V-FUSE: Volumetric Depth Map Fusion with Long-Range Constraints

Nathaniel Burgdorfer Philippos Mordohai
Stevens Institute of Technology

Abstract

We introduce a learning-based depth map fusion framework that accepts a set of depth and confidence maps generated by a Multi-View Stereo (MVS) algorithm as input and improves them. This is accomplished by integrating volumetric visibility constraints that encode long-range surface relationships across different views into an end-to-end trainable architecture. We also introduce a depth search window estimation sub-network trained jointly with the larger fusion sub-network to reduce the depth hypothesis search space along each ray. Our method learns to model depth consensus and violations of visibility constraints directly from the data; effectively removing the necessity of fine-tuning fusion parameters. Extensive experiments on MVS datasets show substantial improvements in the accuracy of the output fused depth and confidence maps. Our code is available at https://github.com/nburgdorfer/V-FUSE

1. Introduction

Much like other areas of computer vision, Multi-View Stereo (MVS) has benefited from the advent of deep learning. Progress has been driven by the creation of end-to-end systems, unifying all aspects of the MVS pipeline, and by replacing heuristics in the components of the pipeline with optimized network modules. An aspect of MVS that requires further investigation is depth map fusion, which is still implemented as a sequence of heuristic operations.

Considering that the top performing MVS systems in terms of geometric accuracy[1] use depth map collections as the representation, depth map fusion can be a crucial step for obtaining the final 3D reconstruction of the scene. As has been shown by conventional fusion research [23], fusing depth maps, guided by geometric constraints, improves the precision of correct depth estimates by blending them with supporting estimates for the same part of the surface, detects and removes outliers, and reduces redundancy in the final 3D model. Current deep MVS approaches [3, 11, 21, 25, 42, 40, 41], however, bypass depth map fusion and proceed directly to *filtering fusion*, which includes various heuristic post-processing steps to obtain a global point cloud by filtering the point cloud reconstructed from the set of depth maps. This approach has been successful; however, without depth map fusion, not all geometric information from the scene is utilized. Our motivation in this work is to build an end-to-end fusion network that can generate much more accurate depth and confidence maps.

Figure 1. Point cloud reconstructions from DTU [1] and Tanks & Temples [19] datasets using depth maps from NP-CVP-MVSNet [40] and UCSNet [3] as input to V-FUSE.

Filtering fusion that operates on local 3D neighborhoods is unable to leverage relationships among distant surface primitives, such as a surface being occluded from a faraway object. Similarly, convolution networks have a limited receptive fields and can only reason about local interactions. We present V-FUSE, an approach that allows a 3D convolutional network to benefit from such geometric information, in a differentiable manner, controlled by learnable hyperparameters.

V-FUSE considers three types of constraints, inspired by the work of Merrell et al. [23]: *support* among consistent depth estimates across multiple views, *occlusions* and *free-space violations* that provide evidence against depth estimates contradicting surfaces estimated in different depth maps. Free-space violations provide the added benefit of

[1] NeRF [26] has inspired a vastly expanding class of algorithms that produce superior results in view synthesis, but not in 3D reconstruction. We consider NeRF a separate line of work from MVS.

encoding conflicts with respect to surfaces that may be invisible in the frame of the reference camera. There are three substantial differences between our approach and that of Merrell et al.: (i) theirs operates in $2\frac{1}{2}$D while ours operates in a 3D volume, (ii) their algorithms make decisions per pixel without considering context, and (iii) all parameters in our approach are learned end-to-end. Specifying visibility constraints in the fusion volume allows V-FUSE to reason based on interactions among depth estimates along the rays, as well as spatially among neighboring voxels. In the absence of these constraints, only the latter would have been possible via 3D convolutions, which cannot reason about long-range conflicts.

Reducing the storage and computational requirements of deep MVS networks is a necessity for increasing the resolution and quality of 3D reconstruction. 3D convolutional networks operating on cost volumes are forced to downsample high resolution inputs. Since our framework is also volumetric, we propose a technique for achieving high resolution near the surfaces while keeping memory requirements manageable. Specifically, we learn to generate a per-pixel, narrow depth search window by examining the input depth and confidence estimates. Unlike previous networks that iteratively refine the depth search space, our framework leverages the availability of input depth and confidence estimates to determine a reduced search space in a single pass.

Our main contributions are:

- An end-to-end learning-based method for the fusion of depth and confidence maps, leveraging long-range, volumetric visibility constraints encoded into a *visibility constraint volume (VCV)*.
- A pixel-wise search window estimation sub-network to refine the depth search space.

We provide extensive evaluation of V-FUSE on MVS benchmarks [1, 19, 43], using 2D and 3D error metrics.

2. Related Work

In this section, we review related work on learning-based MVS, as well as conventional and learning-based depth map fusion.(Unfortunately, no recent surveys on these topics are available, to the best of our knowledge.)

The combination of deep learning and plane-sweeping stereo has inspired a new generation of MVS algorithms. The plane-sweeping volume (PSV) [10] allows the use of cost aggregation and disparity estimation techniques developed for binocular stereo [18] in multi-view settings. The first deep learning-based plane-sweeping algorithm was DeepStereo [8] that addresses view synthesis in a self-supervised manner. Supervised formulations targeting depth map estimation are largely influenced by MVSNet [42] and concurrent work [14, 15]. We also adopt the PSV structure in this work for our fusion volume.

Several methods [3, 11, 21, 25, 35, 40, 41, 44] aim to improve memory efficiency in deep MVS through multi-resolution, iterative schemes that refine the depth search space with each increase in resolution. This is achieved via regular incremental reductions in search range [11, 41], or with a range set using only confidence estimates [3]. We have developed a non-iterative method for estimating per-pixel depth search windows based on information extracted from the distribution of input depth and confidence maps.

