PatchmatchNet: Learned Multi-View Patchmatch Stereo Supplementary Material

Fangjinhua Wang[1] Silvano Galliani[2] Christoph Vogel[2] Pablo Speciale[2] Marc Pollefeys[1,2]
[1]Department of Computer Science, ETH Zurich
[2]Microsoft Mixed Reality & AI Zurich Lab

1. Why not use 3D cost volume regularization?

The adaptive evaluation of our learning-based Patchmatch utilizes 3D convolution layers with $1 \times 1 \times 1$ kernels for the matching cost computation as well as the pixel-wise view weight estimation. This is in contrast to common previous works [3, 6, 10, 14, 15, 16, 17] where a 3D U-Net regularizes the cost volume. Similarly, arguing that the distribution of cost volume itself being not discriminative enough [4, 12], PVSNet [15] also applies a 3D U-Net for predicting the visibility per source view.

The problem with such regularization framework is that it requires a regular spatial structure in the volume. Although we concatenate the matching costs per pixel and depth hypothesis into a volume-like shape as other works [3, 6, 10, 14, 15, 16, 17], we do not possess such a regular structure: (i) the depth hypotheses for each pixel and its spatial neighbors are different, which makes it difficult to aggregate cost information in the spatial domain; (ii) the depth hypotheses of each pixel are not uniformly distributed in the inverse depth range as CIDER [14], which makes it difficult to aggregate cost information along depth dimension.

Recall that during the computation of the pixel-wise view weights in the initial iteration of Patchmatch, depth hypotheses are randomly distributed in the *inverse* depth range, *i.e*. the hypotheses are spatially different per pixel. In each subsequent iteration (on stage k), we perform local perturbation by generating per pixel N_k depth hypotheses uniformly in the normalized inverse depth range R_k, which is centered at the previous estimate. Consequently, the hypotheses of spatial neighbors can differ significantly, especially at depth discontinuities and thin structures. Including the hypotheses obtained by adaptive propagation, that are, moreover, not uniform in the inverse depth range, will increase these effects further.

In the end, however, the main reason for our approach to avoid 3D cost volume regularization altogether is efficiency. In a coarse-to-fine framework, running such regularization frameworks over multiple iterations of Patchmatch on each stage would increase memory consumption and run-time significantly and mitigate our main contribution of building a high-performance, but particularly lightweight framework that can operate with a high computation speed.

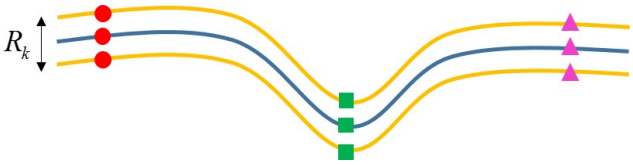


Figure 1: Illustration of local perturbation in the (normalized) inverse depth range. The blue and orange lines represent previous estimation and new hypotheses respectively.

2. How to set the normalized inverse depth range R_k in the local perturbation step of Patchmatch?

After the initial iteration, our set of hypothesis is obtained by adaptive propagation and by local perturbation of the previous estimation. Recall that our local perturbation procedure enriches the set of hypothesis by generating per pixel N_k depth hypotheses uniformly in the normalized inverse depth range R_k, which is illustrated in Fig. 1.

The objective is two-fold. Especially at the beginning, at low resolution, this helps to further explore the search space. More importantly, our adaptive propagation implicitly assumes front-to-parallel surfaces, since we do not explicitly include tangential surface information (due to an implied heavy memory consumption) like [2, 5, 13]. Sampling in the local vicinity of the previous estimation will refine the solution locally and mitigate potential disadvantages from not explicitly modeling tangential surface information. We find it helpful to apply these perturbations already at an early stage to inject the positive effects into hypothesis propagation and note that a-posteriori refinement at the finest level alone cannot recover the same quality. In practice, we again operate in coarse-to-fine manner and set R_k accordingly, based on the hierarchy level.

Fig.2 shows the cumulative distribution function of the normalized absolute error in the inverse depth range on

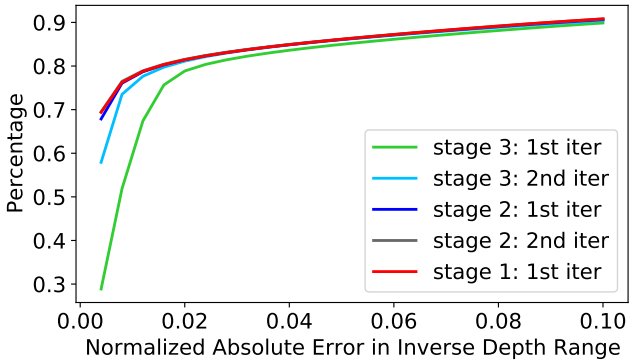


Figure 2: Cumulative distribution function of normalized absolute errors in the inverse depth range on DTU's evaluation set [1]. '*Stage k, n^{th} iter*' denotes the result of the n^{th} iteration of Patchmatch on stage k.

