The Journal of Supercomputing (2022) 78:16512–16528
https://doi.org/10.1007/s11227-022-04512-5

3D map reconstruction using a monocular camera for smart cities

Yuxi Hu[1] · Taimeng Fu[1] · Guanchong Niu[1] · Zixiao Liu[1] · Man-On Pun[1,2]

Accepted: 29 March 2022 / Published online: 7 May 2022

Abstract

Large-scale high-resolution three-dimensional (3D) maps play a vital role in the development of smart cities. In this work, a novel deep learning-based multi-view-stereo method is proposed for reconstructing the 3D maps in large-scale urban environments by exploiting a monocular camera. Compared with other existing works, the proposed method can perform 3D depth estimation more efficiently in terms of computational complexity and graphics processing unit memory usage. As a result, the proposed method can practically perform depth estimation for each pixel before generating 3D maps for even large-scale scenes. Extensive experiments on the well-known DTU dataset and real-life data collected on our campus confirm the good performance of the proposed method.

Keywords 3D reconstruction · Deep learning · Dense map · Smart cities

1 Introduction

Recent advances in the field of three-dimensional (3D) reconstruction have inspired many emerging applications for the development of smart cities [1–5]. However, substantial research efforts are required to implement 3D

✉ Man-On Pun
simonpun@cuhk.edu.cn

Yuxi Hu
yuxihu@link.cuhk.edu.cn

Taimeng Fu
taimengfu@link.cuhk.edu.cn

Guanchong Niu
guanchongniu@link.cuhk.edu.cn

Zixiao Liu
zixiaoliu@link.cuhk.edu.cn

[1] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China

[2] Shenzhen Research Institute of Big Data, Shenzhen 518172, China

reconstruc- tion in practice [6–9]. One of the most critical challenges is to reconstruct 3D models for outdoor large-scale scenes that incur formidable computational complexity [6, 10]. To cope with this problem, one conventional approach is to use map representations. Some of the most representative methods include the point cloud-assisted grid-based methods and volumetric methods. Despite their many advantages, these conventional methods suffer from many drawbacks. For instance, the point cloud-assisted methods require a large number of sample points to produce high-quality reconstruction, which may not be practical for applications with a smaller point cloud [11]. Furthermore, the sparse representation without meshing cannot satisfy the requirements for real-life applications such as path planning. In contrast, the volume representation can be directly applied to the path planning task by using an implicit function to represent 3D geometry as a regularly sampled 3D grid. However, the genera- tion of the volume representation often incurs heavy memory usage [12], which makes these methods computationally infeasible.

In 3D reconstruction, depth, monocular and stereo cameras are com- monly employed. In particular, the depth cameras are capable of providing high-quality depth measurements for indoor environments. However, their per- formance for outdoor environments is rather disappointing. Unfortunately, 3D map reconstruction for real-life cities is often conducted in complex outdoor environments, which presents major challenges to most existing 3D mapping algorithms. To overcome this problem, machine learning-based multi-view- stereo (MVS) methods have been proposed to construct 3D cost volumes by exploiting 3D convolutional neural networks (CNN) to regularize and gener- ate the depth information [13, 14]. Since 3D CNN is usually both time and memory-consuming, [15, 16] proposed to apply down-sampling during feature extraction to reduce the computational complexity. However, the resulting depth map is of low resolution, which fails to take full advantage of the origi- nal high-resolution pictures [17]. To alleviate the challenges incurred by heavy memory usage and time consumption, R-MVSNet proposed to first satisfy memory requirements and then cover full depth ranges in sequence at the cost of additional runtime penalty [18]. More specifically, the additional runtime is spent on cascading 3D features to predict the coarse and fine depth maps [19]. Motivated by the discussions aforementioned, this work proposes a novel MVS network structure to reduce memory consumption and high-resolution runtime by exploiting a cascade-based learning formulation. In particular, an iterative algorithm is developed to approximate the nearest neighbor calcula- tion using the inherent spatial coherence of depth maps. The advantages of the proposed deep learning-based MVS method include low memory requirements, parallax range-independence and implicit smoothing effects. Our contributions are summarized as follows:

- An end-to-end trainable MVS framework based on deep learning is pro- posed by exploiting a coarse-to-fine approach to expedite the computation. The proposed learning-based method can greatly improve the model inter- pretability as compared to conventional methods;
- A feature pyramid network is adopted to enhance image feature extraction;
- Extensive experiments on the well-known DTU dataset and real-life data

collected on our campus are used to confirm the good performance of the proposed method.

2 Related work

In this section, some related works reported in the fields of stereo match- ing, depth estimation, and 3D reconstruction are first reviewed in Sect. 2.1 before conventional multi-view stereo(MVS) methods are discussed in Sect. 2.2. Furthermore, learning-based MVS methods are further elaborated and ana- lyzed in Sect. 2.3. Finally, the related works on high-resolution images and 3D reconstruction for smart cities are highlighted in Sects. 2.4 and 2.5, respectively.