Recent work has addressed MVS by: combined classification and regression for depth estimation [29, 34], sequential depth interval selection [32], an adaptation of RAFT (Recurrent All-Pairs Field Transforms) [22], operating over adaptive intervals along epipolar lines instead of discrete depths [21], and the use of a non-parametric depth distribution model to mitigate shortcomings of unimodal depth models [40]. Transformers for MVS [6, 12, 33, 36] leverage the intra- and inter-attention mechanisms to achieve more accurate feature matching than previous architectures.

Conventional Depth Map Fusion Conventional fusion methods reduce errors and inconsistencies in MVS pipelines. Merrell et al. [23] propose two algorithms for fusing depth maps by selecting depth estimates with large degrees of support from the input depth maps that outweigh violations of visibility constraints. We employ similar constraints, but in a volumetric formulation while the conventional approach [23] reasons on $2\frac{1}{2}$D depth maps. Hu and Mordohai [13] extend the aforementioned method [23] by modeling geometric uncertainty, in addition to confidence, and by considering multiple depth candidates per pixel.

A popular choice for fusing depth maps among deep MVS pipelines is the work of Galliani et al. [9]. It is based on the projection of depth estimates onto several supporting depth maps to accumulate consensus subject to criteria on reprojection error and surface normal inconsistency. The dense COLMAP pipeline [31] also includes a fusion module that rejects outliers based on lack of photometric and geometric support and clusters inliers. Both techniques require setting several thresholds and parameters, and are limited to filtering depth maps into a final 3D model without improving the underlying depth maps.

Some deep MVS systems introduce custom fusion and filtering steps which are not included in the end-to-end trainable pipeline. These include P-MVSNet [20] that considers pixel and depth reprojection errors, and D^2HC-RMVSNet [39] that includes geometric consistency scores. Instead of relying on filtering and averaging depth estimates, our work aims to refine and fuse depth maps before they are filtered and projected into a point cloud.

Learning-Based Depth Map Fusion Most learning-based fusion methods follow the seminal work of Curless and Levoy [5] and adopt a volumetric representation of the truncated signed distance function (TSDF). Learning-based ap-

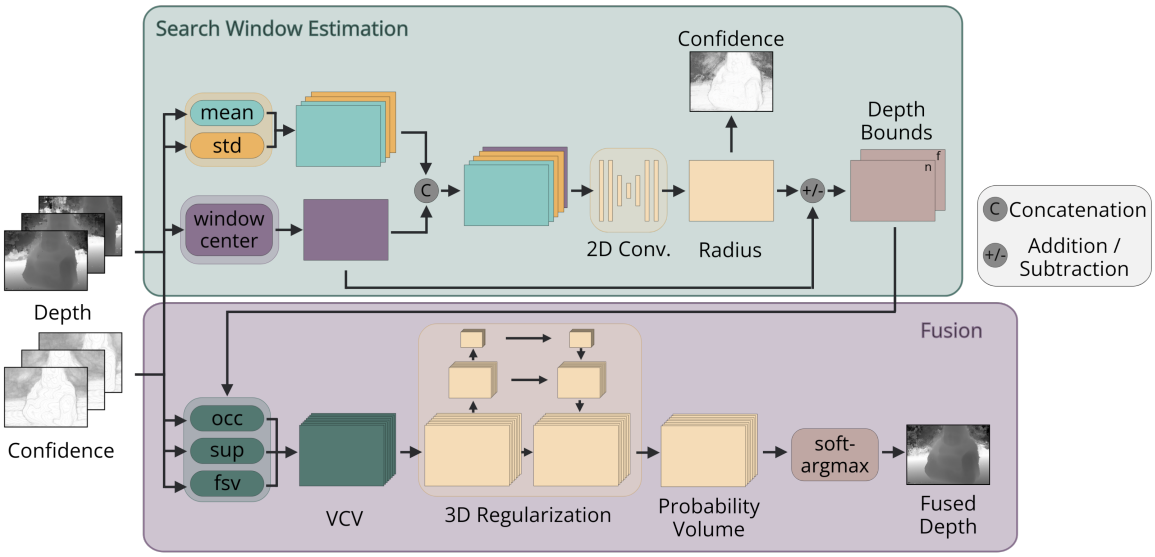


Figure 2. Overview of the V-FUSE architecture. The network is split into two major sub-networks: *Search Window Estimation (SWE)* and *Fusion*. Both sub-networks take in a set of depth maps and confidence maps for a given set of camera views. The SWE sub-network is responsible for estimating a refined depth search window on each ray of the reference camera. The Fusion sub-network uses these refined depth hypotheses to build a *visibility constraint volume (VCV)*, encoding long-range, volumetric visibility constraints into each voxel of the VCV. After passing this volume through a convolutional network and a soft-argmax operator, we regress the final fused depth map.

proaches relying on implicit representations [2, 24, 27, 30] model surfaces as continuous decision boundaries of a deep classifier, and are thus ill-suited for open scenes like the ones we reconstruct.

Recent methods [4, 37, 38] propose fusing streams of input depth maps by learning TSDF volume updates [37], by fusing in the latent space and learning a translator to produce a final TSDF volume [38], or by learning pose invariant scene volumes jointly with a MVS sub-network [4]. Volumetric methods suffer from large storage requirements. Our method is volumetric but allows for a very thin volume in the direction of the optical axis.

