest (RF) to fuse the features, followed by dynamic programming optimization (*CBMV (SGM)*) or Graphcut (*CBMV (GraphCuts)*) which has the same implementation with *GraphCuts*. The code is based on C++, CUDA (SGM implementation) and Python (Batsos, 2020).

5. *DeepFeature*: A deep learning network is used to obtain the features, then optimize with dynamic programming (Luo et al., 2016), the code is based on C++, CUDA (SGM implementation) and Lua torch (Luo, 2020).

6. *PSM net*: A pyramid stereo matching network using a spatial pyramid pooling and a 3D CNN to train an end-to-end model (Chang, Chen, 2018). The code is based on Pytorch (Chang, 2020).

7. *HRS net*: A high resolution stereo network structure improving the accuracy without downscaling the cost volume

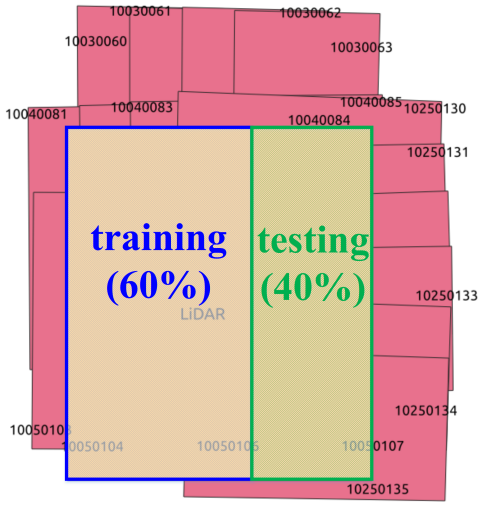


Figure 3. Training and testing area on Vahingen.

(Yang et al., 2019a). The code is based on Pytorch (Yang, 2020).

8. *DeepPruner*: A 3D cost volume is pruned with a differential patch match method to avoid a full cost volume calculation and evaluation (Duggal et al., 2019). The code is based on Pytorch (UberResearch, 2020).

3.1 Experimental setup

The disparity search range is an important parameter for stereo dense matching. Some methods do not need this parameter, i.e., *MICMAC* and *DeepPruner*. In *SGM(GPU)*, the range is set to 128 and is dictated by the implementation. For other methods, it is set to 192.

For machine learning based methods, the training data and hyper-parameters impact significantly the results. For the Random Forest based method *CBMV*, 54 epipolar pairs are used for training. For deep learning based methods, all the training data is used. For the evaluation, all the testing data is used for all methods.

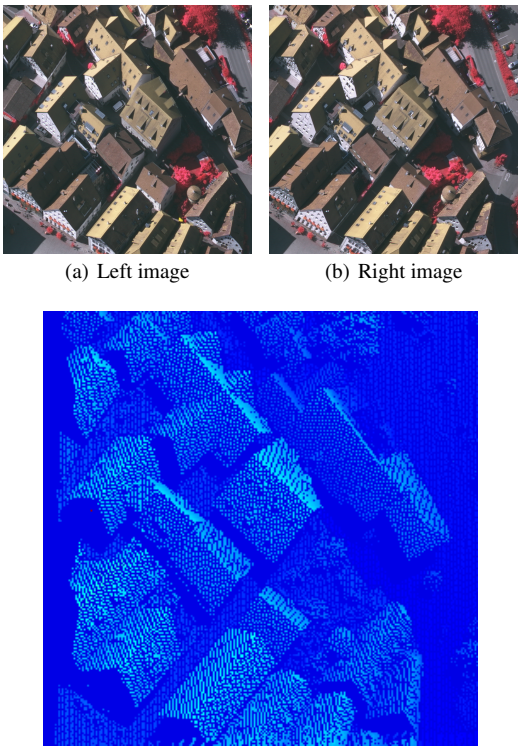
In our experiments, we compare deep learning methods pretrained on the KITTI dataset and fine tuned on our Vahingen ground truth disparity. We found that batch size has a strong impact on the memory requirements and on the accuracy, as shown in Table 2 for *PSM net*. We decided to use the default batch size proposed in the implementation: 12 for *PSM net*, 28 for *HRS net* and 16 for *DeepPruner*. For the fine-tuning experiments on Vaihingen dataset, we did the same for the number of epochs: 20 for *DeepFeature*, 500 for *PSM net*, 700 for *HRS net* and 900 for *DeepPruner*.

