ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing (Supplementary Material)

1. Indoor Object Experiment: Rendering Details

We describe additional details regarding the rendering of the SUNCG dataset [8] for our experiment. In addition to Mitsuba [3] for rendering photo-realistic textures, we also utilize the OpenGL toolbox provided by Song *et al*. [8], which supports rendering of instance segmentation.

Candidate object selection. For each of the provided camera viewpoints from Zhang *et al*. [10], we render an instance segmentation of all objects visible in the camera viewpoint. For each of these objects, we also separately render a binary object mask by removing all other existing objects (including the floor/ceiling/walls).

We use these information to exclude objects that are not ideal for our compositing experiment, including those that are too tiny or only partially visible in the camera view. Therefore, we include objects into the candidate selection list that match the criteria:

- The entire object mask is visible within the camera.
- The object mask occupies at least 10% of all pixels.
- At least 50% of the object mask is visible within the instance segmentation mask.
- The object belongs to one of the NYUv2 [7] categories of refrigerators, desks, bookshelves, cabinets, beds, dressers, sofas, or chairs.

Occlusion removal. For all the objects in the candidate list, we remove the occluding objects (from the associated camera viewpoint) by overlapping the object mask onto the instance segmentation mask. All overlapped pixels with different instance labels are detected to be associated with an occluding object. Since there may be "hidden" occlusions that are occluded in the first place, we repeat the same process after the initial detected occlusions are removed to reveal the remaining occlusions. This is repeated until no more occluding objects w.r.t. the candidate object is present.

In order to create a cleaner space for compositing objects, we also use a "thicker" object mask for the above removal procedure. To achieve this, we dilate the object mask with a 3×3 all-ones kernel for 10 times (*i.e*. "thicken" the object mask by 10 pixels).

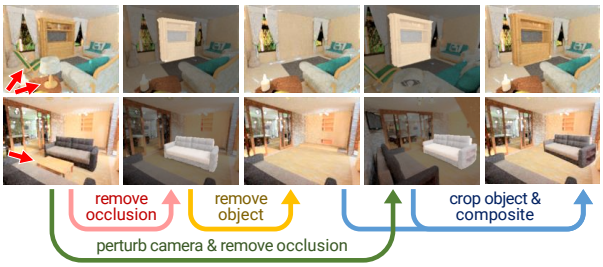

Figure 1: **Rendering pipeline.** Given an indoor scene and a candidate object, we remove occluding objects to create an occlusion-free scenario, which we do the same at another perturbed camera pose. We further remove the object to create a training sample pair with mismatched perspectives.

Camera perturbation. For each of the provided camera viewpoints, we generate a camera perturbation by adding a random 3D-translation sampled from $\text{Unif}(-1, 1)$ in the forward-backward direction, one sampled from $\text{Unif}(-1, 1)$ in the left-right direction (both scaled in meters as defined in the dataset), and a random azimuth rotation sampled from $\text{Unif}(-30, 30)$ (degrees).

After generating a camera perturbation, the same occlusion removal process described above is performed to ensure the wholeness of the object from the perturbed perspective. The candidate object rendered from the perturbed view serves as the foreground source for our experiment. However, if it becomes only partially or not visible, then the rendering is discarded.

Fig. 1 is retaken from the paper for illustration purposes.

Rendering. We use Mitsuba to render 120×160 realistic textures and the OpenGL toolbox to render object masks at 240×320 followed by $\times 2$ downscaling for anti-aliasing.

2. Warp Parameterization Details

We follow Mei *et al*. [6] to parameterize homography with the $\mathfrak{sl}(3)$ Lie algebra. Given a warp parameter vector $\mathbf{p} = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8]^\top \in \mathfrak{sl}(3)$, the transforma-

tion matrix $\mathbf{H} \in \mathbb{SL}(3)$ can be written as

$$\mathbf{H}(\mathbf{p}) = \exp\left(\begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & -p_1 - p_8 & p_5 \\ p_6 & p_7 & p_8 \end{bmatrix}\right) , \qquad (1)$$

where exp is the exponential map (*i.e.* matrix exponential). $\mathbf{H}$ is the identity transformation when $\mathbf{p}$ is an all-zeros vector. Warp composition can thus be expressed as the addition of parameters, *i.e.* $\mathbf{p}_a \circ \mathbf{p}_b \equiv \mathbf{p}_a + \mathbf{p}_b \quad \forall \mathbf{p}_a, \mathbf{p}_b \in \mathfrak{sl}(3)$; furthermore, $\det(\mathbf{H}) = 1 \quad \forall \mathbf{H} \in \mathbb{SL}(3)$.

