MVSCRF: Learning Multi-view Stereo with Conditional Random Fields

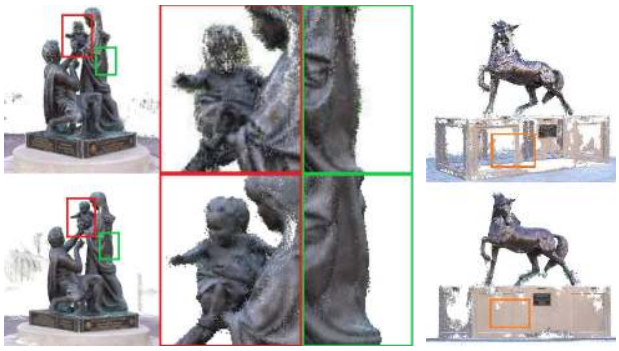
Youze Xue, Jiansheng Chen*, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, Jiayu Bao
Department of Electronic Engineering, Tsinghua University
xueyz19@mails.tsinghua.edu.cn, jschenthu@mail.tsinghua.edu.cn
{wwt16, huang-yq17, yuc18, ltp16, bjy19}@mails.tsinghua.edu.cn

Abstract

We present a deep-learning architecture for multi-view stereo with conditional random fields (MVSCRF). Given an arbitrary number of input images, we first use a U-shape neural network to extract deep features incorporating both global and local information, and then build a 3D cost volume for the reference camera. Unlike previous learning-based methods, we explicitly constraint the smoothness of depth maps by using conditional random fields (CRFs) after the stage of cost volume regularization. The CRFs module is implemented as recurrent neural networks so that the whole pipeline can be trained end-to-end. Our results show that the proposed pipeline outperforms previous state-of-the-arts on large-scale DTU dataset. We also achieve comparable results with state-of-the-art learning-based methods on outdoor Tanks and Temples dataset without fine-tuning, which demonstrates our method's generalization ability.

1. Introduction

In a multi-view stereo (MVS) system, images of the same scene or object taken from different views are processed to reconstruct the 3D model. Traditional MVS methods formulate the task as an optimization problem by utilizing the projection relationship among multiple views [21][22]. Recent success of deep learning has inspired researchers to exploit learning-based MVS methods. Some research exploit the volumetric representation of 3D models, and regress each voxel's occupancy with deep convolutional neural networks (CNNs) [13][14][6]. However, restricted by its huge memory consumption, volumetric representation cannot be scaled up and thus leads to very low resolution of the reconstructed space. Another approach to reconstruct a 3D scene is to first estimate depth map per view, and then fuse depth maps to form the point cloud. Recent work [29][30][10][5] based on depth map estimation have achieved excellent results on public MVS benchmarks such as *DTU* [12] and *Tanks and Temples* [15], demonstrating the effectiveness of using depth maps as intermediate representations in MVS.



(a) Family (b) Horse

Figure 1. Qualitative comparison between MVSNet (top) and the proposed MVSCRF (bottom) on two selected scenes of the *Tanks and Temples* dataset. The visual results of MVSNet are directly cited from the original paper [29].

A commonly used pipeline in binocular stereo was extended to deep learning based MVS in [29], in which Yao et al. proposed an end-to-end architecture MVSNet. The MVSNet first extracts deep features of input images, then builds a 3D cost volume for the reference camera using differentiable homography warping, and finally regularizes the cost volume to regress the depth map. The MVSNet achieves comparable or even better performance compared to optimization based traditional methods on public benchmarks. Recently, Yao et al. further proposed the R-MVSNet [30] which essentially reduces the inference memory requirement by re-designing the cost volume regularization of MVSNet as recurrent neural networks. As such, depth sampling rate can be significantly increased, leading to preciser 3D predictions as expected.

Another way to improve the performance of depth estimation, which may be orthogonal to that used in R-MVSNet, is to further exploit the intrinsic characteristics of depth maps of natural scenes. Inspired by semantic seg-

*Corresponding author

mentation [4], which is also a pixel-wise prediction task, we argue that the combination of local and global features may be effective in depth map estimation. On the one hand, rich local information helps to localize a pixel precisely. On the other hand, pixels in texture-less or reflective regions rely more on global information for their reconstruction. In both MVSNet and R-MVSNet, a sequential stack of convolutional layers is implemented to extract deep features, which cannot effectively incorporate semantic cues from different scales. To solve this problem, Huang et al. proposed to combine semantic features extracted by a pretrained VGG-19 [23] network with features extracted by a UNet-like structure in DeepMVS [10]. However, since DeepMVS is based on sequential processing of image patches, it cannot exploit the information of the whole image. We therefore propose to use a six-scale U-shape structure to extract deep features from the original input images. The encoder-decoder structure provides large receptive fields to extract global information, and the skip connections from shallow layers to deep layers help to merge rich local information with global semantic cues.

