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

Supplementary Material

7. Implementation Details

7.1. Hybrid Path Tracing

Hybrid path tracing refers to deferred rendering where G-buffers, including depth, normal, albedo, and roughness, are rasterized by Gaussian splatting, while shadows and indirect illumination are evaluated by path tracing on the underlying mesh. Specifically, secondary rays are shot from G-buffers according to the BRDF, intersected with the mesh to evaluate visibility, and further reflected from the hits to account for multi-bounce global illumination. Additionally, shadow rays are cast at each path vertex to directly sample the environment map.

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 $\boldsymbol{p}$, G-Buffer normal $\boldsymbol{n}$ and G-buffer position $\boldsymbol{P}$ rendered by PGSR, mesh acceleration structure $\mathcal{M}$, environment map $\mathcal{E}$, clamp threshold ϵ

Output: outgoing radiance $L(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{o})$

$T \leftarrow 1, \quad L(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{o}) \leftarrow 0$

for $m \leftarrow 1$ **to** $maxBounces$ **do**

- Sample BRDF in direction $\boldsymbol{\omega}_\mathrm{i}$, obtain $f_\mathrm{r}(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$ and pdf p
- trace ray $(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{i})$ in $\mathcal{M}$
- **if** *`CandidateTriangleRayT()`* $> \epsilon$ ***or*** *`CandidateTriangleFrontFace()`* **then**
  - `CommitNonOpaqueTriangleHit()`
- **if** *ray hits the scene* **then**
  - $T \leftarrow T f_\mathrm{r}(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})(\boldsymbol{n}(\boldsymbol{p}) \cdot \boldsymbol{\omega}_\mathrm{i})/p$
- **else**
  - $L(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{o}) \leftarrow T f_\mathrm{r}(\boldsymbol{P}(\boldsymbol{p}), \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o}) L_\mathrm{env}(\boldsymbol{\omega}_\mathrm{i}) \cdot (\boldsymbol{n}(\boldsymbol{p}) \cdot \boldsymbol{\omega}_\mathrm{i})/p$

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 $\boldsymbol{x}$, the normal $\boldsymbol{n}$, the texture coordinate $\boldsymbol{t}$, the visibility flags v and v_NEE, and the indirect radiance $\boldsymbol{L}_\mathrm{ind}$.

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 ($v_\mathrm{NEE} = \texttt{true}$), or the BRDF sample of the primary hit is visible to the environment map ($v = \texttt{true}$), 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 $\boldsymbol{C}$ and the target image $T(\boldsymbol{p})$ (`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. Rendering Performance Comparison

We report the rendering time of the first view under the first environment map during relighting for R3DG, IRGS, and our method at the same spp (256). Our GHPT can still render in real-time at such high spp.

Table 5. Rendering time (ms) at 256 spp.

	SYNTHETIC4RELIGHT				TENSOIR SYNTHETIC			
	Balloons	Chair	Hotdog	Jugs	Armadillo	Ficus	Hotdog	Lego
R3DG	194	356	203	191	131	134	206	330
IRGS	1348	716	1772	940	457	349	1386	917
Ours	19.6	12.7	26.2	15.9	12.7	15.9	26.8	25.9

9. More Qualitative Comparisons

Relighting We show additional qualitative comparisons across scenes on SYNTHETIC4RELIGHT and TENSOIR SYNTHETIC datasets in Figs. 7 to 12. R3DG suffers from unrealistic transitions between highlights and shadows. IRGS optimizes lower roughness, leading to an overly smooth appearance. R3DG and TensoIR 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 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 $\boldsymbol{p}$, 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

