CHOSEN: Contrastive Hypothesis Selection for Multi-View Depth Refinement

Di Qiu *Google XR*
Yinda Zhang *Google XR*
Thabo Beeler *Google XR*
Vladimir Tankovich *Work done while at Google XR*
Christian Häne *Work done while at Google XR*
Sean Fanello *Google XR*
Christoph Rhemann *Google XR*
Sergio Orts Escolano *Google XR*

***Abstract*—We propose CHOSEN, a simple yet flexible, robust and effective multi-view depth refinement framework. It can provide significant improvement in depth and normal quality, and can be integrated in existing multi-view stereo pipelines with minimal modifications. Given an initial depth estimation, CHOSEN iteratively re-samples and selects the best hypotheses. The key to our approach is the application of contrastive learning in an appropriate solution space and a carefully designed hypothesis feature, based on which positive and negative hypotheses can be effectively distinguished. We integrated CHOSEN in a basic multi-view stereo pipeline, and show that it can deliver impressive quality in terms of depth and normal accuracy compared to many other top deep learning based multi-view stereo pipelines.**

***Index Terms*—multi-view stereo, 3D reconstruction, depth refinement.**

I. Introduction

Geometry acquisition through multi-view imagery is a crucial task in 3D computer vision. In the multi-view stereo matching (MVS) framework, image patches or features are matched and triangulated to find 3D points [8], [11]. The dense geometry is usually represented as a depth map for each view. Due to view dependent appearance, occlusion and self-similarity in the real world scenarios, the matching signal is often too noisy to directly give accurate and complete geometry. Traditionally, this problem have been attacked by some carefully designed optimization and filtering scheme [2], [35], directly on the depth map or indirectly inside a matching cost volume, where they impose certain priors on the smoothness of the resulting geometry, making use of the confidence of matching and appearance in a larger spatial context.

With the advancement of deep learning, many research works have combined the ideas with traditional MVS. Early works [43], [44] typically involve extracting convolutional features from the images, building one cost volume (indexed by a sequence monotone depth hypotheses d_i for each pixel), or several in a multi-resolution hierarchy, and transform these cost volumes using 3D convolutions into a "probability volume" so that depth map can be obtained by taking the expectation

$$\hat{d} = \sum_i p_i d_i \tag{1}$$

More recent works have more complicated designs regarding cost computation and aggregation [40], visibility and uncertainty estimation [7], [48], loss functions [30], feature backbones [4], feature fusion [50], and consequently have yielded impressive progress on various benchmarks. The evaluation has been limited to point cloud reconstruction, and little attention was paid to, and thus the lack thereof, the raw output's quality in terms of depth and normal accuracy. Looking closely at the predicted depth and its associated normals, we can often find quantization-like artifacts. Moreover, the fixed training and testing benchmarks also result in limited discussion on the learning and inference task under the variance of the multi-view acquisition setups, including different metric scales, camera relative positioning and lenses (focal lengths). Therefore, for a new capture setup substantially different from the training set, one may need to determine what hyper-parameters to use, especially the number of depth proposals to sample from a certain depth range, and retrain the model of choice to get the optimal performance.

To some extent, the problem can be attributed to the approach of using Eq.(1) to obtain depth estimations: the distribution of depths hypotheses at different pixels, or across different

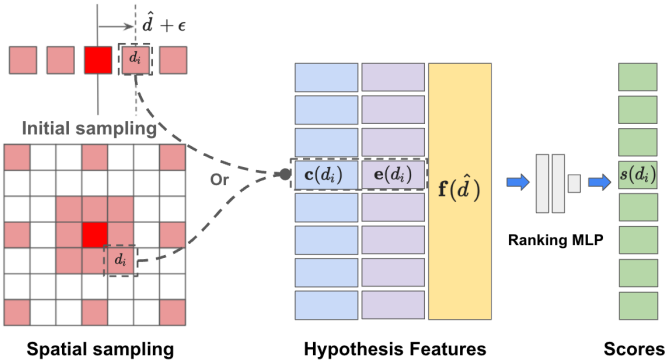

Fig. 1: CHOSEN's hypotheses sampling and ranking mechanism. Assuming an initial depth estimation, we gather depth hypotheses from its perturbations or its spatial neighbors. For each hypothesis, we combine the matching cost, a second order smoothness term and a context feature to form the input of a learned score function represented by an MLP. The one with the highest score will be selected to update refined depth. The process is performed iteratively.

datasets, can be arbitrary. It is thus difficult to guarantee the learned probability weights to produce consistent results: in a smooth spatial neighborhood, small perturbations in the d_i results in non-smoothness in the predicted depth and normal; across different datasets, the inappropriately spaced d_i that are too sparse or too coarse results in poor generalization.

In this work, we turn our attention to the problem of learning to refine depth with multi-view images while keeping in mind that they can be from very different setups, and how to best sample hypotheses for depth refinement. We will discuss in Sec.III-A the proper solution space to perform the learning task, and present an alternative to the approach of Eq.(1) in Sec.III-B. Much as inspired by the PatchMatch [1], [3] methodology, our framework, named CHOSEN, learns to refine the depth by iteratively re-sampling and selecting the best hypotheses, thus eliminating the need of a "probability volume".

In the following we identify and underline the core design elements that enable CHOSEN to produce significantly improved depth and normal quality:

- Transformation of the depth hypotheses into a solution space defined by the acquisition setup.
- The use of first order approximation to sample spatial hypotheses to propagate good hypotheses to their neighborhood.
- A carefully designed hypothesis feature so that it's informative for contrastive learning.
- Each hypothesis is evaluated independently so that the refinement is robust to arbitrary hypothesis samples.

An overview of CHOSEN is illustrated in Fig.1.

To demonstrate the effectiveness of CHOSEN, in Sec.III-C we describe how we embed it in a minimalist MVS pipeline where the only extra learnable part is the feature extraction using light-weight U-Nets [31]. We perform comprehensive ablation experiments to justify our design, and compare the quality of our refined depths with various recent deep learning based MVS pipelines. Without bells and whistles, our baseline model is easy to train, fast to converge, and already delivers impressive depth estimation quality.

