GHPT: Real-Time Relightable Gaussian Splatting using Hybrid Path Tracing

Supplementary Material

7. Implementation Details

7.1. Hybrid Path Tracing

The pseudocode of our proposed hybrid path tracing is given in Algorithm 1, where NEE and MIS are omitted for clarity. For each pixel, the normal, the position, and the material properties are retrieved from the G-buffer rendered by PGSR. To avoid the self-intersection issues introduced by the differences between the G-Buffer rendered by PGSR and the underlying mesh, we discard the intersections lying on the back face or with a hit distance less than a small threshold ϵ. Specifically, we use `CandidateTriangleRayT` function to compute the hit distance, `CandidateTriangleFrontFace` function to check whether the candidate hit is on the back face, and `CommitNonOpaqueTriangleHit` function to confirm the hit in the Slang shading language.

Algorithm 1: Hybrid Path Tracing

Input: pixel coordinate p, G-Buffer normal n and
G-buffer position P rendered by PGSR,
mesh acceleration structure $\mathcal{M}$, environment
map $\mathcal{E}$, clamp threshold ϵ
Output: outgoing radiance $L(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{o})$

1   $T \leftarrow 1, \quad L(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{o}) \leftarrow 0$
2   **for** $m \leftarrow 1$ **to** $maxBounces$ **do**
3   Sample BRDF in direction $\boldsymbol{\omega}_\mathrm{i}$, obtain
     $f_\mathrm{r}(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$ and pdf p
4   trace ray $(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{i})$ in $\mathcal{M}$
5   **if** `CandidateTriangleRayT()` $> \epsilon$ **or**
     `CandidateTriangleFrontFace()` **then**
6   `CommitNonOpaqueTriangleHit()`
7   **if** *ray hits the scene* **then**
8       $T \leftarrow T f_\mathrm{r}(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})(\boldsymbol{n}(p) \cdot \boldsymbol{\omega}_\mathrm{i})/p$
9   **else**
10      $L(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{o}) \leftarrow$
       $T f_\mathrm{r}(\boldsymbol{P}(p), \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o}) L_\mathrm{env}(\boldsymbol{\omega}_\mathrm{i}) \cdot (\boldsymbol{n}(p) \cdot \boldsymbol{\omega}_\mathrm{i})/p$

7.2. Physically-Based Differentiable Rendering

We implement the physically-based differentiable renderer using a two-pass path tracing framework to compute gradients through the rendering equation for each Monte Carlo sample. In the non-differentiable first pass (`trace` function in Algorithm 2), we trace primary rays to determine surface intersections and record the visibility and other necessary information for the computation of direct illumination in the

following pass, including the hit point x, the normal n, the texture coordinate t, the visibility flags v and v_NEE, and the indirect radiance L_ind.

In the differentiable second pass (`shade` function in Algorithm 2), we evaluate the direct illumination using MIS between BRDF sampling and environment map sampling, combining both components into the final outgoing radiance. If the NEE sample of the primary hit is visible to the environment map ($v_\mathrm{NEE} = \mathtt{true}$), or the BRDF sample of the primary hit is visible to the environment map ($v = \mathtt{true}$), we evaluate the BRDF of the primary hit and compute the contribution from the environment map. By using the same random seed, we can ensure that `trace` and `shade` functions have the same samples without explicitly storing the sampling directions.

Finally, we compute the per-pixel L1 loss between the rendered image C and the target image $T(p)$ (`perPixelLoss` function in Algorithm 2), enabling gradient backpropagation to jointly optimize the albedo A, the roughness R, and the environment map L_env by calling `perPixelLoss.bwds` in SlangPy.

8. More Qualitative Comparisons

Relighting   We show additional qualitative comparisons across scenes on SYNTHETIC4RELIGHT and TENSOIR SYNTHETIC datasets in Figs. 7 to 12. R3DG [5] suffers from unrealistic transitions between highlights and shadows. IRGS [13] optimizes lower roughness, leading to an overly smooth appearance. R3DG [5] and TensoIR [17] exhibit excessively high roughness, resulting in overly dim specular reflections. In contrast, our method demonstrates proper roughness, which are closer to the ground truth.

Albedo   We give the qualitative comparisons of estimated albedo maps across scenes from SYNTHETIC4RELIGHT and TENSOIR SYNTHETIC datasets in Figs. 13 and 14. Our method exhibits better material and lighting decomposition results with the ground truth and produces clean albedo maps.

Environment Map   We also conduct qualitative comparisons of the estimated environment maps across scenes from SYNTHETIC4RELIGHT and TENSOIR SYNTHETIC datasets in Figs. 15 and 16. IRGS [13] generates blurred and noisy environment maps, while our method can reconstruct high-fidelity environment maps that preserve finer details and less noise, which is closest to the ground truth in terms of illumination distribution and direction of the sunlight.

Algorithm 2: Differentiable Path Tracing

Input: target image T, pixel coordinate p, mesh acceleration structure $\mathcal{M}$, albedo texture A, roughness texture R, environment map L_{env}

Output: L1 loss L_1 between rendered color and target color

1 **function** perPixelLoss(T, c, $\mathcal{T}$, A, R, L_{env})
2     $C \leftarrow \mathbf{0}$
3   **for** $i = 1$ **to** N_{samples} **do**
4         $(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{t}, v, v_{\text{NEE}}, \boldsymbol{L}_{\text{ind}}) \leftarrow$ trace($i, \mathcal{M}, \boldsymbol{p}, A, R, \boldsymbol{L}_{\text{env}}$)
5         $C{+}{=}$ shade($i, \boldsymbol{p}, A, R, \boldsymbol{L}_{\text{env}}, \boldsymbol{x}, \boldsymbol{n}, \boldsymbol{t}, v, v_{\text{NEE}}, \boldsymbol{L}_{\text{ind}}$)
6     $C \leftarrow C/N_{\text{samples}}$
7     $L_1 \leftarrow \|C - \boldsymbol{T}(\boldsymbol{p})\|_1$
8   **return** L_1