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

function `trace` (i, $\mathcal{T}$, $\boldsymbol{c}$, A, R, $\boldsymbol{L}_{\text{env}}$)
- $v \leftarrow$ `false`
- $v_{\text{NEE}} \leftarrow$ `false`
- Trace primary ray
- **if** *ray hits* $\mathcal{T}$ **then**
  - Read position $\boldsymbol{x}$, normal $\boldsymbol{n}$, texture coordinate $\boldsymbol{t}$, material (a, r) of the primary hit
  - Initialize indirect radiance $\boldsymbol{L}_{\text{ind}} \leftarrow \boldsymbol{0}$
  - **for** $j = 1$ **to** $maxBounces$ **do**
    - Sample environment map in $\boldsymbol{\omega}$
    - Trace ray $(\boldsymbol{x}, \boldsymbol{\omega})$
    - **if** ***not*** *ray* $(\boldsymbol{x}, \boldsymbol{\omega}_{\text{i}})$ *hits* $\mathcal{T}$ **then**
      - Evaluate BRDF and accumulate the contribution of environment map to $\boldsymbol{L}_{\text{ind}}$
      - **if** $j = 1$ **then**
        - $v_{\text{NEE}} \leftarrow$ `true`
    - Sample BRDF in direction $\boldsymbol{\omega}_{\text{i}}$
    - Trace ray $(\boldsymbol{x}, \boldsymbol{\omega}_{\text{i}})$
    - **if** *ray* $(\boldsymbol{x}, \boldsymbol{\omega}_{\text{i}})$ *hits* $\mathcal{T}$ **then**
      - Update throughput and hit info
    - **else**
      - Evaluate BRDF and accumulate the contribution of environment map to $\boldsymbol{L}_{\text{ind}}$
      - **if** $j = 1$ **then**
        - $v \leftarrow$ `true`
      - **break**
- **return** $(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{t}, v, v_{\text{NEE}}, \boldsymbol{L}_{\text{ind}})$

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

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.