Table 2. Influence of the batch for *PSM net*.

Method	Batch size	Accuracy[%]		
		<2-pixel	<3-pixel	<5-pixel
PSM net	3	81.988	87.022	91.501
	12	**84.065**	**88.324**	**92.395**

3.2 Pixel error

In stereo dense matching evaluation, the pixel error (error between the estimated disparity and the ground truth) is an important metric to analyze the performance. In our experiment, we compute the 2, 3 and 5-pixel error. Accuracy is the percentage of positive pixels in valid pixels within the ground truth.



(a) Left image (b) Right image

(c) Disparity map

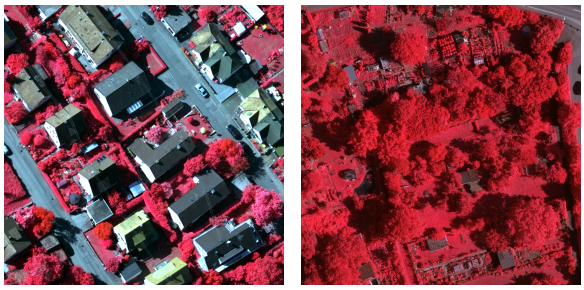
Figure 4. An example of the dataset. In Figure 4(c), pixels with valid disparity values are displayed in bright.

Table 3. Evaluation of methods on testing data. For *SGM(GPU)*, only evaluate disparity smaller than 128, for other methods maximum disparity is 192.

Method	Accuracy[%]		
	<2-pixel	<3-pixel	<5-pixel
MICMAC	67.169	74.283	81.429
SGM(GPU)	71.564	78.539	84.799
GraphCuts	71.704	76.404	80.951
CBMV(SGM)	74.941	80.540	85.342
CBMV(GraphCuts)	76.387	82.229	87.227
DeepFeature	78.265	83.982	88.878
PSM net	**84.065**	**88.324**	**92.395**
HRS net	79.135	85.243	91.238
DeepPruner	83.568	87.893	92.223

As shown in Table 3, the result shows that machine learning based methods have a significantly better performance than traditional methods. While slower, Graphcut optimization achieves better results than dynamic programming. *PSM net* has the best result among all tested deep learning methods.

As for the deep learning methods, the data used for learning highly influences the results. A comparison between the results obtained with a pretrained model and with a model finetuned on our learning set is provided in Table 4. This experiment shows that the dependency of deep learning methods on the training data depends on the method. Comparing to *DeepFeature* which



(a) Building area (b) Tree area

Figure 5. Two examples of the testing data: building area and tree area.

is a hybrid method, end-to-end methods depend more on the training data.

Table 4. Evaluation of deep learning based methods on fine tune dataset.

Method	Training data	Accuracy[%] <2-pixel	<3-pixel	<5-pixel
DeepFeature(Pre)	KITTI2015	**70.888**	**78.918**	**82.921**
DeepFeature	Vaihingen	78.265	83.982	88.878
PSM net(Pre)	KITTI2015	38.532	62.589	80.801
PSM net	Vaihingen	**84.065**	**88.324**	**92.395**
HRS net(Pre)	KITTI2015	69.009	78.462	86.692
HRS net	Vaihingen	79.135	85.243	91.238
DeepPruner(Pre)	KITTI(full)	52.278	63.242	73.407
DeepPruner	Vaihingen	83.568	87.893	92.223

In our experiments, we found that the base height ratio B/H of the epipolar stereo pairs also influences the result, as shown in Table 5. Images along-strip are usually used for dense matching because they have a large overlap and small B/H. Images across-strip have a larger B/H which can increase the intersection accuracy, but leads to more errors because of a larger perspective distortion and more occlusions (Tola et al., 2008).

Table 5. Base height ratio for the testing dataset.

image type	base height ratio(B/H) minimum	maximum	average
along-strip	0.225	0.231	0.229
across-strip	0.380	0.390	0.385

In order to investigate the effect of the B/H, we experimented two different trainings of the model: using all the training dataset and only the along-strip images. As shown in Table 6, the result in across-strip is not as good as along-strip. For the along-strip based training, the dataset size is smaller than the all-based training, but there is no big difference on the results of along-strip testing data. This means that the along-strip images are sufficient for the fine-tuning. However, using images across-strip to fine-tune can improve the performance on across-strip pairs. Especially, the 1-pixel error can be improved significantly for all the methods.

