Eurographics Symposium on Rendering (2025)
B. Wang and A. Wilkie (Editors)

Supplementary: An evaluation of SVBRDF Prediction from Generative Image Models for Texturing 3D Scenes

A. Gauthier[1], V. Deschaintre[2], A. Lanvin[1], F. Durand[3], A. Bousseau[1], and G. Drettakis[1]

[1]Inria & Université Côte d'Azur, France [2]Adobe Research, UK [3]MIT, USA

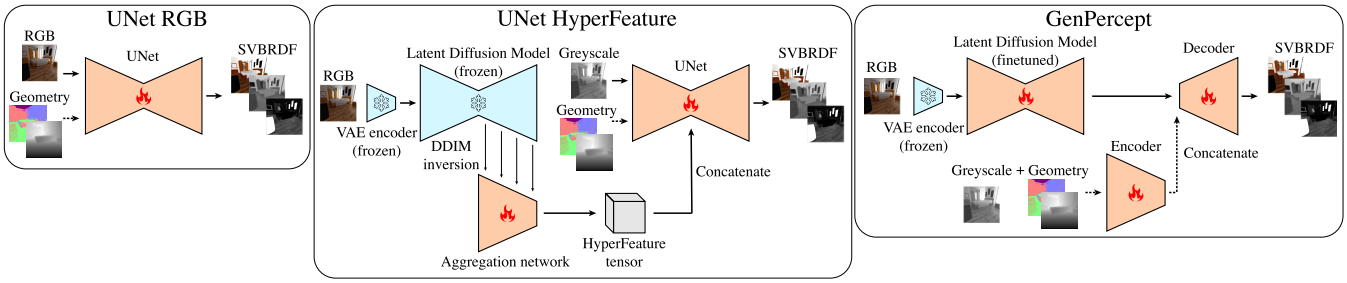

Figure 1: Architectures used in our evaluation. The dotted line box indicates the optional inputs to each architecture (geometry and grayscale input).

In this supplementary material, we provide additional details and results to complement the main article.

1. SVBRDF Predictors architectures

1.1. UNet-RGB

For the UNet-RGB predictor (Fig.1, left), we base our implementation on the diffusers library [vPPL*22]. This codebase provides us with the implementation of the denoising U-Net of Stable Diffusion. For our predictor, we replace the default "AttnDownBlock2D" and "AttnUpBlock2D" by simpler "DownBlock2D" and "UpBlock2D" for a more efficient training. We use a Group Norm value of 1 ("norm_num_groups") as it helps to stabilize training. We leave the rest of the parameters to their default values. When trained solely on single images, the network (UNet-RGB†) is provided with the tonemapped, sRGB corrected image (renormalized in [-1,1]) as input and directly outputs a 5-channel image containing basecolor, roughness, and metallic. When trained with additional geometry cues (depth and normals), the UNet-RGB takes as input a 7-channel image formed by the concatenation of the RGB image, the normalized depth and the normal map.

1.2. UNet-HF

For the UNet-HF predictor (Fig.1, middle), we extract activation features of Stable Diffusion's denoising U-Net during inverse DDIM sampling [SME21], similar to [LDP*23]. The author leverages an aggregation network, which processes upsampled activation features to the size of the biggest feature map in the decoder layers. This results in what they call a 'Hyperfeature tensor'. The UNet-HF predictor shares the same U-Net architecture as the UNet-RGB network but mixes its middle bottleneck layer with the so-called 'Hyperfeature tensor'. To do so, we encode the Hyperfeature tensor using a Conv2D block with TanH activation, inspired by [LDW*24], at half the original middle bottleneck depth resolution, and encode the original middle bottleneck of the U-Net with a similar Conv2D block with TanH activation. Both outputs are concatenated to produce a middle block of the same size as the original middle bottleneck. The aggregation network of [LDP*23] and the prediction UNet are trained end-to-end, to extract the relevant features for SVBRDF prediction. We extract the diffusion features of a given RGB image by inverting it into Stable Diffusion using inverse DDIM sampling. Contrary to the UNet-RGB, we only provide the SVBRDF predictor with a grayscale version of the input image, forcing it to extract colored information from the Hyperfeature tensor rather than the image itself. We train two versions: UNet-HF, trained with input depth and normals, and UNet-HF†, trained only on grayscale images.

