

PIR: Photometric Inverse Rendering with Shading Cues Modeling and Surface Reflectance Regularization

Supplementary Material

1. More Details for the Method

1.1. Network Architectures

Neural SDF: $f_{\Theta_g}(\mathbf{x}) = (s, \mathbf{f}_{\text{geo}})$. We employ an 8-layer MLP featuring a hidden dimension of 256 and incorporate a skip connection at the fourth layer. The network input is the 3D coordinate \mathbf{x} encoded with a frequency of 6, to output the SDF value and an implicit local geometric feature. Before optimization, we perform geometric initialization on the network, as described by [1].

Neural diffuse albedo: $f_{\Theta_d}(\mathbf{x}, \mathbf{n}, \mathbf{f}) = \rho_d$. We use an 8-layer MLP featuring a hidden dimension of 256 and a skip connection at the fourth layer. The network inputs include the 3D coordinate \mathbf{x} encoded with 10 frequencies, surface normal, and geometric features. It outputs the diffuse albedo for point \mathbf{x} .

Neural specular albedo: $f_{\Theta_s}(\mathbf{x}, \mathbf{n}, \mathbf{f}) = \rho_s$. We employ a 4-layer MLP with a width of 256. The input 3D coordinate \mathbf{x} is encoded using 6 frequencies.

Neural roughness: $f_{\Theta_r}(\mathbf{x}, \mathbf{n}, \mathbf{f}_{\text{geo}}) = \rho_r$. We deploy a 4-layer MLP with a width of 256. The input 3D coordinate \mathbf{x} is encoded using 6 frequencies.

Blending scalar: $\gamma(\|\mathbf{x} - \mathbf{x}'\|, \mathbf{w}_i \cdot \mathbf{n}, \rho_r) = \gamma$. We use a 4-layer MLP with a width of 128. The dot product of normal and view direction uses 6 frequencies.

Neural DINO feature: $f_{\Theta_{\text{dino}}}(\mathbf{x}) = \mathbf{f}_{\text{dino}}$. We utilize a 4-layer MLP with a width of 256, where the input location \mathbf{x} is encoded with 6 frequencies, and the output features a dimension of 384.

1.2. Visibility Computation

For joint optimization of object geometry and light position, we determine the visibility of a surface point \mathbf{x} by uniformly sampling $N = 128$ points $\{\mathbf{x}_i\}_{i=1}^N$ along the path from surface point \mathbf{x} to the light source. We obtain the discrete opacity values $\{\alpha_i\}_{i=1}^N$ for these points using the unbiased SDF density conversion method introduced by NeuS [63]:

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{p}(t_i))) - \Phi_s(f(\mathbf{p}(t_{i+1})))}{\Phi_s(f(\mathbf{p}(t_i)))}, 0\right). \quad (10)$$

The light visibility of point \mathbf{x} in the direction of incident light \mathbf{w}_i is represented by the residual transmittance:

$$f_v(\mathbf{w}_i; \mathbf{x}) = 1 - \sum_{j=1}^N \alpha_j T_j, \quad (11)$$

where α_j is the density value at point \mathbf{x}_j , and $T_j = \prod_{k=1}^{j-1} (1 - \alpha_k)$ is the light transmittance at point \mathbf{x}_j in the direction \mathbf{w}_i .

1.3. Inter-reflection Computation

Importance Sampling To model the indirect illumination in scenes dynamically captured with directional lighting, we introduce an online computation approach that combines a differentiable layer and an importance sampling strategy. For a point \mathbf{x} and view direction \mathbf{w}_i , we consider a single light bounce and employ a ray marching towards the reflective direction:

$$\mathbf{w}_r = 2 \times \mathbf{n} - \mathbf{w}_i. \quad (12)$$

We then identify the secondary intersection point \mathbf{x}' . To determine if \mathbf{x}' is occluded from the light source, we uniformly sample 20 points along the path between the light source and the intersection point. The light is considered occluded by another surface if any of the sampled points exhibit a negative SDF value.

Figure 10 illustrates the process of inter-reflection modeling. If the secondary intersection point \mathbf{x}' is unobstructed, we compute the outgoing radiance at \mathbf{x}' using the flash-light's incoming radiance. The outgoing result is then combined with the blending coefficient to represent the indirect illumination.

