# Supplementary material for Multi-View Stereo by Temporal Nonparametric Fusion

## A. Encoder–decoder architecture

We have included details on the network architecture that was used for the encoder and decoder models. Table 5 lists the netowrk components. 'Ch. I/O' refers to the channel number of the input/output. All '*_up' means the upsampled features, and the upsample layers use bilinear interpolation. The plus sign '+' refers to the concatenation operation. The encoder consists of layers from 'conv1' to 'conv5_1' , and the output of the 'conv5_1' layer is $\mathbf{z}$ in Fig. 2, which will be transformed by the GP. The layers from 'upconv4' to 'disp0' are part of the decoder, and the 'disp0' generates the final inverse depth prediction. All layers are followed by batch normalization and ReLU, except the 'disp*' layers.

Additionally, Fig. 8 visualizes the encoder–decoder arcitecture as a block diagram. The orange blocks are parts of the encoder, and the blue blocks form the decoder. The purple blocks indicate four 'disp*' layers. Except 'disp*' layers, each block is followed by a darker block which indicate batch normalization and ReLU layers. There are four skip connections in the figure, which corresponds to feeding the outputs of 'conv*_1' layers into 'iconv*' layers.

Table 5. Details of the encoder–decoder network structures.

| Name | Kernel | s | Ch. I/O | Input |
|---|---|---|---|---|
| conv1 | 7×7 | 1 | 67/128 | reference image + cost volume |
| conv1_1 | 7×7 | 2 | 128/128 | conv1 |
| conv2 | 5×5 | 1 | 128/256 | conv1_1 |
| conv2_1 | 5×5 | 2 | 256/256 | conv2 |
| conv3 | 3×3 | 1 | 256/512 | conv2_1 |
| conv3_1 | 3×3 | 2 | 512/512 | conv3 |
| conv4 | 3×3 | 1 | 512/512 | conv3_1 |
| conv4_1 | 3×3 | 2 | 512/512 | conv4 |
| conv5 | 3×3 | 1 | 512/512 | conv4_1 |
| conv5_1 | 3×3 | 2 | 512/512 | conv5 |
| upconv4 | 3×3 | 1 | 512/512 | conv5_1(after GP)_up |
| iconv4 | 3×3 | 1 | 1024/512 | conv4_1+upconv4 |
| upconv3 | 3×3 | 1 | 512/512 | iconv4_up |
| iconv3 | 3×3 | 1 | 1024/512 | conv3_1+upconv3 |
| disp3 | 3×3 | 1 | 512/1 | iconv3 |
| upconv2 | 3×3 | 1 | 512/256 | iconv3_up |
| iconv2 | 3×3 | 1 | 513/256 | conv2_1+upconv2+disp3_up |
| disp2 | 3×3 | 1 | 256/1 | iconv2 |
| upconv1 | 3×3 | 1 | 256/128 | iconv2_up |
| iconv1 | 3×3 | 1 | 257/128 | conv1_1+upconv1+disp2_up |
| disp1 | 3×3 | 1 | 128/1 | iconv1 |
| upconv0 | 3×3 | 1 | 128/64 | iconv1_up |
| iconv0 | 3×3 | 1 | 65/64 | upconv0+disp1_up |
| disp0 | 3×3 | 1 | 64/1 | iconv0 |

## B. Additional examples

In addition to those in the main paper, we show additional qualitative comparisons of our method and other methods in Fig. 9. In these example frames, our method predicts noiseless dense depth maps with more details compared with other methods. For example, with our method, the shape of arms of chairs in row 3 and row 4 is more clearer. Moreover, our method provide more accurate prediction in both near and far parts of the scene. For instance, the bag in row 6 and the chair in row 5 show the better performance in close by areas, and the table in row 4 and the fridge in row 6 show the better performance in slightly farther areas. Fig. 13 shows more 3D reconstruction results by applying TSDF fusion on 25 predicted depth maps, which prove that our method have better performance on temporal consistency.

