Rethinking Depth Estimation for Multi-View Stereo: A Unified Representation

Rui Peng  1  Rongjie Wang  2  Zhenyu Wang  1  Yawen Lai  1  Ronggang Wang  1,2
1School of Electronic and Computer Engineering, Peking University  2Peng Cheng Laboratory
ruipeng@stu.pku.edu.cn  rgwang@pkusz.edu.cn
https://github.com/prstrive/UniMVSNet

Abstract

Depth estimation is solved as a regression or classification problem in existing learning-based multi-view stereo methods. Although these two representations have recently demonstrated their excellent performance, they still have apparent shortcomings, e.g., regression methods tend to overfit due to the indirect learning cost volume, and classification methods cannot directly infer the exact depth due to its discrete prediction. In this paper, we propose a novel representation, termed Unification, to unify the advantages of regression and classification. It can directly constrain the cost volume like classification methods, but also realize the sub-pixel depth prediction like regression methods. To excavate the potential of unification, we design a new loss function named Unified Focal Loss, which is more uniform and reasonable to combat the challenge of sample imbalance. Combining these two unburdened modules, we present a coarse-to-fine framework, that we call UniMVSNet. The results of ranking first on both DTU and Tanks and Temples benchmarks verify that our model not only performs the best but also has the best generalization ability.

1. Introduction

Multi-view stereo (MVS) is a vital branch to extract geometry from photographs, which takes stereo correspondence from multiple images as the main cue to reconstruct dense 3D representations. Although traditional methods [2,7,25,26] have achieved excellent performance after occupying researchers for decades, more and more learning-based approaches [4,5,9,34–36] are proposed to promote the effectiveness of MVS due to their more powerful representation capability in low-texture regions, reflections, etc. Concretely, they infer the depth for each view from the 3D cost volume, which is constructed from the warped feature according to a set of predefined depth hypotheses. Compared with hand-crafted similarity metrics in traditional methods, the 3D cost volume can capture more discriminative features to achieve more robust matching. Without loss of integrity, existing learning-based methods can be divided into two categories: Regression and Classification.

Regression is the most primitive and straightforward implementation of the learning-based MVS method. It’s a group of approaches [5,9,20,34,35,39] to regress the depth from the 3D cost volume through Soft-argmin, which softly weighting each depth hypothesis. More specifically, the model expects to regress greater weight for the depth hypothesis with a small cost. Theoretically, it can achieve the sub-pixel estimation of depth by weighted summation of discrete depth hypotheses. Nevertheless, the model needs to learn a complex combination of weights under indirect constraints performed on the weighted depth but not on the weight combination, which is non-trivial and tends to overfit. You can imagine that there are many weight combinations for a set of depth hypotheses that can be weighted and summed to the same depth, and this ambiguity also implicitly increases the difficulty of the model convergence.

Classification is proposed in R-MVSNet [36] to infer the optimal depth hypothesis. Different from the weight estimation in regression, classification methods [10,33,36] predict the probability of each depth hypothesis from the 3D cost volume and take the depth hypothesis with the maximum probability as the final estimation. Obviously, these methods cannot infer the exact depth directly from the model like regression methods. However, classification methods directly constrain the cost volume through the cross-entropy loss executed on the regularized probability volume, which is the essence of ensuring the robustness of MVS. Moreover, the estimated probability distribution can directly reflect the confidence, which is difficult to derive from the weight combination intuitively.

In this paper, we seek to unify the advantages of regression and classification, that is, we hope that the model can accurately predict the depth while maintaining robustness. There is a fact that the depth hypothesis close to the ground-truth has more potential knowledge, while that of other remaining hypotheses is limited or even harmful due to the wrong induction of multimodal [42]. Motivated by this, we present that estimating the weights for all depth hypotheses is redundant, and the model only needs to do regression on the optimal depth hypothesis that the representative depth interval (referring to the upper area until the next
larger depth hypothesis) contains the ground truth depth. To achieve this, we propose a unified representation for depth, termed Unification. As shown in Fig. 1, unlike regression, the loss is executed on the regularized probability volume directly, and different from classification, our method estimates the Unity (What we call), whose label is composed by at most one non-zero continuous target ($0 \sim 1$), to simultaneously represent the location of optimal depth hypothesis and its offset to the ground-truth depth. We take proximity (defined as the complement of the offset between ground-truth and optimal depth hypothesis) to characterize the non-zero target in unity label, which is more efficient than purely using offset. The detailed comparisons are in Supp. Mat.