II. Related work

MVS algorithms are often categorized by the representation used to reconstruct the output scene, e.g. volume [18], [19], [24], point cloud [11], [16], [25], [26], depth map [13], [33], [38], etc. However, depth map is probably the most flexible and efficient representation among existing ones. While depth map can be considered as a particular case of point cloud representation, (e.g., pixel-wise point), it reduces the reconstruction task to a per-view (2D) depth map estimation. These MVS approaches can be further grouped in hand-crafted (traditional) methods or learning based solutions. Traditional MVS pipelines extend the two-view case [32] by introducing a view-selection step that aggregates the cost from multiple images to a given reference view. The view selection can be performed per-camera [12] or per-pixel [33]. These approaches [12], [13], [33], [38] then rely on well engineered photometric cost functions (ZNCC, SSD, SAD, etc.) to estimate the scene 3D geometry, by selecting the best depth hypothesis that leads to the lowest aggregated cost. However these cost functions usually perform poorly on texture-less or occluded areas, and under complex lighting environments where photometric consistency is unreliable. Hence further post-processing and propagation steps are used to improve the final estimate [12], [33]. We refer readers to [10] for additional details regarding traditional multi-view stereo pipelines.

Recently, MVS algorithms have showed impressive quality of 3D reconstructions in terms of accuracy and completeness, mostly thanks to the increase popularity of deep learning based solutions [4], [14], [40], [43], [44], [46], [50]. These methods often make use of multi-scale feature extractors [6], [7], [36], cost-volumes [20], [21], [49], and guided refinement [21], [29], [43] to retrieve the final 3D estimate. Typically, they leverage U-Nets [31] to build a single or a hierarchy of cost volumes with predetermined depth hypotheses. Then, these cost volumes are regularized using 3D convolutions and the final depth map is regressed from the regularized probability volume. However, to achieve high resolution depth accuracy, it requires sampling a large number of depth planes, which is limited by memory consumption.

Researchers are also opening other frontiers in deep learning based MVS. UCS-Net [7] and Vis-MVSNet [48] use some uncertainty estimate for an adaptive generation of cost volumes. PVSNet [42] and PatchmatchNet [40] learn to predict visibility for each source image. The approach of Eq.(1) is considered critically in [30], although it was only considered from the loss function perspective. GeoMVSNet [50] proposes a geometry fusion mechanism in the MVS pipline. Most recently, transformer based methods [4], [5], [9] exploits the attention mechanism for more robust matching and context awareness. In particular, MVSFormer [4] has combined this approach with the powerful pre-trained DINO features [28].

It is worth mentioning that existing methods usually need to hand-tune hyper-parameters such as the depth range, hypotheses spacing, and number of hypotheses to ensure sufficient and accurate coverage for the new application. Some of these methods adopt 2D convolutional neural networks to obtain final depth estimations, using RGB image to guide depth up-sampling and refinement [15], [21], [40]. Consequently, these methods often generalize poorly to new camera setups or new scenes.

III. Framework

There are two important aspects in our CHOSEN framework. In Sec.III-A we discuss how to define a suitable solution space for the depth hypotheses and how to sample them. This has to accommodate the fact that input depths can be of arbitrary scale, and are computed from different multi-view systems. Then in Sec.III-B we detail our design of a ranking module that is able to process arbitrarily sampled hypotheses, and how it can be trained to effectively and robustly distinguish the good hypotheses from the bad ones.

A. Hypothesis representation and sampling

a) Pseudo disparity representation: We find it crucial to operate in a transformed inverse depth representation, which we call pseudo disparity - akin to disparity for rectified stereo pairs. Denoting the metric depth as D, the pseudo disparity writes

$$d = \frac{f * b}{D} \tag{2}$$

where we choose f to be the focal length (unit in pixels) of the reference camera, and b the metric distance between the reference camera and the *closest* source camera. The scaling factor $f * b$ converts the inverse depth into the *pixel space*, and approximates the correct accuracy level of the capture setup in use, granted that the cameras have sufficient frustum overlap and similar focal lengths. This representation not only allows us to build hypothesis feature insensitive to the variance in metric or intrinsic scales, but also defines the correct granularity in the solution space where we distinguish the positive and negative samples for contrastive learning, thus enabling our depth refinement model to learn and infer across various acquisition setups.

b) Initial hypotheses sampling: The initial depth hypotheses are constructed as a $H \times W \times N$ volume where we will look for the optimal hypothesis per pixel that has the lowest matching cost. In case there is no initial depth available, we initialize the volume using uniformly spaced slices in the range $[d_{\text{min}}, d_{\text{max}}]$, where the number of slices can be chosen so that the slices sit 1-px apart from each other, and initialize the depth $\hat{d}$ to be the one with lowest matching cost. We refer to the corresponding cost volume as the **full cost volume**. In case an previous depth $\hat{d}$ is available, the volume will be restricted to be uniformly spaced in $[\hat{d} - M, \hat{d} + M]$. A cost volume built using this set of hypotheses is referred to as a **local cost volume**, and $\hat{d}$ will be updated to be the one with lowest matching cost in the local volume. In addition, we apply uniform random perturbation in $\mathcal{U}[-0.5, 0.5]$ to each hypothesis at each pixel for both volumes to robustify the training and estimation. Note that there are no other particular restrictions on choices of N or M as long as the spacing is appropriate and the coverage is sufficient, and in particular they can be changed without re-training.

c) Spatial hypothesis sampling: Inspired by the PatchMatch framework [1], [3], we use spatial hypotheses to expand good solutions into their vicinity. Specifically, we sample a set of hypotheses for each pixel by collecting the *propagated* depths from its spatial neighbors. The propagation is facilitated through first order approximation. Denote $\partial d = (\partial_x d, \partial_y d)$ to be gradient of the depth d, $\mathbf{p}' = \mathbf{p} + \Delta\mathbf{p}$ be a spatial neighbor of the pixel $\mathbf{p}$. The propagated depth from $\mathbf{p}'$ to $\mathbf{p}$ is defined as

$$d_{\mathbf{p}' \to \mathbf{p}} = d_{\mathbf{p}'} - \partial d_{\mathbf{p}'} \cdot \Delta\mathbf{p} \tag{3}$$

In practice we use a fixed set of $\Delta\mathbf{p}$'s for each pixel, though it is more of a convenience than requirement. We conduct sampling and best hypothesis selection in multiple iterations and resolutions.