Fig. 12 presents one failure example. As we mentioned, one risk of our method is that wrong predictions can also be propagated forward. In this case, the wrong predictions inside red boxes exist among the first three successive frames, but the erroneous resuts decay away for the latter two frames, as the GP only bring a prior for the latent space and the observations quickly overwhelm it.

## C. Ablation study

In Sec. 4.2, we presented several ablation studies. Here we provide additional qualitative comparisons (to supplement the metrics in the main paper) of different choices of kernel function in Fig. 10 and Fig. 11. We visualize the TD kernel, exponential kernel, and Matérn kernel. The results show that the TD kernel is limited to considering the consecutive two neighbour frames as it uses a different distance metric. Additionally, the Matérn kernel has stronger coupling than exponential kernel. For each example, we show results of three frames, including both near neighbour and far neighbour. We use red lines to label the selected frames in the kernel images. It shows that for the far neighbour (see frame 39 in Fig. 10 and frame 160 in Fig. 11), the results of the TD kernel and w/o GP are worse than the results of exponential kernel and Matérn kernel, as they cannot leverage information from distant past frames though they share similar views. Comparing the exponential kernel and the Matérn kernel, the results of Matérn have sharper edges.

## D. Supplementary video

The project page (https://aaltoml.github.io/GP-MVS) features a supplementary video with example sequences from 7SCENES (*office-04*) and SUN3D (*mit_46_6lounge*). The benefits of the GP model are apparent especially in the cases where the camera stays still. Furthermore, we have included two example sequences captured from our iPad implementation, where the inference runs in real-time on the device.

Note that there are no view selection heuristics and we only need to store the previous frame. The effect of the GP can be seen clearly when the app starts up and the GP first accumulates information over frames.

## E. Inference time

We also evaluated the inference time on our desktop mentioned in Sec. 4: Our method (online) $0.076 \pm 0.003$ s, MVDepthNet $0.066 \pm 0.008$ s, DeepMVS $4.9 \pm 0.1$ s, MVS-Net $3.2 \pm 0.1$ s, and COLMAP $4.5 \pm 0.5$ s. As we discussed in Sec. 4.1, these results proves that the improvement introduced by GP come at almost no additional cost.

