Intrinsic Scene Properties from a Single RGB-D Image Supplementary Material

Jonathan T. Barron and Jitendra Malik
UC Berkeley
{barron, malik}@eecs.berkeley.edu

1. Pseudo-synthetic Dataset

To quantitatively evaluate our model we must first produce a dataset of "scene"-like images: images with occlusion and spatially varying illumination. We will do this by layering the objects in the MIT intrinsic images dataset [4] (using the ground-truth shapes produced by Berkeley [2]) into a scene, and then rendering the resulting shapes using a randomly sampled spatially-varying illumination. For each scene we also produce a noisy Kinect-like depth map for use as input to our model.

The creation of our dataset is as follows: The 20 objects in the MIT dataset are downsampled by a factor of two, such that our resulting scenes can be reasonably sized. We split the objects into training and test sets, using the same split as in [1, 2]. We then synthesize 10 training and 10 test scenes, where training scenes are composed of training objects, and test scenes are composed of test objects. To generate a scene, we iteratively layer objects on top of each other, taking care to place each object in the largest part of the scene that is still empty. For each scene we generate a layered depth map, normal map, and reflectance image. Given these, we then synthesize a natural illumination to generate a shading image. All of our scenes are 256×256 pixels.

Natural scenes have complicated, spatially-varying illumination due to attenuation, shadowing, interreflection, etc. Therefore, to make this dataset a somewhat realistic surrogate for real images of natural scenes, we cannot simply use one global model of illumination, as was done in [1]. We will instead synthesize our own spatially-varying illumination as a mixture of many attenuated point-light sources. For each scene, we generate 50 lights, each of whose position is generated in a uniform region twice the size of the volume enveloped by the objects in the scene, and whose color c_i is randomly sampled from the average RGB value of an image in the sIBL Archive[1]. For each light, we compute the distance d_i of each depth-map pixel to the light source and compute an attenuation $a_i = 1/(1 + d_i^2/40000)$, where 40000 (which controls the amount of attenuation) is chosen manually such that the resulting scenes look reasonable. For each pixel, we compute the unit vector ℓ_i from that pixel's location in the scene to the light source's position. We then render the scene according to attenuated Lambertian reflectance ($c_i \times \max(0, a_i(n_i \cdot \ell_i))$) for each light source, and take the sum of all of these renderings (after dividing by the max intensity) as the complete shading image. The final image, which is this shading image multiplied by the previously-generated reflectance image, will be the RGB input to our system.

In parallel with rendering these images, we render a "light probe" surface that has the same depths as the scene, but whose normal field is chosen to uniformly sample the space of orientations. We will evaluate the fidelity of our recovered illumination by rendering this "light probe" surface according to our recovered illumination, and comparing it to the ground-truth produced during this synthesis.

In the paper we described a probabilistic model of the noise present in Kinect depth maps, for use during inference. This noise model is not sufficient for our experiments: we must be able to synthetically generate noisy depth maps that are similar to Kinect depth maps, for the purposes of tuning our model parameters and empirically evaluating the accuracy of our model on pseudo-synthetic data. We therefore present a generative model to construct a noisy depth map: Given a "true" depth-map Z^* in centimeters, we first construct a disparity map. We replace each pixel in the disparity map with the bilinearly interpolated value of a location near that pixel where the shift is drawn from a normal distribution $\sigma = 1/2$, then we add IID Gaussian noise to each pixel of the disparity map ($\sigma = 1/6$), and then we translate all pixels in the disparity map by a shift drawn from a normal distribution ($\sigma = 1/4$). The first shuffling and injection of noise is intended to simulate sensor noise in the Kinect, and the second image-wide shuffling is intended to simulate a slightly inaccurate alignment between the image and the depth map, which we often see in even well-aligned Kinect data. We quantize the shuffled disparity by rounding it to the nearest integer, and then convert

[1] http://www.hdrlabs.com/sibl/archive.html

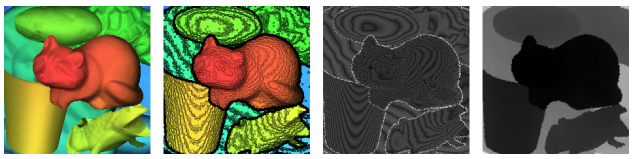


Figure 1. A visualization of how we introduce noise to Kinect images. In the first column we have a ground-truth depth-map, and in the second we have our corrupted version of it that we will use as a proxy for Kinect data. The third column shows the difference between the two, where we see the stripe-like noise introduced by quantization, as well as the noise near the boundaries of the objects introduced by shuffling and mis-aligning the image. The fourth column is a visualization of our error model, where we see error increases with depth, in accordance with our understanding of binocular stereo.

