Material Palette: Extraction of Materials from a Single Image

Supplementary Material

In this supplementary material, we provide experimental details on Material Palette and baselines in Sec. A, and a description of our TexSD texture-generated dataset in Sec. B. Further, we report additional qualitative results in Sec. C and discuss limitations in Sec. D. Sources of our scenes and visuals are reported in Sec. E. We also illustrate the use of our extracted materials for 3D scene editing in the video available on the project page.

A. Experimental details

In this section, additional information is provided to ensure the reproducibility of our method and provide further insights. First, we highlight in Sec. A.1 details about the extraction of textures, how to ensure tileability, and also address issues with color shifts. We report decomposition details in Sec. A.2, and elaborate on the use of segmenters in Sec. A.3. Finally, Sec. A.4 contains evaluation details.

A.1. Tileable texture extraction

Crop selection. We extract crops from $\mathcal{R}$ at different resolutions $c_x = \{2^5, 2^6, 2^7, 2^8\}$ using a moving window with stride $c_x/5$. We retain crops of the largest scale that remain within the confines of $\mathcal{R}$, enabling us to select the optimal scale while accommodating the variable size of $\mathcal{R}$. This way, the maximum amount of information within each region is kept to learn the concept S^*.

Seamless generation. The Stable Diffusion checkpoint we use – version 1.5 [1] – is trained at 512px. Interestingly, we found that the VAE decoder can output at a resolution of 1024px without any noticeable performance drop. To ensure tileability, we use noise unrolling [9] and apply Poisson solving [5] to remove seams on the borders [2].

High-res generation. When generating textures larger than 1024×1024px, we stitch texture patches together. Although noise unrolling ensures the continuity of generated texture patches, seams still remain if patches are directly concatenated (Fig. 12, *left*). Similar to ControlMat [9], we decode overlapping 512×512 patches and apply a weighted average between them to remove visible seams from the resulting stitched texture (*center*). The patches overlap by 8 pixels on all sides ('blending' row). We present denoising times over 10 runs in Tab. 3 at 4 output resolutions (1K up to 8K) on a Tesla V100-16GB. Denoising times increase quadratically w.r.t. resolution size.

[1] https://huggingface.co/runwayml/stable-diffusion-v1-5
[2] https://github.com/bchao1/fast-poisson-image-editing

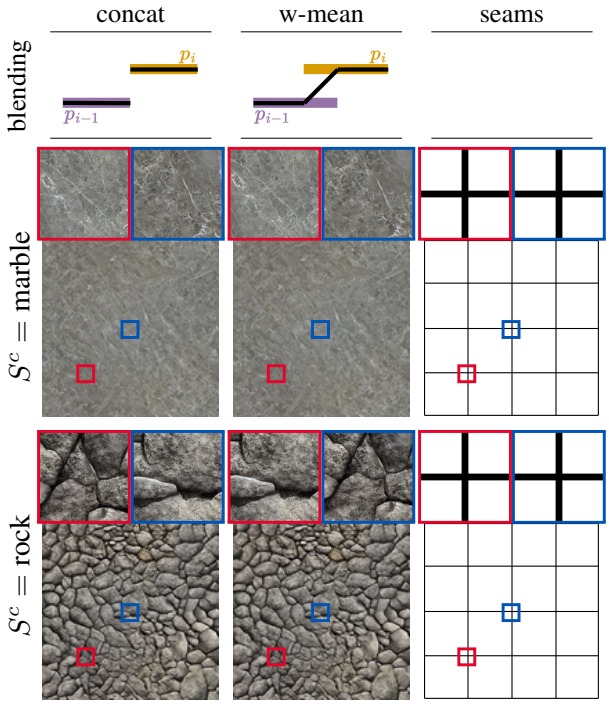

Figure 12. **Patched decoding.** Assembling patches decoded separately by SD. *left*: a direct concatenation of the patches leads to visible seams; *middle*: blending patches with a weighted average; *right* the location of the seams. The blending function shown *above* represents how two adjacent patches p_{i-1} and p_i are stitched together. We zoom in on two areas: red and blue.

Concept learning. To learn S^*, we fine-tune Stable Diffusion (v1.5 weights) [6] and run LoRA [2] Dreambooth [7] using PEFT [3]. We train for 800 steps with a batch size of 1, and a learning rate of $1e{-}4$. Input crops are resized to $c_{in} = 256$ (*cf.* Sec. 4.5) and we observed that augmentations are important to learn S^* without overfitting to the training views. Since textures are not all rotation invariant, we only apply random flip augmentations. Optimization times take no more than 5 minutes per S^* on a single Tesla V100-16GB. In Fig. 13, convergence times are ablated. With 400 steps our strategy yields good results.

Color shift correction. We solve the common color shift of diffusion model outputs [9] by normalizing the generated patch with the per-channel statistics of $\mathcal{R}$. We ablate the importance of this operation in Fig. 14 showing results of a LoRA DreamBooth trained on crops extracted from $\mathcal{I}$ with and without color correction. In practice, we directly renormalize the generated samples with the mean and std

