Supplementary: Monocular Neural Image Based Rendering with Continuous View Control

Xu Chen* Jie Song* Otmar Hilliges
AIT Lab, ETH Zurich
{xuchen, jsong, otmar.hilliges}@inf.ethz.ch

In these supplementary materials, accompanying the main paper, we discuss further architecture details and provide additional quantitative and qualitative results.

1. Handling Dis-occlusions or Invisible Parts

When parts of the scene or object are invisible or occluded in the source view, the depth based approach alone can not fully reconstruct the target view. Here we propose a simple extension to our main network, to handle these issues.

1.1. Method

We leverage two additional branches in our architecture that predict a) an image $I_{t,p}$ in the target view directly and b) a mask M for the fusion of depth and image branch predictions such that the final result yields a complete structure. As shown in Fig. 1, both of these predictions are directly decoded from the transformed latent code. The final output is then given by:

$$I_t(x_t, y_t) = M \odot I_{t,d} + (\mathbb{1} - M) \odot I_{t,p}, \quad (1)$$

where $\odot$ denotes the element-wise multiplication.

The weights of this fusion network are optimized via minimization of the L_1 loss between the predicted target view $\hat{I}_t$ and the ground truth I_t. All three branches are trained via a reconstruction loss, applied on intermediate outputs and the final blended images.

$$\mathcal{L}_{recon} = \left\| I_t - \hat{I}_{t,p} \right\|_1 + \left\| I_t - \hat{I}_{t,d} \right\|_1 + \left\| I_t - \hat{I}_t \right\|_1 \quad (2)$$

In addition, to improve realism of the pixel branch, we apply a least square adversarial loss [2] and a perceptual loss [1]:

$$\mathcal{L}_{adv} = (1 - Dis(I_{t,p}))^2 \quad \text{and} \quad \mathcal{L}_{vgg} = \left\| F(I_{t,p}) - F(\hat{I}_{t,p}) \right\|_2 \quad (3)$$

where Dis is a discriminator and F is a pre-trained VGG [4] based feature extractor.

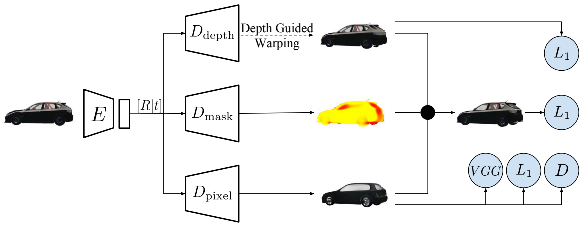

Figure 1: **Architecture overview with fusion.** The architecture combines depth-based (top) and direct pixel predictions (bottom) via a weighted average. Weights are encoded in a per-pixel mask (middle). The network is trained end-to-end in a self-supervised fashion.

1.2. Results

Figure. 2 shows the synthesized views produced by both the pixel and depth branch, as well as the blended results. The network tends to use information from the depth branch as much as possible, and uses the pixel branch to fill-in parts that are not in the source view via a learned image prior. The mask branch correctly predicts the visibility of source view pixels in the target view. This mask can be used not only to fuse results from the pixel and depth branches, but also to combine information from multiple source views when they are available.

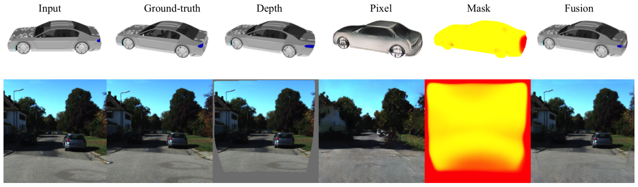

Figure 2: **Channel fusion examples for car and scene.** Images from the pixel branch are in the correct shape but lacks detail. The visibility branch correctly predicts the invisible part. By fusion we yield realistic, detail preserving and complete outputs.

*Equal contribution.

2. Fixed-view Comparison

While our focus is on fine-grained viewpoint control, it is also interesting to evaluate our performance in the original fixed view setting as proposed in [7]. This setting has been commonly used to evaluate previous view synthesis methods. The training data remains the same as described in Section 4.1 (main paper), and during testing the network generates novel views, but only for discrete viewpoints that are present in the training data (car and chair) or along the same trajectory as in training data (KITTI).