2.1 Three-dimensional (3D) reconstruction

Epipolar Geometry is a constraint on stereo vision modeling, describing the internal projective relationship between two views. By exploiting the epipolar geometry, the optimal solutions to the stereo matching or depth estimation problem are established. More specifically, the epipolar geometry is charac- terized by the camera internal parameters and the relative pose between two views, independent on the external scene. Thus, if the image point on the imag- ing plane of a real-world object under study is known, the image point of the object on another imaging plane that constitutes the epipolar plane must be on the epipolar line. As a result, it becomes more computationally efficient during the feature point matching process by exploiting the fact that all feature points should fall within certain straight lines, in lieu of the whole image.

Furthermore, the homography transformation is a projection method widely used in image transformation, projecting a point in one space into another. Given four point pairs in two images, the homography transformation relationship between the two images can be found by solving a system of linear equations. Built upon the homography transformation, the planning sweeping method models calibrated images captured by multiple cameras of objects in a scene by dividing the objects into a series of dense planes with equal spacing. If the parallel planes are sufficiently finely divided, any point on the surface of the object must be on a certain plane. Assuming that all objects have only diffuse reflection without taking in account the lighting transformation in the scene, the image pixel values recorded by all cameras at this point must be equal while the pixel values projected onto each camera are different for points on the surface of the object. By exploiting these observations, the plane scanning algorithm assumes that a point on the plane is likely to be a point on the surface of the object if its pixel value projected to each camera is identical.

Finally, stereo matching, also known as disparity estimation or binocular depth estimation, estimates the depth value corresponding to each pixel of the reference image through a pair of left and right views captured at the same moment and corrected by epipolar lines. The correlation between the pixel to be matched and the candidate pixel is calculated by the matching cost function. To reduce the required computational complexity, the algorithm limits the disparity search range to a certain interval. For each pixel in the reference image, a three-dimensional matrix of dimension $W \times H \times D$ known as Disparity Space Image (DSI) is used to store the matching cost of each pixel within the disparity range. The disparity map obtained through disparity calculation can be further optimized by methods such as removing error errors, proper smoothing, and sub-pixel accuracy optimization. However, since DSI is computed based on local pixel information within a certain window size, it is susceptible to noise and illumination conditions, particularly when the image is in a repetitive or weak texture area. Thus, it is critical to optimize DSI by jointly exploiting local and global information.

2.2 Traditional multi-view stereo

Generally speaking, existing MVS methods can be classified into four categories, namely the point cloud-based methods, the voxel-based methods, the mesh-based methods and the depth map-based methods [20]. Point cloud- based methods usually start with a sparse set of matching feature points before expanding the initial matching to surrounding pixels and eliminating incorrect matches using visibility constraints [21]. This process is repeated for multiple times until dense reconstruction is achieved in low-texture regions. However, the performance of these point cloud-based methods is not satisfactory due to their demanding requirements on the quality of extracted feature points, par- ticularly when the texture distribution is uneven. In contrast, the voxel-based methods first compute the bounding box containing the scene before discretiz- ing the 3D space into an irregular grid to find voxels near the scene surface [22]. However, the memory consumption of the voxel-based methods increases with the voxel resolution that determines the reconstruction accuracy. Fur- thermore, the mesh-based methods initialize the surface evolution following iterative improvement on the multi-view photometric consistency [23]. Finally, the depth map-based methods such as Gipuma [24] and Colmap [25] first decompose the entire reconstruction into several single-view depths before fusing all the depth maps into the point cloud under one coordinate system. More specifically, PatchMatch is commonly employed to iteratively optimize the esti- mation of the depth map of each image. In addition, since the depth map-based methods utilize only one reference image and a small number of source images to estimate a single depth map, these methods are very computationally effi- cient. However, PatchMatch cannot take into account the efficiency of sampling propagation and the accuracy of view selection at the same time, which makes PatchMatch ineffective in handling weak-textured areas. To resolve this prob- lem, [25] proposed a distributed method based on distributed motion average and global camera consensus. Furthermore, Gipuma proposed in [24] capital- izes on a red-black checkerboard pattern to streamline the message-passing process by exploiting a massive number of parallel multi-view stereos. In con- trast, COLMAP proposed in [25] jointly estimates pixel-wise view selection, depth map and surface normal. In practice, Gipuma has higher computational efficiency as compared to the COLMAP algorithm. However, the COLMAP algorithm adopts a more complex view selection strategy based on the Markov chain model, while the Gipuma algorithm lacks view selection. As a result, the performance of Gipuma is in general worse than that of the COLMAP algorithm.

Furthermore, the saliency of the texture information of an image varies at different scales. In order to better extract saliency information for ambigu- ous regions, researchers proposed a multi-scale geometric consistency guided framework to perceive salient information at different scales and convey through geometric consistency across multiple views. Finally, ACMM was pro- posed in [26] by exploiting adaptive checkerboard sampling, multi-hypothesis joint view selection and multi-scale geometric consistency guidance.

2.3 Multi-view stereo (MVS)

As 3D reconstruction requires key geometric information extracted from mas- sive images and data, some pioneering methods based on data mining and machine learning have been proposed to extract image features, or to con- struct cost volumes. In particular, learning-based MVS methods have drawn considerable research interest due to their excellent performance. For instance, SurfaceNet [27] and DeepMVS [28] pre-warped multi-view images into 3D space before applying deep networks in regularization and aggregation. Fur- thermore, 3D cost-based MVS methods were proposed by exploiting warped 2D image features from multiple views as well as 3D CNN-based cost reg- ularization and depth regression [29]. Since 3D CNN is graphics processing unit (GPU) memory-intensive, these 3D cost-based MVS methods usually use downsampled cost volumes. It should be emphasized that our proposed cas- cading cost volume can be easily integrated into these methods to achieve high-resolution cost volume and further improve their performance in terms of accuracy, computation speed and GPU memory efficiency.