The exponential map is also Taylor-expandable as

$$\mathbf{H}(\mathbf{p}) = \exp(\mathbf{X}(\mathbf{p})) = \lim_{K \to \infty} \sum_{k=0}^{K} \frac{\mathbf{X}^k(\mathbf{p})}{k!} . \qquad (2)$$

We implement the $\mathfrak{sl}(3)$ parameterization using the Taylor approximation expression with $K = 20$.

3. Training Details

For all experiments, we set the batch size for all experiments to be 20. Unless otherwise specified, we initialize all learnable weights in the networks from $\mathcal{N}(0, 0.01)$ and all biases to be 0. All deep learning approaches are trained with Adam optimization [4]. We set $\lambda_{\text{grad}} = 10$ following Gulrajani *et al*. [2].

We describe settings for specific experiments as follows.

3D cubes. We create 4000 samples of 3D cube/room pairs with random colors, as described in the paper. For the initial warp $\mathbf{p}_0$, we generate random homography perturbations $\mathbf{p}_0$ by sampling each element of $\mathbf{p}_0$ from $\mathcal{N}(0, 0.1)$, *i.e.* $\mathbf{p}_0 \sim \mathcal{N}(\mathbf{0}, 0.1\mathbf{I})$. This is applied to a canonical frame with x and y coordinates normalized to $[-1, 1]$ and subsequently transformed back to the image frame. We train ST-GAN with 4 sequential warps, each for 50K iterations (with perturbations generated on the fly) with the learning rates for both $\mathcal{G}$ and $\mathcal{D}$ to be 10^{-4}. We set $\lambda_{\text{update}} = 0.1$ in this experiment.

Indoor objects. For the self-supervised baselines (HomographyNet [1] and SDM [9]), we generate random homography perturbations $\mathbf{p}_0$ using the same noise model as that from the 3D cubes experiment.

We train HomographyNet for 200K iterations (with perturbations generated on the fly) with a learning rate of 10^{-4}. For SDM, we vectorize the grayscale images to be the feature as was practiced for image alignment [5]; in our case, we concatenate those of the background and masked foreground as the final extracted feature. We generate 750K perturbed examples (more than 10 perturbed examples per training sample) to train each linear regressor. Also as was practiced [9, 5], we add an ℓ_2 regularization term to the SDM least-squares objective function and search for the penalty factor by evaluating on a separate validation set.

We initialize each of the ST-GAN generators $\mathcal{G}_i$ with the pretrained HomographyNet as we find it to be better-conditioned. During adversarial training, we train each $\mathcal{G}_i$ for 40K iterations with the learning rate for $\mathcal{G}_i$ to be 10^{-6} and that of $\mathcal{D}$ to be 10^{-4}. In the final end-to-end fine-tuning stage, we train all $\mathcal{G}_i$ for 40K iterations using the same learning rates (10^{-6} for all $\mathcal{G}_i$ and 10^{-4} for $\mathcal{D}$). The non-sequential ST-GAN baseline is trained for 160K iterations with the same learning rates. We set $\lambda_{\text{update}} = 0.3$ in this experiment.

Glasses. For data augmentation, we perturb the faces with random similarity transformations from $\mathcal{N}(0, 0.1)$ for rotation (radian) and $\mathcal{N}(0, 0.05)$ for translation (scaled by the image dimensions, in both x and y directions). The glasses are perturbed using the same random homography noise model as used in the 3D cubes experiment.

We train ST-GAN with 5 sequential warps, each for 50K iterations with the learning rates for both $\mathcal{G}$ and $\mathcal{D}$ to be 10^{-5}. As a preconditioning step, we also pretrain the discriminator $\mathcal{D}$ using only the initial fake samples and real samples for 50K iterations with the same learning rate. We set $\lambda_{\text{update}} = 1$ in this experiment.

4. Additional Indoor Object Results

We include additional qualitative results from the indoor object experiment in Fig. 2. Compared to the baselines, ST-GAN consistently predicts more realistic geometric corrections in most cases.

5. Additional Glasses Results

We also include additional qualitative results from the glasses experiment in Fig. 3. We re-emphasize that the training data here is unpaired and there is no information in the dataset about where the glasses are placed. Despite these, ST-GAN is able to consistently match the initial glasses foreground to the background faces.

