GeoFill: Reference-Based Image Inpainting with Better Geometric Understanding
–Supplementary Material–

Yunhan Zhao[1]*, Connelly Barnes[2], Yuqian Zhou[2,3], Eli Shechtman[2], Sohrab Amirghodsi[2], Charless Fowlkes[1]
[1]UC Irvine [2]Adobe Research [3]IFP, UIUC
{yunhaz5, fowlkes}@ics.uci.edu {cobarnes, elishe, tamirgho}@adobe.com yuqian2@illinois.edu

Outline

In the supplementary document, we provide additional ablation studies to further support our findings, as well as details of our experiments and more visualizations. Below is the outline.

- **Section 1: Convergence criteria.** We describe the convergence criteria of our optimization step in detail.
- **Section 2: Joint optimization evaluation: inpainting performance.** We demonstrate the importance of the joint optimization module by comparing the performance of GeoFill with parameters estimated before and after the joint optimization.
- **Section 3: Joint optimization evaluation: depth and pose accuracy.** We further evaluate the performance of the optimization module by measuring the accuracy of depth maps and camera poses individually against ground-truth using ScanNet.
- **Section 4: Performance w.r.t intrinsic parameters.** Quantitative comparisons of our approach with different focal lengths.
- **Section 5: Further analyses of initial alignments.** We provide additional qualitative and quantitative evaluations of GeoFill and TransFill without the CST module.
- **Section 6: Visualizations of hole ablation study.** We include qualitative comparisons between GeoFill and TransFill under various hole sizes.
- **Section 7: Average running time of GeoFill.** We report the average running time of each step in GeoFill.
- **Section 8: User study against other baselines.** A user study of GeoFill against OPN, ProFill, and TransFill.
- **Section 9: Handling appearance changes from camera movement.** We show qualitative results of GeoFill handling some common appearance changes due to camera movement.
- **Section 10: Failure cases.** Visual examples of failure cases of GeoFill.
- **Additional Visual Results.** We include more inpainting results in Fig. 6 and 7.

*Work done while an intern at Adobe.

1. Convergence Criteria

The convergence criteria define when the optimization should stop. Our optimization halts the loop at a given scale and continues to the next scale if the following condition is met or the predefined maximum number of iteration is achieved. The formula below measures the objective function value changes within the last m iterations.

$$\epsilon_i = \frac{|\sum_{i-(m/2)-1}^{i} l_i - \sum_{i-(m/2)}^{i-m-1} l_i|}{\sum_{i-(m/2)-1}^{i} l_i}, \tag{1}$$

where i represents the i^{th} iteration. If ϵ_i is smaller than a predefined ϵ_{opt}, we assume the objective function has converged. Since we adopt a coarse-to-fine optimization strategy, we check the same condition at every level of the pyramid. In other words, we move to the finer scale level only if Eqn. 1 is met or maximum number of iterations at the current level is reached. We also keep track of the optimal parameters at each level and use them as the initialization in the next level. In practice, we set the convergence threshold ϵ_{opt} to 10^{-6} for all levels. The number of loss values to track in computing convergence criteria is $m = 10$.

2. Joint Optimization Evaluation: Inpainting Performance

We show the importance of our optimization module by comparing the performance of GeoFill with initial estimated parameters and optimized parameters. As shown in Table 1, GeoFill with optimized parameters has substantially better performance. Initial parameters are computed from SIFT and pretrained models such as OANet, which can make erroneous predictions, especially for image pairs with holes. Experimental results demonstrate our optimization module successfully mitigates such errors and improves the inpainting performance.

Table 1: Quantitative comparison of our method with initially estimated parameters and optimized parameters.

Model	PSNR↑	SSIM↑	LPIPS↓
GeoFill (optim)	**31.47**	**0.9748**	**0.0525**
GeoFill (init)	30.66	0.9719	0.0548

Table 2: Relative camera pose evaluation of initial guess and our optimized results, where R and t represent the rotation and translation, respectively.

