SMVNet: Deep Learning Architectures for Accurate and Robust Multi-View Stereopsis

1st Shizeng Yao  
Dept. of EECS  
University of Missouri - Columbia  
Columbia, MO, USA  
shizeng.yao@mail.missouri.edu

2nd Yangyang Wang  
Dept. of EECS  
University of Missouri - Columbia  
Columbia, MO, USA  
yw6p8@mail.missouri.edu

3rd Hadi AliAkbarpour  
Dept. of EECS  
University of Missouri - Columbia  
Columbia, MO, USA

4th Guna Seetharaman  
US Naval Research Laboratory  
Washington, D.C., USA

5th Raghuveer Rao  
US Army Research Laboratory  
USA

6th Kannappan Palaniappan  
Dept. of EECS  
University of Missouri - Columbia  
Columbia, MO, USA

Abstract—We describe Spatial Voxel-Net (SVNet) and Multi-View Voxel-Net (MVNet), a cascade of two novel deep learning architectures for calibrated multi-view stereopsis that reconstructs complicated outdoor 3D models accurately. Both networks use a sequence of RGB images based on ordered camera poses in a coarse-to-fine fashion. SVNet extracts summarized features and analyzes the spatial relationship among a block of 3D voxels using 3D convolutions, then predicts block-level occupancy information. MVNet then receives the occupancy information together with RGB images to predict the final voxel-level occupancy information. SMVNet is an end-to-end trainable network, which can reconstruct complex outdoor 3D models and be applied to large-scale datasets in a parallel fashion without the need of estimating or fusing multiple depth maps, typical of other approaches. We evaluated SMVNet on the complex outdoor Tanks and Temples dataset, in which outperformed two well-known state-of-the-art MVS algorithms.

Index Terms—Computer Vision, MVS, Deep Learning, City-Scale

I. INTRODUCTION

Multi-view stereopsis (MVS) is a core problem in computer vision, which takes a set of scene views together with known camera poses, then produces a geometric representation of the underlying 3D model. This representation can be a set of disparity or depth maps, a 3D point cloud, signed distance fields, etc. Many methods [1]–[4] have shown the ability to reconstruct small objects or large scenes using classical computer vision algorithms, while many modern approaches have successfully reconstructed small objects or small indoor scenes using learning-based techniques [5]–[8].

However, predicting individual voxel occupancy information is non-trivial. In this paper, we present a cascade of two novel deep learning architectures: Spatial Voxel-Net and Multi-View Voxel-Net (SMVNet) - an end-to-end trainable deep-learning system which is able to accurately and robustly reconstruct large outdoor 3D models given only RGB images and their camera poses. Compared to other deep learning based MVS approaches, one of our key novelties is the development of SVNet, which uses the coarse spatial relationship to hierarchically predict the block occupancy information, which is further refined in MVNet. By using SVNet, we could reduce both the number of operations and the computation time at inference while keeping the same level of accuracy. Later, MVNet generates a multi-view cube by projecting voxels into image views, then utilizes the cube together with the probability score from SVNet to get the 3D occupancy information without the need of estimating or fusing multiple depth maps. The proposed approach utilizes a coarse-to-fine parallel fashion and can be scaled up to sub-meter resolution.

We train and evaluate the proposed SMVNet on the large-scale complex outdoor Tanks and Temples dataset [9], and compared with furu [4] and COLMAP [10], [11] for quantitative evaluation. Examples showing the training image and reconstructed result can be found in Figure 1.

The rest of this paper is organized as follows: Section II presents the related works. Section III presents our system overview and describes the details about our approach. Experiments, quantitative evaluations and discussions are presented in Section IV and Section IV-B. Section V concludes the paper with some future work.

II. RELATED WORK

Extracting 3D information and reconstructing 3D models from images are one of the classical problems in computer vision.
vision. Early works mainly focus on classical computer vision techniques. We refer the readers to [12] for an overview of classical MVS algorithms. Early algorithms could be generally divided into the following categories: Algorithms based on deformable polygonal meshes [2], [13] demand a good starting point to initialize the corresponding optimization process. Algorithms using multiple depth maps [3], [14], [15] require merging individual depth maps into one single 3D model. Algorithms with patch-based [4], [16] methods are simple, but often suffer from the lack of texture information. In addition, when they are applied on large city-scale datasets, a post-processing to merge separately generated point clouds is often needed [17]. Voxel-based approaches [18]–[20] whose accuracy is limited to the pre-defined grid resolution. Although our method is also applied on a 3D voxelized volume, the accuracy of our method is not limited to the grid resolution since we could continue dividing our grid into higher resolution and apply our system on the subspace.