the disparity measurements back into depth measurements. Formally, our procedure for corrupting a ground-truth depth map is:

$$\hat{Z} \leftarrow \frac{35130}{\lfloor 35130/(\text{shuffle}(Z^*)) + \mathcal{N}(0, (1/6)^2) + 0.5 \rfloor} \quad (1)$$

The constant 35130 is derived from the baseline of the Kinect sensors. See Figure 1 for a visualization of these synthetic noisy depth maps. Though our model for generating noise is different from our model for inference in the face of noise, manual investigation of the data suggests that the models largely agree — the marginal distribution of the noisy depth maps generated from Equation 1 appears to match the prior posed in the paper, in terms of the width of the uniform distribution and the shape of the hyper-laplacian tail. Perhaps more importantly, our synthetically noisy depth maps look similar to the real depth maps in the NYU Depth Dataset [6].

2. Error Metrics

Our error metrics are a revision of those in [1, 2]. We use six error metrics: two for shape, one for shading, one for reflectance, one joint error metric for shading and reflectance introduced in [4], which we will refer to as rs-MSE (though which the original authors call "LMSE"), and one for illumination.

Our first shape error metric is the shift-invariant mean absolute error between our recovered shape $\hat{Z}$ and the true shape Z^*:

$$Z\text{-MAE}(\hat{Z}, Z^*) = \frac{1}{n} \min_{\beta} \sum_{x,y} \left| \hat{Z}_{x,y} - Z^*_{x,y} + \beta \right| \quad (2)$$

Our second shape error metric is the mean angle (in radians) between our recovered normal field $\hat{N}$ and the true normal field N^*:

$$N\text{-MAE}(\hat{N}, N^*) = \frac{1}{n} \sum_{x,y} \arccos\left(\hat{N}_{x,y} \cdot N^*_{x,y} \right) \quad (3)$$

Together these two error metrics measure the important quantities of the recovered depth map. Z-MAE is sensitive to the overall shape of the depth map, modulo any global shift (which cannot be directly observed except in the Kinect depth map) and N-MAE is sensitive to the local orientation of each pixel in the depth map.

For shading and reflectance, we use the scale-invariant mean squared error of our recovered shading image $\hat{s} = \exp(\hat{S})$ and reflectance image $\hat{r} = \exp(\hat{R})$

$$s\text{-MSE}(\hat{s}, s^*) = \frac{1}{n} \min_{\alpha} \sum_{x,y} \left\| \alpha \hat{s}_{x,y} - s^*_{x,y} \right\|_2^2 \quad (4)$$

$$r\text{-MSE}(\hat{r}, r^*) = \frac{1}{n} \min_{\alpha} \sum_{x,y} \left\| \alpha \hat{r}_{x,y} - r^*_{x,y} \right\|_2^2 \quad (5)$$

These error metrics are invariant to absolute scaling of the entire image, but not to each RGB channel individually, and therefore measure errors in overall color and white-balance.

We also use the error metric introduced with the MIT intrinsic images dataset [4], which the authors refer to as LMSE, but which we will call rs-MSE to avoid confusion with L-MSE. This metric measures local scale-invariant error for each channel of both reflectance and shading, thereby measuring high-frequency errors in both that are insensitive to overall intensity and color.

Our error metric for illumination is the scale-invariant MSE of a rendering of our recovered illumination $\{\hat{\boldsymbol{L}}, \hat{\boldsymbol{V}}\}$ on a "light-probe"-like surface, compared to a ground-truth light-probe surface that was rendered alongside the image when generating the data:

$$L\text{-MSE}(\hat{\boldsymbol{L}}, \hat{\boldsymbol{V}}, S^p) = \frac{1}{n} \min_{\alpha} \sum_{x,y} ||\alpha S(N^p, \hat{\boldsymbol{L}}, \hat{\boldsymbol{V}})_{x,y} - S^p_{x,y}||_2^2 \quad (6)$$

Where N^p is the light-probe normal field we use, and S^p is a rendering of that normal field under the ground-truth attenuated point light-sources used in generating our pseudo-synthetic imagery.

