CGI2019 manuscript No.
(will be inserted by the editor)

Joint Bilateral Propagation Upsampling for Unstructured Multi-View Stereo

Mengqiang Wei · Qingan Yan · Fei Luo · Chengfang Song · Chunxia Xiao*

Abstract In this paper, we explore a new way to accelerate and densify unstructured Multi-View Stereo (MVS). While many unstructured MVS algorithms have been proposed, we discover that the image-guided resizing can easily and significantly benefit their 3D reconstruction results in both efficiency and completeness. Therefore, we build our framework upon a novel selective joint bilateral upsampling and depth propagation strategy. First we downsample the input unstructured images into lower resolution ones and perform the MVS calculation to efficiently obtain depth and normal maps from these resized pictures. Then the proposed algorithm upsamples the normal maps with the guidance of input images, and jointly take them into consideration to recover the low resolution depth maps into high resolution with geometry details simultaneously enriched. Finally by adaptively fusing the reconstructed depth and normal maps, we construct the final dense 3D scene. Quantitative results validate the efficiency and effectiveness of the proposed method.

Keywords Multi-View Stereo · unstructured images · 3D reconstruction · joint bilateral propagation upsampling

1 Introduction

In the past few decades, Structure-from-Motion [1,16, 22,23,28] for sparse modeling and Multi-View Stereo (MVS) [8,10,11,24,39] for dense modeling have made remarkable progress in image-based modeling field, regardless of accuracy, completeness and robustness.

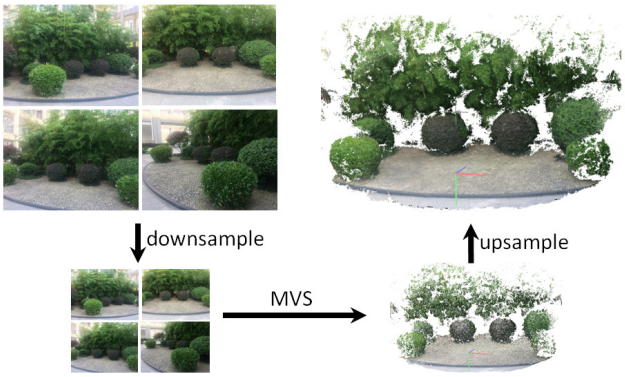


Fig. 1: The proposed selective joint bilateral propagation strategy can easily and significantly benefit 3D reconstruction results in both efficiency and completeness.

Image-based modeling offers a low-cost alternative to laser-based scans [29], and provides easily accessible reconstruction results benefiting a wide scope of applications, such as classification [27], image-based rendering [5], autonomous vehicles [12], point cloud geometry completion [33,36], city-scale modeling and so on. Different from monocular and binocular stereo, MVS is able to provide more visually consistent geometries and alleviated the occlusion problem [11]. While current MVS methods have shown good results for many reality scenes, they still suffer from high computation and memory requirements when the number and scale of images become large. Moreover, MVS relies on finding feature correspondences with epipolar constraints via the patch-based stereo strategies. Thus there are defi-

Mengqiang Wei · Fei Luo · Chengfang Song · Chunxia Xiao
School of Computer Science, Wuhan University, Wuhan, China, 430072. Email: 2292507220@qq.com, luofei@whu.edu.cn, Songchf@whu.edu.cn, cxxiao@whu.edu.cn. *Corresponding to Chunxia Xiao.

Qingan Yan
JD.com American Technologies Corporation, CA, 94043. Email: qingan.yan@jd.com.

nitely a large amount of feature correspondences cannot be found in dealing with high resolution images, which consequently leads to a less complete reconstructed 3D models.

In this paper, we propose a novel selective joint bilateral propagation upsampling method for dense unstructured Multi-View Stereo, which can both benefit its runtime efficiency and reconstruction completeness. By downsampling the input high resolution images, we use a state-of-the-art MVS method (such as COLMAP [21]) to produce depth and normal maps in low resolution with higher efficiency. Next we develop a novel joint bilateral propagation framework to upsample the low resolution depth and normal maps into high resolution ones with geometry details simultaneously enriched, as shown in Figure 1. Finally, we fuse these resized maps to get final refined 3D models. We have conducted extensive experiments on ETH3D benchmark [25] compared with related methods. The results show that our method can not only greatly reduce the reconstruction time of the traditional 3D reconstruction methods, but also get more dense 3D models.

In summary, the main advantages of the proposed method are as follows:

- Our method is efficient, and can handle high resolution images for MVS to achieve high performance.
- Our upsampling method can produce more complete dense 3D reconstruction models.
- Our upsampling method is simple and easy to implement, and has the generalization ability to be applied to many different MVS systems.

This paper is organized as follows, in section 2, we give the related works. we present the technical details behind this approach in section 3, and we provide the experimental results, comparisons and discussions in section 4. We conclude our paper in section 5, and also present the future research directions.

2 Related Work

With solid theory foundation [14,31] and advanced computational methods [2,9,32], a series of MVS systems [6,7,20,21] have been developed. These MVS methods can be divided into four major categories [26]: voxel-based methods, surface-based methods, feature-based methods and depth-based methods. In this paper, we only review the depth map fusing based methods since they are most relevant to our work.

These methods compute depth maps for each image firstly and then fuse them together to get dense 3D point models. [13] presented a MVS algorithm for scene reconstruction from community photos, which iteratively grew surface to produce depth maps from sparse points reconstructed from SIFT [18] feature points. Furukawa and Ponce [10] presented a PatchMatch-based MVS algorithm named PMVS, which iteratively generated scene geometry by expansion and filtering steps from an initial set of oriented patches. The follow-up work CMVS [8] clustered the scene to multiple independent sub-problems, which can be processed by PMVS individually at the cost of computation time, but can not handle the effective scale of input images and is not progressive. [39] presented a MVS algorithm within a probabilistic framework that jointly modeled pixel-level view selection and depth estimation, solving by EM-based view selection probability inference and depth propagation in a Hidden Markov Chain. [11] presented a massively parallel MVS algorithm, which iteratively generated depth and normal estimation at the same time within a slanted support window by a modified, diffusion-like propagation scheme. [24] presented one of the state-of-the-art MVS methods, which improved the work of [39] and developed a robust open software system. Compared to these prior works, our approach takes the least time obtaining more dense reconstruction models by performing reconstruction in the corresponding low resolution images. Recent learning-based methods [37,38] also show advanced sides to improve depth and get desired point clouds, whereas their performance relies highly on training data and requires low-resolution inputs. In contrast, our approach focuses on a different aspect that explores the capability of 3D reconstruction via high-resolution image guidance.

