A Novel Recurrent Encoder-Decoder Structure for Large-Scale Multi-view Stereo Reconstruction from An Open Aerial Dataset

Jin Liu and Shunping Ji*
School of Remote Sensing and Information Engineering, Wuhan University
{liujinwhu, jishunping}@whu.edu.cn

Abstract

A great deal of research has demonstrated recently that multi-view stereo (MVS) matching can be solved with deep learning methods. However, these efforts were focused on close-range objects and only a very few of the deep learning-based methods were specifically designed for large-scale 3D urban reconstruction due to the lack of multi-view aerial image benchmarks. In this paper, we present a synthetic aerial dataset, called the WHU dataset, we created for MVS tasks, which, to our knowledge, is the first large-scale multi-view aerial dataset. It was generated from a highly accurate 3D digital surface model produced from thousands of real aerial images with precise camera parameters. We also introduce in this paper a novel network, called RED-Net, for wide-range depth inference, which we developed from a recurrent encoder-decoder structure to regularize cost maps across depths and a 2D fully convolutional network as framework. RED-Net's low memory requirements and high performance make it suitable for large-scale and highly accurate 3D Earth surface reconstruction. Our experiments confirmed that not only did our method exceed the current state-of-the-art MVS methods by more than 50% mean absolute error (MAE) with less memory and computational cost, but its efficiency as well. It outperformed one of the best commercial software programs based on conventional methods, improving their efficiency 16 times over. Moreover, we proved that our RED-Net model pre-trained on the synthetic WHU dataset can be efficiently transferred to very different multi-view aerial image datasets without any fine-tuning. Dataset and code are available at `http://gpcv.whu.edu.cn/data`.

1. Introduction

Large-scale and highly accurate 3D reconstruction of the Earth's surface, including cities, is mainly realized from dense matching of multi-view aerial images implemented and dominated by commercial software such as Pix4D [24], Smart3D [8], and SURE [27], all of which were developed from conventional methods [33, 3, 13]. Recent attempts at multi-view stereo (MVS) matching with deep learning methods are found in the literature [14, 16, 36, 37, 15]. While these deep learning approaches can produce satisfactory results on close-range object reconstruction, they have two critical limitations when applied to Earth surface reconstruction from multi-view aerial images. The first limitation is the lack of aerial dataset benchmarks, which makes it difficult to train, discover, and improve the appropriate networks through between-method comparison. In addition, most of the existing MVS datasets are images of laboratory, and models trained on them cannot be satisfactorily transferred to a bird's eye view of a terrestrial scene. The second limitation of these methods is their high GPU memory demand in recent MVS networks [36, 15, 25, 34], which makes them less suitable for large-scale and high-resolution scene reconstruction. The state-of-the-art R-MVSNet method [37] has achieved depth inference with unlimited depth-wise resolution, however, the resolution quality of its results is not high as the output depth map is down-sampled four times.

In this paper, we present a synthetic aerial dataset we created for large-scale MVS matching and Earth surface reconstruction. Each image in the dataset was simulated from a complete and accurate 3D urban scene produced from a real multi-view aerial image collection with software and careful manual editing. The dataset includes thousands of simulated images covering an area of 6.7×2.2 km^2, along with the ground truth depth and camera parameters for multi-view images, as well as disparity maps for rectified epipolar images. Due to the large size of the aerial images (5376×5376 pixels), there are subsets provided consisting of cropped sub-blocks that can be used directly for training CNN models on a single GPU. Note that the simulated camera parameters are unbiased and the provided ground truths are absolutely complete even in occluded regions, which ensures the accuracy and reliability of the dataset for detailed 3D reconstruction.

*Corresponding author

We also introduce in this paper an MVS network, called RED-Net, we created for large scale MVS matching. A recurrent encoder-decoder (RED) architecture is utilized to sequentially regularize cost maps obtained from a series of convolutions on multi-view images. When compared to the state-of-the-art method [37], we achieved higher efficiency and accuracy using less GPU memory while maintaining unlimited depth resolution, which is beneficial to city-scale reconstruction. Our experiments confirmed that RED-Net outperformed all the comparable methods evaluated on the WHU aerial dataset.

We had a third aim for our work beyond addressing the two limitations of the existing methods. That goal was to demonstrate that our MVS network could be generalized for cross-dataset transfer learning. We demonstrate here that RED-Net pre-trained on our WHU dataset could be directly applied on another quite different aerial dataset with slightly better accuracy than one of the best commercial software programs with efficiency improved 16 times over.