3.3 Error analysis

Pixel error is a statistical metric. When the scene is complex, errors can show different patterns on buildings (man-made structure), ground, vegetation and so on. To visualize this, in Figure 5 we provide the error maps on different scenes. Because the ground truth is sparse, the error maps are interpolated with the nearest point in order to facilitate the interpretation. The disparity map is visualized using an ambient occlusion shading implemented in MicMac as shown in Figure 6. Figures 6(p) and 6(r) demonstrate that the Graphcut based methods are smooth on building's roofs. Figures 6(k) to 6(m) demonstrate that the deep learning based methods perform well on building's boundary (disparity discontinuity).

DeepPruner proves to have the second best result among these methods, but the pre-trained result is poor as shown in Figure 7(i), which means that *DeepPruner* is highly dependent on the training dataset. Because there is a plane constraint in the *GraphCuts* method, the result are not satisfactory on trees, as shown in Figure 7(c).

4. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new stereo benchmark dataset adapted to deep learning and evaluate some existing traditional and learning based methods on this dataset. Experiments show that:

- Learning based methods perform better than traditional methods.
- Fine-tuning deep learning architectures by transfer learning on a specific dataset improves the results significantly.
- Along-strip images and across-strip images should be considered in training.

Our future work will focus on extending the benchmark to satellite images, but also studying how training on different scenes and sensor types affect the performance of learning-based methods.

ACKNOWLEDGEMENTS

This research has been funded by AI4GEO project.The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF).

REFERENCES

Batsos, K., 2020. Cbmv. https://github.com/kbatsos/CBMV.

Batsos, K., Cai, C., Mordohai, P., 2018. Cbmv: A coalesced bidirectional matching volume for disparity estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2060–2069.

Bethmann, F., Luhmann, T., 2015. Semi-Global Matching in Object Space. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Bevilacqua, M., Aujol, J.-F., Biasutti, P., Brédif, M., Bugeau, A., 2017. Joint inpainting of depth and reflectance with visibility estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 125, 16–32.

Table 6. Evaluation of deep learning methods on different stereo pair modes.

Method	Training data	Testing data	Accuracy[%]			
			<1-pixel	<2-pixel	<3-pixel	<5-pixel
DeepFeature	Vaihingen(along-strip)	Vaihingen(along-strip)	69.552	82.135	86.723	90.588
DeepFeature	Vaihingen(along-strip)	Vaihingen(across-strip)	33.781	65.030	75.806	83.247
DeepFeature	Vaihingen(all)	Vaihingen(along-strip)	67.854	81.529	86.588	90.750
DeepFeature	Vaihingen(all)	Vaihingen(across-strip)	52.742	71.710	78.750	85.119
PSM net	Vaihingen(along-strip)	Vaihingen(along-strip)	**76.609**	85.402	89.178	92.837
PSM net	Vaihingen(along-strip)	Vaihingen(across-strip)	44.010	74.271	81.456	87.230
PSM net	Vaihingen(all)	Vaihingen(along-strip)	76.054	**85.392**	**89.350**	**93.067**
PSM net	Vaihingen(all)	Vaihingen(across-strip)	**69.389**	**81.399**	**86.264**	**91.046**
HRS net	Vaihingen(along-strip)	Vaihingen(along-strip)	68.510	80.904	86.603	92.093
HRS net	Vaihingen(along-strip)	Vaihingen(across-strip)	39.744	68.830	78.006	86.036
HRS net	Vaihingen(all)	Vaihingen(along-strip)	68.856	81.140	86.809	92.298
HRS net	Vaihingen(all)	Vaihingen(across-strip)	58.567	75.118	82.099	89.110
DeepPruner	Vaihingen(along-strip)	Vaihingen(along-strip)	75.780	84.775	88.669	92.629
DeepPruner	Vaihingen(along-strip)	Vaihingen(across-strip)	40.539	73.143	80.525	86.683
DeepPruner	Vaihingen(all)	Vaihingen(along-strip)	76.045	84.961	88.972	92.937
DeepPruner	Vaihingen(all)	Vaihingen(across-strip)	68.926	80.772	85.728	90.790

Biasutti, P., Bugeau, A., Aujol, J.-F., Brédif, M., 2019. Visibility estimation in point clouds with variable density. *International Conference on Computer Vision Theory and Applications (VISAPP)*.