	$R\downarrow$		$t\downarrow$	
	mean (°)	med (°)	mean (°)	med (°)
GeoFill (optim)	**1.588**	**1.062**	**3.688**	**3.457**
GeoFill (init)	7.378	7.807	11.861	11.096

3. Joint Optimization Evaluation: Depth, Pose Accuracy

In this section, we further demonstrate the effectiveness of our joint optimization step by measuring the depth and pose accuracy before and after the optimization. Since both RealEstate10K and MannequinChallenge do not have ground-truth labels, we choose the ScanNet [2] dataset which comes with ground-truth camera poses and depth maps. We randomly sampled 75 pairs of images with approximately 30 frame difference. We generate random holes in the same manner as described in our main paper line 388. Each image pair comes from a unique scene in the dataset. Note that ScanNet includes images with heavy motion blur, which we manually filtered out. We evaluate depth and relative camera pose *separately* by providing the ground-truth for one of these (depth or camera pose) when evaluating the accuracy of the other one. For example, when evaluating the accuracy of depth maps, we first follow the same pipeline described in the main paper. Then, instead of estimating the relative pose, we provide the ground-truth camera pose and evaluate the accuracy of the depth map determined by our pipeline before and after the optimization. Note that we only optimize scale and offset when evaluating depth accuracy. A similar analogy applies when evaluating the accuracy of camera poses: we provide the ground truth depth map to our pipeline and then evaluate the accuracy of the relative pose before and after optimization. For pose evaluation, we report the geodesic errors [1] for both rotations and translation directions. For depth evaluation, we follow the commonly adopted metrics used in the literature [3, 4]. As shown in Table 2 and Table 3, both depth and pose errors are significantly reduced after the optimization module, which demonstrates the ability of the optimization module to find more accurate depth and poses in our challenging case where the images have holes.

4. Performance w.r.t Intrinsic Parameters

GeoFill handles incoming image pairs using fixed camera intrinsic parameters instead of explicitly knowing the ground-truth camera intrinsic parameters. In the main paper, we use fixed camera intrinsic parameters by setting the focal length of all images to 750 pixels and the principal point to the center of the image. It is intuitive to set the principal point to the center of the images with unknown intrinsic parameters, therefore, we focus on studying the effect of focal lengths. We compare the performance of GeoFill with the camera focal lengths of 600, 750, 900, 1050, and 1200 pixels. As shown in Fig. 1, GeoFill with different focal lengths has very slight differences in terms of PSNR. There is a slight trend that the performance drops as the focal length increases. We believe this indicates that the ground-truth focal length is close to 600 and higher focal lengths make the optimization have a harder time finding improved relative poses. Nevertheless, GeoFill can still adapt to different focal lengths by jointly optimizing depth scale, offset, and relative pose, therefore, it still can render similar images across a variety of focal lengths.

Table 3: Depth evaluation of our initial guess and optimized results. The evaluation metrics include absolute relative difference (Abs^r), squared relative difference (Sq^r), root mean squared log error (RMS-log), and accuracy with a relative error threshold of $\delta^k < 1.25^k$, k = 1, 2.

Models	Abs^r ↓	Sq^r ↓	RMS-log↓	δ^1 ↑	δ^2 ↑
GeoFill (optim)	**.258**	**.233**	**.302**	**.683**	**.851**
GeoFill (init)	.372	.766	.403	.609	.788

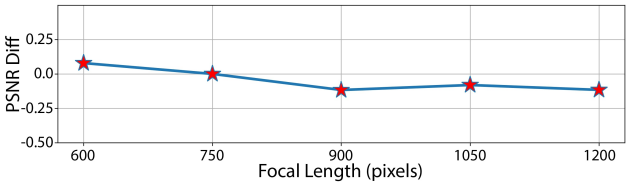

Figure 1: Visual plots showing the performance of GeoFill with different focal lengths. PSNR diff is computed by using GeoFill with new focal length subtract GeoFill with focal length equals to 750.

Table 4: Initial alignment comparisons of our method compared to TransFill without the CST Module on MannequinChallenge dataset (FD=10).

Model	PSNR↑	SSIM↑	LPIPS↓
GeoFill	**28.85**	**0.9702**	**0.0553**
GeoFill (no CST)	26.84	0.9626	0.0615
TransFill	28.01	0.9680	0.0569
TransFill (no CST)	24.04	0.9526	0.0760

5. Further Analyses of Initial Alignments

GeoFill adopts the CST module from TransFill to further improve any small residual spatial misalignments and correct color and exposure differences. We first visually compare the quality of our single proposal to the merged proposal from TransFill without the CST on RealEstate10K

Figure 2: Qualitative comparisons of our approach against TransFill with different hole sizes. **Please zoom in** to see that ours looks good but there are broken structures, ghosting, and distortion artifacts in TransFill.

dataset. As shown in Fig. 3, the single proposal from GeoFill is significantly more accurate than merged heuristic proposals from TransFill, demonstrating the superiority of our approach over TransFill. Additionally, we also show the quantitative comparisons of GeoFill and TransFill without the CST module on the MannequinChallenge dataset. As shown in Table 4, we find GeoFill without the CST module has a huge advantage over TransFill merged homographies.