2. Related Work

2.1. Datasets

Two-view datasets. Middlebury [28] and KITTI [9] are two popular datasets for stereo disparity estimation. However, these datasets are too small for current applications, especially when training deep learning models, and the lack of sufficient samples often leads to overfitting and low generalization. Considering this situation, [21] created a large synthetic dataset that consists of three subsets: FlyingThings3D, Monkaa, and Driving, which provide thousands of stereo images with dense and complete ground truth disparities. However, a model pre-trained on this synthetic dataset cannot easily be applied to a real scene dataset due to the heterogeneous data sources.

Multi-view datasets. The Middlebury multi-view dataset [31] was designed for evaluating MVS matching algorithms on equal ground and is a collection of calibrated image sets from only two small scenes in a laboratory environment. The DTU dataset [1] is a large scale close-range MVS benchmark that contains 124 scenes with a variety of objects and materials under different lighting conditions, which make it well-suited for evaluating advanced methods. The Tanks and Temples benchmark [18] provides high-resolution data with large-size images acquired in complex outdoor environments. A recent benchmark called ETH3D [30] was created for high-resolution stereo and multi-view reconstruction, which consists of artificial scenes and outdoor and indoor scenes and represents various real-world reconstruction challenges.

Reconstructing the Earth's surface and cities is mainly realized with matching multi-view aerial images. The ISPRS Association and the EuroSDR Center jointly provided two small aerial datasets called München and Vaihingen [11], which consist of dozens of aerial images; however, these datasets are currently not publicly accessible. In our work, we created a large-scale synthetic aerial dataset with accurate camera parameters and complete ground truths for MVS method evaluation and urban scene reconstruction.

2.2. Networks

Inspired by the success of the deep learning based stereo methods [23, 17, 38, 4], some researchers attempted to apply CNNs to the MVS task. Hartmann et al. [12] proposed an N-way Siamese network to learn the similarity score over a set of multi-patches. The first end-to-end learning network designed for MVS was SurfaceNet [15] by building colored voxel cubes outside the network to encode the camera parameters through perspective projection, which combined multi-view images to a single cost volume. The Learnt Stereo Machine (LSM) [16] ensures end-to-end MVS reconstruction by differentiable projection and unprojection operations. The features are unprojected into 3D feature grids with known camera parameters, and 3D CNN then is used to detect the surface of the 3D object in the voxel. Both SurfaceNet and LSM utilize volumetric representation; nevertheless, they only reconstruct low-resolution objects and have a huge GPU memory consumption of 3D voxel; for example, they created the world grid at a resolution of $32 \times 32 \times 32$.

3D cost volume has its advantage in encoding camera parameters and image features. DeepMVS [14] generates a plane-sweep volume for each reference image, and an encoder-decoder structure with skip connections is used to aggregate the cost and estimate depths with fully-connected conditional random field (Dense-CRF) [19]. [36] built a 3D cost volume by differentiable homography warping. Its memory requirement grows cubically with the depth quantization number, which makes it unrealistic for large scale scenes. The state-of-the-art method, R-MVSNet [37], regularized 2D cost maps sequentially across depths via a convolutional gated recurrent unit (GRU) [5] instead of 3D CNNs, which reduced the memory consumption and made high-resolution reconstruction possible. However, R-MVSNet regularized the cost maps with a small 3×3 receptive field in the GRUs and down-sampled the output depth four times, which resulted in contextual information loss and coarse reconstruction.

Our RED-Net approach follows the idea of sequentially processing 2D features along the depth direction for wide-depth range inference. However, we introduce a recurrent encoder-decoder architecture to regularize the 2D cost maps rather than simply stacking the GRU blocks as in [37]. The RED structure provides multi-scale receptive fields

Figure 1: The dataset. Area 0: the complete dataset consists of 1,776 virtual aerial images each 5376×5376 pixels in size. For facilitating machine learning methods, areas 1/4/5/6 were allocated for the training set, which consisted of 261 images. Areas 2 and 3, which consisted of 93 images, were used as the test set. In the training and testing area, the images also were cropped into tiles of 768×384 pixel-size for a single GPU.

to exploit neighborhood information effectively in fine resolution scenes, which allows us to achieve large-scale and full-resolution reconstruction with higher accuracy and efficiency and lower memory requirements.

3. WHU Dataset

This section describes the synthetic aerial dataset we created for large-scale and high-resolution Earth surface reconstruction call the WHU dataset. The aerial images in the dataset were simulated from a 3D surface model that was produced by software and refined by manual editing. The dataset includes a complete aerial image set and cropped sub-image sets for facilitating deep learning.