9 **function** trace(i, $\mathcal{T}$, c, A, R, L_{env})
10     $v \leftarrow$ false
11     $v_{\text{NEE}} \leftarrow$ false
12   Trace primary ray
13   **if** *ray hits $\mathcal{T}$* **then**
14   Read position $\boldsymbol{x}$, normal $\boldsymbol{n}$, texture coordinate $\boldsymbol{t}$, material (a, r) of the primary hit
15   Initialize indirect radiance $\boldsymbol{L}_{\text{ind}} \leftarrow \mathbf{0}$
16   **for** $j = 1$ **to** $maxBounces$ **do**
17   Sample environment map in $\boldsymbol{\omega}$
18   Trace ray $(\boldsymbol{x}, \boldsymbol{\omega})$
19   **if** ***not** ray $(\boldsymbol{x}, \boldsymbol{\omega}_{\text{i}})$ hits $\mathcal{T}$* **then**
20   Evaluate BRDF and accumulate the contribution of environment map to $\boldsymbol{L}_{\text{ind}}$
21   **if** $j = 1$ **then**
22                     $v_{\text{NEE}} \leftarrow$ true
23   Sample BRDF in direction $\boldsymbol{\omega}_{\text{i}}$
24   Trace ray $(\boldsymbol{x}, \boldsymbol{\omega}_{\text{i}})$
25   **if** *ray $(\boldsymbol{x}, \boldsymbol{\omega}_{\text{i}})$ hits $\mathcal{T}$* **then**
26   Update throughput and hit info
27   **else**
28   Evaluate BRDF and accumulate the contribution of environment map to $\boldsymbol{L}_{\text{ind}}$
29   **if** $j = 1$ **then**
30                     $v \leftarrow$ true
31   **break**
32   **return** $(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{t}, v, v_{\text{NEE}}, \boldsymbol{L}_{\text{ind}})$

33 **function** shade(i, p, A, R, L_{env}, $\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{t}, v, v_{\text{NEE}}, \boldsymbol{L}_{\text{ind}}$)
34   Fetch material: $a \leftarrow A(\boldsymbol{t})$, $r \leftarrow R(\boldsymbol{t})$
35     $C \leftarrow \boldsymbol{L}_{\text{ind}}$
36   **if** v_{NEE} **then**
37   Evaluate BRDF and accumulate the contribution of environment map to C
38   **if** v **then**
39   Evaluate BRDF and accumulate the contribution of environment map to C
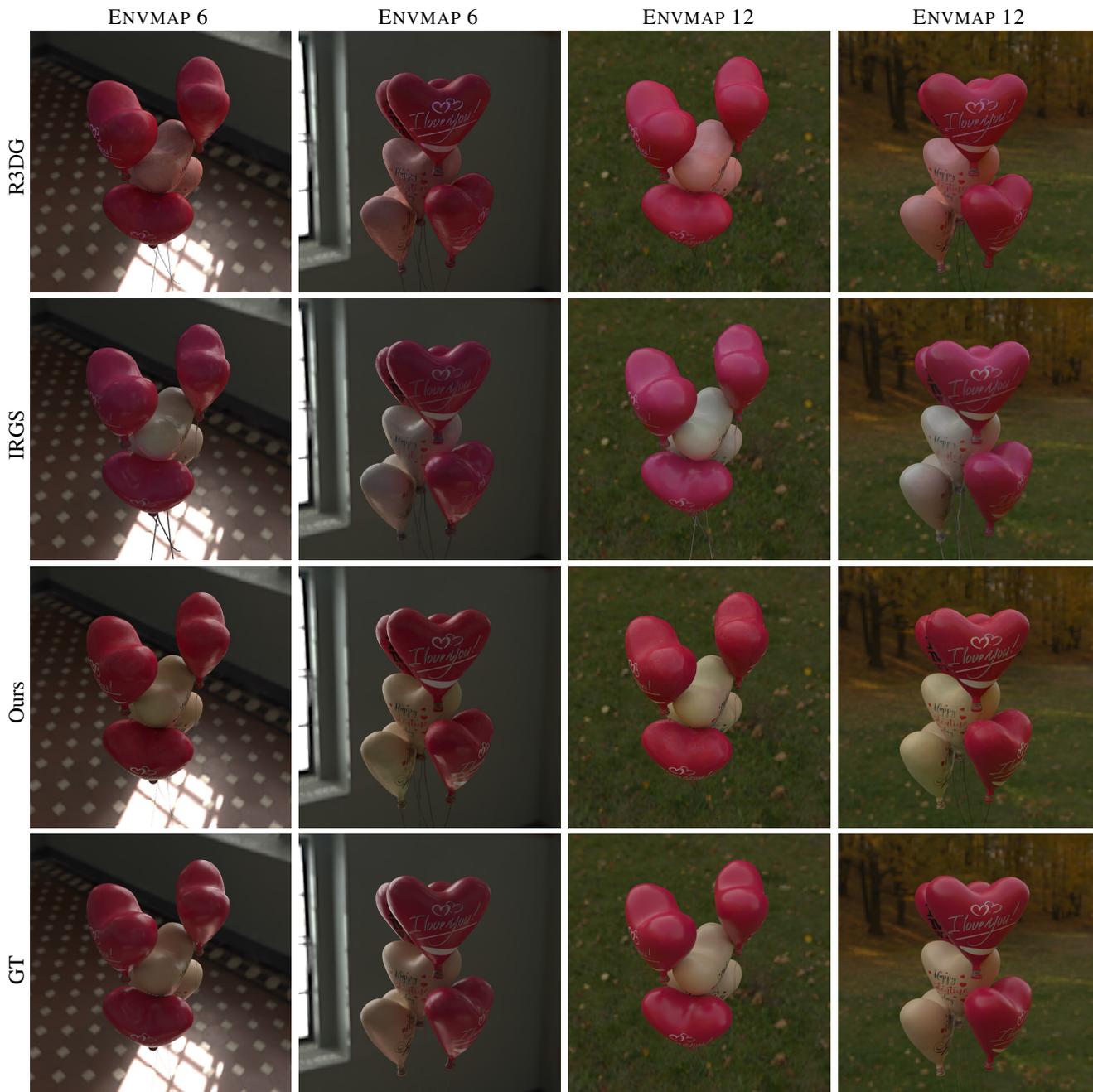40   **return** C