2.4 High-resolution output in MVS

To cope with the memory-intensive problem incurred in high-resolution MVS, some innovative attempts have been made. For instance, Point MVSNet [30] reduces the required memory usage without voxel grids by generating coarse depth maps using a small amount of cost while employing a point-based iter- ative refinement network to output full-resolution depth. Furthermore, a 3D cost volume is built based on warped 2D image features from multiple views and 3D CNNs are applied for cost regularization and depth regression.

To reduce the intensive GPU memory usage required by the 3D CNNs, a common approach reported in the literature is to downsample the cost volume. As a result, cascade MVSNet can use less running time and GPU memory than Point MVSNet to output full-resolution depth with higher accuracy. The cascade cost volume can be easily integrated into these learning-based methods to enable high-resolution cost volumes.

2.5 3D reconstruction for smart cities

Highly accurate 3D city models are indispensable for the development of smart cities. As it is both costly and time-consuming to develop 3D city models, it is particularly important to explore and adopt new technologies to pro- vide low-cost and computationally efficient 3D model reconstruction. Driven by the rapid advances in virtual reality and augmented reality, terrain 3D model data is widely employed in surveying, mapping and other areas. For instance, smart mines capitalize on terrain 3D models to provide panoramic visual management for mine managers while combining multiple tasks including automatic transportation, emergency rescue, mineral analysis, fire and explosion protection. Furthermore, in smart tourism, natural or cultural land- scape is reconstructed through a 3D digital model before being integrated into the tourism industry's attraction recommendation, journey routes, cultural exchanges, business consumption and so on. Finally, the game, film and tele- vision production industries use terrain 3D model data to provide rich and changeable scenes for large-scale interactive games and digital movies.

3 Methodology

In this section, we introduce the structure of the proposed network as illus- trated in Fig. 1. It consists of multi-scale feature extraction, learning-based coarse-to-fine depth estimation, and a spatial refinement module.

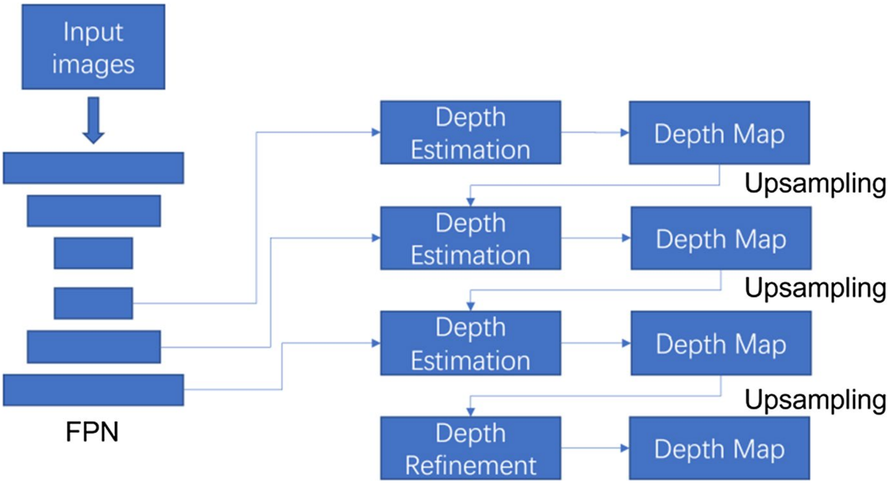


Fig. 1 Framework of the proposed network consisting of three components, namely multiscale feature extractors, learning-based depth estimation and depth refinement

3.1 Feature pyramid network

In order to obtain high-resolution depth (or disparity) maps, previous works often generate a comparatively low-resolution depth (or disparity) map using the standard cost volume followed by upsampling and refining with 2D CNNs. The standard cost volume is constructed using the top-level feature maps of high-level semantic features without low-level finer representations. Here, we adopt the Feature Pyramid Network (FPN) proposed in [31] to generate feature maps with improved spatial resolutions and establish cost volumes of higher resolution. For instance, when applying cascade cost volume to MVSNet [32], we build three scale cost volumes from the FPN feature maps with spatial reso- lutions of *{*1/16, 1/4, 1*}* of the input image size, respectively. More specifically, given N input images of size $W \quad H$, we denote by $\mathbf{I}_0$ and $\left\{\mathbf{I}_i\right\}_{i=1}^{N-1}$ reference and source images, respectively. We extract pixel features from input hierar- chically at multiple resolutions, which allows us to progressively improve the depth map estimation in a coarse-to-fine manner, following a similar approach as proposed in [33].

3.2 Learning-based depth estimation

3.2.1 Hypothesis plane

We denote by R_k and I_k the depth range and the plane interval of the k-th stage for $k = 1, 2,..$, respectively. Assuming that R_1 covers the entire depth range of the input scene, the scope of the hypothesis in the current stage can be predicted based on the output of the previous stage as below:

$$R_{k+1} = R_k \cdot w_k, \tag{1}$$

where $w_k < 1$ is the reducing factor of the hypothetical range.