3.1. Data Source

A 3D digital surface model (DSM) with OSGB format [35] was reconstructed using Smart3D software [8] from a set of multi-view aerial images captured from an oblique five-view camera rig mounted on an unmanned aerial vehicle (UAV). One camera was pointed straight down and the optical axis of the other four surrounding cameras was at a 40° tilt angle, which guaranteed most of the scenes, including the building façade, could be well captured. We manually edited some errors in the surface model to improve its resemblance to the real scene. The model covered an area of about 6.7×2.2 km^2 over Meitan County, Guizhou Province in China with about 0.1 m ground resolution. The county contains dense and tall buildings, sparse factories, mountains covered with forests, and some bare ground and rivers.

3.2. Synthetic Aerial Dataset

First, a discrete 3D points set on a $0.06 \times 0.06 \times 0.06$ m^3 grid covering the whole scene was generated by interpolating the OSGB mesh. Each point includes the object position *(X, Y, Z)* and the texture *(R, G, B)*.

Then, we simulated the imaging process of a single-lens camera. Given the camera's intrinsic parameters (focal length f, principal point x_0, y_0, image size W, H, and sensor size) and the exterior orientation (camera center *(Xs, Ys, Zs)* and three rotational angles (φ, ω, κ)). We projected the 3D discrete points onto the camera to obtain a virtual image, and the depth map was simultaneously retrieved from the 3D points. Note that the depth map was complete even on the building façade since the 3D model had full scene mesh. The virtual image was taken at 550 m above the ground with 10 cm ground resolution. A total of 1,776 images (5376×5376 in size) were captured in 11 strips with 90% heading overlap and 80% side overlap, with corresponding 1,776 depth maps as ground truth. We set the rotational angles at (0,0,0), and two adjacent images therefore could be regarded as a pair of epipolar images. A total of 1,760 disparity maps along the flight direction also were provided for evaluating the chosen stereo matching methods. We provided 8-bit RGB images and 16-bit depth maps with the lossless PNG format and text files that recorded the orientation parameters that included the camera center *(Xs, Ys, Zs)* and the rotational matrix **R**.

3.3. Sub-Dataset for Deep Learning

In addition to providing the complete dataset, we selected six representative sub-areas covering different scene types as training and test sets for deep learning methods, which are shown in Figure 1. "Area 1" is a flat suburb with large and low factory buildings. "Area 2" contains trees, roads, buildings, and open spaces. "Area 3" is a residential area with a mixture of low and high buildings. "Area 4" and "Area 5" are the town center covering dense buildings with complex rooftop structures. "Area 6" is a mountainous area covered by agricultural land and forests. A total of 261 virtual images of Areas 1/4/5/6 were used as the training set, and 93 images from Area 2

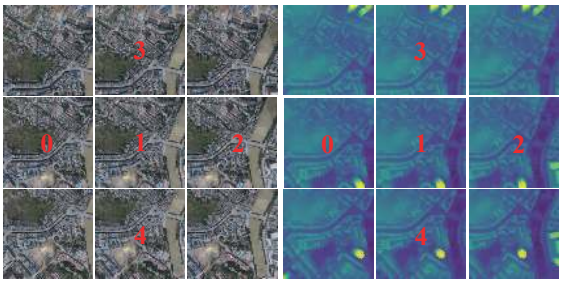
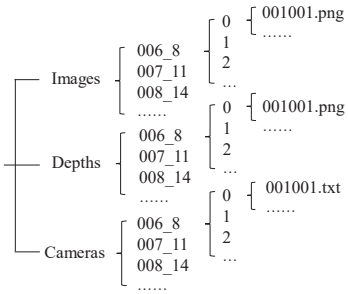


Figure 2: The images and depth maps from different viewpoints. A five-view unit took the Image with ID 1 as the reference image, the images with ID 0 and 2 in the heading direction and the images with ID 3 and 4 in the side strips as the search images. The three-view set consisted of images with ID 0, 1, and 2. In the stereo dataset, Image 1 and Image 2 were treated as a pair of stereo epipolar images.

(a)



(b)

Figure 3: (a) A five-view sub-set with size of 768×384 pixels. The three sub-images in red rectangle comprise the three-view set. (b) The organization of images, depths, and camera files in the MVS dataset.

and Area 3 comprised the test set. The ratio of the training to the test set was roughly 3:1. For a direct application of the deep learning-based MVS methods on the sub-dataset, we additionally provided a multi-view and a stereo sub-set by cropping the virtual aerial images into sub-blocks as an image of 5376×5376 pixels may not be fed into a current single GPU.

Multi-view Dataset. A multi-view unit consists of five images as shown in Figure 2. The central image with ID 1 was treated as the reference image, and the images with ID 0 and 2 in the heading direction and the images with ID 3 and 4 in the side strips were the search images. We cropped the overlapped pixels into the sub-block at a size of 768×384 pixels. A five-view unit yielded 80 pairs (400 sub images) (Figure 3(a)). The depth maps were cropped at the same time. The dataset was ultimately organized as Figure 3(b). The virtual images, depth maps, and camera parameters were in the first level folder. The second level folders took the name of the reference image in a five-view unit; for example, 006_8 represented the eighth image in the sixth strip. The five sub-folders were named as 0/1/2/3/4 to store the sub images generated from the five-view virtual images respectively. In addition, there was a three-view dataset that consisted of the images with ID 0, 1, and 2.