More importantly, the depth values of neighboring pixels are usually highly correlated. In the inner area of an object, depth values tend to be continuous. While on the borders of an instance, depth values usually vary drastically. In MVSNet, a 3D UNet is implemented to aggregate neighboring information. However, no explicit constraint is imposed on the smoothness property of depth maps. DeepMVS uses Dense-CRF [16] in post-processing to refine the depth map with smoothness constraints explicitly. We argue that an more effective way is to incorporate the smoothness constraint in an end to end trainable style in MVS. We therefore propose to use the output of the cost volume regularization as the input of multi-scale conditional random fields (MSCRFs), which are implemented as recurrent neural networks so that the whole pipeline can be trained end-to-end. The end-to-end training encourages the feature extractor and the cost volume regularizer to produce output which complies with the smoothness constraints of depth maps, leading to more smooth as well as complete depth estimation results as is shown in Fig. 1.

2. Related work

Traditional MVS methods optimize the depth value of each pixel by using projection relationship among multiple views. Schonberger et al. presented Colmap [21][22], which uses hand-crafted features in patch matching and optimizes the depth value pixel-by-pixel. Colmap performs well on diverse scenarios including public multi-view benchmarks and internet photos. However, traditional methods like Colmap are time-consuming because they optimize depth values of pixels one-by-one which is hard to be implemented in parallel. Besides, deep image features extracted by CNNs have been proved to be more expressive and informative than hand-crafted ones in many vision tasks such as image classification [25] and semantic segmentation [4]. Recent learning-based methods[10][29][30] outperform traditional methods on public benchmarks and greatly reduce the time consumption for more than 10 times.

There are mainly two different kinds of approaches for the learning-based MVS. One is based on voxels and the other uses depth maps as intermediate representations. Voxels-based methods split the space into regular grids and directly estimate the occupancy of each grid. Choy et al. proposed 3D-R2N2 [6], an end-to-end pipeline which regards the 32^3 voxels as hidden variables of a convolutional LSTM [9], and feeds in image features as input. Kar et al. proposed to use differentiable projection to build feature volumes from image features, which explicitly incorporates geometry prior defined by projections [14]. Voxel based methods usually suffer from huge memory requirement caused by volumetric representations, thus the space resolution of their reconstructed models is usually no more than 256^3. To enlarge space resolution, Ji et al. proposed to split the whole space into smaller Colored Voxel Cubes (CVCs) and regress the surface identity cube-by-cube [13]. However, this leads to high time complexity.

Compared to voxel cubes, depth maps are two-dimensional representations which consume much less memory during computation. In binocular stereo, a vision task highly related to MVS, disparity map between a pair of images is used as the intermediate represent, which is basically a form of depth estimation. Learning-based binocular stereo methods commonly build cost volume for a pair of images to estimate the disparity map. Inspired by this, MVS methods such as [29], [10] and [5] use CNNs to extract image features, build 3D cost volume with images from multiple views and regress the depth maps for each view. To extend from the pair-wise cost volume building to a multi-view setting, MVS methods usually select one image as the reference image once and the other input images are called the source images. For each pair of the reference image and one of its source images, a cost volume is built. Choi et al. proposed to compute the weighted sum of these cost volumes as the final cost volume for the reference image [5]. Huang et al. proposed to use a max operation to merge the cost volumes [10]. Yao et al. proposed to compute the variance of cost volumes alternatively [29].

UNet was initially proposed to deal with medical image segmentation [20]. By modifying the fully convolutional neural network (FCN) [17], UNet uses a down-sampling path followed by a symmetric up-sampling path to produce per-pixel segmentation results. The down-and-up architecture and its skip connections between shallow layers and deep layers have been widely used in different segmentation networks. Considering that depth estimation is in na-

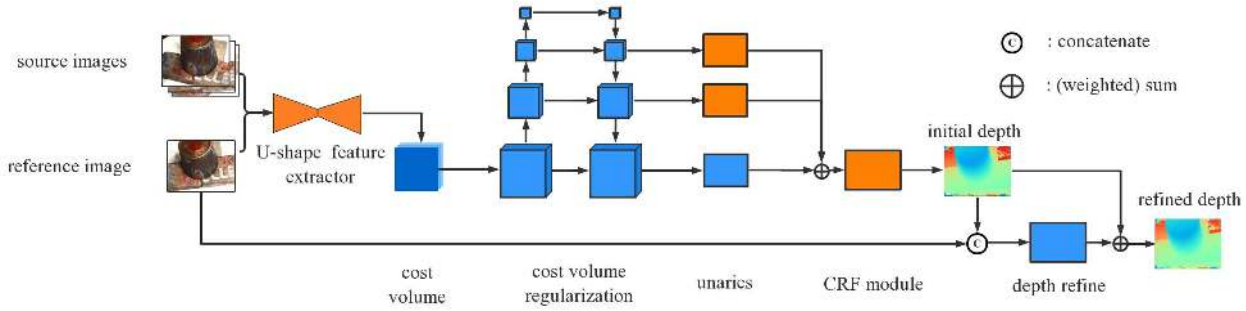


Figure 2. The overall architecture of MVSCRF.

ture similar to segmentation, we adopt the U-shape structure and skip connections to form our feature extractor.