When evaluating these error metrics, we take the geometric mean of each metric over each image in the test set. When producing our "average" error metric, we take the geometric mean of each error metric. The geometric mean is used because it is insensitive to scalings of the constituent error metrics, so arbitrarily large error metrics or unusually difficult images don't have more influence. The geometric mean is therefore difficult to trivially minimize in practice.

References

[1] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. *ECCV*, 2012.

[2] J. T. Barron and J. Malik. Shape, albedo, and illumination from a single image of an unknown object. *CVPR*, 2012.

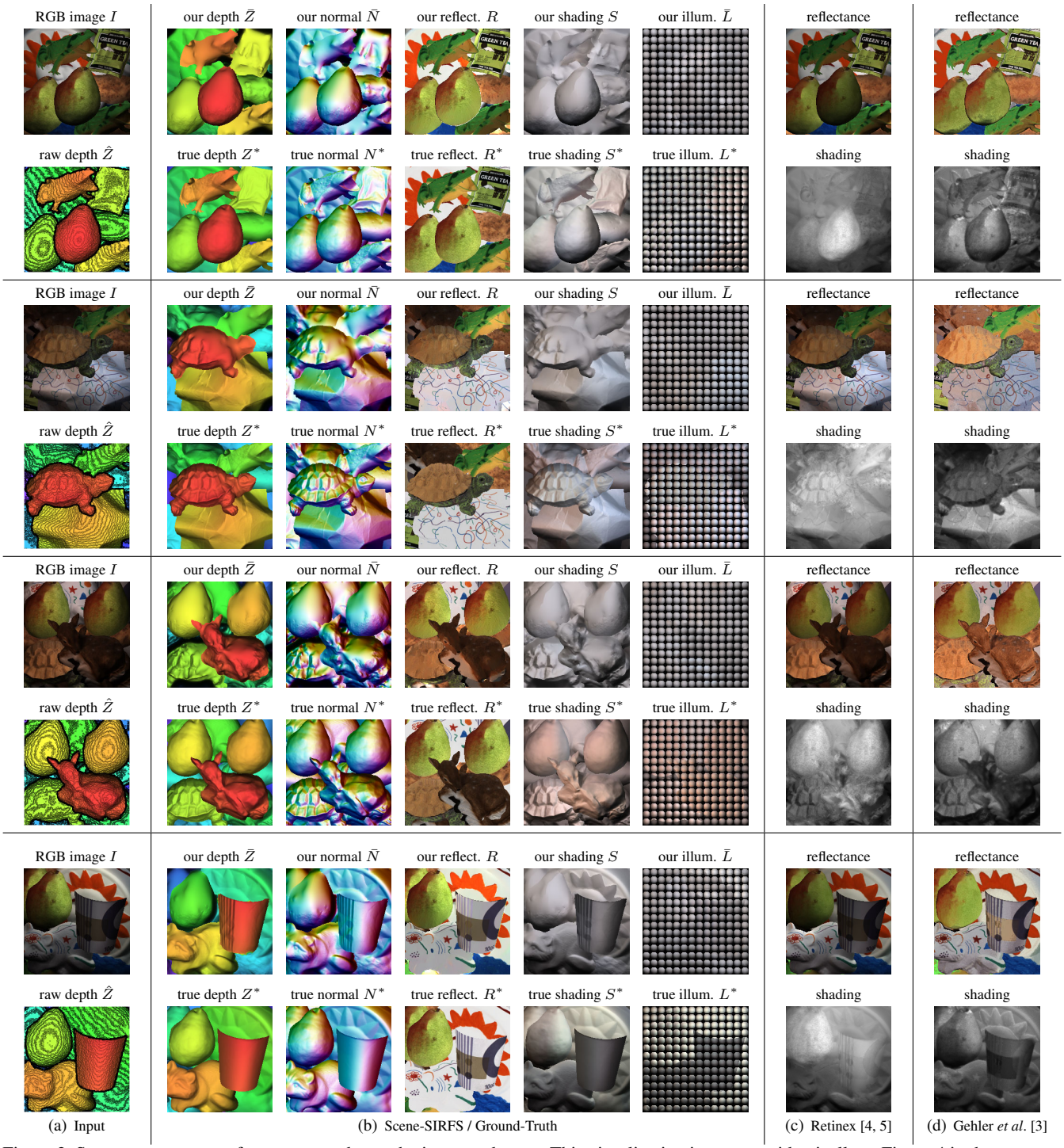


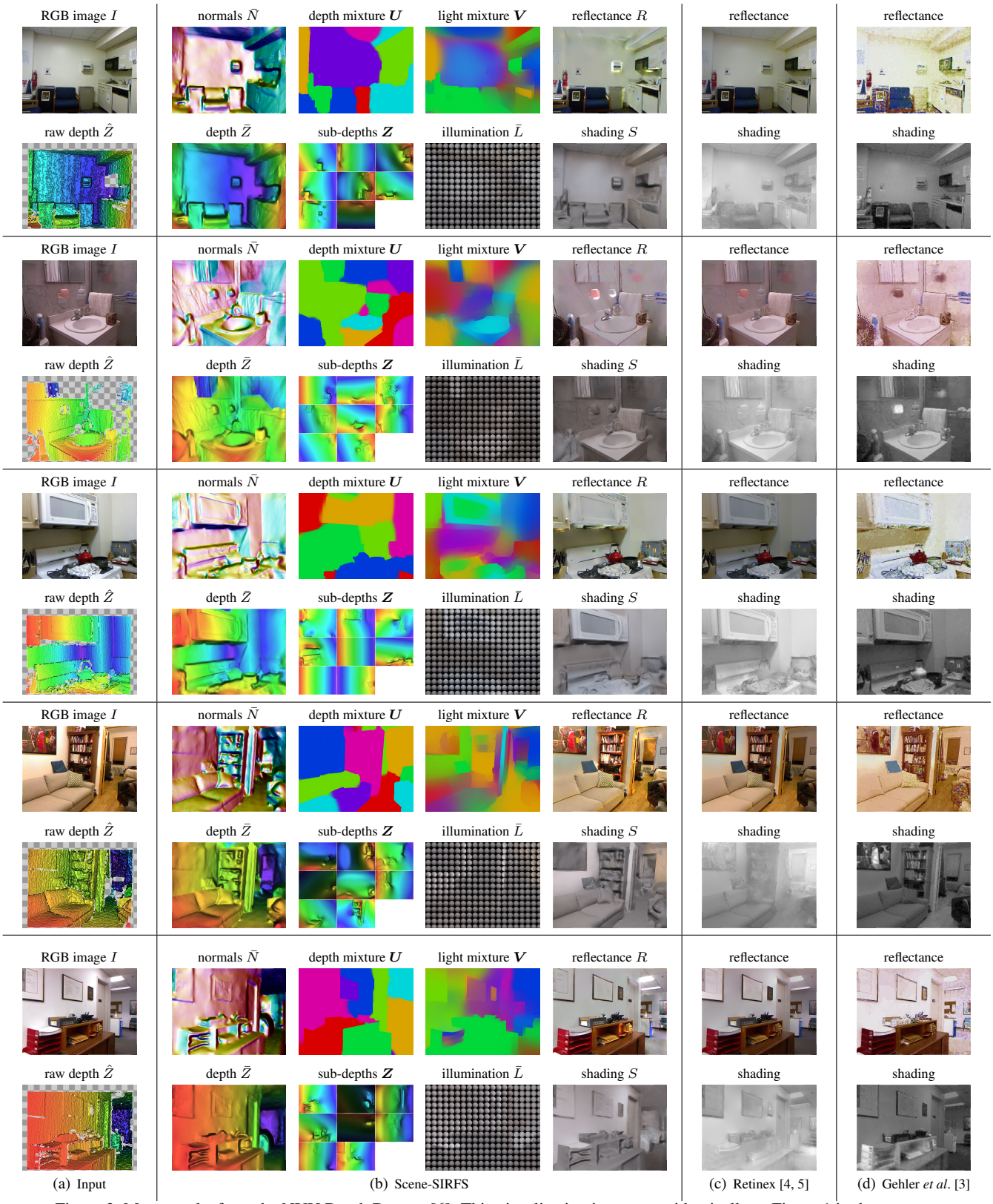
Figure 2. Some test-set scenes from our pseudo-synthetic scene dataset. This visualization is structure identically to Figure 4 in the paper.

