Semi-supervised Three-dimensional Reconstruction Framework with Generative Adversarial Networks

Chong Yu
NVIDIA Semiconductor Technology Co., Ltd.
No.5709 Shenjiang Road, No.26 Qiuyue Road, Shanghai, China 201210
dxxzdxxz@126.com, chongy@nvidia.com

Abstract

*Because of the intrinsic complexity in computation, three-dimensional (3D) reconstruction is an essential and challenging topic in computer vision research and applications. The existing methods for 3D reconstruction often produce holes, distortions and obscure parts in the reconstructed 3D models, or can only reconstruct voxelized 3D models for simple isolated objects. So they are not adequate for real usage. From 2014, the Generative Adversarial Network (GAN) is widely used in generating unreal datasets and semi-supervised learning. So the focus of this paper is to achieve high-quality 3D reconstruction performance by adopting the GAN principle. We propose a novel semi-supervised 3D reconstruction framework, namely SS-3D-GAN, which can iteratively improve any raw 3D reconstruction models by training the GAN models to converge. This new model only takes real-time 2D observation images as the weak supervision and doesn't rely on prior knowledge of shape models or any referenced observations. Finally, through the qualitative and quantitative experiments & analysis, this new method shows compelling advantages over the current state-of-the-art methods on the **Tanks & Temples** reconstruction benchmark dataset.*

1. Introduction

In computer graphics and computer vision areas, three-dimensional (3D) reconstruction is the technique of recovering the shape, structure and appearance of real objects. Because of its abundant and intuitional expressive force, 3D reconstruction is widely applied in construction [3], geomatics [16], archaeology [11], game [8], virtual reality [20] areas, etc. Researchers have made significant progress on 3D reconstruction approaches in the past decades. The 3D reconstructed targets can be some isolated objects [2, 25] or large scale scene [9, 22, 27]. For different reconstructed targets, researchers attempt to represent 3D objects based on voxels [2], point clouds [22], or meshes and textures [23]. The state-of-the-art 3D reconstruction methods can be divided into following categories.

- Structure from motion (SFM) based method
- RGB-D camera based method
- Shape prior based method
- Generative-Adversarial based method

In this paper, we propose a semi-supervised 3D reconstruction framework named SS-3D-GAN. It combines latest GAN principle as well as advantages in traditional 3D reconstruction methods like SFM and multi-view stereo (MVS). By the fine-tuning adversarial training process of 3D generative model and 3D discriminative model, the proposed framework can iteratively improve the reconstruction quality in semi-supervised manner. The main contribution of this paper can be summarized as following items.

- SS-3D-GAN is a weakly semi-supervised framework. It only takes collected 2D observation images as the supervision, and has no reliance of 3D shape priors, CAD model libraries or any referenced observations.
- Unlike many state-of-the-art methods which can only generate voxelized objects or some simple isolated objects such as table, bus, SS-3D-GAN can reconstruct complicated 3D objects, and still obtains good results.
- By establishing evaluation criterion of 3D reconstructed model with GAN, SS-3D-GAN simplifies and optimizes the training process. It makes the application of GAN to complex reconstruction possible.

2. SS-3D-GAN for Reconstruction

2.1. Principle of SS-3D-GAN

Imagine the following situation, a person wants to discriminate the real scene and artificially reconstructed scene model. So firstly, he observes in the real 3D scene. Then he observes in the reconstructed 3D scene model at exactly the same positions and viewpoints as he observes in the real 3D

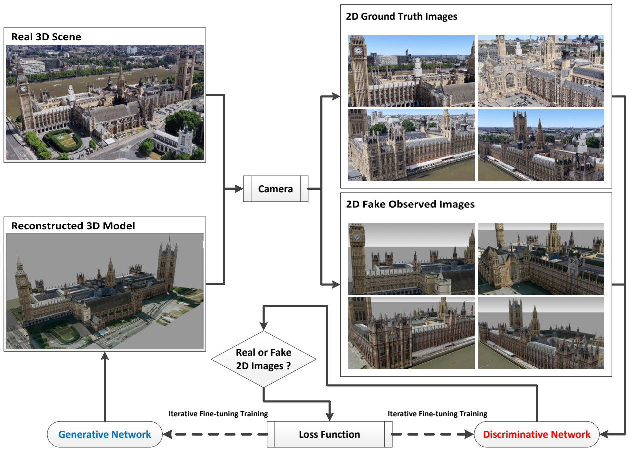


Figure 1. Principle and workflow chart of SS-3D-GAN

scene. If all the observed 2D images in the reconstructed 3D scene model are exactly the same as the observed 2D images in the real 3D scene. Then this person can hardly differentiate reconstructed 3D scene model from the real 3D scene. For the purpose of 3D reconstruction, we can accumulate the difference between each observed 2D image in the reconstructed 3D model and the observed 2D image in the real 3D scene. If the difference at each position and viewpoint is small enough, we can regard it as a high-quality 3D reconstruction result. Fig. 1 illustrates this concept.

To combine the purpose of 3D reconstruction and GAN model, we propose the novel 3D reconstruction framework, namely SS-3D-GAN. For the proposed SS-3D-GAN model, it consists of the 3D generative network and the 3D discriminative network. Here, we can imagine the discriminative network as the observer. So the purpose of the generative network is to reconstruct new 3D model which is aligned with the real 3D scene, and attempts to confuse the discriminative network, i.e., the observer. While the purpose of the discriminative network is to classify reconstructed 3D model by the generative network and the real 3D scene. When the SS-3D-GAN model achieves Nash Equilibrium, i.e., the generative network can reconstruct 3D model which exactly aligns with the character and distribution of real 3D scene. And at the same time, the discriminative network returns the classification probability 0.5 for each observation pair of generated and real 3D scene. This is also aligned with the evaluation criterion of 3D reconstructed. In conclusion, solving the 3D reconstruction problem is equal to making the SS-3D-GAN model well-trained and converged.