Stereo Dataset. Each adjacent image pair in a strip was also epipolar images. Similar to the multi-view set, we cropped each image and disparity map into 768×384 pixels and obtained 154 sub-image pairs in a two-view unit.

4. RED-Net

We developed a network, which we named RED-Net, that combines a series of weight-shared convolutional layers that extract the features from separate multi-view images and recurrent encoder-decoder (RED) structures that sequentially learn regularized depth maps across both the depth and spatial directions for large-scale and high-resolution multi-view reconstruction. The framework was inspired by [37]. However, instead of using a stack of three GRU blocks, we utilized a 2D recurrent encoder-decoder structure to sequentially regularize the cost maps, which not only significantly reduced the memory consumption and greatly improved the computational efficiency, but also captured the finer structures for depth inference. The output of RED-Net has the same resolution as the input reference images rather than being downsized by four as in [37], which ensures high-resolution reconstruction for large-scale and wide depth range scenes. The network structure is illustrated in Figure 4.

2D Feature Extraction. RED-Net infers a depth map with depth sample number D from N-view images where N is typically no less than three. The 2D convolution layers first are separately used to extract the features of the N input images with shared weights, which can be seen as an N-way Siamese network architecture [6]. Each branch consists of five convolutional layers with 8, 8, 16, 16, 16 channels, respectively, and a 3×3 kernel size and a stride of 1 (except for the third layer, which has a 5×5 kernel size and a stride of 2). All of the layers are followed by a rectified linear unit (ReLU) [10] except for the last layer. The 2D network yields 16-channel feature representations for each input image half the width and height of the input image.

Cost Maps. A group of 2D image features are back-projected onto successive virtual planes in 3D space to build cost maps. The plane sweep methods [7] were adopted to warp these features into reference camera viewpoint, which is described as differentiable homography warping