Image upsampling is an important operation in computer vision and graphics communities. Naive upsampling method such as bilinear interpolation will suffer from sharp edge blurring due to the smoothness prior inherent in the linear interpolation filters. Because of the limitation of this upsampling method, many improved upsampling methods are proposed. Based on the idea of bilateral filtering [30], joint bilateral upsampling [17] adopted the image color information and distance information to upsample the image, and can well preserve the edge characteristics of the image. Different from our approach, this method use all data in a window for weighted average interpolation regardless of the differences among pixels. Bilateral guided upsampling [4] utilized local information of the image operation kernel to fit a bilateral-space affine model between the low resolution input image and the low resolution output image, and then produced the high resolution output image by evaluating the model on the high resolution input image. Although this method is faster, their results relying on the interpolation of surround-

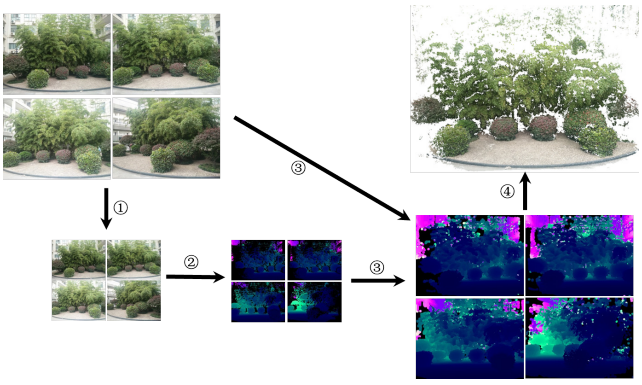


Fig. 2: Algorithm overview. For high resolution input unstructured images, we perform the following steps to obtain the final dense 3D scenes: 1 Downsampling, 2 MVS reconstruction, 3 selective joint bilateral propagation upsampling and 4 final depth and normal map fusion.

ing big size girds are less accurate than ours which not only uses bilateral weights but also propagates depth along the tangential planes in 3D space. The fast bilateral solver [3] is an edge-aware fast filter algorithm using domain-specific optimization in the bilateral space, which can be used in several different computer vision tasks. In contrast, our approach avoids solving optimization problems by selectively interpolating via jointing source color images and normal vectors information.

3 Fast Multi-View Stereo Upsampling

In Figure 2, we present the overview of the proposed method. We first downsample the input high resolution unstructured images into low resolution ones, and perform MVS (such as COLMAP [21]) on these low resolution images to obtain the low resolution depth and normal maps, Then, we develop a novel selective joint bilateral propagation upsampling method to upsample the low resolution depth and normal maps into high resolution ones. Finally, we construct the final dense 3D models by adaptively fusing the depth and normal maps. In the following subsections, we will provide the technical details for each step.

3.1 Initial Multi-View Stereo

[24] developed a robust open software system named COLMAP [21]. The MVS method of their system utilized the probabilistic framework of [39] which jointly estimates the depth and normal information, performs pixelwise view selection based on photometric and geometric priors using Generalized Expectancy Maximization (GEM) method of variational inference, and fuses the final dense models after filtering using photometric and geometric consistency constraints. COLMAP uses the GEM method to compute the Normalized Cross Correlation (NCC) between patches, which requires lots of computation and takes up a lot of time. In addition, this system needs to store all related image information in GPU memory, which is intractable when handling high resolution images. To avoid these problems, we perform MVS of COLMAP in corresponding low resolution images instead in the first stage of our reconstruction.

The MVS method of COLMAP applied photometric and geometric consistency strategies to filter the depth and normal maps before producing the 3D dense model in the final fusion step. In practice, given a correct depth value at a pixel in one image, sometimes the NCC between this image and the corresponding another image at this pixel is very low due to occlusions or illumination aberration. On the other hand, because of the repetitive scene structure or homogeneous texture region [15,35], an incorrect depth value may register high image similarity in another image, which makes it not reasonable to use photometric consistency to determine whether the depth value is accurate. Therefore, to obtain more accurate depth and normal maps, we need to denoise them before upsampling.

3.1.1 Depth Map Denoising

The depth and normal maps produced by COLMAP tend to have heavy-tailed noise distribution, which will produce heavy outliers when performing direct upsampling. We find that some sparse noise regions in the depth and normal maps are similar to salt-and-pepper noise. We use median filtering to denoise the depth maps, which a depth value will be replaced by the extracted median depth value if this depth value is sufficiently different from the median depth value (i.e., not within a factor of [0.95, 1.05]). Note that, we use a modified median filtering that pixels without depth values are not considered when extracting the median depth value in the window, as the pixels without depth values are set to zero in the depth maps which will disturb the median when including them in the computing. Figure 3 shows the depth map filtering result for an image with size of 400×300.

3.1.2 Normal Map Denoising

Accurate normal vectors can improve the accuracy and completeness of the reconstruction results, so it is nec-

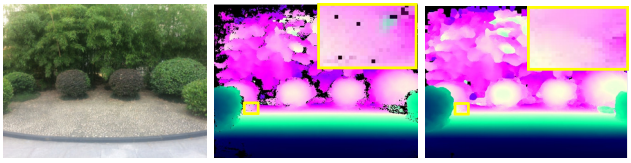


Fig. 3: Filtering the depth maps using median filter. The left one is the color image, the middle one is the depth map before median filtering, and the right one is the depth map after median filtering.

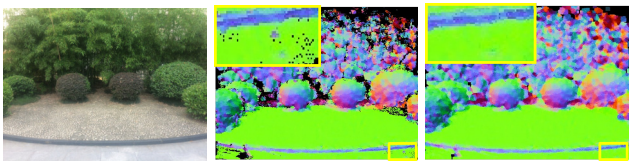


Fig. 4: Filtering the normal maps using median filter. The left one is the color image, the middle one is the normal map before median filtering, and the right one is the normal map after median filtering.

essary to denoise the normal maps as well. There are many normal filtering methods, such as mean and median normal filtering [34], bilateral normal filtering [40], L1-Median filtering [19] and so on. Our goal is simply to filter some sparse outlier normal vectors, so we use median normal filtering [34] to denoise the normal maps. Analogous to the depth map median filtering, we also exclude the pixels without normal vectors when extracting the median normal vector in the window. Figure 4 shows the denoised result of the normal map using median normal filtering method.

3.2 Joint Bilateral Propagation Upsampling

After above denoising processing, we can get more accurate low resolution depth and normal maps, the next step is to upsample the low resolution depth and normal maps to produce high resolution ones. There are many upsampling methods, including bilinear interpolation, joint bilateral upsampling [17], bilateral guided upsampling [4], The fast bilateral solver [3] and so on. Inspired by the joint bilateral upsampling, we propose a more effective upsampling method that couples the image content and normal vectors information to upsample the low resolution maps.