Because we use convolutional neural networks, both UNet-RGB, and UNet-HF can predict SVBRDF at any resolution, when provided with the appropriate maps as input (the same resolution as the input RGB). Since the size of the Hyperfeature tensor is fixed to a height and width of size 64*64, we simply interpolate it to the same size as the bottleneck layer of the SVBRDF predictor to output maps at the appropriate resolution.

1.3. GenPercept

For the GenPercept predictor (Fig.1, right), we use the implementation of the original paper [XGL*25] using the customized decoder variant to account for our 5-channels outputs. For the decoder architecture, we simply output 5 channels out of the DPT head for SVBRDF estimation instead of a single one for monocular depth estimation. When trained with additional geometry cues (depth and normals), we first use a fully connected geometry head with ReLU activations to go from 5, to 16, to 32 features. We use a second head for the output of the diffusion UNet, taken from the original [XGL*25] codebase, to which we remove the last Conv2D layer. Finally, we concatenate both 32 features of the geometry head and the former one, to pass them into a two-layered fully connected output head (64, to 32, to 5 features) with a ReLU activation in between and an Identity activation in the end (to stay as close as possible to the original implementation). For training, we use an 'effective_batch_size' of 32 and a 'max_train_batch_size' of 8. Both UNet and decoder learning rates are set to 3e-5. We set 'max_epoch' to 10.000 and the 'lr_scheduler' to 'IterExponential' with parameters total_iter=25000, final_ratio=0.01, and warmup_steps=100.

2. Additional Texturing result

In Fig.2, we illustrate another texturing example on a kitchen scene with a basket, some fruits, cutting boards and pots.

3. Additional Stop-The-Pop metrics

We provide the full Stop-The-Pop metric graphs Fig.4 to 8 for the 5 scenes illustrated Fig. 3, which we used in the main article.

4. Additional SVBRDF Predictions

In Figures 9 to 11, we illustrate more SVBRDF predictions results with all the methods we evaluate on 5 additional examples.

References

[LDP*23] LUO G., DUNLAP L., PARK D. H., HOLYNSKI A., DARRELL T.: Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems* (2023). 1

[LDW*24] LUO G., DARRELL T., WANG O., GOLDMAN D. B., HOLYNSKI A.: Readout guidance: Learning control from diffusion features. 1

[SME21] SONG J., MENG C., ERMON S.: Denoising diffusion implicit models. In *International Conference on Learning Representations* (2021). 1

[vPPL*22] VON PLATEN P., PATIL S., LOZHKOV A., CUENCA P., LAMBERT N., RASUL K., DAVAADORJ M., NAIR D., PAUL S., BERMAN W., XU Y., LIU S., WOLF T.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1

[XGL*25] XU G., GE Y., LIU M., FAN C., XIE K., ZHAO Z., CHEN H., SHEN C.: What matters when repurposing diffusion models for general dense perception tasks? In *Proc. of the IEEE International Conf. on Learning Representations* (2025). 2