Conditional random fields (CRFs) [2] are a kind of probability graph model. Pixels in an image have some relationship that cannot be automatically modeled by neural networks. CRFs can serve as human priors by constraining the output of pixel-wise predictions explicitly. Segmentation research usually use CRFs to model relationship of labels among different pixels [31]. Ristovski et al. proposed to model an image as a fully connected graph to solve problems of image denoising in remote sensing [19]. In the area of monocular depth estimation, Xu et al. implemented CRFs as sequential deep networks so that the whole pipeline can be trained end-to-end [27]. Huang et al. used Dense-CRF in the post-processing of the estimated depth map in MVS [10]. The vanilla CRFs are optimization-based, so it cannot be jointly trained with nerual networks directly. Zheng et al. provided a way to model CRFs as recurrent neural networks for segmentation tasks so that the CRFs module can be trained end-to-end with neural networks [31]. Different from segmentation task, the number of depth samples, whose counterparts are the semantic labels in segmentation task, is expected to be flexible for different scenarios. We therefore re-design the RNN-formed CRFs module so that the model parameters are independent of the number of depth samples.

3. MVSCRF

We adopt MVSNet as the baseline architecture to which modifications are mainly made in the stages of feature extraction and cost volume regularization as shown in Fig. 2.

3.1. Revisiting MVSNet

The MVSNet pipeline can be divided into five stages: pre-processing, feature extraction, cost volume building, cost volume regularization and post-processing. In fact, this is the de facto standard pipeline for most depth map based MVS such as DeepMVS, and more recently, R-MVSNet. In the pre-processing step,the camera's intrinsics and extrinsics, the depth range and the selection of source images are determined by using Colmap or other traditional method like OpenMVG[18] for sparse reconstruction. Then the feature extractor extracts deep features from the reference image and its source images.

Next, a cost volume is built upon the reference camera's frustum by warping a source image's feature map to some depth hypotheses of the reference image. All the cost volumes for a single reference image are merged by computing the variance among them. After that, the merged cost volume is fed in into a regularizer which is a 3D UNet in MVSNet and convolutional gated recurrent units in R-MVSNet to produce the initial depth map. MVSNet presents a refinement module after the initial output of the regularizer and finally, the depth maps from different views are merged together to produce the point cloud.

3.2. The U-shape feature extractor

We use a six-scale U-shape structure to extract deep features, as shown in Fig. 3. The down-sampling path consists of six different scales, each of which is two times smaller than the higher one. Hence, the smallest feature map is 32 times smaller than the original input. The up-sampling path is designed to be exactly symmetric to the down-sampling part. Two more convolutional layers are implemented between these two parts. In each scale of the networks, a convolutional (or deconvolutional) layer with stride 2 is designed to down-sample (or up-sample) the feature map instead of pooling layers according to [24]. After each convolutional (or deconvolutional) layer with stride 2, a convolutional layer with stride 1 is implemented to extract features at this scale. The original UNet concatenates two feature maps from the same scale of the down-sampling path and the up-sampling path to implement skip connections. In order to save memory, our feature extractor is designed to directly add two feature maps. After the U-shape structure, the feature maps are down-sampled by two convolutional layers with stride 2, each of which is followed by a

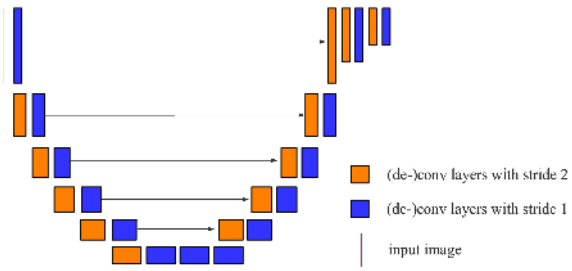


Figure 3. The architecture of the U-shape feature extractor.

convolutional layer with stride 1. Except for the last layer, batch-normalization [11] operation and ReLU activation are implemented for each layer. The final output of the feature extractor is of the size $H/4 * W/4 * 32$.

The U-shape structure provides features of different scales of resolution. And with 32 times down-sampling, the receptive fields are much larger than the original MVS-Net, so the features contain more global information. Meanwhile, the skip connections from shallow layers to deep layers help to retain rich local information to facilitate precise localization. Experiments show that the proposed feature extractor helps improve the performance of depth estimation significantly.