B. Contrastive learning for hypothesis ranking

It is crucial to keep in mind that we *don't assume any structure in the sampled set of hypotheses*. In fact, many among them can be wildly bad samples, especially if the initial depth is too noisy. Hence it implies that each hypothesis should be evaluated independently, and our task is to learn to distinguish the good hypotheses from the bad ones.

To achieve our objective, we designate the ranking model to be a small MLP that takes a **hypothesis feature** and its **context feature**, and outputs a score for the input hypothesis. Among all the hypotheses at a particular pixel, the one with the highest score will be selected to be the updated depth estimation $\hat{d}$. The ranking and selection goes on iteratively to refine the depth estimation, where the new $\hat{d}$ will be used to generate better hypotheses with the updated context feature.

Our design for the hypothesis feature and context update mechanism is illustrated in Fig.1. There are three general guidelines we have followed when we design the hypothesis feature: (1) It should inform how well the matching is given by the hypothesis; (2) It should inform how well the hypothesis fits into the current spatial context; (3) It should be robust to different camera setups and resolutions. With these in mind, we propose to use the concatenation of the following three simple quantities computed from a hypothesis d_i:

$$[\mathbf{c}(d_i), \mathbf{e}(d_i), \mathbf{f}(\hat{d})] \tag{4}$$

The first term $\mathbf{c}(d_i)$ consists of the matching costs linearly interpolated from the cost volume for a fixed set of perturbations

$$\mathbf{c}(d_i) = [c(d_i + \epsilon) \text{ for } \epsilon \in \{-1, 0, 1\}] \tag{5}$$

Note that ϵ is defined in the pseudo disparity space.

The second term $\mathbf{e}(d_i)$ is a "tamed" version of the second-order error in the one-ring neighborhood $\mathcal{N}$ of the pixel $\mathbf{p}$. Note that the error is computed against $\hat{d}$ in the pseudo disparity space so that it can work across different metric and intrinsic scales.

$$\mathbf{e}(d_i) = [\tanh(\hat{d}_{\mathbf{p}+\Delta p} - \partial\hat{d}_{\mathbf{p}'} \cdot \Delta p - d_i) \text{ for } \Delta\mathbf{p} \in \mathcal{N}] \tag{6}$$

Finally, $\mathbf{f}(\hat{d})$ is a learned context feature of the previous refined depth, whose inputs are the concatenation of $\mathbf{c}(\hat{d})$, $\mathbf{e}(\hat{d})$ and the feature from a U-Net describing the reference view's high level appearance. We use a two-layer convolution network to initialize $\mathbf{f}(\hat{d})$, and subsequently we use a convolutional gated recurrent unit (GRU) to update $\mathbf{f}(\hat{d})$ given the updated $\hat{d}$. We remark that the overall iterative update is similar to the RAFT [37] framework, with the addition of the geometric error term $\mathbf{e}(d)$. We will discuss its crucial contribution in our ablation study in Sec.IV-B0c.

We use a contrastive loss to supervise the ranking module. Let d_{gt} be the ground truth depth and $\mathcal{D} = d_i$ be the set of

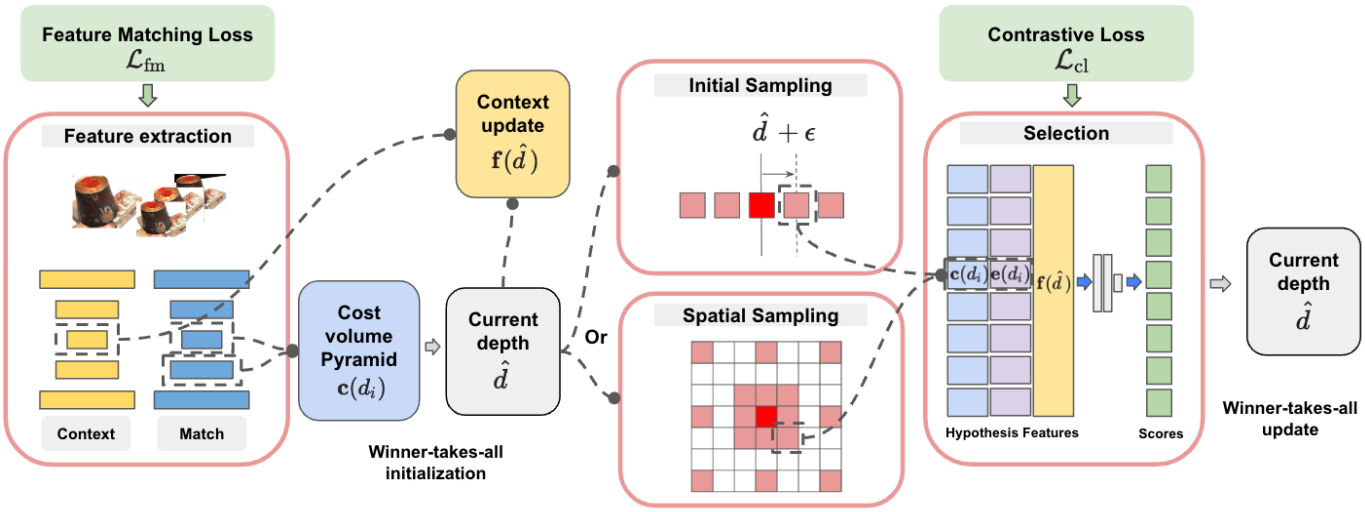

Fig. 2: Overview of the baseline MVS with our CHOSEN depth refinement. The winner hypothesis is from the initial full range cost volume, followed by applying the hypotheses sampling and best hypothesis selection. Initial sampling and spatial sampling are applied in an alternating fashion. Spatial sampling is facilitated through first order propagation as defined in Eq.(3). Key to this pipeline is the design of the hypothesis feature, defined in Sec.IV-B0c. The refined depth is upsampled to the higher resolution using nearest neighbor, and the same refinement procedure will be applied.

input hypotheses. We define the positive sample group to be those within 1-pseudo disparity of the ground truth

$$\mathcal{D}^+ = \{d_i \in \mathcal{D} : |d_i - d_{\text{gt}}| <= 1\} \tag{7}$$

Denote the score of d_i to be $s(d_i)$. Our contrastive loss is defined as

$$\mathcal{L}_{\text{cl}} = -\log\Big(\sum_{d_i \in \mathcal{D}^+} \frac{\exp(s(d_i))}{\sum_{d_i \in \mathcal{D}} \exp(s(d_i))}\Big) \tag{8}$$

As result, the ranking module will learn to put most of the weight on the set $\mathcal{D}^+$. We found this formulation most effective in training, and much superior compared to learning to put all weight on the closest sample to the ground truth. We believe it's due to that during spatial propagation, many samples will be already close to the ground truth and supervising on the closest sample introduces unnecessary competition among good samples.