Gradient Backpropagation Given that the blending coefficient is conditioned on the roughness property of the 3D point, there exists a correlation between the roughness property and the blending coefficient, introducing additional ambiguity in material estimation. In our experiments, we found that detaching the roughness of the secondary intersection point \mathbf{x}' prior to its input into the blending coefficient network leads to a more precise material decomposition. Moreover, the process of gradient backpropagation starts from the image loss, through the residual component, and into the material networks of the secondary intersection point \mathbf{x}' , fostering the alignment of secondary point radiance with inter-reflection cues. In our experiments, we discovered that disabling the optimization of local geometry at the secondary point reduces the complexity of the optimization process, leading to improved geometric reconstruction, especially in concave areas.

1.4. BRDF Renderer

Our BRDF implementation closely adheres to the Mitsuba roughplastic BRDF model [22], with the distribution pa-

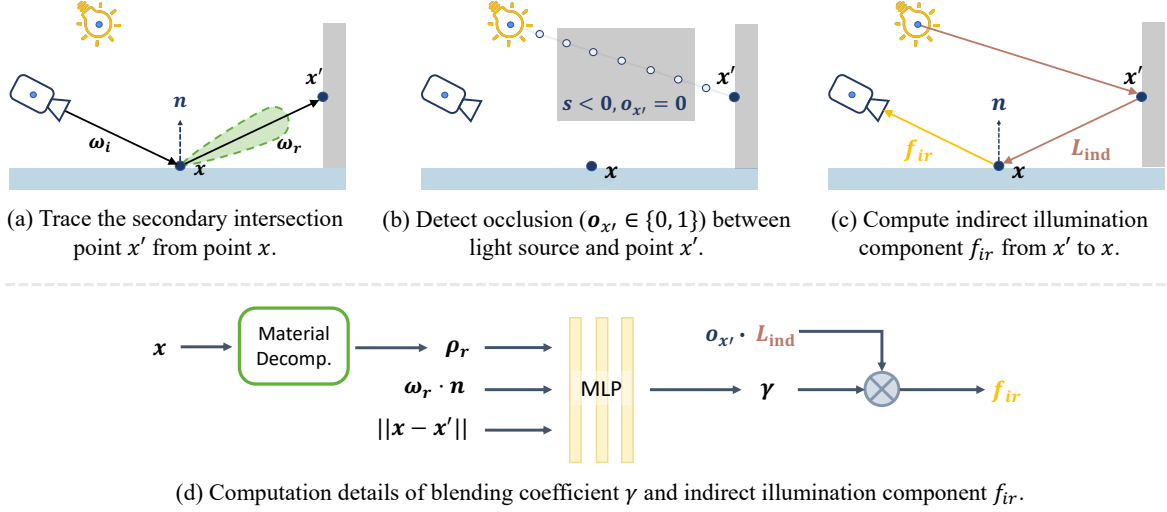


Figure 10. The illustration of inter-reflection modeling. We take into account rays that are physically rendered at secondary intersection points near the reflective direction of point x , as these rays contribute significantly to the indirect illumination.

parameter specifically set to ‘ggx’. For simplicity, we refer to our configuration as the roughplastic model. Default values are maintained for the internal and external Indices of Refraction and the nonlinear parameter.

Previous methods (IRON [84] and WildLight [12]) employing the renderer relied on an oversimplified BRDF model within an idealized setting where the camera and flashlight are collocated, neglecting the deviations between the camera and light source present in our capture setup. To address this limitation, we have enhanced the simplified roughplastic model to accommodate a broader range of scenarios, allowing for variations in both incident and outgoing light directions.

1.5. Training Details

The training process requires approximately 9 hours on a single RTX3090 GPU with 24GB of memory. We start by training NeuS over 100,000 iterations to initialize the geometry and diffuse albedo networks. For each training iteration, we utilize 512 randomly sampled pixels, employing an ℓ_1 loss along with an eikonal regularization loss. Prior to the rendering phase, we derive the feature maps of images by the pre-trained ViT-S/8 model [9] and executed 10,000 iterations with λ_4 set to 1.0 to extract the DINO feature from 2D feature maps to 3D surfaces. During the physics-based surface rendering stage with a total of 50,000 iterations, we fixed the geometry and lighting to warm up the BRDFs network for 2,000 iterations to stabilize the process, and subsequently, we carried out a joint optimization of the lighting, geometry, and BRDFs. The training of the blending coefficient network started at the 10,000th iteration. We set the size of rendered image patch as 128×128 and loss weights to $\lambda_1 = 10^{-4}$, $\lambda_2 = 0.1$, $\lambda_3 = 10^{-5}$ and $\lambda_4 = 10^{-5}$. All

networks are optimized by corresponding Adam optimizers with learning rate 10^{-4} .

