11 Supplementary Material

11.1 Details on Geometric Reconstruction

Once our approach convert the groups of input images into feature maps, they are directly fed as input to existing multi-view stereo techniques (be it COLMAP, NeuS or Neuralangelo), with only minor modifications: the number of channels of an input image is changed from 3 (RGB) to 12, to match the dimension of our features. The objective function stays the same as in respective reconstruction approaches. Moreover, for methods based on inverse rendering such as NeuS or Neuralangelo, we further modify their rendering process to produce 12-channel output images, in order to compute the loss against our input feature maps. This is done by changing the output dimension of the final fully connected layer in NeuS/Neuralangelo from 3 to 12.

11.2 Details on Appearance Reconstruction

After geometric reconstruction, we establish a uv-parameterization over object surfaces, and compute BRDF parameters at each valid texel via differentiable optimization. While not being tied to any specific model, we adopt the anisotropic GGX BRDF in this paper:

$$f(\omega_\mathbf{i};\omega_\mathbf{o},\mathbf{p}) = \frac{\rho_d}{\pi} + \rho_s \frac{D_{GGX}(\omega_\mathbf{h};\alpha_x,\alpha_y)F(\omega_\mathbf{i},\omega_\mathbf{h})G_{GGX}(\omega_\mathbf{i},\omega_\mathbf{o};\alpha_x,\alpha_y)}{4(\omega_\mathbf{i}\cdot\mathbf{n_p})(\omega_\mathbf{o}\cdot\mathbf{n_p})}.$$

Here ρ_d/ρ_s are the diffuse/specular albedo, α_x/α_y are the roughness parameters, and $\omega_\mathbf{h}$ is the half vector. D_{GGX} is the microfacet distribution function, F is the Fresnel term and G_{GGX} accounts for shadowing/masking effects. The BRDF model is defined in the local frame $\mathbf{n_p}/\mathbf{t_p}$ of $\mathbf{p}$, where $\mathbf{n_p}/\mathbf{t_p}$ are the normal and tangent, respectively.

To fit BRDF parameters for a particular texel, we first project its corresponding 3D position to all visible views to gather its image measurements. Next, we employ a 16D latent vector to represent the BRDF parameters: a decoder network is also trained to transform the latent vector to the parameters $(\rho_d, \rho_s, \alpha_x, \alpha_y, \mathbf{n_p}, \mathbf{t_p})$. These parameters will be used to produce rendering results, whose difference with the aforementioned image measurements is minimized. All latent vectors and the corresponding decoder are jointly optimized. Finally, we convert the latent vector at each texel to anisotropic GGX BRDF parameters, and store them in texture maps as the appearance result (as visualized in Fig. 16).

11.3 Features Incorporating Correlated Factors

According to Tab. 1, diffuse albedos and normals are mostly correlated with our learned features. Here we test the impact of replacing part of our learned features with the predictions of these highly correlated factors. Specifically, we encourage our network to learn to explicitly predict the first 6D of the output feature as diffuse albedo and normal, with the following modified loss:

$$L = \lambda_0 L_0 + \lambda_1 L_1 + \lambda_2 L_2 + \lambda_p L_p + \lambda_{\text{reg}} L_{\text{reg}},$$

where

$$L_{\text{reg}} = L_{\text{diffuse}} + L_{\text{normal}}.$$

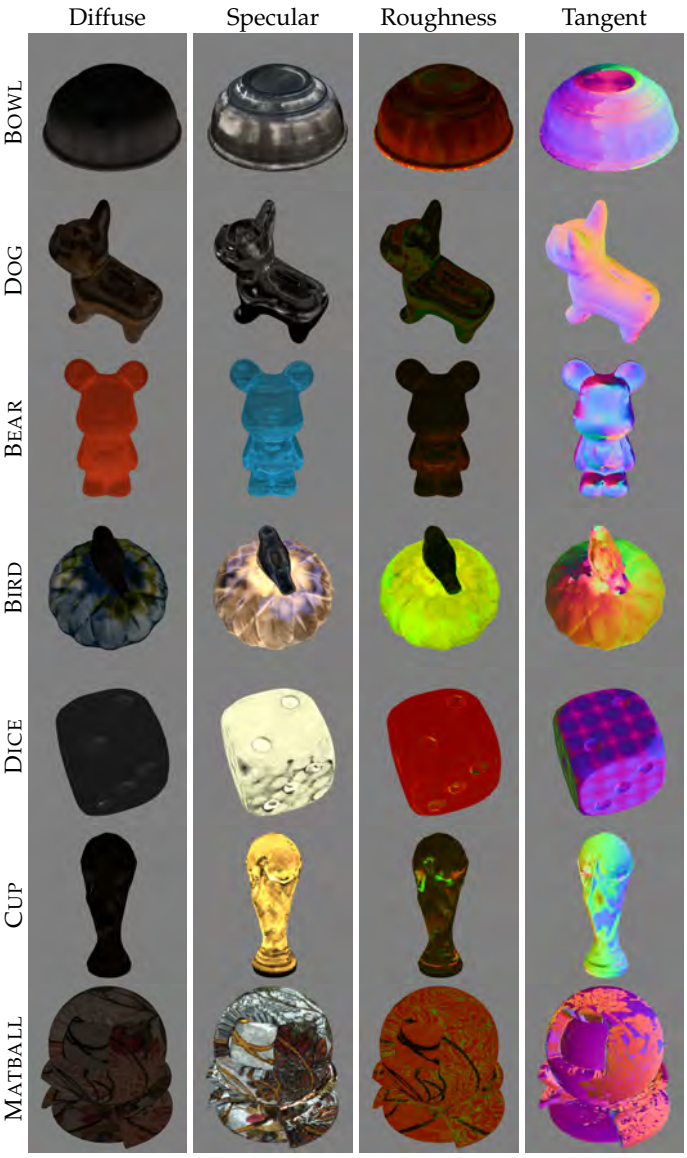

Fig. 16: Reconstructed SVBRDF parameters. For visualization purpose, each tangent is added with $(1,1,1)$ and then divided by 2 to fit to the range of $[0,1]$; the specular albedo is re-scaled; and α_x/α_y are visualized in the red/green channel.

We reserve the first 6 dimensions of the final feature for diffuse and normal predictions, and leave the remaining dimensions for data-learned features. Here L_{diffuse} represents the mean squared error (MSE) between the first three dimensions of the final feature and the ground-truth diffuse albedo, while L_{normal} is the MSE between the next three dimensions of the final feature and the ground-truth normal. We set $\lambda_{\text{reg}} = 5$ in our experiment.

We test the new features on reconstructing the geometry of Matball. Its Chamfer distance increases from 5.13 (our features) to 5.28 (new features). We find that while it is faster to train the new features due to the extra regularization term, the reconstruction quality is reduced, as the features are not completely learned from data.