IterMVS: Iterative Probability Estimation for Efficient Multi-View Stereo

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

1. Matching Similarities with Multi-scale Features

When performing the iterative updates, we compute the matching similarities from features on all levels to include multi-scale information. For a pixel $\mathbf{p}$ with coordinates (x, y) in the depth-map $\mathbf{D} \in \mathbb{R}^{W/4 \times H/4}$, we first find its corresponding position $\mathbf{p}_l$ for level l ($l = 1, 2, 3$) in the reference feature map at the coordinates $(x/2^{l-2}, y/2^{l-2})$. Then, the reference feature of $\mathbf{p}_l$, $\mathbf{F}_{0,l}(\mathbf{p}_l)$, is found via bilinear interpolation. Afterwards, with N_l new depth hypotheses, known camera parameters (for each level l) and source features $\{\mathbf{F}_{i,l}\}_{i=1}^{N-1}$, we warp (using differentiable warping) $\mathbf{p}_l$ into the respective source view and compute the matching similarities between reference and each source view. Finally, we use the $2\times$ upsampled pixel-wise view weights to compute the integrated matching similarities and pass them through a level-wise 2D U-Net to aggregate the neighborhood information.

2. Depth Upsampling

Following RAFT [6], we upsample the depth map from $1/4$ to full resolution. Specifically, the depth of each pixel in the high resolution depth map is a convex combination of its 9 neighbors at the coarse resolution. The weights are learned from the reference feature map. Fig. 1 illustrates the upsampling process.

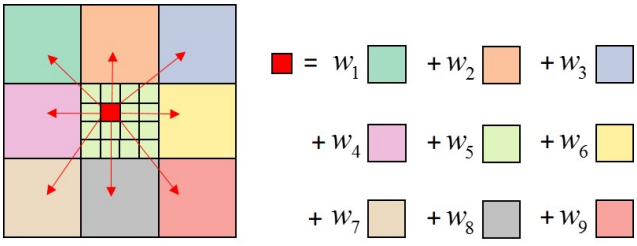


Figure 1. Illustration of depth upsampling. In the high resolution depth map, the depth of each pixel is the weighted sum of its 9 coarse resolution neighbors.

3. Point Cloud Reconstruction

Before fusing the depth maps, we filter out unreliable depth estimates, following MVSNet [7]. There are two filtering steps: geometric consistency filtering and confidence filtering.

Geometric Consistency Filtering. Following MVSNet [7], we apply a geometric constraint to measure the consistency of depth estimates among multiple views. For each pixel $\mathbf{p}$ in the reference view, we project it, using its depth d_0, to a pixel $\mathbf{p}_i$ in the i-th source view. After looking up its depth d_i in the source view, we reproject $\mathbf{p}_i$ into the reference view, and look up the depth d_{reproj} at this location, $\mathbf{p}_{\text{reproj}}$. We consider pixel $\mathbf{p}$ and its depth d_0 as consistent to the i-th source view, if the distances, in image space and depth, between the original estimate and its reprojection satisfy:

$$|\mathbf{p}_{\text{reproj}} - \mathbf{p}| < \delta, |d_{\text{reproj}} - d_0|/d_0 < \varepsilon, \quad (1)$$

where $\delta = 1$ and $\varepsilon = 0.01$ are two thresholds. Finally, we accept the estimations as reliable, if they are consistent in at least N_{geo} source views.

Confidence Filtering. Since our learned confidence indicates how close the estimation is to the ground truth depth, we use it to filter out estimations with high uncertainty. Specifically, we use a confidence threshold $\tau = 0.3$ throughout the experiments to filter out all the pixels with confidence lower than it.