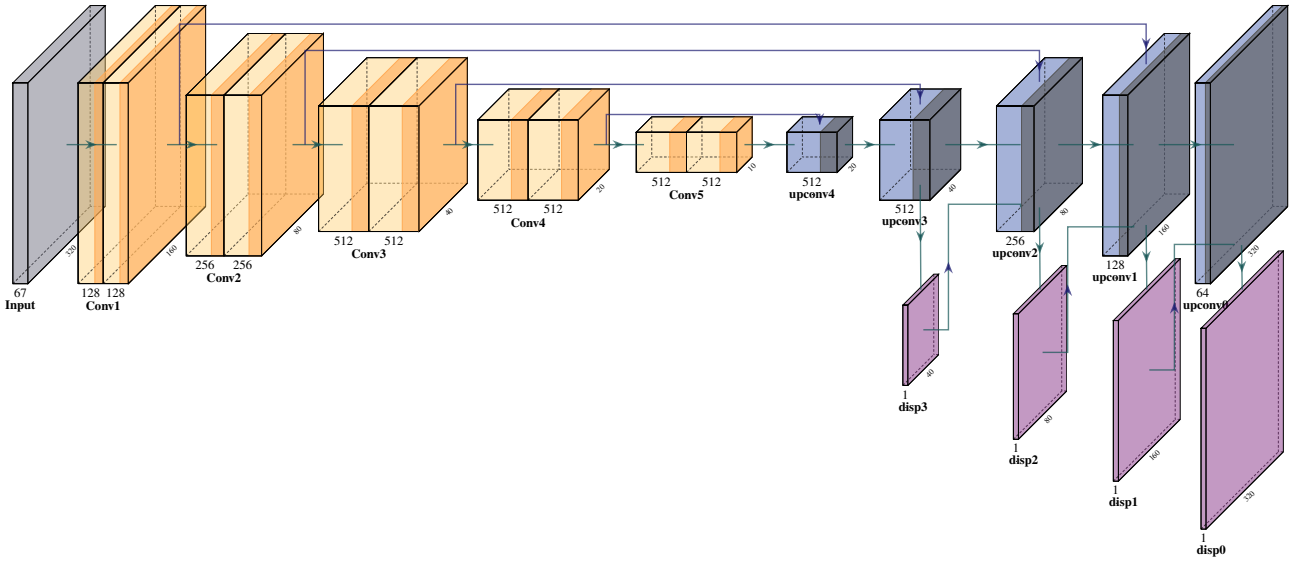



Figure 8. The architecture of the encoder–decoder in our method. The orange blocks are parts of the encoder, and the blue blocks are the decoder. The purple blocks indicate four 'disp*' layers. Except 'disp*' layers, each block are followed by a darker block which refer to the batch normalization and ReLU layers.