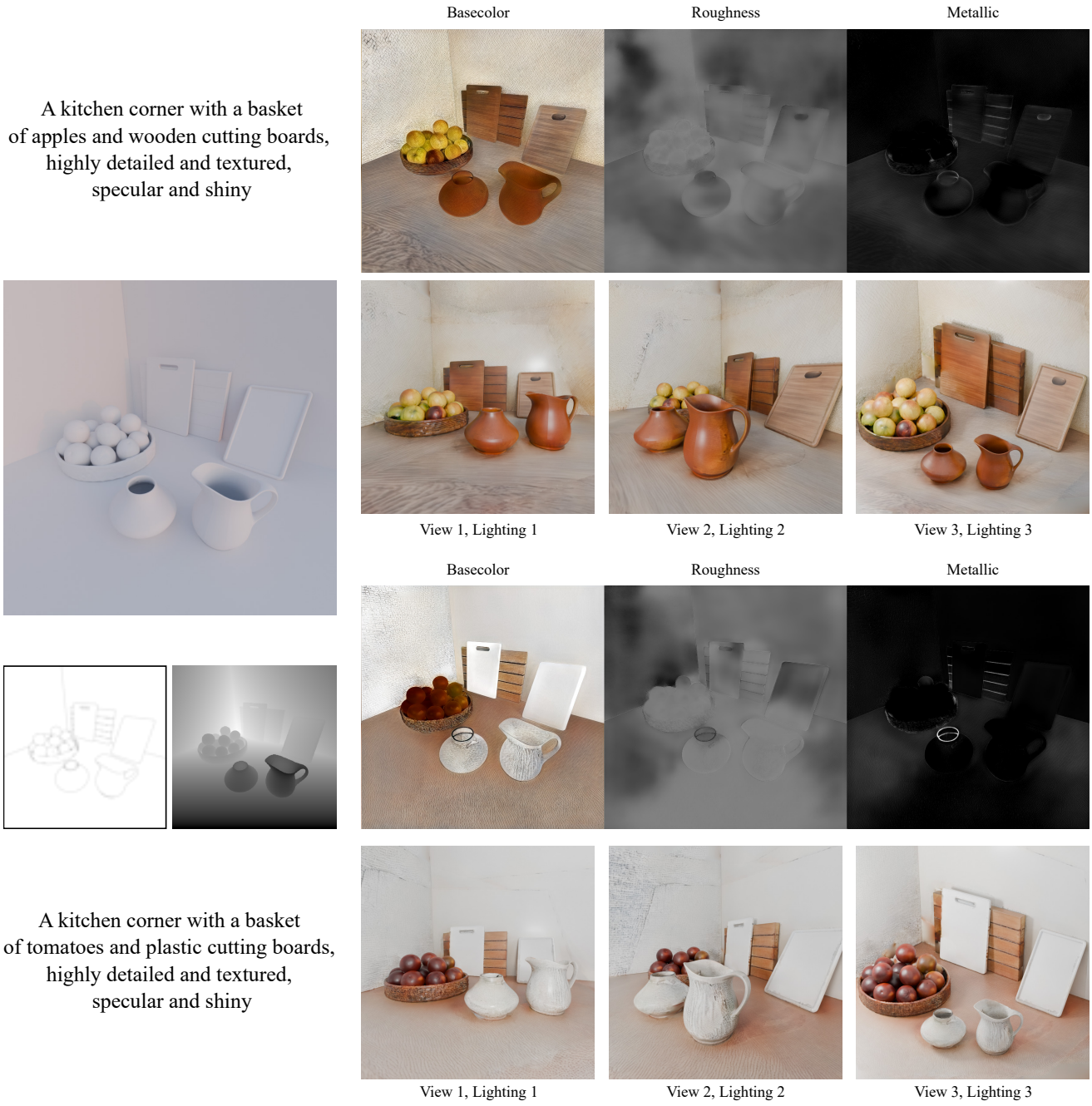

Figure 2: An example of material design for a kitchen scene. On the left, we show the two different prompts used to design the appearance of the scene. For each prompt we show the material maps extracted from the generated image, and below the maps three different viewing and lighting conditions (moving lightsource).

(a) bedroom scene (b) 'kitchen-003' scene (c) 'kitchen-005' scene

(d) 'kitchen-007' scene (e) 'kitchen-010' scene

Figure 3: Scenes used to compute the Stop-The-Pop metrics.