Figure 7. Qualitative comparison of relighting results under different environment maps on AIR BALLOONS from SYNTHETIC4RELIGHT dataset.
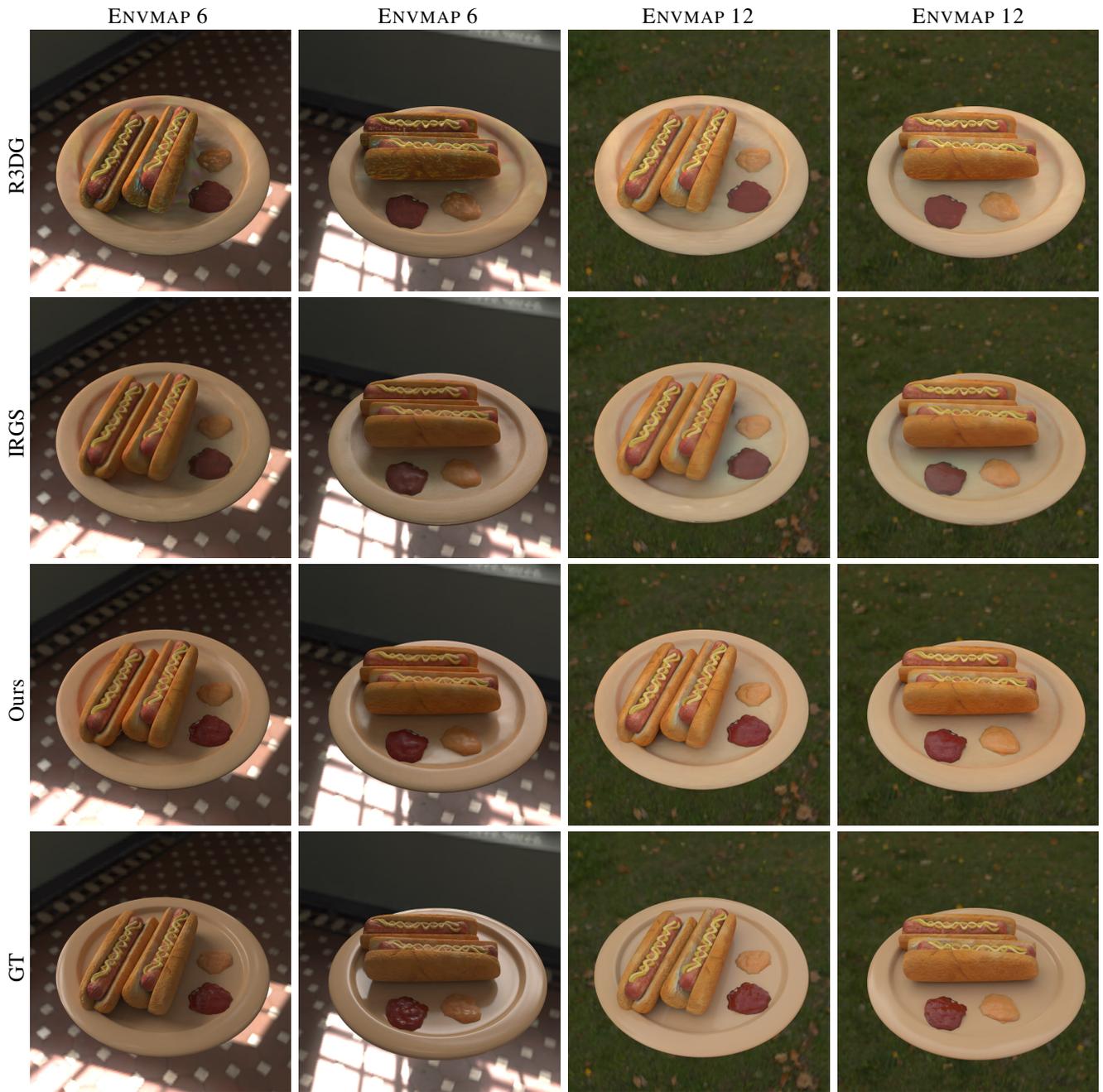
Figure 8. Qualitative comparison of relighting results under different environment maps on HOTDOG from SYNTHETIC4RELIGHT dataset.