[3] https://github.com/huggingface/peft

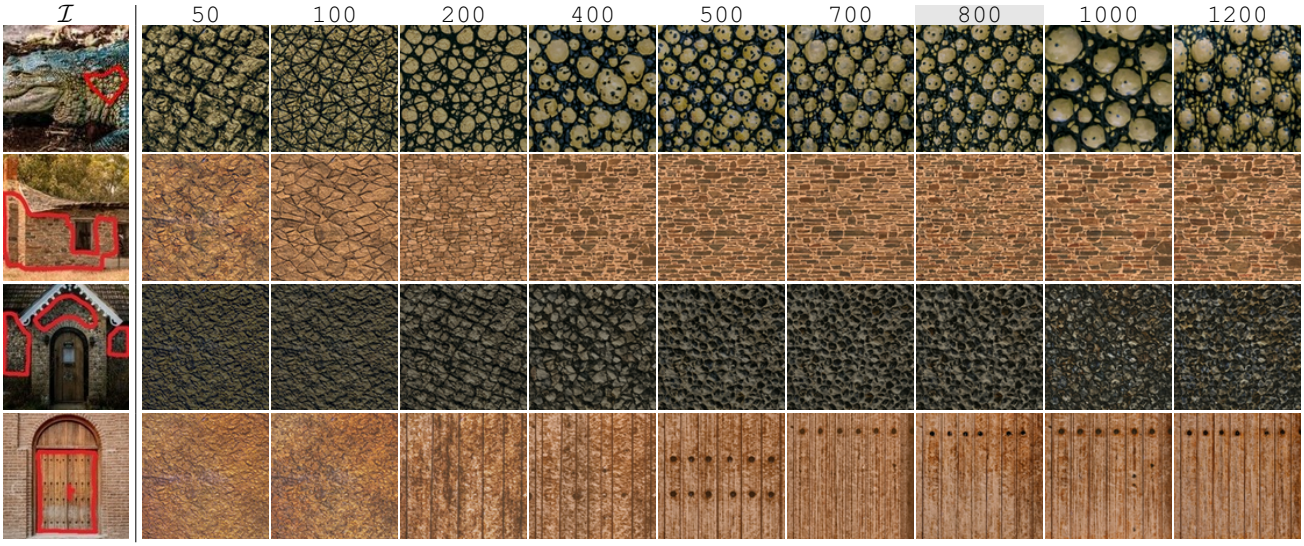

Figure 13. **Convergence.** Generations at different timesteps of the low-rank adaptation [2] of SD when learning S^* with Dreambooth [7]. Although in some cases, fewer timesteps are needed, we generally found that `800` steps work best to learn S^*.

	1K `1,024x1,024`	**2K** `2,048x2,048`	**4K** `4,096x4,096`	**8K** `8,192x8,192`
n	4	16	64	256
t	11s	43s	170s (≈3min)	685s (≈11min)

Table 3. **Denoising times.** All resolutions require 16GB. When dealing with resolutions larger than `1024px`, latent features and textures are processed in batches of 16 patches. For 50 denoising steps, $t \approx n * 2.7\text{s}$, with n the total number of patches.

of $\mathcal{R}$. We highlight that the lack of renormalization on outputs (cols 2-3) leads to a noticeable color shift (darker tone). While applying this operation (cols 4-5) allows us to get a color that matches the input region $\mathcal{R}$.

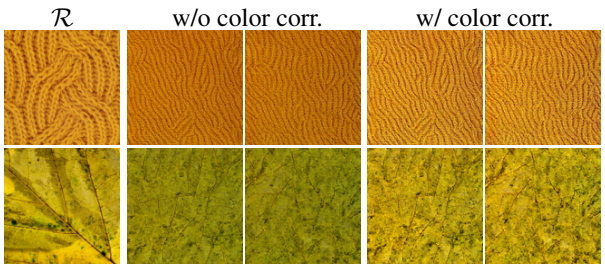

Figure 14. **Color correction.** We show the purpose of our color correction applied to generated textures. We show two generations per concept. The right-most outputs better match the color of $\mathcal{R}$.

A.2. Decomposition training

The decomposition network (*i.e.*, f) described in Sec. 4.1 is illustrated in Fig. 15. We train the source model $f^{\mathcal{S}}$ on $\mathcal{S}$ for 1,000 epochs and finetune on $\mathcal{S} + \mathcal{T}$ for 100 epochs. For augmentations, we apply random 512x512 crops, random horizontal/vertical flips as well as random 90° increment rotations. The decomposition maps A, N, and R are predicted by three separate decoders with $tanh$ activation functions, each outputting spatial 3-, 2-, and 1-channel maps, respectively. The z-axis of N is set to 1 and normalized. Finally A and R are mapped to $[0, 1]$. We apply the log on A and R, as well as all rendered views before computing the $L1$ losses to penalize larger errors. For source training, we use a batch size of 4, while when training on both source and target, we use batches of 2+2. Adam [3] is chosen as optimizer with a learning rate of $1e{-}4$ and λ is set to 0.1 [1].

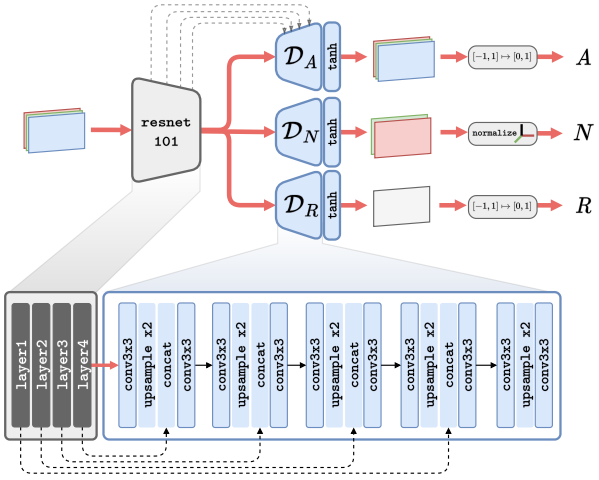

Figure 15. **Network architecture.** We use a Resnet-101 backbone with U-Net skip connections (*left*) to three separate custom decoders (*center*), one for each decomposed map. We apply different normalizations depending on the output type.