2.2. Workflow of SS-3D-GAN

Firstly, to start the training process of SS-3D-GAN, we generate a rough 3D reconstructed model as the initialization of generative network. The representation of the 3D model is aligned with "ply" model format. The vertex and color info are separately stored in triple structures. To generate this initial 3D model, we use the camera to collect video stream as ground truth. The video stream is served as the raw data to generate 2D observed images, camera trajectory, as well as the original rough 3D model with spatial mapping method [17]. This method generates 3D model based on depth sensing estimation by comparing the differentials between adjacent frames. The 2D observed images captured from video stream are also served as ground-truth image dataset.

After the initialization, we can start the iterative fine-tuning training process of generative network and discriminative network in SS-3D-GAN. The overall workflow of SS-3D-GAN is also shown in Fig. 1.

As SS-3D-GAN needs to get the observed 2D images in the reconstructed 3D scene model, we import the reconstructed 3D model into Blender (a professional and open-source 3D computer graphics software toolset) and OpenDR [14]. OpenDR is a differentiable renderer that approximates the true rendering pipeline for mapping 3D models to 2D scene images, as well as back-propagating the gradients of 2D scene images to 3D models. The differentiable renderer is necessary. Because GAN structure needs to be fully differentiable to pass the discriminators gradients to update the generator.

In the Blender, we setup a virtual camera with the same optical parameters as the real camera to collect video stream in real 3D scene. As the camera trajectory is calculated while processing ground truth video stream, we move the virtual camera along this trajectory, and use renderer to capture the 2D images at the same positions and viewpoints as in the real 3D scene. Hence, we are able to generate the same number of 2D fake observed images in the reconstructed 3D model and 2D ground truth images captured from video stream.

When the 2D scene images of ground truth and fake observation are ready, we use the discriminative network to classify them as the real or fake 2D images. At the same time, we calculate the overall loss value through loss function. With the overall loss, SS-3D-GAN will continue fine-tuning training process, and create new 3D generative network and 3D discriminative network. The new trained 3D generative network will generate a new reconstructed 3D model for virtual camera to observe. And the new observed fake 2D images as well as the ground-truth images will be fed into the new 3D discriminative network for classification. The workflow of SS-3D-GAN will iteratively train and create new 3D generative and discriminative networks, until the overall loss converges to the desired value.

2.3. Loss Function Definition

The overall loss function of SS-3D-GAN consists of two parts: reconstruction loss L_{Recons} and cross entropy loss $L_{SS-3D-GAN}$. So the loss function is written as follows:

$$L_{Overall} = L_{Recons} + \lambda \cdot L_{SS-3D-GAN}, \qquad (1)$$

where λ is parameter to adjust percentages between reconstruction loss and cross entropy loss.

In the SS-3D-GAN framework, the reconstruction quality is judged by the discriminative network. So the reconstruction loss is provided by calculating the differences between real and fake 2D scene image pairs from the discriminator. In this paper, three quantitative image effect indicators are applied to measure the differences [26]. Peak Signal to Noise Ratio (PSNR) indicator is applied to assess the effect difference from the gray-level fidelity aspect. Structural Similarity (SSIM) [21] indicator which is an image quality assessment indicator based on the human vision system is applied to assess the effect difference from the structure-level fidelity aspect. Normalized Correlation (NC) indicator which represents the similarity between the same dimension images is also taken into consideration. The definitions of these three evaluation indicators are as follows.

$$PSNR(\boldsymbol{x},\boldsymbol{y}) = 10\log_{10}\left(\frac{(MAX_I)^2}{MSE\,(\boldsymbol{x},\boldsymbol{y})}\right), \tag{2}$$

where MAX_I is the maximum possible pixel value of scene images: $\boldsymbol{x}$ and $\boldsymbol{y}$. $MSE(\boldsymbol{x},\boldsymbol{y})$ represents the Mean Squared Error (MSE) between scene images: $\boldsymbol{x}$ and $\boldsymbol{y}$.

$$SSIM(\boldsymbol{x},\boldsymbol{y}) = \frac{(2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{y}} + C_1)\,(2\sigma_{\boldsymbol{xy}} + C_2)}{\left(\mu_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{y}}^2 + C_1\right)\left(\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{y}}^2 + C_2\right)}, \tag{3}$$

where $\mu_{\boldsymbol{x}}$ and $\mu_{\boldsymbol{y}}$ represent the average grey values of scene images. Symbol $\sigma_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{y}}$ represent the variances of scene images. Symbol $\sigma_{\boldsymbol{xy}}$ represents covariance between scene images. Symbol C_1 and C_1 are two constants which are used to prevent unstable results when either $\mu_{\boldsymbol{x}}^2+\mu_{\boldsymbol{y}}^2$ or $\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2$ is very close to zero.

$$NC(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}, \tag{4}$$

where symbol $\boldsymbol{x}\cdot\boldsymbol{y}$ indicates the inner product of scene images, operation $\|*\|$ indicates *Euclidean* norm of $\boldsymbol{x}$ and $\boldsymbol{y}$.

SSIM indicator value of two images is in the range of 0 to 1. NC indicators value is in the range of -1 to 1. If the value of SSIM indicator or NC indicator is closer to 1, it means there is less difference between image $\boldsymbol{x}$ and image $\boldsymbol{y}$. For PSNR indicator, the common value is in the range of 20 to 70 dB. So we apply the extended sigmoid function to regulate its value to the range of 0 to 1.

$$E\_Sigm\,(PSNR\,(\boldsymbol{x},\boldsymbol{y})) = \frac{1}{1+e^{-0.1(PSNR(\boldsymbol{x},\boldsymbol{y})-45)}}, \tag{5}$$

So the reconstruction loss is written as follows:

$$L_{Recons} = \sum_{j=1}^{N}\Big\{\alpha\cdot\left[1 - E\_Sigm\left(PSNR_{G_jF_j}\right)\right] + \beta\cdot\left(1 - SSIM_{G_jF_j}\right) + \gamma\cdot\left(1 - NC_{G_jF_j}\right)\Big\} \tag{6}$$

where α,β,γ are the parameters to adjust the percentages among the loss values from PSNR, SSIM and NC indicators. The subscript G_jF_j represent the pair of ground truth and fake observed 2D scene images. The symbol N represents the total amount of 2D image pairs. In the next session, we will discuss details of cross entropy loss for SS-3D-GAN.