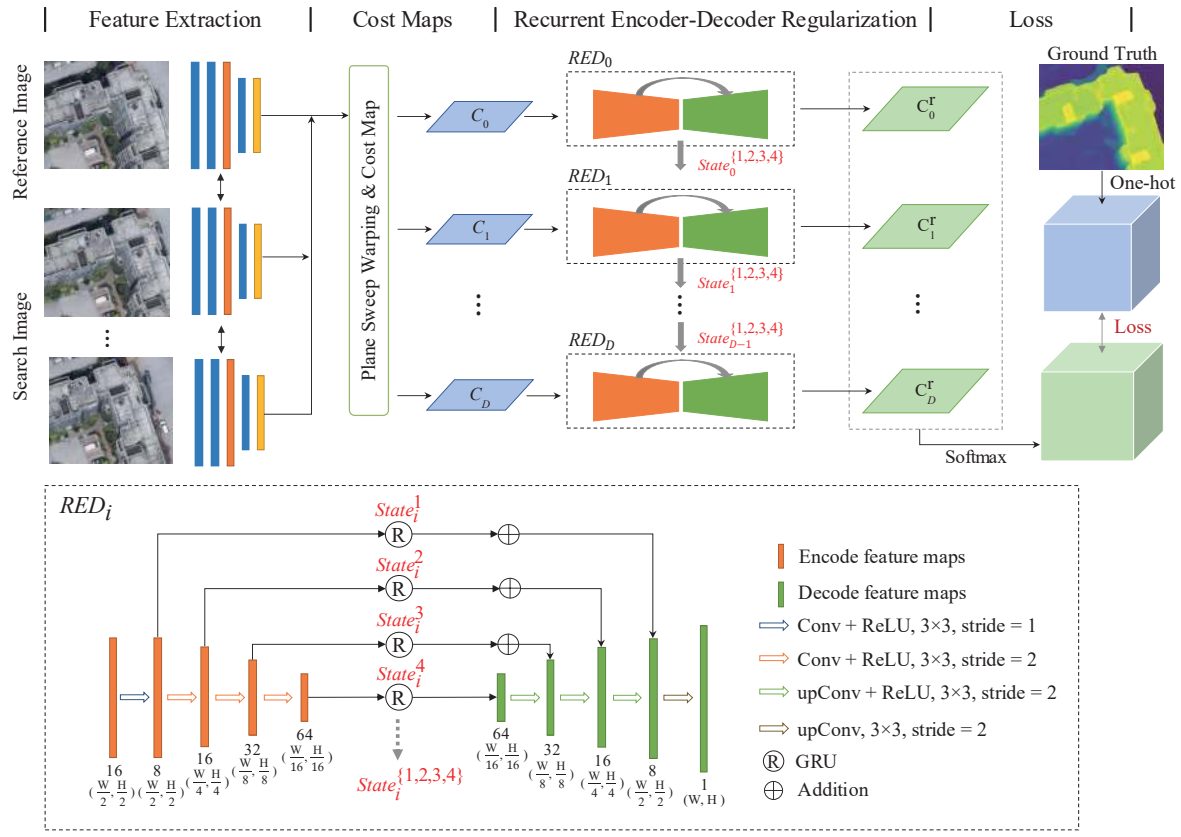


Figure 4: The structure of the RED-Net. W, H, and D are the image width, height, and depth sample number, respectively.

in [36, 37]. The variance operation [36] was adopted to concatenate multiple feature maps to one cost map at a certain depth plane in 3D space. Finally, D cost maps are built at each depth plane.

Recurrent Encoder-Decoder Regularization. Inspired by the U-Net [26], GRU [5], and RCNN [2], in this paper we introduce a recurrent encoder-decoder architecture to regularize the D cost maps that are obtained from the 2D convolutions and plane sweep methods. In the spatial dimension, one cost map C_i is the input to the recurrent encoder-decoder structure at a time, which is then processed by a four-scale convolutional encoder. Except for the first convolution layer with stride 1 and channel number 8, we doubled the feature channels at each downsampling step in the encoder. The decoder consists of three up-convolutional layers, and each layer expands the feature map generated by the previous layer and halves the feature channels. At each scale, the encoded feature maps are regularized by a convolutional GRU [37], which are then added to the corresponding feature maps at the same scale in the decoder. After the decoder, an up-convolutional layer is used to upsample the regularized cost maps to the input image size and reduce channel number to 1.

In the depth direction, the contextual information of the sequential cost maps is recorded in the previous regulated GRUs and transferred to current cost map C_i. There are four GRU state transitions in the laddered encoder-decoder structure, denoted as *state*, to gather and refine the contextual features in different spatial scales.

By regularizing the cost maps in the spatial direction and aggregating the geometric and contextual information in the depth direction by the recurrent encoder-decoder, RED-Net realized globally consistent spatial/contextual representations for multi-view depth inference. Compared to a stack of GRUs [37], our multi-scale recurrent encoder-decoder exploits multi-scale neighborhood information with more details and less parameters.

Loss computation. A cost volume is obtained by stacking all the regularized cost maps together. We turned it into a probability volume by utilizing a *softmax* operator along the depth direction as accomplished in previous works [17]. From this probability volume, the depth value can be estimated pixel-wise and compared to the ground truth with the cross-entropy loss, which is the same as [37].

To maintain an end-to-end manner, we did not provide a post-processing process. The inferred depth maps are translated into dense 3D points according to the camera parameters, all of which constitute the complete 3D scene. However, many classic post-processing methods [22] can be applied for refinement.

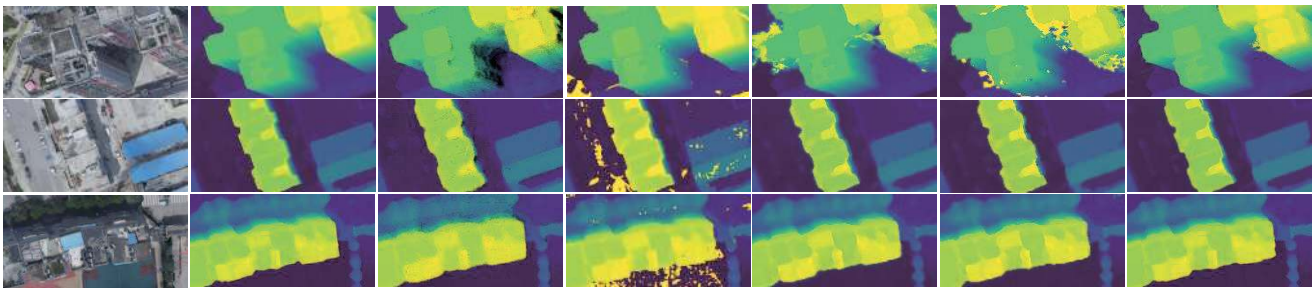


Figure 5: The inferred depth maps of three sub-units in the WHU test set. Our method produced the finest depth maps.