C. Baseline MVS design

Here we will demonstrate the design of a simple end-to-end MVS pipeline with our depth refinement method. We have deliberately designed it such that *there are no learnable components for cost volume construction, depth refinement, or confidence prediction. The only learnable parts are the ranking MLP, context update convolutional layers, the light-weight U-Nets used for extracting matching feature as well as the high level appearance of the reference view*. Hence our cost volume construction should be inferior and more noisy compared to many existing MVS models that are equipped with cost volume filtering. Even with our simple pipeline, we show in Sec.IV-C that it can achieve much more superior quality in depth and normal compared to many state-of-art MVS works on the DTU dataset.

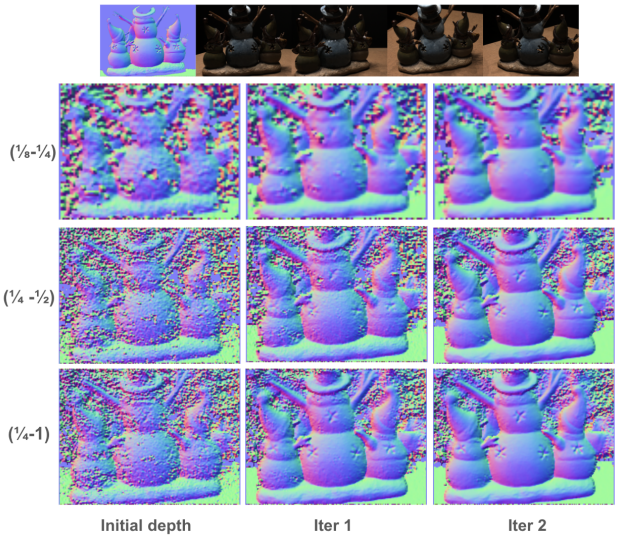

Fig. 3: Evolution of the refined depth in our baseline MVS model. The first row shows the ground truth and input images, and we mark each row with the cost volume pyramid configuration. As one can see, the winner-take-all initialization from the cost volume is usually very noisy, but nevertheless contain some accurate estimations. By iteratively re-sample and selecting the best hypotheses, the depth and quality are significantly improved.

a) Feature extraction: We use a two-stream feature extraction architecture similar to [37]. First, we use a matching feature network extracts distinctive features that are then used to

evaluate the cost of depth hypotheses between the reference and other source views. Second, we use a context feature network that learns high-level spatial cues for learnable hypothesis selection, applied only on the reference image. Both networks are based on a light-weight U-Net architecture [31] which extracts features from coarse to fine.

b) Matching cost aggregation: We choose the negative correlation as the matching cost:

$$c_v(d) = -\text{corr}(\mathbf{f}_{v\to\text{ref}}(d), \mathbf{f}_{\text{ref}}) \tag{9}$$

where $\mathbf{f}_{v\to\text{ref}}(d)$ denotes the v-th source view feature warped to the reference view through the hypothesis d. We aggregate the costs from all source views to have a single cost volume simply as

$$c(d) = \frac{1}{\Sigma_v w_v(d)} \Sigma_i w_v(d) c_v(d) \tag{10}$$

where

$$w_i(d) = \text{Sigmoid}(\alpha(\delta - c_i(d))^3) \tag{11}$$

computes the weighting for each view and hypothesis, and α is chosen so that it downplays the contribution from higher cost, increases the importance of lower cost, and not sensitive if the cost is close to δ, thanks to the cubic exponent.

c) Cost volume pyramid: As an optional benefit, we observe that it's possible to compute the matching costs for lower resolution depth using higher resolution feature. Specifically, we construct a grid $(x_p, y_p)_p$, indexed by lower resolution pixels, which samples the corresponding locations in the higher resolution feature. We can then compute the warped pixel location and the matching costs using the camera intrinsics and features corresponding to the higher resolution.

In practice, we typically compute one additional cost volume using higher resolution features. The new hypotheses are centered around the hypothesis of lowest cost in the coarse cost volume, spaced according the scale defined by the higher resolution pseudo disparity space. These two cost volumes now form a **cost volume pyramid**, which will be used to provide both coarse and fine matching information to the our ranking module through $\mathbf{c}(d)$ in Eq.(5). We will demonstrate in Sec.IV-B0a that using the higher resolution feature can unblock a higher level of accuracy even at lower resolution output, but it's otherwise *not* crucial to the success of our hypothesis ranking module.

d) Overall pipeline: Our baseline MVS pipeline, as depicted in Figure 2, starts by extracting features from a reference view and multiple source views. The first depth estimation $\hat{d}$ is obtained as the lowest cost hypothesis from the full cost volume. A cost volume pyramid is then constructed around the previous depth estimation. The refinement is performed on a hierarchy of resolutions. The refined depth from lower resolution will be upsampled using nearest-neighbor and set to be the initial depth for the next higher resolution.