2. More Details for the Dataset

DRV Dataset We acquired the DRV dataset [5] from the authors, comprising five scenes: *Dragon*, *Girl*, *Pony*, *Tree*, and *Cartoon*. Each scene has approximately 400 images, split between training and test sets. The dataset captures images in a darkroom, utilizing a nearly collocated camera-light setup.

Luan Dataset The Luan dataset [40] was captured using a casual smartphone. We noticed that the images exhibit significant noise and motion blur, together with varying exposure times and white balance settings during capture. This inconsistency introduces challenges in maintaining multi-view consistency. We evaluated the scene *Xmen*, which includes 136 images, to compare novel view rendering and material decomposition against the IRON method.

Self-captured Dataset For capturing real-world images, we employed an iPhone 15 to shoot in RAW format, ensuring a linear camera response. Across all photos, we maintained consistent settings for the camera’s exposure time, focus, and white balance. Specifically, the ISO value and shutter speed (exposure time) were fixed at 100 and 1/250s, respectively, with the white balance adjusted to 3,800 Kelvin degrees. Our collection encompasses 5 scenes: *Toy*, *fruit*, *Panda*, *Assassin*, and *Bear*, with the number of images per scene varying from 120 to 400. Camera poses were derived using COLMAP [50], and objects were scaled to fit within a unit sphere based on the reconstructed point cloud. The photography sessions took place in a darkroom, positioning the camera 0.15 to 0.3 meters away from



Figure 11. Qualitative comparisons with state-of-art methods on the synthetic dataset (*dragon* and *horse*). The materials of NeILF++[82] are *Base Color*, *Metallic*, *Roughness* defined by simplified Disney principled BRDF.

each object. To achieve comprehensive coverage, we systematically moved the camera in a spiral pattern around the subjects. The separation between the camera lens and the flashlight on the iPhone, roughly 0.015m, results in an approximate 3-degree variation between viewing and lighting angles at a standard distance of 0.25m from the object.

3. More Results and Comparisons

We primarily compare our results with those from IRON [84] and WildLight [12]. Notably, WildLight was unable to reconstruct the synthetic data for *duck*, and as such, its results are not presented in the table within the main paper.

3.1. Results on Synthetic Data

In Table 4, we offer comprehensive results for each synthetic scene captured under casual conditions. Additionally, we provide a qualitative comparison of novel view rendering and material decomposition between our method and earlier methods, as illustrated in Fig. 11. For the *dragon* scene, our method produces a diffuse albedo with less indirect illumination incorporated into the materials. In the *horse* scene, our material decomposition results demonstrate a reduced influence of self-shadows, showing a closer

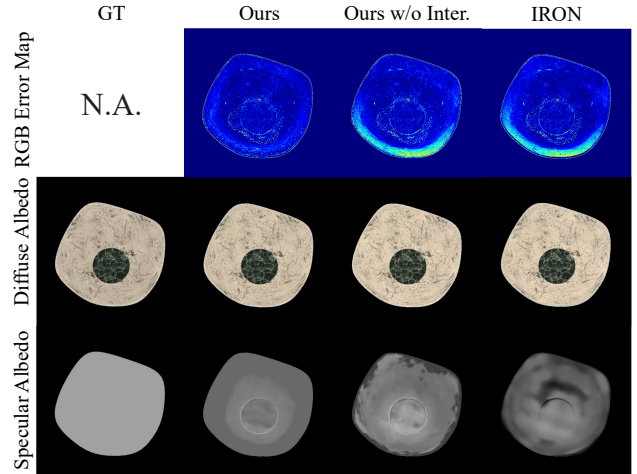


Figure 12. Ablation study on inter-reflection in *marble bowl*. alignment with the ground truth than those obtained with IRON. Even in extremely concave regions, our method is more robust than the previous method, as shown in Fig. 12.