3.3. Conditional random fields

Formally, the depth estimation can be considered as a multi-label classification problem and each depth hypothesis corresponds to a different label. Our intuition is that nearby pixels in the inner area of an object tend to have similar labels (depth hypotheses), and pixels near borders or edges may have significantly different labels. We regard a depth map as fully connected pairwise conditional random fields conditioned on the corresponding image $\mathbf{I}$, in which each pixel is to be assigned with a depth label.

Let $\mathbf{x} = \{x_1, x_2, ..., x_N\}$ be the label vector of N pixels in a depth map. Component x_i belongs to $\{1, 2, ..., D\}$ where D is the number of depth samples. The probability of the label assignment is defined in the form of Gibbs distribution as $P(\mathbf{x}|\mathbf{I}) = \frac{1}{Z} exp(-E(\mathbf{x}|\mathbf{I}))$, where $E(\mathbf{x})$ is the energy function which describes the cost of label assigning, and Z is a normalization factor. For convenience we drop the notation of condition $\mathbf{I}$ from now on. Following the formulation in [31], the energy function defined in (1) includes a unary term and a pairwise term, where ψ_u defines the self-energy of assigning label x_i, and ψ_p defines the mutual-energy of a pair of labels.

$$E(\mathbf{x}) = \sum_{i=1}^{N} \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{1}$$

The output C of the cost volume regularizer provides a convenient measurement of the cost of labeling a pixel. We defines C_i as the cost vector of the i-th pixel, and $\psi_u(x_i)$ is set to be $C_i(x_i)$. In this way, the influence of the CNNs is embedded into the probability distribution of the depth map. As is suggested in [31], the mutual-energy term is defined in (2), where $\mu(x_i, x_j)$ is a symmetric distance measurement between two labels. In (2), $\omega^{(1)}$, $\omega^{(2)}$ are two weights for two different Gaussian kernels $k^{(1)}$ and $k^{(2)}$, and $\mathbf{f}_i^{(m)}$ denotes the feature describing the i-th pixel in the four times down-sampled input image, for instance the coordinates or the RGB values of the pixel. We adopt the same features and gaussian kernels as in [31]. Till now, the joint probability distribution of label assignment is completely defined. The distribution combines information extracted from neural networks and smoothness constraints reflecting the intrinsic characteristics of depth maps.

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{2} \omega^{(m)} k^{(m)}(\mathbf{f}_i^{(m)}, \mathbf{f}_j^{(m)}) \tag{2}$$

Generally speaking, the exact maximization of the original probability distribution is intractable. To approximate the distribution, $P(\mathbf{x})$ is usually decoupled into products of distributions of each pixel as $\prod_{i=1}^{N} Q_i(x_i)$. As such, it can be solved in an iterative manner using mean-field inference, which is shown in Alg. 1 in detail. Zheng et al. implemented the mean-field inference as recurrent neural networks so that the CRFs module can be trained with the whole pipeline in an end-to-end manner [31]. In brief, the Gaussian filtering is modeled as convolutional layers, the normalization operation is modeled as soft-max operation, and the weights $\mu(x_i, x_j)$ are modeled as parameters of $1*1$ convolutions. The recurrent neural networks iterate for T iterations for a forwarding-pass.

Algorithm 1 Mean-field inference

$Q_i(l) = \frac{1}{Z_i} exp(C_i(l))$, **for** $i = 1, 2, \ldots, N$
for $t = 1 : T$
  $\widetilde{Q}_i^{(m)}(l) = \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$,
  $\check{Q}_i(l) = \sum_{m=1}^{2} \omega^{(m)} \widetilde{Q}_i^{(m)}(l)$,
  $\hat{Q}_i(l) = \sum_{l'=1}^{D} \mu(l, l') \check{Q}_i(l)$,
  $\check{Q}_i(l) = C_i(l) - \hat{Q}_i(l)$,
  $Q_i(l) = \frac{1}{Z_i} exp(\check{Q}_i(l))$
end

Different from the semantic segmentation task in which the number of labels (object classes) are usually fixed, the number of depth samples may need to be changed for different scenarios. Therefore, the parameters $\mu(x_i, x_j)$ are better to be independent to the depth number D. In [31], the parameters $\mu(x_i, x_j)$ are learned during training, and the learned results show that the matrix converges to be

close to a diagonal-like matrix with small values lying on the diagonal line and large values in other positions. Based on this observation, we simplify the distance measurement of different depth hypotheses to a dualistic problem, as is shown in (3) where μ_0 and μ_1 are two learnable scalars. In this way, since the matrix $\mu(i,j)$ only contains two scalar parameters, it is completely decoupled with the number of depth samples D.

$$\mu(i,j) = \begin{cases} \mu_0 & |i-j| \leq 1 \\ \mu_1 & otherwise \end{cases} \tag{3}$$

To further utilize information from different resolutions, we add two branches of convolutional layers following the two lower scales of the 3D UNet as shown in Fig. 1. Each branch outputs a cost volume containing information of its scale. Denote them as C^1 and C^2, and re-denote the original output C of cost volume regularization as C^3, then we replace the unary term of CRFs with $\sum_{i=1}^{3} \alpha_i * C^i$ to build the multi-scale CRFs. The weights α_i are learnable parameters and we constraint $\sum_i^3 \alpha_i = 1$. The multi-scale version of CRFs incorporates more global information from the lower scales of the 3D UNet. The output of our proposed CRFs module takes the role of the softmax of $-C$ in original MVSNet. And the following architecture of the pipeline remains unchanged to MVSNet.