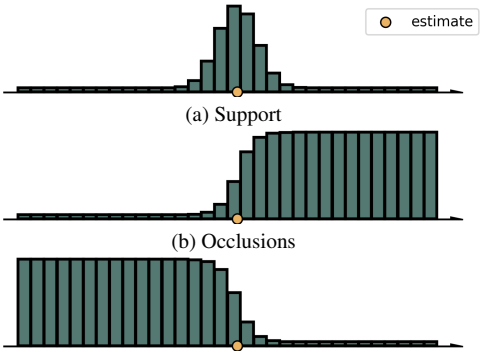
Donné and Geiger [7] developed a non-volumetric data-driven approach for fusing depth maps, estimated conventionally or by a learning-based technique. DeFuSR filters out wrong depth estimates, but also improves correct ones via refinement and inpainting sub-networks. It operates on re-projected depth estimates and image features at high resolution, depending entirely on 3D convolutions to reason about consensus.

3. Method

In this section, we introduce the architecture of V-FUSE (an overview can be found in Figure 2). Our network takes as input a reference depth map $D_0 \in \mathbb{R}^{HxW}$ and the corresponding confidence map $C_0 \in \mathbb{R}^{HxW}$, and $N-1$ source depth and confidence maps $\{D_v\}_{v=1}^{N-1} \in \mathbb{R}^{HxW}$ and $\{C_v\}_{v=1}^{N-1} \in \mathbb{R}^{HxW}$ respectively. We begin by rendering the input source maps into the reference view to obtain $\{D_v^{ref}\}_{v=1}^{N-1}$ and $\{C_v^{ref}\}_{v=1}^{N-1}$. With this set of rendered maps, we build a *visibility constraint volume (VCV)*, whose structure follows that of the plane-sweeping volume. The VCV encodes long-range, volumetric, constraints at each voxel. We use a 3D convolutional network to regularize the VCV and regress the final fused depth map. As output, our network produces fused depth and confidence maps for the reference view, D^f and C^f. The construction of the VCV and the 3D convolutional network are supervised using an l_1 loss between the estimated and ground truth depth maps. For memory and run-time efficiency, we introduce a novel *search window estimation (SWE)* sub-network in order to restrict the depth search space used as input in the construction of the VCV. This sub-network is supervised through a novel loss that we discuss in Section 4.2.

3.1. Visibility Constraint Volume

Similar to most deep MVS frameworks, a core component of our network is the construction of a cost volume along the reference camera frustum. However, instead of encoding warped image features, our volume encodes visibility constraints for the purpose of measuring multi-view depth estimate consensus and inconsistency. Specifically, we compute three separate metrics measuring support, occlusions, and free-space violations, and aggregate each metric into separate channels in the VCV. Essentially, each voxel is a collection of the response values for all three constraints from each input view. The constraints are aggregated over all views, each view contributing equally (without favoring the reference view) up to a confidence weighting. We discuss each constraint in detail below. Figure 3



(a) Support

(b) Occlusions

(c) Free-Space

Figure 3. Visualization of each constraint response curve. Support and occlusions are encoded along the ray of the reference camera, while free-space violations are encoded along the ray of each source camera. The constraint curves are centered around each depth estimate. The support response activates near the depth estimate (encoded as a Gaussian). The occlusion response activates past the depth estimate and the free-space response activates before the depth estimate (both encoded as sigmoids).

shows a visualization of the three constraints.

The network takes as input a set of M initial depth hypotheses $h \in \mathbb{R}^M$. This is the set of depth values measured along the ray of the reference camera. We define p to be a given pixel index and q to be the corresponding voxel index at the d^{th} hypothesis. Using the set h, we build a hypothesis volume $S \in \mathbb{R}^{HxWxM}$ by tiling h at every pixel, meaning that each ray uses the same set of depth hypotheses. Using S, we can compute a 3-channel VCV, $V \in \mathbb{R}^{HxWxMx3}$.

Support We first compute the support response for each voxel in the VCV. Intuitively, support is an encoding of the multi-view depth consensus for the rendered depth maps. For a given voxel, the higher the support response, the more probable the true depth value exists at that voxel. For support, we employ a Gaussian distribution centered at each depth estimate in the rendered depth map D_v^{ref} for each view, encoded along the ray of the reference view.

$$V_{q,0} = \frac{1}{K} \sum_{v=0}^{K-1} C_{v,p}^{ref} \exp\left(\frac{-(S_q - D_{v,p}^{ref})^2}{2\sigma_p^2}\right) \quad (1)$$

Here, $V_{q,0}$ is the support response for the depth hypothesis at voxel S_q. The subscript 0 is used to indicate that the support response is encoded in the first channel of V_q. The confidence $C_{v,p}^{ref}$ rendered into the reference view for view v and pixel p is used to weigh the support response. The standard deviation for the Gaussian distribution σ_p is a function of the per-pixel window radius (discussed in Section 3.3), which allows the level of support to vary depending on the size of the search window. The formulation of σ_p (see supplement) includes a learned hyper-parameter in order for the support window to be learned from the data. Lower values of σ_p correspond to a sharper response boundary.

Note that due to perspective distortion and due to some 3D points projected from the source depth maps falling out of bounds, a pixel of the reference view may receive fewer than N rendered depths and confidences. Therefore, we define $K \leq N$ to be the number of views that provide a response for the depth hypothesis at voxel S_q.