2.4. SS-3D-GAN Network Structure

As aforementioned, the 3D model learned in SS-3D-GAN is mesh data. The traditional method to handle mesh 3D data is sampling it into voxel representations. Then mature convolutional neural network (CNN) concept can be applied to this grid-based structured data, such as volumetric CNN [18]. However, the memory requirement is $O(M^3)$, which will dramatically increase with the size of target object. The memory boundary also leads to the low resolution and poor visual quality of 3D models.

Here, 3D mesh data can be represented by vertices and edges. Because vertices and edges are basic elements of graph, so we use the graph data structure to represent the 3D model in SS-3D-GAN as $\boldsymbol{G}_{3D} = (\boldsymbol{V},\boldsymbol{A})$, where $\boldsymbol{V}\in\boldsymbol{R}^{N\times F}$ is the matrix with N vertices and F features each. $\boldsymbol{A}\in\boldsymbol{R}^{N\times N}$ is the adjacency matrix, which defines the connections between the vertices in $\boldsymbol{G}_{3D}$. The element a_{ij} is defined as 1 if there is an edge between vertex i and j. Other elements are 0 in matrix $\boldsymbol{A}$ if no edges are connected. The memory requirement of $\boldsymbol{G}_{3D}$ is $O(N^2+FN)$, which is an obvious memory saving over the voxel representation memory cost [4].

Then we can apply Graph CNN [4] to $\boldsymbol{G}_{3D}$. We allow a graph be represented by L adjacency matrices at the same time instead of one. This can help SS-3D-GAN to learn more parameters from the same sample and apply different filters to emphasize different aspects of the data. The input data for a graph convolutional layer with C filters includes:

$$\boldsymbol{V}_{in}\in\boldsymbol{R}^{N\times F}, \boldsymbol{A}\in\boldsymbol{R}^{N\times N\times L}, \boldsymbol{H}\in\boldsymbol{R}^{L\times F\times C}, \boldsymbol{b}\in\boldsymbol{R}^{C}, \tag{7}$$

where $\boldsymbol{V}_{in}$ is an input graph, $\boldsymbol{A}$ is a tensor to represent L adjacency matrices for a particular sample, $\boldsymbol{H}$ is the graph filter tensor, and $\boldsymbol{b}$ is the bias tensor. The filtering operation is shown as follows [4].

$$\boldsymbol{V}_{out} = (\boldsymbol{A}\times\boldsymbol{V}_{in}^T)_{(2)}\boldsymbol{H}_{(3)}^T + \boldsymbol{b}, \boldsymbol{V}_{out}\in\boldsymbol{R}^{N\times C} \tag{8}$$

Like traditional CNN, this operation can be learned through back-propagation and it is compatible with operations such as ReLU, batch normalization, etc.

For SS-3D-GAN, the discriminative network needs brilliant classification capability to handle the complex 2D scene images which is the projection of 3D space. So we apply the 101-layer ResNet [10] as the discriminative network. The structure of generative network is almost the same as the discriminative network. Because the generative network needs to reconstruct the 3D model, so we change

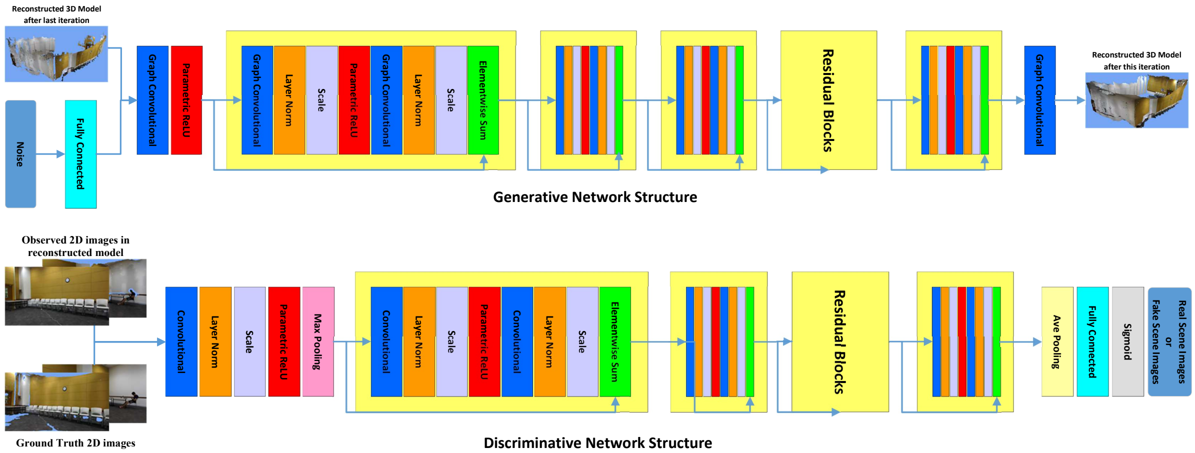


Figure 2. Details of generative network structure and discriminative network structure in SS-3D-GAN

all the convolutional layers to graph convolutional layers. The typical ResNet applies batch normalization to achieve the stable training performance. However, the introduction of batch normalization makes the discriminative network to map from a batch of inputs to a batch of outputs. In the SS-3D-GAN, we want to keep the mapping relation from a single input to a single output. We replace batch normalization by layer normalization for the generative and discriminative networks to avoid the correlations introduced between input samples. We also replace ReLU with parametric ReLU for the generative and discriminative networks to improve the training performance. Moreover, to improve the convergence performance, we use Adam solver instead of stochastic gradient descent (SGD) solver. In practice, Adam solver can work with a higher learning rate when training SS-3D-GAN. The detailed network structures are shown in Fig. 2.