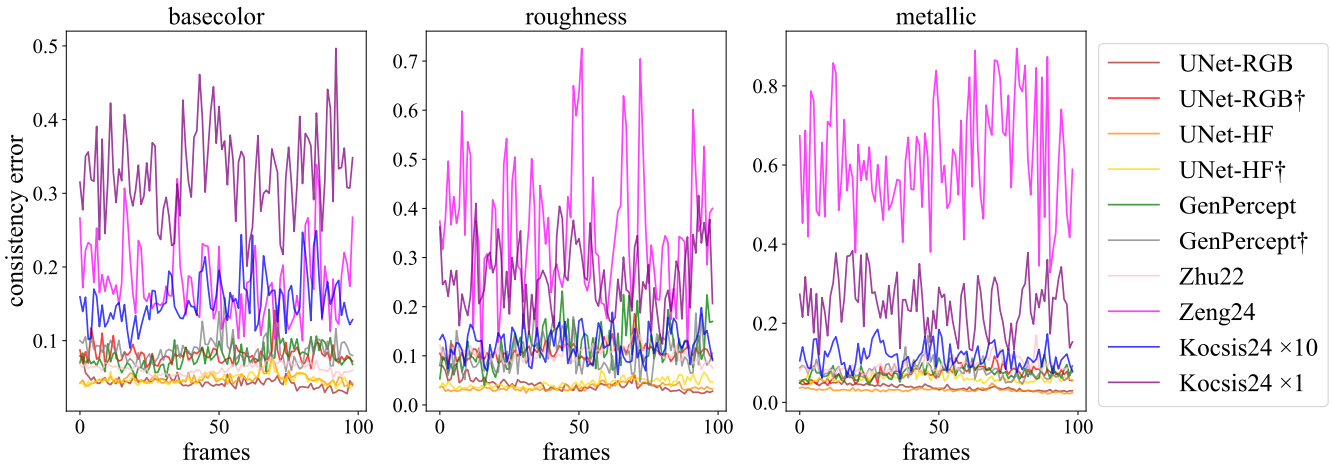

Figure 4: Stop-The-Pop metrics for the bedroom scene

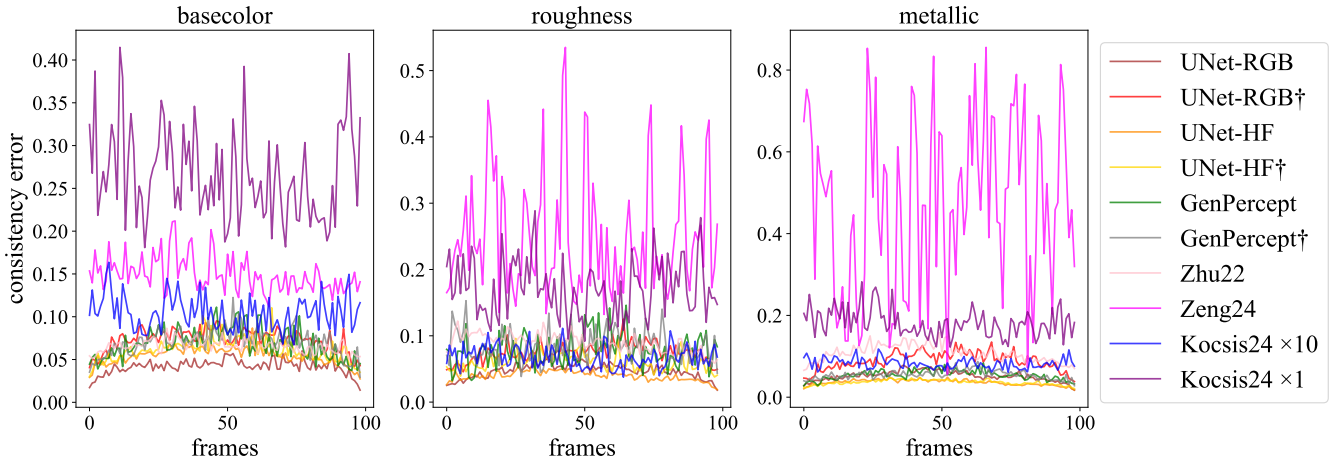

Figure 5: Stop-The-Pop metrics for the 'kitchen-003' scene