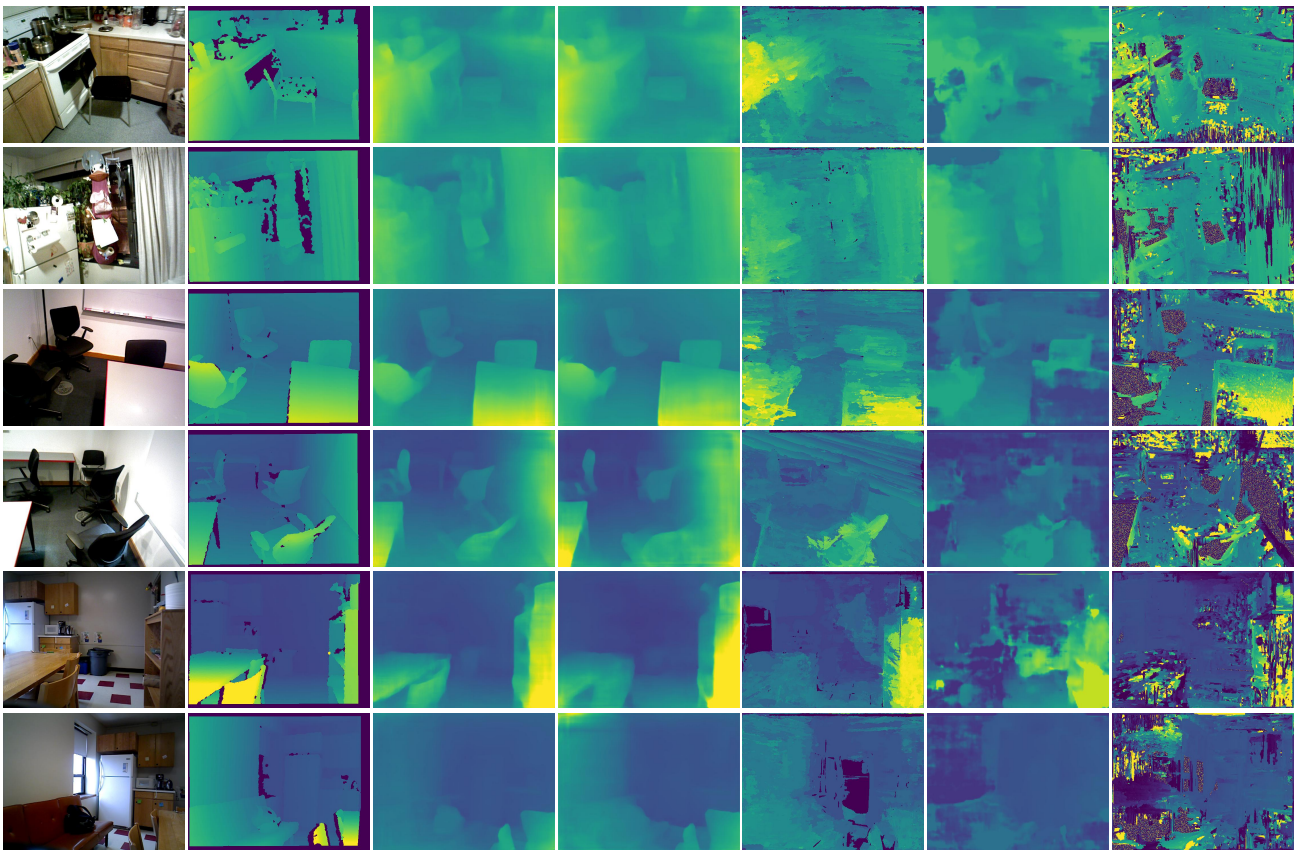



Figure 9. Qualitative results on SUN3D.