Bosch, M., Foster, K., Christie, G., Wang, S., Hager, G. D., Brown, M., 2019. Semantic stereo for incidental satellite images. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 1524–1532.

Bosch, M., Kurtz, Z., Hagstrom, S., Brown, M., 2016. A multiple view stereo benchmark for satellite imagery. *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, IEEE, 1–9.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9), 1124–1137.

Cavegn, S., Haala, N., Nebiker, S., Rothermel, M., Tutzauer, P., 2014. Benchmarking high density image matching for oblique airborne imagery. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(3), 45.

Chang, J.-R., 2020. Psmnet. https://github.com/JiaRenChang/PSMNet.

Chang, J.-R., Chen, Y.-S., 2018. Pyramid stereo matching network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5418.

Cho, W., Jwa, Y.-S., Chang, H.-J., Lee, S.-H., 2004. Pseudo-grid based building extraction using airborne LIDAR data. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 35, 378–381.

Cournet, M., Sarrazin, E., Dumas, L., Michel, J., Guinet, J., Youssefi, D., Defonte, V., Fardet, Q., 2020. Ground Truth Generation and Disparity Estimation for Optical Satellite Imagery. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 127–134.

Duggal, S., Wang, S., Ma, W.-C., Hu, R., Urtasun, R., 2019. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. *Proceedings of the IEEE International Conference on Computer Vision*, 4384–4393.

Flötotto, J., 2020. 2D and surface function interpolation. *CGAL User and Reference Manual*, 5.2 edn, CGAL Editorial Board.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 3354–3361.

Hernandez-Juarez, D., 2020. sgm. https://github.com/dhernandez0/sgm.

Hernandez-Juarez, D., Chacón, A., Espinosa, A., Vázquez, D., Moure, J. C., López, A. M., 2016. Embedded real-time stereo estimation via semi-global matching on the GPU. *Procedia Computer Science*, 80, 143–153.

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

Hirschmuller, H., Scharstein, D., 2007. Evaluation of cost functions for stereo matching. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1–8.

Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., Yang, R., 2019. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10), 2702–2719.

Jain, R., Kasturi, R., Schunck, B. G., 1995. *Machine vision*. 5, McGraw-hill New York.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H., 2014. Large scale multi-view stereopsis evaluation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 406–413.