5. Experiments

5.1. Experimental Settings and Results

We evaluated our proposed RED-Net on our WHU dataset and compared it to several recent MVS methods and software, including COLMAP [29] and commercial software SURE [27] (aerial version for trial [32]), which are based on conventional methods, and the MVSNet [36] and R-MVSNet [37], which are based on deep neural networks. We directly applied COLMAP and SURE to the WHU test set, which contained 93 images (5376×5376 in size) and output depth maps or dense clouds. We trained the CNN-based methods, which includes our method, with the WHU training set, which contained 3,600 sub-units (768×384 in size) and then evaluated them on the WHU test set, which contained 1,360 sub-units with the same image size. The input view numbers were N=3 and N=5 for WHU-3 and WHU-5, respectively, with depth sample number D=200. The depth range can vary in each image, so we evaluated the initial depth with COLMAP and set the depth range accordingly for each image. In the test set, the depth number was variable and we set the interval at 0.15 m. The performances of the different methods were compared on the depth maps without any post-processing. For SURE, the generated dense point clouds were translated to depth maps in advance.

In the training stages of RED-Net, RMSProp [20] was chosen as the optimizer, and the learning rate was set at 0.001 with a decay of 0.9 for every 5k iterations. The model was trained for three epochs with a batch size of one, which involved about 150k iterations in total. All the experiments were conducted on a 24 GB NVIDIA TITAN RTX graphics card and TensorFlow platform.

We used four measures to evaluate the depth quality: 1) **Mean absolute error (MAE)**: the average of the L1 distances between the estimated and true depths, and only the distances within 100 depth intervals were counted in order to exclude the extreme outliers; 2) $<$ **0.6m**: the percentage of pixels whose L1 error were less than the 0.6 m threshold; 3) **3-interval-error** ($<$ **3-interval**): the percentage of pixels whose L1 error was less than three depth intervals; 4) **Completeness**: the percentage of pixels with the estimated depth values in the depth map.

Method	Train & Test	MAE (m)	<3-interval (%)	<0.6m (%)	Comp.
COLMAP	/	0.1548	94.95	95.67	98%
SURE	/	0.2245	92.09	93.69	94%
MVSNet	WHU-3	0.1974	93.22	94.74	100%
	WHU-5	0.1543	95.36	95.82	100%
R-MVSNet	WHU-3	0.1882	94.00	94.90	100%
	WHU-5	0.1505	95.64	95.99	100%
RED-Net	WHU-3	0.1120	97.90	**98.10**	100%
	WHU-5	**0.1041**	**97.93**	98.08	100%

Table 1: The quantitative results on WHU dataset.

Our quantitative results are shown in Table 1. RED-Net outperformed all the other methods for all the indicators and obtained at least 50% MAE improvement compared to the second-best R-MVSNet. For the 3-interval-error and 0.6 m threshold indicators, our method exceeded all the other methods at least 2%. Our qualitative results in Figure 5 show that RED-Net's reconstructed depth map was the cleanest and most similar to the ground truth.

5.2. GPU Memory and Runtime

The GPU memory requirement and running speed of RED-Net, MVSNet, and R-MVSNet on the WHU dataset are listed in Table 2. The memory requirement of MVSNet increased with depth sample number D, whereas that of RED-Net and R-MVSNet were constant at D. The occupied memory of RED-Net was nearly half that of R-MVSNet, and RED-Net could reconstruct a depth map with full resolution, which was 16-time larger than the latter.

The runtime was related to the depth sample number, input image size, and image number. Given the same N-view images, (R-)MVSNet generated a depth map down-sampled by 4 and was slightly faster, while RED-Net kept the same resolution with input inference. Therefore, considering the output resolution, our network was much more efficient than the others.

Methods	Input size	Depth sample number (3-view)				(5-view)	Output size
		D = 800	D = 400	D = 200	D = 128	D = 200	
MVSNet	384 × 768	17085M 1.1s	8893M 0.6s	4797M 0.3s	2749M 0.2s	4797M 0.5s	96 × 192
R-MVSNet	384 × 768	4419M 1.2s	4419M 0.6s	4419M 0.4s	4419M 0.3s	4547M 0.6s	96 × 192
RED-Net	384 × 768	2493M 1.8s	2493M 0.95s	2493M 0.6s	2493M 0.5s	2509M 0.8s	384 × 768

Table 2: Comparisons of memory requirement and runtime between (R-)MVSNet and RED-Net. Our method requires less memory but achieves full-resolution reconstruction.