3.2. Results on Real Data

In Fig. 14, we present our dataset’s novel view rendering and material decomposition outcomes. The IRON

Scene	Method	Roughness	Diffuse Albedo			Specular Albedo			Novel View Synthesis		
		MSE $\times 10^{-3}$ ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
<i>duck</i>	WildLight	-	-	-	-	-	-	-	-	-	-
	IRON	2.640	24.693	0.9631	0.0483	18.669	0.9017	0.1075	31.845	0.9855	0.0320
	Ours	1.059	34.871	0.9852	0.0355	23.402	0.9474	0.0730	35.164	0.9912	0.0273
<i>maneki</i>	WildLight	93.59	18.151	0.7351	0.4472	12.413	0.8114	0.2497	29.913	0.9400	0.0787
	IRON	1.455	35.367	0.9805	0.0238	18.967	0.8369	0.1732	30.087	0.9550	0.0468
	Ours	0.467	36.098	0.9880	0.0184	22.245	0.9371	0.0726	32.979	0.9729	0.0340
<i>horse</i>	WildLight	40.23	24.625	0.9507	0.1032	16.997	0.8401	0.2056	32.032	0.9669	0.0520
	IRON	2.198	31.903	0.9826	0.0363	29.323	0.8701	0.1275	31.713	0.9808	0.0366
	Ours	1.509	33.573	0.9880	0.0194	33.071	0.9259	0.0917	34.206	0.9831	0.0321
<i>marble bowl</i>	WildLight	165.2	22.613	0.8862	0.1379	15.600	0.8261	0.2135	28.219	0.9252	0.0981
	IRON	0.321	29.258	0.9623	0.0518	35.035	0.8947	0.1553	27.403	0.9602	0.0583
	Ours	0.172	29.881	0.9647	0.0493	39.972	0.9729	0.0660	29.209	0.9640	0.0591
<i>dragon</i>	WildLight	120.8	33.679	0.9208	0.1108	14.432	0.7840	0.2453	26.546	0.9078	0.1155
	IRON	2.815	36.470	0.9675	0.0575	30.902	0.7610	0.2295	25.516	0.9257	0.0876
	Ours	0.923	38.720	0.9735	0.0390	34.894	0.8152	0.1772	27.870	0.9406	0.0766
<i>armchair</i>	WildLight	117.9	30.116	0.9242	0.1282	15.748	0.8260	0.2136	29.126	0.9100	0.1200
	IRON	1.612	41.361	0.9818	0.0416	21.960	0.8333	0.1938	27.752	0.9555	0.0734
	Ours	1.155	41.518	0.9833	0.0370	25.442	0.8959	0.1438	31.916	0.9655	0.0642

Table 4. Complete results on the synthetic dataset.

Method	<i>Pony</i>		<i>Girl</i>		<i>Tree</i>		<i>Dragon</i>		<i>Cartoon</i>		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
IRON [84]	29.269	0.9150	27.136	0.9326	31.641	0.9464	32.421	0.9317	30.773	0.9587	30.248	0.9369
Ours	30.092	0.9414	27.589	0.9365	31.765	0.9464	32.251	0.9306	30.975	0.9589	30.534	0.9428

Table 5. Quantitative comparison of novel view rendering on DRV dataset.

method often incorporates indirect illumination into the diffuse albedo, particularly in concave regions, as observed in the *Fruit* scene. Additionally, specular albedoes produced by the IRON method are adversely affected by self-shadows and inter-reflections, as highlighted in specific boxes.

In Table 5, we provide a quantitative comparison that underscores the enhanced performance of our method compared to IRON in terms of novel view rendering within the DRV real dataset. Figure 15 and Fig. 16 complement this with side-by-side qualitative comparisons of our method against IRON regarding material decomposition. Leveraging DINO regularization for surface decomposition, which effectively clusters similar materials, our approach produces more accurate results for material decomposition, especially in scenarios with a skewed view distribution. We observe that IRON’s evaluation metrics for the *Dragon* scene slightly exceed those of our method, this disparity is primarily due to its collocated camera-lighting setup, which inherently minimizes the occurrence of self-shadows within the scene.