As shown in Table. 1, in this fixed-view setting, our method still outperforms other previous methods. The proposed network achieves the best structural similarity results among all previous flow [7, 3] or pixel [6] based methods on both ShapeNet and KITTI datasets. On KITTI dataset flow based methods produce higher L1 error than pixel based method due to invisible parts. The extended fusion network described in Section. 1 (supplementary) outperforms methods that uses completion [3] or fusion [5]. It shows that our T-AE and depth-guided warping improves not only the precision and granularity of viewpoint control, but also the image quality at fixed training views.

Methods	**Car**		**Chair**		**KITTI**	
	L1	SSIM	L1	SSIM	L1	SSIM
Tatarchenko et al.[6]	.139	.875	.223	.882	.295	.505
Zhou et al.[7]	.148	.877	.229	.871	.418	.504
Park et al.[3]	.119	.913	.202	.889	–	
Sun et al.[5]	.098	.923	.181	.895	.203	.626
Ours	.083	.919	.159	.889	.384	.638
Ours (w. fusion)	**.066**	**.932**	**.141**	**.898**	**.191**	**.722**

Table 1: **Quantitative comparison on the fixed view setting as proposed in [7]**

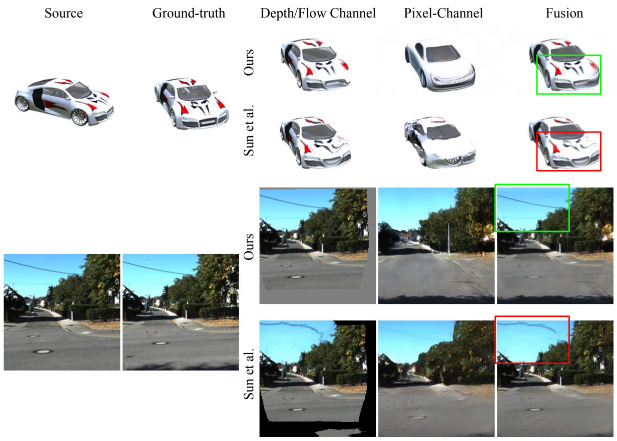

Figure 3: **Direct comparison with [5]** on fixed view synthesis. Ours produces sharp and geometrically correct views (see green highlights). In contrast, [5] can produce blurry results since the method does not reason about geometry (see red highlights).

3. Comparison with Forward Warping

Predicting source view depth and then obtaining the target view by forward warping is in principle another way to achieve the task of view synthesis. Many methods for the prediction of depth from RGB images have been proposed recently and therefore one may be tempted to leverage such methods. However, such an approach would produce holes in the target view as shown In Fig. 4. On the other hand, our method produces dense and geometrically correct target views.

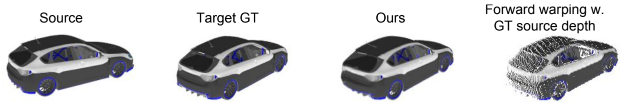

Figure 4: **Comparison with forward warping.** Predicting flow (or depth) in the source view is an easier task (and used in many depth from RGB approaches). However, predicting in source view causes banding artifacts when projecting the point-cloud into an image as seen from a different viewpoint. This causes loss of information in the final result.

4. Flow and Depth

As shown in Table 3 (main paper), the guidance of depth improves the accuracy of flow predictions and consequently improves the accuracy of target views. We show a qualitative example in Fig. 5. Flow-based methods only consider the appearance and hence can warp information from the wrong region in the input. This is solved in our method by the guidance attained from the depth prediction in target view. We also show more results for unsupervised depth predictions 6.

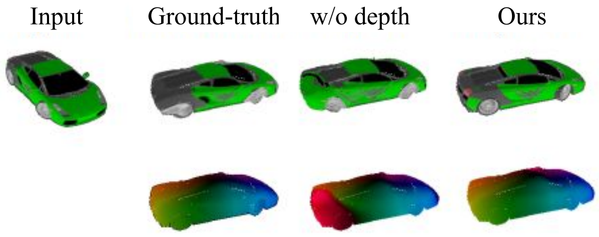

Figure 5: **Comparison of flow maps.** Predictions from our method and flow-based method. *Top row*: final pixel prediction. *Bottom row*: corresponding flow maps.