Occlusions To identify conflicting depth estimates, we include occlusion and free-space violation responses as separate channels in our VCV. Occlusions are events in which the reference depth hypothesis at voxel S_q is farther away from the reference camera than the rendered supporting depth estimate $D_{v,p}^{ref}$. To encode occlusions, we use a sigmoid computed along the ray of the reference view. The sigmoid is centered at each depth estimate in the rendered depth map D_v^{ref} for each view and activates behind the estimate. In this way, the response for occlusions contributes a sigmoid response to V_q with magnitude depending on the difference in depth. The response is high for depth hypotheses that are beyond each estimate and low for hypotheses that are in front of each estimates.

$$V_{q,1} = \frac{1}{K} \sum_{v=0}^{K-1} C_{v,p}^{ref} \frac{1}{1 + \exp\left(-\lambda_p (S_q - D_{v,p}^{ref})\right)} \quad (2)$$

Here, $V_{q,1}$ is the occlusion response for the depth hypothesis at voxel S_q encoded into the second channel of V_q. The input confidence values are used to weigh the occlusion responses. The multiplier λ_p is a function of the per-pixel window radius and is used to adjust the slope of the sigmoid function. The definition for λ_p (see supplement) also includes a learned hyper-parameter so that the slope of the sigmoid is learned from the data.

Free-Space Violations In contrast to support and occlusion, free-space violations are measured with respect to the source views. They occur when a depth hypothesis S_q^v (rendered into the source view v) is closer to the source camera than the depth estimate from the original, non-rendered source depth map $D_{v,p}$. In this context, we state that the depth hypothesis S_q^v violates the free-space of depth estimate $D_{v,p}$. Much like occlusions, we use a sigmoid to encode free-space violations. The sigmoid function is defined along the ray of projection of the original depth map D_v for each view and activates in front of the depth estimates, contributing a sigmoid response to V_q with magnitude depending on the difference in depth.

$$V_{q,2} = \frac{1}{K} \sum_{v=0}^{K-1} C_{v,p} \frac{1}{1 + \exp\left(-\lambda_p (D_{v,p} - S_q^v)\right)} \quad (3)$$

Here, $V_{q,2}$ is the free-space violation response for the depth hypothesis at voxel S_q encoded into the third channel of V_q.

The response values are weighted by the original confidence values $C_{v,p}$ for view v and pixel p. The multiplier λ_p is the same parameter used in the encoding of occlusions.

3.2. Evidence Aggregation and Depth Estimation

In order to aggregate neighboring information, we regularize the VCV using a 3D UNet similar to MVSNet [42]. This includes several layers of 3D convolutions with downsampling and skip-connections to incorporate global context in the latent space, producing a probability volume P. We then apply a soft-argmax operator along the depth dimension. The final fused depth map is generated using the depth-wise expectation of probabilities for each depth hypothesis,

$$D_p^f = \sum_d S_{p,d} P_{p,d} \tag{4}$$

Here, we write $S_{p,d}$ and $P_{p,d}$ using explicit index notation instead of S_q and P_q to clearly indicate the reduction over the depth dimension.

3.3. Dynamic Depth Search Windows

As input, our VCV construction process takes a set of depth hypotheses per ray. Instead of a single constant set of hypotheses for all rays, we aim to formulate a hypothesis set per ray that is learned from the data. For the sake of run-time and memory efficiency, it is important to limit the number of depth hypotheses. Therefore, we look to reduce the search space while maintaining a high probability that it encompasses the true depth.

Our Search Window Estimation (SWE) sub-network takes as input the N rendered depth and confidence maps. We compute the mean and standard deviation of both the depth and confidence maps per-pixel. Similar to the formulations of the constraints, we average these metrics over the set of valid inputs $K \leq N$. In order to center the search windows around an initial value D^{center}, we use the input confidence values to select the most confident depth estimate from the N input views. See the supplement for the motivation behind this choice.

The input to our search window estimation sub-network is the concatenation of the pixel-wise depth and confidence statistics with D^{center}. We run this 5-channel feature map through several 2D convolutional layers, followed by a sigmoid activation function. The output is used for estimating the search window radius,

$$R_p = r_{min} + r_{max} O_p \tag{5}$$

where R_p is the window radius at pixel p, $r_{min} = \psi_{min}(b_{max} - b_{min})$ and $r_{max} = \psi_{max}(b_{max} - b_{min})$ are the minimum and maximum allowable bound for the window radius respectively, and O_p is the output of the 2D convolutional network at pixel p. The scalars ψ_{min} and ψ_{max} are used to select a percentage of the full input hypothesis range $(b_{max} - b_{min})$ as the minimum and maximum allowable search window radii. These parameters are in place to prevent the network from estimating extreme radius values.

Using this estimated window radius, we can define the depth hypothesis bounds centered around the initial window center estimates.

$$B_p^{min} = D_p^{center} - R_p \tag{6}$$

$$B_p^{max} = D_p^{center} + R_p \tag{7}$$

Here, B_p^{min} and B_p^{max} are the minimum and maximum depth bounds defining the search window at pixel p. We then interpolate between these new bounds to obtain M depth hypotheses, $S_p = [B_p^{min}, ..., B_p^{max}] \in \mathbb{R}^M$. The new hypothesis volume S, with per-pixel hypotheses sets, is then used to build the VCV as described in Section 3.1.

4. Loss Function

We train our network in a supervised manner on the output depth and confidence maps of MVS frameworks. We formulate two loss functions, one for each sub-network.