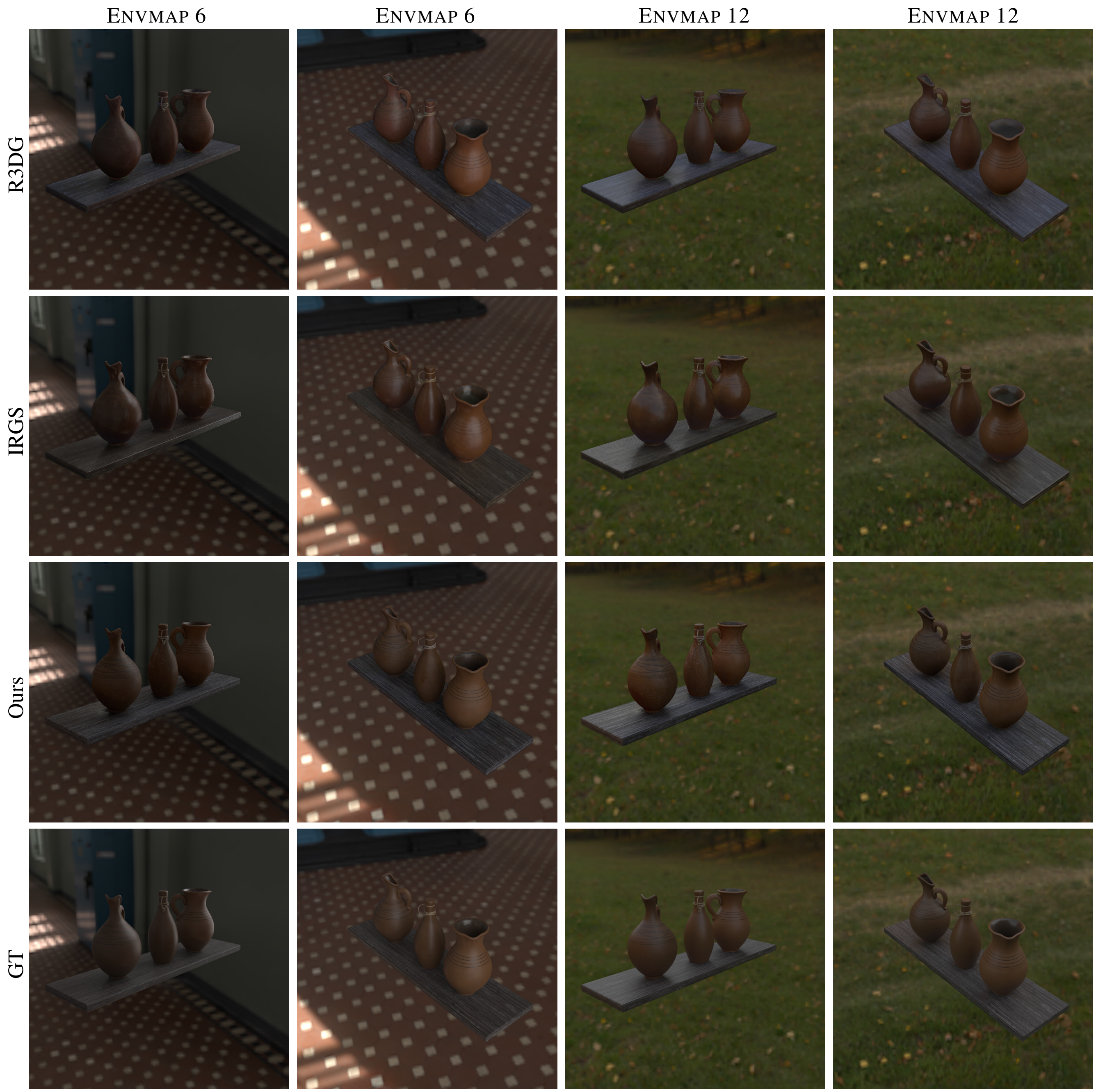

Figure 9. Qualitative comparison of relighting results under different environment maps on JUGS from SYNTHETIC4RELIGHT dataset.

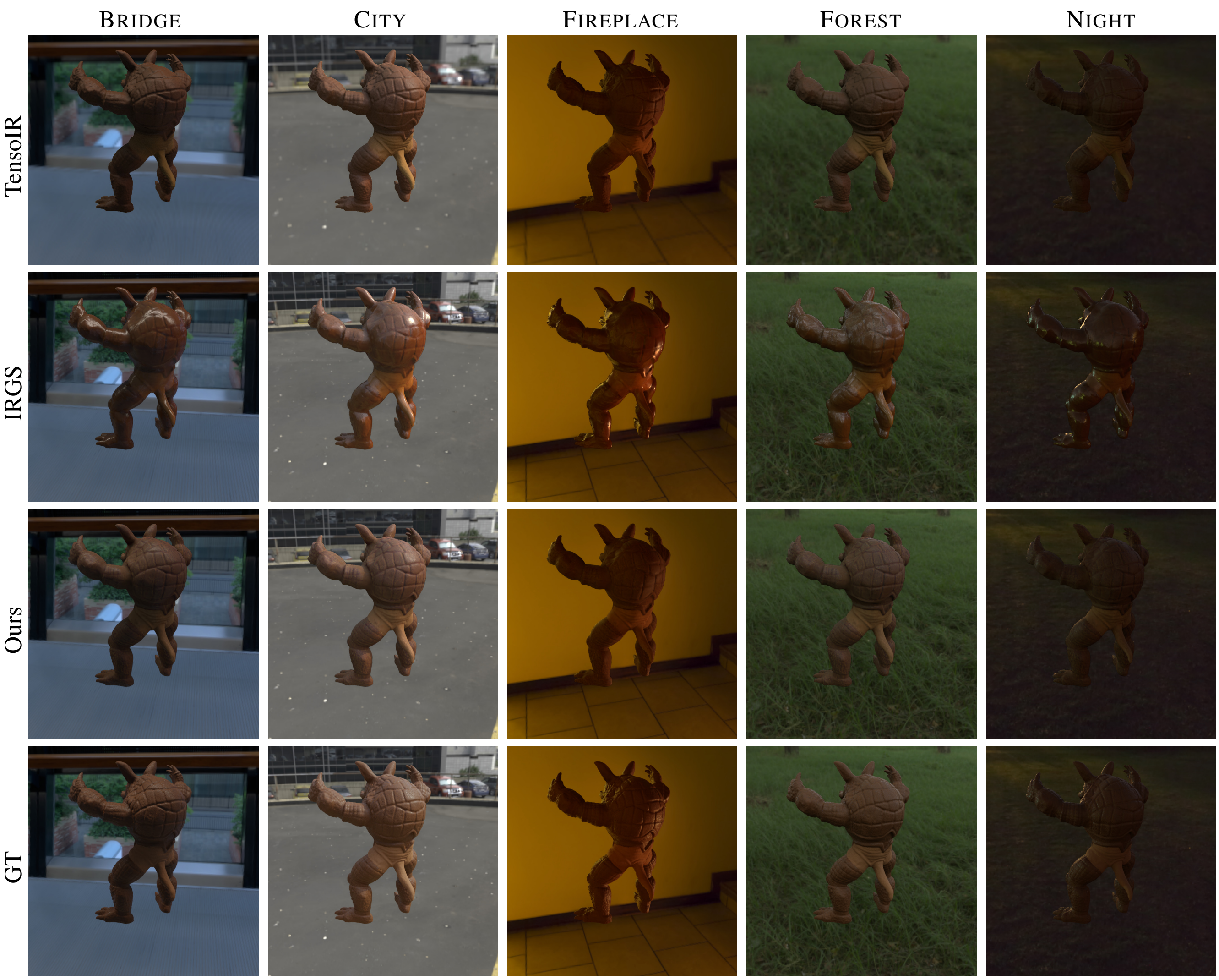

Figure 10. Qualitative comparison of relighting results under different environment maps on ARMADILLO from TENSOIR SYNTHETIC dataset.