A.3. Segmenters

SAM. When using SAM [4], we retrieve the 16 largest masks from the image. We notice that SAM tends to output masks encompassing large portions of the image which are undesired for our task since they are likely to encompass more than one material. Instead, smaller masks are more prone to have a single material displayed. Thus, for overlapping masks with IoU > 0.5 we discard the *largest*.

Materialistic. Materialistic [8] requires an initial query point, so we select a random point within each of our user-defined regions and segment using the pre-trained weights.

User. Regions are defined by a single human annotator.

A.4. Evaluation

In order to use the annotation masks from the ACG dataset, in "Resemblance to SVBRDF dataset." evaluation (Tab. 2, left), we apply a class mapping between ACG and OS [10]. The mapping consists of **14** classes detailed in Tab. 4.

"brick" (**01**) → "brick"	"plastic" (**17**) → "plastic"
"cardboard" (**02**) → "cardboard"	"polished stone" (**18**) → "marble"/"granite"
"carpet" (**03**) → "carpet"	"stone" (**21**) → "paving stone"
"fabric" (**05**) → "fabric"	"tile" (**22**) → "tiles"
"laminate" (**11**) → "wood floor"	"wallpaper" (**23**) → "wallpaper"
"leather" (**12**) → "leather"	"wicker" (**24**) → "wicker"
"paper" (**16**) → "paper"	"wood" (**25**) → "wood"

Table 4. **Class mapping.** For evaluation purposes, we construct a mapping as the intersection of class sets from ACG and OS. Numbers correspond to label ids from OpenSurfaces [10].

Inference heuristics. For visualization *only*, we discard regions for which material images P_{SD} have a LPIPS > 0.65 w.r.t. P_C. This heuristic helps filter incorrect materials.

B. TexSD – Texture generated dataset

For generating our support dataset from Stable Diffusion, named TexSD, we compose an ontology (Tab. 5) of material classes starting from material categories in ACG (*top row*), complemented by classes obtained by providing ChatGPT (*bottom row*) the list of ACG classes and querying it to generate a list of "complementary classes" and "additional materials". We use SD [6] v1.5 and apply noise unrolling, outputting images directly at 1024px (*cf.* Sec. A). At inference, we use the same prompt templates as when conditioning with a S^*, except here a class name S^c is used. For instance, given S^c = pebbles, we can query SD with *"realistic pebbles texture in top view"* to generate a texture of pebbles. We recall the templates `PromptsGen`:

- *"realistic S^c texture in top view"* (■)
- *"high resolution realistic S^c texture in top view"* (■)
- *"top view realistic S^c texture"* (☒)
- *"top view realistic texture of S^c"* (☒)

We show some samples from TexSD in Fig. 16.

AmbientCG	*asphalt, bamboo, bark, bricks, candy, cardboard, carpet, chipboard, clay, concrete, cork, fabric, facade, foam, glazed terracotta, grass, gravel, ground, ice, ivory, lava, leather, marble, moss, paint, painted bricks, painted plaster, painted wood, paper, paving stones, planks, plaster, plastic, porcelain, road, rock, rocks, roofing tiles, shells, snow, sponge, tactile paving, terrazzo, tiles, wallpaper, wicker, wood, wood chips, wood floor, wood siding.*
ChatGPT	*acrylic, bamboo flooring, brocade, burlap, burnout velvet, canvas, carbon fiber, ceramic tiles, checkered, chiffon, cobblestone, concrete blocks, concrete pavers, corduroy, cotton, crocheted, crushed velvet, crystal, damask, denim, dupioni, embossed, felt, fiberboard, fur, gingham, giraffe fur, glass, granite, hammered velvet, hempcrete, herringbone, jacquard, knitted, lace, linen, linoleum, linoleum, mahogany wood, mosaic tiles, oak wood, octagonal paving, onyx, organza, particle board, pine wood, plaid, plywood, quartz top, quartzite top, rammed earth, reptile skin, resin, ribbed, rubber, rustic wood, sand, sandstone, sateen, satin, shantung, silk, silk velvet, slate, snake skin, straw bale, stucco, suede, taffeta, teak wood, terracotta, tiger fur, travertine, tweed, velvet, vinyl, vinyl, woodgrain, wool, woven, zebra fur.*

Table 5. TexSD **ontology.** Classes used to generate the dataset. We complement the classes existing in ACG using ChatGPT.

Figure 16. TexSD**.** The class names S^c are given left-to-right. **1st row:** *bambo*, *candy*, *brick wall*, *tiger fur*; **2nd row:** *zebra fur*, *corduroy*, *veneer*; **3rd row:** *fur*, *fabric*, *giraffe fur*, *granite*, *checkered pattern*; **4th row:** *onyx*, *gravel*, *leather*, *marble*; **5th row:** *marble*, *marble mosaic*, *oak wood*; **6th row:** *pebbles*, *painted wood*, *porcelain*, *quartzite top*, *wood floor*; **7th row:** *terrazzo*, *rustic wood*, *slate*, *rock*; **last row:** *damask*, *resin*, *fur*.

C. Qualitative results

In this section, we present additional qualitative results, for easing visualization of the high-quality extraction of textures and related materials.

Texture extraction from planar surfaces In this experiment, we further highlight possible applications of our texture extraction. Assuming an available top-view material picture, we can sample crops from the whole image (hence $\mathcal{R} = \mathcal{I}$) and generate a tileable texture with our proposed texture extraction pipeline. The resulting images in Fig. 17 are readily usable for any CG application.