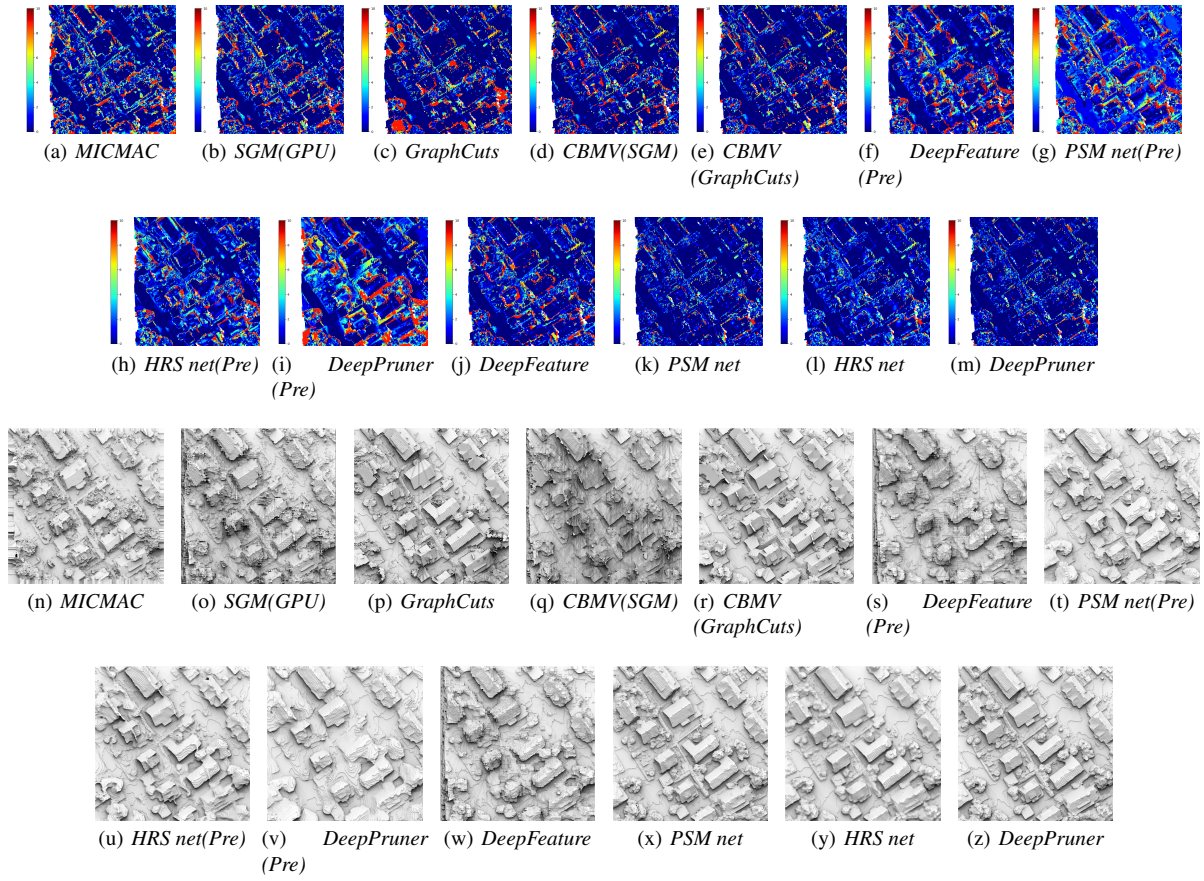


Figure 6. Error map and disparity visualization on building area. From (a)-(m) is the error map, and from (n)-(z) is the disparity map visualization. The left image is shown in Figure 5(a).

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A., 2017. End-to-end learning of geometry and context for deep stereo regression. *Proceedings of the IEEE International Conference on Computer Vision*, 66–75.

Knöbelreiter, P., Vogel, C., Pock, T., 2018. Self-supervised learning for stereo reconstruction on aerial images. *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 4379–4382.

Laga, H., Jospin, L. V., Boussaid, F., Bennamoun, M., 2020. A Survey on Deep Learning Techniques for Stereo-based Depth Estimation. *arXiv preprint arXiv:2006.02535*.

Li, Y., Huttenlocher, D. P., 2008. Learning for stereo vision using the structured support vector machine. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1–8.

Luo, W., 2020. cvpr16_stereo_public. https://bitbucket.org/saakuraa/cvpr16_stereo_public/src/master/.

Luo, W., Schwing, A. G., Urtasun, R., 2016. Efficient deep learning for stereo matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5695–5703.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T., 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4040–4048.

micmacIGN, 2020. micmac. https://github.com/micmacIGN/micmac.

Patil, S., Comandur, B., Prakash, T., Kak, A. C., 2019. A new stereo benchmarking dataset for satellite images. *arXiv preprint arXiv:1907.04404*.

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

Pierrot-Deseilligny, M., Paparoditis, N., 2006. A multiresolution and optimization-based image matching approach: An application to surface reconstruction from spot5-hrs stereo imagery. *ISPRS Workshop On Topographic Mapping From Space*, 36(1), Ankara, Turkey.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1), 293–298.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., Westling, P., 2014. High-resolution stereo datasets with subpixel-accurate ground truth. *German conference on pattern recognition*, Springer, 31–42.

Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3), 7–42.