Moreover, we note that this unified representation faces an undeniable sample imbalance in both category and hardness. While Focal Loss (FL) [19] is the common solution proposed in the detection field, which is tailored to the traditional discrete label, the more general form (GFL) is proposed in [17, 40] to deal with the continuous label. Even though GFL has demonstrated its performance, we hold the belief that it has an obvious limitation in distinguishing hard and easy samples due to ignoring the magnitude of ground-truth. To this end, we put forward a more reasonable and unified form, called Unified Focal Loss (UFL), after thorough analysis to better address these challenges. In this way, the traditional FL can be regarded as a special case of UFL, while GFL is its imperfect expression.

To demonstrate the superiority of our proposed modules, we present a coarse-to-fine framework termed UniMVSNet (or UnifiedMVSNet), named for its unification of depth representation and focal loss, which replaces the traditional representation of recent works [5,9,34] with Unification and adopts UFL for optimization. Extensive experiments show that our model surpasses all previous MVS methods and achieves state-of-the-art performance on both DTU [1] and Tanks and Temples [14] benchmarks.

2. Related Works

Traditional MVS methods. Taking the output scene representation as an axis of taxonomy, there are mainly four types of classic MVS methods: volumetric [15, 27], point cloud based [7, 16], mesh based [6] and depth map based [3,8,24,25,31]. Among them, the depth map based method is the most flexible one. Instead of operating in the 3D domain, it degenerates the complex problem of 3D geometry reconstruction to depth map estimation in the 2D domain. Moreover, as the intermediate representation, the estimated depth maps of all individual images can be merged into a consistent point cloud [22] or a volumetric reconstruction [23], and the mesh can even be further reconstructed.

Learning-based MVS methods. While the traditional MVS pipeline mainly relies on hand-crafted similarity metrics, recent works apply deep learning for superior performance on MVS. SurfaceNet [11] and LSM [12] are the first proposed volumetric learning-based MVS pipelines to regress surface voxels from 3D space. However, they are restricted to memory which is the common drawback of the volumetric representation. Most recently, MVSNet [35] first realizes an end-to-end memory low-sensitive pipeline based on 3D cost volumes. This pipeline mainly consists of four steps: image feature extraction by 2D CNN, variance-based cost aggregation by homography warping, cost regularization through 3D CNN, and depth regression. To further excavate the potential capacity of this pipeline, some variants of MVSNet have been proposed, e.g., [5,9,34,36] are proposed to reduce the memory requirement through RNN or coarse-to-fine manner, [20, 38] are proposed to adaptively re-weight the contribution of different pixels in cost aggregation. Meanwhile, all existing methods are based on one of the two complementary of classification and regression to infer depth. In this paper, we propose a novel unified representation to integrate their advantages.
3. Methodology

This section will present the main contributions of this paper in detail. We first review the common pipeline of the learning-based MVS approach in Sec. 3.1, then introduce the proposed unified depth representation in Sec. 3.2 and unified focal loss in Sec. 3.3, and finally describe the detailed network architecture of our UniMVSNet in Sec. 3.4.

3.1. Review of Learning-based MVS

Most end-to-end learning-based MVS methods are inherited from MVSNet [35], which constructs an elegant and effective pipeline to infer the depth \( D \in \mathbb{R}^{H \times W} \) of the reference image \( I_1 \). Given multiple images \( \{I_i \in \mathbb{R}^{C \times H \times W}\}_{i=1}^{N} \) of a scene taken from \( N \) different viewpoints, image features of all images \( \{F_i \in \mathbb{R}^{C' \times H' \times W'}\}_{i=1}^{N} \) are first extracted through a 2D network with shared weights. As mentioned above, the learning-based method is based on the 3D cost volume, and the depth hypothesis of \( M \) layers \( \{d_i \in \mathbb{R}^{H \times W}\}_{i=1}^{M} \) is sampled from the whole known depth range to achieve this, where \( d_1 \) represents the minimum depth and \( d_M \) represents the maximum depth. With this hypothesis, feature volumes \( \{V_i \in \mathbb{R}^{M \times C' \times H' \times W'}\}_{i=1}^{N} \) can be constructed in 3D space via differentiable homography by warping 2D image features of source images to the reference camera frustum. The homography between the feature maps of \( i_{th} \) view and the reference feature maps at depth \( d \) is expressed as:

\[
H_i(d) = dK_iT_i^{-1}K_i^{-1}
\]  

(1)

where \( K \) and \( T \) refer to camera intrinsics and extrinsics respectively.