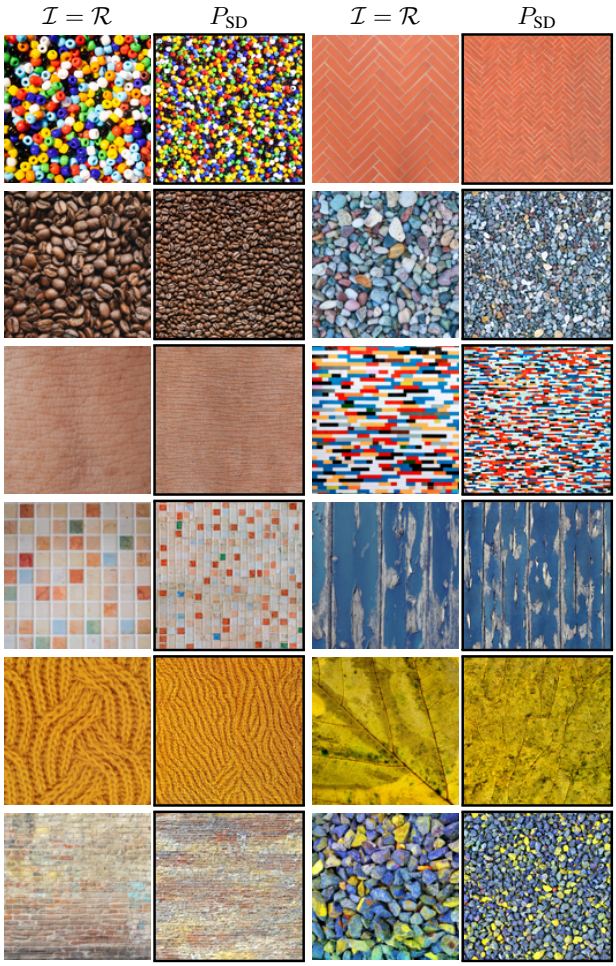

Figure 17. **Texture extraction.** Here, textures are extracted from planar objects that present interesting patterns. Overall S^* has the powerful ability to encapsulate the pattern. This conditioning allows generating tileable high-resolution textures (here 1024^2) which can be a valuable tool for CG artists.

Large textures extraction. In Fig. 18, we present high-resolution examples of extracted textures from real images. This shows further evidence of how Material Palette is able to correctly reproduce the learned texture and extend it to arbitrary resolutions. We highlight the high quality and the absence of visible seams even at 8K resolution.

Comparison with ACG renderings. We compare qualitatively materials in ACG and materials extracted by Material Palette in Figs. 19 and 20. We highlight how common materials emerge in both cases (*e.g.*, bricks, stones, etc.) while our extracted materials exhibit unique ones (*e.g.*, fur, fruits) challenging to extract from controlled scenarios.

Additional visualization. We also extend the visualization present in the main paper. In Fig. 21, we present web-collected images and corresponding material palettes (similar to main paper Fig. 7) but this time showing three samples per material. In Fig. 22, we propose additional results for the end-to-end estimation task (main paper, Fig. 8)

Blender rendering. All 3D renders are made in Blender, using the Principled BSDF shader with the predicted albedo, roughness, and normals map. To improve realism, we use displacement maps inferred from our surface normals using DeepBump [4]. The renderings are done using the rendering engine 'Cycles', with displacement as bumps. Some objects (*e.g.* floor) have a lower displacement scale than others to avoid irregular surfaces that would be unrealistic. Importantly, we do not adjust the shader parameters *per*-material to preserve the true appearance of our extracted materials.