Furthermore, the initial hypothesis plane interval is comparatively larger to generate a coarse depth estimation as compared to the conventional single cost volume formulation. For latter stages, finer hypothesis plane intervals are applied to recover more detailed outputs. Thus, we have:

$$I_{k+1} = I_k \cdot p_k, \tag{2}$$

where $p_k < 1$ is the reducing factor of hypothesis plane interval.

3.2.2 Depth estimation

The proposed depth estimation module includes the following three main steps:

1. Initialization: Generate random hypotheses;
2. Propagation: Propagate the hypothesis to neighbors;
3. Evaluation: Calculate the matching cost of all hypotheses.

First, a random initialization step is performed to promote diversification. After that, the proposed depth estimation alternates between the propaga- tion and evaluation steps until convergence is achieved [34]. The assumption of spatial consistency of depth values is generally valid only for pixels from the same physical surface [35]. Thus, propagation is performed in an adaptive manner, collecting hypotheses from the same surface [36]. Finally, the eval- uation module completes differentiable warping, matching cost computation, adaptive spatial cost aggregation and depth regression. The detailed structure of depth estimation module is illustrated in Fig. 2.

Most learning-based MVS methods adopt the plane sweep stereo approach to establish front-to-parallel planes at sampled depth hypotheses and warp the feature maps of source images into the planes. We denote by $\{\mathbf{Q}_i\}_{i=0}^{k}$ and $\left[\mathbf{R}_{0,i}\mathbf{t}_{0,i}\right]_{i=1}^{K}$ the intrinsic matrices of image i and the relative transfor- mations of reference view 0 and source view i, respectively. Thus, the pixel $\mathbf{p}_{i,j}: = \mathbf{p}_i\ (d_j)$ in the source view for a pixel $\mathbf{p}$ in the reference view in homogeneous coordinates is given by:

$$\mathbf{p}_{i,j} = \mathbf{Q}_i \cdot \mathbf{R}_{0,i} \cdot \mathbf{Q}_0^{-1} \cdot \mathbf{p} \cdot d_j + \mathbf{t}_{0,i},$$

where $d_j := d_j(\mathbf{p})$ is the depth hypothesis.

Finally, denoted by $\mathbf{F}_i\ (\mathbf{p}_{i,j})$, the warped source feature maps of view i and the j-th set of depth hypotheses are obtained through differentiable bilinear interpolation. It is worth noting that this step has to integrate information from an arbitrary number of source views into a single cost per pixel $\mathbf{p}$ and depth hypothesis d_j for MVS methods. To this end, we propose to compute the cost per hypothesis using group-wise

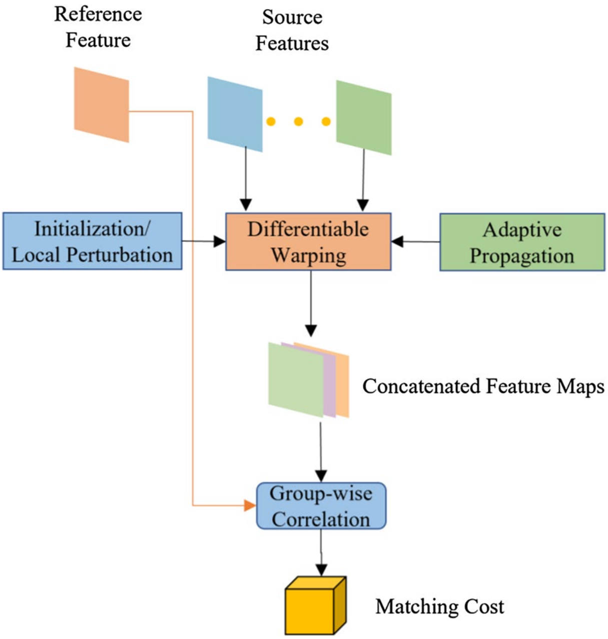


Fig. 2 The depth estimation module of the system

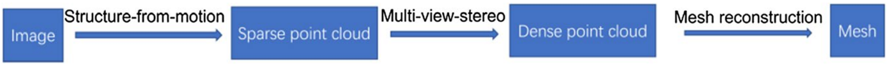


Fig. 3 Block diagram of a typical 3D reconstruction system

correlation before aggregating the views with a pixel-wise view weighting coefficient. By doing that, we can employ visibility information during cost aggregation to achieve better robust performance. Finally, the per-group costs are projected into a single number, per reference pixel and hypothesis, through a small network.