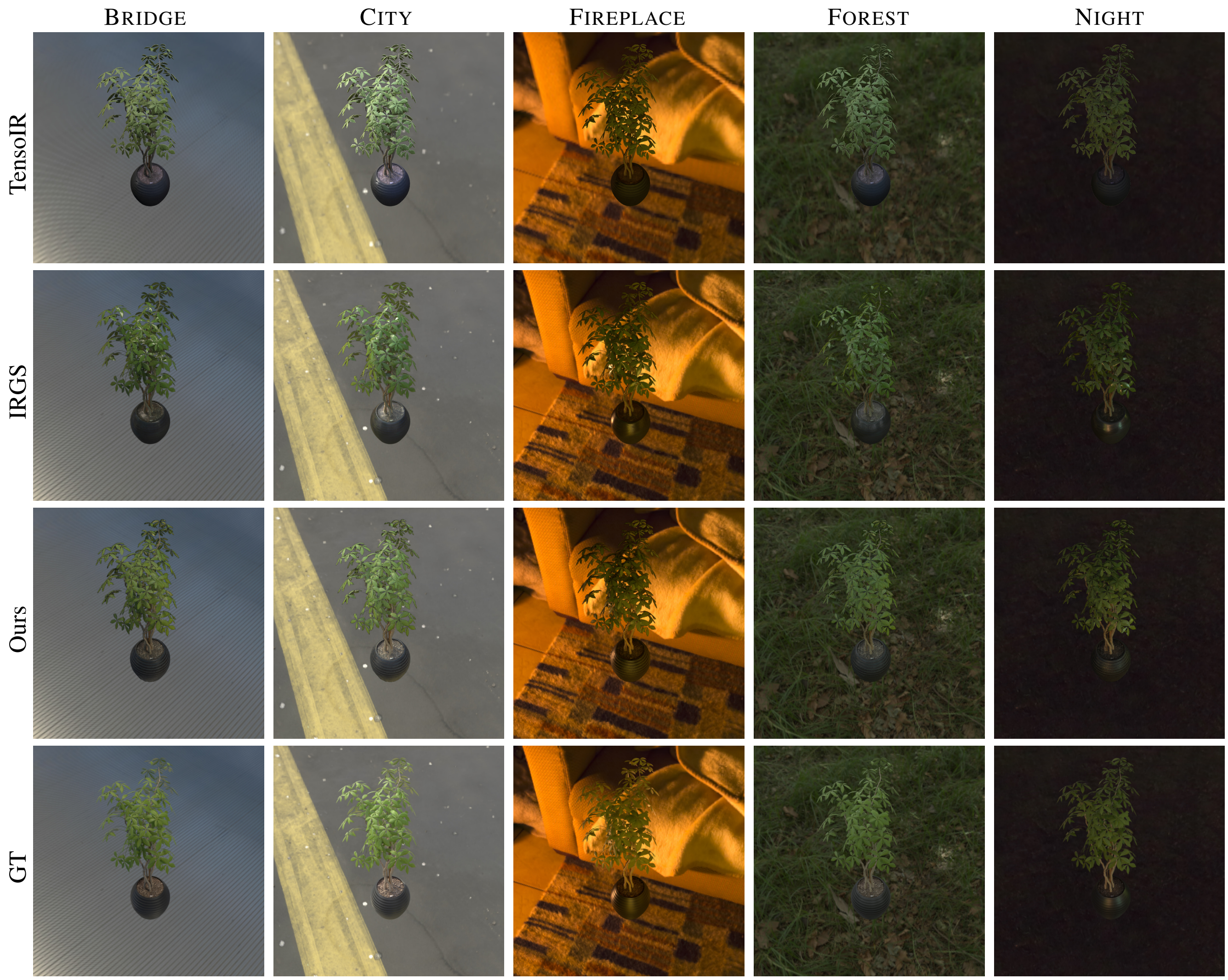

Figure 11. Qualitative comparison of relighting results under different environment maps on FICUS from TENSOIR SYNTHETIC dataset.