To handle arbitrary number of input views, the multiple feature volumes \( \{V_i\}_{i=1}^{N} \) need to be aggregated to one cost volume \( C \in \mathbb{R}^{M \times C' \times H' \times W'} \). The aggregation strategy consists two dominant groups: statistical and adaptive. The variance-based mapping is a typical statistical aggregation:

\[
C = \frac{1}{N} \sum_{i=1}^{N} (V_i - \bar{V})^2
\]

(2)

Where \( \bar{V} \) denotes the average feature volume. Furthermore, the adaptive aggregation is proposed to re-weight the contribution of different pixels, which can be modeled as:

\[
C = \frac{1}{N-1} \sum_{i=2}^{N} W_i \odot (V_i - V_1)^2
\]

(3)

where \( W \) is the learnable weight generated by an auxiliary network, and \( \odot \) denotes element-wise multiplication.

The matching cost between the reference view and all source views under each depth hypothesis has been encoded into the cost volume, which is required to be further refined to generate a probability volume \( P \in \mathbb{R}^{M \times H' \times W'} \) through a softmax-based regularization network. Concretely, the probability volume is treated as the weight of depth hypotheses in regression methods and the depth at pixel \((x, y)\) is calculated as the sum of the weighted hypotheses as:

\[
D^{x,y} = \sum_{d\in d_{1:M}} dP(d)^{x,y}
\]

(4)

and the model is constrained by the \( L1 \) loss between \( D \) and the ground-truth depth. In classification methods, \( P \) refers to the probability of depth hypotheses and the depth is estimated as the hypothesis whose probability is maximum:

\[
D^{x,y} = \arg \max_{d \in \{d_{1:M}\}} P(d)^{x,y}
\]

(5)

and the model is trained by the cross-entropy loss between \( P \) and the ground-truth one-hot probability volume.

In traditional one-stage methods, compared with original input images, the depth map is either downsized during feature extraction [35] or before the input [38] to save memory, while in the coarse-to-fine method [5, 9, 34], it’s a multi-scale result \( \{D_i\}_{i=1}^{L} \) with incremental resolution generated by repeating the above pipeline \( L \) times. The multi-scale is realized by a FPN-like [18] feature extraction network, and the depth hypothesis with decreasing depth range is sampled based on the depth map generated in the previous stage.

3.2. Unified Depth Representation

As aforementioned, the regression method tends to over-fit due to its indirect learning cost volume and ambiguity in the correspondence between depth and the weight combination. For the classification method, although it can constrain the cost volume directly, it cannot predict exact depth like regression methods due to its discrete prediction. In this paper, we found that they can complement each other, and we unify them successfully through our unified depth representation, as shown in Fig. 2. We recast the depth estimation as a multi-label classification task, in which the model needs to classify which hypothesis is the optimal one and regress the
proximity for it. In other words, we first adopt classification to narrow the depth range of the final regression, but they are executed simultaneously in our implementation. Therefore, the model in our Unification representation is able to estimate an accurate depth like regression methods, and it also directly optimizes the cost volume like classification methods. Below, we will introduce how to generate the ground-truth unity from ground-truth depth (Unity generalization), and how to regress the depth from the estimated unity (Unity regression).

**Unity generation:** As shown in Fig. 1, ground-truth unity \( \{ U_i \}_{i=1}^M \) is a more general form of one-hot label peaked at the optimal depth hypothesis whose depth interval contains ground-truth depth. The at most one non-zero target is a continuous number and represents the proximity of the optimal hypothesis to ground-truth depth. The detail of unity generation is shown in Algorithm 1, which is one more step of proximity calculation than the one-hot label generation in classification methods.

**Algorithm 1: Unity Generation**

*Input:* Ground-truth depth \( D_{gt} \in \mathbb{R}^{H' \times W'} \); Depth hypotheses \( \{ d_i \in \mathbb{R}^{H' \times W'} \}_{i=1}^M \).

*Output:* Ground-truth Unity \( \{ U_i \in \mathbb{R}^{H' \times W'} \}_{i=1}^M \).

*Initialization:* Depth interval \( r = 0 \).