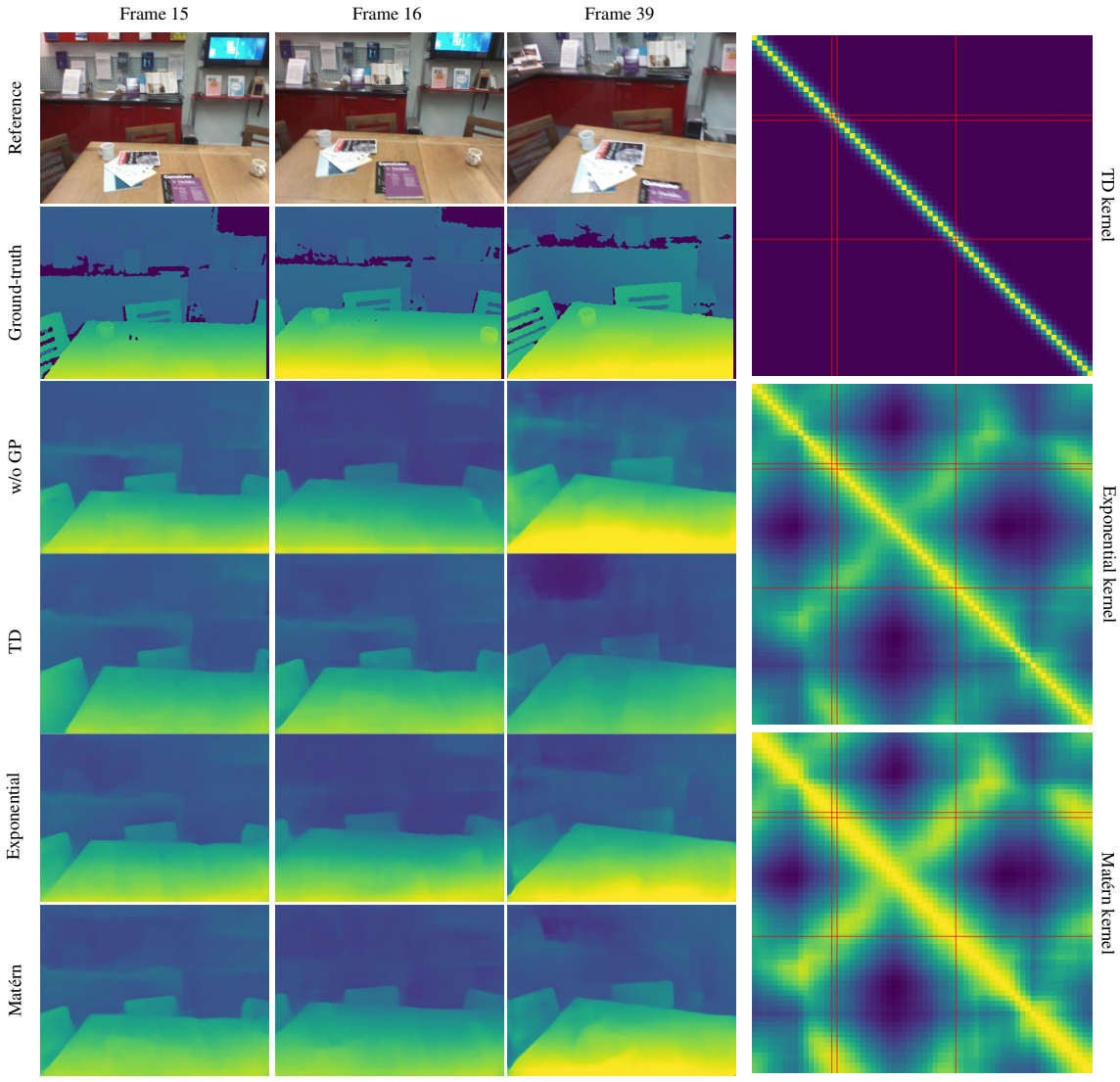



Figure 10. Results comparison of different choices of kernel function on the *redkitchen* sequence in 7SCENES.