6. Visualizations for Hole Ablation Study

In the main paper, we show the quantitative comparisons between our approach and TransFill with various hole sizes. Here, we provide visual comparisons to better understand the performance boost for larger holes. We simulate larger holes by generating the same hole shape with larger stroke width. As shown in Fig. 2, GeoFill has a robust performance while TransFill has ghosting artifacts and misalignments as the hole grows larger.

Table 5: User study results of GeoFill against ProFill, OPN, and TransFill.

	Filtered		Non-Filtered	
Model	PR	p-value	PR	p-value
ProFill	100%	$p < 10^{-6}$	96.25%	$p < 10^{-6}$
OPN	97.37%	$p < 10^{-6}$	95.00%	$p < 10^{-6}$
TransFill	70.90%	$p < 2 \times 10^{-3}$	68.13%	$p < 2 \times 10^{-3}$

7. Average Running Time of GeoFill

We randomly sampled 50 images at 1280x720 pixels and compute the average time of each step. Monocular depth estimation takes 3.83s, sparse correspondence estimation takes 0.596s, triangulation takes 0.0009s, initial relative pose takes 0.0052s, joint optimization takes 58.2s, mesh rendering takes 1.03s, refinement and merging step takes 2.53s. The reported time uses default parameters described in the experiment section. Although the joint optimization step takes up the vast majority of the time, its current implementation is naive and not optimized. If desired, various

Figure 3: Visual comparisons of GeoFill and TransFill without the CST module.

Figure 4: Visual examples of failure cases of GeoFill.

engineering optimizations could be made such as using a custom kernel with proper low-level optimizations such as fusion for the renderer instead of a naive pure PyTorch implementation, using FP16 mode, using only the sparse edge map pixels during optimization (these are quite sparse so significant acceleration should be possible), using second-order optimization techniques that could potentially converge in fewer steps, carefully tuning input resolution, number of pyramid levels, iteration limits, break thresholds, etc. We considered these to be lower-level engineering details that we did not focus on in our paper's implementation, since we were focusing more on research aspects.

8. User Study

To better evaluate the performance of GeoFill against other baselines, we conduct a user study via Amazon Mechanical Turk (AMT). We compare our method against OPN, ProFill, and TransFill by showing users image pairs with binary choice questions. The users are requested to choose the inpainting results that look more realistic and faithful. To improve the quality of collected data, we adopt a qualification test with trivial questions to filter noisy results. For each method pair, we randomly sampled 80 examples in RealEstate10K dataset with FD=50, and each example was evaluated by 7 independent users. We present

Figure 5: Qualitative results of GeoFill handling some common appearance changes such as in white balance and exposure due to camera movement.

two approaches to computing the preference rate. The first one is the filtered approach, in which we filter the responses to retain only those where one method is "preferred" if 6 or more users select it. The filtering helps suppress noise in the responses of Mechanical Turk workers, whose work quality can vary. The second one is the non-filtered approach where we retain all responses and choose the method as "preferred" where a simple majority of 4 or more users select it. We reported GeoFill's Preference Rate (PR) in Table 5. GeoFill has much higher preference rates against OPN and ProFill. Compared against TransFill, we receive a PR around 70% on filtered and non-filtered approaches. TransFill is still very robust on small holes and relatively small camera motions in the randomly sampled data. Therefore, GeoFill is favored by users over TransFill but less strongly than in the other comparisons. We performed a one sample permutation t test with 10^6 simulations using the null hypothesis that each pair are preferred equally by users: the p-values are all sufficiently small that the preference for our method is statistically significant.

9. Handling Appearance Changes from Camera Movement

As we stated in the main paper, we focus on the common scenario of capturing photos with the same camera *freely* moving around. However, there are potential appearance changes of the same parts of the scene due to the camera movement, for example, changes due to automatic exposure or automatic white balance between source and target images. This is a common yet non-trivial challenge when applying GeoFill in real-world applications. In this section, we show some visual examples of image pairs with appearance changes in the dataset. As shown in Figure 5, GeoFill still inpaints plausible results even when the appearance of the same part of the scene is different between source and target images.