3.3. Failure Case

Like many neural surface reconstruction methods, both COLMAP and NeuS presuppose Lambertian observation to guarantee multi-view consistency. Following the same approach as IRON, our method primarily depends on NeuS for geometry initialization but struggles to reconstruct objects with reflective surfaces, as depicted in Fig. 13. The surfaces

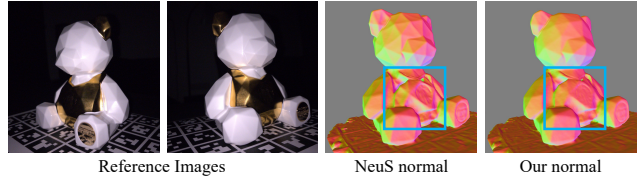


Figure 13. A failure case on *bear* with reflective surfaces.

reconstructed by NeuS and our method exhibit holes within reflective regions.

4. Video Demos

In the video, we present more comprehensive results to demonstrate the effectiveness of our design, along with additional comparison cases between our method and other inverse rendering methods. Furthermore, we render the reconstructed 3D assets using a traditional graphical pipeline to illustrate their practical applications.

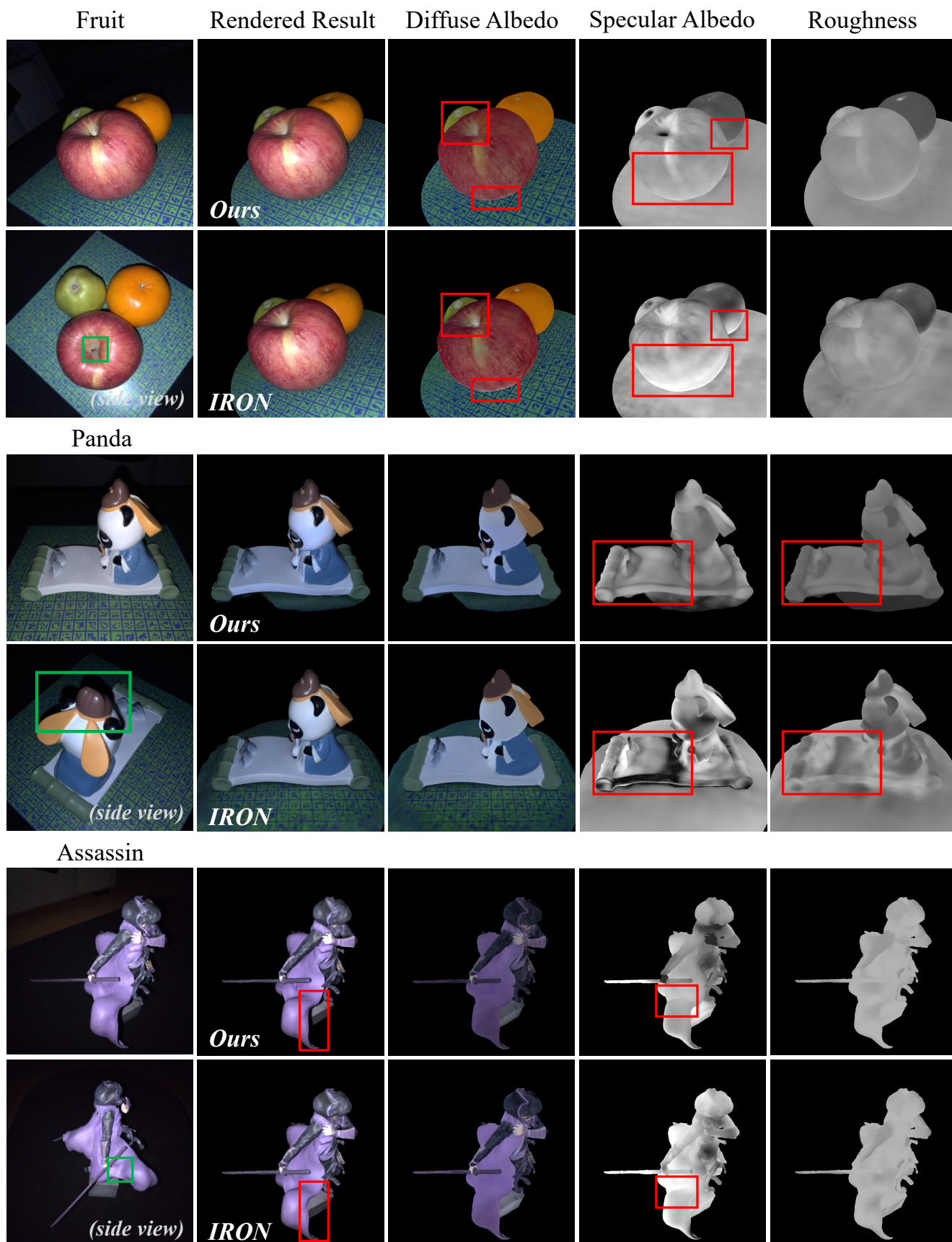


Figure 14. More visual results of material decomposition on our dataset.