References

[1] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 2

[3] W. Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 1

[4] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[5] C.-H. Lin, R. Zhu, and S. Lucey. The conditional lucas & kanade algorithm. In *European Conference on Computer Vision*, pages 793–808. Springer, 2016. 2

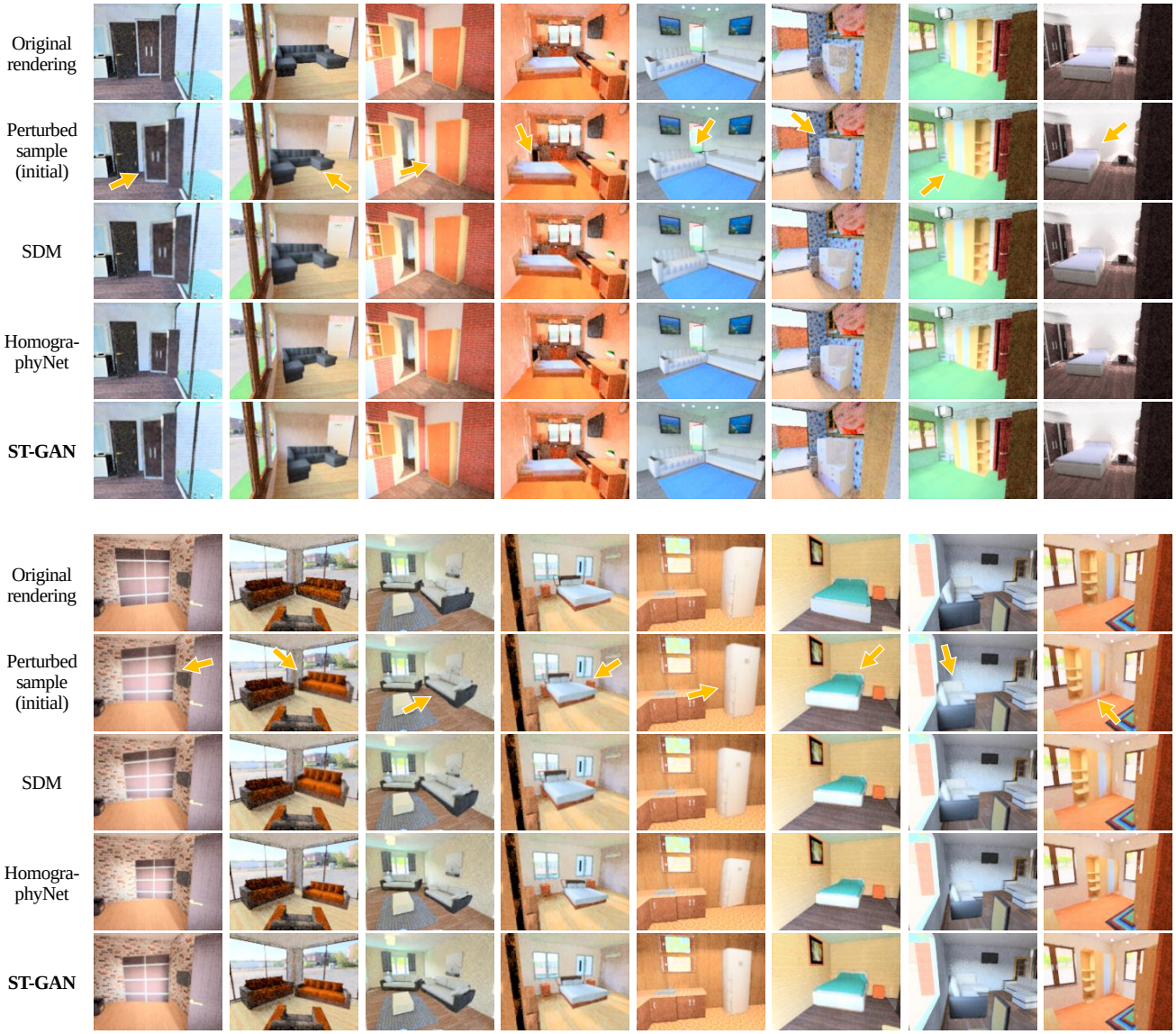

Figure 2: Additional qualitative results from the indoor object experiment (test set). The yellow arrows in the second row point to the composited foreground objects.