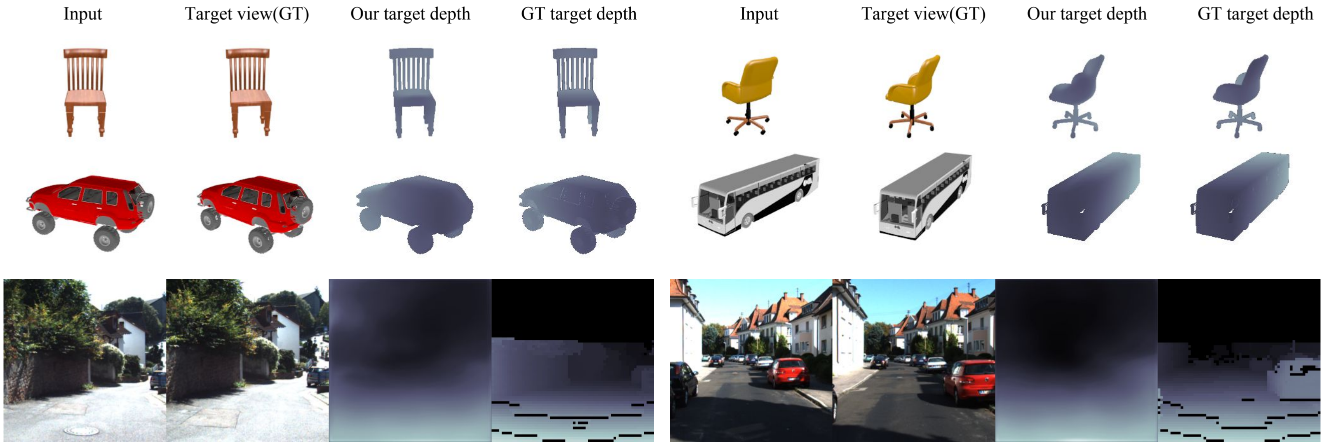

Figure 6: **More results on unsupervised depth predictions.** Our method recovers depth of individual objects and natural scenes faithfully without any direct depth supervision. Note that the predictions on KITTI are smooth but do contain geometric detail (see cars parked on the right hand side).

5. Implementation Details

The encoder consists of seven convolutional layers, each of which downsamples the feature map by 2. Each conv layer is followed by a batch normalization layer and a leaky ReLU layer. The output of the encoder is converted to a vector of length 600 via a fully connected layer, and then reshaped to a 200x3 matrix. The matrix is then multiplied with the rotation and the translation is appended. The transformed code is reshaped to a vector which is fed to the decoder. The decoder consists of seven layers and each one upsamples the feature map by 2 via a bilinear upsampling layer followed by a convolution layer, a batch normalization layer and a leaky ReLU layer respectively.

The original implementation in Zhou et al. [7] and Sun et al. [5] does not support continuous viewpoint input for objects. They represent viewpoint parameters as one hot encoded vectors of length 18, corresponding to 18 fixed azimuth values of training views. To allow for continuous input for comparison, we replace their one hot encoded representation with cosine and sine values of the viewing angles. For the sake of fairness, we use the same encoder and decoder for all compared methods.

6. Additional Qualitative Results

Figure. 7,8,9,10 shows more examples. Figure. 7 and 8 match Figure 3 in the main paper. Figure 9 and 10 match Figure 5 and 6 in the main paper respectively.

References

[1] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 1

[2] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 1

[3] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[4] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[5] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 2, 3, 4, 5, 6, 7

[6] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *Proc. of the European Conf. on Computer Vision (ECCV)*. Springer, 2016. 2, 4, 5