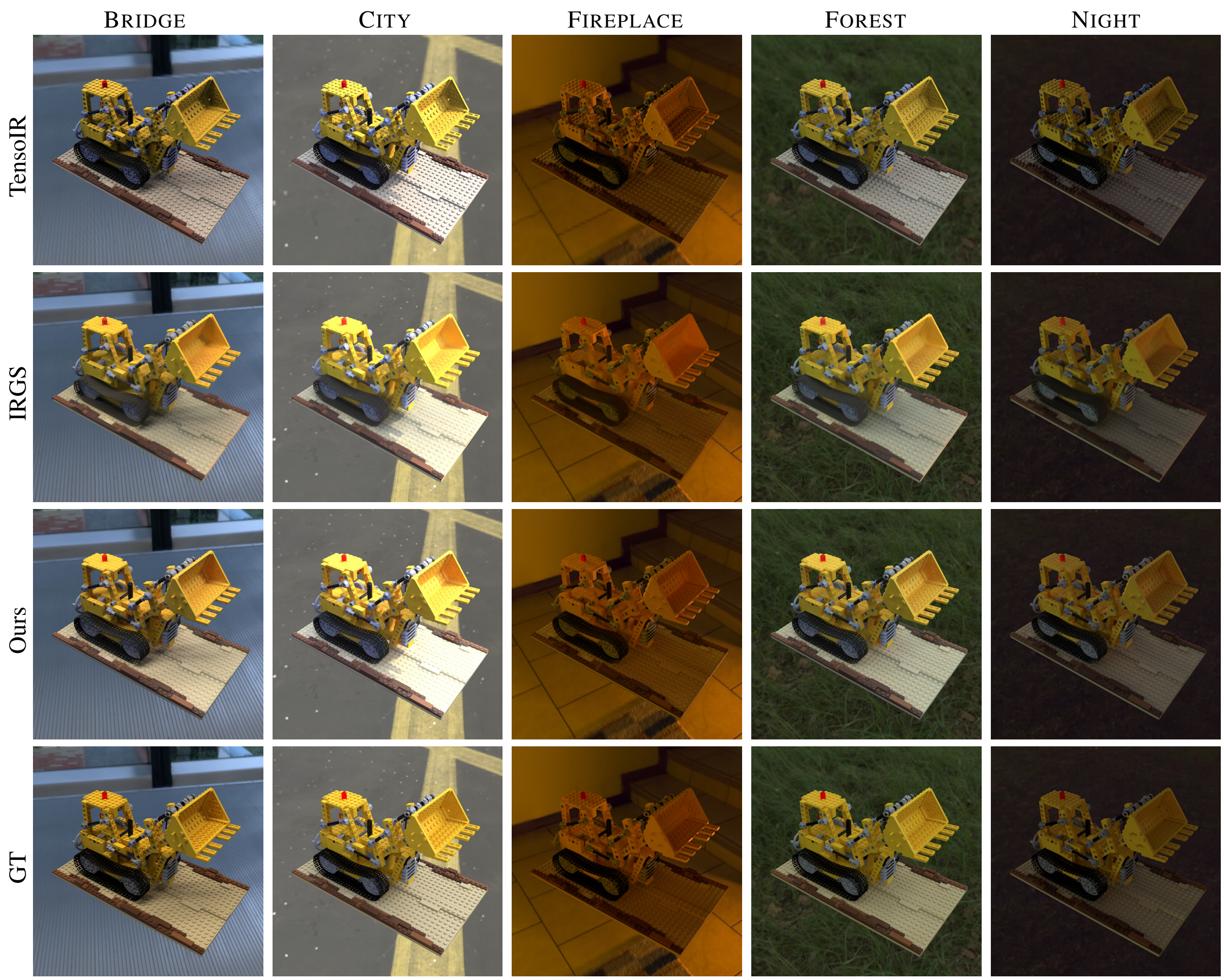

Figure 12. Qualitative comparison of relighting results under different environment maps on LEGO from TENSOIR SYNTHETIC dataset.