[4] https://github.com/HugoTini/DeepBump

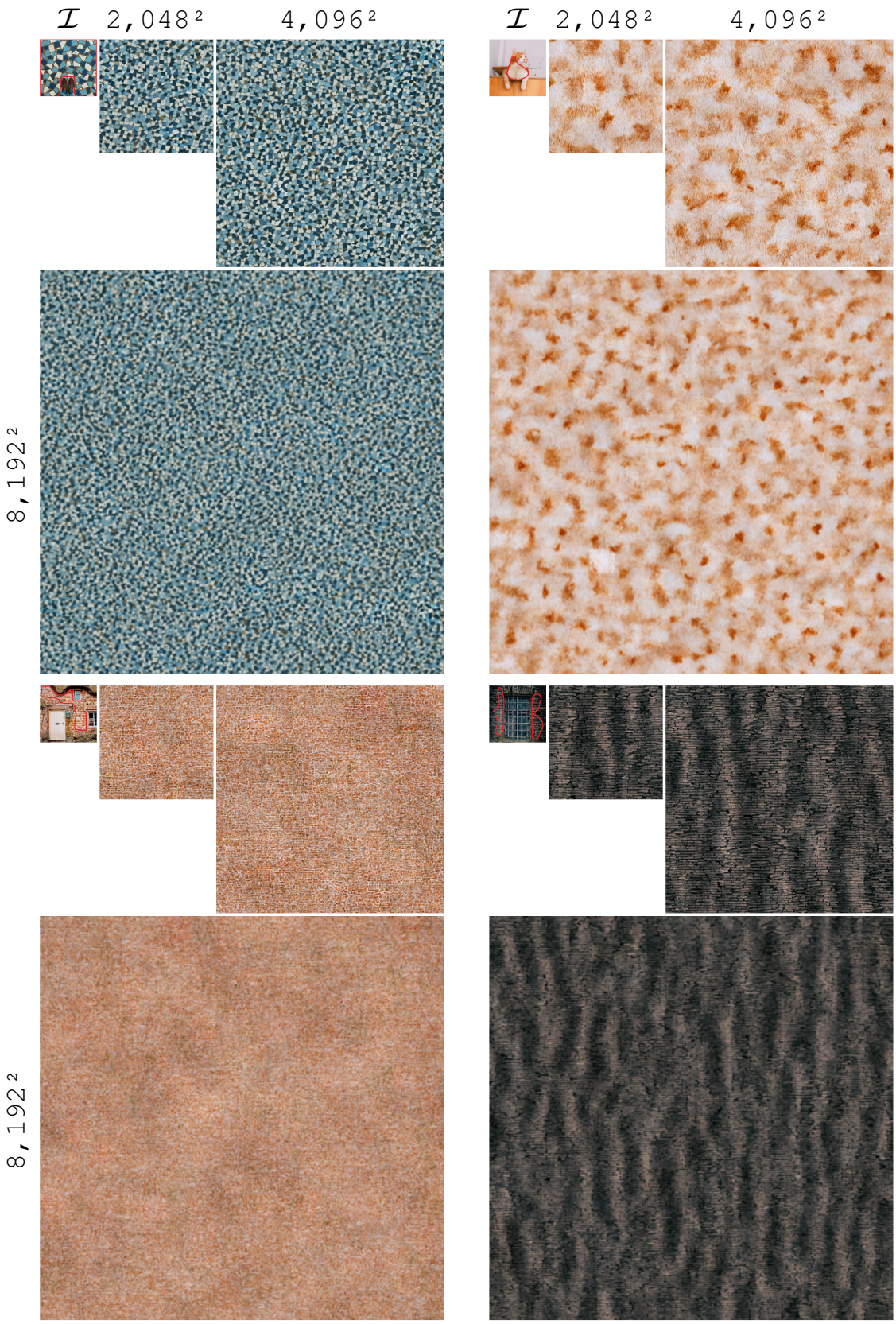

Figure 18. **High resolution outputs.** We provide generations for each image at three resolutions.

Figure 19. **Sphere renderings (a).** Showing spheres of materials from AmbientCG (*left*) and ones extracted from real-world images using our proposed pipeline, Material Palette (*right*). Renderings are done in Blender with a HDRI map.

Figure 20. **Sphere renderings (b).** Showing spheres of materials from AmbientCG (*left*) and ones extracted from real-world images using our proposed pipeline, Material Palette (*right*). Renderings are done in Blender with a HDRI map.

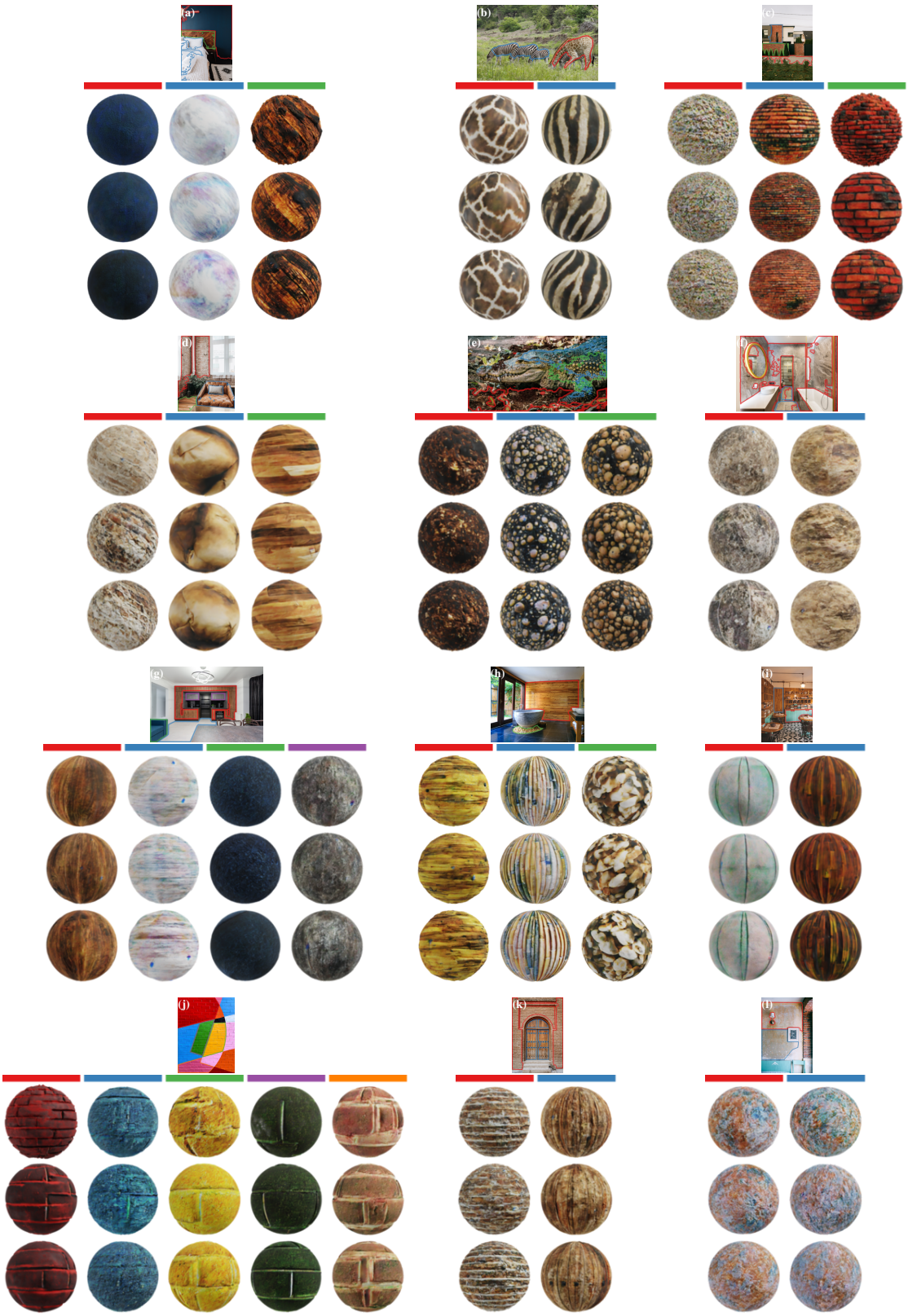

Figure 21. **Material palettes.** We present some results of our method Material Palette on images gathered on the web. Different from the palettes presented in the main paper, here we show three variations per extracted material.