10. Failure Cases

We show some failure cases of GeoFill under extreme conditions. Fig. 4 shows three common failure cases of GeoFill. The image pair on the left contains transparent surfaces in the images. These objects often cause monocular depth estimators to fail and can lead to bad optimization results. In the second failure case, the drastic changes in the lighting environment affect the feature correspondence matching and depth prediction, which makes the final result from GeoFill less accurate. In the last case, dynamic objects, e.g., pedestrians, make our optimization module estimate inaccurate parameters. We discuss in the last section of our main paper ways that future work might address these issues.

Additional Visual Results

We include additional qualitative comparisons of GeoFill against other baselines in Fig. 6. Additionally, we also show the inpainting performance of GeoFill on user-provided images, RealEstate10K, and MannequinChallenge dataset in Fig. 7.

References

[1] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3258–3268, 2021.

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[3] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[4] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

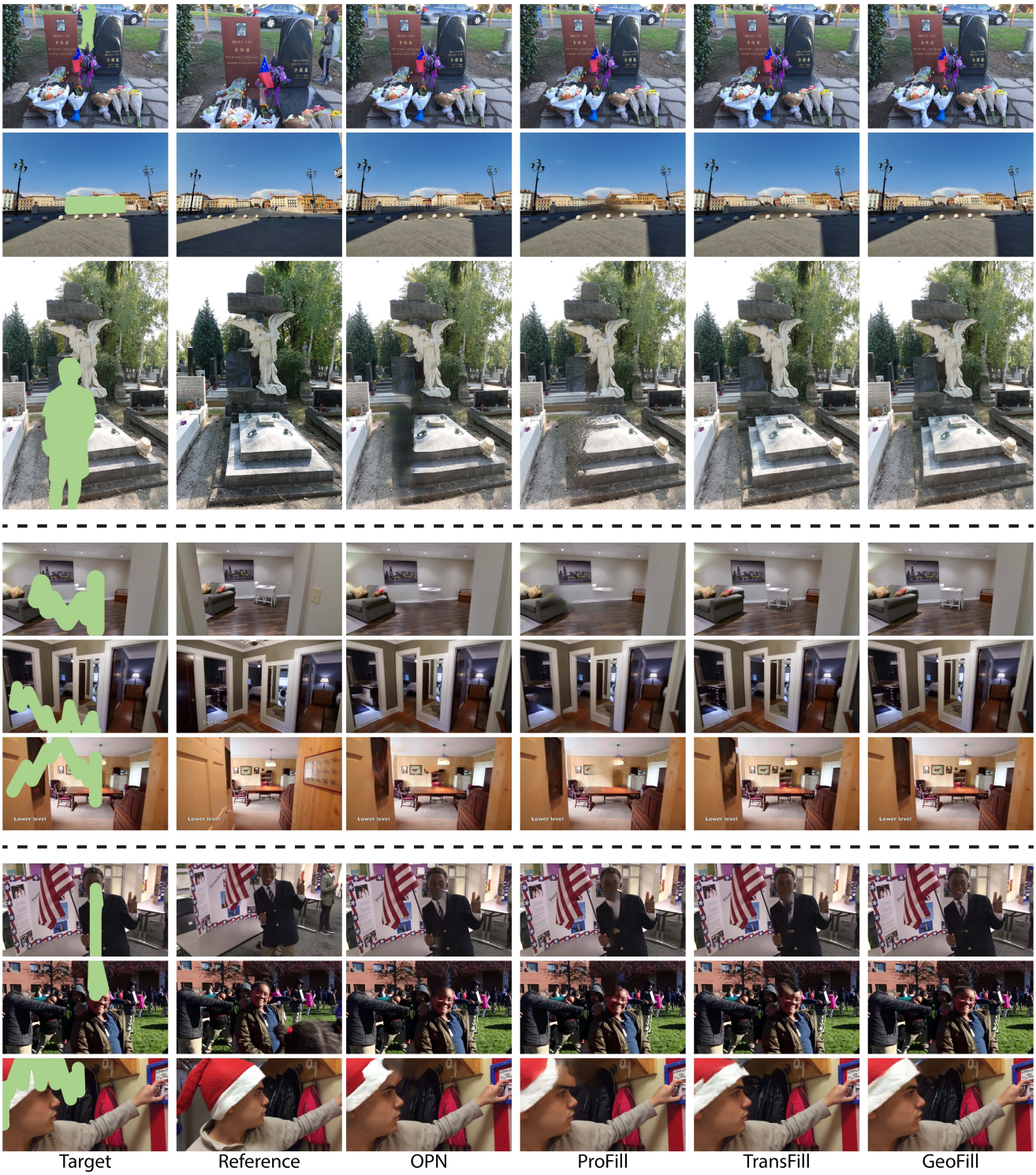

Figure 6: Qualitatively comparison of GeoFill against other baselines on user-provided images (top 3 rows), RealEstate10K (mid 3 rows), and MannequinChallenge dataset (last 3 rows).

