Semi-supervised Three-dimensional Reconstruction Framework with GAN

Chong Yu
NVIDIA Corporation
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 dataset and semi-supervised learning. So the focus of this paper is to achieve high quality 3D reconstruction performance by adopting 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 Tanks & Temples reconstruction benchmark dataset.

1 Introduction and Related Work

In computer graphics and computer vision areas, 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 [Dai *et al.*, 2017], geomatics [Nex and Remondino, 2014], archaeology [Johnson-Roberson *et al.*, 2017], game and virtual reality [Sra *et al.*, 2016] areas, etc. Researchers have made significant progress on 3D reconstruction in the past decades. The state-of-the-art 3D reconstruction methods can be divided into four categories: Structure from motion (SFM) based, RGB-D camera based, Shape prior based and Generative-Adversarial based methods.

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, SS-3D-GAN can iteratively improve the reconstruction quality in semi-supervised manner. The main contribution of this paper includes:

- 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.

1.1 SFM and MVS Based Method

In the traditional SFM and MVS method, two-view reconstructed results are firstly estimated upon the feature matching between two images. Then, 3D models are reconstructed by initializing from successful two-view reconstructed results, iteratively adding new matched images, triangulating feature matches, and bundle-adjusting the structure and motion. The time complexity of traditional SFM and MVS method is often known as $O(n^4)$ with respect to the number of cameras. The representative work namely VisualSFM, further improves the performance. The time complexity is reduced to $O(n)$ on many major time-consuming steps including bundle adjustment. However, this method has obvious restrictions. They come from the key technical assumption that features are able to be matched across multi-views. If the viewpoints are separated by a large baseline, feature matching will be extremely problematic because of local appearance changes or self-occlusions. Another key limitation is that if the surface of the reconstructed objects lacks of texture or has specular reflections, the feature matching will also be in vain.

1.2 RGB-D Camera Based Method

The most famous work is KinectFusion [Newcombe *et al.*, 2011]. With the depth information provided, the method

can continuously tracks the six degrees-of-freedom pose of a RGB-D camera. The tracking accuracy relies on the feature matching between RGB frames for pose tracking. Reconstructed model is obtained by iteratively integrating depth and tracking information into a global dense volumetric model. Whelan [Whelan *et al.*, 2016] further improves the KinectFusion in tracking accuracy, robustness and reconstruction quality. The improvement is achieved by adopting dense frame-to-model camera tracking and windowed surfel-based fusion coupled with frequent model refinement through non-rigid surface deformations. The main restriction of this method type is that there are obvious holes, distortions and obscure parts exist in the 3D reconstructed model due to self-occlusion, light reflection, fusion error of depth sensor data, etc.

1.3 Shape Prior Based Method

Representative work is 3D-R2N2 [Choy *et al.*, 2016]. It uses deep convolutional neural network to learn the mapping relation from observed 2D images to corresponding underlying 3D shapes of target objects from a large collection of training data. Taking the advantages of LSTM network, it takes in one or more images of an object instance from arbitrary viewpoints and outputs the reconstructed result in the 3D voxelized form. The main advantage is this method can be applied to single-view and multi-view 3D reconstruction by selectively updating hidden representations with the control of input gates and forget gates. It can get the 3D reconstructed model even the information from different viewpoints are partly conflicted. However, the restrictions are also obvious. The success of 3D-R2N2 depends on training dataset of 3D CAD models and the corresponding 2D observations. And it can only reconstruct some categories of isolated objects into 3D voxelized form.

1.4 Generative-Adversarial Based Method

Representative work is 3D-GAN [Wu *et al.*, 2016]. 3D-GAN introduces generative-adversarial loss as the criterion to classify whether an object is real or reconstructed. Because 3D objects are highly structured, generative-adversarial criterion has better performance on capturing the structural difference of 3D objects than traditional voxel-wise independent heuristic criterion. It also solves the criterion-dependent overfitting problem. Most obvious advantage comes from GAN principle. With GAN framework, it learns the mapping relation from low-dimensional probabilistic space to 3D objects space. So the reconstruction process doesn't depend on training dataset of 3D CAD models and corresponding 2D reference observations. But limitation is also caused by GAN principle. Currently, even many works have improved the training process of GAN, but it is still hard to converge even in 2D space. Because of the complexity of 3D space, it can only reconstruct simply isolated objects into 3D voxelized form, which is limited in size, color, texture style, and quality.

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.