Joint bilateral upsampling [17] is an edge-preserving upsampling method and can produce compelling results for color images, while using this method to upsample the depth maps will introduce some troublesome problems, especially for depth maps containing pixels without depth values. In contrast, we notice that instead of

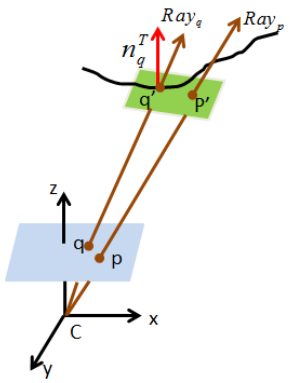


Fig. 5: Depth propagation on the local 3D tangent plane represented by a 3D point q' and its normal n_q^T in camera coordinate.

estimating the depth values of the pixels only depending on the spatial and range weights, propagating the depth values of these points on the 3D local tangent planes will deliver more accurate depth values. Thus, based on the spatial and range filter kernels, we further investigate the normal vectors information and introduce a selective joint bilateral propagation upsampling method.

Given a high resolution original color image $\tilde{I}$, low resolution depth map d and normal map n, the upsampling scheme is defined as follows:

$$\tilde{d}_p = \begin{cases} d_{p_\downarrow} & \text{if } p_\downarrow \text{ is integer} \\ \frac{1}{W}\sum_{i=1}^{n}(Pro(\tilde{d}_{q_i} \rightarrow \tilde{d}_p) f(||p-q_i||) g(||\tilde{I}_p - \tilde{I}_{q_i}||)) & \\ & \text{if } \tilde{d}_p = 0, \end{cases} \tag{1}$$

$$\tilde{n}_p = \begin{cases} n_{p_\downarrow} & \text{if } p_\downarrow \text{is integer} \\ n_{q_1} & \text{if } \tilde{d}_p = 0, \end{cases} \tag{2}$$

where $\tilde{d}_p$ and $\tilde{n}_p$ are the estimated depth and normal values in the high resolution image $\tilde{I}$ at pixel p. Let p and q denote coordinates of pixels in $\tilde{I}$, $p_\downarrow$ and $q_\downarrow$ denote the corresponding coordinates (may be fractional) in the low resolution image. f is the spatial Gaussian kernel defining the Euclidean distance between two pixels, and g is the range filter kernel defining the pixel intensity value difference between two pixels. W is a normalizing factor. $Pro(\tilde{d}_q \rightarrow \tilde{d}_p)$ is the propagation function we will describe later.

It involves two main steps. First, if $p_\downarrow$ is an integer, it means that the pixel p corresponds to pixel $p_\downarrow$ exactly. Under this circumstance, instead of interpolation with surrounding depth and normal values, we assign

the depth and normal values located at $p_{\downarrow}$ in low resolution maps to the depth and normal of corresponding p in the high resolution maps, which will produce more accurate results. With the first step, we can obtain the high resolution depth and normal maps with discrete values. In the second step, we apply a new interpolation method for those pixels without values in the high resolution depth and normal maps. $\tilde{d}_p = 0$ means that there is no depth and normal values located at p in high resolution maps. Let Ω represent a window located at p, and $q_i \in \Omega$ are neighboring pixels that have depth and normal values. Different from joint bilateral upsampling method [17], we perform selective weighted average interpolation using the most appropriate neighboring pixels. We first sort all q_i in descending order according to bilateral weight $f * g$, then select n pixels with largest weights to be considered as the candidate pixels for the interpolation at p. We do not estimate depth and normal for p when no pixel having depth and normal values in its neighborhood Ω. For the depth map, we use the weighted average of n candidate values as the depth of p. Because the requirements of the normal vectors are relatively weak than depth values in the final fusion step, we just use the normal of the pixel with the largest weight (named q_1) as the normal of p. That means, compared to the joint bilateral upsampling using all pixels in the window for weighted average interpolation, we only select fewer pixels with the largest weights for weighted average interpolation, which will get more accurate values, especially if the window contains pixels without depth and normal values or the depth and normal values of the window are discontinuous.

$$Pro(\tilde{d}_q \rightarrow \tilde{d}_p) = \frac{\tilde{d}_q Ray_q n_q^T}{Ray_p n_q^T} \tag{3}$$

Now we come to define $Pro(\tilde{d}_q \rightarrow \tilde{d}_p)$ shown as Equation 3. As shown in Figure 5, C is the camera center in 3D space. The blue plane is image plane, q is the neighboring pixel of p that has depth value and normal vector, its depth value is $\tilde{d}_q$, normal vector is n_q^T, its corresponding 3D point is q'. We can define a local 3D tangent plane at q' with its normal vector n_q^T. Ray_q and Ray_p are rays that connecting q and p with the camera center C, respectively. p' is the intersection 3D point of Ray_p with the tangent plane of q'. The equation means propagating the depth value of q (that is $\tilde{d}_q$) to p along the tangent plane defined by the 3D point of q' and its normal vector n_q^T (the green plane of the Figure 5). For all selected neighboring pixels of p, the propagated depth values are considered as candidate depth values for the interpolation at p. We use the propagated depth value instead of the depth value $\tilde{d}_q$ in our upsampling method, as propagating the depth values of adjacent points along their 3D tangent planes are closer to the truth depth values of the object, which can produce more accurate depth maps.

3.3 Depth and Normal Map Fusion

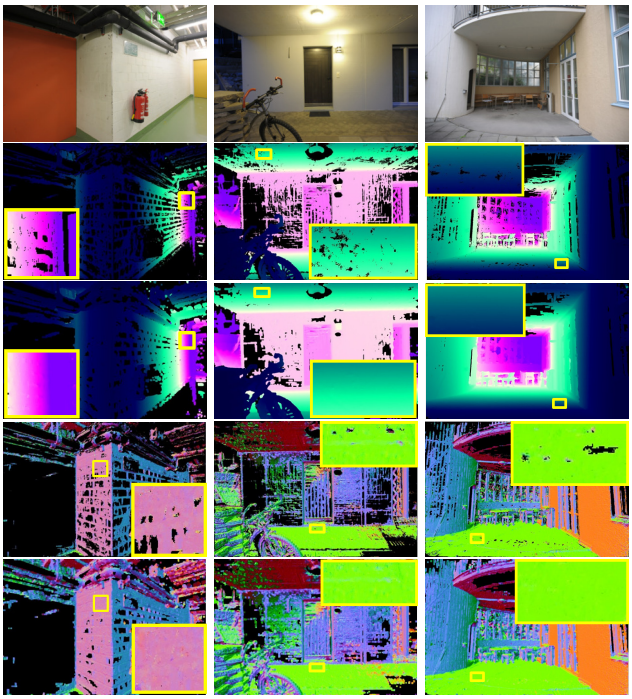
With above denoising and upsampling operation for low resolution depth and normal maps, we obtain more complete and accurate high resolution depth and normal maps. We use the fusion method of COLMAP to obtain the dense 3D model from the upsampling depth and normal maps. This fusion method is based on photometric and geometric constraints from multiple views. An inlier observation should meet the depth error constraint, the normal error constraint and the reprojection error constraint in the defined directed graph of the consistent pixels. When there is no remaining node that satisfies the three constraints in the graph, COLMAP fuses the cluster's elements of consistent pixels into a 3D point, which has median location and mean normal over all cluster elements. Please refer to [24] for more details.