[7] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 2, 3, 4, 5, 6, 7

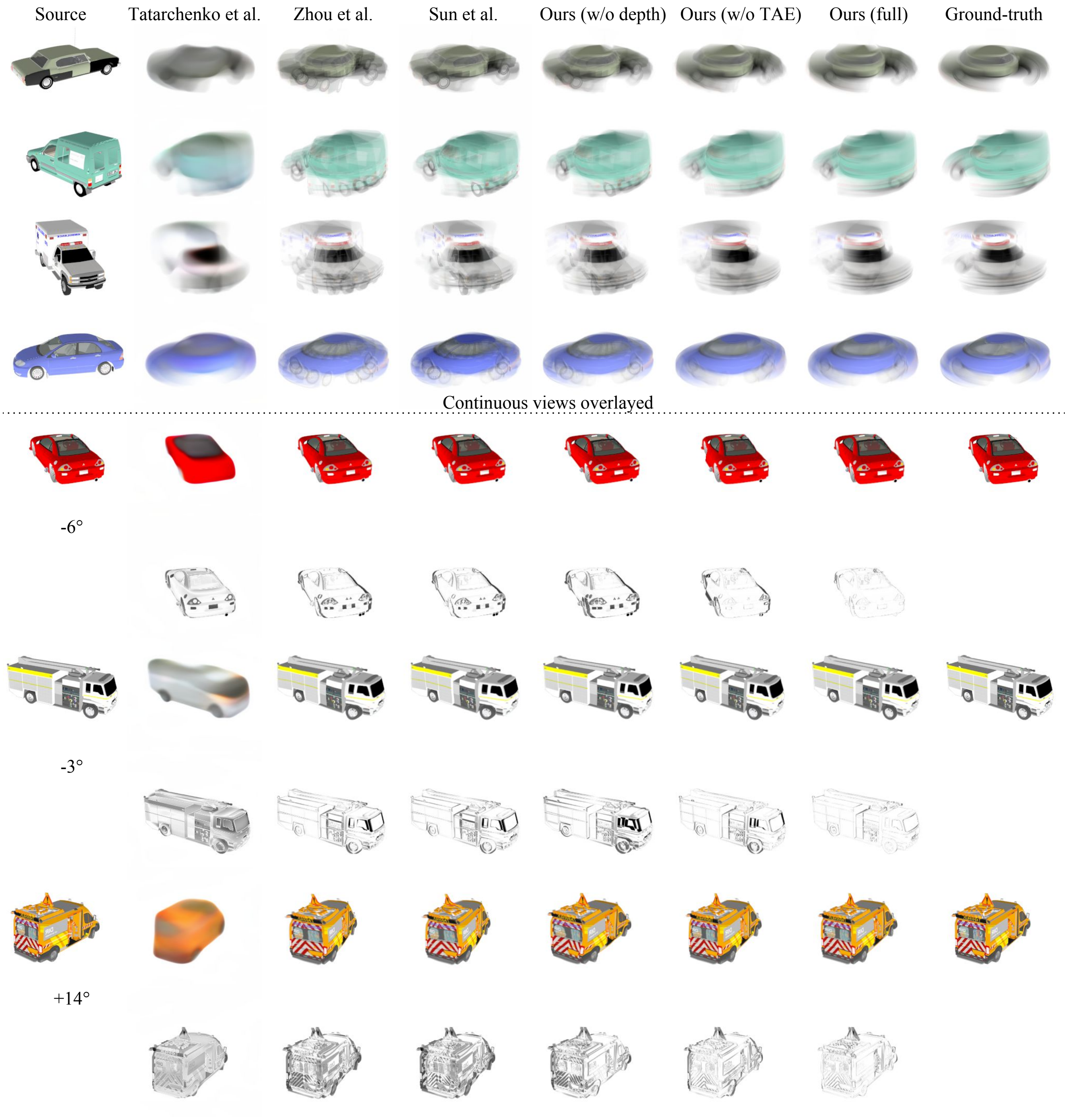

Figure 7: **Qualitative results for granularity and precision of viewpoint control on ShapeNet (Car)** In the top four rows, we generate and overlay 80 continuous views with step size of 1° from a single input. Our method exhibits similar spin pattern as the ground truth, whereas other methods mostly converge to the fixed training views (see wheels of the car and chair indicated in the box). In the bottom, a close look at a specific view is given, which reveals that previous methods display distortions or converge to neighboring training views (Zhou et al.[7], Sun et al.[5]). The image generated by Tatarchenko et al.[6] is heavily blurred. Corresponding error maps are also depicted.