Based on the experiments in [9], Wasserstein GAN (WGAN) with gradient penalty can succeed in training the complicated generative and discriminative networks like ResNet. So we introduce the improved training method of WGAN into SS-3D-GAN training process. The target of training the generative network G and discriminative network D is as follows.

$$\min_G \max_D \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathbf{P}_r}\left[D\left(\boldsymbol{x}\right)\right] - \mathop{\mathbf{E}}_{\tilde{\boldsymbol{x}}\sim\mathbf{P}_g}\left[D\left(\tilde{\boldsymbol{x}}\right)\right], \tag{9}$$

where symbol $\mathbf{P}_r$ is the real scene images distribution and symbol $\mathbf{P}_g$ is the generated scene images distribution. Symbol $\tilde{\boldsymbol{x}}$ is implicitly generated by generative network G. For the raw WGAN training process, the weight clipping is easy to result in the optimization difficulties including capacity underuse, gradients explosion or vanish. For improvement, the gradient penalty as a softer constraint is adopted instead. So the cross entropy loss for SS-3D-GAN is written as follows.

$$\begin{aligned} L_{SS-3D-GAN} = \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathbf{P}_r}\left[D\left(\boldsymbol{x}\right)\right] - \mathop{\mathbf{E}}_{\tilde{\boldsymbol{x}}\sim\mathbf{P}_g}\left[D\left(\tilde{\boldsymbol{x}}\right)\right] - \\ \theta \cdot \mathop{\mathbf{E}}_{\hat{\boldsymbol{x}}\sim\mathbf{P}_{\hat{\boldsymbol{x}}}}\left[\left(\left\|\nabla_{\hat{\boldsymbol{x}}} D\left(\hat{\boldsymbol{x}}\right)\right\|_2 - 1\right)^2\right], \end{aligned} \tag{10}$$

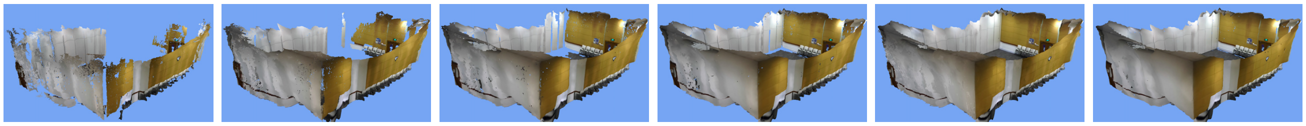
where θ is the parameter to adjust the percentage of gradient penalty in the cross entropy loss. $\mathbf{P}_{\hat{\boldsymbol{x}}}$ is implicitly defined as the dataset which is uniformly sampled along straight lines between pairs of points come from $\mathbf{P}_r$ and $\mathbf{P}_g$ distributions. The value of this cross entropy loss can quantitatively indicate the training process of SS-3D-GAN.

3. Experimental Results

3.1. Qualitative Performance Experiments

In qualitative experiments, we adopt ZED stereo camera as data collection tool. The ground truth dataset is collected by using stereo camera to scan over a meeting room. With the recorded video streams, we can extract the 2D scene images as the ground truth. At the same time, we can calculate the camera trajectory based on depth estimation by stereo camera. With the 2D scene images captured from stereo camera and the corresponding camera trajectory, we use spatial mapping to generate original rough 3D reconstructed model. Spatial mapping method represents the geometry of target scene as a single 3D triangular mesh. The triangular mesh is created with vertices, faces and normals attached to each vertex. To recover the surface of the 3D model, the 3D mesh should be colored by projecting the 2D images captured during spatial mapping process to mesh faces. During the spatial mapping, a subset of the camera images is recorded. Then each image is processed and assembled into a single texture map. Finally, this texture map will be projected onto each face of the 3D mesh using automatically generated UV coordinates [1].

With the initial rough 3D reconstructed model generated



(a) (b) (c) (d) (e) (f)

Figure 3. Reconstructed results of SS-3D-GAN. The reconstructed scene is an assembly hall. The size of the hall is about 23 meters in length, 11 meters in width and 5 meters in height. (a) shows the rough 3D model generated by spatial mapping method in the initialization stage. (b) to (f) show the reconstructed 3D models in the iterative fine-tuning training process of SS-3D-GAN. (b): 15 epochs, (c): 45 epochs, (d): 90 epochs, (e): 120 epochs. We can find the reconstructed models are from coarse to fine. Holes, distortions and obscure parts are greatly reduced by the SS-3D-GAN. (f) shows the ultimate reconstructed 3D model with small value in loss function (150 epochs).