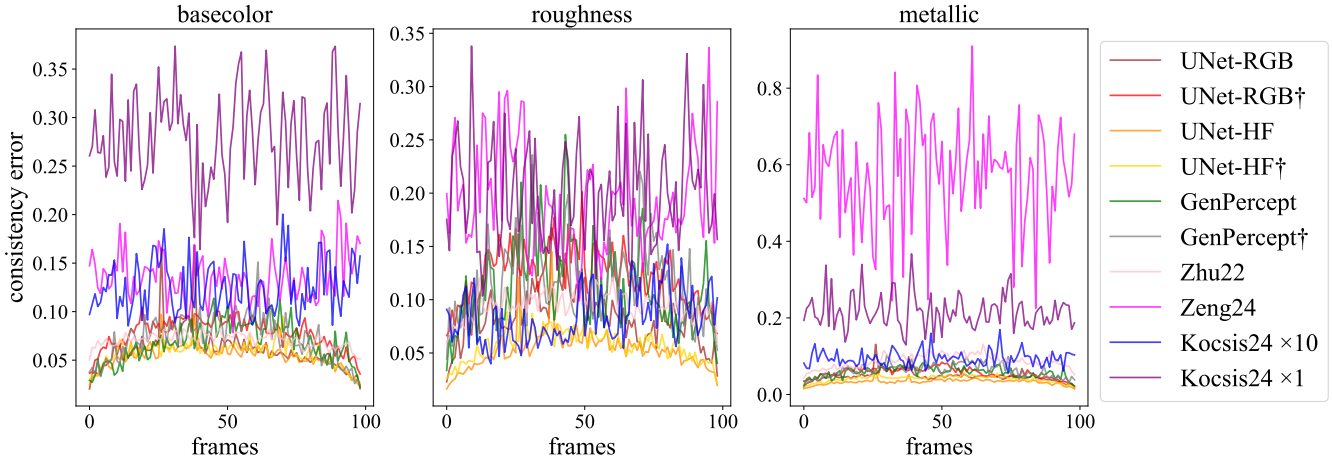

Figure 6: Stop-The-Pop metrics for the ’kitchen-005’ scene

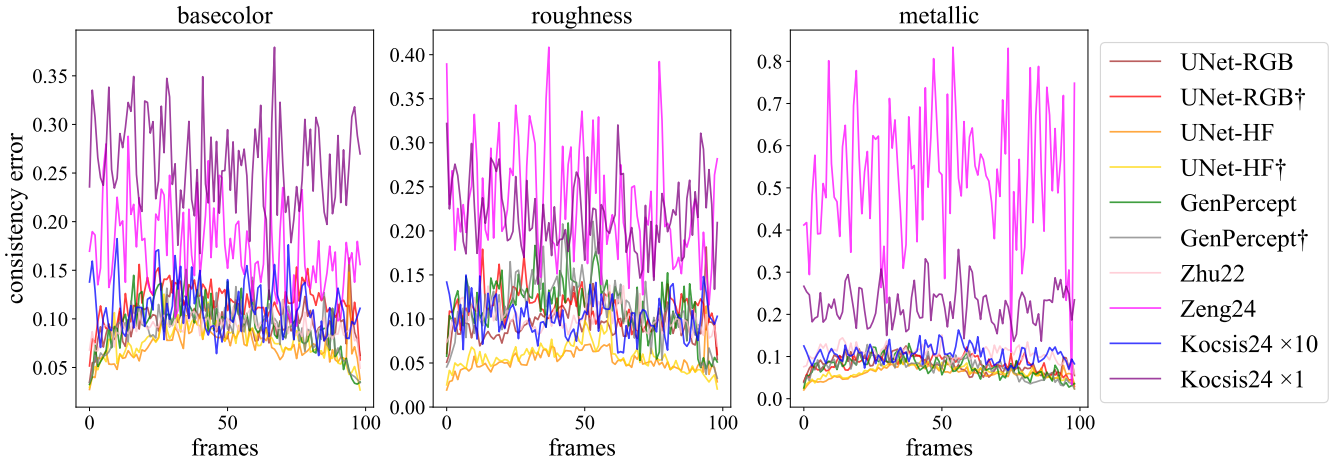

Figure 7: Stop-The-Pop metrics for the ’kitchen-007’ scene