e) Training: Similar to the contrastive loss used in [36], we supervise the feature matching using a simple contrastive loss

$$\mathcal{L}_{\text{fm}} = c(d_{\text{gt}}) - \text{clip}(c(d^-), [\alpha, \beta]) \tag{12}$$

where $c(d_{\text{gt}})$ is the ground truth depth, and d^- is the negative sample that is at least 1-pseudo disparity away from the ground truth and has the lowest matching cost

$$d^- = \arg\min_{|d-d_{\text{gt}}|>1} c(d) \tag{13}$$

The loss will not penalize cost of d^- already lower than α or higher than β. We choose $\beta = 1$ in lower resolution and $\alpha = 0$ for all resolution, so that the learned features will be as distinctive as possible. We found it better to choose a lower β in higher resolution (e.g. 0.8) due to the frequent appearance of local texture-less regions. Note that there is no any kind of cost volume regularization. Furthermore, we enforce the matching feature network to be only trained by the above loss, meaning it will not be influenced by the depth refinement. Therefore, the total loss of this baseline model is simply

$$\mathcal{L}_{\text{fm}} + \mathcal{L}_{\text{cl}} \tag{14}$$

where $\mathcal{L}_{\text{fm}}$ only updates the matching feature U-Net, and $\mathcal{L}_{\text{cl}}$ updates the context feature U-Net in the reference view, the ranking MLP and the convolutional layers that initialize and update $\mathbf{f}(\hat{d})$ in Eq.(4).

D. Integration in MVS pipeline

It's clear to see that CHOSEN can be embedded end-to-end in existing deep learning based MVS framework with little modification needed, since CHOSEN only needs the cost, second order error term and context feature to rank the depth hypotheses. One may substitute our simplistic yet noisy cost volume construction in the baseline with the techniques proposed in recent works, such as cost volume filtering [43] and other feature backbones [4], etc. The main thing to consider for the integration is that the depth hypotheses must be converted to the pseudo disparity representation in order to be processed by CHOSEN. Since the spacing of the hypotheses in the cost volume is crucial for contrastive learning, it might be necessary to retrain the model with CHOSEN from scratch since the original model might be sensitive to different spacing. For more results please refer to the expanded version of this paper[1].

IV. Experiments

A. Implementation details

We evaluate our depth refinement method using the baseline MVS pipeline described in Sec.III-C. Typically, we extract matching features at $1/8$, $1/4$, $1/2$ and 1 of the original resolution, and context features at $1/8$ and $1/4$ of the original resolution. The depth is initialized by winner-take-all from the full cost volume at $1/8$ resolution that has $N = 128$ slices, and a cost volume pyramid is built on $1/8$ & $1/4$ resolution matching

[1]https://arxiv.org/abs/2404.02225

Method	$\% < 1$mm ↑	MAE(@$<$ 1mm) ↓	$\% < 5°$(@$<$ 1dsp) ↑	$\% < 10°$(@$<$ 1dsp) ↑
baseline: (1/8,1/4,1/4)-(1/4,1/2,1)	**70.16**	**0.3669**	**45.53**	**73.34**
baseline: (1/8,1/4,1/4)-(1/4,1/2,1) + BlendedMVS	67.01	0.3992	43.77	71.73
baseline: (1/8,1/8,1/8)-(1/4,1/2,1)	**71.03**	**0.3558**	**68.16**	**84.01**
baseline: (1/8,1/8,1/8)-n.a.	63.62	0.3978	57.31	79.64
baseline w/o selection: (1/8,1/8,1/8)-(1/4,1/2,1)	59.33	0.4282	45.38	74.41
baseline w/o $\mathbf{e}(d)$ term: (1/8,1/8,1/8)-(1/4,1/2,1)	55.26	0.4223	41.51	66.63
baseline w/o CHOSEN: (1/8,1/8,1/8)-(1/4,1/2,1)	46.46	0.4653	20.90	49.00
MVSFormer [4]	61.92	0.4248	18.96	44.17
GeoMVSNet [50]	63.93	0.4089	15.18	38.40
GBi-Net [27]	34.01	0.4531	13.78	33.41
IterMVS [39]	51.09	0.4524	16.67	37.50
PatchMatchNet [40]	56.19	0.4260	18.03	41.82
UCSNet [14]	59.03	0.3942	31.31	54.39

TABLE I: Comparison for estimated depths and normals on DTU testing set. The metric is computed on all the valid ground truth pixels. We mark our best results separately for $1/8$ and $1/4$ resolutions. The results from other methods are evaluated at $1/4$ resolutions with nearest neighbor down-sampling. Notice that so long as the finest resolution matching features are used, the final depth accuracy metrics for these two output resolutions are very similar. Our simple baselines offer significant improvement in terms of depth quality even compared to the strongest state-of-art MVS pipelines.

features. One refinement stage consists of 4 iterations, where twice on initial sampled hypotheses and twice on spatially sampled hypotheses, in an alternating fashion. After upsampling the refined depth using nearest neighbor, we repeat the same refinement stage at $1/4$ resolution, with a $1/4$ & $1/2$ pyramid. Finally, the last refinement stage is performed with a $1/4$ & 1 pyramid with output at $1/4$ resolution. Each refinement stage has their own parameters for the ranking MLP and context update networks. We denote this default configuration as $(1/8, 1/4, 1/4)$ & $(1/4, 1/2, 1)$. Different choices for cost volume pyramid configurations and their effects will be discussed in Sec IV-B0a.

We set $M = 4$ in the initial hypotheses sampling, giving $2M + 1 = 9$ hypotheses. For spatial sampling, we use a set of offsets $\{\Delta\mathbf{p}\}$ consists of dilated 3×3 regular patches with dilation rate of 1 and 3, without repeating the patch center, giving in total 17 hypotheses for each pixel. This offset configuration is the same as the one illustrated in the spatial sampling part of Fig.1.

Our baseline model is trained and tested on a single NVIDIA A100 GPU (40G). The total trainable parameter count is about 1.1 million, including the matching and context U-Nets. Our selection module's learnable parameter count is about 781k. We first train with batch size 4, at input resolution 600×800, up to 200k iterations using the default Adam optimizer [22] at a learning rate of 0.001. We then fine-tune for up to 50k iterations at input resolution 1200×1600 at a learning rate of 0.0005. During training, we fix the closest source view and randomly sample two from the remaining source views, which is the only data augmentation we used. For testing, it takes around 0.6 seconds to run at 1200×1600 with 5 source views, with a total of 12 iterations of hypotheses ranking (4 iterations for each cost volume config). We provide results both from models trained on DTU [17] training set only, and on DTU mixed with the BlendedMVS [45] dataset, which will be tagged specifically.

a) Depth & normal metrics: We use the following metrics to measure the quality of depth and its induced normal map on the DTU evaluation dataset:

- $\% < x$mm: the percentage of pixels that have less than x-mm of absolute error.
- MAE(@$< x$mm): the mean absolute error on pixels that have less than x-mm of absolute error.
- $\% < x°$(@$< y$-dsp): the percentage of pixels where the normal is within $x°$ of angular error out of all pixels whose absolute error is less than y-pseudo disparity.

