Group 9 Final Project Proposal

Optimization to the Unification & Unified Focal Loss of UniMVSNet

311605004 劉子齊、311553014 張亭萱

I. Goal

Some of the parts or mathematical calculations of the original UniMVSNet were produced by mistake based on the works of the creator of the present Unified MVSNet. To improve the original Unified MVS Net, we suggested a method that was optimized.

In this research, we will modify our baseline, the UniMVSNet, to emphasize unity generation, unity regression, and unified focal loss in an effort to improve the performance of computer vision's depth estimation.

II. Related Work

A. Traditional MVS methods

- **Volumetric based**

  Volumetric approaches partition the three-dimensional space into regular grids and then estimate whether or not each voxel adheres to the surface. This format has two drawbacks: space discretization inaccuracy and excessive memory consumption.

- **Point cloud based**

  Point cloud-based approaches depend on the propagation strategy to gradually densify the reconstruction and act directly on 3D points. As the propagation of point clouds occurs sequentially, these approaches are difficult to parallelize and often need a considerable amount of time to complete.

- **Depth map based**

  In contrast, the depth map is the most adaptable of all the representations. It decouples the large MVS problem into relatively

tiny per-view depth map estimate problems, concentrating simultaneously on only one reference and a few source pictures.

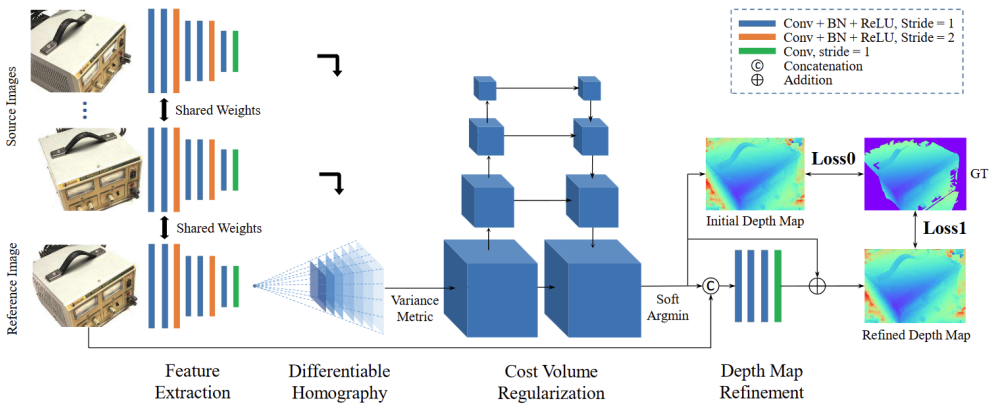
In addition, depth maps may be readily merged with the point cloud or volumetric reconstructions, and the mesh can be rebuilt further.

B. MVSNet & R-MVSNet

MVSNet introduced a deep architecture for depth map estimation, which substantially improves the reconstruction's accuracy and quality.

Cost volume regularization is one of the most significant advantages of learning-based MVS, with a large percentage of networks employing multi-scale 3D CNNs to regularize the 3D cost volume. However, this phase is exceedingly memory costly: it acts on three-dimensional volumes, and its memory requirements expand cubically with model resolution.

R-MVSNet is a recurrent neural network-based variant of MVSNet. The proposed network is based on the MVSNet architecture. However, the cost volume is progressively regularized using the convolutional gated recurrent unit (GRU) rather than 3D CNNs. With sequential processing, the algorithm's online memory needs are lowered from cubic to quadratic to the model resolution. Consequently, R-MVSNet is applicable to high-resolution 3D reconstructions with limitless depth-wise resolution.

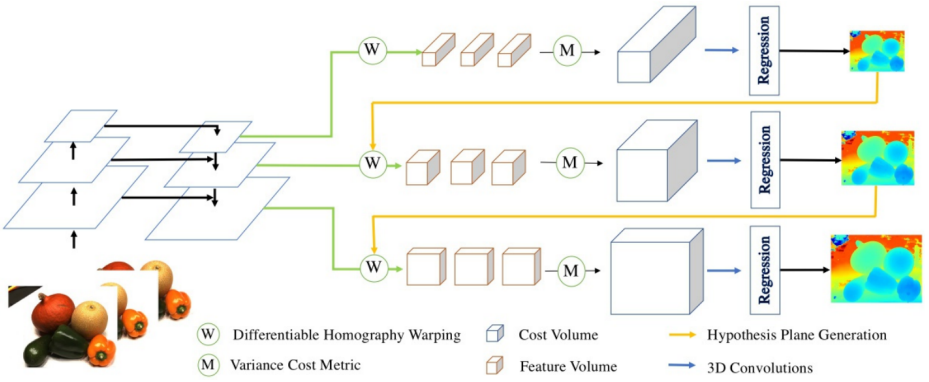


C. Cascade MVSNet

The cascade MVSNet presents a memory- and time-efficient formulation of 3D cost volumes that is supplementary to existing multi-view stereo (MVSNet) and stereo matching techniques based on 3D cost volumes.