[3] P. Gehler, C. Rother, M. Kiefel, L. Zhang, and B. Schoelkopf. Recovering intrinsic images with a global sparsity prior on reflectance. *NIPS*, 2011.

[4] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. *ICCV*, 2009.

[5] B. K. P. Horn. Determining lightness from an image. *Computer Graphics and Image Processing*, 1974.

[6] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012.



(a) Input (b) Scene-SIRFS (c) Retinex [4, 5] (d) Gehler *et al.* [3]

Figure 3. More results from the NYU Depth Dataset [6]. This visualization is structure identically to Figure 1 in the paper.

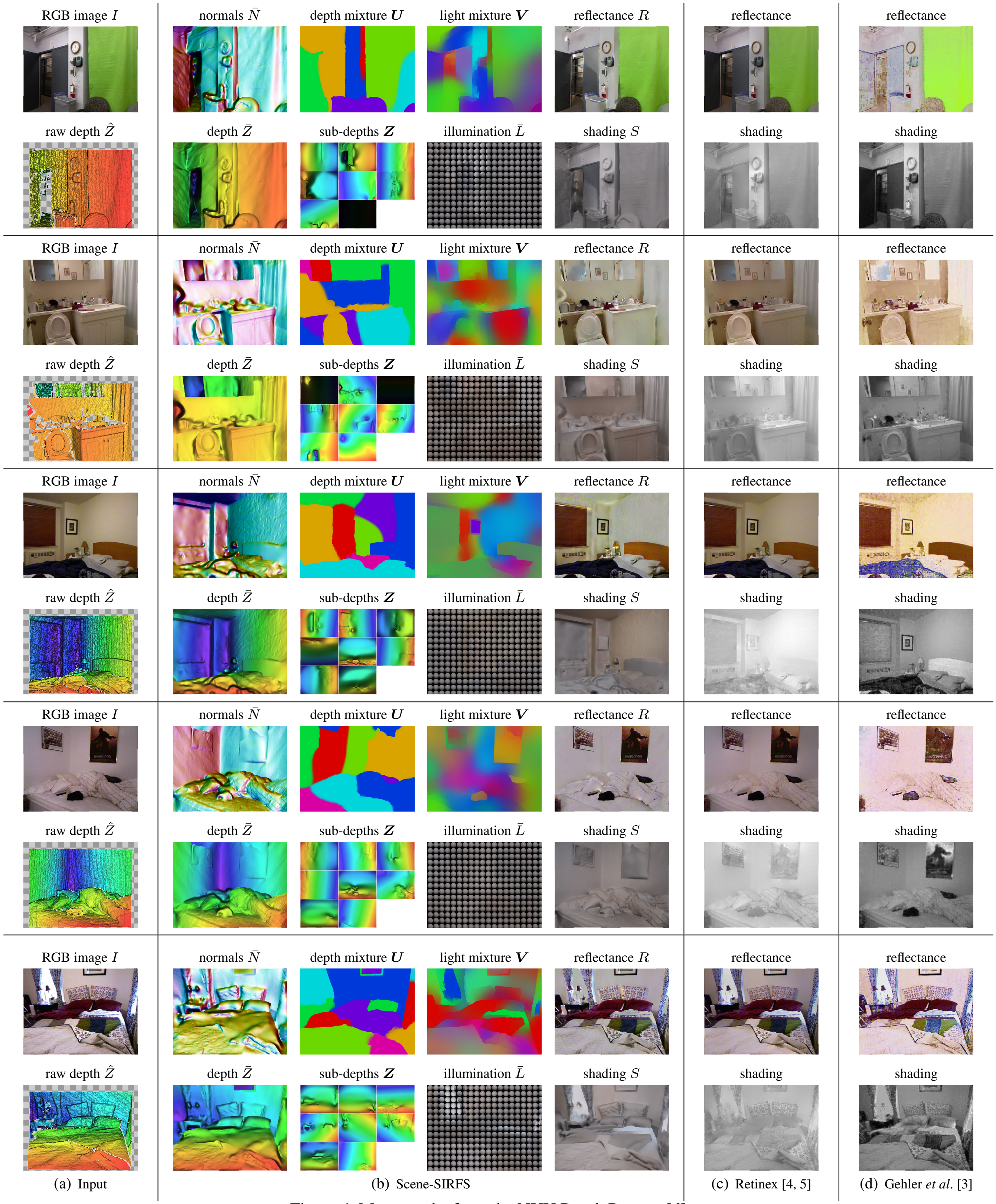


Figure 4. More results from the NYU Depth Dataset [6].