3.4. Post-processing

Similar to the geometric filtering criteria in MVSNet, we use the re-projection error to measure the confidence of depth estimation. In MVSNet, the output depth map is also filtered by a photometric confidence map computed from the probability distribution. In our implementation, the CRFs module changes the property of probability distribution of depth and the method to generate confidence map is no longer suitable. Nevertheless, experiments show that the geometric filtering itself is already good enough for filtering depth maps and is even not that necessary for simple scenarios such as scenes in the DTU dataset [12].

4. Experiments

4.1. Datasets

Experiments in this work are carried out on two public datasets: the indoor DTU dataset [12] and the outdoor $Tanks\ and\ Temples$ dataset [15]. DTU dataset contains more than 100 scenes captured on an experimental platform. Each scene has 49 or 64 images of different views under 7 different lighting conditions. The image is of size $1600 * 1200$, and the depth range of a scene is between $425mm$ and $935mm$. Point clouds with normal information are provided so that ground truth depth maps can be generated. $Tanks\ and\ Temples$ dataset contains two sets of scenes, the intermediate one and the advanced one. We only use the intermediate set for evaluation. In the intermediate set, there are 8 different scenes, each of which corresponds to a short video. A set of 2148 preselected images are provided as inputs. DTU dataset was collected in well-controlled laboratory conditions, while $Tanks\ and\ Temples$ dataset was collected in real outdoor scenarios, which is much more complex than that of DTU. The area of the scenes varies from $5m^2$ to more than $100m^2$, and the natural lighting conditions are also very different from the well-controlled experimental setups.

4.2. Training

MVSNet provides the pre-processed training data of DTU dataset. For fair comparison, we trained our model on DTU dataset following the training configurations of MVSNet[29]. The input images are resized to $640 * 512$. For each reference image, two source images are selected accordingly. The whole dataset is split into a training set, a validation set and an evaluation set. The training set consists of 27097 training samples with each image being used as the reference image. The depth hypotheses are uniformly sampled between $425mm$ to $935mm$. The number of depth samples D is set to 256 in MVSNet. However, we find that there is very little difference of performance when our proposed model is trained with $D = 128$. To speed up the training process, our models are all trained with 128 depth samples. The CRFs module in our model is trained together with the whole networks in an end-to-end manner. The iteration number T for the recurrent networks is set to be 5. All other parameters in the model are to be learned during training.