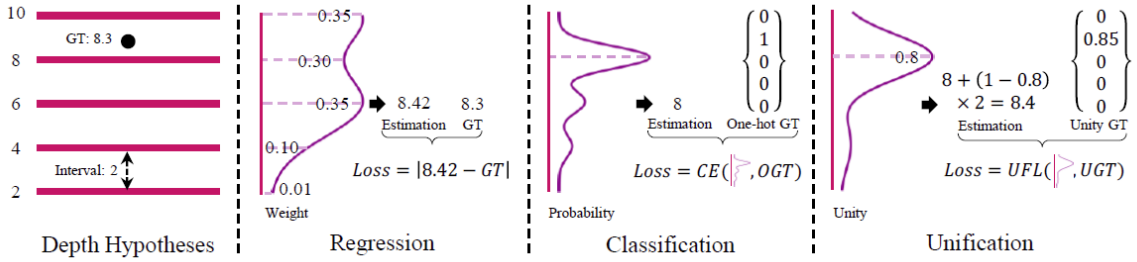
Initially, the suggested cost volume is based on a pyramid of features that encodes geometry and context at progressively smaller sizes. Then, it can narrow each stage's depth (or disparity) range based on the forecast of the prior stage.

The output is recovered from coarse to fine using ever more expensive volume resolution and adaptive modification of depth (or disparity) intervals.



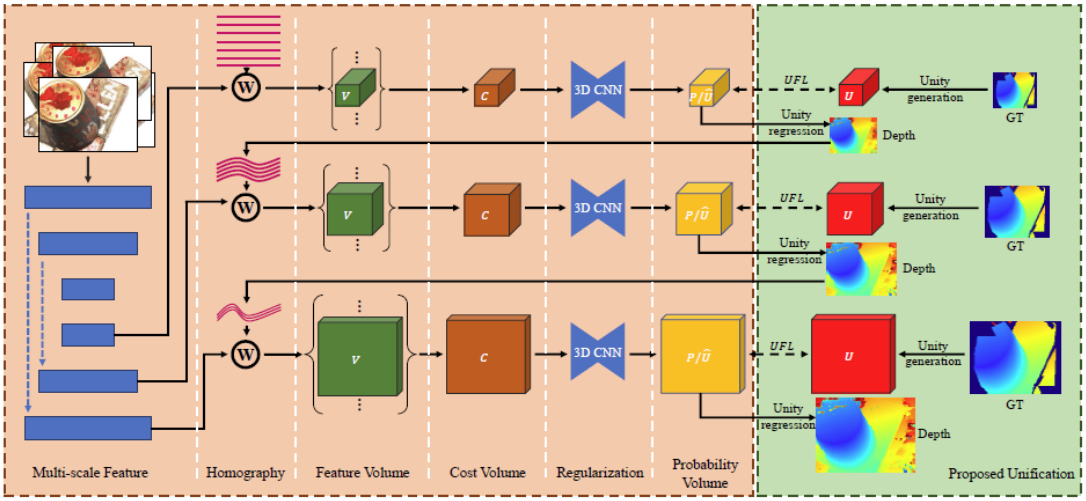
D. UniMVSNet

Regression and classification are two learning-based multi-view stereo methods. Although these two representations have recently demonstrated their excellent performance, they still need to improve, 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 their discrete prediction.



UniMVSNet offers a unique representation known as Unification to combine the benefits of **regression** and **classification**. It may directly confine the cost volume, similar to classification techniques, and forecast sub-pixel depth, similar to regression methods.

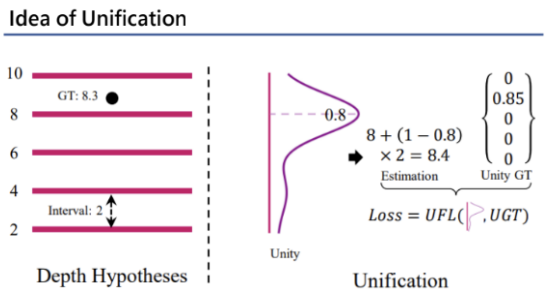
To uncover the potential of unification, it also created a new loss function, dubbed Unified Focal Loss, which is more consistent and rational in its approach to the sample imbalance difficulty.