3.3 Loss function

The proposed loss function denoted by L_{total} takes into account the losses among all the depth estimation and rendered ground truth with same resolution as follow:

$$L_{\text{total}} = \sum_{k=1}^{3} \sum_{i=1}^{n_k} L_i^k + L_{ref}^0, \tag{3}$$

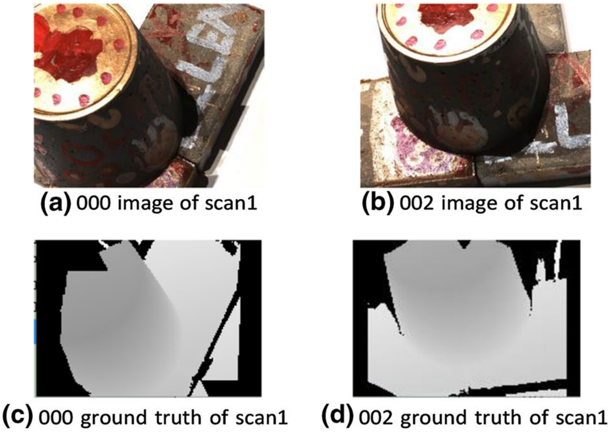
L_i^k is the loss of the i-th iteration of estimation on the k-th stage whereas L_{ref}^0 stands for the loss for the final refined depth map. Furthermore, n_k is the number of views on the k-th stage. We propose to adopt the smooth $l1$ loss for L_i^k.

3.4 MVS in 3D reconstruction

Figure 3 shows the 3D reconstruction step-by-step procedures. First, sparse point cloud, the camera's intrinsic and extrinsic parameters are extracted from the input image through a structure-from-motion process such as COLMAP. However, since the point cloud is sparse, it is not sufficient to support accurate 3D reconstruction. Therefore, multi-view stereo is applied to the sparse point cloud to generate dense point cloud before mesh reconstruction is utilized to generate high-quality 3D reconstruction. Thus, it is apparent that the multi- view stereo step plays a critical role in the 3D reconstruction. In particular, the proposed method is able to create denser point cloud, which enables a simpler mesh reconstruction method without requiring tedious refinement and optimization.

4 Experiment

In this section, we evaluate our proposed network using the well-known DTU dataset as well as some building photos taken on the campus of the Chinese University of Hong Kong, Shenzhen (CUHKSZ). In the following, Sect. 4.1 first elaborates the DTU dataset and the CUHKSZ data collection process. After that, the software and hardware environment for conducting the experiments are detailed in Sect. 4.2 while the comparison and analysis of the reconstruction results are provided in Sect. 4.3.


(a) 000 image of scan1

(b) 002 image of scan1

(c) 000 ground truth of scan1

(d) 002 ground truth of scan1

Fig. 4 Sample experimental data from DTU

4.1 Dataset

The dataset DTU is a large MVS dataset consisting of 124 different scenes scanned in seven different lighting conditions at 49 or 64 positions [11]. In our experiments, we use the DTU training set to train our proposed network while testing the performance with the DTU evaluation set. For evaluation purposes, the dataset also contains the ground-truth point cloud for each scene. A sample of the dataset is shown in Fig. 4.

To verify the generalization capability of our proposed network, the trained network was tested on real-life data collected on the CUHKSZ campus. First, a drone was used to shoot some scenes on the campus. After extracting some frames from the video at certain intervals, we obtained a series of images without time series. Together with the pose information and similar frames derived, these images were used as the input to COLMAP.

More specifically, the dataset contains three types of information. First, the dataset provides photos of objects to be reconstructed. Second, the corresponding camera intrinsic and extrinsic parameters used to take those photos are also available in the dataset. Finally, the dataset contains matching pairs of neigh- bor frames. The DTU dataset provides these three types of information for training and testing whereas we used COLMAP to extract the corresponding information when reconstructing real scenes.

4.2 Implementation

Our simulation platform was implemented with PyTorch, running on Nvidia 3090 GPU. The detailed environment configuration is shown in Table 1.

We set the image resolution to 640 512 and the number of input images to $N = 5$. We set the iteration number of depth estimation module on stages 1, 2, 3 as 1, 2, 2, respectively. We trained our model with Adam for 16 epochs with a learning rate of

Table 1 Environment configuration

Type	Model or version
Operating system	Ubuntu 18.04
CPU	Intel Core i9-10900X
GPU	GeForce RTX 3090
Python	3.7.0
CUDA	11.0
PyTorch	1.7.1

Table 2 Quantitative comparison using scan #1 of DTU[11]

Methods	Point of Scan #1
MVSNet	3,736,559
Cascade MVSNet	29,275,994
Proposed	57,989,243

0.001 and a batch size of four. After depth estimation, we reconstructed point clouds following an approach similar to Cascade-MVSNet [37].

4.3 Comparison and analysis of results

In the first experiment, we input images at their original size, i.e., 1600 1200, and set the number of views $N = 5$. We follow the evaluation metrics provided by the DTU dataset in our performance evaluation as shown in Table 2.

Inspection of Table 2 suggests that the proposed network outperforms the other two existing MVSNet methods in terms of completeness. As a high completeness implies a denser point cloud with finer details, the proposed network can achieve competitive performance in overall quality. The improved performance can be also observed in Fig. 5 where the boundaries and thin structures in the models reconstructed by our proposed network appear more visible than of that by Cascade MVSNet.

More reconstruction examples using the DTU dataset are shown in Figs. 6 and 7. All of these point clouds show good details and little white noise.

To quantitatively compare the performance attained by different methods, three performance indices, namely *Accuracy (Acc.)*, Completeness *(Comp.)* and *Overall score (Overall)*, are employed. Specifically, *Acc.* computes the aver age distance between the reconstructed point cloud and the real point cloud, which represents the quality of the point cloud reconstructed by MVS. In addition, *Comp.* calculates the difference between the real point cloud and the reconstructed point cloud, which represents the completeness of point clouds. *Overall* averages the first two metrics. For all three metrics, a smaller score stands for better performance.