4 Experiments and Discussions

To demonstrate the effectiveness of our proposed method, we have conducted extensive experiments on the high resolution MVS datasets of ETH3D benchmark [25], which is the latest and one of the most standard 3D reconstruction datasets including both indoor and outdoor scenes. Our algorithm is implemented in standard, single-threaded C++, and our experimental environment is a single PC machine with an Intel(R) Core(TM) i7-6700k CPU and 16GB RAM. We compare our method with state-of-the art methods both quantitatively and qualitatively.

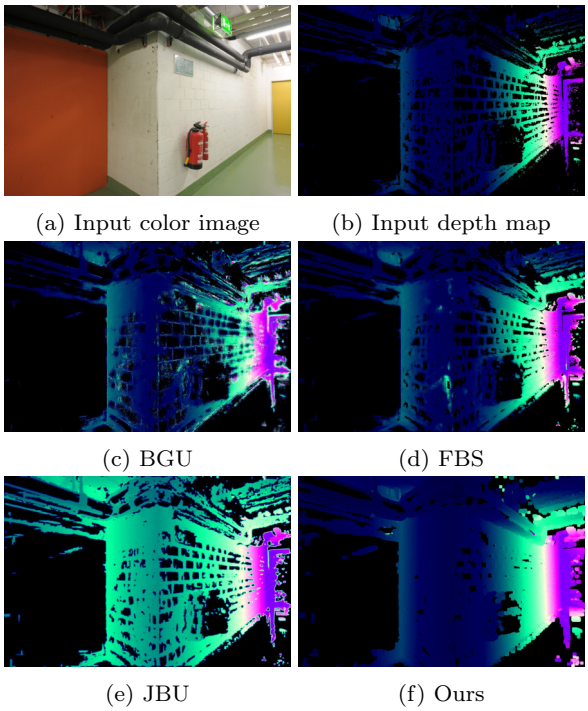
4.1 Denoising Evaluation

We first illustrate the advantage of using median filtering to denoise the depth and normal maps. The depth and normal maps produced by MVS methods usually contain noise, especially when the 3D scene structures are slim or heavily occluded (flowers and trees, etc). For those outliers containing salt-and-pepper noise, we use modified median filtering described in 3.1 to prune them. As shown in Figure 6 (window size is set as 5 for 1555×1035 depth and normal maps), the depth and normal maps are much better after preforming filtering.



(a) (b) (c)

Fig. 6: Depth and normal maps denoising results. (a)(b)(c) are pipes, terrace2 and terrace datasets of ETH3D benchmark. From top to bottom are the original color images, the depth maps before denoising, the depth maps after denoising, the normal maps before denoising and the normal maps after denoising.



(a) Input color image (b) Input depth map

(c) BGU (d) FBS

(e) JBU (f) Ours

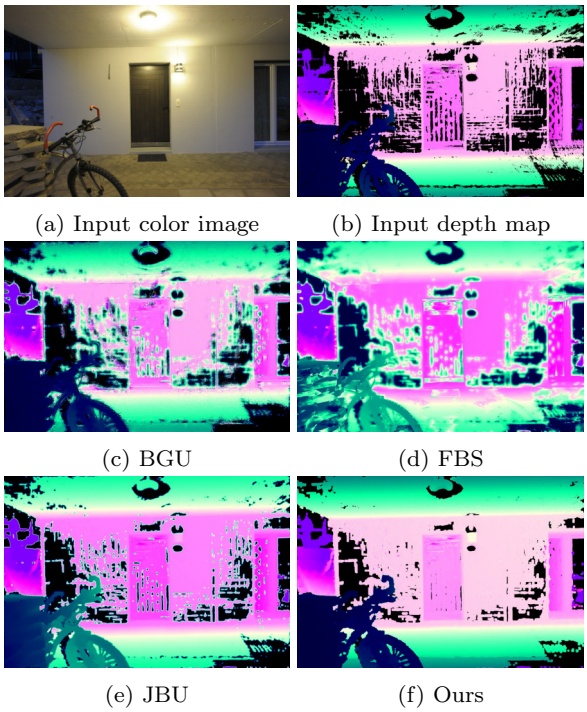
Fig. 7: Depth map upsampling comparisons using different methods on pipes dataset.

4.2 Upsampling Evaluation

We have compared our proposed method with several popular edge-aware upsampling methods, such as joint bilateral upsampling (JBU) [17], bilateral guided upsampling (BGU) [4], and the fast bilateral solver (FBS) [3]. For fair comparison, we use the depth and normal maps after our denoising optimization processing when applying these upsampling operators, and set the same value for the window size of the spatial filtering kernel. **Qualitative Evaluation:** Figure 7 and Figure 8 show the depth maps upsampling comparisons between our method and JBU, BGU, FBS (low resolution depth map is 1555×1035, the high resolution depth map is 6220×4141, the upsampling scale is 4, the window radius used for upsampling is 15, the spatial filter Gaussian σ_s and the range filter Gaussian σ_r are both 10). Figure 7(a) and Figure 8(a) are the low resolution color images and Figure 7(b) and Figure 8(b) are the corresponding low resolution depth maps estimated using COLMAP (Due to runtime and memory constraints of GPU, we set "num_iterations" from 5 to 3, and reduce the number of source images in the patch-match.cfg file from "__auto__, 20" to "__auto__, 15", the other parameters are all the same as the default parameters).

Figure 7(c)-(f) and Figure 8(c)-(f) are the upsampling depth maps using the above methods. From Figure 7(c) and Figure 8(c), we can observe that BGU does not work well on depth map upsmpling. As BGU is based on the idea that nearby pixels with similar color in the input also has similar color in the output, which works well for color images. But obviously the depth map does not match the corresponding color image in this means, so the affine models cannot be well fitted in the bilateral girds defined in this method. Similar to the BGU, FBS method performs upsampling by solving optimization problems in the simplified bilateral grids constructed by reference color image, which will be disturbed when the input depth map is discontinuous. In addition, the reference color image information is overused in the optimization, which will make the resulting depth values deviate from the true depth values.