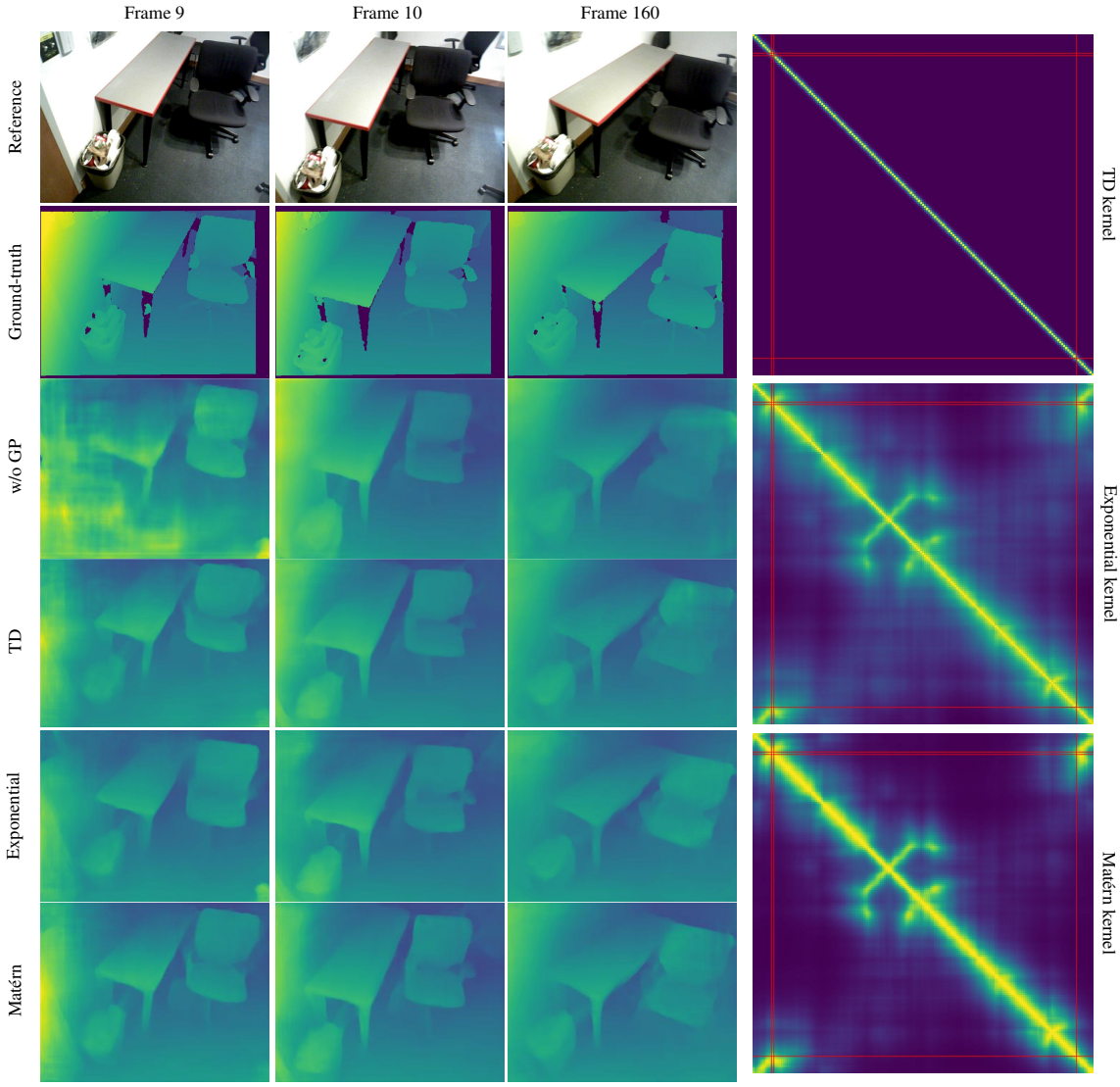



Figure 11. Results comparison of different choices of kernel function on SUN3D.