III. Proposed Method

A. Optimization to the Unification

Originally, the predicted depth of the model must be larger than the previous depth interval before it can be predicted correctly. If the value of ground truth is small today, the prediction will be biased, which will lead to the situation that the depth cannot be accurately determined. Therefore, we proposed the optimized unity generation and unity regression to change the way we encode the ground truth, which makes the encoded value lie between 0.5 and 1, and this brings a solution to this problem.



Original Unity Generation & Regression

$$U_i^{x,y} = 1 - \frac{D_{gt}^{x,y} - d_i^{x,y}}{r} \qquad off = \left(1 - \hat{U}_o^{x,y}\right) * r$$

Optimized Unity Generation & Regression

$$U_i^{x,y} = 1 - \frac{D_{gt}^{x,y} - d_i^{x,y}}{2 * r} \qquad off = \left(1 - \hat{U}_o^{x,y}\right) * (2 * r)$$

B. Optimization of the Unified Focal Loss

Carrying on from the previous section, we now solve the problem that the relative error is too large when the ground truth is too small. We also suggest our optimal unified focal loss, which is depicted on the right side of the picture below. We want to utilize this enhanced version to differentiate between complicated and simple samples using a more straightforward relative error.

Original Unified Focal Loss

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

Optimized Unified Focal Loss

$$UFL(u,q) = \begin{cases} \alpha^{+}T_1\left(\frac{|q-u|}{q^{+}}\right)^{\gamma} BCE(u,q), & q > 0 \\ \alpha^{-}T_2\left(\frac{u}{q^{+}}\right)^{\gamma} BCE(u,q), & else \end{cases}$$

IV. Result

We've run numerous tests, the lists below are our experiment results. The first thing we do is reduce the training epoch from 16 to 6, which cuts the training time in half from 80 to 30 hours due to time restrictions. Due to VRAM limitations on the RTX-2080Ti's (12G) VRAM for training, we also decreased the batch size from 2 to 1 and the number of views from 5 to 3. The outcome we retrained using these hyperparameters is the baseline result.

Our version of unity generation produced a result that was 71% better than when we originally utilized it. The result is significantly better when we utilize the optimized form of unity regression, reducing by 1.5% over the starting value (UniMVSNet with optimized unity generation). The best result we obtained from these experiments comes from removing the sigmoid function from the unified focal loss in optimized UniMVSNet to speed up the model's convergence.

To speed up convergence, we also reduce the range of the unified focal loss. But the outcome wasn't much better. One probable explanation is that when we reduced the difference between the losses, we shortened the range of the unified focal loss, which made it more challenging to converge.

And since we achieved a superb performance on the optimized unified MVSNet after only six training epochs, we decided to add another six to see if we could beat the SOTA (overall 0.315, made by UniMVSNet). We, unfortunately, failed to arrive. We may have a problem because we did not alter additional hyperparameters during training. We earmarked this problem

for later work. We think that if the model is improved, we will be able to outperform UniMVSNet.

Finally, we conduct some tests on changing the activation function. One is ReLU, and the other one is Leaky ReLU. Both functions improved the result significantly but didn't beat our optimized UniMVSNet without sigmoid. Among them, the performance of Leaky ReLU was not as good as ReLU, possibly because the calculation method of Leaky ReLU caused a slower convergence speed, resulting in lower performance compared to ReLU.

	Baseline	Baseline + OUG	Opti. Baseline	Opti. Baseline w/o Sigmoid	Opti. Baseline w/o Sigmoid in range [1, 3]	Opti. Baseline w/o Sigmoid (12 epoch)	Baseline w/ ReLU	Baseline w/ Leaky ReLU
Acc.	1.6048	0.4623	0.419	0.3886	0.4179	0.6287	0.3915	0.4175
Com.	1.1208	0.3322	0.3283	0.3229	0.3472	0.4612	0.4722	0.5392
Overall	1.3628	0.3973	0.3737	0.3557	0.3826	0.5449	0.4318	0.4783

V. Conclusion

We outperformed UniMVSNet in an early stage of training, as the results above demonstrate. Even though we weren't able to outperform UniMVSNet in 12 epochs, this model drastically cuts down on training time, especially for models with more views. It is therefore easier to employ in actual practice.

The faster training pace of optimized UniMVSNet has an advantage over other models because most models today, including this one, have already attained an error level in depth estimate that is acceptable for the application.

Additionally, altering the activation function can boost the model's performance, though it couldn't match ours. While we did not change any further hyperparameters, there may yet be the possibility for performance enhancement after tuning. We'll take this into account for a future project.