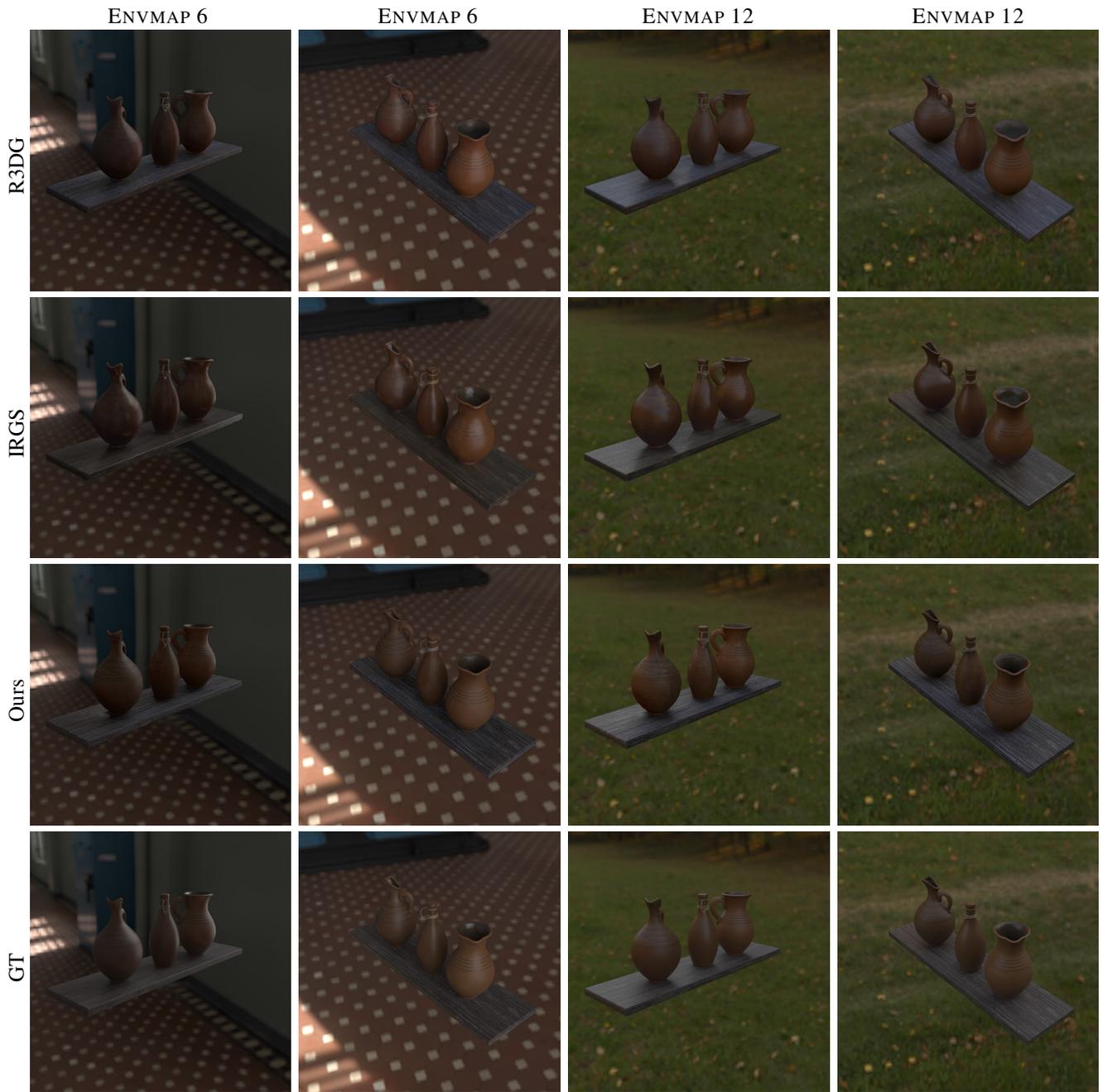
Figure 9. Qualitative comparison of relighting results under different environment maps on JUGS from SYNTHETIC4RELIGHT dataset.
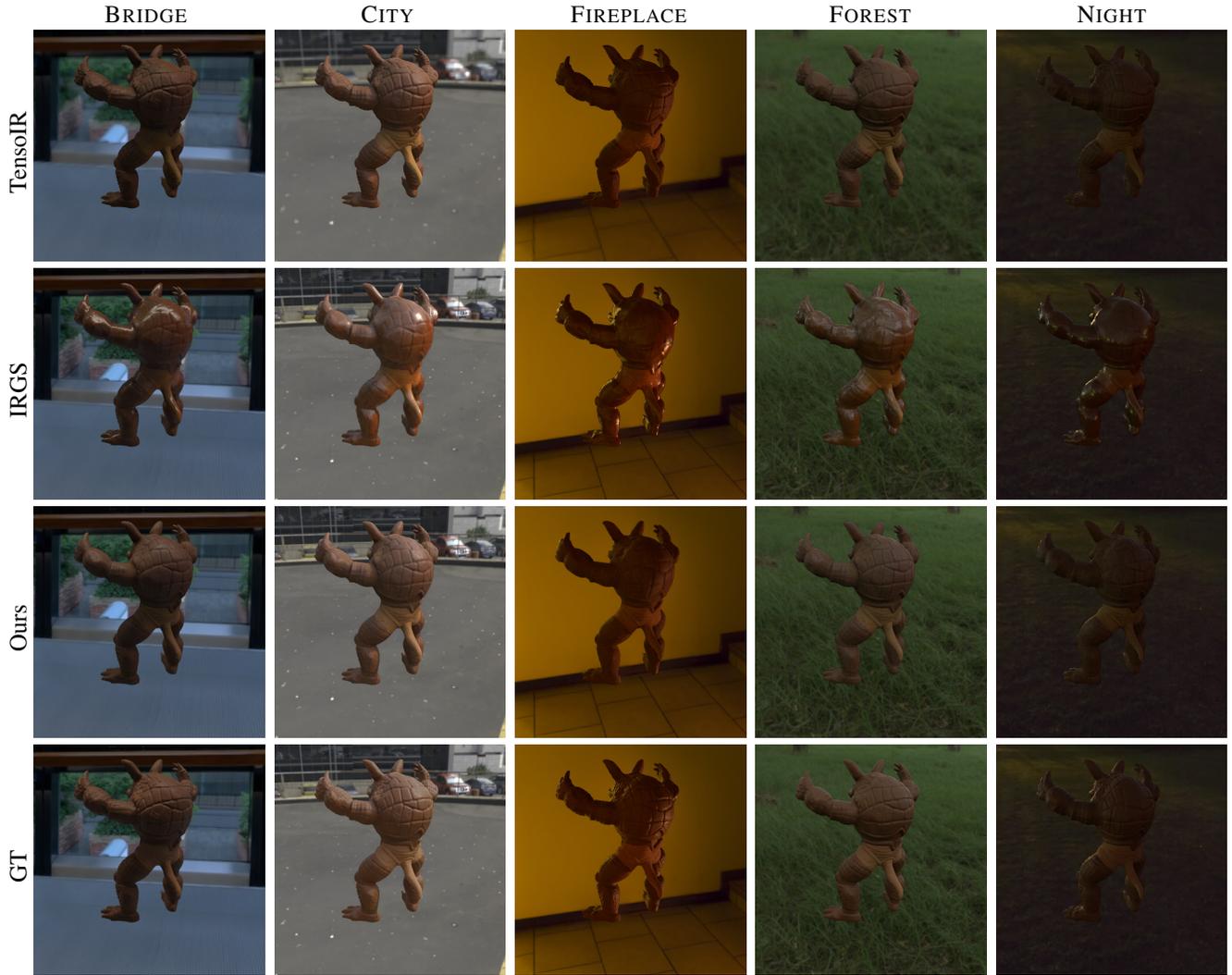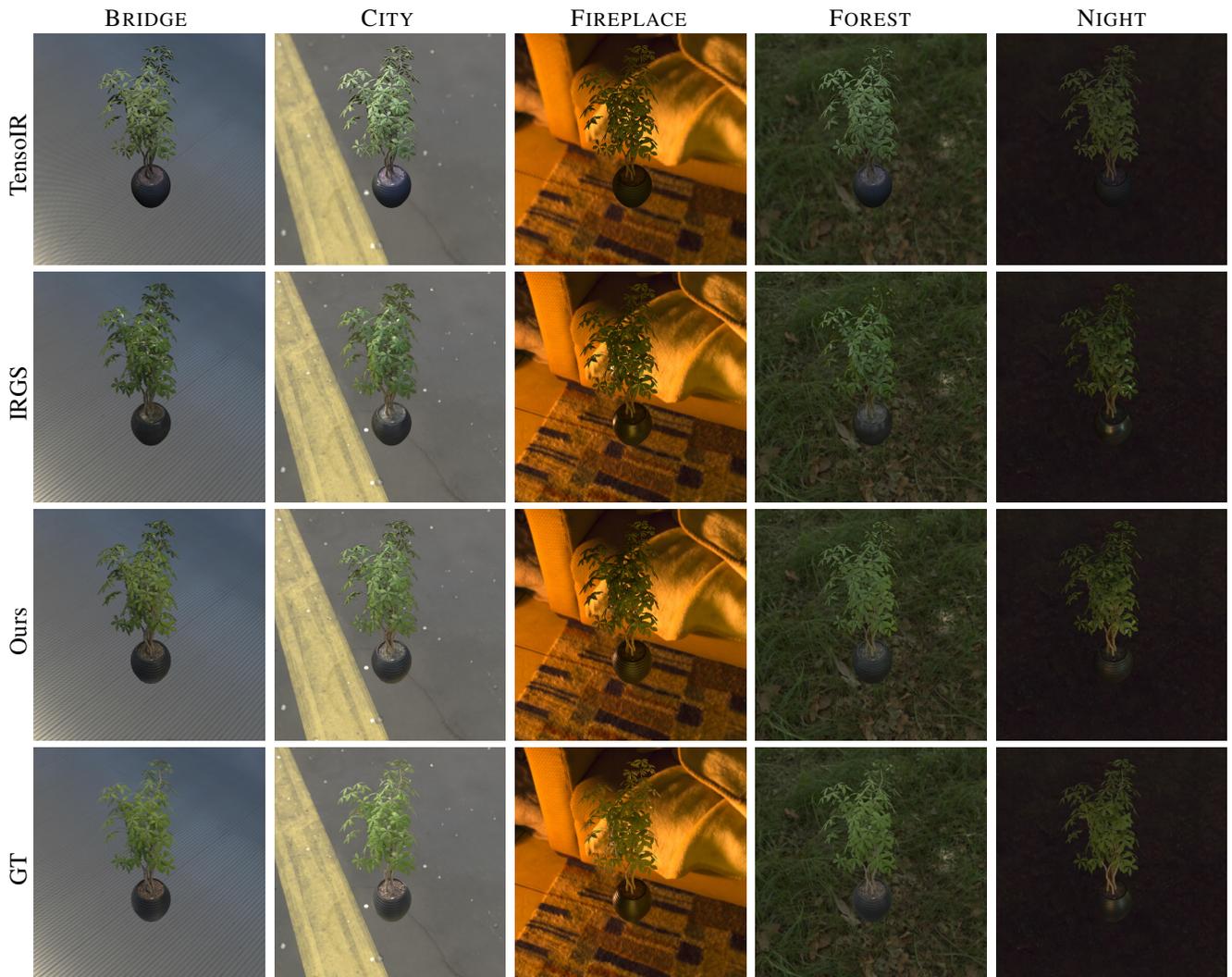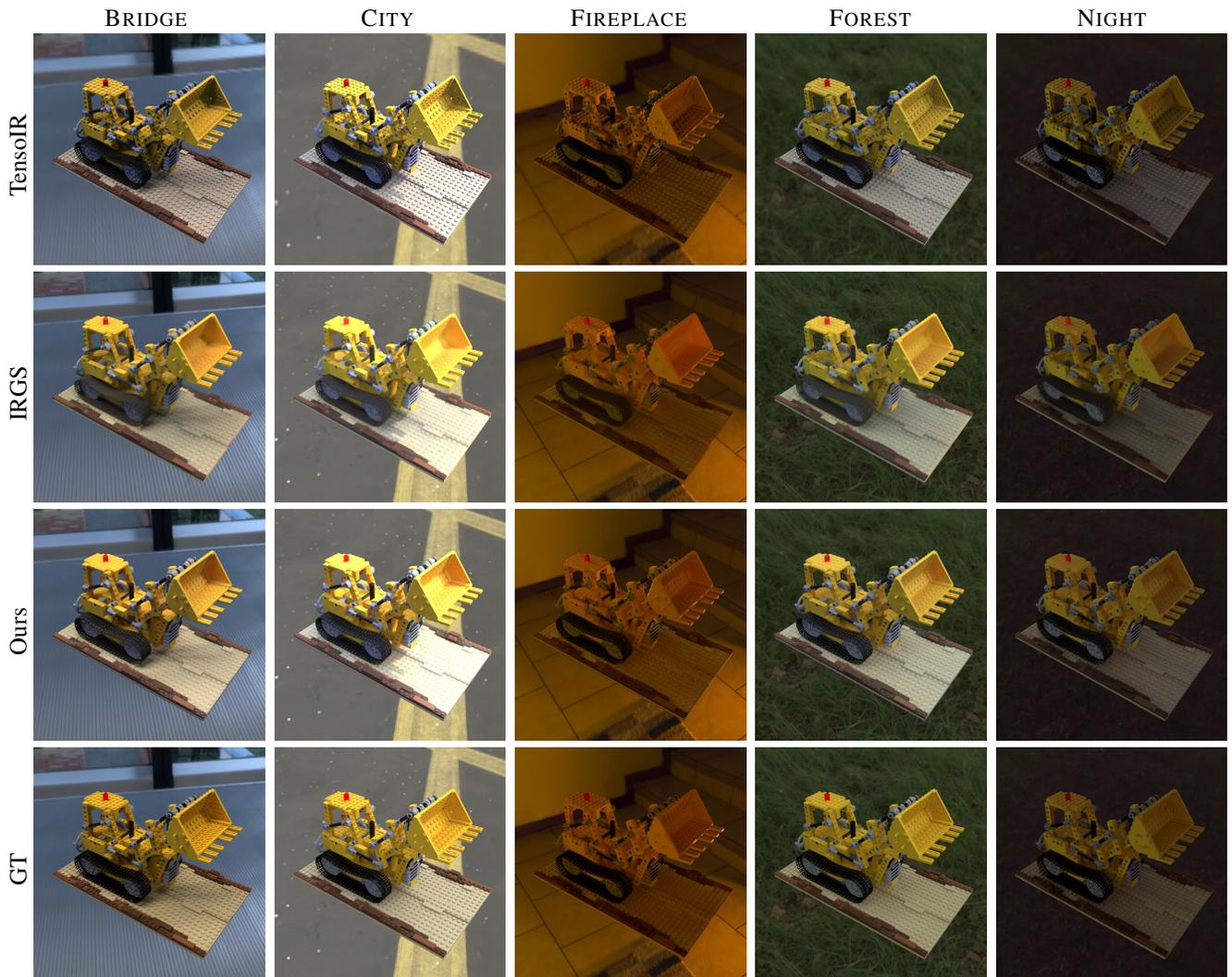