4. Visualization of Probability

Our GRU-based probability estimator encodes the per-pixel probability distribution of depth with the hidden state. A 2D CNN is applied on the hidden state to estimate the probability of D_2 samples that are uniformly distributed in the *inverse* depth range for each pixel. We visualize this probability distribution for various scenes in Fig. 2. For pixels with distinct features, the probability has a single dominant peak and the estimation is precise. For some challenging situations, where distributions are non-peaky or multi-modal, *e.g.* in textureless areas, our hybrid depth estimation strategy can still robustly produce estimations as accurate

as possible. We also visualize the update process of probability distribution in Fig. 3. We observe that the probability distribution becomes more focused, several local maxima get suppressed and the estimation becomes more precise with more GRU iterations. In each iteration, multi-scale matching information is injected into the hidden state. This allows the hidden state to more accurately model the per-pixel probability distribution of depth with each iteration.

5. Visualization of Pixel-wise View Weight

Several examples for our estimated pixel-wise view weights are depicted in Fig. 4. Comparing the view weights with the visible areas in the reference validates that the all visible areas receive higher weights, while occluded and invisible parts have very low weights. Interestingly, in the first two images, pixels on the windows have low weights. Here, especially the upper row of windows mirror the surrounding buildings and cannot provide reasonable matching information. The other images have low weights in visible regions at areas with strong perspective and specular reflections as well as occlusions, while fronto-parallel and textured regions achieve higher weights. We conclude that our pixel-wise view weight is capable to determine co-visible areas between the reference and source images.

6. Visualization of Point Clouds

We visualize the reconstructed point clouds from DTU's evaluation set [1], Tanks & Temples dataset [2] and ETH3D benchmark [5] in Fig. 5, 6 and 7.

7. Future Work

Currently, the learned confidence is only used to filter out unreliable estimates before depth fusion. However, we believe it will be a promising direction to further exploit the confidence in each GRU iteration. For example, one can refine the depth of unconfident areas with the information propagated from those confident areas [3, 4]. Another idea would be to focus more effort on unconfident areas only, while leaving the confident areas unchanged, which further should improve efficiency.