[6] C. Mei, S. Benhimane, E. Malis, and P. Rives. Homography-based tracking for central catadioptric cameras. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 669–674. IEEE, 2006. 2

[7] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *Computer Vision–ECCV 2012*, pages 746–760, 2012. 1

[8] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[9] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013. 2

[10] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

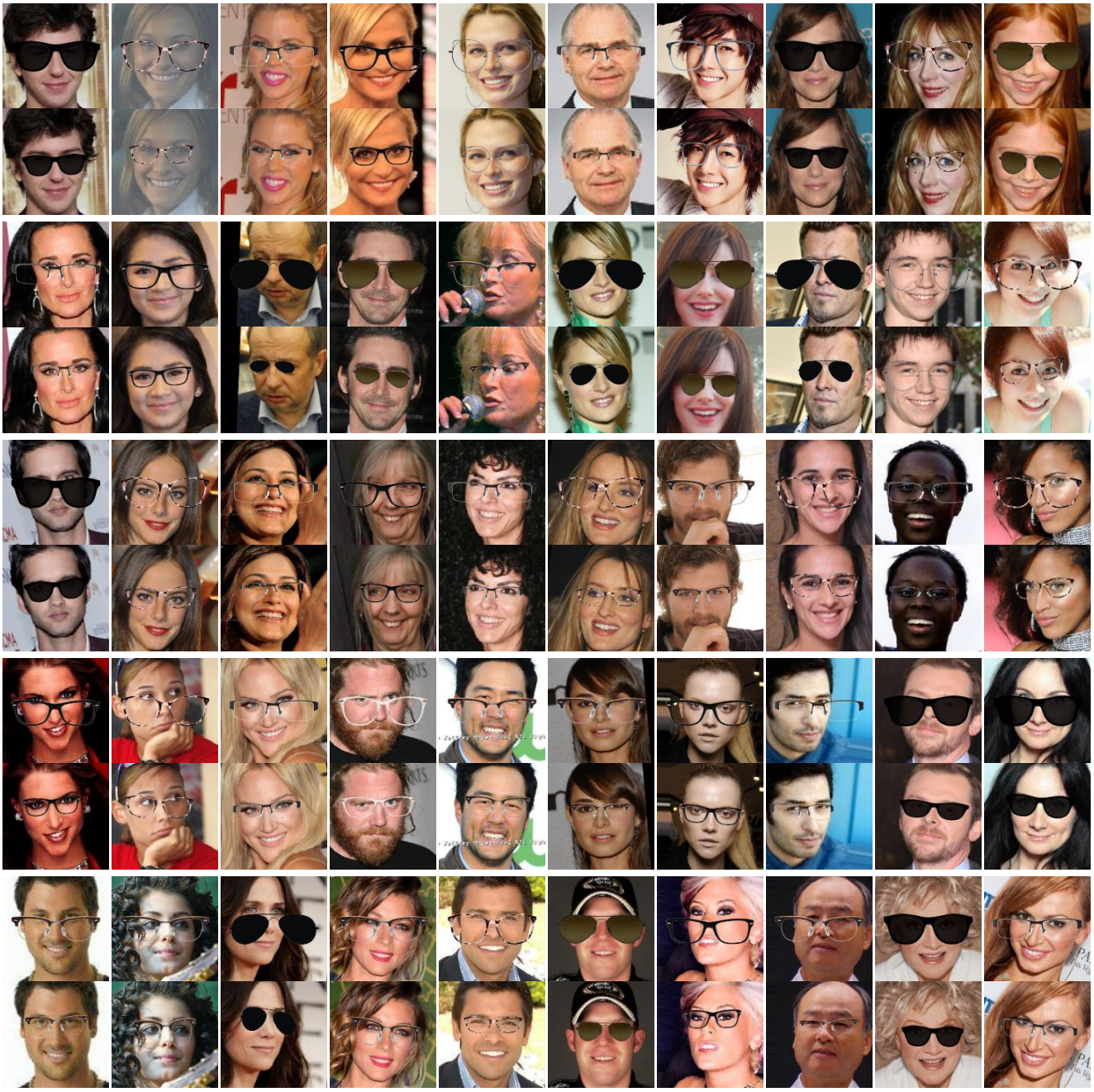

Figure 3: Additional qualitative results from the glasses experiment (test set). The top row indicates the initial composite, and the bottom row indicates the ST-GAN output.