4.1. Depth Regression Loss

We specify the depth regression loss as the l_1 loss between the estimated fused depth maps D^f and the ground truth depth maps D^{gt}.

$$L_d = \sum_{p \in \Omega_p} |D_p^f - D_p^{gt}| \tag{8}$$

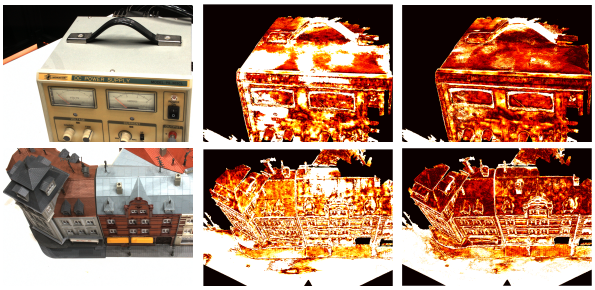
Here, Ω_p is the set of all valid pixels where ground truth depths are available. This loss is mainly used to supervise the construction of the VCV and the regularization network; however, there are no barriers in place to prevent back-propagation through the SWE sub-network. That being said, it is not sufficient to rely on the regression loss to supervise our SWE sub-network.

4.2. Depth Search Window Loss

In order to supervise the SWE network module, we formulate two objective functions. The first term, named the *coverage loss*, penalizes estimated search windows that do not encompass the ground truth depth.

$$L_c = \sum_{p \in \Omega_p} \frac{|D_p^{center} - D_p^{gt}|}{R_p} \tag{9}$$

Using the coverage loss in isolation would not prevent the network from learning to simply maximize the window radius. Therefore, as a regularizing term, we add the magnitude of the window radius to the joint loss function.



(c) Image (b) Input Error (b) Fused Error

Figure 4. Error comparison between the input depth maps from NP-CVP-MVSNet [40] and the fused maps from V-FUSE. The errors are encoded as heat maps, with brighter colors corresponding to higher errors. V-FUSE helps recover from inconsistencies due to texture-less regions, such as the top of the power supply and sections of the roof.

(a) Image (b) Input Depth (c) Fused Depth

Figure 5. Qualitative examples comparing the input depth maps with the fused output depth maps from the DTU [1] dataset using GBi-Net [25] as input. The depth maps generated by V-FUSE improve fine details, texture-less surfaces, and estimates near depth discontinuities.

$$L_r = \sum_{p \in \Omega_p} R_p \tag{10}$$

This term directly penalizes large window radii, guiding the SWE sub-network to produce tight search windows that include the ground truth depth. The formulation of the loss in this manner bears some similarity to the work of Kendall and Gal [17], with the window radii used as a proxy for uncertainty.

Our total loss is the weighted sum of these three objective functions.

$$L = \lambda_d L_d + \lambda_c L_c + \lambda_r L_r \tag{11}$$

5. Experiments

5.1. Datasets

DTU The DTU dataset [1] is an indoor dataset that contains images of 124 scenes taken from a camera mounted on an industrial robot arm. All scenes share the same camera trajectories, with ground-truth point clouds captured via a structured light scanner. We follow the training, validation, and evaluation split used by Yao et al. [42].

Method	DTU				
	MAE↓	$< 0.125 \uparrow$	$< 0.25 \uparrow$	$< 0.50 \uparrow$	$< 1.00 \uparrow$
MVSNet [42]	9.200	9.55	18.67	34.55	55.35
+ Conventional [23]	9.050	13.19	25.57	45.34	65.16
+ V-FUSE	**6.838**	**15.00**	**28.57**	**48.45**	**66.51**
UCSNet [3]	12.071	9.99	19.52	35.90	56.24
+ Conventional [23]	13.633	12.45	24.06	42.59	61.15
+ V-FUSE	**9.667**	**12.85**	**24.85**	**43.96**	**62.69**
NP-CVP-MVSNet [40]	12.897	11.76	23.16	42.92	64.27
+ Conventional [23]	11.933	12.79	25.35	47.42	68.94
+ V-FUSE	**8.566**	**16.47**	**31.49**	**53.06**	**70.58**
GBi-Net [25]	5.845	12.77	24.89	45.10	65.94
+ Conventional [23]	5.009	17.30	32.93	55.52	73.66
+ V-FUSE	**4.196**	**18.41**	**34.79**	**57.50**	**74.66**

Table 1. Quantitative comparison of the 2D depth map errors on the evaluation set of DTU [1] benchmark. All threshold values are measured in *mm*. Conventional fusion improves almost all inputs, even those from recent state-of-the-art methods, in terms of average error over all pixels with ground truth and also by increasing the number of inliers for each threshold. Learned fusion via V-FUSE leads to even larger improvements in *all* cases.

Tanks & Temples The Tanks & Temples dataset [19] is a large-scale, mostly outdoor dataset containing video sequences of challenging scenes. The dataset is divided into a training set and two evaluations sets; intermediate and advanced.

BlendedMVS The BlendedMVS [43] dataset is a large-scale, synthetic dataset containing images processed by blending original images with rendered images from each scene mesh. The dataset is split into training and validation sets, containing 106 and 7 scenes respectively.

5.2. Implementation Details

Training Details We implement the model with PyTorch [28] and train on the output depth and confidence maps of the DTU [1] dataset from several deep MVS methods, separately. For improved generalization, we follow the robust training strategy used by PatchmatchNet [35], in which we randomly choose $N-1$ of the 10 best source views to use as training for a given reference view. We train on an NVIDIA RTX A6000 GPU for 30 epochs. The model has approximately 300,000 parameters and training takes 1 hour per epoch for the high resolution data. We use the Adam optimizer with a learning rate of 0.0003 using an exponential decay of 0.95 every 2 epochs. For additional model parameters, please see the supplement.