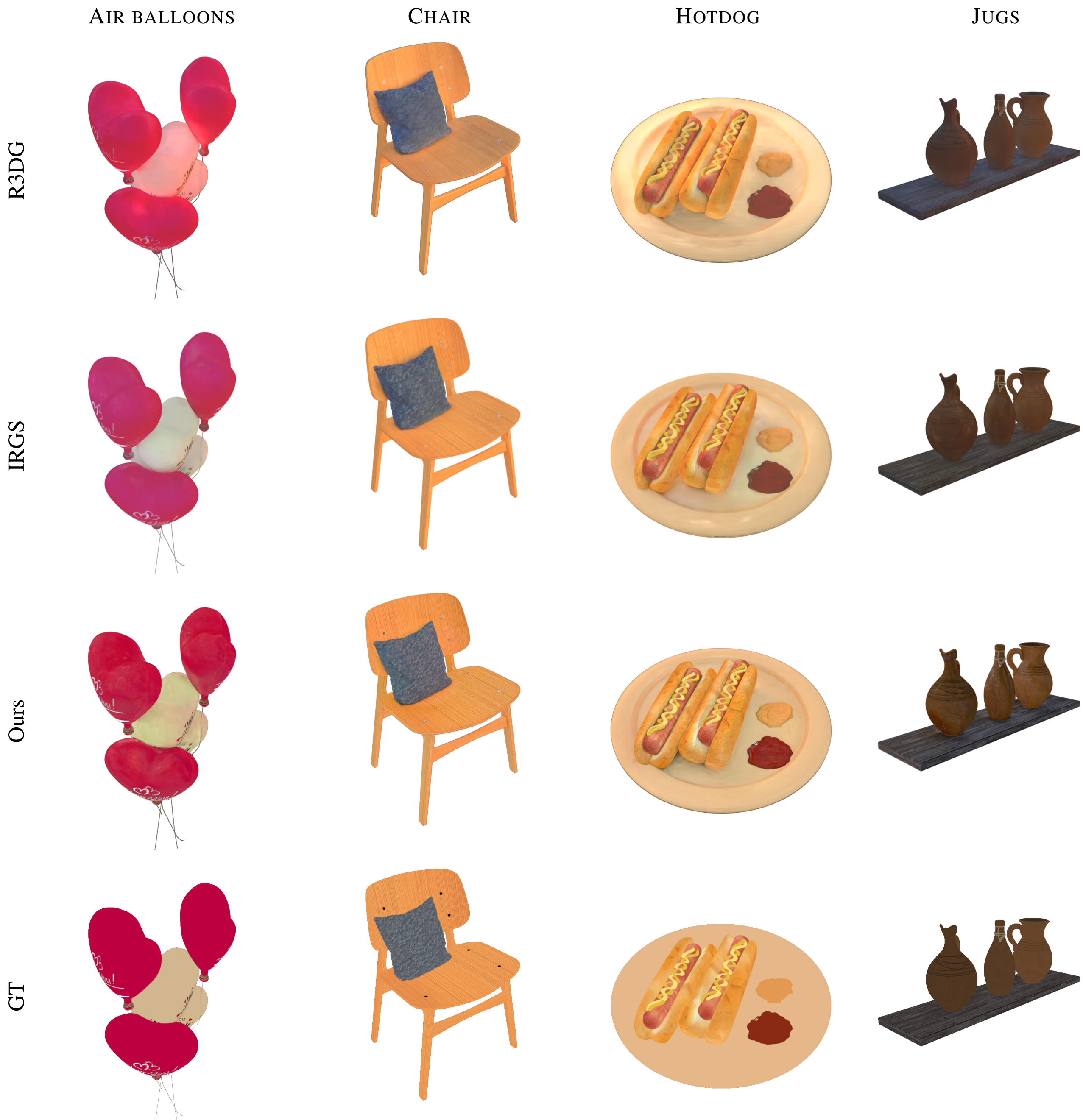

Figure 13. Qualitative comparison of the estimated albedo on SYNTHETIC4RELIGHT dataset.