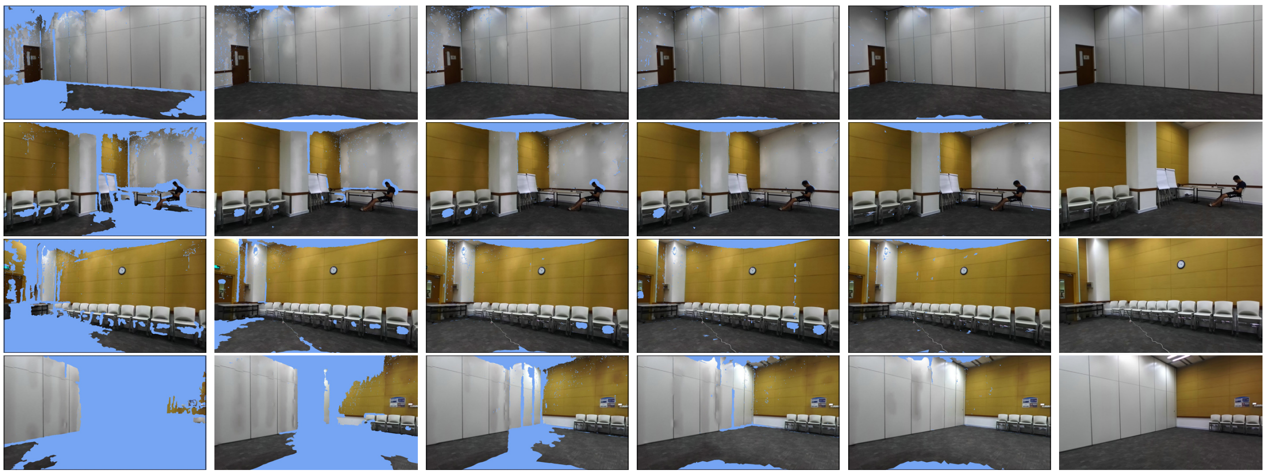


Figure 4. Observed 2D images in the reconstructed 3D models and in the real scene. We take four representative 2D images in each 3D model as the observed examples to illustrate the quality of 3D reconstructed models (They are shown in the same column). Column 1-5 are observed 2D images corresponding to 3D reconstructed models in Fig. 3(b-f). Column 6 are ground truth images which are observed in the real scene. The images in the same row are observed in the same position and viewpoint.

by spatial mapping (shown in Fig. 3(a)), we initialize parameters in loss functions. We set the value of parameters as follows: $\lambda = 0.7, \alpha = 0.25, \beta = 0.6, \gamma = 0.15, \theta = 10$. In this experiment, we use 600 scene images as weak supervision. The learning rate of generative and discriminative networks is 0.063. We use PyTorch as the framework, and train the SS-3D-GAN with the iterative fine-tuning process of 150 epochs.

Typical samples of reconstructed 3D model results are shown in Fig. 3. Comparison results of observed 2D images in the reconstructed 3D model and real scene are shown in Fig. 4. The results shown in Fig. 3 and Fig. 4 can prove the high quality of reconstructed 3D model and the corresponding 2D observations of SS-3D-GAN framework in qualitative aspect.

Typical samples of reconstructed 3D models of *Tanks and Temples* dataset are shown in Fig. 5 $\sim$ 7. Compared with ground truth provided by benchmark, it also proves the reconstruction capability of SS-3D-GAN framework in qualitative aspect.

3.2. Quantitative Comparative Experiments

We compare SS-3D-GAN with the state-of-the-art 3D reconstruction methods in various scenes benchmark. Here are the dataset we used in quantitative experiments.

***Tanks and Temples* dataset** This dataset [12] is designed for evaluating image-based and video-based 3D reconstruction algorithms. The benchmark includes both outdoor scenes and indoor environments. It also provides the ground truth of 3D surface model and its geometry. So it can be used to have a precise quantitative evaluation of 3D reconstruction accuracy.

As most of the state-of-the-art works in the shape prior based and generative-adversarial based method categories are target for single object reconstruction, and cannot handle the complicated 3D scene reconstruction. Moreover, their results are mainly represented in voxelized form without color. So for fair comparison, we just take the state-of-the-art works in SFM & MVS based and RGB-D camera based method categories which have similar 3D reconstruction capability and result representation form into comparative experiments. We choose VisualSFM [24], PMVS [6], MVE [5], Gipuma [7], COLMAP [19], OpenMVG [15] and SMVS [13] to compare with SS-3D-GAN. Beyond these, we also evaluate some combinations of methods which provides compatible interfaces.

Evaluation Process For comparative evaluation, the first step is aligned reconstructed 3D models to the ground truth.

Figure 5. Reconstructed ***Truck*** models in *Tanks and Temples* dataset (With different view angles and details). Column 1 shows ground truth. Column 2 shows the reconstructed 3D model with ***SS-3D-GAN***. Column 3 shows the reconstructed 3D model with ***COLMAP*** method.

Table 1. *Precision* (%) for *Tanks and Temple* Dataset

Algorithms	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train	Auditorium	Ballroom	Courtroom	Museum	Palace	Temple
COLMAP	56.02	34.35	40.34	41.07	**53.51**	39.94	38.17	**41.93**	31.57	24.25	38.79	45.12	27.85	**34.30**
MVE	37.65	18.74	11.15	27.86	3.68	25.55	12.01	20.73	6.93	9.65	21.39	25.99	12.55	14.74
MVE + SMVS	30.36	17.80	15.72	29.53	34.54	29.59	11.42	22.05	8.29	10.62	21.24	18.57	11.45	12.76
OpenMVG + MVE	38.88	22.44	18.27	31.98	31.17	31.48	23.32	26.11	14.21	19.73	25.94	28.33	10.79	17.94
OpenMVG + PMVS	61.26	**49.72**	37.79	47.92	47.10	52.88	41.18	37.20	26.79	29.10	42.70	47.82	23.78	28.58
OpenMVG + SMVS	31.87	21.36	16.69	31.63	34.71	33.83	32.61	26.32	16.45	14.72	22.92	20.05	12.81	15.07
SS-3D-GAN	**66.63**	48.99	**42.15**	**50.07**	53.35	52.89	**46.30**	41.21	**38.01**	29.08	**43.04**	**48.23**	30.59	33.45
VisualSfM + PMVS	59.13	38.67	35.25	48.92	53.20	**53.74**	46.02	33.69	37.57	**29.75**	41.31	40.36	**31.16**	18.69

Because the methods can estimate the reconstructed camera poses, so the alignment is achieved by registering them to ground-truth camera poses [12].

The second step is sampled the aligned 3D reconstructed model using the same voxel grid as the ground-truth point cloud. If multiple points fall into the same voxel, the mean of these points is retained as sampled result.