5.3. Evaluation

Metrics As our method is focused on generating depth and confidence maps, we mainly focus our evaluation on 2D metrics. For depth map evaluation, we report the *mean absolute error (MAE)* between the estimated and ground truth depth maps. We also report the percentage of pixels with depth estimates within several error thresholds. We also present 3D metrics on output point clouds generated

Method	DTU [Sparse] (mm) ↓			DTU [Dense] (mm) ↓		
	Acc.	Comp.	**Overall**	Acc.	Comp.	**Overall**
MVSNet [42]						
+ Gipuma [9]	**0.396**	0.527	0.462	0.419	0.383	0.401
+ V-FUSE	0.432	**0.390**	**0.411**	**0.388**	**0.349**	**0.368**
UCSNet [3]						
+ Gipuma [9]	**0.338**	0.349	0.344	0.320	**0.261**	0.290
+ V-FUSE	0.354	**0.329**	**0.342**	**0.265**	0.276	**0.270**
NP-CVP-MVSNet [40]						
+ Gipuma [9]	0.356	**0.275**	0.316	0.288	0.194	0.241
+ V-FUSE	**0.337**	0.277	**0.307**	**0.256**	**0.181**	**0.219**
GBi-Net [25]						
+ ∼ COLMAP [31]	0.315	**0.262**	**0.289**	0.254	**0.173**	0.214
+ V-FUSE	**0.310**	0.274	0.292	**0.227**	0.180	**0.204**

Table 2. Chamfer distances (lower is better) of the final fused point clouds from the evaluation set of DTU [1] benchmark. We evaluate the final models using the official script that enforces a sparse minimum point-spacing of 0.2mm (left). Since the errors are approaching this threshold, we also evaluate the models enforcing a dense minimum point-spacing of 0.03mm (right). MVSNet [42], UCSNet [3], and NP-CVP-MVSNet [40] use Gipuma [9] to fuse depth estimates into a final 3D model. GBi-Net [25] uses an adaptation of the fusion approach of COLMAP, in which geometric and photometric filters are used to filter and average consistent depth estimates across views.

from the fused depth maps. We evaluate our point clouds on the DTU benchmark [1], measuring *accuracy*, *completeness*, and *overall* scores. Accuracy is the mean distance between every point in the estimated point cloud to the closest point in the ground truth model and completeness is the mean distance between every point in the ground truth point cloud to the closest point in the estimated model. The overall score is the average of these metrics. We show a variation of these metrics when comparing to DeFuSR [7] following the evaluations performed in their work. Donné and Geiger [7] report the Chamfer distances as the percentage of points within a threshold of $\tau = 2.0mm$. We also evaluate our point clouds on the Tanks & Temples benchmark. We report the *f-score* for each scene, as well as the mean f-score for all scenes.

MVS Baselines We compare the results of applying V-FUSE on the outputs of MVSNet [42], UCSNet [3] as a representative multi-resolution algorithm and two state-of-the-art methods, NP-CVP-MVSNet [40] and GBi-Net [25].

Fusion Baselines We compare the results of V-FUSE with the conventional fusion approach of Merrell et al. [23] for 2D evaluations, and Gipuma [9] for 3D evaluations, since Gipuma is the method of choice by *state-of-the-art* MVS frameworks to produce final 3D models. We also provide comparisons to the learning-based fusion method, DeFuSR [7]. Methods operating on implicit TSDF volumes, such as VolumeFusion [4], RoutedFusion [37], and NeuralFusion [38] are not included in our evaluations, since they are better suited for reconstructing closed, watertight objects. These papers do not provide any quantitative evaluations on DTU or Tanks & Temples, with NeuralFusion presenting qualitative-only results on select scenes from Tanks & Temples.

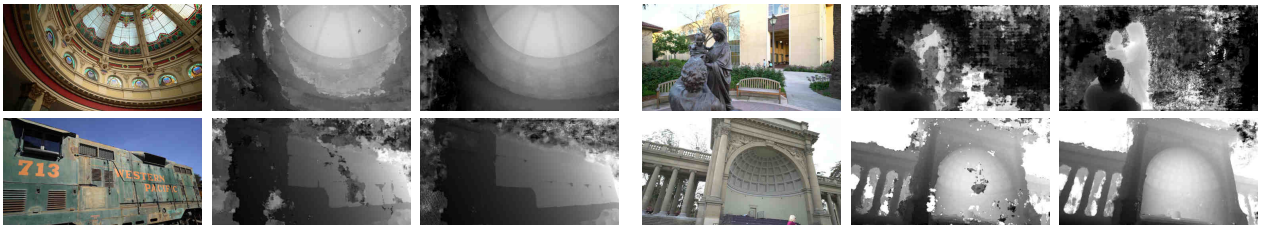
Method	DTU (full) ↑		
	Acc. (%)	Comp. (%)	Mean (%)
MVSNet [42]	88	66	77
+ DeFuSR [7]	86	65	76
+ V-FUSE	**98**	**98**	**98**

Table 3. Chamfer distances (lower is better) of the final 3D models of MVSNet [42] using DeFuSR [7] and V-FUSE for fusion. Here, accuracy and completeness are reported as the percentage of points with accuracy and completeness scores within $\tau = 2.0mm$.

Evaluation on DTU Dataset We first compute ground truth depth maps for DTU in the same manner as MVSNet [42]. Specifically, we run screened Poisson surface reconstruction (SPSR) [16] on the provided ground truth point clouds for each scene and produce a watertight mesh. We then render this mesh into all cameras to obtain ground truth depth maps. To produce our final point clouds, we use heuristic filtering, similar to the post-processing presented in GBi-Net. We first filter out depth estimates that have a confidence value below a threshold. We then project each estimate into neighboring views, using the depth estimates in each view to reproject back to the reference view, measuring the pixel reprojection error and filtering out estimates whose error is above a threshold.