DTU's evaluation set [1]. After the first iteration of Patchmatch on stage 3, the estimation error decreases remarkably: the normalized error is already smaller than 0.1 for 90.0% percent of the cases. Visibly, the performance keeps improving after each iteration. To correct errors in estimation and refine the results on stage k, we set R_k to compensate most of estimation errors. For example, we set $R_3 = 0.38$ for Patchmatch on stage 3 after first iteration so that we can cover most ground truth depth in the hypothesis range and then refine the results. Besides, adaptive propagation will further correct those wrong estimations with the depth hypotheses from neighbors when sampling in R_k fails in refinement (*c.f.* Fig. 6 from the paper).

3. Why not include propagation for last iteration of Patchmatch on stage 1?

Similar to MVSNet [17], the point cloud reconstruction mainly consists of photometric consistency filtering, geometric consistency filtering and depth fusion. Photometric consistency filtering is used to filter out those depth hypotheses that have low confidence. Based on MVSNet [17], we define the confidence as the probability sum of the depth hypotheses that fall in a small range near the estimation. We use the probability $\mathbf{P}$ (*c.f.* Eq. 7 from the paper) from the last iteration of Patchmatch on stage 1 for filtering. In this iteration, we only perform local perturbation, without adaptive propagation. At stage 1, operating at a quarter the image resolution and with the algorithm almost converged, the hypotheses obtained via propagation from spatial neighbors are usually very similar to the current solution. Such irregular sampling of the probability space causes bias in the regression (*c.f.* Eq. 7 from the paper) and the estimate becomes over-confident at the current solution, where most propagated samples are located. In contrast, by performing only the local perturbation, the depth hypotheses are uniformly distributed in the inverse depth range. Contrary to previous iterations, we compute the estimated depth at pixel $\mathbf{p}$, $\mathbf{D}(\mathbf{p})$, by utilizing the inverse depth regression [14], which is based on the *soft argmin* operation [8]:

$$\mathbf{D}(\mathbf{p}) = \Big(\sum_{j=0}^{D-1} \frac{1}{d_j} \cdot \mathbf{P}(\mathbf{p}, j)\Big)^{-1}, \tag{1}$$

where $\mathbf{P}(\mathbf{p}, j)$ is the probability for pixel $\mathbf{p}$ at the j-th depth hypothesis. Then we compute the probability sum of four depth hypotheses that are nearest to the estimation to measure the confidence [17].

4. Weighting in the Adaptive Spatial Cost Aggregation

Recall that in Eq. 6 of the paper we utilize two weights to aggregate our spatial costs, $\{w_k\}_{k=1}^{K_e}$ based on spatial feature similarity and $\{d_k\}_{k=1}^{K_e}$ based on the similarity of depth hypotheses. The feature weights $\{w_k\}_{k=1}^{K_e}$ at a pixel $\mathbf{p}$ are based on the feature similarity at the sampling locations around $\mathbf{p}$, measured in the reference feature map $\mathbf{F}_0$. Given the sampling positions $\{\mathbf{p} + \mathbf{p}_k + \Delta\mathbf{p}_k\}_{k=1}^{K_e}$, we extract the corresponding features from $\mathbf{F}_0$ via bilinear interpolation. Then we apply group-wise correlation [7] between the features at each sampling location and $\mathbf{p}$. The results are concatenated into a volume on which we apply 3D convolution layers with $1 \times 1 \times 1$ kernels and sigmoid non-linearities to output normalized weights that describe the similarity between each sampling point and $\mathbf{p}$.

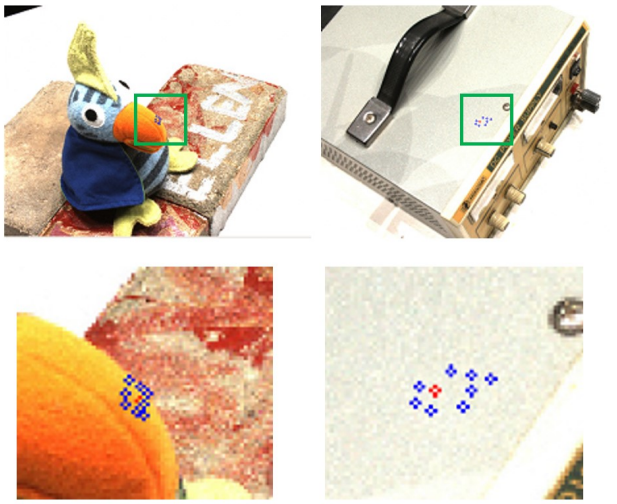
As discussed in *Sec.* 1, neighboring pixels will be assigned different depth values throughout the estimation process. For pixel $\mathbf{p}$ and the j-th depth hypothesis, our depth weights $\{d_k\}_{k=1}^{K_e}$ take this into account and downweight the influence of samples with large depth difference, especially when located across depth discontinuities. To that end, we collect the absolute difference in inverse depth between each sampling point and pixel $\mathbf{p}$ with their j-th hypotheses, and obtain the weights by applying a sigmoid function on the, again, inverted differences for normalization.