Table 3 provides the quantitative results derived from each method. Inspection of Table 3 suggests that the proposed method outperformed the MVSNet and

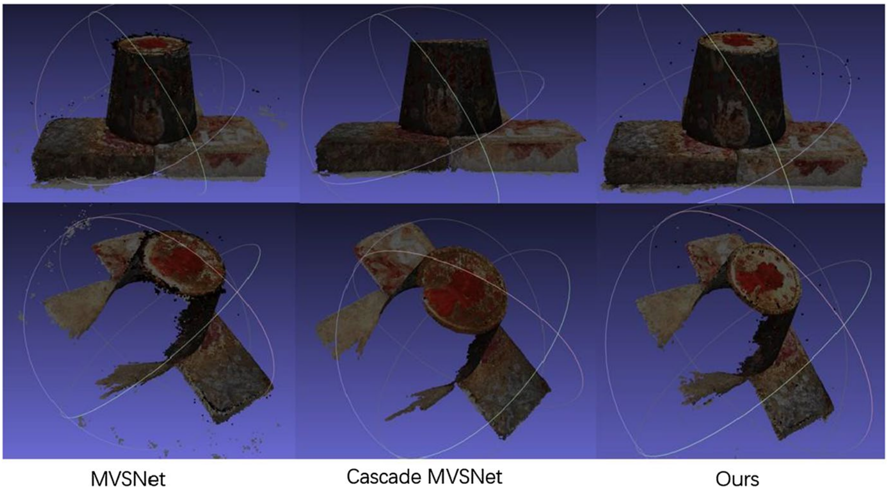


Fig. 5 Qualitative comparison using scan #1 of DTU[11]

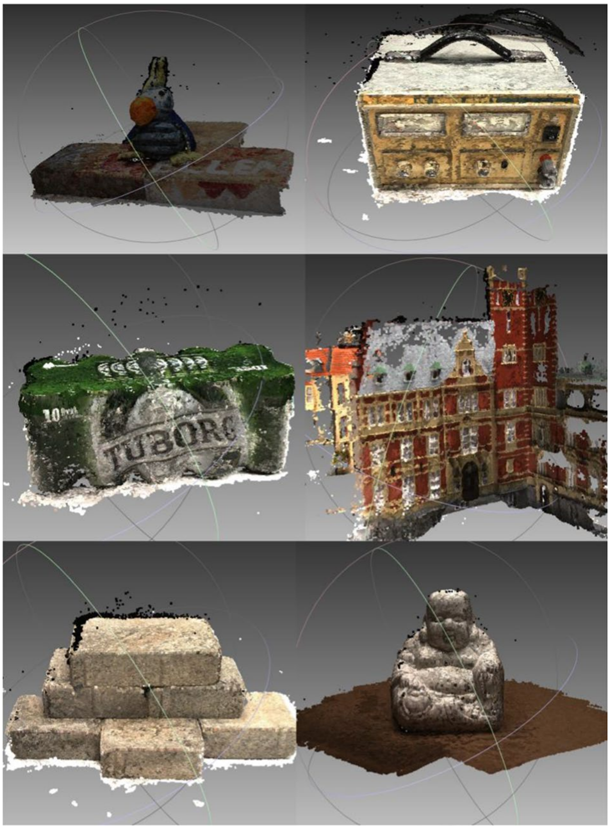


Fig. 6 Point cloud results on DTU [11]

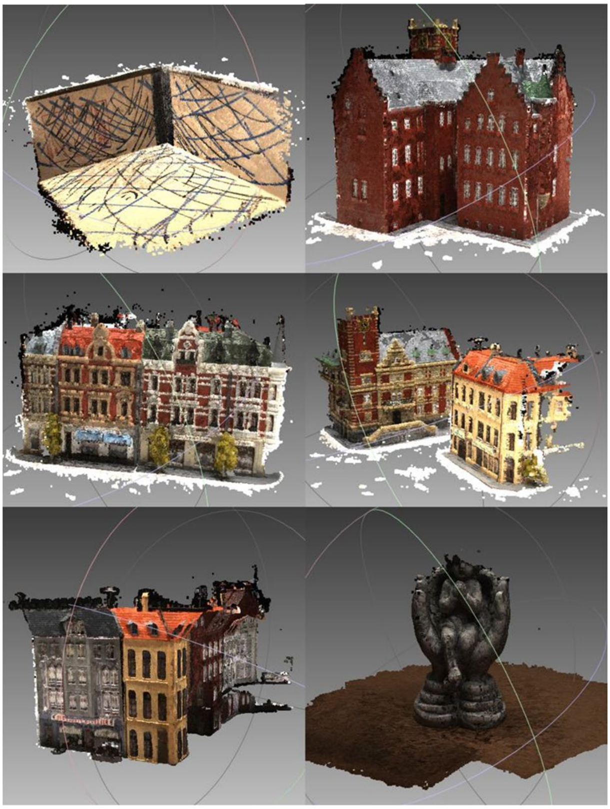


Fig. 7 Point cloud results on DTU [11]

CasMVSNet in all indices, which implies that the proposed method can reconstruct more accurate points without artifacts.