The ground truth depth are resized to the same resolution of the outputs using nearest neighbor. The normals are computed from the 3D coordinate's gradients using the Sobel filter. All the above metrics are computed only on the valid pixels where the ground-truth is available.

B. Ablation studies

a) Cost volume pyramid design: Here we study the effects of different cost volume pyramid configurations. We choose the following variants of the configuration:

- (1/8,1/4,1/4)-(1/4,1/2,1): There are 3 stages of refinement with $1/8$ & $1/4$, $1/4$ & $1/2$ and $1/4$ & 1 cost volume pyramids, and output refined depth at $1/4$ resolution. This is our default configuration and we also include the results trained on data mixed BlendedMVS.
- (1/8,1/8,1/8)-(1/4,1/2,1): There are 3 stages of refinement with $1/8$ & $1/4$, $1/8$ & $1/2$ and $1/8$ & 1 cost volume pyramids, and output refined depth at $1/8$ resolution.
- (1/8,1/8,1/8)-n.a: Only the coarsest resolution matching feature is used. There are 3 stages of refinement with $1/8$ resolution feature, and output refined depth at $1/8$ resolution.

Quantitative results on DTU testing set are reported in the first four rows in Tab.I. First of all, notice that even without cost volume pyramid, our method performs reasonably well, indicating that the pyramid design *mainly unblocks higher accuracy at lower resolution output, but otherwise is an optional feature*. Second, we can observe from the first and third rows in Tab.I that the $\% < 1$mm metrics are similar for the outputs at $1/8$ resolution and $1/4$ resolution, so long as the full resolution matching features are used. Comparing $1/8$ (third and fourth rows) and $1/4$ (first and second rows) output

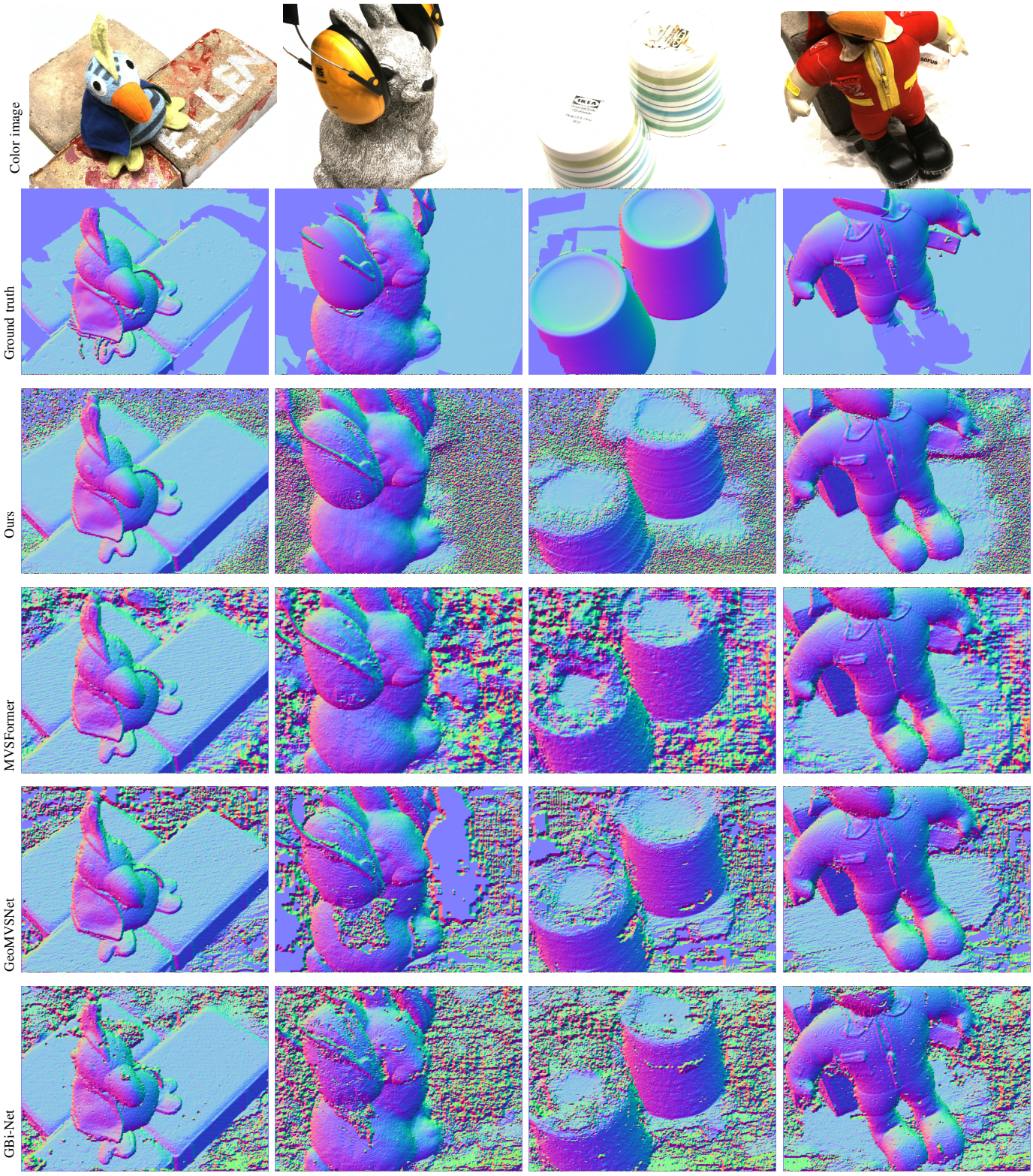

Fig. 4: Normal quality comparisons on DTU. Our simple baseline MVS trained only on DTU produces significantly more accurate normals.