1. for \( i = 1 \) to \( M \) do
   2. for \( (x, y) = (1, 1) \) to \( (H', W') \) do
      3. if \( i < M \) then
         4. \( r = d_{i+1}^x - d_i^x \);
         5. end
      6. if \( d_i^{x, y} \leq D_{gt}^{x, y} \) and \( d_i^{x, y} + r > D_{gt}^{x, y} \) then
         7. \( U_i^{x, y} = 1 - \frac{D_{gt}^{x, y} - d_i^{x, y}}{r} \);
         8. else
         9. \( U_i^{x, y} = 0 \);
         10. end
   11. end
12. end
13. return \( \{ U_i \}_{i=1}^M \).

**Algorithm 2: Unity Regression**

*Input:* Estimated unity \( \{ \hat{U}_i \in \mathbb{R}^{H' \times W'} \}_{i=1}^M \); Depth hypotheses \( \{ d_i \in \mathbb{R}^{H' \times W'} \}_{i=1}^M \).

*Output:* Regressed Depth \( D \in \mathbb{R}^{H' \times W'} \).

*Initialization:* Depth interval \( r = 0 \).

1. for \( (x, y) = (1, 1) \) to \( (H', W') \) do
   2. Optimal hypothesis index \( o = \arg \max_{i \in \{1, \ldots, M\}} \hat{U}_i^{x, y} \);
   3. Optimal hypothesis \( d = d_o^{x, y} \);
   4. if \( o < M \) then
      5. \( r = d_{o+1}^{x, y} - d_o^{x, y} \);
      6. else
      7. \( \text{// previous interval for the last hypothesis} \)
      8. \( r = d_o^{x, y} - d_{o-1}^{x, y} \);
      9. end
   10. Offset \( o f f = (1 - \hat{U}_o^{x, y}) \times r \);
   11. Depth \( D^{x, y} = d + o f f \);
12. end
13. return \( D \).

**Unity regression:** Unlike the traditional way of predicting the probability volume by softmax operators, unification representation estimated it through sigmoid operators. Here, we disassemble the estimated probability volume \( \hat{P} \) into the estimated unity \( \{ \hat{U}_i \}_{i=1}^M \) along the \( M \) dimension. To regress the depth, we first select the optimal hypothesis with the maximum unity at each pixel, then calculate the offset to ground-truth depth, and finally fuse the estimated depth. The detailed procedure is shown in Algorithm 2.

### 3.3. Unified Focal Loss

Generally, the depth hypothesis of MVS models will be sampled quite densely to ensure the accuracy of the estimated depth, which will cause obvious sample imbalance due to only one positive sample (the at most one non-zero target) among hundreds of hypotheses. Meanwhile, the model needs to pay more attention to hard samples to prevent overfitting. Relevant Focal Loss (FL) [19] has been proposed to solve these two problems, which auto-

---

[8648]
matically distinguishes hard samples through the estimated unity \( u \in [0, 1] \) and rebalances the sample through tunable parameter \( \alpha \) and \( \gamma \). Here, we discuss a certain pixel for convenience. The typical definition of FL is:

\[
\text{FL}(u, q) = \begin{cases} 
- \alpha(1 - u)^\gamma \log(u), & q = 1 \\
-(1 - \alpha)u^\gamma \log(1 - u), & \text{else}
\end{cases}
\]  

where \( q \in \{0, 1\} \) is the discrete target. Therefore, the traditional FL is not suitable for our continuous situation. To enable successful training under the case of our representation, we borrow the main idea from FL. Above all, the binary cross-entropy \(-\log(u)\) or \(-\log(1 - u)\) needs to be extended to its complete form \(\text{BCE}(u, q) = -q \log(u) - (1 - q) \log(1 - u)\). Correspondingly, the scaling factor should also be adjusted appropriately. The generalized FL form (GFL) obtained through these two steps is:

\[
\text{GFL}(u, q) = \begin{cases} 
\alpha|q - u|^\gamma \text{BCE}(u, q), & q > 0 \\
(1 - \alpha)|u|^\gamma \text{BCE}(u, q), & \text{else}
\end{cases}
\]

where \( q \in [0, 1] \) is the continuous target. This advanced version is currently adopted by some existing methods with different names, e.g., QFL in [17] or VFL in [40]. But in this paper, we point out that this implementation is not perfect in scaling hard and easy samples, because they ignore the magnitude of the ground-truth.