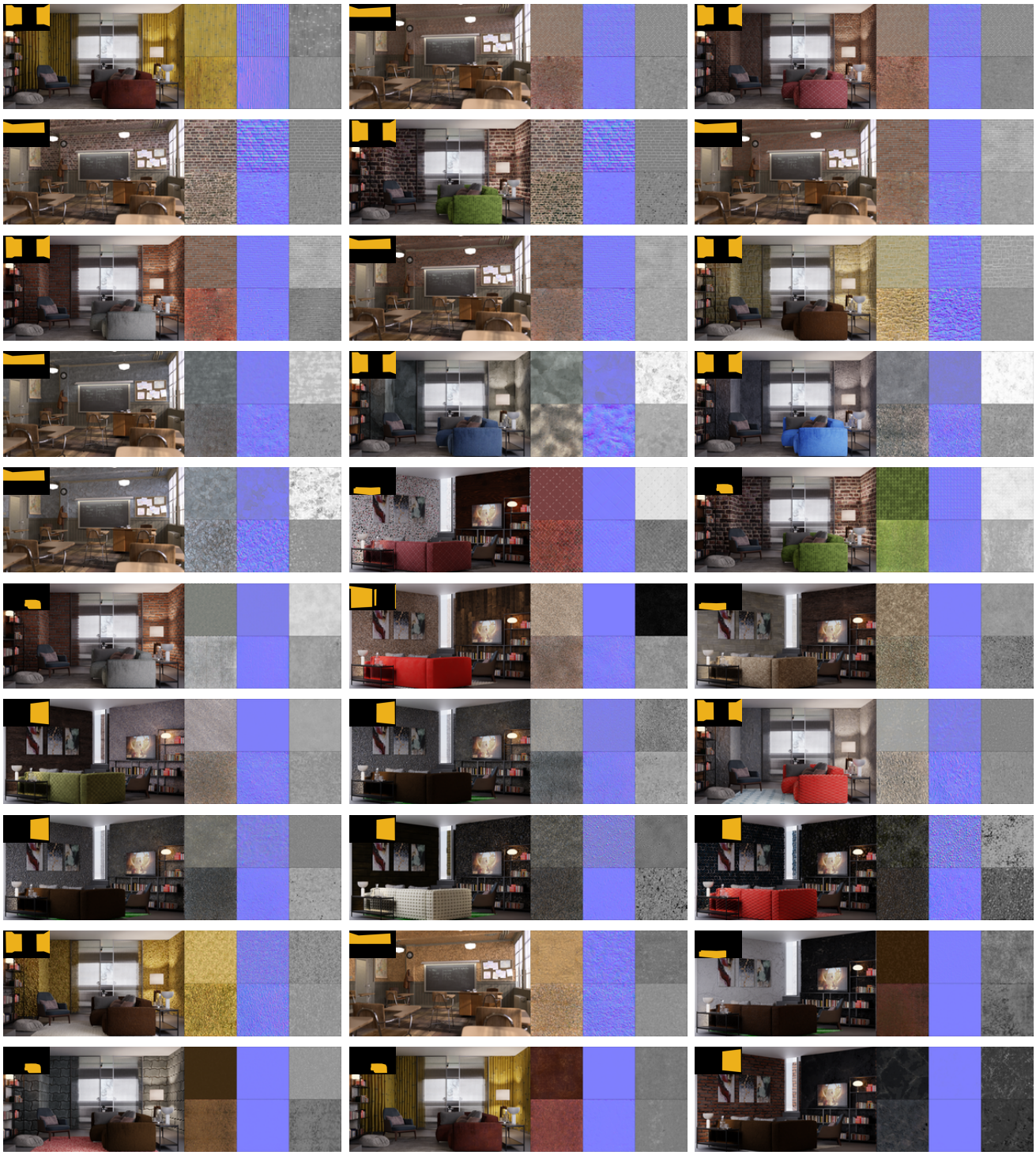

Figure 22. **End to end evaluation.** We edit 3D scenes with ACG materials rendering different scenes and viewpoints (each column, left). The edited regions are shown in the inset. Then, we use Material Palette to extract edited materials. This allows us to compare Material Palette extractions w.r.t. ACG ground truth (right).

D. Limitations

Material Palette lacks contextual information during the material decomposition stage and therefore may perform poorly with strong shading due to lighting or complex geometry. For complex materials, we show failure cases in Fig. 23. Overall, we notice that our pipeline may present incorrect S^* estimation for complex patterns with strong geometrical constraints (rows 1-2). Although our approach corrects geometry and homogenizes light information, errors may still be present with highly tilted surfaces or strong shadows (rows 3-4). Finally, the SD network tends to generate artifacts for highly saturated materials (last two rows). However, for uniform materials as in Fig. 24, we note we are still performing well even in complex lighting.