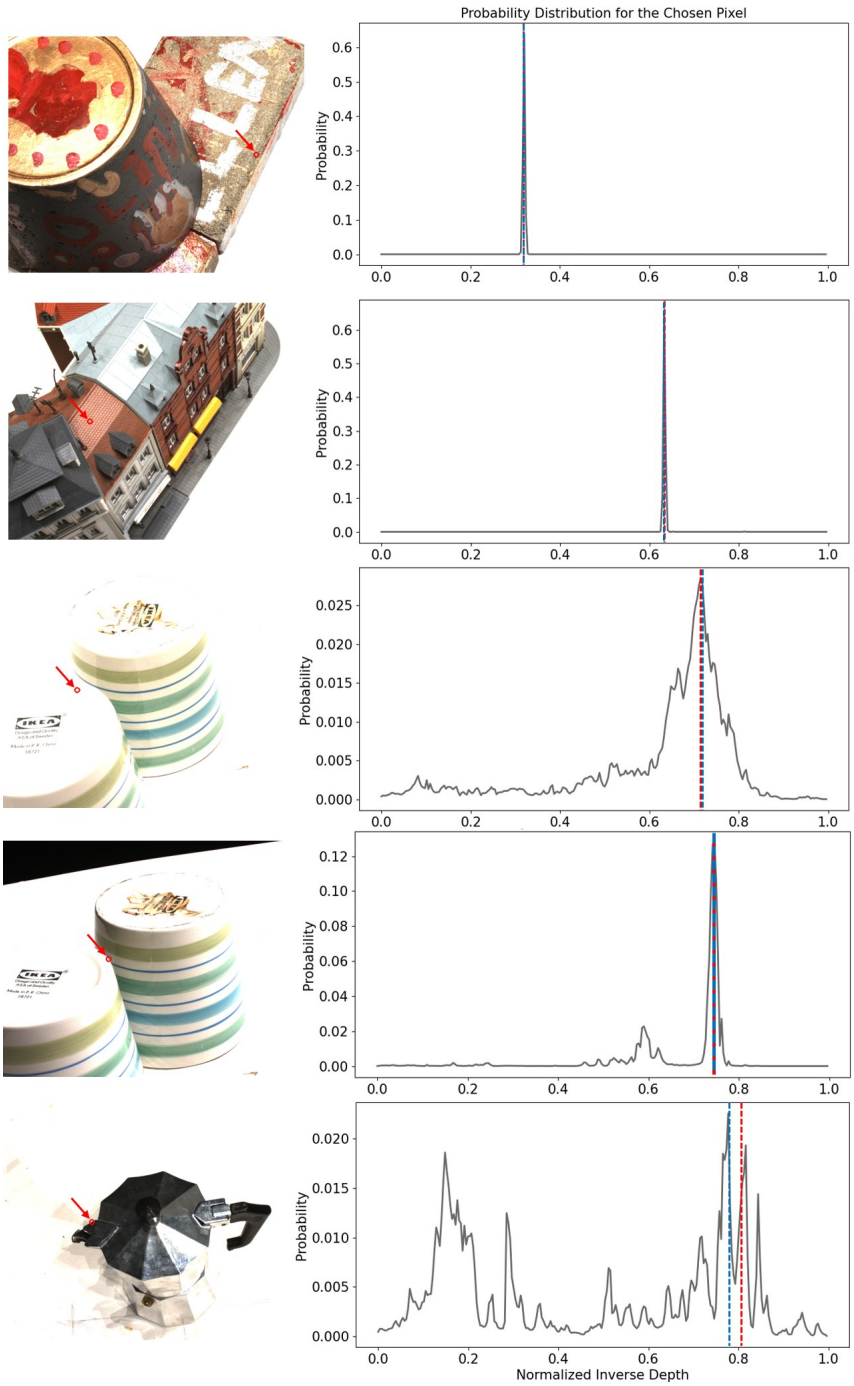
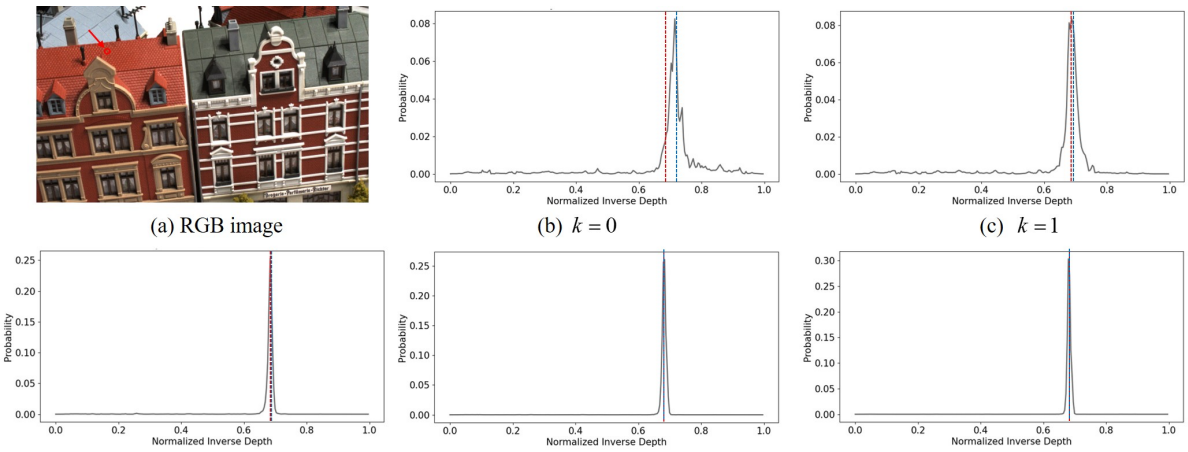


Figure 2. Visualization of probability. Left: Reference images (the chosen pixels are highlighted). Right: Probability distribution of depth for the chosen pixels. Red line denotes ground truth depth and blue line denotes our estimation.