As shown in Fig. 3, the first two samples will be considered the hardest under the absolute error \( |q - u| \) measurement in GFL. However, the absolute error cannot distinguish samples with different targets. Even if the first two samples in Fig. 3 have the same absolute error, this error obviously has a smaller effect on the first sample due to its larger ground-truth. To solve this ambiguity, we further improve the scaling factor in GFL through relative error and propose our naive version of Unified Focal Loss (UFL) just as:

\[
\text{UFL}(u, q) = \begin{cases} 
\alpha\left(\frac{|q - u|}{q^{+}}\right)^\gamma \text{BCE}(u, q), & q > 0 \\
(1 - \alpha)\left(\frac{|u|}{q^{-}}\right)^\gamma \text{BCE}(u, q), & \text{else}
\end{cases}
\]

where \( q^{+} \in (0, 1] \) is the positive target. It can be seen from Eq. (8) that FL is a special case of UFL when the positive target is the constant 1.

Moreover, we noticed that the range of scaling factor \( \frac{|q - u|}{q^{+}} \in [0, +\infty) \), which may lead to a special case like the last sample in Fig. 3. Even a small number of such samples will overwhelm the loss and computed gradients due to their huge scaling factor. In this paper, we solve this problem by introducing a dedicated function to control the range of the scaling factor. Meanwhile, to keep the precious positive learning signals, we adopt an asymmetrical scaling strategy. And the complete UFL can be modeled as:

\[
\text{UFL}(u, q) = \begin{cases} 
\alpha^{+}\left(S_{b}\left(\frac{|q - u|}{q^{+}}\right)\right)^\gamma \text{BCE}(u, q), & q > 0 \\
\alpha^{-}\left(S_{b}\left(\frac{|u|}{q^{-}}\right)\right)^\gamma \text{BCE}(u, q), & \text{else}
\end{cases}
\]

where the dedicated function \( S_{b}(x) \) is designed as the sigmoid-like function \((1/(1 + b^{-x}))\) with a base of \( b \) in this paper.
3.4. UniMVSNet

It’s straightforward to apply our Unification and UFL to existing learning-based MVS methods. To illustrate the effectiveness and flexibility of the proposed modules, we build the UniMVSNet, whose framework is depicted in Fig. 4, based on the coarse-to-fine strategy. This pipeline abides by the procedure reviewed in Sec. 3.1, except the depth representation and optimization.

Inherited from CasMVSNet [9], We adopt a FPN-like network to extract multi-scale features, and uniformly sample the depth hypothesis with a decreasing interval and a decreasing number. To better handle the unreliable matching in non-Lambertian regions, we adopt an adaptive aggregation method with negligible parameters increasing like [38] to aggregate the feature volumes warped by the differentiable homography. Meanwhile, we also apply multi-scale 3D CNNs to regularize the cost volume, and the generated probability volume \( P \) at each stage is treated as the estimated Unity \( \{ \bar{U}_i \}_{i=1}^M \) here, which can be further regressed to accurate depth as shown in Algorithm 2. It can be seen from Fig. 4 that UniMVSNet directly optimizes cost volume through UFL, which can effectively avoid the overfitting of indirect learning strategy in regression methods.

Training Loss. As shown in Fig. 4, we apply UFL to all stages and fuse them with different weights. The total loss can be defined as:

\[
    Loss = \sum_{i=1}^{L} \lambda_i UFL_i
\]

where \( UFL_i \) is the average of UFL of all valid pixels at stage \( i \) and \( \lambda_i \) denotes the weight of UFL at \( i_{th} \) stage.

4. Experiments

This section demonstrates the start-of-the-art performance of UniMVSNet with comprehensive experiments and verifies the effectiveness of the proposed Unification and UFL through ablation studies. We first introduce the datasets and implementation and then analyze our results.