We use three metrics to evaluate the reconstruction quality. The *precision* metric quantifies the accuracy of reconstruction. Its value represents how closely the points in reconstructed model lie to the ground truth. We use $\boldsymbol{R}$ as the point set sampled from reconstructed model and $\boldsymbol{G}$ as the ground truth point set. For a point $\boldsymbol{r}$ in $\boldsymbol{R}$, its distance to the ground truth is defined as follows.

$$d_{\boldsymbol{r}\to\boldsymbol{G}} = \min_{\boldsymbol{g}\in\boldsymbol{G}} \|\boldsymbol{r}-\boldsymbol{g}\| \tag{11}$$

Then the *precision* metric of the reconstructed model for any distance threshold e is defined as follows.

$$P(e) = \frac{\sum_{\boldsymbol{r}\in\boldsymbol{R}} [d_{\boldsymbol{r}\to\boldsymbol{G}} < e]}{|\boldsymbol{R}|}, \tag{12}$$

where $[\cdot]$ is the Iverson bracket. The *recall* metric quantifies the completeness of reconstruction. Its value represents to what extent all the ground-truth points are covered. For a ground-truth point $\boldsymbol{g}$ in $\boldsymbol{G}$, its distance to the reconstruction is defined as follows.

$$d_{\boldsymbol{g}\to\boldsymbol{R}} = \min_{\boldsymbol{r}\in\boldsymbol{R}} \|\boldsymbol{g}-\boldsymbol{r}\| \tag{13}$$

The *recall* metric of the reconstructed model for any distance threshold e is defined as follows.

$$R(e) = \frac{\sum_{\boldsymbol{g}\in\boldsymbol{G}} [d_{\boldsymbol{g}\to\boldsymbol{R}} < e]}{|\boldsymbol{G}|} \tag{14}$$

Precision metric alone can be maximized by producing a very sparse point set of precisely localized landmarks. While *recall* metric alone can be maximized by densely covering the whole space with points. To avoid the situation, we combine *precision* and *recall* together in a summary metric *F-score*, which is defined as follows.

$$F(e) = \frac{2P(e)R(e)}{P(e)+R(e)} \tag{15}$$

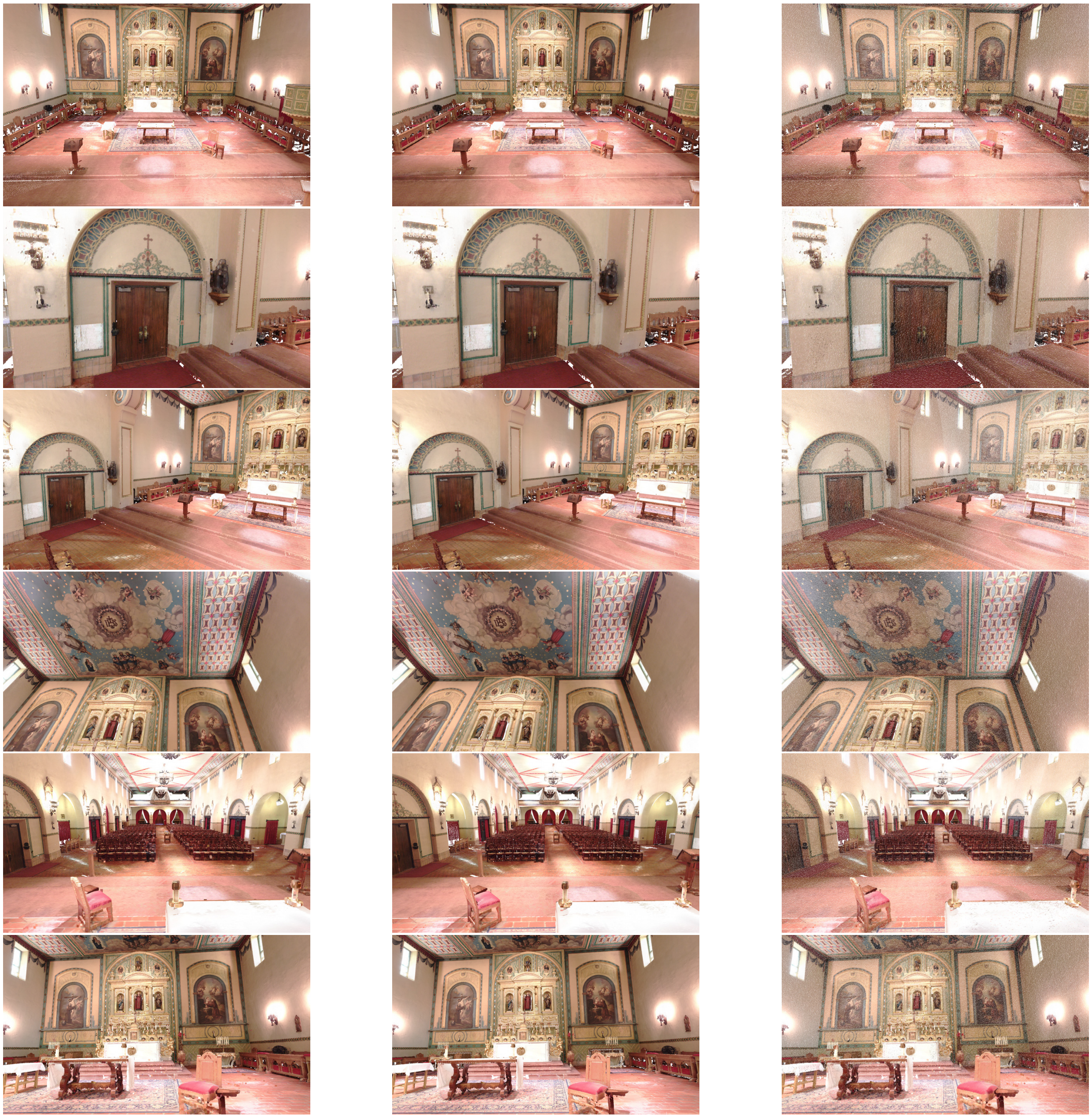


Figure 6. Reconstructed ***Church*** models in *Tanks and Temples* dataset. Column 1 shows ground truth. Column 2 shows the reconstructed 3D model with ***SS-3D-GAN***. Column 3 shows the reconstructed 3D model with ***COLMAP*** method.