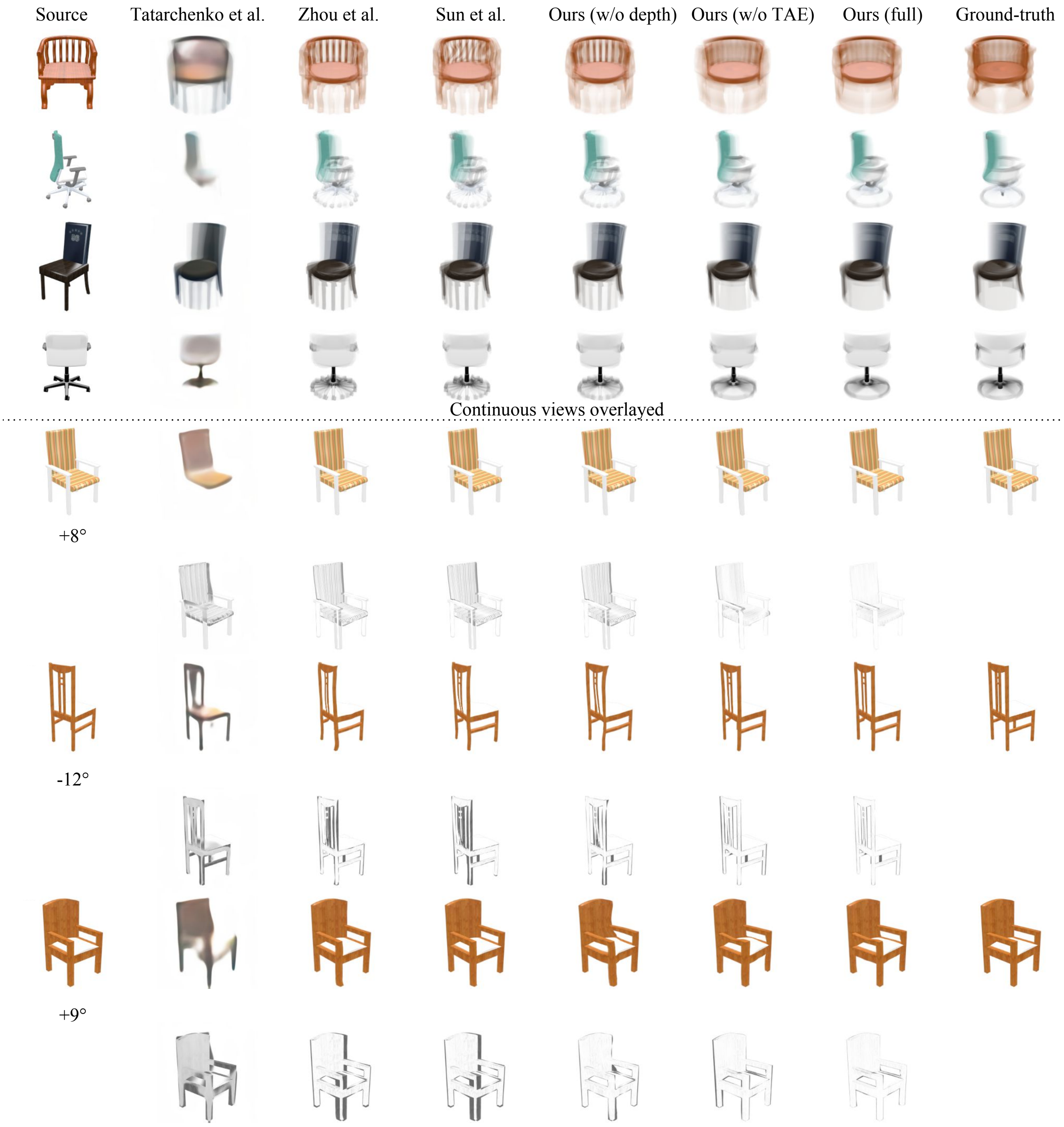

Figure 8: **Qualitative results for granularity and precision of viewpoint control on ShapeNet (Chair)** In the top four rows, we generate and overlay 80 continuous views with step size of 1° from a single input. Our method exhibits similar spin pattern as the ground truth, whereas other methods mostly converge to the fixed training views. The bottom rows, depict individual view angles, which reveal that previous methods display distortions or converge to neighboring training views (Zhou et al.[7], Sun et al.[5]). The image generated by Tatarchenko et al.[6] is heavily blurred. Corresponding error maps are also depicted, consistently showing lower errors for ours.