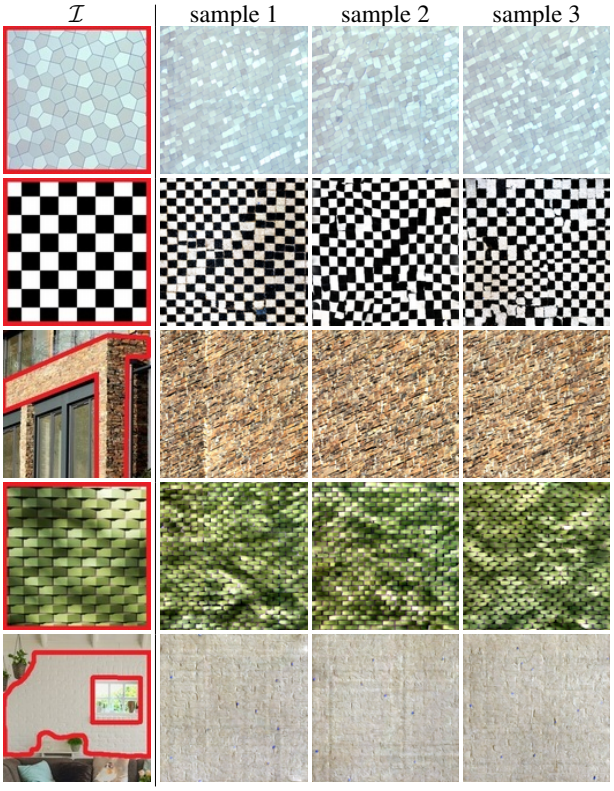

Figure 23. **Failure cases.** Complex patterns are difficult to extract (*first two rows*). Highly tilted surfaces create a bias in the optimization and are not properly rectified (*third row*). Shadows are entangled in the extraction (*fourth row*). We notice some weaknesses when trying to estimate from highly saturated objects (notice the blue spots *last row*).

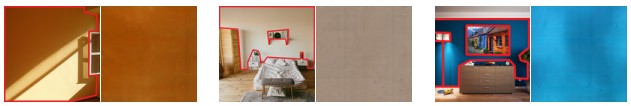

Figure 24. **Complex lighting.** Uniform material extraction in complex lighting

E. Acknowledgements

We provide the list of resources (images and 3D scenes) that may have been used during this project in Tab. 6.