Figure 10. Qualitative comparison of relighting results under different environment maps on ARMADILLO from TENSOIR SYNTHETIC dataset.
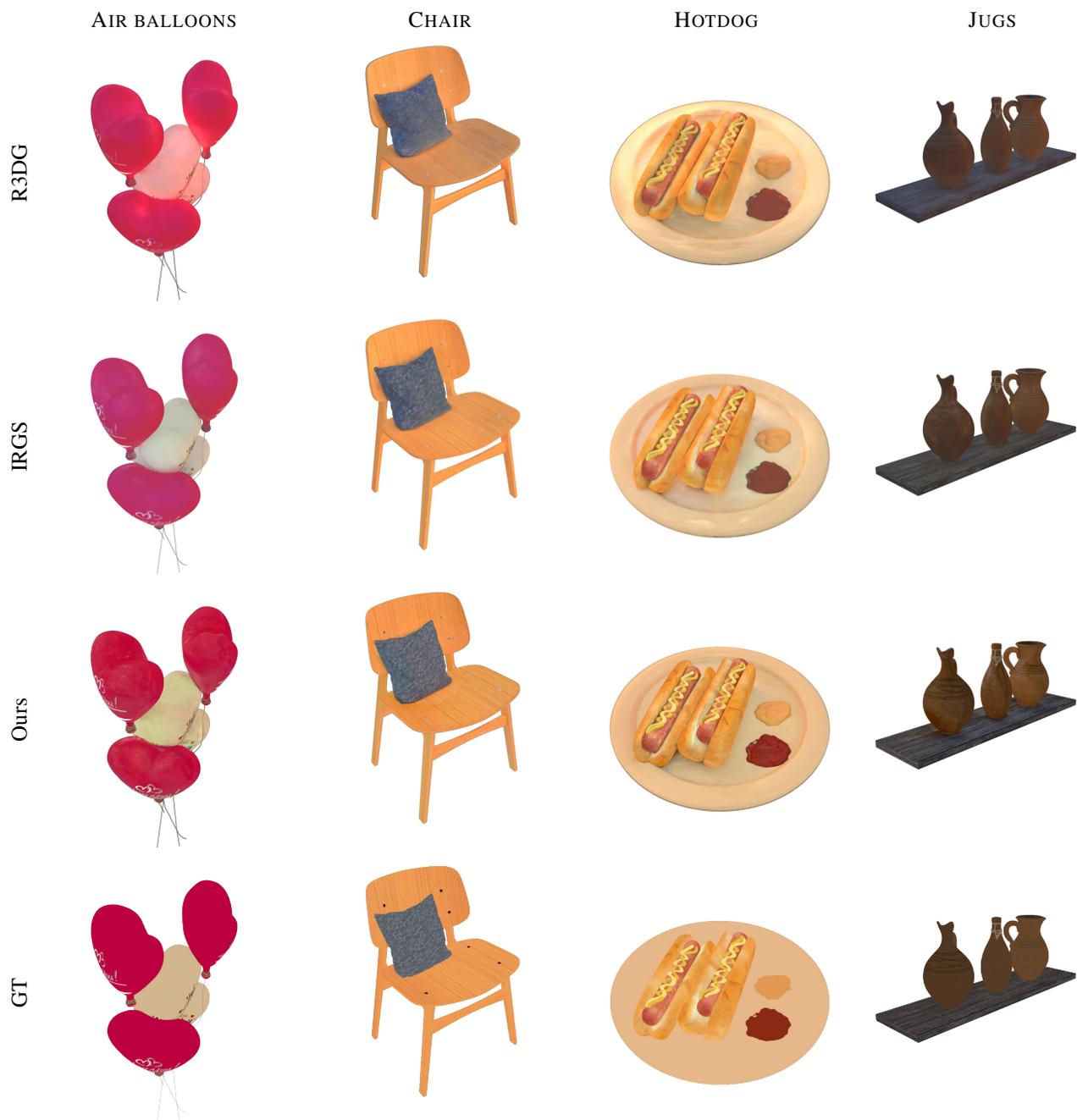
Figure 11. Qualitative comparison of relighting results under different environment maps on FICUS from TENSOIR SYNTHETIC dataset.