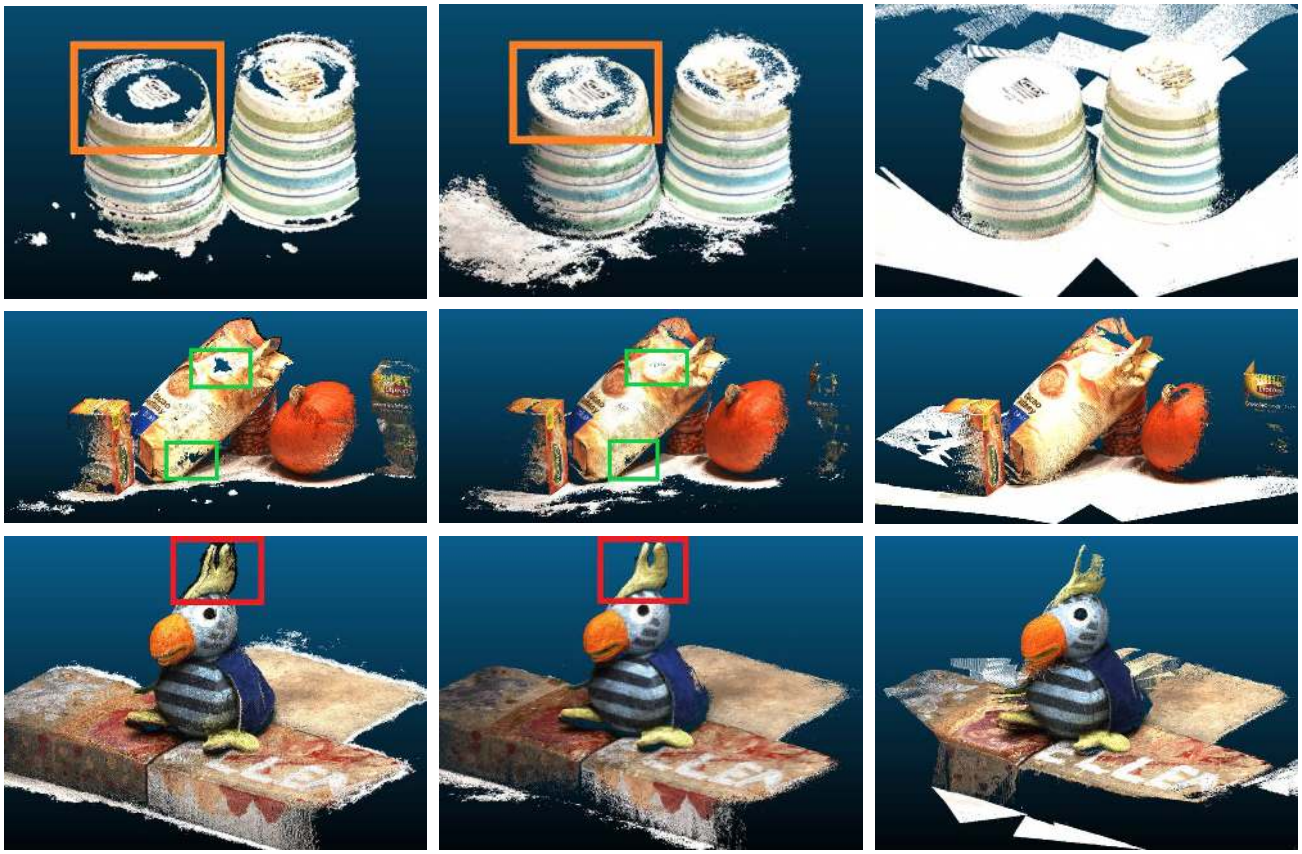
4.3. Testing on DTU dataset

The evaluation set of DTU contains 22 different scenes. We use our proposed model to generate depth maps for each image, and then merge them into point clouds by using $fusibile$ [8]. For DTU dataset, the backgrounds are clear, so there is no need to filter the depth maps before point clouds merge. Following MVSNet, the input image is of size $1600 * 1184$, the number of input views is 5, and the number of depth samples for inference is 256. We calculate $accuracy$ $(acc.)$ and $completeness$ $(comp.)$ using the official code provided by DTU dataset. Also, a percentage measurement [15] is computed. Table 1 shows the quantitative results of our method. Our method generally outperforms previous methods including MVSNet and its recent extension R-MVSNet.

Qualitative comparisons on DTU are shown in Fig. 4. More visual results are presented in the supplementary material. As is shown in the red/orange boxes, MVSCRF reduces the outliers compared to R-MVSNet, leading to the reduction of $accuracy$ distance. It may due to more global semantic cues incorporated by our U-shape feature extrac-

	Mean distance(mm)			Percentage($< 1mm$)			Percentage($< 2mm$)		
	acc.	*comp.*	*overall*	*acc.*	*comp.*	f-*score*	*acc.*	*comp.*	f-*score*
Camp*[3]	0.835	0.554	0.695	71.75	64.94	66.31	84.83	67.82	73.02
Furu*[7]	0.613	0.941	0.777	69.55	61.52	63.26	78.99	67.88	70.93
Tola*[26]	0.342	1.19	0.766	90.49	57.83	68.07	93.94	63.88	73.61
Gipuma*[8]	**0.283**	0.873	0.578	**94.65**	59.93	70.64	**96.42**	63.81	74.16
Colmap*[22]	0.400	0.664	0.532	-	-	-	-	-	-
SurfaceNet[13]	0.450	1.04	0.745	83.8	63.38	69.95	87.15	67.99	74.4
MVSNet[29]($D = 256$)	0.396	0.527	0.462	86.46	71.13	75.69	91.06	75.31	80.25
R-MVSNet[30]($D = 512$)	0.383	0.452	0.417	-	-	-	-	-	-
MVSCRF	0.371	**0.426**	**0.398**	83.82	**78.49**	**80.02**	87.64	**82.03**	**83.84**

Table 1. Quantitative results on DTU. Our $overall$ mean distance (smaller is better) and f-$scores$ (larger is better) are the best among all methods including learning-based methods and traditional methods(*). Noted that although R-MVSNet samples more depth hypotheses, our method still outperforms it both in accuracy and completeness.