name	author	resource link
private flat	Flavio Della Tommasa	https://blenderartists.org/t/private-flat/1294642
classroom	Christophe Seux	https://download.blender.org/demo/test/classroom.zip
brown wooden table	Skitterphoto	https://www.pexels.com/photo/brown-wooden-center-table-584399/
white couch table	QuarkStudio	https://www.pexels.com/photo/white-padded-couch-in-front-of-black-wooden-center-table-2506986/
wooden kitchen	Max Rahubovskiy	https://www.pexels.com/photo/contemporary-kitchen-and-dining-room-interior-with-shiny-chandelier-7045364/
empty black chair	Saviesa Home	https://www.pexels.com/photo/empty-two-black-chair-2089696/
mosque man	Mihai Vlasceanu	https://www.pexels.com/photo/mosque-door-surrounded-by-decorative-patterns-18538944/
fes door front	Gül Işık	https://www.pexels.com/photo/arabian-front-door-with-arch-and-mosaic-13794309/
door front	Faruk Tokluoğlu	https://www.pexels.com/photo/decorative-patterns-on-entrance-to-mosque-13811595/
innsbruck	Waldemar	https://www.pexels.com/photo/colourful-houses-in-innsbruck-austria-5052366/
four stools	Pixabay	https://www.pexels.com/photo/four-gray-bar-stools-in-front-of-kitchen-countertop-279648/
ivy brick	Boys In Bristol SmokZ	https://www.pexels.com/photo/ivy-growing-on-building-brick-wall-15869273/
fall modern	Tanya Pro	https://unsplash.com/photos/brown-and-white-concrete-building-oPyu636ASPw
brown brick house	pixabay	https://www.pexels.com/photo/brown-wall-paint-house-near-at-garden-209315/
hold interior home	Maria Orlova	https://www.pexels.com/photo/interior-of-cozy-room-in-old-house-4906484/
colored brick	Fatih	https://unsplash.com/photos/pink-and-yellow-wall-paint-kgqu_qs3B78
joint homes	Erol Ahmed	https://unsplash.com/photos/two-white-wooden-doors-with-grills-FTy5VSGIfiQ
cat	Camilo Ospina	https://www.pexels.com/photo/cat-on-furniture-in-room-18597909/
parrot	Dušan veverkolog	https://unsplash.com/photos/blue-and-yellow-parrot-perched-on-tree-during-daytime-2HgyU4YwraQ
crocodile	Philipp Deus	https://www.pexels.com/photo/a-crocodile-in-close-up-photography-3741492/
church	David Besh	https://www.pexels.com/photo/white-and-black-concrete-house-969260/
okapi	Magda Ehlers	https://www.pexels.com/photo/okapi-animal-in-zoo-12788062/
brick modern	Expect Best	https://www.pexels.com/photo/facade-of-modern-building-against-clear-sky-323781/
blue sofa	Max Rahubovskiy	https://www.pexels.com/photo/interior-of-spacious-living-room-with-minimalist-furniture-6492397/
coffee shop	Emre Can Acer	https://www.pexels.com/photo/three-brown-wooden-dinette-sets-2079448/
damaged bricks	Pixabay	https://www.pexels.com/photo/abandoned-ancient-antique-architecture-235986/
modern tub	Max Rahubovski	https://www.pexels.com/photo/bathroom-interior-with-sink-on-counter-near-mirror-and-bath-6444967/
mountain tub	Max Rahubovskiy	https://www.pexels.com/photo/interior-of-bathroom-with-mirror-above-sink-7061423/
merrigum house	wikimedia	https://commons.wikimedia.org/wiki/File:MerrigumHouse3.JPG
modern mansion	Max Rahubovskiy	https://www.pexels.com/photo/contemporary-villa-against-cloudy-blue-sky-7031595/
oldschool kitchen	Polina Kovaleva	https://www.pexels.com/photo/kitchen-room-with-ornamental-plants-5644353/
sofa space	Charlotte May	https://www.pexels.com/photo/sofa-with-cushions-in-cozy-apartment-5825404/
vintage living	Derick McKinney	https://unsplash.com/photos/grey-pillow-on-chair-Tc349sxFa2U
wooden door	Hisham Yahya	https://www.pexels.com/photo/wooden-door-of-a-bricked-wall-2909959/
wooden furniture	PNW Production	https://www.pexels.com/photo/brown-wooden-cabinet-beside-white-wall-8251248/
rundown home	Graham Meyer	https://unsplash.com/photos/brown-brick-house-near-green-trees-during-daytime-nV1phwYx3J0
stone house	Bernard Hermant	https://unsplash.com/photos/brown-and-white-concrete-house-near-green-grass-field-during-daytime-SLW8vO8Erwc
river stone	Pixabay	https://pixabay.com/photos/house-river-running-waterfall-2548478/
cottage	Alex Baber	https://unsplash.com/photos/closed-brown-house-door-tTKgcVWPdLM
ivy bricks	Liv Cashman	https://unsplash.com/photos/blue-wooden-door-on-brown-brick-house-oJAM9h0158c
green home	Binyamin Mellish	https://www.pexels.com/photo/lighted-beige-house-1396132/
wooden door	Tirachard Kumtanom	https://www.pexels.com/photo/photo-of-wooden-door-near-window-887822/
large garage	Curtis Adams	https://www.pexels.com/photo/photo-of-house-3555615/
marble bedroom	Max Rahubovskiy	https://www.pexels.com/photo/bedroom-interior-with-vases-with-branch-near-bed-and-mirror-6480209/
concrete wall	Henry Co.	https://www.pexels.com/photo/brown-metal-staircase-and-gray-painted-wall-1246078/
ancient wall	Pixabay	https://www.pexels.com/photo/ancient-architecture-brick-brick-wall-277544/
brick fireplace	Max Rahubovskiy	https://www.pexels.com/photo/brick-fireplace-in-a-living-room-7746461/
bricks paradise	Charles Parker	https://www.pexels.com/photo/weathered-brick-residential-building-on-city-street-5847374/
home couch	Max Rahubovskiy	https://www.pexels.com/photo/room-with-open-kitchen-near-sofa-6782429/
orange wool	Engin Akyurt	https://www.pexels.com/photo/brown-knitted-textile-1487703/
green tiles	Sutee Pheera	https://www.pexels.com/photo/green-woven-pavement-570047/
chevron bricks	Mitchell Luo	https://www.pexels.com/photo/close-up-shot-of-a-brick-wall-4115538/
yellow leaf	Hilary Halliwell	https://www.pexels.com/photo/close-up-photography-of-dried-leaf-612328/
blue polygons	Damir Mijailovic	https://www.pexels.com/photo/abstract-background-of-wall-with-geometric-ornament-3695238/
yellow facade	Athena	https://www.pexels.com/photo/yellow-and-black-pattern-2254103/
blue tiles	Natã Romualdo	https://www.pexels.com/photo/brown-low-top-sneakers-2904284/
human skin	Karolina Grabowska	https://www.pexels.com/photo/close-up-view-of-human-skin-4046567/
beads	Magda Ehlers	https://www.pexels.com/photo/assorted-color-beads-1331705/
colored stones	Engin Akyurt	https://www.pexels.com/photo/close-up-photo-of-assorted-rocks-3129641/
multicolored tiles	Monstera Production	https://www.pexels.com/photo/abstract-backdrop-of-multicolored-tiled-floor-7794425/
red planks	Magda Ehlers	https://www.pexels.com/photo/red-wooden-surface-960137/
legos	Omar Flores	https://unsplash.com/photos/blue-red-and-white-artwork-lQT_bOWtysE
checkered board	Wikimedia Commons	https://commons.wikimedia.org/wiki/Category:SVG_checkered_patterns
old wood	Suzy Hazelwood	https://www.pexels.com/photo/gray-wall-with-cracked-blue-paint-2096543/
coffee beans	Polina Tankilevitch	https://www.pexels.com/photo/brown-coffee-beans-4109743/
crumbling paint	Laura Tancredi	https://www.pexels.com/photo/shabby-brick-wall-with-crumbling-paint-7078669/
graffiti brick	ShonEjai	https://www.pexels.com/photo/closeup-photo-of-brown-brick-wall-1227511/
minerals	msvr	https://www.pexels.com/photo/gray-and-yellow-gravel-stones-997704/
white paint	Henry Co. Henry Co.	https://www.pexels.com/photo/white-painted-wall-1939485/

Table 6. **Resources.** List of images and 3D scenes used in this project. All properties have open licenses.

References

[1] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM TOG*, 2018. 2

[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 2

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2017. 2

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 3

[5] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries*. SIGGRAPH, 2023. 1

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 3

[7] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023. 1, 2

[8] Prafull Sharma, Julien Philip, Michaël Gharbi, William T. Freeman, Fredo Durand, and Valentin Deschaintre. Materialistic: Selecting similar materials in images. In *ACM TOG*, 2023. 3

[9] Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. Controlmat: A controlled generative approach to material capture. *arXiv*, 2023. 1

[10] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 3