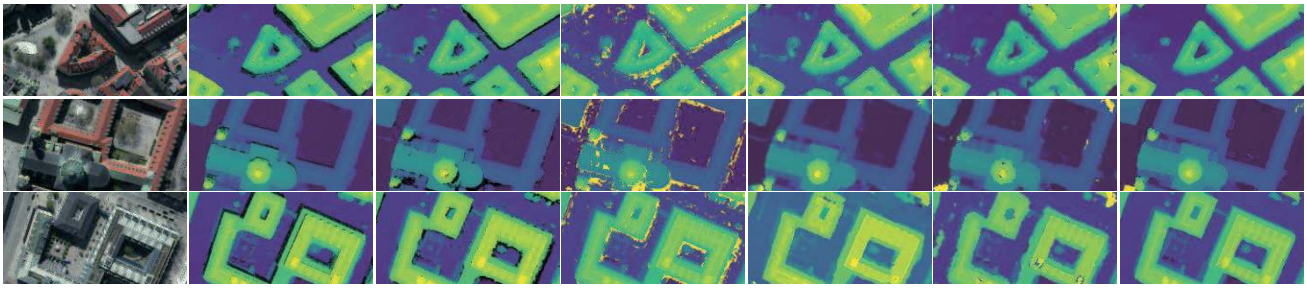


Figure 6: The inferred depth maps of three sub-units on München aerial image set. The deep learning based methods are trained on the WHU-3 training set.

5.3. Generalization

The WHU dataset was created under well-controlled imaging processes. To demonstrate the representation of the WHU dataset for aerial datasets and the generalization of RED-Net, five methods were tested on the real aerial dataset München [11]. The München dataset is somewhat different from the WHU dataset in that it was captured at a metropolis instead of a town. It is comprised of 15 aerial images (7072×7776 in size) and 80% and 60% overlapping in the heading and side directions, respectively. The three CNN-based models were pre-trained on the DTU or WHU datasets without any fine-tuning. The input view number of the München dataset was N=3 and the depth sample resolution was 0.1 m. The quantitative results are shown in Table 3. Some qualitative results are shown in Figure 6. Three conclusions can be drawn from Table 3. First, RED-Net, which was trained on the WHU-3 dataset, performed the best in all the indicators. RED-Net also exceeded the other methods by at least 6% in 3-interval-error. The model trained on the WHU-5 dataset performed almost the same as RED-Net. Second, the WHU dataset guaranteed the generalizability while the indoor DTU dataset could not. When trained on the DTU dataset, all the CNN-based methods performed worse than the two conventional methods. For example, (R-)MVSNet was 30% worse than the two conventional methods in 3-interval-error; however, when trained on the WHU dataset, their performances were comparable to the latter. Finally, the recurrent encoder-decoder structure in RED-Net led to better generalizability compared to the stack of GRUs in R-MVSNet and the 3D convolutions in MVSNet. When trained on the DTU dataset, our method experienced a 20% improvement over (R-)MVSNet in 3-interval-error.

Methods	Train set	MAE (m)	<3-interval (%)	<0.6m (%)
COLMAP	/	0.5860	73.36	81.95
SURE	/	0.5138	73.71	85.70
MVSNet	DTU	1.1696	43.19	61.26
	WHU-3	0.6169	69.33	81.36
	WHU-5	0.5882	70.43	83.46
R-MVSNet	DTU	0.7809	43.22	70.26
	WHU-3	0.6228	74.33	83.35
	WHU-5	0.6426	74.08	83.68
RED-Net	DTU	0.6867	63.04	78.89
	WHU-3	**0.5063**	**80.67**	**86.98**
	WHU-5	0.5283	80.40	86.69

Table 3: Quantitative evaluation on the München aerial image set with different MVS methods. The deep learning based methods were trained on the WHU or the DTU training set.

6. Discussion

6.1. Advantage of the Recurrent Encoder-Decoder

In this section, we evaluate the effectiveness of the recurrent encoder-decoder in an MVS network. We down-sampled the feature maps by four times in the 2D extraction stage. By doing this, the cost maps in RED-Net were the same size as R-MVSNet. The final output was also changed to 1/16 size of the input to keep consistent with the R-MVSNet. The results are compared in Table 4. On the three aerial datasets, RED-Net demonstrated obvious advantages for all measures, which indicates that the high performance of RED-Net is not only due to improvement of the output resolution, but also to the encoder-decoder structure, which learned spatial and contextual representations better than stacked GRUs.

Figure 7: The point cloud reconstructions of a large area using RED-Net. The right is an enlarged part from the left scene.

6.2. Evaluation on DTU

Although RED-Net is mainly developed for large-scale aerial MVS problem, it surpassed the state-of-the-art R-MVSNet on the close-range DTU dataset. Table 5 shows that, with the same post-processing (photometric and geometric filtering), the overall score of RED-Net outperformed that of R-MVSNet by 18%, and also outperformed the results provided in [37] with full four post-processing methods. Overall score is derived from two representative indicators *accuracy* and *completeness* suggested by the DTU dataset [1] and used in [37].