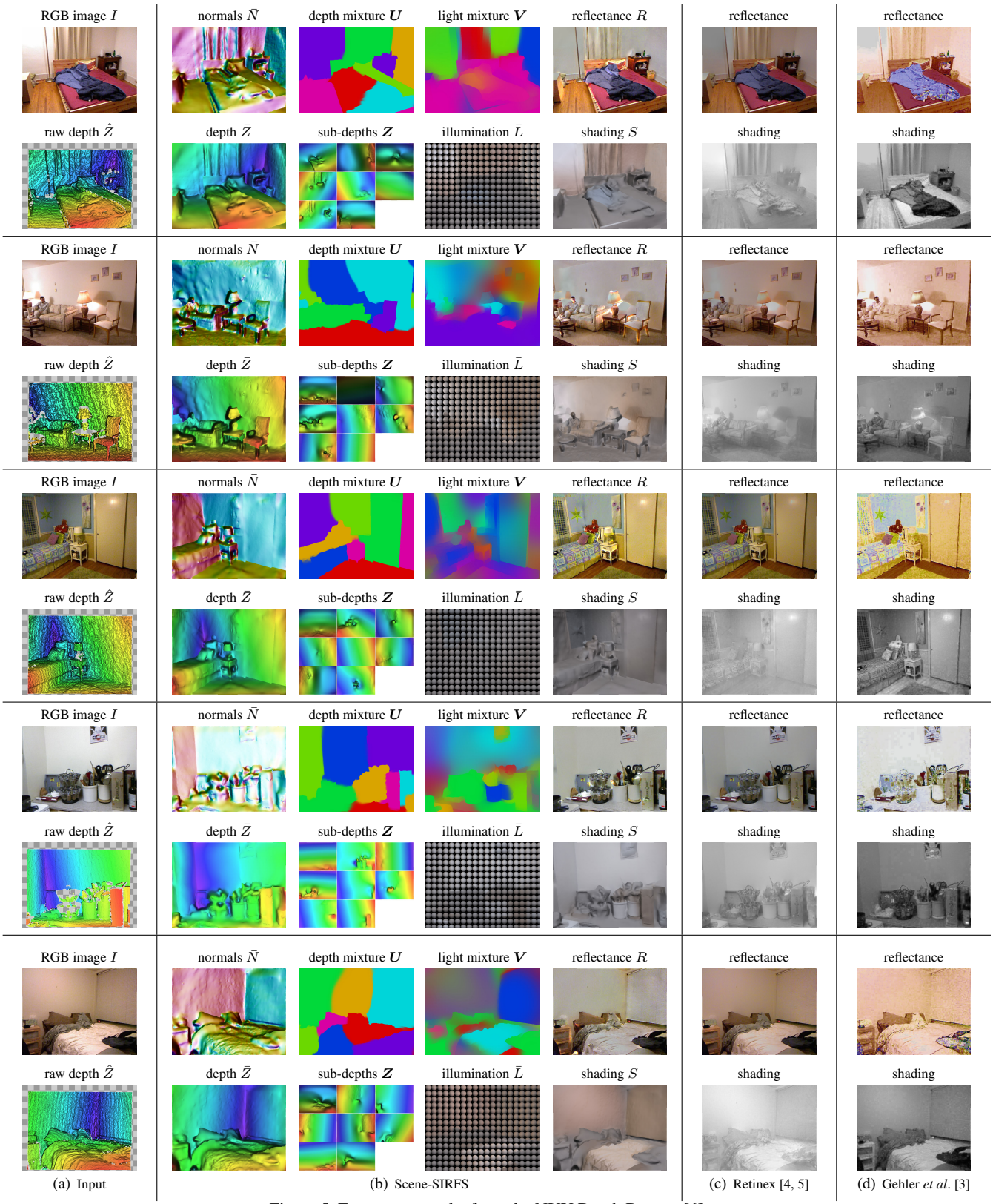


Figure 5. Even more results from the NYU Depth Dataset [6].