JBU works well when dealing with edge-aware image upsampling. Figure 7(e) and Figure 8(e) show that JBU produces better upsampling results than BGU and FBS when performing upsampling of depth maps, but its runtime is longest. The above three methods will introduce erroneous depth values in the resulting depth maps, especially at the edges. This will change the depth ranges of the depth maps, and make the visualization



(a) Input color image (b) Input depth map

(c) BGU (d) FBS

(e) JBU (f) Ours

Fig. 8: Depth map upsampling comparisons using different methods on terrace2 dataset.

of the depth maps different. This is why the color of resulting depth maps generated by the above three methods will vary drastically in some areas, especially at the edges. From Figure 7 and Figure 8, we can observe that the visualization of our results are most similar to the input depth maps, showing that our method will not produce large erroneous depth values. The JBU method using all pixels in the window for weighted average interpolation, which will produce large depth error when the window size is large or the depth values in the window are not continuous. In contrast, we only select n depth values with the largest weights as candidate depth values (In our experiment, we set n to 4). What's more, we use the propagated depth values along their local 3D tangent planes instead of those candidate depth values to perform weighted average interpolation, which will make the depth results more accurate.

In Figure 10, we compare bilinear interpolation (BI), JBU, BGU, and FBS on the final 3D models based on the upsampled depth maps and normal maps, respectively. We also compare with the result of COLMAP performed on the original high resolution images. The upsampled depth and normal maps produced by JBU, BGU, and FBS contain many erroneous values, and many of these erroneous values will be filtered in the final fusion step. While there are relatively less erroneous values in our results, we can observe that our method obtain more dense 3D models, and the geometry details are better reconstructed.

In Figure 11, we present two more comparison results with COLMAP. In these examples, the scenes are relatively larger. From the results, we can observe that our method has greatly improved the completeness of the reconstruction results, and the geometry structures and details are also preserved well.

Quantitative Evaluation: For quantitative evaluation, we compare the accuracy, completeness and F1 score of the final dense 3D models reconstructed from different methods. Accuracy is defined as the fraction of the reconstruction which is closer to the ground truth than the evaluation threshold distance (Larger is better). Completeness is defined as the fraction of the ground truth which is closer to the reconstruction than the evaluation threshold distance (Larger is better). F1 score is defined as the harmonic mean of accuracy and completeness (Larger is better). The quantitative comparisons are performed on the ETH3D benchmark [25], which contains a software system specifically designed for evaluating the accuracy, completeness and F1 score compared with the ground truth. There are five tolerances (0.01m, 0.02m, 0.05m, 0.1m, 0.2m) for computing the accuracy, completeness and the F1 score. Their MVS datasets contain images with size over 6000×4000. We perform 4 times downsampling for images in the datasets, and then use the above methods to upsampling the depth and normal maps obtained by COLMAP. For fair comparison, the parameters of COLMAP for performing in high resolution images and low resolution images are the same.

Table 1 shows the comparison results between our method and COLMAP as well as several other upsampling methods on some datasets. The COLMAP rows are the results of COLMAP performed on the original high resolution images, and the other rows are the results of 4 times upsampling using BI, BGU, FBS, JBU and ours. As illustrated from the table, our method gets the highest completeness and highest F1 scores, while COLMAP gets the highest accuracy because of utilizing the full resolution images. Although our accuracy is not the highest, it is still comparable in these several upsampling methods, except for bilinear interpolation. Because our method performs propagating weighted average interpolation in a relative large window, which will introduce some less accuracy when the useful depth values in the window are less. Bilinear interpolation only interpolates the two nearest neighbor depth values, the results are relatively accurate, while it has little improvement on completeness. For full comparisons, we have conducted experiments on the whole ETH3D

	Accuracy					Completeness					F1 score				
pipes	0.01	0.02	0.05	0.1	0.2	0.01	0.02	0.05	0.1	0.2	0.01	0.02	0.05	0.1	0.2
COLMAP	**0.968**	**0.986**	**0.994**	**0.996**	**0.997**	0.139	0.212	0.344	0.456	0.572	0.244	0.349	0.511	0.626	0.727
BI	0.850	0.912	0.949	0.964	0.982	0.254	0.337	0.441	0.520	0.605	0.391	0.492	0.602	0.675	0.749
BGU	0.619	0.794	0.942	0.982	0.995	0.067	0.106	0.187	0.272	0.375	0.121	0.188	0.312	0.426	0.545
FBS	0.427	0.620	0.832	0.935	0.979	0.054	0.105	0.209	0.316	0.437	0.096	0.180	0.334	0.472	0.604
JBU	0.857	0.917	0.952	0.966	0.984	0.248	0.327	0.430	0.509	0.595	0.385	0.483	0.592	0.666	0.742
Ours	0.839	0.906	0.948	0.964	0.982	**0.275**	**0.368**	**0.476**	**0.553**	**0.631**	**0.415**	**0.523**	**0.634**	**0.703**	**0.768**
kicker	0.01	0.02	0.05	0.1	0.2	0.01	0.02	0.05	0.1	0.2	0.01	0.02	0.05	0.1	0.2
COLMAP	**0.927**	**0.967**	**0.987**	**0.992**	**0.994**	0.167	0.235	0.359	0.498	0.685	0.283	0.378	0.526	0.663	0.811
BI	0.767	0.856	0.919	0.947	0.970	0.370	0.477	0.631	0.765	0.899	0.499	0.612	0.748	0.846	0.933
BGU	0.520	0.695	0.881	0.957	0.987	0.204	0.275	0.389	0.505	0.665	0.293	0.394	0.540	0.662	0.794
FBS	0.320	0.475	0.687	0.821	0.935	0.119	0.181	0.282	0.399	0.598	0.173	0.262	0.400	0.537	0.729
JBU	0.768	0.855	0.919	0.947	0.971	0.363	0.467	0.619	0.752	0.885	0.493	0.604	0.740	0.838	0.926
Ours	0.763	0.854	0.919	0.946	0.969	**0.400**	**0.518**	**0.684**	**0.813**	**0.925**	**0.525**	**0.645**	**0.784**	**0.875**	**0.946**
playground	0.01	0.02	0.05	0.1	0.2	0.01	0.02	0.05	0.1	0.2	0.01	0.02	0.05	0.1	0.2
COLMAP	**0.694**	**0.809**	**0.934**	**0.983**	**0.995**	0.243	0.383	0.595	0.720	0.816	0.360	0.520	0.727	0.831	0.897
BI	0.591	0.741	0.902	0.967	0.987	0.423	0.575	0.750	0.839	0.896	0.493	0.648	0.819	0.898	0.939
BGU	0.438	0.616	0.845	0.946	0.985	0.263	0.412	0.573	0.674	0.772	0.329	0.494	0.683	0.787	0.866
FBS	0.198	0.311	0.543	0.757	0.894	0.130	0.247	0.418	0.551	0.694	0.157	0.275	0.472	0.638	0.781
JBU	0.593	0.743	0.904	0.968	0.989	0.418	0.568	0.741	0.833	0.892	0.490	0.644	0.815	0.896	0.938
Ours	0.592	0.742	0.902	0.966	0.987	**0.431**	**0.588**	**0.766**	**0.857**	**0.909**	**0.499**	**0.656**	**0.829**	**0.908**	**0.946**

Table 1: Quantitative comparisons about accuracy, completeness and F1 score of 3D dense models between several upsampling methods and ours on pipes, kicker and playground datasets.

datasets. Please refer to the supplementary material for more statistics.