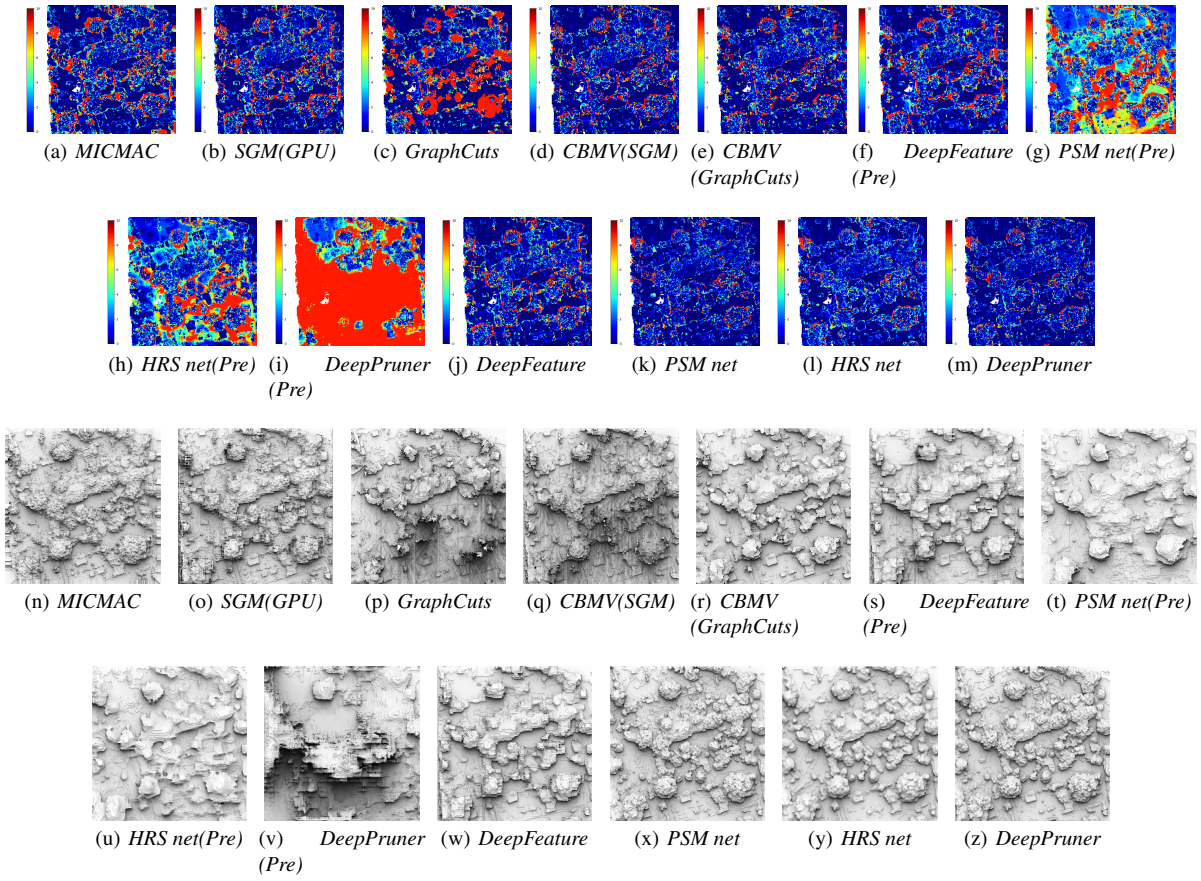


Figure 7. Error map and disparity visualization on tree area. From (a)-(m) is the error map, and from (n)-(z) is the disparity map visualization. The left image is shown in Figure 5(b).

Scharstein, D., Szeliski, R., 2003. High-accuracy stereo depth maps using structured light. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 1, IEEE, I–I.

Schops, T., Schonberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A., 2017. A multi-view stereo benchmark with high-resolution images and multi-camera videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3260–3269.

Seki, A., Pollefeys, M., 2017. Sgm-nets: Semi-global matching with neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 231–240.

Sun, J., Zheng, N.-N., Shum, H.-Y., 2003. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7), 787–800.

Taniai, T., 2020. Localexpstereo. https://github.com/t-taniai/LocalExpStereo.

Taniai, T., Matsushita, Y., Sato, Y., Naemura, T., 2017. Continuous 3D label stereo matching using local expansion moves. *IEEE transactions on pattern analysis and machine intelligence*, 40(11), 2725–2739.

Tola, E., Lepetit, V., Fua, P., 2008. A fast local descriptor for dense matching. *2008 IEEE conference on computer vision and pattern recognition*, IEEE, 1–8.

Tombari, F., Mattoccia, S., Di Stefano, L., Addimanda, E., 2008. Classification and evaluation of cost aggregation methods for stereo correspondence. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1–8.

UberResearch, 2020. Deeppruner. https://github.com/uber-research/DeepPruner.

Van Meerbergen, G., Vergauwen, M., Pollefeys, M., Van Gool, L., 2002. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47(1), 275–285.

Yang, G., 2020. high-res-stereo. https://github.com/gengshan-y/high-res-stereo.

Yang, G., Manela, J., Happold, M., Ramanan, D., 2019a. Hierarchical deep stereo matching on high-resolution images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5515–5524.

Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., Zhou, B., 2019b. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 899–908.

Žbontar, J., LeCun, Y., 2016. Stereo matching by training a convolutional neural network to compare image patches. *The journal of machine learning research*, 17(1), 2287–2318.