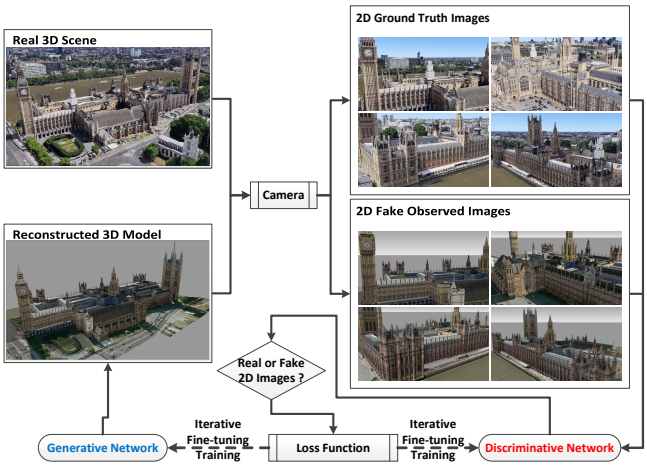

Figure 1: Principle and workflow chart of SS-3D-GAN

Firstly, he observes in the real 3D scene. Then he observes in the reconstructed 3D scene model at exactly the same positions and viewpoints. 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 [Pillai *et al.*, 2016]. 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" [Loper and Black, 2014]. 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 discriminator's 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:

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

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 [Yu, 2016]. Peak Signal to Noise Ratio (PSNR) indicator is applied to assess the effect difference from the gray-level fidelity aspect. Structural Similarity (SSIM) [Wang *et al.*, 2004] 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.

SSIM indicator value of two images is in the range of 0 to 1. NC indicator's value is in the range of -1 to 1. If the value of SSIM or NC is closer to 1, it means there is less difference between image $\boldsymbol{x}$ and $\boldsymbol{y}$. For PSNR , 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.

$$E\_Sigm\left(PSNR\left(\boldsymbol{x},\boldsymbol{y}\right)\right) = \frac{1}{1+e^{-0.1(PSNR(\boldsymbol{x},\boldsymbol{y})-45)}}, \quad (2)$$

So the reconstruction loss is written as follows:

$$L_{Recons} = \sum_{j=1}^{N} \Big\{ \alpha \cdot \left[1 - E\_Sigm\left(PSNR_{G_jF_j}\right)\right] + \beta \cdot \left(1 - SSIM_{G_jF_j}\right) + \gamma \cdot \left(1 - NC_{G_jF_j}\right) \Big\} \quad (3)$$

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 [Qi *et al.*, 2016]. 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.

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 [Dominguez *et al.*, 2017].

Then we can apply Graph CNN [Dominguez *et al.*, 2017] 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

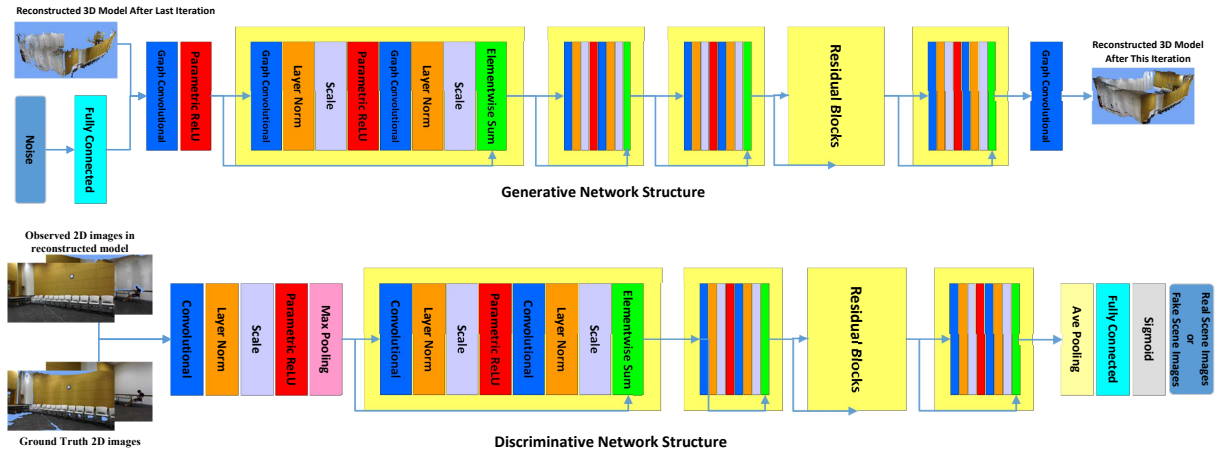

Figure 2: Details of generative network structure and discriminative network structure in SS-3D-GAN

data. The input data for a graph convolutional layer with C filters includes:

$$\boldsymbol{V}_{in} \in \boldsymbol{R}^{N\times F}, \boldsymbol{A} \in \boldsymbol{R}^{N\times N\times L}, \boldsymbol{H} \in \boldsymbol{R}^{L\times F\times C}, \boldsymbol{b} \in \boldsymbol{R}^{C}, \quad (4)$$

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.

$$\boldsymbol{V}_{out} = (\boldsymbol{A} \times \boldsymbol{V}_{in}^{T})_{(2)} \boldsymbol{H}_{(3)}^{T} + \boldsymbol{b}, \boldsymbol{V}_{out} \in \boldsymbol{R}^{N\times C} \quad (5)$$

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 [He *et al.*, 2016] 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 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 [Guizilini and Ramos, 2016], 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.

$$\min_{G}\max_{D} \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathbf{P}_r}[D(\boldsymbol{x})] - \mathop{\mathbf{E}}_{\tilde{\boldsymbol{x}}\sim\mathbf{P}_g}[D(\tilde{\boldsymbol{x}})], \quad (6)$$