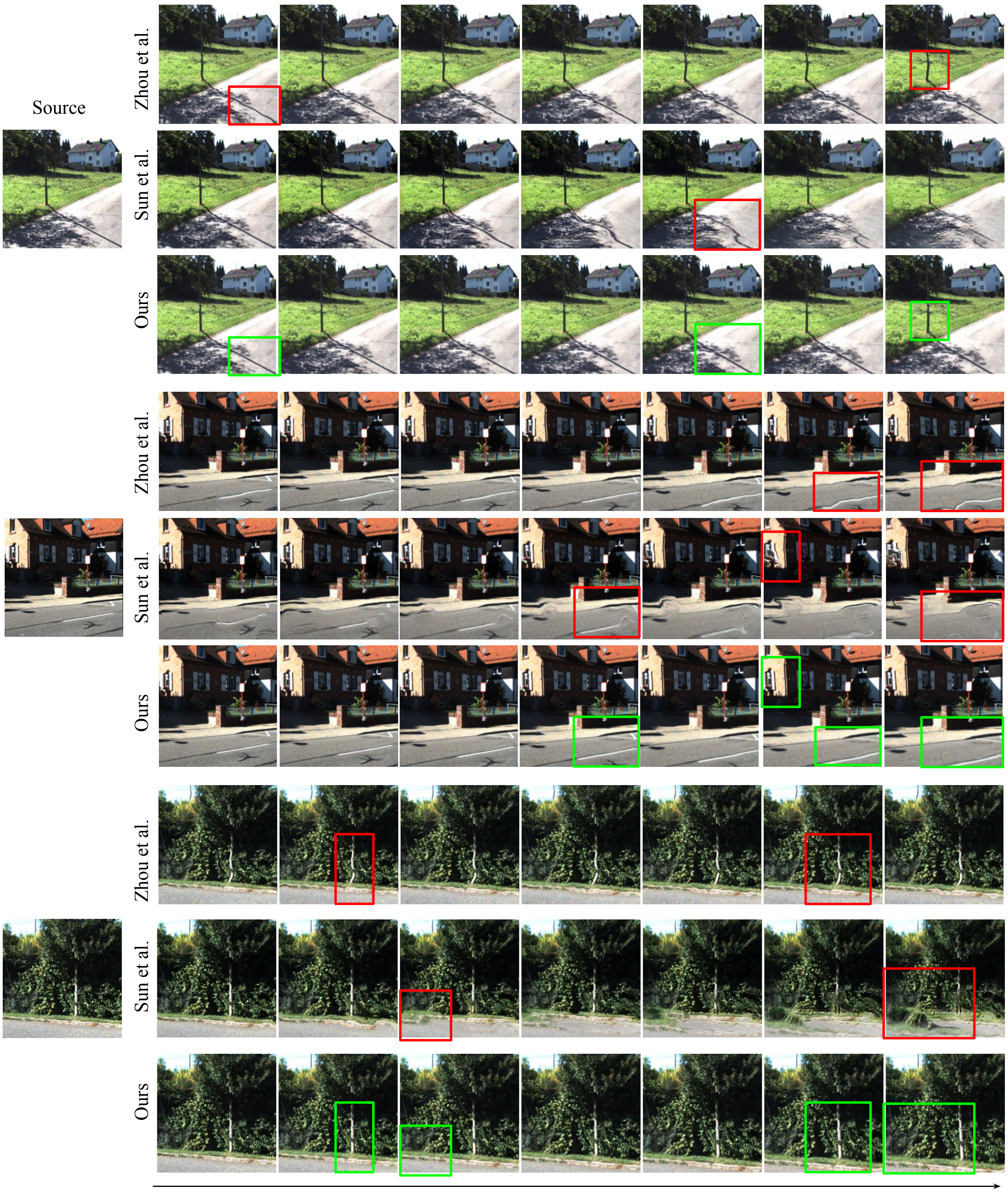

Figure 9: **Qualitative trajectory following results.** Our method produces sharp and correct images while [7, 5] produce distorted images.