**Table 1. Quantitative results on DTU evaluation set.** Best results in each category are in **bold**. Our model ranks first in terms of Completeness and Overall metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC.(mm)</th>
<th>Comp.(mm)</th>
<th>Overall(mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furu [7]</td>
<td>0.613</td>
<td>0.941</td>
<td>0.777</td>
</tr>
<tr>
<td>Gipuma [8]</td>
<td><strong>0.283</strong></td>
<td>0.873</td>
<td>0.578</td>
</tr>
<tr>
<td>COLMAP [24, 25]</td>
<td>0.400</td>
<td>0.664</td>
<td>0.532</td>
</tr>
<tr>
<td>SurfaceNet [11]</td>
<td>0.450</td>
<td>1.040</td>
<td>0.745</td>
</tr>
<tr>
<td>MVSNet [35]</td>
<td>0.396</td>
<td>0.527</td>
<td>0.462</td>
</tr>
<tr>
<td>P-MVSNet [20]</td>
<td>0.406</td>
<td>0.434</td>
<td>0.420</td>
</tr>
<tr>
<td>R-MVSNet [36]</td>
<td>0.383</td>
<td>0.452</td>
<td>0.417</td>
</tr>
<tr>
<td>Point-MVSNet [4]</td>
<td>0.342</td>
<td>0.411</td>
<td>0.376</td>
</tr>
<tr>
<td>AA-RMVSNet [30]</td>
<td>0.376</td>
<td>0.339</td>
<td>0.357</td>
</tr>
<tr>
<td>CasMVSNet [9]</td>
<td>0.325</td>
<td>0.385</td>
<td>0.355</td>
</tr>
<tr>
<td>CVP-MVSNet [34]</td>
<td>0.296</td>
<td>0.406</td>
<td>0.351</td>
</tr>
<tr>
<td>UCS-Net [5]</td>
<td>0.338</td>
<td>0.349</td>
<td>0.344</td>
</tr>
<tr>
<td>UniMVSNet (ours)</td>
<td><strong>0.352</strong></td>
<td><strong>0.278</strong></td>
<td><strong>0.315</strong></td>
</tr>
</tbody>
</table>

**Datasets.** We evaluate our model on DTU [1] and Tanks and Temples [14] benchmark and finetune on BlendedMVS [37]. (a) DTU is an indoor MVS dataset with 124 different scenes scanned from 49 or 64 views under 7 different lighting conditions with fixed camera trajectories. We adopt the same training, validation, and evaluation split as defined in [35]. (b) Tanks and Temples is collected in a more complex realistic environment, and it’s divided into the intermediate and advanced set. While intermediate set contains 8 scenes with large-scale variations, advanced set has 6 scenes. (c) BlendedMVS is a large-scale synthetic dataset, which is consisted of 113 indoor and outdoor scenes and is split into 106 training scenes and 7 validation scenes.

**Implementation.** Following the common practice, we first train our model on the DTU training set and evaluate on DTU evaluation set, and then finetune our model on BlendedMVS before validating the generalization of our approach on Tanks and Temples. The input view selection and data pre-processing strategies are the same as [35]. Meanwhile, we utilize the finer DTU ground-truth as [30]. In this paper, UniMVSNet is implemented in 3 stages with 1/4, 1/2 , and 1 of original input image resolution respectively. We follow the same configuration (e.g., depth interval) of the model at each stage as [9] in both training and evaluation of DTU. When training on DTU, the number of input images is set to \( N = 5 \) and the image resolution is resized to 640 × 512. To emphasize the contribution of positive signals, we set \( \alpha^+ = 1 \), and scale the range of \( S^\alpha \) to \([1,3]\) and \( S^\gamma \) to \([0,1]\). The other tunable parameters in UFL are configured stage-wise, e.g., \( \alpha^+ \) is set to 0.75, 0.5, and 0.25, and \( \gamma \) is set to 2, 1, and 0 from the coarsest stage to the finest stage. We optimize our model for 16 epochs with Adam optimizer [13], and the initial learning rate is set to 0.001 and decayed by 0.5 after 10, 12, and 14 epochs. During the evaluation of DTU, we also resize the input image size to 1152 × 864 and set the number of the input images to 5. We report the standard metrics (accuracy, completeness,
Figure 6. Qualitative results of scan 15 on DTU. The top row shows the point clouds generated by different methods and the ground-truth, and the bottom row shows a more detailed local region corresponding to the red rectangle.