There are some parameters in our selective joint bilateral propagation upsampling method, such as the spatial filter kernel (σ_s), the range filter kernel (σ_r) and the radius (R) of the upsampling window. We evaluate the influences of these parameters in Table 2. In Table 2, we make more rigorous comparison using three minimum tolerances (0.01m, 0.02m, 0.05m) to compute the accuracy, completeness and the F1 score. We can see that the accuracy of the reconstruction results using smaller radius is better than the results using bigger radius, but the completeness of them is adverse. In general, the F1 score of the bigger radius is better. We find that using bigger σ_r will produce more accuracy results, which indicates that spatial weights are more important than the range weights. The table shows that the completeness and F1 score are highest when using smaller σ_s, bigger σ_r and bigger radius to upsampling the depth and normal maps.

Time Complexity: The parameters of COLMAP are the same when performing original high resolution images and corresponding low resolution images. Theoretically, the runtime of processing low resolution images should be equal to the runtime of processing corresponding high resolution images divided by the square of the downsampling rate. As the MVS of COLMAP is performed on the GPU, it requires a large amount of memory of GPU. The GPU requirement will be reduced by square of downsampling rate when processing corresponding low resolution images. Our upsampling

	Completeness			Accuracy			F1 score		
pipes	0.01	0.02	0.05	0.01	0.02	0.05	0.01	0.02	0.05
σ_s=10,σ_r=10,R=10	0.2730	0.3638	0.4696	0.8411	0.9074	0.9479	0.4122	0.5194	0.6281
σ_s=15,σ_r=10,R=10	0.2723	0.3629	0.4692	0.8409	0.9073	0.9478	0.4114	0.5185	0.6277
σ_s=10,σ_r=15,R=10	0.2740	0.3651	0.4700	**0.8415**	**0.9077**	**0.9482**	0.4134	0.5207	0.6285
σ_s=10,σ_r=10,R=15	0.2753	0.3676	**0.4761**	0.8390	0.9058	0.9477	0.4145	0.5230	0.6338
σ_s=15,σ_r=10,R=15	0.2743	0.3663	0.4740	0.8388	0.9058	0.9476	0.4135	0.5217	0.6319
σ_s=10,σ_r=15,R=15	**0.2764**	**0.3692**	**0.4761**	0.8391	0.9061	0.9481	**0.4159**	**0.5246**	**0.6339**

Table 2: Quantitative comparisons about completeness, accuracy and F1 score of 3D reconstruction results produced by our selective joint bilateral propagation upsampling method with different parameters (the spatial filter kernel (σ_s), the range filter kernel (σ_r) and the radius (R) of the upsampling window) on pipes dataset.

method is highly parallelized and suitable for multi-thread implementations, and the runtime can almost be ignored compared with the runtime of COLAMP. So overall the runtime of our MVS is slightly higher than the runtime of COLMAP performed on low resolution images, and is approximately equal to the runtime of COLMAP performed on original resolution images divided by the square of downsampling rate. We run MVS of COLMAP for original high resolution images on our workstation with four NVIDIA GeForce GTX 1080 Ti GPUs (11 GB frame buffer on each card), and the runtime of processing one image on our GPU to get depth and normal maps on pipes datasets is about 220s.

Table 3 shows the runtime of upsampling a depth map using the above several methods (the low resolution depth map image is 1555×1035, and the corresponding high resolution depth map image is 6220×4141, the upsampling scale is 4, the above algorithms

	BGU	FBS	JBU	Ours
Time (s)	6.7	139.8	822	39

Table 3: The runtime comparison of upsampling a low resolution depth map with 1555×1035 to high resolution one with 6220×4141 between several upsampling methods and ours on the pipes dataset.

are processing using a single thread on a single Intel(R) Core(TM) i7-6700k CPU. The spatial filter Gaussian σ_s and the range filter Gaussian σ_r are both 10 for all the upsampling methods, the radius of window used in JBU and ours are 15. Because FBS uses YUV color images to construct simplified bilateral grids, we only set the spatial filter σ_s to 40 and the other parameters are default parameters). BGU method is the fastest because of interpolating on the bilateral grids, but gets poor results. FBS method solves least-squares optimization problem on simplified bilateral grids, the runtime is related to the size of the simplified bilateral grids, which will require a lot of time when the size of simplified bilateral grids are large. JBU method requires the most time for upsampling, because this method performs a large number of coordinate transformation between low resolution images and high resolution images, requiring large amount of time for the division operation. Our method is much faster than FBS and JBU. In addition, our upsampling method performs both depth and normal maps upsampling simultaneously.

Limitations: Our method also has its limitations. First, although the completeness and F1 score are highest for the 3D dense models obtained by our method, the accuracy is not as high as the results obtained by original high resolution images. Second, our method relies on the depth and normal maps obtained by COLMAP. If the depth and normal maps are not accurate enough, our upsampling method will not get accurate results. For example, as shown in Figure 9, the low resolution depth and normal maps lack a lot of data in the box areas, the high resolution depth and normal maps as well as the dense 3D model produced by our method also have no data in the corresponding areas.

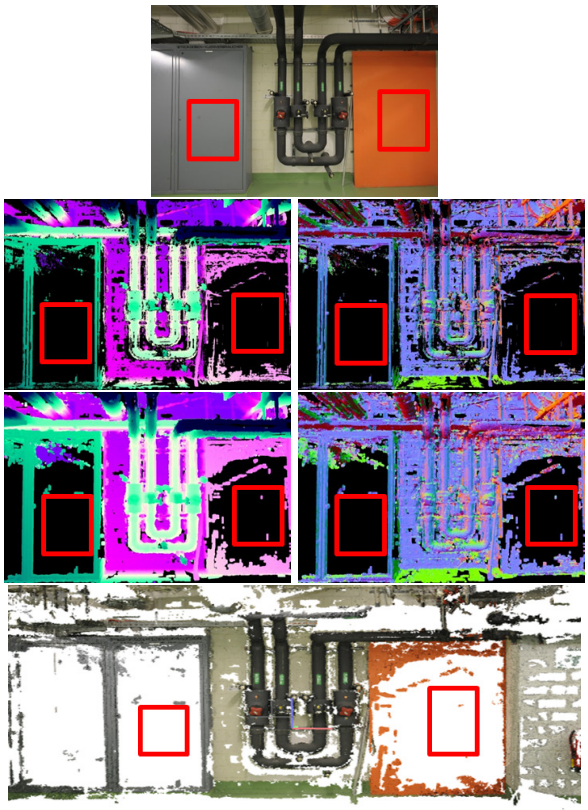


Fig. 9: For an input image (the first row), the depth and normal maps obtained by COLMAP exhibit a lot of missing information in the box areas (the second row). Our upsampled depth and normal maps on the third row and the final 3D dense model on the last row also exhibit missing information in those areas.