5. Evaluation of Multi-stage Depth Estimation

We use multiple stages to estimate the depth map in a coarse-to-fine manner. Here, we analyze the effectiveness of our multi-stage framework. We upsample the estimated depth maps on stages 3, 2 and 1, to the same resolution as the input and then reconstruct the point clouds. As shown in Table 1, the reconstruction quality gradually increases from coarser stages to finer ones. This shows that our multi-stage framework can reconstruct the scene geometry with increasing accuracy and completeness.

Stages	Acc.(mm)	Comp.(mm)	Overall(mm)
3	0.740	0.389	0.564
2	0.471	0.283	0.377
1	0.441	**0.268**	0.354
0	**0.427**	0.277	**0.352**

Table 1: Quantitative results of different stages on DTU's evaluation set [1] (lower is better). The depth maps on stages 3, 2 and 1 are upsampled to reach the same resolution as input images and then used to reconstruct point clouds.



(a) object boundary (b) textureless region

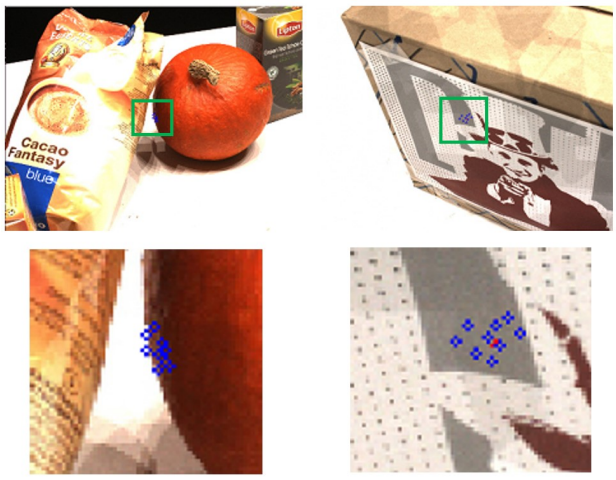
Figure 3: Visualization of sampling locations in adaptive propagation for two typical situations: object boundary and textureless region. The center points and sampling points are shown in red and blue respectively.

6. Visualization of Adaptive Propagation

We visualize the sampling locations in two typical situations, at an object boundary and a textureless region. As shown in Fig. 3, for the pixel $\mathbf{p}$ at the object boundary, all sampling points tend to be located on the same surface as $\mathbf{p}$. In contrast, for the pixel $\mathbf{q}$ in the textureless region, the sampling points are spread over a larger region. By sampling from a large region, a more diverse set of depth hypotheses can be propagated to $\mathbf{q}$ and reduce the local ambiguity for depth estimation in the textureless area. The visualization shows two examples how the adaptive propagation successfully adapts the sampling to different challenging situations.

7. Visualization of Adaptive Evaluation

Here, we again visualize the sampling locations for two situations, at an object boundary and a textureless region. Fig.4 demonstrates that for the pixel $\mathbf{p}$ at the object boundary, sampling points tend to stay within the boundaries of the object, such that they focus on similar depth regions. For the pixel $\mathbf{q}$ in the textureless region, the points are



(a) object boundary (b) textureless region

Figure 4: Visualization of sampling locations in adaptive evaluation for two typical situations: object boundary and textureless region. The center points and sampling points are shown in red and blue respectively.

distributed sparsely to sample from a large context, which helps to obtain reliable matching and to reduce the ambiguity. Again, the visualization demonstrates how our adaptive evaluation adapts the sampling for the spatial cost aggregation to different situations.

8. Visualization of Point Clouds

We visualize reconstructed point clouds from DTU's evaluation set [1], Tanks & Temples dataset [9] and ETH3D benchmark [11] in Fig. 5, 6, 7.

Figure 5: Reconstruction results on DTU's evaluation set [1].

Figure 6: Reconstruction results on Tanks & Temples dataset [9].

Figure 7: Reconstruction results on ETH3D benchmark [11].

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision (IJCV)*, 2016. 2, 3, 4

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, 2011. 1

[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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[4] Zehua Fu and Mohsen Ardabilian Fard. Learning confidence measures by multi-modal convolutional neural networks. In *Winter Conf. on Applications of Computer Vision (WACV)*, 2018. 1

[5] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[6] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[7] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[8] 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 *International Conference on Computer Vision (ICCV)*, 2017. 2

[9] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017. 3, 5

[10] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[11] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 6

[12] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*, 2016. 1

[13] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[14] Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *AAAI*, 2020. 1, 2

[15] Qingshan Xu and Wenbing Tao. PVSNet: Pixelwise visibility-aware multi-view stereo network. *ArXiv*, 2020. 1

[16] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[17] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2