<table>
<thead>
<tr>
<th>Method</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-MVSNet [4]</td>
<td>48.27 61.79 41.15 34.20 50.79 51.97 50.83 52.38 43.06</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>PatchmatchNet [29]</td>
<td>53.15 66.99 52.64 43.24 54.87 52.87 49.54 54.28 43.06</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>UCS-Net [5]</td>
<td>54.83 76.09 53.16 43.03 54.00 55.60 51.49 57.38 47.89</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>CVP-MVSNet [34]</td>
<td>54.03 76.50 47.74 36.34 55.12 57.28 54.28 57.43 47.54</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>P-MVSNet [20]</td>
<td>55.62 70.04 44.64 40.22 65.20 55.08 59.17 60.37 54.29</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>CasMVSNet [9]</td>
<td>56.84 76.37 58.45 46.26 55.81 56.11 54.06 58.18 49.51</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>ACMP [32]</td>
<td>58.41 70.30 54.06 54.11 61.65 54.16 57.60 58.12 57.25</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>UCS-Net [5]</td>
<td>59.20 74.69 56.04 49.42 60.08 59.81 59.61 60.04 53.92</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>D²HC-RMVSNet [33]</td>
<td>60.03 77.40 60.23 47.07 63.44 62.21 57.28 60.54 52.07</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>VisMVSNet [41]</td>
<td>61.51 77.77 59.53 51.53 64.02 64.05 59.47 60.85 55.50</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>AA-RMVSNet [30]</td>
<td>61.68 77.86 60.54 52.96 62.33 61.69 60.34 62.44 55.30</td>
<td>- - - - - - - -</td>
</tr>
<tr>
<td>EPP-MVSNet [21]</td>
<td>64.36 81.20 66.43 53.11 63.46 66.09 64.84 62.23 57.53</td>
<td>38.96 28.33 44.36 39.74 52.89 33.80 34.63</td>
</tr>
<tr>
<td>UniMVSNet (ours)</td>
<td>64.36 81.20 66.43 53.11 63.46 66.09 64.84 62.23 57.53</td>
<td>38.96 28.33 44.36 39.74 52.89 33.80 34.63</td>
</tr>
</tbody>
</table>

Table 2. Quantitative results of F-score on Tanks and Temples benchmark. Best results in each category are in bold. “Mean” refers to the mean F-score of all scenes (higher is better). Our model outperforms all previous MVS methods with a significant margin on both Intermediate and Advanced set.

and overall) proposed by the official evaluation protocol [1]. Before testing on Tanks and Temples benchmark, we fine-tune our model on BlendedMVS for 10 epochs. We take 7 images as the input with the original size of $768 \times 576$. For benchmarking on Tanks and Temples, the number of depth hypotheses in the coarsest stage is changed from 48 to 64, and the corresponding depth interval is set to 3 times as the interval of [35]. We set the number of input images to 11 and report the F-score metric.

4.1. Results on DTU

Similar to previous methods [9, 35, 36], we introduce photometric and geometric constraints for depth map filtering. The probability threshold and the number of consistent views are set to 0.3 and 3 respectively, which is the same as [36]. The final 3D point cloud is obtained through the same depth map fusion method as [9, 35, 36].

We compare our method to those traditional and recent learning-based MVS methods. The quantitative results on the DTU evaluation set are summarized in Tab. 1, which indicates that our method has made great progress in performance. While Gipuma [8] ranks first in the accuracy metric, our method outperforms all methods on the other two metrics significantly. Depth map estimation and point reconstruction of a reflective and low-textured sample are shown in Fig. 5, which shows that our model is more robust on the challenge regions. Figure 6 shows some qualitative results compared with other methods. We can see that our model can generate more complete point clouds with finer details.

4.2. Results on Tanks and Temples

As the common practice, we verify the generalization ability of our method on Tanks and Temples benchmark using the model finetuned on BlendedMVS. We adopt a depth map filtering strategy similar to that of DTU, except for the geometric constraint. Here, we follow the dynamic geometric consistency checking strategies proposed in [33]. Through this dynamic method, those pixels with fewer consistent views but smaller reprojection errors and those with larger errors but more consistent views will also survive.

The corresponding quantitative results on both intermediate and advanced sets are reported in Tab. 2. Our method achieves state-of-the-art performance among all existing MVS methods and yields first place in most scenes. Notably, our model outperforms the previous best model by 2.68 points and 3.24 points on the intermediate and advanced sets. Such obvious advantages just show that our model not only has the best performance but also exhibits the strongest generalization and robustness. The qualitative point cloud results are visualized in Fig. 7.
Figure 7. Qualitative results of some scenes on Tanks and Temples.