Figure 12. Qualitative comparison of relighting results under different environment maps on LEGO from TENSOIR SYNTHETIC dataset.

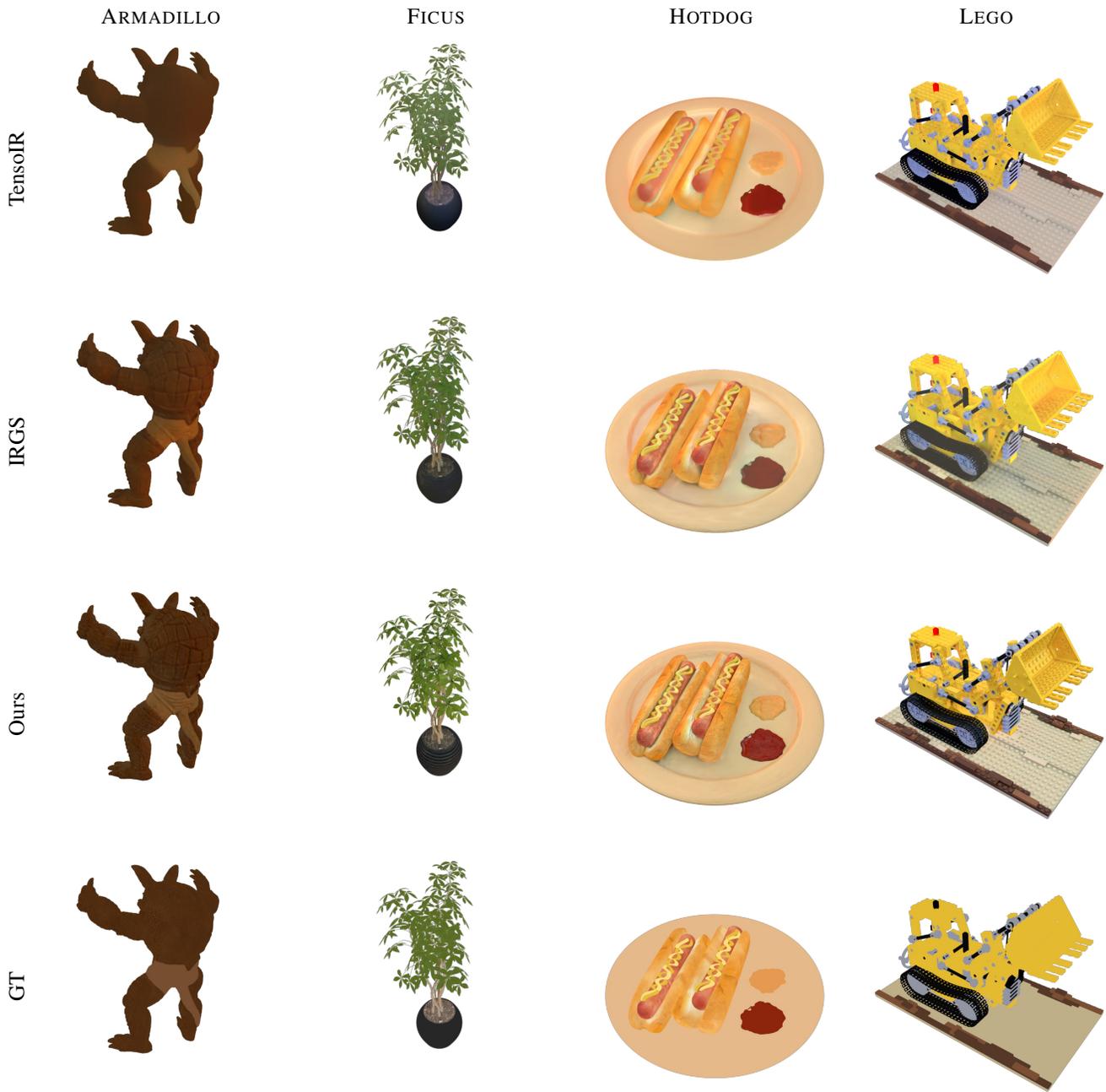Figure 13. Qualitative comparison of the estimated albedo on SYNTHETIC4RELIGHT dataset.