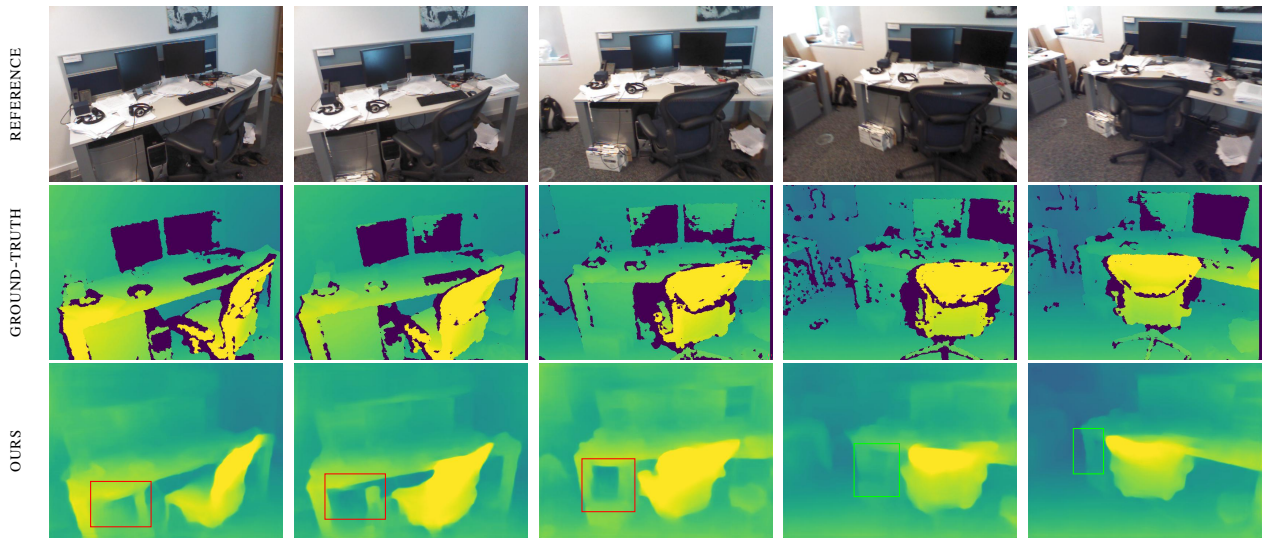
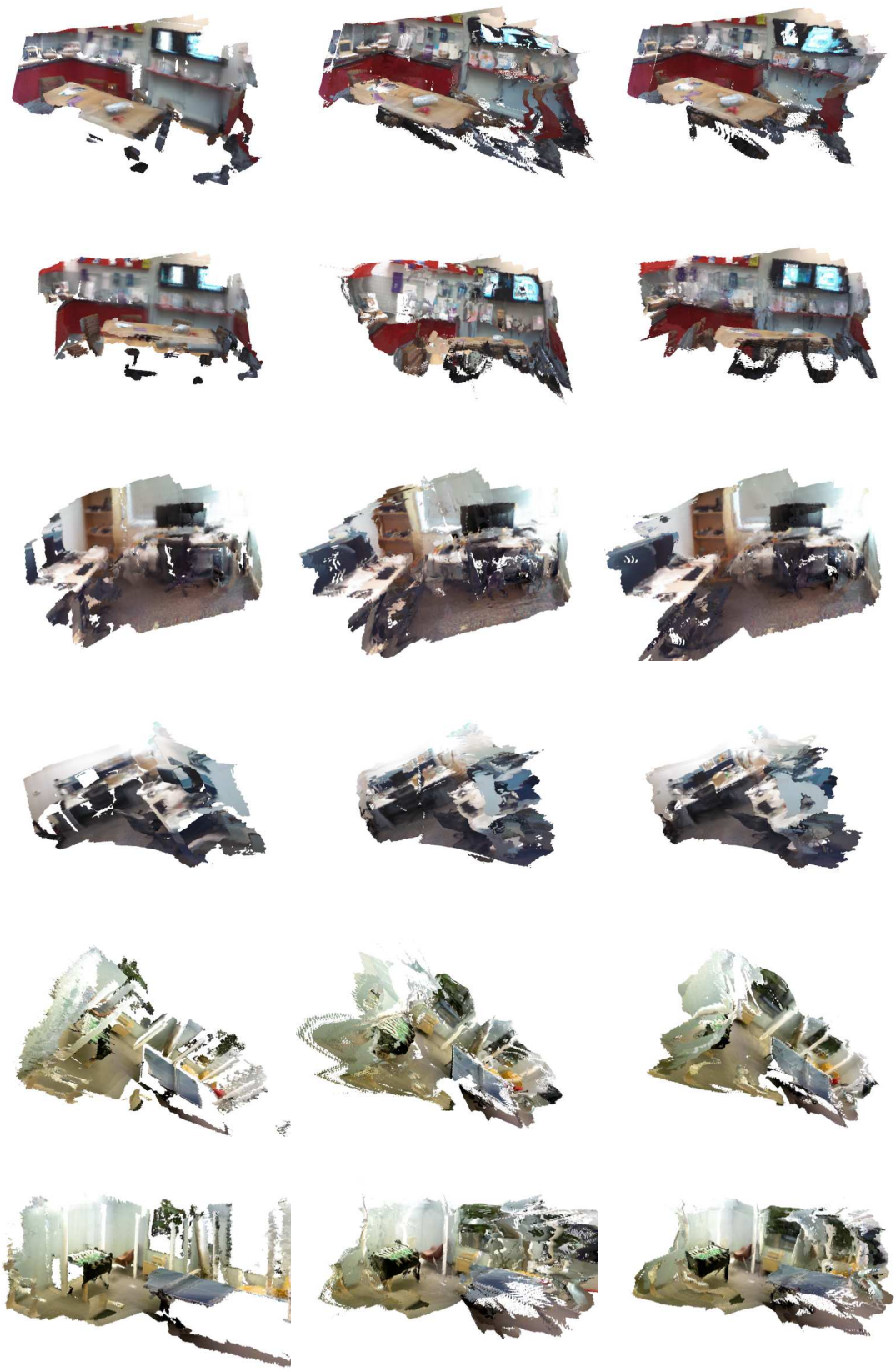



Figure 12. Failure cases example. The wrong predictions might be propagated forward because of the fusion in the latent space. However, as the GP only bring a prior for the latent space, the erroneous depth estimates decay away quickly.



GROUND-TRUTH W/O GP OURS

Figure 13. 3D reconstruction examples. All results are fused from 25 depth maps.