(a) RGB image (b) $k = 0$ (c) $k = 1$

(d) $k = 2$ (e) $k = 3$ (f) $k = 4$

Figure 3. (a) Reference image (the chosen pixel is highlighted). (b)-(f) Probability distribution of depth for the chosen pixel in the k-th GRU iteration ($k = 0$ represents the probability distribution from initial hidden state h_0). Red line denotes ground truth depth and blue line denotes our estimation. Note our estimation becomes more accurate while several local maxima get suppressed in the distribution that converges to a single, more pronounced peak with more iterations.

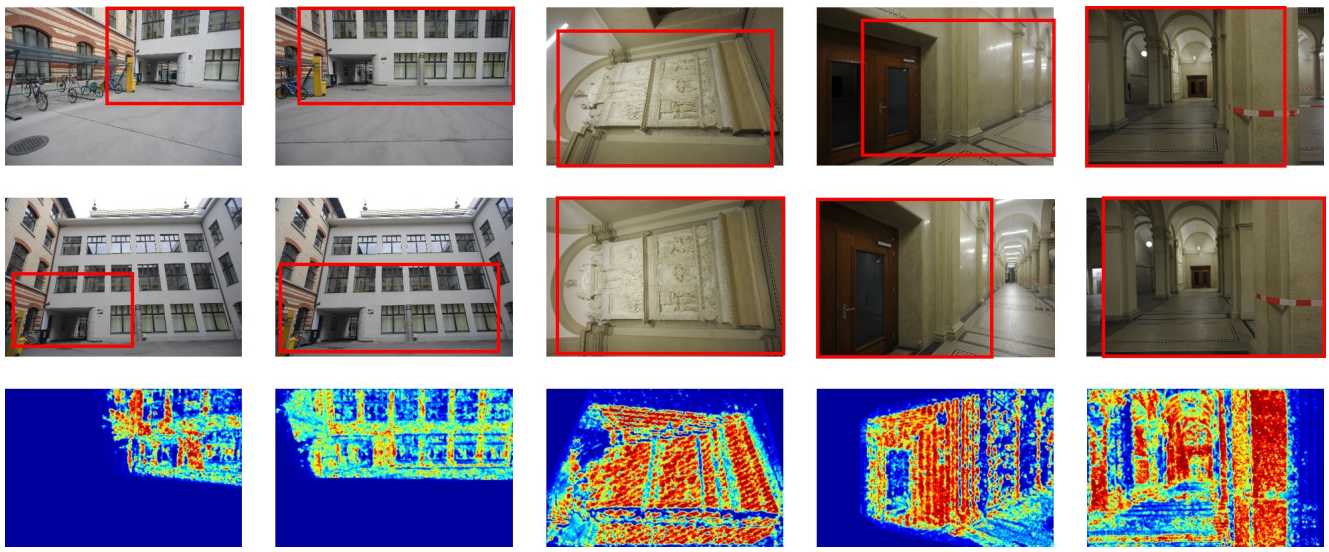


Figure 4. Visualization of our learned pixel-wise view weight on ETH3D [5]. Top row: reference images. Middle row: source images. Bottom row: pixel-wise view weight. Areas marked with boxes in reference images and source images are co-visible.

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

(a) Intermediate dataset

(b) Advanced dataset

Figure 6. Reconstruction results on Tanks & Temples dataset [2].

(a) Training dataset

(b) Test dataset

Figure 7. Reconstruction results on ETH3D Benchmark [5].

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, 5

[2] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *TOG*, 2017. 2, 6

[3] Andreas Kuhn, Christian Sormann, Mattia Rossi, Oliver Erdler, and Friedrich Fraundorfer. Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In *3DV*, pages 404–413. IEEE, 2020. 2

[4] Zhaoxin Li, Wangmeng Zuo, Zhaoqi Wang, and Lei Zhang. Confidence-based large-scale dense multi-view stereo. *TIP*, 29:7176–7191, 2020. 2

[5] 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 *CVPR*, 2017. 2, 4, 7

[6] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 1

[7] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 1