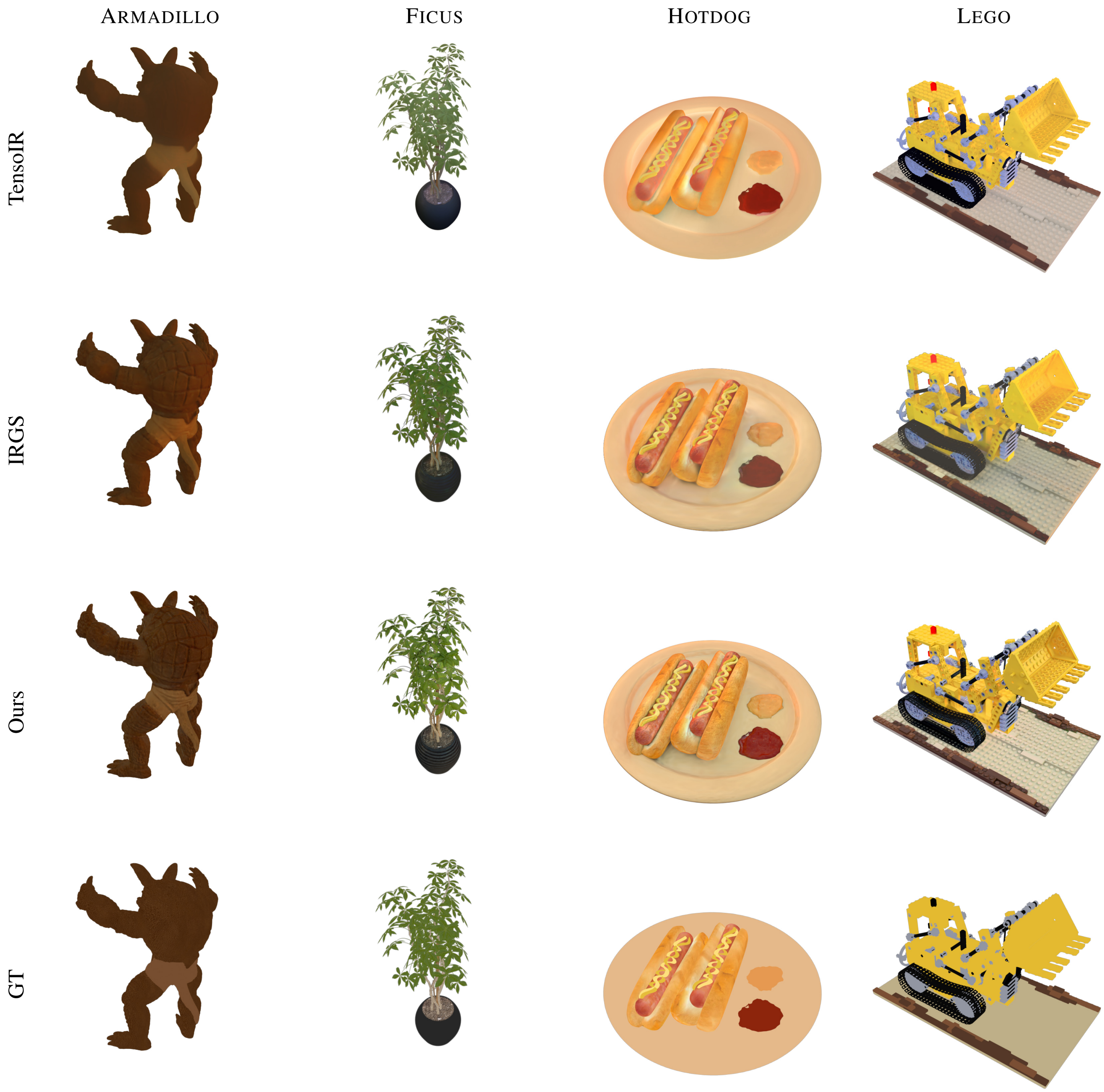

Figure 14. Qualitative comparison of the estimated albedo on TENSOIR SYNTHETIC dataset.