Recently, machine learning based techniques, especially CNN based deep learning methods, have been widely used to solve MVS related problems including 3D semantic segmentation [21], [22], structure-from-motion (SFM) [23], synthetic view generating [24], image-based localization [25], [26], and etc..

Learning-based 3D reconstruction algorithms have also been widely proposed recently, for example: [5], [6] predict the shape of objects from a single image in the form of direct depth map regression.

Learning-based depth map fusion algorithms OctNet [7], MVSNet [27], DeepMVS [28] first extract image features and build multiple depth maps, then fuse the depth maps to obtain a final 3D reconstruction. To improve accuracy, [27] extracts deep visual features instead of common features to create a 3D cost volume for learning-based depth maps estimation. [28] also creates plane-sweep volumes to feed into a convolutional neural network for better disparity maps estimation. Other similar algorithms reconstruct 3D models by learning image patch similarity function [29], [30]. However these algorithms require a fusion step to merge multiple depth maps into one single model.

SurfaceNet [31] describes a network, which receives two cubes containing information from two images from two different viewing angles, and then estimates the surface that lies in this cube. This method achieves color information in each cube by projecting the voxels within this cube onto images. Then the network estimates a surface result for each pair of cubes, and for an arbitrary number of images, they compute a final output by averaging among all results. This process could be highly affected by outliers, for example, occlusion, complex illuminations, and motion blur. Although [31] develops a second shallow network to reject occluded cubes, the occlusion situation may vary among different datasets and the final result may still be affected. In our proposed network, instead of using only two images with color information, we applied a Mixture of Gaussian model to extract color information through the whole data sequence, which can discard outliers automatically.

Secondly, besides the color information, we also used 2D feature points as an extra channel to improve accuracy and stability. For an arbitrary number of images, since we used the whole sequence for color and 2D feature points information extraction, no averaging or other fusion processes are needed in our network.

In our work, we apply our algorithm on a discretized voxelized volume space with a novel set of color and likelihood features to analyze the spatial relationship among voxels, and then hierarchically predict the occupancy information. In addition, in contrast with other existing learning-based MVS methods which focus on reconstructing small objects or indoor scenes, our proposed method mainly focuses on large outdoor scene reconstruction.

### III. Proposed Approach

#### A. System Pipeline

We propose a system consisting of two networks which receives an empty voxelized volume space together with image views as inputs, then predicts fine resolution voxel occupancy information, which is demonstrated in Figure 2.

![SMVNet pipeline](image)

**Fig. 2. SMVNet pipeline:** Inputs are ordered RGB image views together with an empty 3D voxelized volume. **SVNet** predicts coarse occupancy information and **MVNet** refines high resolution occupancy information.

#### B. Spatial Voxel-Net (SVNet)

In large-scale 3D models, due to the fact that foreground surfaces are just thin layers, number of foreground samples and number of background samples are usually extremely unbalanced. As a solution to this problem, our first network SVNet predicts a probability score for blocks of voxels instead of a single voxel, which reduces both the number of operations and the computation time at inference while keeping the same level of accuracy. Another reason of applying SVNet on a block of 3D voxels is because it is then able to exploit spatial coherence information, which allows preserving the spatial information among the groups of voxels, in contrast to many modern learning-based 3D algorithms that are treating the volume as a set of unordered 3D points.

**SVNet** (demonstrated in Figure 3 top row) is applied on a block with size $u \times u \times u$ of 3D voxels. Each 3D voxel contains eight channels of feature including:

- Three channels of color (RGB) obtained through a 3D Mixture of Gaussian model (Sec III-B1)
- Three standard deviations associated with the three color channels described above (Sec III-B1)
- The number of images associated with the dominant cluster in MoG model (Sec III-B1)
Fig. 3. Proposed cascade of two networks for large-scale 3D model reconstruction. SVNet (top row) consumes a block with size $u \times u \times u$ of 3D voxels, where each voxel contains summarized chromatic features and surface likelihood map, and outputs a probability score for current block. MVNet (bottom row) receives a cube of $v \times v \times v$ voxels, where each voxel contains a probability score from SVNet and its full-sequence chromatic information, and outputs a probability score for only the center voxel of this cube.