(a) R-MVSNet (b) MVSCRF (c) Ground Truth

Figure 4. Qualitative comparison on DTU dataset. From the top to the bottom are respectively scan48, scan32 and scan4. It is noted that, the results of our method are more complete in texture-less area and have cleaner boundaries compared to results of R-MVSNet.

tor. Guided by semantic information, the network refines the depth estimation of pixels which are not effectively described by local features. Also, reconstructed models of our proposed MVSCRF are more complete in continuous regions, as indicated by the orange/green boxes. We believe that the completeness improvement is mainly contributed by the CRFs module, since these continuous surfaces comply to the smoothness hypothesis well which is explicitly modeled by CRFs. We further experiment with different settings to verify our analysis. According to Table 3, with the U-shape feature extractor only, the $accuracy$ distance reduces significantly by $0.045mm$. The experiment

method	rank	mean	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
ACMH*[28]	3.00	54.82	69.99	49.45	45.12	59.04	52.64	52.37	58.34	51.61
MVSCRF	7.88	45.73	59.83	30.60	29.93	51.15	50.61	51.45	52.60	39.68
R-MVSNet[30]	8.38	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38
Pix4D†[1]	12.25	43.24	64.45	31.91	26.43	54.41	50.58	35.37	47.78	34.96
MVSNet[29]	12.38	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
Colmap*[22]	12.38	42.14	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04

Table 2. Quantitative results on $Tanks\ and\ Temples$. Our result is better than MVSNet. ACMH [28] and Colmap [22] are optimization-based traditional methods(*) . Pix4D[1] is a commercial software(†).

Figure 5. Qualitative results of $Tanks\ and\ Temples$ dataset.

	acc.	*comp.*	*overall*
baseline(MVSNet)	0.396	0.527	0.462
+U-extractor	**0.351**	0.513	0.432
+U-extractor+MSCRFs	0.371	**0.426**	**0.398**

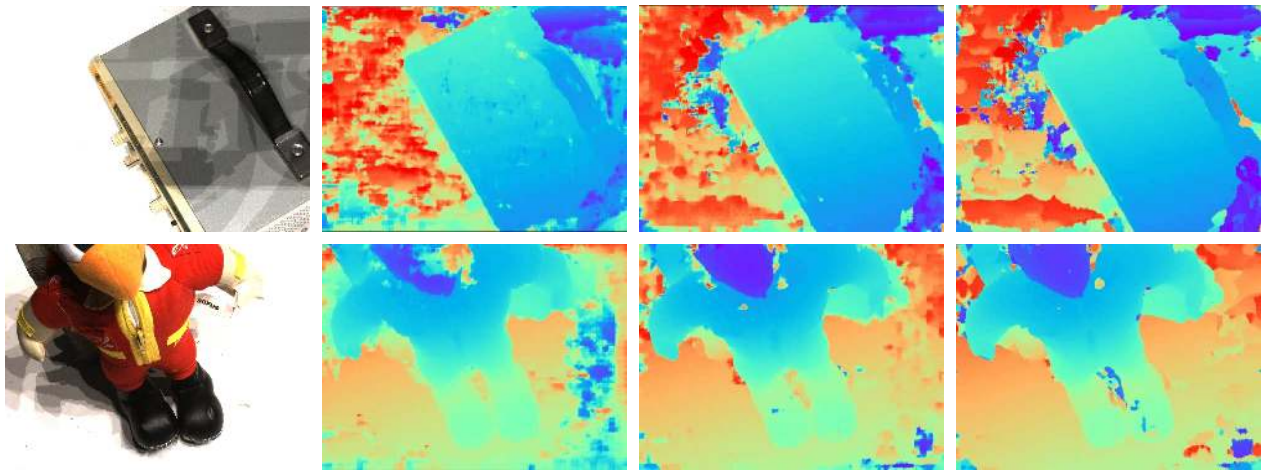
Table 3. Quantitative evaluation of each component. Baseline refers to the MVSNet pipeline. U-extractor refers to the U-shape feature extractor. MSCRFs refers to the multi-scale CRFs module.

demonstrates that the fusion of global features and local features contributes to the reconstruction, especially for the $accuracy$. Further, the multi-scale CRFs module reduces the $completeness$ distance from $0.513mm$ to $0.426mm$. Conclusions can be made that the $completeness$ improvement mainly originates from the explicit smoothness constraints imposed by the CRFs module. Fig. 6 shows the qualitative comparison. It can be noticed that depth maps produced with the U-shape feature extractor and the CRFs module are smoother in inner area and have clearer edges. Noted that R-MVSNet implements 512 depth samples which double our setting. It is always expected that denser depth sampling are effective in producing preciser predictions. However, the DTU scenes are generally of limited depth range, and the shapes of objects are relatively regular. Therefore for this kind of scenarios, our proposal seems to be more effective than increasing depth samples.

4.4. Testing on $Tanks\ and\ Temples$ dataset

$Tanks\ and\ Temples$ dataset is more challenging and we evaluate our method on it without any fine-tuning to demonstrate our method's generalization ability. In outdoor situations, images contain a lot of objects far outside the estimated depth range. Thus, the depth predictions of these background pixels are unreliable. We use the geometric criteria mentioned in Section 3.4 to filter the depth maps before merging them into point clouds. We follow the setups in MVSNet, with the input size being $1920 * 1056$, 5 input views and 256 depth samples.