6.3. Large-scale Reconstruction

RED-Net produced full resolution depth maps with arbitrary depth sample numbers, which particularly can benefit high-resolution large-scale reconstruction of the Earth's surface from multi-view aerial images with a wide depth range. Moreover, RED-Net can handle three-view images with a size of 7040×7040 pixels on a 24GB GPU, taking only 58 seconds to infer a depth map with 128 depth sample numbers. When we inferred the depth of a scene covering 1.8×0.85 km^2 (Figure 7), RED-Net with 3-view input and 200 depth sample numbers took 9.3 minutes while SURE took 150 minutes and COLMAP took 608 minutes.

7. Conclusion

In this paper, we introduced and demonstrated a synthetic aerial dataset, called the WHU dataset, that we created for large-scale and high-resolution MVS reconstruction, which, to our knowledge, is the largest and only available multi-view aerial dataset. We confirmed in this paper that the WHU dataset will be a beneficial supplement to current close-range multi-view datasets and will help facilitate the study of large-scale reconstruction of the Earth's surface and cities.

We also introduced in this paper a new approach we developed for multi-view reconstruction called RED-Net.

Dataset	Methods	MAE (m)	<3-interval (%)	<0.6m (%)
München	R-MVSNet	0.4264	81.43	88.67
	RED-Net*	0.3677	83.63	89.95
WHU-3	R-MVSNet	0.1882	94.00	94.90
	RED-Net*	0.1574	95.52	96.03
WHU-5	R-MVSNet	0.1505	95.64	95.99
	RED-Net*	0.1379	95.89	96.64

Table 4: Results of the R-MVSNet and RED-Net with the same size of inferred depth map on three datasets. '*' means that the cost maps and outputs of our method are downsampled by four as the R-MVSNet. Models are trained and tested on the same dataset respectively.

Methods(D=256)	Mean Acc.	Mean Comp.	Overall(mm)
R-MVSNet [10]	0.385	0.459	0.422
R-MVSNet*	0.551	0.373	0.462
RED-Net	0.456	0.326	0.391

Table 5: Results of the R-MVSNet and RED-Net on DTU benchmark. '*' means our implementation with only photometric and geometric filtering post-processing, the same as in RED-Net.

This new network was shown to achieve highly efficient large-scale and full resolution reconstruction with relatively low memory requirements, and its performance exceeded that of both the deep learning-based methods and commercial software. Our experiments also showed that RED-Net pre-trained on our newly created WHU dataset could be directly applicable to a somewhat different aerial dataset due to the proper training data and model's powerful generalizability, which has sent a signal that deep learning based approaches may take place of conventional MVS methods in practical large-scale reconstruction.

Acknowledgement

This work was supported by the Huawei Company, Grant No. YBN2018095106.

References

[1] H. Aanaes, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016.

[2] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M Taha, and Vijayan K Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*, 2018.

[3] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference 2011*, pages 1–11, 2011.

[4] J. R. Chang and Y. S. Chen. Pyramid stereo matching network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 539–546, 2005.

[7] R. T. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.

[8] ContextCapture. Available: https://www.bentley.com/en/products/brands/contextcapture.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3354–3361, 2012.

[10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[11] Norbert Haala. The landscape of dense image matching algorithms. 2013.

[12] W. Hartmann, S. Galliani, M. Havlena, L. Van Gool, and K. Schindler. Learned multi-patch similarity. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1595–1603, 2017.

[13] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[14] P. H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. B. Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2821–2830, 2018.

[15] M. Q. Ji, J. R. Gall, H. T. Zheng, Y. B. Liu, and L. Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2326–2334, 2017.

[16] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems*, pages 365–376, 2017.

[17] 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)*, pages 66–75, 2017.

[18] A. Knapitsch, J. Park, Q. Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78, 2017.

[19] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[21] 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)*, pages 4040–4048, 2016.

[22] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[23] J. H. Pang, W. X. Sun, J. S. J. Ren, C. X. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *IEEE International Conference on Computer Vision Workshops*, pages 878–886, 2017.

[24] Pix4D. Available: https://www.pix4d.com/.

[25] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6620–6629, 2017.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, volume 9351, pages 234–241, 2015.

[27] Mathias Rothermel, Konrad Wenzel, Dieter Fritsch, and Norbert Haala. Sure: Photogrammetric surface reconstruction from imagery. In *Proceedings LC3D Workshop, Berlin*, page 2, 2012.

[28] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[29] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[30] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera

videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2538–2547, 2017.

[31] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 519–528. IEEE, 2006.

[32] SURE-Aerial. Available: http://www.nframes.com/products/sure-aerial/.

[33] E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.

[34] P. S. Wang, Y. Liu, Y. X. Guo, C. Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics*, 36(4):72, 2017.

[35] Rui Wang and Xuelei Qian. *OpenSceneGraph 3.0: Beginner's Guide*. Packt Publishing Ltd, 2010.

[36] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[37] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

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