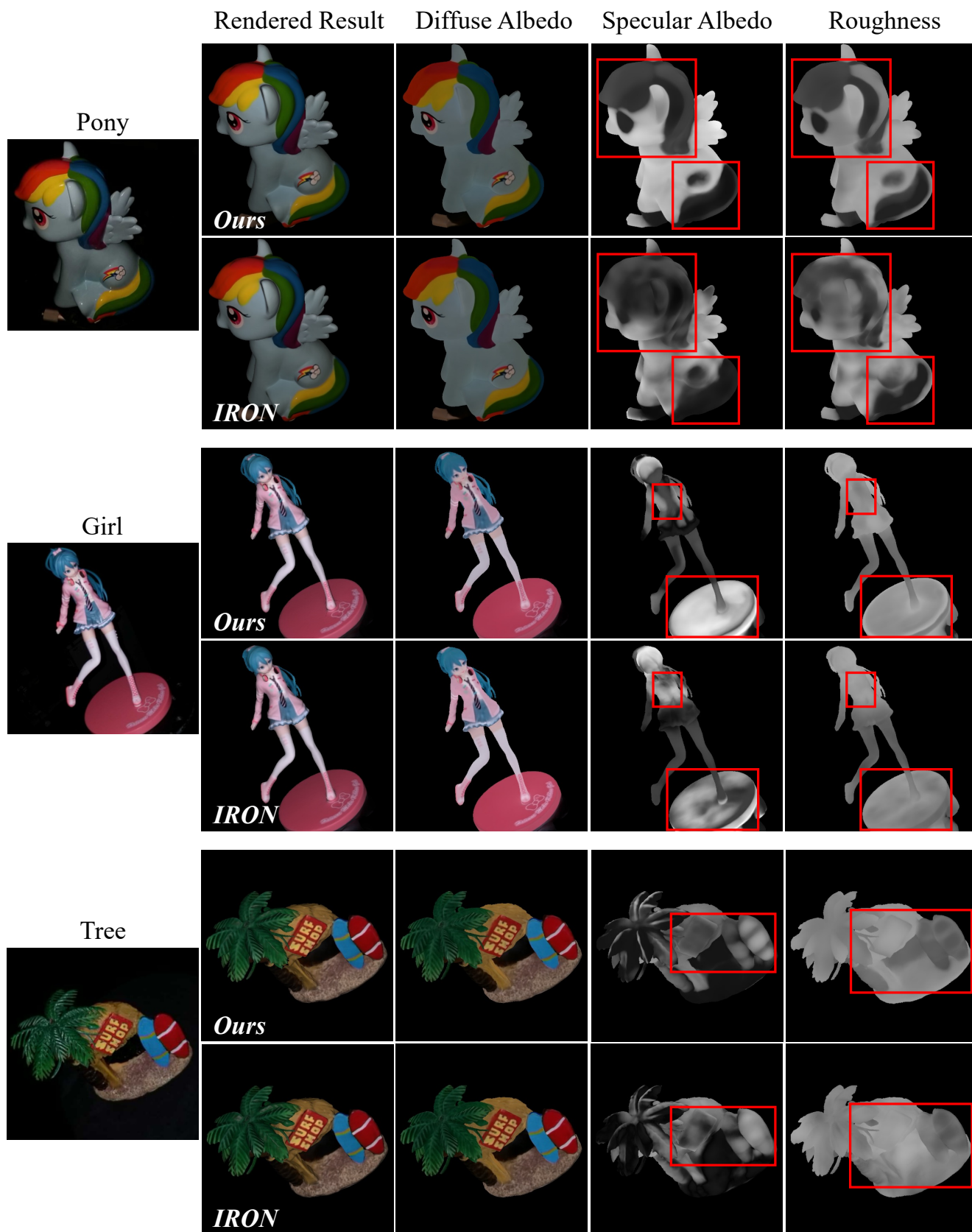


Figure 15. Visual results of material decomposition on DRV dataset.