5 Conclusion and Future Work

In this paper, we have proposed a selective joint bilateral propagation strategy for unstructured Multi-View Stereo. Our upsampling method applies the high resolution color images as guided images, and utilizes the normal vectors to perform depth propagation interpolation on the local tangent planes of 3D scene, which contributes to the accuracy and completeness improvement. Our method also greatly reduces the reconstruction time of the traditional MVS methods.

The proposed framework is universal and can be used in those MVS methods based on depth map fusing as well as feature point growing strategies. In the future, we will further accelerate the proposed method. One idea is to borrow the idea of bilateral guided upsampling [4] into our method, and perform ours method for high resolution images on consumer smartphones in real time.

Acknowledgments

This work was partly supported by The National Key Research and Development Program of China (2017YFB1002600), the NSFC (No. 61672390, No. 41201404), Wuhan Science and Technology Plan Project (No. 2017010201010109), and Key Technological Innovation Projects of Hubei Province (2018AAA062).

Compliance with Ethical Standards

Conflicts of Interest: The authors have no conflict of interest.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10), 105–112 (2011)
2. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: ECCV, pp. 29–42 (2010)
3. Barron, J.T., Poole, B.: The fast bilateral solver. In: ECCV, pp. 617–632 (2016)
4. Chen, J., Adams, A., Wadhwa, N., Hasinoff, S.W.: Bilateral guided upsampling. ACM Transactions on Graphics **35**(6), 203 (2016)
5. Fu, Y., Yan, Q., Yang, L., Liao, J., Xiao, C.: Texture mapping for 3d reconstruction with rgb-d sensor. In: CVPR, pp. 4645–4653 (2018)
6. Fuhrmann, S., Langguth, F., Moehrle, N., Waechter, M., Goesele, M.: Mve: an image-based reconstruction environment. Computers & Graphics **53**, 44–53 (2015)
7. Furukawa, Y.: http://www.di.ens.fr/cmvs (2011)
8. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: CVPR, pp. 1434–1441 (2010)
9. Furukawa, Y., Hernández, C., et al.: Multi-view stereo: A tutorial. Foundations and Trends® in Computer Graphics and Vision **9**(1-2), 1–148 (2015)
10. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(8), 1362–1376 (2010)
11. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: ICCV, pp. 873–881 (2015)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR, pp. 3354–3361 (2012)
13. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: ICCV, pp. 1–8 (2007)
14. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
15. Heinly, J., Dunn, E., Frahm, J.M.: Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction (2014)
16. Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In: CVPR, pp. 3287–3295 (2015)
17. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics **26**(3), 96 (2007)
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
19. Lu, X., Chen, W., Schaefer, S.: Robust mesh denoising via vertex pre-filtering and l1-median normal filtering. Computer Aided Geometric Design **54**, 49–60 (2017)
20. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: Openmvg: Open multiple view geometry. In: International Workshop on Reproducible Research in Pattern Recognition, pp. 60–74 (2016). URL https://github.com/openMVG/openMVG
21. Schonberger, J.L.: https://colmap.github.io/ (2016)
22. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR, pp. 4104–4113 (2016)
23. Schonberger, J.L., Radenovic, F., Chum, O., Frahm, J.M.: From single image query to detailed 3d reconstruction. In: CVPR, pp. 5126–5134 (2015)
24. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV, pp. 501–518 (2016)
25. Schöps, T., Schönberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: CVPR, vol. 3 (2017)
26. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR, vol. 1, pp. 519–528 (2006)
27. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR, pp. 1297–1304 (2011)
28. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM Transactions on Graphics **25**(3), 835–846 (2006)
29. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR, pp. 1–8 (2008)
30. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV, pp. 839–846 (1998)
31. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustmenta modern synthesis. In: International Workshop on Vision Algorithms, pp. 298–372 (1999)
32. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Multicore bundle adjustment. In: CVPR, pp. 3057–3064 (2011)
33. Xiao, C., Zheng, W., Miao, Y., Zhao, Y., Peng, Q.: A unified method for appearance and geometry completion of point set surfaces. The Visual Computer **23**(6), 433–443 (2007)
34. Yagou, H., Ohtake, Y., Belyaev, A.: Mesh smoothing via mean and median filtering applied to face normals. In: GMP, p. 124 (2002)
35. Yan, Q., Yang, L., Zhang, L., Xiao, C.: Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context. In: CVPR (2017)
36. Yang, L., Yan, Q., Xiao, C.: Shape-controllable geometry completion for point cloud models. The Visual Computer **33**(3), 385–398 (2016)
37. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: ECCV (2018)
38. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: CVPR (2019)
39. Zheng, E., Dunn, E., Jojic, V., Frahm, J.M.: Patchmatch based joint view selection and depthmap estimation. In: CVPR, pp. 1510–1517 (2014)
40. Zheng, Y., Fu, H., Au, O.K.C., Tai, C.L.: Bilateral normal filtering for mesh denoising. IEEE Transactions on Visualization and Computer Graphics **17**(10), 1521–1530 (2011)