<table>
<thead>
<tr>
<th>Method</th>
<th>Representation</th>
<th>Loss Function</th>
<th>Aggregation</th>
<th>FGT</th>
<th>Input</th>
<th>ACC.(mm)</th>
<th>Comp.(mm)</th>
<th>Overall.(mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Reg)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>3</td>
<td>0.369</td>
<td>0.317</td>
<td>0.343</td>
</tr>
<tr>
<td>Baseline (Cla)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>0.368</td>
<td>0.312</td>
<td>0.340</td>
</tr>
<tr>
<td>Baseline (Uni)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>0.425</td>
<td>0.285</td>
<td>0.355</td>
</tr>
<tr>
<td>Baseline (Uni) + GFL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>0.372</td>
<td>0.282</td>
<td>0.327</td>
</tr>
<tr>
<td>Baseline (Uni) + UFL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>0.361</td>
<td>0.289</td>
<td>0.325</td>
</tr>
<tr>
<td>Baseline (Uni) + UFL + AA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>0.353</td>
<td>0.287</td>
<td>0.320</td>
</tr>
<tr>
<td>Baseline (Uni) + UFL + AA + FGT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>0.352</td>
<td>0.279</td>
<td>0.317</td>
</tr>
</tbody>
</table>

Table 3. Ablation results on DTU evaluation set. “AA” and “FGT” refer to adaptive aggregation and finer ground-truth respectively. “Baseline (Reg)” is the original CasMVSNet [9]. We set the confidence threshold and the consistent views to 0.3 and 3 for all models.

4.3. Ablation Studies

As aforementioned, we adopt some extra strategies (e.g., adaptive aggregation and finer ground-truth) that have been adopted by recent methods [30, 38] to train our model for a fair comparison with them. However, this may not be fair to those methods inherited only from MVSNet. In this section, we will prove through extensive ablation studies that even if these strategies are eliminated, our method still has a significant improvement. We use our baseline CasMVSNet [9], whose original representation is regression, as the backbone and changing various components, e.g., depth representation, optimization, aggregation, and ground-truth. And we adopt 5 input views for all models for a fair comparison.

Benefits of Unification. As shown in Tab. 3, significant progress can be made even if purely replacing the traditional depth representation with our unification. Meanwhile, unification is more robust when the hypothesis range of the finer stage doesn’t cover the ground-truth depth. In this case, the target unity of the unification representation generated by Algorithm 1 is all zero, which is a correct supervision signal anyway, and the traditional representation will generate an incorrect supervision signal to pollute the model training. Meanwhile, our Unification can also generate sharp depth on object boundaries like [28].

Benefits of UFL. Applying Focal Loss to our representation can effectively overcome the sample imbalance problem. It can be seen from Tab. 3 that GFL has a huge benefit to accuracy, albeit with a slight loss of completeness. And our UFL can further improve the accuracy and completeness significantly on the basis of GFL. More ablation results about UFL are shown in Supp. Mat.

Regression finetuning and less data. Tab. 4 shows that our

<table>
<thead>
<tr>
<th>Method</th>
<th>Confidence Consistent View</th>
<th>ACC.(mm)</th>
<th>Comp.(mm)</th>
<th>Overall.(mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Uni) + UFL + 50% data</td>
<td>✓</td>
<td>0.372</td>
<td>0.282</td>
<td>0.327</td>
</tr>
</tbody>
</table>

Table 4. Some other Ablation results.

Unification performs better and is more concise compared to other strategies. Meanwhile, we still achieve excellent performance even with only 50% of the training data.

5. Conclusion

In this paper, we propose a unified depth representation and a unified focal loss to promote the effectiveness of multi-view stereo. Our Unification can recover finer 3D scene benefits from the direct learning cost volume, and UFL is able to capture more fine-grained indicators for rebalancing samples and deal with continuous labels more reasonably. What’s more valuable is that these two modules don’t impose any memory or computational costs. Each plug-and-play module can be easily integrated into the existing MVS framework and achieve significant performance improvements, and we have shown this through our UniMVSNet. In the future, we plan to explore the integration of our modules into stereo matching or monocular field and look for more concise loss functions.

Acknowledgements. Thanks to the National Natural Science Foundation of China 62072013 and U21B2012, Shenzhen Research Projects of JCYJ20180503182128089 and 201806080921419290, Shenzhen Cultivation of Excellent Scientific and Technological Innovation Talents RCJC20200714114435057, Shenzhen Fundamental Research Program (GXWD20201231165807007-20200806163656003). In addition, we thank the anonymous reviewers for their valuable comments.
References