resolution, we can observe that it becomes more difficult to get accurate normals at higher resolution. This is due to the fact that high frequency details only emerge in higher resolution, which may be impossible to obtain without a photometric

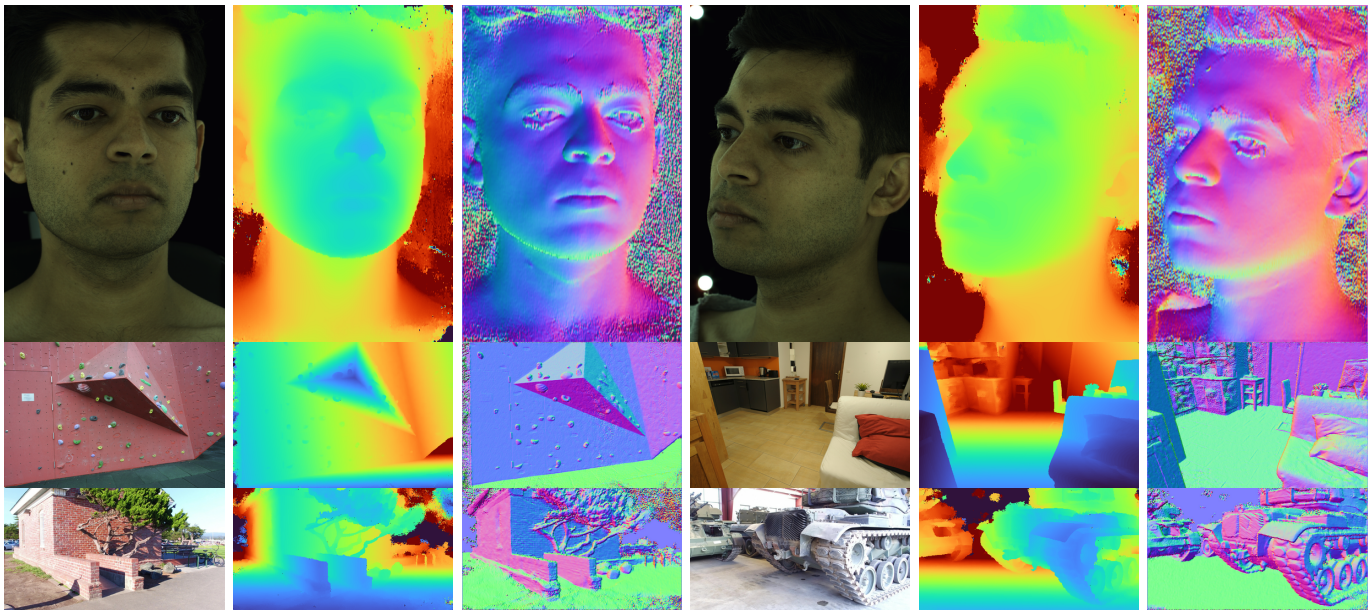

Fig. 5: Direct application of our DTU + BlendedMVS trained baseline model on instances from MultiFace [23], Tanks & Temples and ETH3D [34] datasets. Our simple baseline achieves consistent generalization ability even though trained on substantially different data.

appearance model. Lastly, we remark that the worse results from mixed data training is likely due to that BlendedMVS contains images with severe aliasing artifacts, which could hurt our model training with $\mathcal{L}_{\text{fm}}$.

b) Selection v.s. expectation: The ability to rank arbitrary hypotheses is essential to the success of our depth refinement. Here we illustrate the point by comparing with an approach where the learned score s_i for each hypothesis d_i is used for taking a weighted average. In this ablated approach, the refined depth is obtained by

$$\hat{d} = \frac{1}{\sum_i \exp(s_i)} \sum_i \exp(s_i) \cdot d_i$$

and we use smoothed L^1 loss between $\hat{d}$ and d_{gt} in place of $\mathcal{L}_{cl}$. Everything else in the pipeline stays the same. Quantitative results are reported in the fifth row of Tab.I. We note that this alternative approach based on weighted average is much less accurate than the our approach based on ranking and selection. This can be attributed to the difficult task of learning the weights for arbitrary hypotheses, while learning to *classify* the hypotheses is a much easier task.

c) Hypothesis feature design: The "tamed" second order error term $\mathbf{e}(d)$ in Eq.(4) can be viewed as an extra component compared to the input feature design in the RAFT [37] framework for optical flow estimation. Here we show that it is a vital component in the hypothesis ranking model that improves the overall smoothness and accuracy. Quantitative results on DTU testing set are reported in the sixth row of Tab.I. Since each hypothesis is evaluated independently, without the information about how well a particular hypothesis fits in the local geometry, the model struggles to select the best hypothesis solely based on matching and context information.

C. Comparisons on depths and normals

We collect the testing results from various recent deep learning based MVS works in the last part of Tab.I. The outputs for the candidate models and ground truths are resized to $1/4$ of input resolution using nearest neighbor. Our simple baseline MVS pipeline significantly outperforms the strongest state-of-art in terms depth and normal quality. Visual comparisons are shown in Fig.4.

D. Qualitative results for baseline

We demonstrate the excellent generalization ability of our simple baseline on various datasets including MultiFace [41], Tanks & Temples [23] and ETH3D [34]. Results in Fig.5 use the same baseline model trained on mixture of DTU and BlendedMVS reported in Tab.I.

V. Limitations and Conclusion

We have demonstrated that CHOSEN is simple yet effective for multi-view depth refinement. One limitation of CHOSEN is that it becomes more expensive to sample and select from a large number of hypotheses at high resolution. This is part of the reason why we settled at $1/4$ resolution to arrive at a sweet spot of both performance and run time. Another limitation is that we did not focus on bundle adjustment and joint filtering of depths for point cloud and surface reconstruction, which we believe should be a subject of its own interests, especially considering the proliferation of volumetric based methods such as [47].