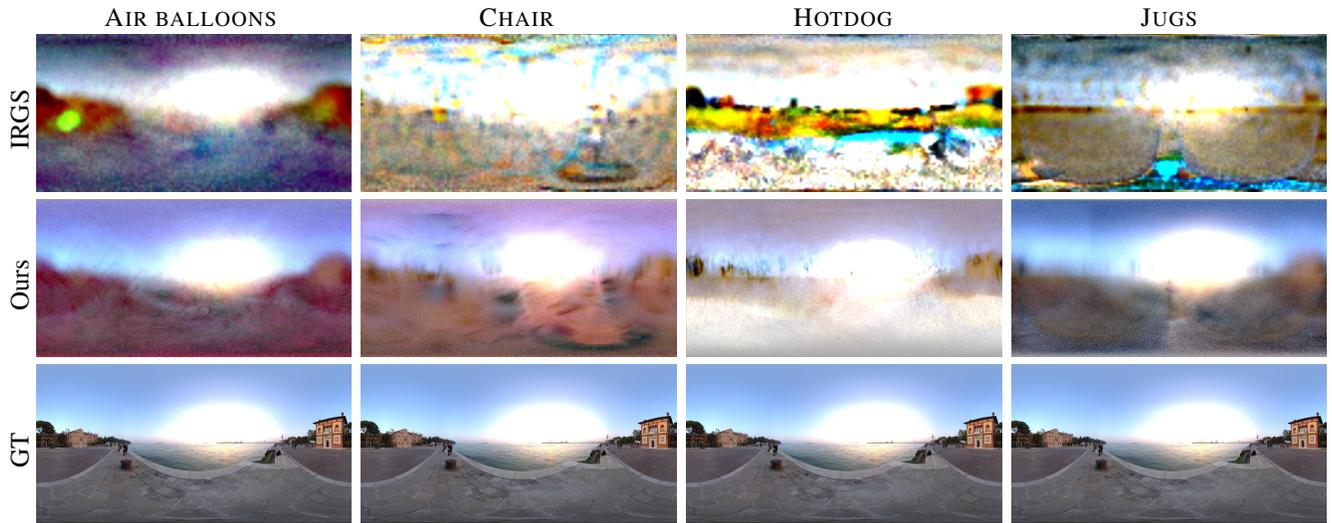Figure 14. Qualitative comparison of the estimated albedo on TENSOIR SYNTHETIC dataset.