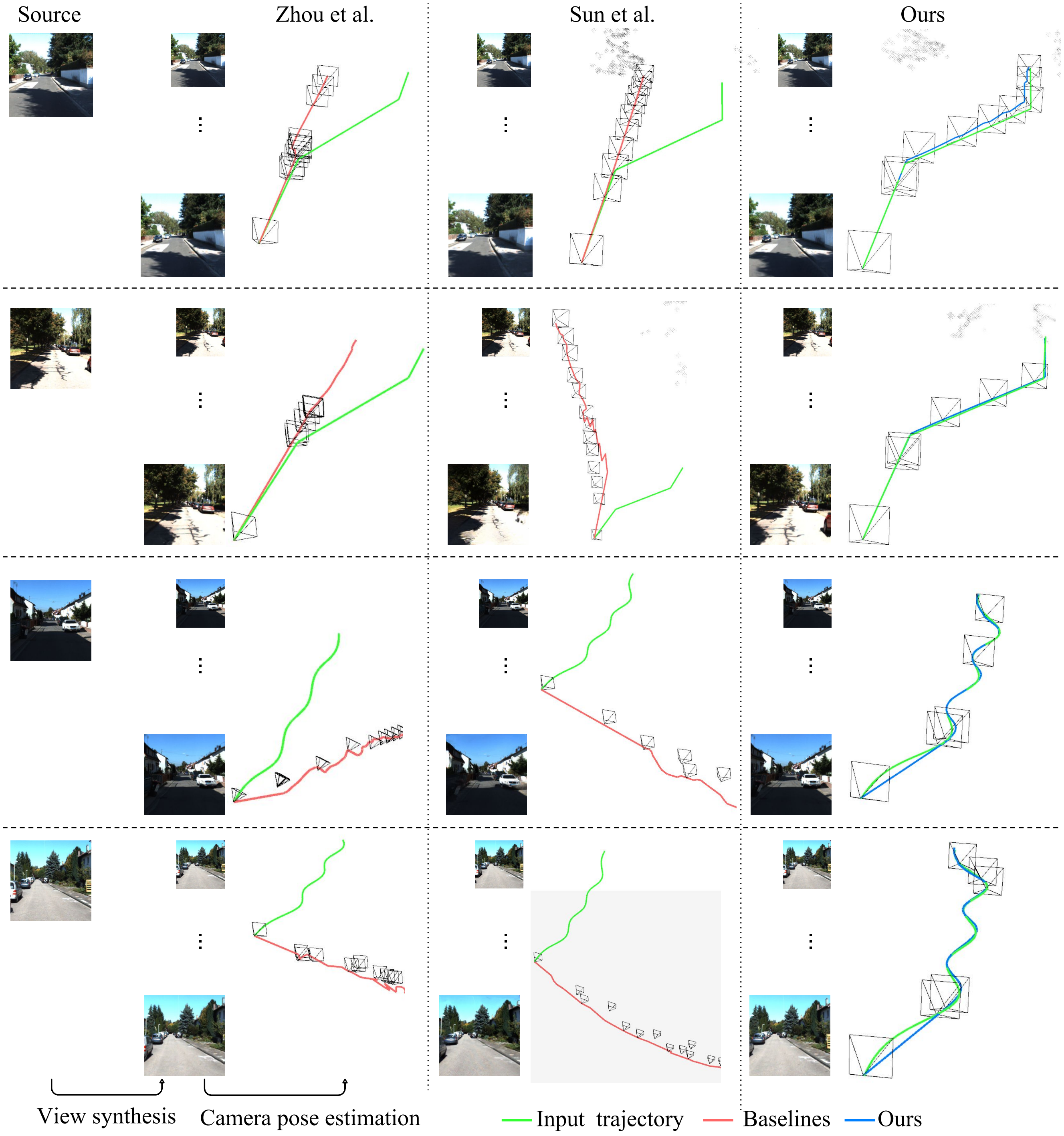

Figure 10: **Additional trajectory recovery results.** Setting: given a source view and an input trajectory, a continuous sequence of views is synthesized along the user defined trajectory (green). Trajectories are estimated via a state-of-the-art visual odometry system [**?**] and compared to the desired trajectory. Right: ours. Left: state-of-the-art [7, 5]. The trajectory estimated from *Ours* align well with the ground-truth, while [7, 5] mostly produce straight forward or wrong motion regardless of the input.