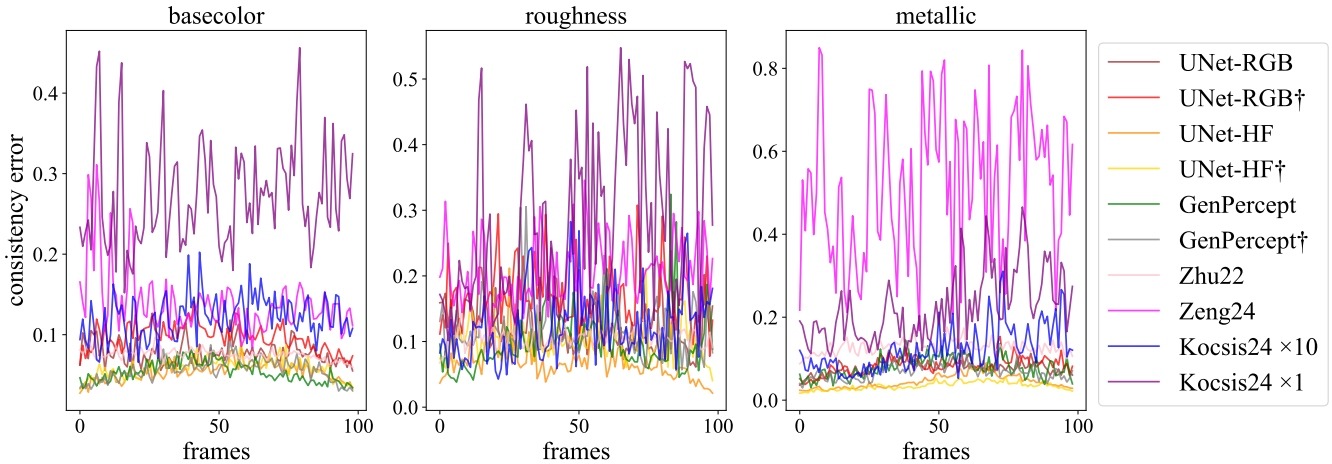

Figure 8: Stop-The-Pop metrics for the ’kitchen-010’ scene

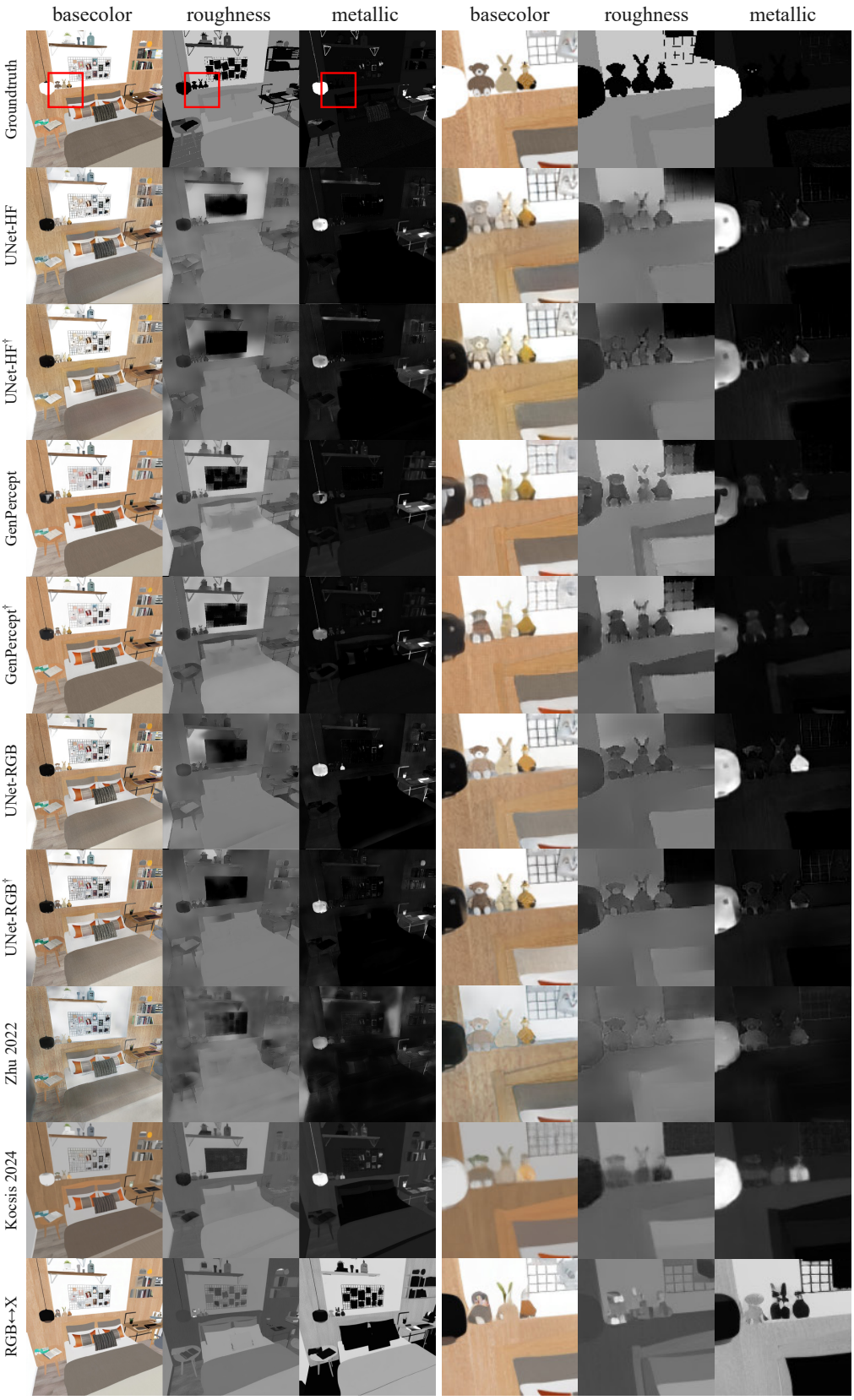

Figure 9: scene: L3D124S21ENDIMODSYQUI5NFSLUF3P3XG888; image 009