Table 1 shows a comparison of the depth map errors between all baseline methods and V-FUSE. Observing the fused depth map errors, we can see that even using the low resolution inputs of MVSNet, V-FUSE can generate depth maps with a lower MAE than UCSNet and NP-CVP-MVSNet. Additionally, V-FUSE produces depth maps with more inliers at all threshold values compared to the input depth maps generated by all baseline methods. A comparison of error maps is shown in Figure 4. Qualitative depth map results can be seen in Figure 5. We can observe that V-FUSE removes much of the noise in the input depth maps, while producing better estimates near depth discontinuities. In Table 2, we evaluate the final 3D models of all MVS baselines and compare the fusion choice from each method to V-FUSE. V-FUSE shows clear improvements in the *overall* results in both Sparse and Dense evaluation scenarios. In the case of GBi-Net, the improvements realized by V-FUSE are more noticeable without the sampling procedure used in the Sparse evaluation. DeFuSR [7] provides results evaluating fusion of COLMAP [31] and MVSNet [42] inputs on DTU. We provide a comparison according to the evaluation protocol used in [7] in Table 3. The threshold used by DeFuSR is $\tau = 2.0mm$, which is quite large. V-FUSE outperforms DeFuSR by a substantial margin, which is expected as the authors state they are not able to refine the MVSNet inputs much.

Evaluation on Tanks & Temples Dataset We use the model trained on the DTU output depth and confidence maps of each network without any fine-tuning for evaluation. In order to evaluate the depth maps on Tanks & Tem-



(a) Image (b) Input Depth (c) Fused Depth (d) Image (e) Input Depth (f) Fused Depth

Figure 6. Qualitative comparison of depth maps for scenes from the Tanks & Temples benchmark [19] using GBi-Net [25] as input.

Method	intermediate ↑								
	Mean	Fam.	Franc.	Horse	Light.	M60	Pan.	Play.	Train
UCSNet [3]									
+ Gipuma [9]	54.83	**76.09**	53.16	43.03	54.00	55.60	**51.49**	**57.38**	**47.89**
+ V-FUSE	**55.03**	75.64	**57.60**	**46.03**	**54.35**	**55.78**	49.42	56.02	45.37
GBi-Net [25]									
+ ∼ COLMAP [31]	**61.42**	**79.77**	**67.69**	**51.81**	**61.25**	**60.37**	**55.87**	**60.67**	**53.89**
+ V-FUSE	59.08	78.92	65.23	49.96	59.16	57.08	53.13	58.58	50.61

Table 4. F-score (higher is better) of the final fused point clouds from the evaluation sets of the Tanks & Temples [19] benchmark. The best results between the baseline and V-FUSE are marked as bold.

Method	Tanks & Temples			
	MAE↓	$< \tau \uparrow$	$< 2\tau \uparrow$	$< 4\tau \uparrow$
UCSNet [3]	0.175	11.83	19.69	**30.17**
UCSNet + V-FUSE	**0.167**	**12.18**	**19.75**	29.84
NP-CVP-MVSNet [40]	0.177	15.49	**25.16**	37.38
NP-CVP-MVSNet + V-FUSE	**0.155**	**15.68**	25.13	**37.57**
GBi-Net [25]	**0.240**	12.02	19.51	29.24
GBi-Net + V-FUSE	0.243	**12.88**	**20.57**	**30.22**

Table 5. Quantitative comparison of depth map errors on the training set of Tanks & Temples [19]. All methods have been trained on DTU. The threshold value τ is selected per-scene and is derived from the thresholds provided by the benchmark.

ples, we use the training set and the provided ground truth point clouds, computing ground truth depth maps the same way they are computed for DTU. Table 5 shows the depth map errors between each baseline and V-FUSE. The fused depth maps are more accurate overall for UCSNet and NP-CVP-MVSNet. For GBi-Net, we show improved accuracy for estimates within the error thresholds. See Figure 6 for a qualitative comparison of depth maps. Table 4 shows the f-scores for the final point clouds on the Tanks & Temples intermediate set. We show comparable results to both input MVS baselines. We provide the *precision* and *recall* split for each method in the supplement.

Additional Experiments We provide results on the validation set of the BlendedMVS [43] dataset in the supplement. Using the outputs of GBi-Net trained on the BlendedMVS training set and the V-FUSE model trained on DTU without any fine-tuning, V-FUSE produces higher quality depth maps for all scenes, with a mean MAE of *0.288* compared to *0.319* for GBi-Net. We also provide evaluations of the output confidence maps, reporting the *AUC* of all methods. Using GBi-Net as input, the AUC of V-FUSE is *2.480* compared to *3.690* for GBi-Net. We show several ablation studies in the supplement, testing the individual contributions of different aspects of the network architecture. Specifically, we evaluate the contributions of the visibility constraints, as well as the efficiency gains of the SWE sub-network. As detailed in the supplement, introducing the SWE sub-network results in $8.5\times$ memory and $9\times$ run-time efficiency gains, as well as a 20% decrease in MAE.

6. Conclusion

We have presented an end-to-end depth map fusion network that leverages long-range visibility constraints encoded into a learnable pipeline. Our method improves input depth and confidence maps generated by MVS networks, integrating multi-view consensus and inconsistency measures. We also present a novel depth search space refinement sub-network that estimates a narrow search window along each ray to increase memory and run-time efficiency, as well as allow for high resolution depth estimation near surfaces. The combination of these concepts is able to obtain fused depth maps that are quantitatively and qualitatively much better than the inputs. While the depth map fusion in our work is end-to-end, merging the depth estimates into a unified point cloud remains a heuristic-driven process. We aim to incorporate a more principled point cloud reconstruction procedure from a collection of depth maps in future work. We also aim to explore the generalization ability of learning-based fusion.