Next, we proceed to compare the memory consumption and run-time of the proposed network and several state-of-the-art learning-based methods. These existing methods generally utilize a cascade formulation of 3D cost volumes and output depth maps at the same resolution as the input images.

Table 4 shows our experiment results on memory consumption and run- time comparison among different methods. First, we observe that memory consumption and run-time increased much slower for the proposed network.

For instance, memory consumption for the resolution of 1152 864 is reduced by 45.7% as compared to Cascade MVSNet. It is worth pointing out that these existing methods cannot be fit into the memory of our GPU when higher-resolution images are used for evaluation. Inspection of Table 4 suggests that the proposed network is much more computationally efficient in terms of memory consumption and run-time while providing very competitive performance.

Finally, we collected some large-scale pictures on the CUHKSZ campus to test the generalization capability of the proposed network as shown in Fig. 8. It

Table 3 Quantitative results on DTU dataset

Method	Acc	Comp	Overall
Gipuma	**0.283**	0.873	0.578
SurfaceNet	0.450	1.040	0.745
MVSNet	0.456	0.646	0.551
CasMVSNet	0.365	0.385	0.375
Ours	0.350	**0.374**	**0.362**

Table 4 Quantitative results of different methods on DTU dataset (lower is better)

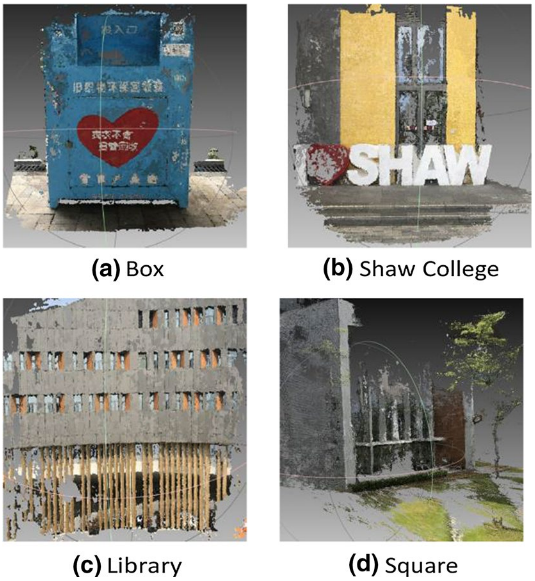
Methods	GPU	Mem(MB)	Run-time(s)
R-MVSNet	7577		1.28
Point-MVSNet	8731		3.35
MVSNet	10,823		1.210
Cascade MVSNet	5345		0.492
Ours	2900		0.510
Comp. with Cascade MVSNet	45.7%		−3.5%

is interesting to observe that the proposed method demonstrated good fea- ture extraction performance over low-textured regions (such as glass) and areas with repeating texture (such as similar building structures).

5 Conclusion

In this work, a novel cascade formulation of learning-based MVSNet has been proposed by augmenting learned adaptive propagation and evaluation modules based on deep features. Unlike conventional learning based methods, the pro- posed network does not rely on 3D cost volume regularization, which makes the proposed network more computationally efficient in terms of memory consump- tion and runtime. Extensive experimental results on the DTU dataset and our own data collected on the CUHKSZ campus have confirmed the excellent performance of the proposed network. The proposed network is particularly attractive for devices with limited computing resources and time-critical appli- cations. There are several extensions of this study that can be further explored. One possible drawback of the proposed network is that some noise is gener- ated during the reconstruction of very large scenes. To cope with this problem, the attention mechanism can be adopted to constrain multiple views. In addi- tion, novel feature extraction and regression methods should be devised for further performance improvement. Finally, the computational complexity of reconstruction increases with the number of points in the point cloud as it takes a longer time to synthesize these points. Thus, it is highly desirable to devise a computationally efficient method to handle dense point cloud.

Fig. 8 Reconstruction effect of real large scale data



(a) Box **(b)** Shaw College

(c) Library **(d)** Square

Acknowledgements This work was supported by the Shenzhen Science and Technology Innovation Committee under Grant No. JCYJ20190813170803617.

References

1. Huang G, Ma Y, Liu X, Luo Y, Lu X, Blake MB (2014) Model-based automated navigation and composition of complex service mashups. IEEE Trans Serv Comput 8(3):494–506
2. Huang G, Liu X, Ma Y, Lu X, Zhang Y, Xiong Y (2016) Programming situational mobile web applications with cloud-mobile convergence: an internetware-oriented approach. IEEE Trans Serv Comput 12(1):6–19
3. Chen X, Lin J, Ma Y, Lin B, Wang H, Huang G (2019) Self-adaptive resource allocation for cloud-based software services based on progressive qos prediction model. Science China Inf Sci 62(11):1–3
4. Chen X, Wang H, Ma Y, Zheng X, Guo L (2020) Self-adaptive resource allocation for cloud-based software services based on iterative qos pre- diction model. Futur Gener Comput Syst 105:287–296
5. Huang G, Xu M, Lin FX, Liu Y, Ma Y, Pushp S, Liu X (2017) Shuffle- dog: Characterizing and adapting user-perceived latency of android apps. IEEE Trans Mob Comput 16(10):2913–2926
6. Chen X, Li A, Guo W, Huang G et al (2015) Runtime model based approach to iot application development. Front Comp Sci 9(4):540–553