Table 2. *Recall* (%) for *Tanks and Temple* Dataset

Algorithms	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train	Auditorium	Ballroom	Courtroom	Museum	Palace	Temple
COLMAP	45.82	16.46	18.79	49.34	59.69	57.01	66.61	42.15	10.73	26.29	31.40	38.44	13.36	23.56
MVE	68.52	32.75	14.74	68.59	8.14	75.40	3.83	49.32	2.92	18.26	40.21	52.05	**14.79**	19.51
MVE + SMVS	30.47	15.62	7.82	41.06	45.20	52.71	1.34	20.86	0.51	4.96	14.13	21.03	5.84	5.80
OpenMVG + MVE	**69.70**	37.91	24.01	**73.21**	**71.15**	77.41	**84.71**	**57.69**	15.22	**39.72**	43.42	**55.74**	2.20	31.41
OpenMVG + PMVS	30.85	10.77	7.73	28.73	30.04	24.19	25.88	22.58	2.48	7.63	13.93	20.99	3.94	8.18
OpenMVG + SMVS	31.99	18.66	13.65	43.16	45.51	54.02	39.91	24.02	4.41	9.54	17.46	24.11	6.82	10.35
SS-3D-GAN	69.31	**38.11**	**25.12**	72.89	69.97	**77.60**	83.55	55.72	**15.47**	37.66	**43.59**	54.83	14.74	**32.28**
VisualSfM + PMVS	28.02	7.77	6.73	27.83	34.36	25.07	28.86	8.25	2.49	6.63	10.20	13.30	4.15	1.13

Either aforementioned situation will drive *F-score* metric to 0. A high *F-score* can only be achieved by the reconstructed model which is both accurate and complete.

The *precision*, *recall* and *F-score* metrics for *Tanks & Temples* benchmark dataset are shown in Table 1 $\sim$ 3, respectively. According to the *F-score* metric obtained on each of the benchmark scenes in this dataset, SS-3D-GAN outperforms all other state-of-the-art 3D reconstruc-

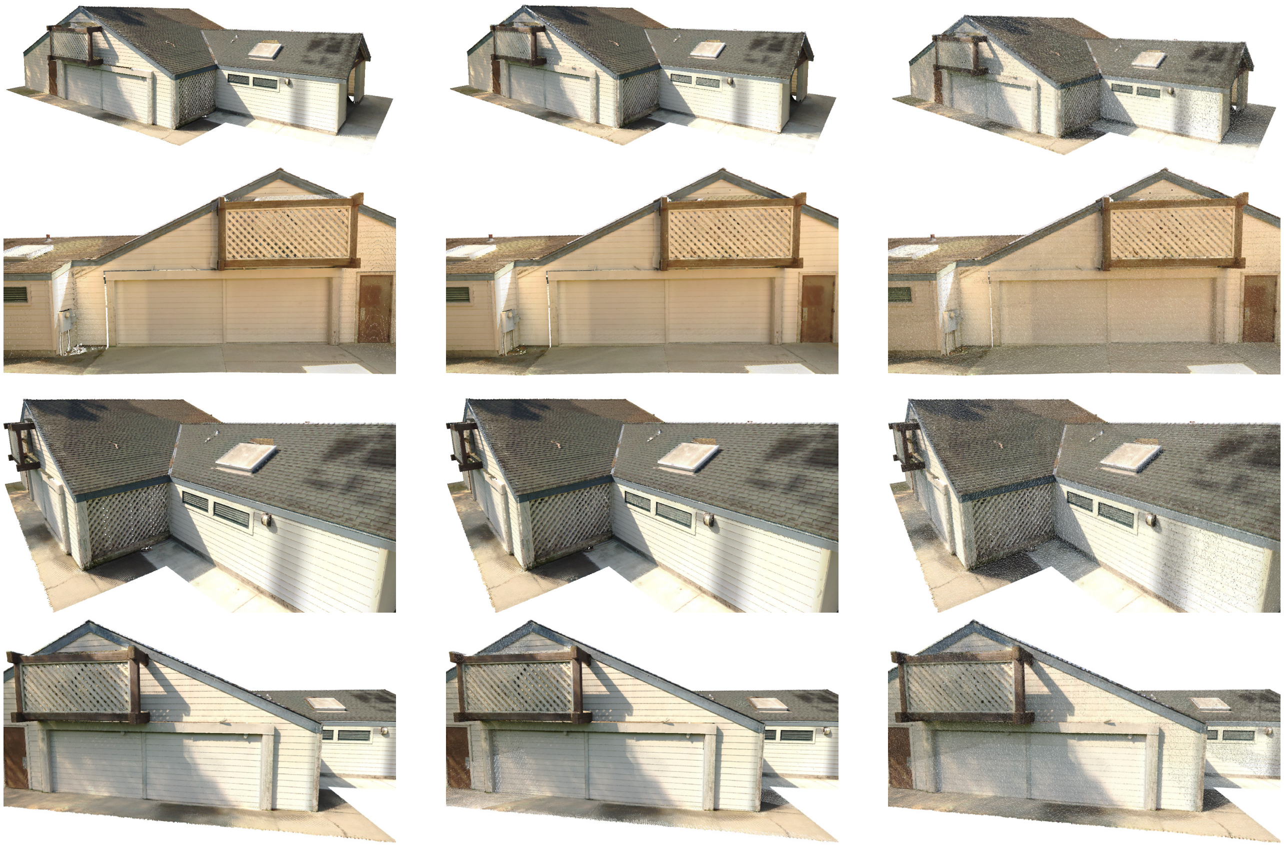


Figure 7. Reconstructed ***Barn*** models in *Tanks and Temples* dataset. Column 1 shows ground truth. Column 2 shows the reconstructed 3D model with ***SS-3D-GAN***. Column 3 shows the reconstructed 3D model with ***COLMAP*** method.