REFERENCES

[1] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (Proc. SIGGRAPH) **28**(3) (Aug 2009)

[2] Barron, J.T., Poole, B.: The fast bilateral solver. In: European conference on computer vision. pp. 617–632. Springer (2016)

[3] Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: Bmvc. vol. 11, pp. 1–11 (2011)

[4] Cao, C., Ren, X., Fu, Y.: Mvsformer: Multi-view stereo by learning robust image features and temperature-based depth. Transactions of Machine Learning Research (2023)

[5] Cao, C., Ren, X., Fu, Y.: Mvsformer++: Revealing the devil in transformer's details for multi-view stereo. arXiv preprint arXiv:2401.11673 (2024)

[6] Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

[7] Cheng, S., Xu, Z., Zhu, S., Li, Z., Li, L.E., Ramamoorthi, R., Su, H.: Deep stereo using adaptive thin volume representation with uncertainty awareness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2524–2534 (2020)

[8] Collins, R.T.: A space-sweep approach to true multi-image matching. In: Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 358–363. IEEE (1996)

[9] Ding, Y., Yuan, W., Zhu, Q., Zhang, H., Liu, X., Wang, Y., Liu, X.: Transmvsnet: Global context-aware multi-view stereo network with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8585–8594 (2022)

[10] Furukawa, Y., Hernández, C.: Multi-view stereo: A tutorial. Found. Trends Comput. Graph. Vis. **9**(1-2), 1–148 (2015)

[11] Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)

[12] Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: ICCV (2015)

[13] Galliani, S., Lasinger, K., Schindler, K.: Gipuma: Massively parallel multi-view stereo reconstruction. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V **25**(361-369), 2 (2016)

[14] Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 2492–2501. IEEE (2020)

[15] Hui, T., Loy, C.C., Tang, X.: Depth map super-resolution by deep multi-scale guidance. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9907, pp. 353–369. Springer (2016)

[16] Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. pp. 2807–2817 (2018)

[17] Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 406–413 (2014)

[18] Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2326–2334. IEEE Computer Society (2017)

[19] Kar, A., Häne, C., Malik, J.: Learning a multi-view stereo machine. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 365–376 (2017)

[20] Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: IEEE International Conference on Computer Vision (ICCV) (2017)

[21] Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., Izadi, S.: StereoNet: Guided hierarchical refinement for edge-aware depth prediction. In: European Conference on Computer Vision (ECCV) (2018)

[22] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[23] Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)

[24] Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. Int. J. Comput. Vis. **38**(3), 199–218 (2000). https://doi.org/10.1023/A:1008191222954, https://doi.org/10.1023/A:1008191222954

[25] Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 418–433 (2005)

[26] Lin, C., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3d object reconstruction. pp. 7114–7121. AAAI Press (2018)

[27] Mi, Z., Di, C., Xu, D.: Generalized binary search network for highly-efficient multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)

[28] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

[29] Pang, J., Sun, W., Ren, J., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: International Conference on Computer Vision-Workshop on Geometry Meets Deep Learning (ICCVW 2017) (2017)

[30] Peng, R., Wang, R., Wang, Z., Lai, Y., Wang, R.: Rethinking depth estimation for multi-view stereo: A unified representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8645–8654 (2022)

[31] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. MICCAI (2015)

[32] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision (2002)

[33] Schönberger, J.L., Zheng, E., Frahm, J., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9907, pp. 501–518. Springer (2016)

[34] Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3260–3269 (2017)

[35] Taniai, T., Matsushita, Y., Sato, Y., Naemura, T.: Continuous stereo matching using local expansion moves. arXiv preprint arXiv:1603.08328

[36] Tankovich, V., Hane, C., Zhang, Y., Kowdle, A., Fanello, S., Bouaziz, S.: Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14362–14372 (2021)

[37] Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)

[38] Tola, E., Strecha, C., Fua, P.: Efficient large-scale multi-view stereo for ultra high-resolution image sets. Mach. Vis. Appl. **23**(5), 903–920 (2012)

[39] Wang, F., Galliani, S., Vogel, C., Pollefeys, M.: Itermvs: Iterative probability estimation for efficient multi-view stereo (2022)

[40] Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: Patchmatchnet: Learned multi-view patchmatch stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14194–14203 (2021)

[41] Wuu, C.h., Zheng, N., Ardisson, S., Bali, R., Belko, D., Brockmeyer, E., Evans, L., Godisart, T., Ha, H., Huang, X., Hypes, A., Koska, T., Krenn, S., Lombardi, S., Luo, X., McPhail, K., Millerschoen, L., Perdoch, M., Pitts, M., Richard, A., Saragih, J., Saragih, J., Shiratori, T., Simon, T., Stewart, M., Trimble, A., Weng, X., Whitewolf, D., Wu, C., Yu, S.I., Sheikh, Y.: Multiface: A dataset for neural face rendering. In: arXiv (2022). https://doi.org/10.48550/ARXIV.2207.11243, https://arxiv.org/abs/2207.11243

[42] Xu, Q., Tao, W.: Pvsnet: Pixelwise visibility-aware multi-view stereo network. CoRR **abs/2007.07714** (2020), https://arxiv.org/abs/2007.07714

[43] Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 767–783 (2018)

[44] Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: IEEE Conference

on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 5525–5534 (2019)
[45] Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. Computer Vision and Pattern Recognition (CVPR) (2020)
[46] Yu, Z., Gao, S.: Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 1946–1955. IEEE (2020)
[47] Zhang, J., Yang, G., Tulsiani, S., Ramanan, D.: Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. Advances in Neural Information Processing Systems **34**, 29835–29847 (2021)
[48] Zhang, J., Li, S., Luo, Z., Fang, T., Yao, Y.: Vis-mvsnet: Visibility-aware multi-view stereo network. International Journal of Computer Vision **131**(1), 199–214 (2023)
[49] Zhang, Y., Khamis, S., Rhemann, C., Valentin, J., Kowdle, A., Tankovich, V., Schoenberg, M., Izadi, S., Funkhouser, T., Fanello, S.: ActiveStereoNet: End-to-end self-supervised learning for active stereo systems. European Conference on Computer Vision (ECCV) (2018)
[50] Zhang, Z., Peng, R., Hu, Y., Wang, R.: Geomvsnet: Learning multi-view stereo with geometry perception. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21508–21518 (2023)