Figure 15. Qualitative comparison of the estimated environment maps on SYNTHETIC4RELIGHT dataset.
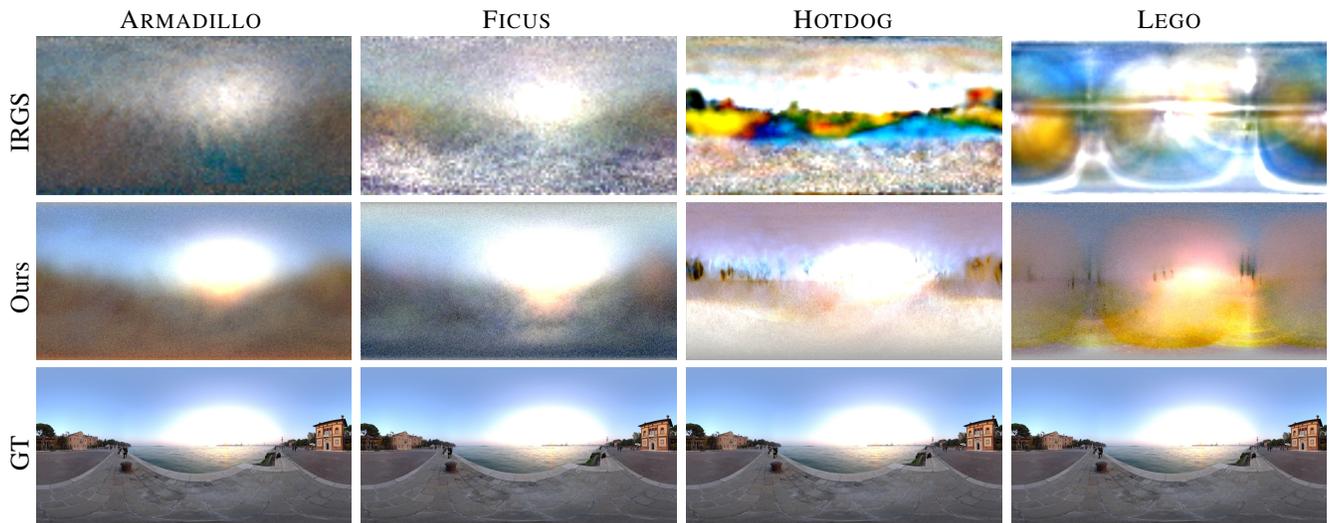

Figure 16. Qualitative comparison of the estimated environment maps on TENSOIR SYNTHETIC dataset.