Table 3. *F-score* (%) for *Tanks and Temple* Dataset

Algorithms	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train	Auditorium	Ballroom	Courtroom	Museum	Palace	Temple
COLMAP	50.41	22.26	25.64	44.83	56.43	46.97	48.53	42.04	16.02	25.23	34.71	41.51	18.06	27.93
MVE	48.60	23.84	12.70	39.63	5.07	38.17	5.81	29.19	4.11	12.63	27.93	34.67	13.58	16.79
MVE + SMVS	30.41	16.64	10.44	34.35	39.16	37.90	2.40	21.44	0.96	6.76	16.97	19.72	7.73	7.98
OpenMVG + MVE	49.92	28.19	20.75	44.51	43.35	44.76	36.57	35.95	14.70	26.36	32.48	37.57	3.65	22.84
OpenMVG + PMVS	41.04	17.70	12.83	35.92	36.68	33.19	31.78	28.10	4.54	12.09	21.01	29.17	6.76	12.72
OpenMVG + SMVS	31.93	19.92	15.02	36.51	39.38	41.60	35.89	25.12	6.96	11.58	19.82	21.89	8.90	12.27
SS-3D-GAN	**67.94**	**42.87**	**31.48**	**59.36**	**60.54**	**62.91**	**59.58**	**47.38**	**21.99**	**32.82**	**43.31**	**51.32**	**19.89**	**32.85**
VisualSfM + PMVS	38.02	12.94	11.30	35.48	41.75	34.19	35.47	13.25	4.67	10.84	16.36	20.01	7.32	2.13

tion methods based on SFM & MVS and RGB-D camera.

In the *Tanks & Temples* dataset, for *precision* metric, the closest competitor is COLMAP and VisualSFM + P-MVS algorithms. For *recall* metric, the closest competitor is OpenMVG + MVE algorithm. But for the aggregate *F-score* metric, SS-3D-GAN can still achieve 1.1X∼1.5X relative improvement over the second highest *F-score* algorithms.

4. Conclusion and Future Works

We propose the novel 3D reconstruction framework to achieve high quality 3D reconstructed models of complicated scene. SS-3D-GAN transfers the traditional 3D reconstruction problem to the training and converge issue of GAN model. Due to its weakly semi-supervised principle, SS-3D-GAN has no reliance on 3D shape priors. So it is very suitable to complicated industrial and commercial reconstruction applications in real business. SS-3D-GAN also provides the quantitative indicators to measure the quality of 3D reconstructed model from human observation view angle. So it can also be used to mentor human's design work in the 3D modeling software, such as role modeling for video games, special visual effects for films, simulator design for autonomous driving, etc.

The SS-3D-GAN modle is trained from initial rough 3D reconstructed model [17]. So the quality of initial rough 3D model will affect the final result of SS-3D-GAN. In the Fig. 3(a), we provide the rough 3D model generated in the initialization stage. It gives a visualized quality of the rough model. There are large parts missing or with holes in the rough model. In the future, we will make quantitative analysis of the influence of initial rough model to SS-3D-GAN. Also lighting influence will be analysed in the future work.

References

[1] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 4

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016. 1

[3] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017. 1

[4] M. Dominguez, F. P. Such, S. Sah, and R. Ptucha. Towards 3d convolutional neural networks with meshes. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3929–3933. IEEE, 2017. 3

[5] S. Fuhrmann, F. Langguth, and M. Goesele. Mve-a multi-view reconstruction environment. In *GCH*, pages 11–18, 2014. 5

[6] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. 5

[7] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015. 5

[8] P. F. Gotardo, T. Simon, Y. Sheikh, and I. Matthews. Photogeometric scene flow for high-detail dynamic 3d reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 846–854, 2015. 1

[9] V. Guizilini and F. Ramos. Large-scale 3d scene reconstruction with hilbert maps. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3247–3254. IEEE, 2016. 1, 4

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[11] M. Johnson-Roberson, M. Bryson, A. Friedman, O. Pizarro, G. Troni, P. Ozog, and J. C. Henderson. High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology. *Journal of Field Robotics*, 34(4):625–643, 2017. 1

[12] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 5, 6

[13] F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. Shading-aware multi-view stereo. In *European Conference on Computer Vision*, pages 469–485. Springer, 2016. 5

[14] M. M. Loper and M. J. Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 2

[15] P. Moulon, P. Monasse, R. Perrot, and R. Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 5

[16] F. Nex and F. Remondino. Uav for 3d mapping applications: a review. *Applied geomatics*, 6(1):1–15, 2014. 1

[17] S. Pillai, S. Ramalingam, and J. J. Leonard. High-performance and tunable stereo reconstruction. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3188–3195. IEEE, 2016. 2, 8

[18] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 3

[19] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[20] M. Sra, S. Garrido-Jurado, C. Schmandt, and P. Maes. Procedurally generated virtual reality from 3d reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 191–200. ACM, 2016. 1

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3

[22] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 1

[23] C. Wu. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International conference on*, pages 127–134. IEEE, 2013. 1

[24] C. Wu et al. Visualsfm: A visual structure from motion system. 2011. 5

[25] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 1

[26] C. Yu. Steganography of digital watermark based on artificial neural networks in image communication and intellectual property protection. *Neural Processing Letters*, 44(2):307–316, 2016. 3

[27] Q.-Y. Zhou, S. Miller, and V. Koltun. Elastic fragments for dense scene reconstruction. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 473–480. IEEE, 2013. 1