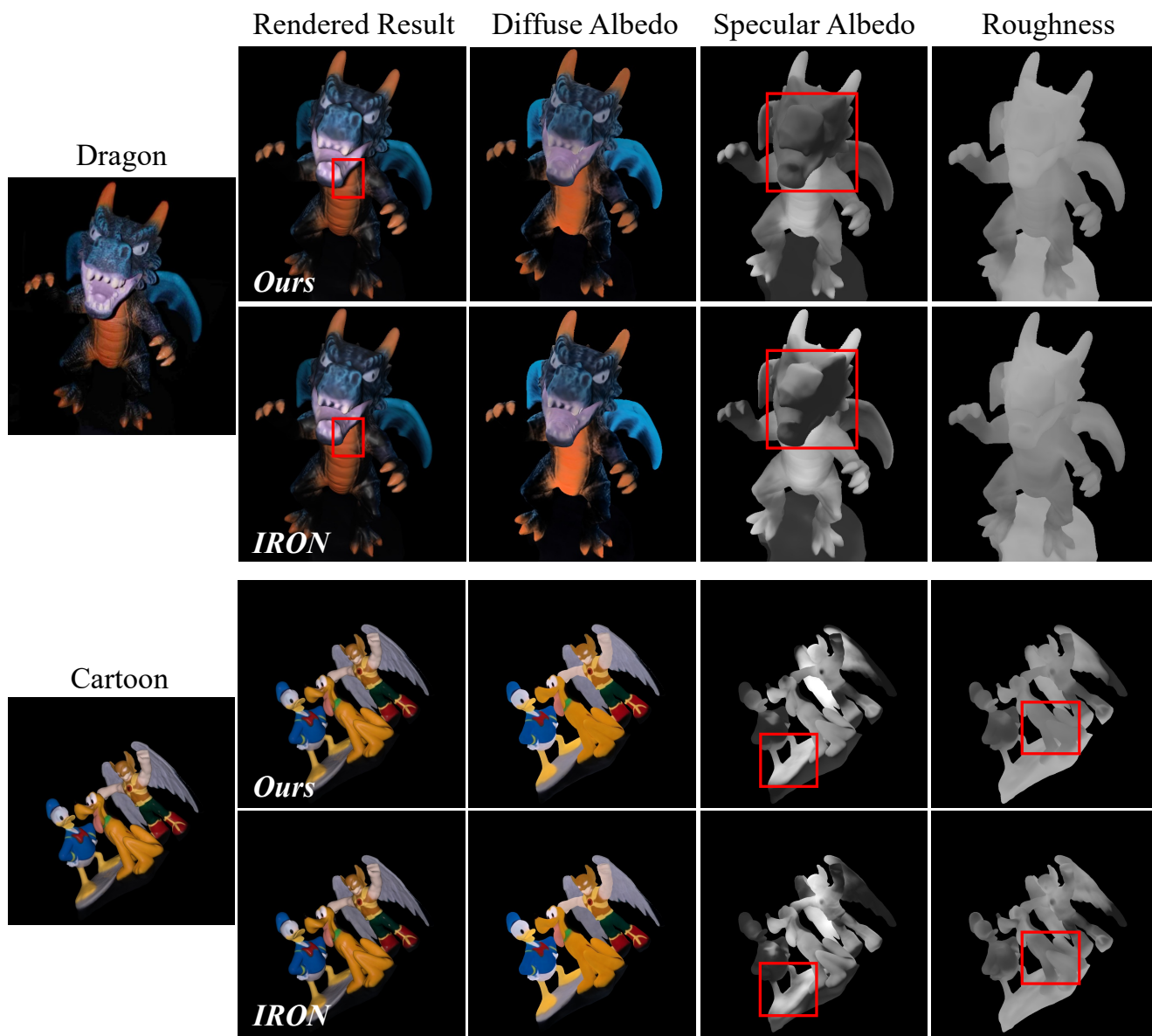


Figure 16. Visual results of material decomposition on DRV dataset (continued).