- One feature channel represents the likelihood of current 3D voxel belonging to a 3D surface (Sec III-B2)

SVNet includes six 3D convolutional layers, one max-pooling layer, and two fully connected layers. The parameters for each layer can be found in Figure 3 top row. SVNet outputs a probability score for the current block, where a higher score represents a higher probability that this block contains certain foreground structures. By preserving the blocks with higher probability scores, a coarse resolution model can be exported as an intermediate result.

1) Voxel Chromatic Features Extraction: The first 7 channels of features for each voxel are obtained by the following chromatic feature extraction process. We firstly create this $u \times u \times u$ 3D voxelized block. Using camera poses we then project each 3D voxel in this block to the set of 2D image views and gather corresponding color information. The process results in a queue of RGB color values for each voxel, which are further divided and summarized into $K$ color clusters through a 3D Mixture of Gaussian model derived from the work of F. Nielsen [32]. More concretely, after we obtain the queue of RGB color values, we create $K$ clusters for each 3D voxel ($K = 4$ in our experiments). During the procedure, all clusters are updated respectively: each time a new color is dequeued from the color queue, its distances $d_{clusterK}$ from the current $K$ color clusters are computed as:

$$d_{clusterK} = \sum_{c \in \{R, G, B\}} |color_{clusterK}^c - color_{new}^c| / SD_{clusterK}$$  \hspace{1cm} (1)$$

where $c$ is the three color channels and $K$ represents a certain cluster label. After computing the distances, we attempt to update the cluster with the minimum distance if the minimum distance is lower than a pre-defined threshold. Otherwise, the cluster with the least number of images is replaced by the new dequeued color. After applying this model through all the colors in the color queue, the cluster with the most number of images is selected as the dominant cluster, whose information is assigned to current 3D voxel as its 7 channels of chromatic features. The pseudo code describing this process is demonstrated in Algorithm 1.

Algorithm 1 3D Mixture of Gaussian Model

1: procedure
2: Input: Voxels in the given 3D Euclidean space $V$
3: Output: Color features associated with the given voxels
4: for Each voxel $v$ in $V$ do
5: Project $v$ into all image views based on camera parameters
6: Create color queue $q$
7: Create $K$ clusters
8: for Each cluster in $K$ do
9: $color_{candidate} = 0$
10: $SD = \infty$
11: Number of Images = 0
12: end for
13: for Each color in $q$ do
14: Compute distance $d_{clusterK}$ for all $K$ clusters using equation 1
15: Compute minimum distance $d_{min}$ among $K$ clusters
16: if $d_{min} < \text{Threshold}$ then
17: Update cluster with $d_{min}$
18: else
19: Reset cluster with the least images by new color
20: end if
21: end for
22: dominant cluster = the cluster with the most number of images
23: Assign information from the dominant cluster to voxel $v$
24: end for
25: end procedure

During this chromatic feature extraction process, noisy information such as occlusion, complex illumination, and fast motion blur is discarded automatically and hence will not introduce outliers into network. Figure 4 demonstrates that this chromatic information could be used to distinguish occupied blocks and unoccupied blocks.

2) Voxel Surface Likelihood Computation: The last feature channel of each voxel is obtained by a surface likelihood
map. We firstly detect 2D Structure Tensor feature points [33], which is a matrix representation of partial derivative information, in all image views. Then from each 2D feature point on each image view, a ray is cast from 2D image view into the 3D block based on the camera pose. The likelihood value of any voxel lying on this ray is incremented by 1 (Figure 5a). A surface likelihood map is then built after applying this procedure through all image views in the image sequence. This likelihood map is used as the last feature channel for each voxel for that view. For arbitrary number of images, we down-sample/up-sample the color queue length to $n$ (n is set to 54 in our experiments).