7. Chen C-M, Chen L, Gan W, Qiu L, Ding W (2021) Discovering high utility-occupancy patterns from uncertain data. Inf Sci 546:1208–1229
8. Chen C-M, Huang Y, Wang K-H, Kumari S, Wu M-E (2020) A secure authenticated and key exchange scheme for fog computing. Enterpr Inf Syst, 1–16
9. Liu X, Huang G, Zhao Q, Mei H, Blake MB (2014) imashup: a mashup- based framework for service composition. Science China Inf Sci 57(1):1–20
10. Lin B, Huang Y, Zhang J, Hu J, Chen X, Li J (2019) Cost-driven off- loading for dnn-based applications over cloud, edge, and end devices. IEEE Trans Industr Inf 16(8):5456–5466
11. Aanæs H, Jensen RR, Vogiatzis G, Tola E, Dahl AB (2016) Large- scale data for multiple-view stereopsis. Int J Comput Vision 120(2):153–168
12. Wang P-S, Liu Y, Guo Y-X, Sun C-Y, Tong X (2017) O-cnn: Octree- based convolutional neural networks for 3d shape analysis. ACM Trans- actions On Graphics (TOG) 36(4):1–11
13. Pang, J., Sun, W., Ren, J.S., Yang, C., Yan, Q.: Cascade residual learn- ing: A two-stage convolutional neural network for stereo matching. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 887–895 (2017)
14. Wu Z, Wu X, Zhang X, Wang S, Ju L (2019) Semantic stereo match- ing with pyramid cost volumes. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7484–7493
15. Liang Z, Feng Y, Guo Y, Liu H, Chen W, Qiao L, Zhou L, Zhang J (2018) Learning for disparity estimation through feature constancy. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2811–2820
16. Kar A, H¨ane C, Malik J (2017) Learning a multi-view stereo machine. arXiv preprint arXiv:1708.05375
17. Tola E, Strecha C, Fua P (2012) Efficient large-scale multi-view stereo for ultra high-resolution image sets. Mach Vis Appl 23(5):903–920
18. Yao Y, Luo Z, Li S, Shen T, Fang T, Quan L (2019) Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5525–5534
19. Riegler G, Osman Ulusoy A, Geiger A (2017) Octnet: Learning deep 3d rep- resentations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3577–3586
20. Lhuillier M, Quan L (2005) A quasi-dense approach to surface reconstruc- tion from uncalibrated images. IEEE Trans Pattern Anal Mach Intell 27(3):418–433
21. Furukawa Y, Ponce J (2009) Accurate, dense, and robust multiview stereopsis. IEEE Trans Pattern Anal Mach Intell 32(8):1362–1376
22. Sinha SN, Mordohai P, Pollefeys M (2007) Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: 2007 IEEE 11th international conference on computer vision, pp 1–8. IEEE
23. Furukawa Y, Ponce J (2006) Carved visual hulls for image-based modeling. In: European conference on computer vision, pp 564–577 Springer
24. Galliani, S., Lasinger, K., Schindler, K.: Gipuma: Massively parallel multi- view stereo reconstruction
25. Schonberger JL, Frahm, J-M (2016) Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4104–4113
26. Huang Y, Wang L (2019) Acmm: Aligned cross-modal memory for few- shot image and sentence matching. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5774–5783
27. Ji M, Gall J, Zheng H, Liu Y, Fang L (2017) Surfacenet: an end-to-end 3d neural network for multiview stereopsis. In: Proceedings of the IEEE international conference on computer vision, pp 2307–2315
28. Huang P-H, Matzen K, Kopf J, Ahuja N, Huang J-B (2018) Deepmvs: Learning multi-view stereopsis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2821–2830
29. 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. In: Proceedings of the IEEE international conference on computer vision, pp 66–75
30. Chen R, Han S, Xu J, Su H (2019) Point-based multi-view stereo network. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1538–1547

31. Lin T-Y, Doll´ar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117– 2125
32. Yao Y, Luo Z, Li S., Fang, T., Quan L (2018) Mvsnet: Depth infer- ence for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV), pp 767–783
33. Song X, Zhao X, Hu H, Fang L (2018) Edgestereo: a context integrated residual pyramid network for stereo matching. In: Asian Conference on Computer Vision, pp 20–35. Springer
34. Duggal S, Wang S, Ma W-C, Hu R, Urtasun R (2019) Deeppruner: learning efficient stereo matching via differentiable patchmatch. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 4384–4393
35. Sun J, Zheng N-N, Shum H-Y (2003) Stereo matching using belief propa- gation. IEEE Trans Pattern Anal Mach Intell 25(7):787–800
36. Sch¨onberger JL, Zheng E, Frahm J-M, Pollefeys M (2016) Pixelwise view selection for unstructured multi-view stereo. In: European conference on computer vision, pp 501–518. Springer
37. Gu X, Fan Z, Zhu S, Dai Z, Tan F, Tan P (2020) Cascade cost volume for high-resolution multiview stereo and stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2495–2504

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.