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.

$$\begin{aligned} L_{SS-3D-GAN} = & \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathbf{P}_r}[D(\boldsymbol{x})] - \mathop{\mathbf{E}}_{\tilde{\boldsymbol{x}}\sim\mathbf{P}_g}[D(\tilde{\boldsymbol{x}})] - \\ & \theta \cdot \mathop{\mathbf{E}}_{\hat{\boldsymbol{x}}\sim\mathbf{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1\right)^2\right], \end{aligned} \quad (7)$$

where θ is the parameter to adjust the percentage of gradient penalty in the cross entropy loss. $\mathbf{P}_{\hat{x}}$ 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

With the initial rough 3D reconstructed model generated by spatial mapping, 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 models of *Tanks and Temples* dataset are shown in Fig. 3. Compared with ground truth provided by benchmark, it proves the reconstruction capability of SS-3D-GAN framework in qualitative aspect.

Figure 3: Reconstructed results in *Tanks and Temples* dataset. Column 1 shows ground truth. Column 2 shows the reconstructed 3D model with COLMAP method. Column 3 shows the reconstructed 3D model with SS-3D-GAN. Row 4 shows the details of truck examples.

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

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

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 [Knapitsch *et al.*, 2017] 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 [Wu and others, 2011], PMVS [Furukawa and Ponce, 2010], MVE [Fuhrmann *et al.*, 2014], Gipuma [Galliani *et al.*, 2015], COLMAP [Schönberger and Frahm, 2016], OpenMVG [Moulon *et al.*, 2016] and S-MVS [Langguth *et al.*, 2016] to compare with SS-3D-GAN. Beyond these, we also evaluate some combinations of methods which provides compatible interfaces.

For comparative evaluation, the first step is aligned reconstructed 3D models to the ground truth. Because the methods can estimate the reconstructed camera poses, so the alignment is achieved by registering them to ground-truth camera poses [Knapitsch *et al.*, 2017].

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

Table 2: *Recall* (%) 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

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

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

$$P(e) = \frac{\sum_{\boldsymbol{r}\in\boldsymbol{R}} [d_{\boldsymbol{r}\to\boldsymbol{G}} < e]}{|\boldsymbol{R}|}, \tag{8}$$

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

$$R(e) = \frac{\sum_{\boldsymbol{g}\in\boldsymbol{G}} [d_{\boldsymbol{g}\to\boldsymbol{R}} < e]}{|\boldsymbol{G}|} \tag{9}$$

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.

$$F(e) = \frac{2P(e)R(e)}{P(e) + R(e)} \tag{10}$$

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

In the *Tanks & Temples* dataset, for *precision* metric, the closest competitor is COLMAP and VisualSFM + PMVS 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$\sim$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.

In this paper, we use *Blender* as the tool to operate the 3D reconstructed model. However the APIs provided by *Blender* is not user-friendly. It leads to the extra time consumptions for training SS-3D-GAN. In the future, we will solve this issue to improve the reconstruction efficiency.

The SS-3D-GAN modle is trained from initial rough 3D reconstructed model [Pillai *et al.*, 2016]. So the quality of initial rough 3D model will affect the final result of SS-3D-GAN. 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

[Choy *et al.*, 2016] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio 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.

[Dai *et al.*, 2017] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017.

[Dominguez *et al.*, 2017] Miguel Dominguez, Felipe Petroski Such, Shagan Sah, and Raymond Ptucha. Towards 3d convolutional neural networks with meshes. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3929–3933. IEEE, 2017.

[Fuhrmann *et al.*, 2014] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. Mve-a multi-view reconstruction environment. In *GCH*, pages 11–18, 2014.

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

[Galliani *et al.*, 2015] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[Guizilini and Ramos, 2016] Vitor Guizilini and Fabio 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.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Johnson-Roberson *et al.*, 2017] Matthew Johnson-Roberson, Mitch Bryson, Ariell Friedman, Oscar Pizarro, Giancarlo Troni, Paul Ozog, and Jon C Henderson. High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology. *Journal of Field Robotics*, 34(4):625–643, 2017.

[Knapitsch *et al.*, 2017] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017.

[Langguth *et al.*, 2016] Fabian Langguth, Kalyan Sunkavalli, Sunil Hadap, and Michael Goesele. Shading-aware multi-view stereo. In *European Conference on Computer Vision*, pages 469–485. Springer, 2016.

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

[Moulon *et al.*, 2016] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[Newcombe *et al.*, 2011] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

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

[Pillai *et al.*, 2016] Sudeep Pillai, Srikumar Ramalingam, and John J Leonard. High-performance and tunable stereo reconstruction. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3188–3195. IEEE, 2016.

[Qi *et al.*, 2016] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas 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.

[Schönberger and Frahm, 2016] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Sra *et al.*, 2016] Misha Sra, Sergio Garrido-Jurado, Chris Schmandt, and Pattie 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.

[Wang *et al.*, 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[Whelan *et al.*, 2016] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[Wu and others, 2011] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011.

[Wu *et al.*, 2016] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh 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.

[Yu, 2016] Chong 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.