C. Multi-View Voxel-Net (MVNet)

MVNet receives a cube of $v \times v \times v$ voxels, centered at the current testing voxel. Each voxel in this cube contains a color queue of $n$ RGB color values and 1 probability score computed from SVNet. The $n$ RGB color values are obtained by projecting the voxels onto all image views, then ordered based on camera positions (see Figure 6a). Figures 6b and 6c show two different color appearances of the same cube from the same viewing angle but from two different cameras. When a special situation occurs, such as any voxel is outside of an image view, we assign color values ”-1,-1,-1” to that voxel for that view. For arbitrary number of images, we down-sample/up-sample the color queue length to $n$ (n is set to 54 in our experiments).

MVNet includes six 3D convolutional layers, two max-pooling layers, and three fully connected layers. The parameters for each layer can be found in Figure 3 bottom row. MVNet outputs a probability score for the center voxel in this cube, where a higher score represents a higher probability that this voxel belongs to a surface. By preserving the voxels with higher probability scores, a fine resolution 3D model can be generated.

D. Coloring & Post-Processing

After applying the cascade of networks, a fine-resolution large-scale 3D model is reconstructed by preserving the voxels with higher probability scores. We then apply a simple post-processing to filter out noise and assign a real-world color to each remaining voxel by applying our 3D Mixture of Gaussian model again with two more constraints:

- For each 3D voxel in the final model, we check its visibility in all image views and update clusters using only the images in which the voxel is not occluded.
- After updating through the whole image sequence, a voxel will be removed if the number of images in the dominant cluster is lower than a pre-defined threshold value.

A large-scale colored point cloud model is then exported after this procedure as the final result.

IV. EXPERIMENTS AND EVALUATIONS

A. Training Strategy

We used Tanks and Temples multi-view stereo dataset [9] for training and evaluation, which is a large scale complex outdoor MVS benchmark. These benchmark sequences were acquired outside the lab, in realistic conditions. Ground-truth data was captured using an industrial laser scanner. We used training datasets from the Tanks and Temples for training step (Figure 1 a), and Intermediate dataset for quantitative evaluation.

In the training step of SVNet, $u$ is set to 30 (voxel distance is set to 1 cm), and we use a sliding window with stride size $u/6$ to acquire training samples. In the training step of MVNet, $v$ is set to 9 (voxel distance is set to 1 cm). Both networks use sigmoid function as the activation function, stochastic gradient descent as the optimization algorithm, and binary cross-entropy function as the loss function.

All the feature extraction procedures and final coloring module are implemented in C++ with GPU acceleration while both networks are implemented in Python using Tensorflow-GPU and Keras libraries [35], [36]. The training process of SVNet takes 3808.1 seconds on average for each epoch and the
training process of MVNet takes 3554.9 seconds on average for each epoch using a Nvidia GeForce GTX 1080 Ti GPU. We trained 100 epochs for each of the two networks.

B. Quantitative Evaluations

The point cloud results of the intermediate set from Tanks and Temples dataset can be found in Figure 7. To quantitatively evaluate our method, we compared our results with Bundler [34] + PMVS [4] and COLMAP baseline [10], [11] using the evaluation methodology proposed in Tanks and Temples MVS benchmark [9]. The precision score \( P(d) \), recall score \( R(d) \), and F-score \( F(d) \) are calculated using Equation 2.

\[
P(d) = \frac{100}{|R|} \sum_{e \in R} [e_{r \rightarrow G} < d] \\
R(d) = \frac{100}{|G|} \sum_{g \in G} [e_{g \rightarrow R} < d] \\
F(d) = \frac{2P(d)R(d)}{P(d) + R(d)}
\]

The F-scores of each image sequence dataset are shown in Table I. From the table we could notice that our proposed method outperforms the other two methods in half of the sequences, and achieved the highest mean F-score among the three approaches. The reason that our method doesn’t perform well in the dataset such as Playground and Lighthouse is because the provided imagery very often focuses on a small portion of the whole scene, and since our method uses the Mixture of Gaussian model to extract dominant features, the network was not able to classify the correct occupancy information.

V. CONCLUSIONS

SMVNet is a cascade of two novel Convolutional Neural Networks, which is an end-to-end trainable deep learning architecture that performs multi-view stereopsis at a large-scale. One of the key novelties is the use of coarse spatial relationships among a group of 3D voxels and hierarchical predicting of the occupancy information in a coarse-to-fine approach. SMVNet can be scaled in a parallel fashion to large outdoor scenes for high resolution. We validated the approach