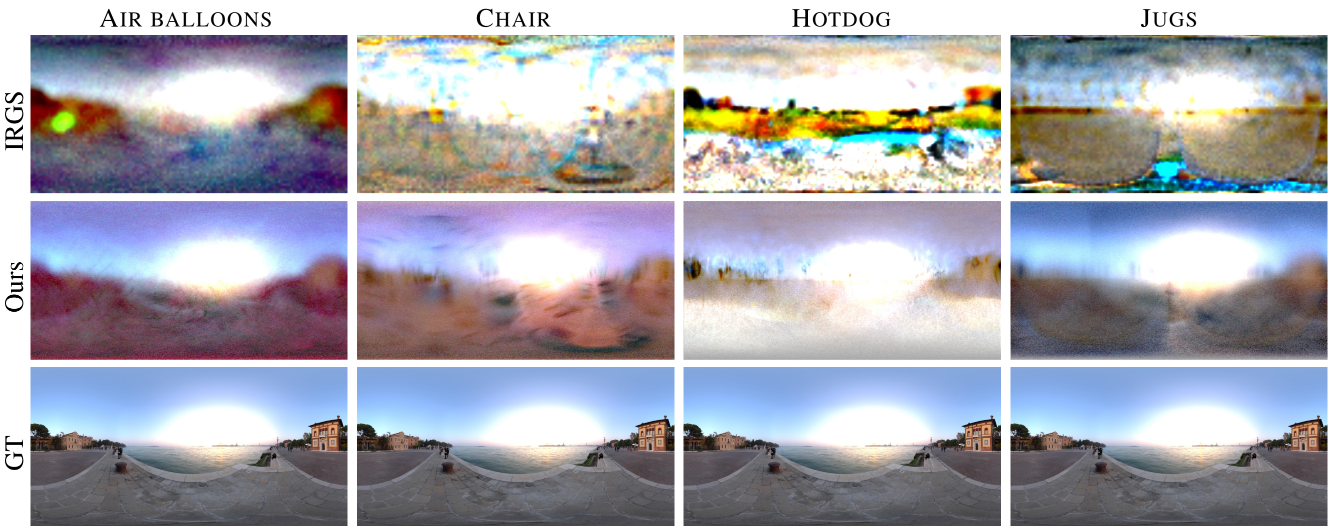

Figure 15. Qualitative comparison of the estimated environment maps on SYNTHETIC4RELIGHT dataset.

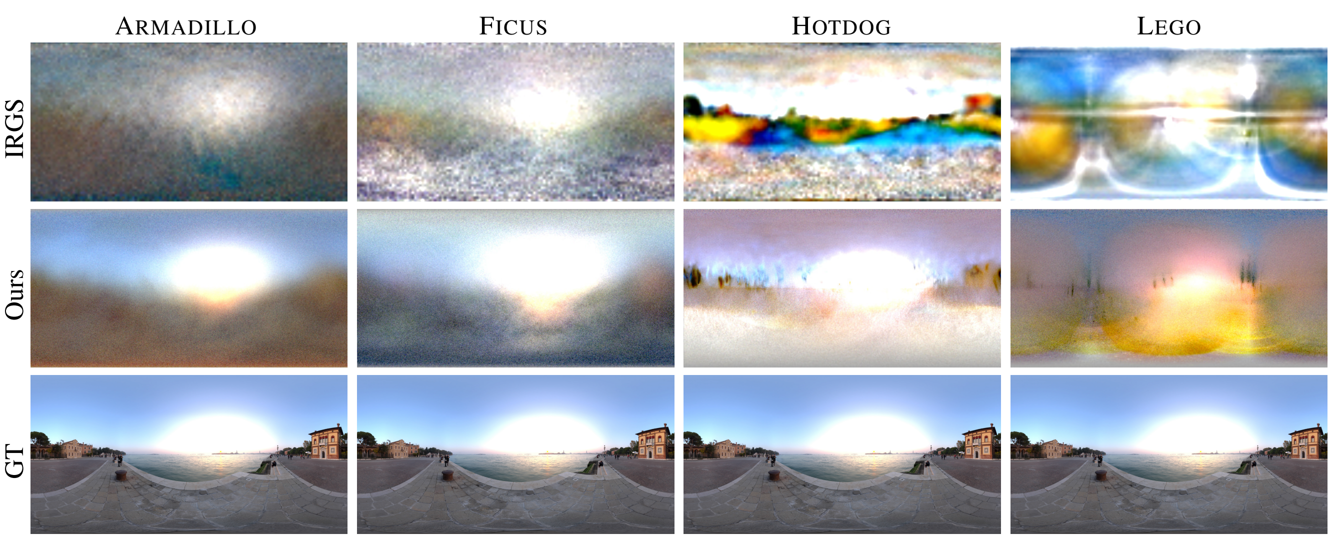

Figure 16. Qualitative comparison of the estimated environment maps on TENSOIR SYNTHETIC dataset.