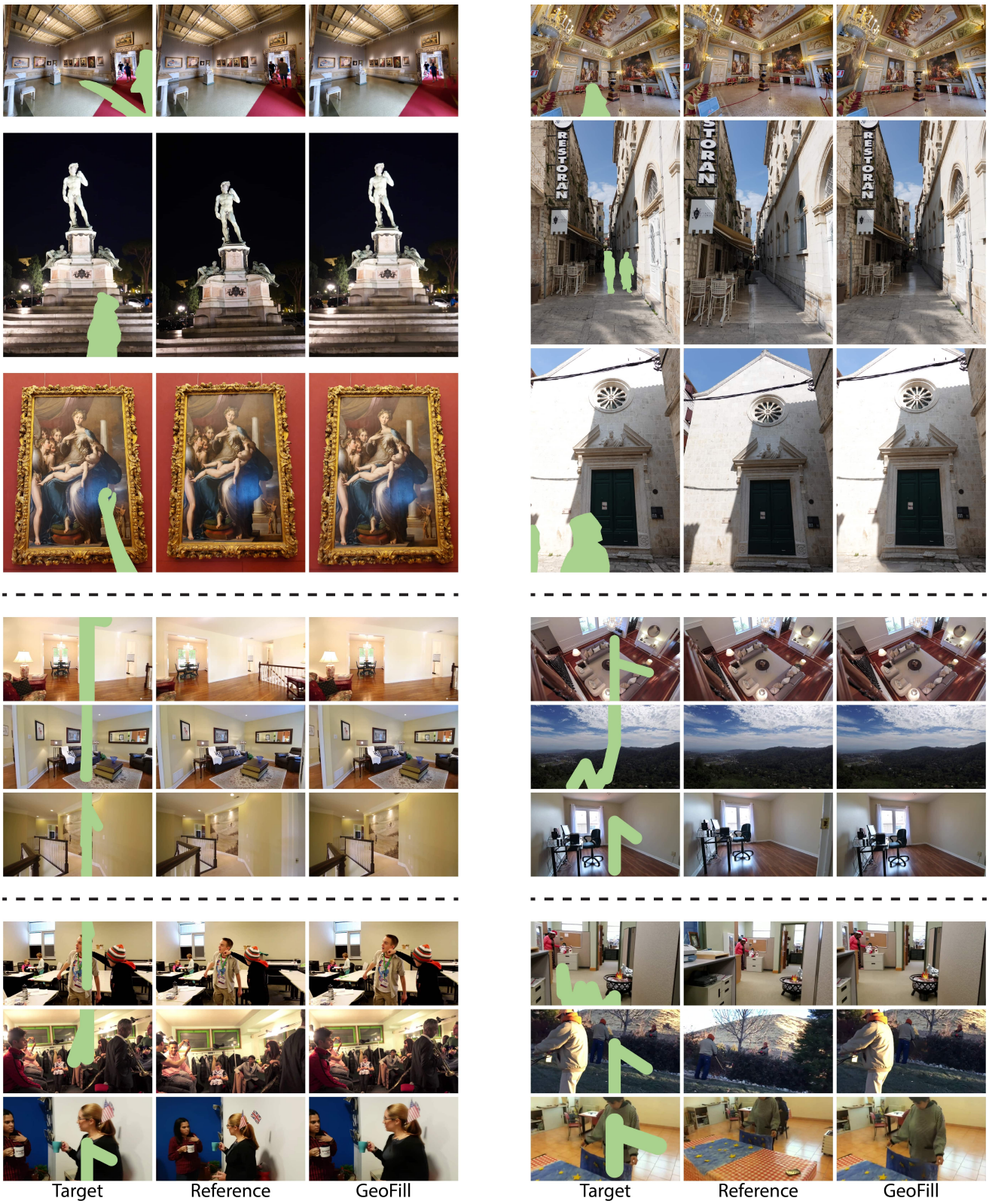

Figure 7: Visual illustration of inpaiting performance of GeoFill on user-provided images, RealEstate10K, and MannequinChallenge dataset.