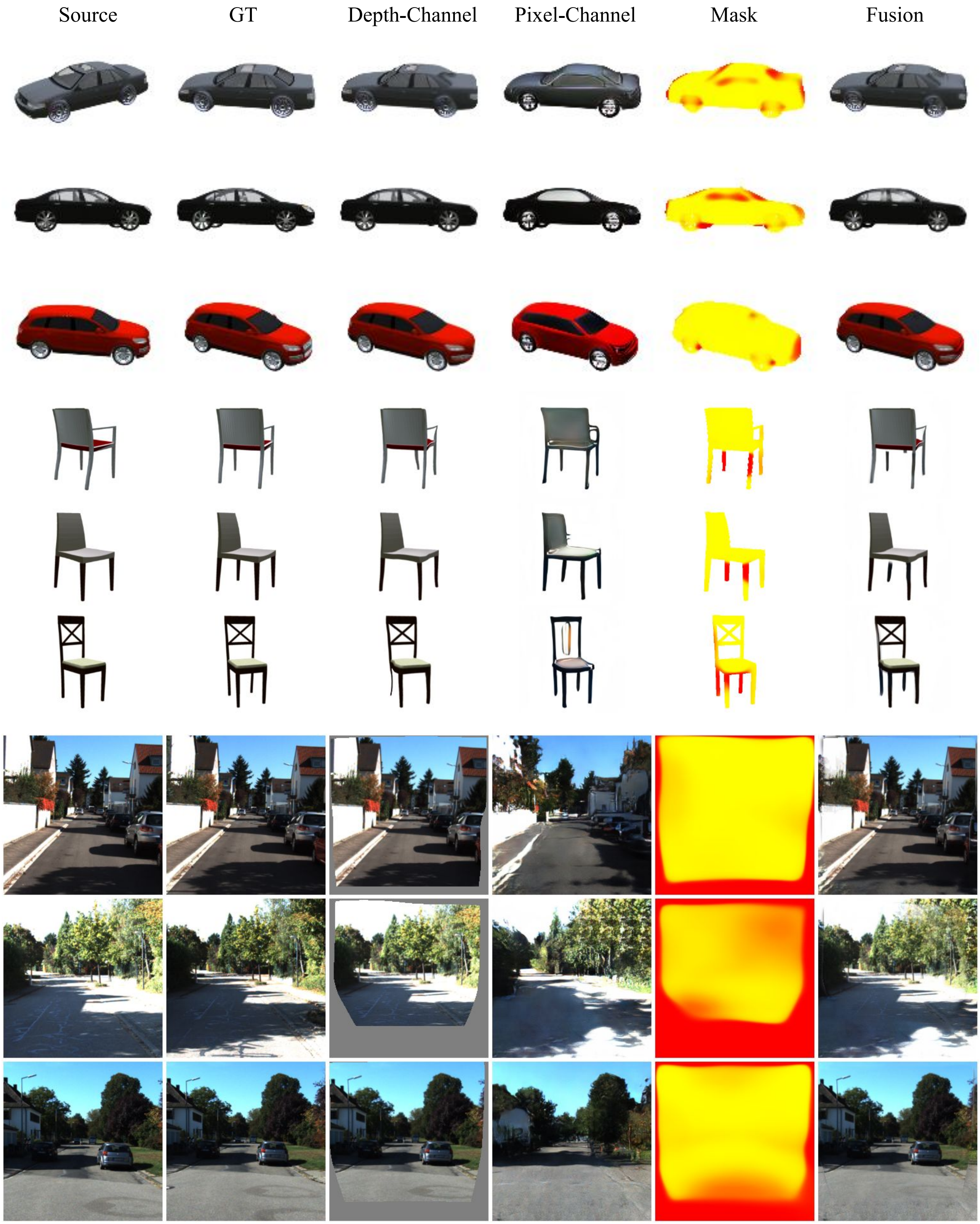

Figure 11: **More fusion results.** In the mask, pixels shown in red indicate high weights for the pixel-channel, whereas pixels in yellow are taken from the depth branch.