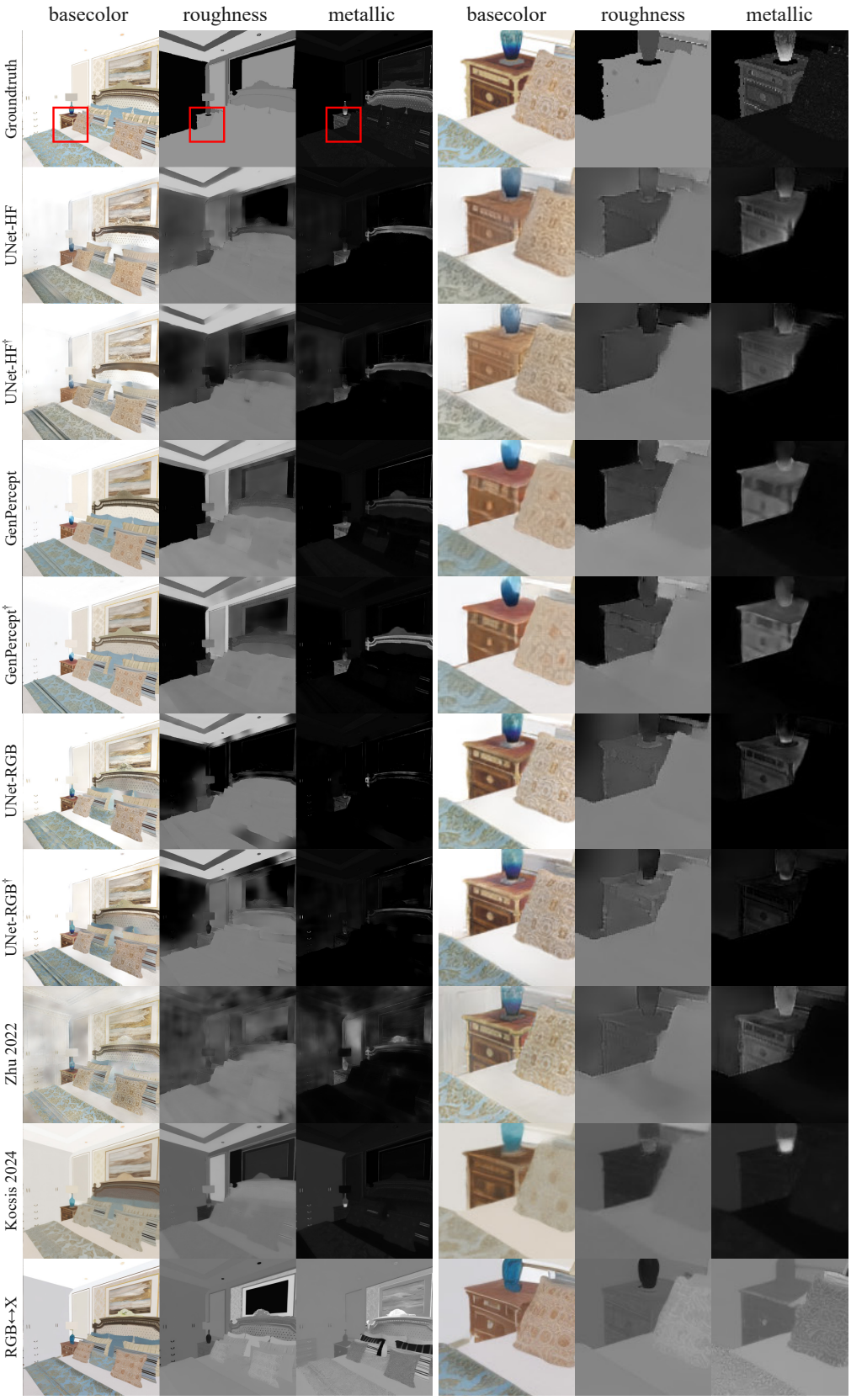

Figure 10: scene: L3D187S8ENDIDQLVHQUI5NYALUF3P3XK888; image 006

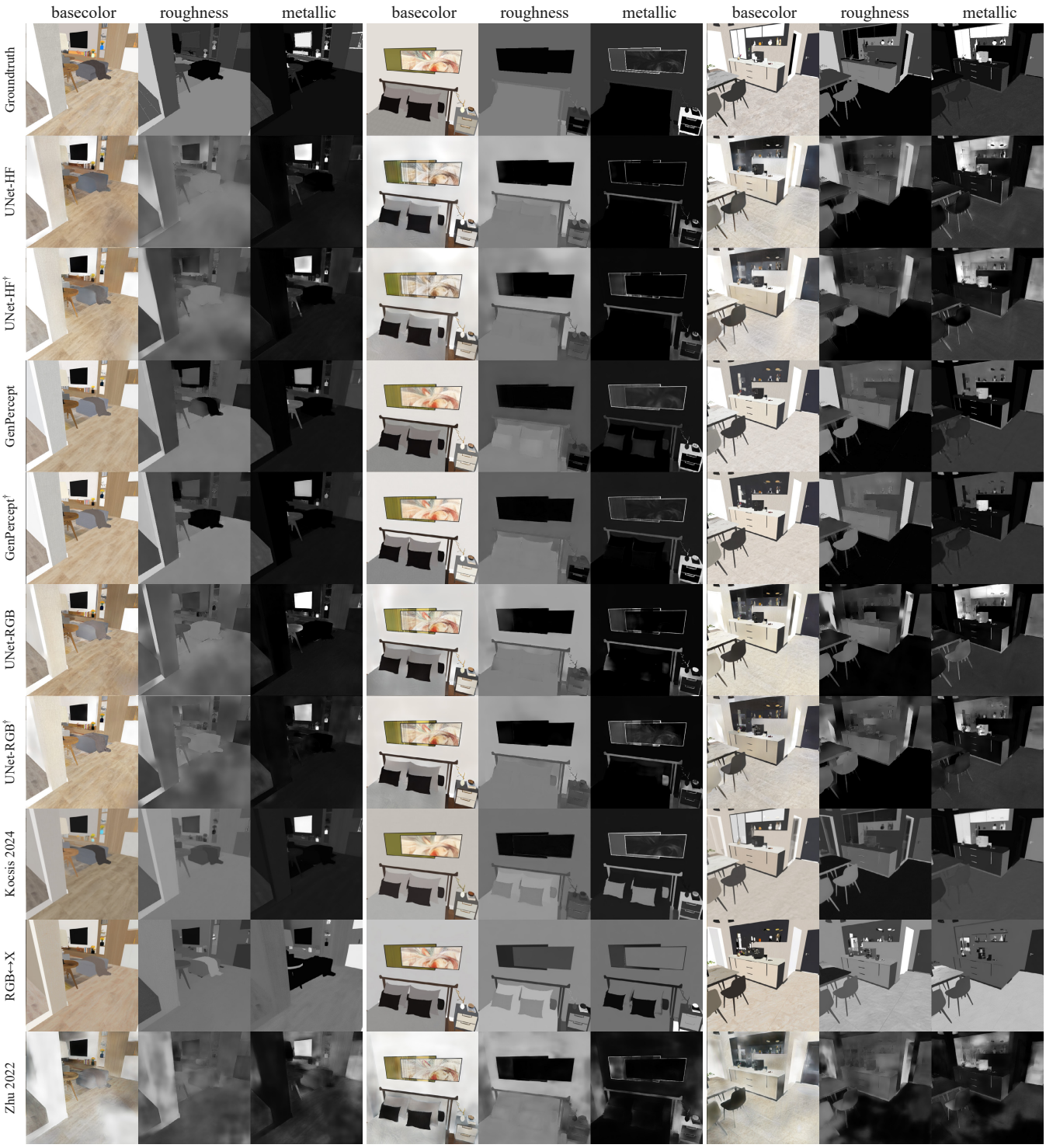

Figure 11: Scene names (from left to right):
scene 1: L3D124S21ENDIDQO5UIUI5NFSLUF3P3XA888 - image 004
scene 2: L3D124S8ENDIDRGRTQUI5L7GLUF3P3WW888 - image 003
scene 3: L3D124S8ENDIMLAOPIUI5NYALUF3P3XS888 - image 004