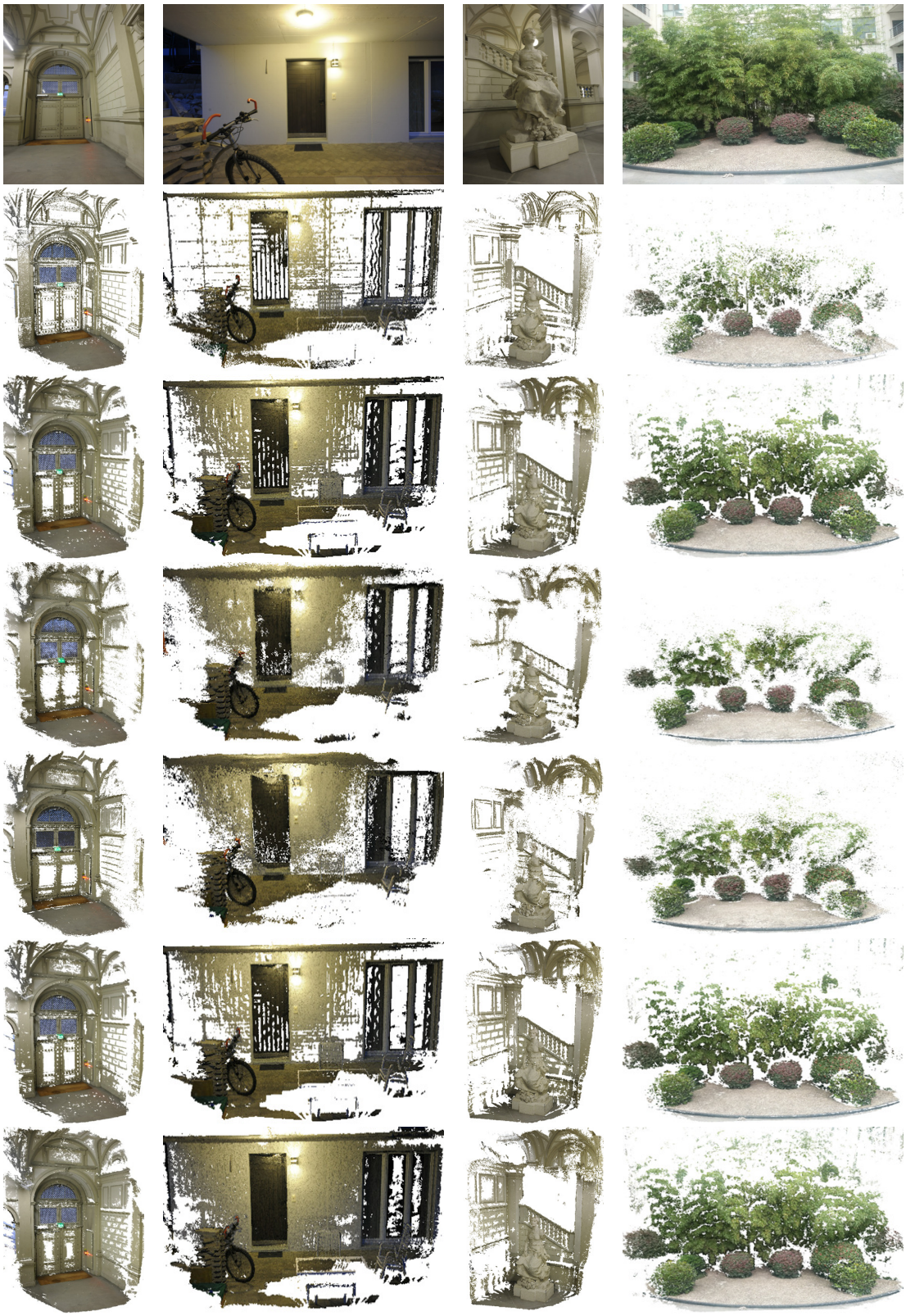


Fig. 10: Multi-View Stereo comparisons. Those datasets are door, terrace2, statue and bamboo from left to right (the first three are from the datasets of ETH3D benchmark, the last one is ours). From top to bottom, the first row is the original images, the second row is the result of COLMAP performed on the original high resolution images, the third to sixth rows are the results of BI, BGU, FBS, JBU and ours (4 time upsample).

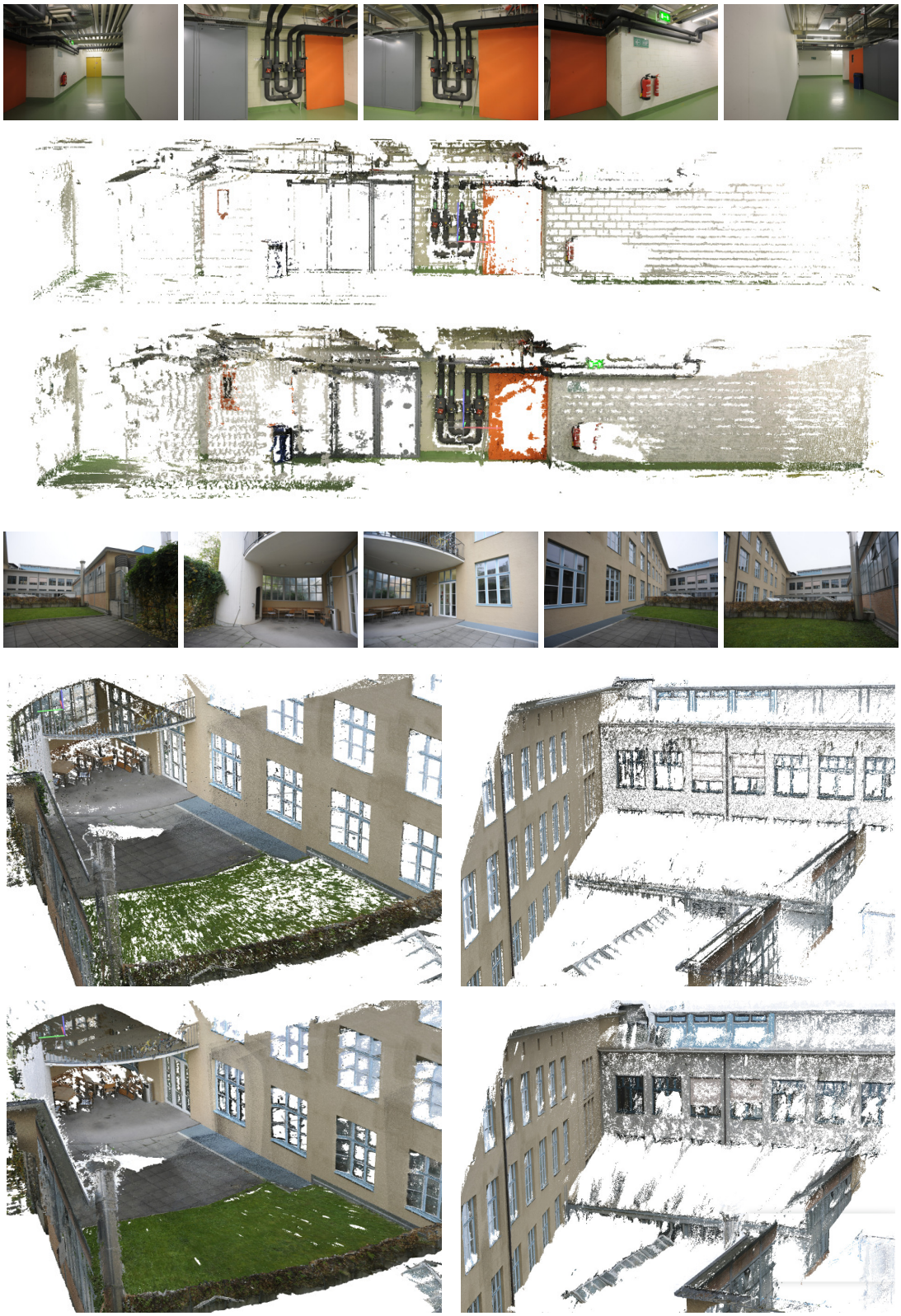


Fig. 11: Multi-View Stereo comparisons. The top row presents some color images of pipes dateset, the second row is the result of COLMAP on original color images, and the third row is our result. The fourth row presents some color images of terrace dataset, the fifth row presents the results of COLMAP on original color images (two parts of the scene), and last row presents our results (two parts of the scene).