Acknowledgment This research has been supported in part by the National Science Foundation under award 2024653.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *IJCV*, 2016.

[2] Zhiqin Chen and Hao Zhang. Learning Implicit Fields for Generative Shape Modeling. In *CVPR*, 2019.

[3] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep Stereo Using Adaptive Thin Volume Representation With Uncertainty Awareness. In *CVPR*, 2020.

[4] Jaesung Choe, Sunghoon Im, Francois Rameau, Minjun Kang, and In So Kweon. VolumeFusion: Deep Depth Fusion for 3D Scene Reconstruction. In *ICCV*, pages 16086–16095, 2021.

[5] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images. In *SIGGRAPH*, pages 303–312, 1996.

[6] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global Context-Aware Multi-View Stereo Network with Transformers. In *CVPR*, pages 8585–8594, 2022.

[7] Simon Donné and Andreas Geiger. Learning Non-Volumetric Depth Fusion Using Successive Reprojections. In *CVPR*, 2019.

[8] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to Predict New Views From the World's Imagery. In *CVPR*, 2016.

[9] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *ICCV*, 2015.

[10] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time Plane-sweeping Stereo with Multiple Sweeping Directions. In *CVPR*, 2007.

[11] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching. In *CVPR*, 2020.

[12] Vitor Guizilini, Rareș Ambruș, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-Frame Self-Supervised Depth with Transformers. In *CVPR*, pages 160–170, 2022.

[13] Xiaoyan Hu and Philippos Mordohai. Least Commitment, Viewpoint-Based, Multi-view Stereo. In *3DIMPVT*, pages 531–538, 2012.

[14] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning Multi-View Stereopsis. In *CVPR*, 2018.

[15] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: End-To-End Deep Plane Sweep Stereo. In *ICLR*, 2019.

[16] Michael Kazhdan and Hugues Hoppe. Screened Poisson Surface Reconstruction. *ACM TOG*, 32(3), 2013.

[17] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *NeurIPS*, 30, 2017.

[18] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-To-End Learning of Geometry and Context For Deep Stereo Regression. In *ICCV*, pages 66–75, 2017.

[19] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM TOG*, 36(4), 2017.

[20] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-MVSNet: Learning Patch-Wise Matching Confidence Aggregation for Multi-View Stereo. In *ICCV*, 2019.

[21] Xinjun Ma, Yue Gong, Qirui Wang, Jingwei Huang, Lei Chen, and Fan Yu. EPP-MVSNet: Epipolar-Assembling Based Depth Prediction for Multi-View Stereo. In *ICCV*, pages 5732–5740, 2021.

[22] Zeyu Ma, Zachary Teed, and Jia Deng. Multiview Stereo with Cascaded Epipolar RAFT. In *ECCV*, 2022.

[23] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nister, and Marc Pollefeys. Real-Time Visibility-Based Fusion of Depth Maps. In *ICCV*, pages 1–8, 2007.

[24] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *CVPR*, pages 4460–4470, 2019.

[25] Zhenxing Mi, Chang Di, and Dan Xu. Generalized Binary Search Network for Highly-Efficient Multi-View Stereo. In *CVPR*, pages 12991–13000, 2022.

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, pages 405–421. Springer, 2020.

[27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions For Shape Representation. In *CVPR*, pages 165–174, 2019.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An Imperative Style, High-Performance Deep Learning Library. *NeurIPS*, 32, 2019.

[29] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *CVPR*, pages 8645–8654, 2022.

[30] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional Occupancy Networks. In *ECCV*, pages 523–540, 2020.

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

[32] Christian Sormann, Mattia Rossi, Andreas Kuhn, and Friedrich Fraundorfer. IB-MVS: An Iterative Algorithm for Deep Multi-View Stereo Based on Binary Decisions. In *BMVC*, 2021.

[33] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z Jane Wang, and Rabab Ward. Multi-view 3d reconstruction with transformers. In *ICCV*, pages 5722–5731, 2021.

[34] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. IterMVS: Iterative Probability Estimation for Efficient Multi-View Stereo. In *CVPR*, pages 8606–8615, 2022.

[35] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. PatchmatchNet: Learned Multi-View Patchmatch Stereo. In *CVPR*, pages 14194–14203, 2021.

[36] Xiaofeng Wang, Zheng Zhu, Fangbo Qin, Yun Ye, Guan Huang, Xu Chi, Yijia He, and Xingang Wang. MVSTER: Epipolar Transformer for Efficient Multi-View Stereo. In *ECCV*, 2022.

[37] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R. Oswald. RoutedFusion: Learning Real-Time Depth Map Fusion. In *CVPR*, 2020.

[38] Silvan Weder, Johannes L. Schonberger, Marc Pollefeys, and Martin R. Oswald. NeuralFusion: Online Depth Fusion in Latent Space. In *CVPR*, pages 3162–3172, 2021.

[39] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense Hybrid Recurrent Multi-View Stereo Net With Dynamic Consistency Checking. In *ECCV*, pages 674–689. Springer, 2020.

[40] Jiayu Yang, Jose M. Alvarez, and Miaomiao Liu. Non-Parametric Depth Distribution Modelling Based Depth Inference for Multi-View Stereo. In *CVPR*, pages 8626–8634, 2022.

[41] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost Volume Pyramid Based Depth Inference for Multi-View Stereo. In *CVPR*, 2020.

[42] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth Inference for Unstructured Multi-view Stereo. In *ECCV*, 2018.

[43] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799, 2020.

[44] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-Aware Multi-View Stereo Network. *BMVC*, 2020.