$Tanks\ and\ Temples$ benchmark measures the overall quality of reconstructed point clouds by using a metric called $f\text{-}score$. Briefly speaking, the $f\text{-}score$ is a combination of $accuracy$ and $completeness$. Table 2 shows the quantitative results of our method and other published state-of-the-arts. Our method outperforms MVSNet obviously on 7 out 8 scenes. Fig. 5 shows the reconstructed models for the 8 scenes. We achieve comparable results with R-MVSNet in terms of rank, but for scenes $Family$ and $Francis$, R-MVSNet produces much better results than



(a) reference image (b) baseline (c) baseline+U (d) baseline+U+MSCRFs

Figure 6. Qualitative evaluation of each component. We use the codes and the trained model provided by Yao et al [29].

ours. Take scene $Francis$ for example, it consists of thin structure including a delicate spire, four pillars and a statue in it, which do not comply to our smoothness hypothesis. R-MVSNet uses an average of 898 depth samples to achieve its results while we only sample 256 depth hypotheses. On the contrary, for scene $Lighthouse$, our method's performance is obviously better. It may be explained that the scene $Lighthouse$ contains planes such as roofs and walls, which are suitable for our intuitions.

Our results on two different datasets show that, in scenes having restricted depth range and relatively smaller local depth changes, such as DTU scenes or some scenes in $Tanks\ and\ Temples$, incorporating priors of depth maps has even larger influence on the quality of reconstructed models than increasing the sampling rate. But for those delicate models with thin structure, denser samplers are more favorable. Our proposal of utilizing depth maps' priors may be combined with denser samplers to achieve better results.

4.5. Discussion

Complexity: The time/memory consumption are evaluated using DTU validation settings: $H * W = 512 * 640$,$D = 256$.In terms of parameter counts and run time, MVSCRF is between MVSNet and R-MVSNet as is shown in Table 4. MVSCRF consumes slightly more memory than MVSNet because of the UNet and CRF module, however the space complexity of MVSCRF and MVSNet is the same.

	Param.	Time	Memory	Complexity
MVSCRF	571K	1.8s	5.43GB	$O(H \times W \times D)$
MVSNet	363K	0.9s	5.28GB	$O(H \times W \times D)$
R-MVSNet	812K	2.2s	4.39GB	$O(H \times W)$

Table 4. Comparison of computational efficiency.

Post-processing: For fair comparison, we reproduce MVSNet with our post-processing step as well as the $fusibile$ depth fusion, and achieve a mean distance of $0.462mm$ which is close to that of MVSNet($0.460mm$).

	multi-scale	single scale	MVSNet
mean f-score	45.73%	44.00%	43.48%

Table 5. Performance comparison using different CRF settings.

CRF settings: In multi-scale CRF, matching results of multi-level image features are used to generate a more accurate self-energy term, leading to more appropriate smoothness constraints. Table 5 shows that this is critical in producing high quality depth map, especially for complicate scenes containing structures of different scales in $Tanks\ and\ Temples$. For simple scenes like those in DTU, however, the performance gain of multi-scale CRF is very marginal (f-score: $79.77\% \rightarrow 80.02\%$).

5. Conclusions

We present an end-to-end pipeline for multi-view stereo with conditional random fields (MVSCRF). Following the basic architecture of MVSNet, we incorporate priors of depth maps into the network design. A U-shape feature extractor is designed to extract informative deep features combining both local and global information. And conditional random fields are implemented as the form of recurrent neural networks to explicitly define the smoothness constraints on depth maps. Quantitative and qualitative results on public dataset DTU and $Tanks\ and\ Temples$ demonstrate the effectiveness of our method.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (61673234).

References

[1] https://pix4d.com.

[2] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[3] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, pages 766–779. Springer, 2008.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018.

[5] Sungil Choi, Seungryong Kim, Kihong Park, and Kwanghoon Sohn. Learning descriptor, confidence, and depth estimation in multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 276–282, 2018.

[6] 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.

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

[8] 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.

[9] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[10] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning-Volume 37*, pages 448–456. JMLR. org, 2015.

[12] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[13] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2307–2315, 2017.

[14] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017.

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

[16] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[18] 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.

[19] Kosta Ristovski, Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic. Continuous conditional random fields for efficient regression in large fully connected graphs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[21] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[22] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518. Springer, 2016.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[26] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.

[27] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

[28] Qingshan Xu and Wenbing Tao. Multi-view stereo with asymmetric checkerboard propagation and multi-hypothesis joint view selection. *arXiv preprint arXiv:1805.07920*, 2018.

[29] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *ECCV*, 2018.

[30] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. *arXiv preprint arXiv:1902.10556*, 2019.

[31] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.