EG UK Computer Graphics & Visual Computing (2024)
A. Slingsby and D. Hunter (Editors)

Semantic UV mapping to improve texture inpainting for 3D scanned indoor scenes

J. Vermandere [1], M. Bassier [1], S. Cuypers [1], M. Vergauwen [1]

[1]KU Leuven, Belgium

Abstract

This work aims to improve texture inpainting following clutter removal in scanned indoor meshes. This is achieved through a new UV mapping pre-processing step that leverages semantic information from indoor scenes to more accurately align the UV islands with the 3D representations of distinct structural elements, such as walls and floors. Semantic UV Mapping enhances traditional UV unwrapping algorithms by incorporating not only geometric features but also visual features derived from the existing texture. This segmentation improves UV mapping and simultaneously simplifies the 3D geometric reconstruction of the scene after the removal of loose objects. Each segmented element can then be reconstructed separately, using the boundary conditions of the adjacent elements. Since this is performed as a pre-processing step, other specialized methods for geometric and texture reconstruction can be employed in the future to further enhance the results.

CCS Concepts

• ***Computing methodologies*** → ***Mesh geometry models; Texturing;***

1. Introduction

Empty 3D indoor environments, captured from real locations, are in high demand in the gaming and Architecture, Engineering, Construction, and Operations (AECO) industries [VBV22]. These environments can be used for a wide variety of applications, such as remodeling, renovations, and interactive simulations. These environments can be captured and processed using different methods. One of the more popular methods involves using a 3D scanner to capture a full 3D point cloud accompanied by panoramic images that add more information. For efficient consumption, these models are converted to meshes, which retain much of the geometric detail while also embedding the textural information of the surfaces [BVGW24]. However, these environments are rarely empty when captured. Loose objects present during the capture process can lead to occlusions, either because they are placed against a permanent element or because they block the view of another part of the room. Current semantic instance segmentation methods can automatically detect these objects [DRB*18], enabling an automated removal process. Removing these objects from the scene reveals occlusions and holes, resulting in an incomplete environment. Therefore, there is a need to complete these missing parts.

Holes and missing regions in meshes can be completed in two steps: first geometrically and then texturally. Geometric hole filling has been a field of much research, leading to very robust tools and algorithms [DRB*18, MCST22, BG14] for filling these holes. Image inpainting has recently gained popularity due to the rise of diffusion models, which dramatically improve inpainting results [LDF*22]. However, there are still some obstacles to using this method on 3D model textures. The visual appearance of a 3D object is created by using a texture map, which projects the faces onto a 2D plane. This projection is called UV projection. The projection process creates a disconnect between the 3D mesh and the UV texture map [VBV23], as the location of a face in 3D space does not necessarily correspond to the same location on the UV plane. This means adjacent faces do not always remain adjacent in 2D.

Current SOTA works approach this problem in different ways. Works like [GSZ*21, SGC*24, WFR23] use a 2D inpainting approach to paint on the camera views and reproject them onto the mesh. While this works well for objects close to walls or distant from the camera, these methods struggle with large occlusions due to the complex room geometry and multiple objects covering the views. Other works [FN22, OMN*19] aim to directly predict the color in 3D space. However, these models are limited in resolution due to the use of vertex colors or texture fields.

The main goal of this work is to improve the UV projection of the scene by leveraging semantic instance segmentation to separate loose parts from the scene, as well as distinct structural elements like walls and floors. Using these masks, the loose objects can be removed from the scene, and the resulting missing geometry can be reconstructed element by element. Furthermore, the segmented structural elements also allow for better UV mapping, ensuring the resulting UV map more closely matches the 3D mesh, minimizing distortion and keeping adjacent faces together. This new UV map

DOI: 10.2312/cgvc.20241221

will improve the texture inpainting process, leading to a more visually appealing mesh.

2. Background and related work

2.1. Texture inpainting

Restoring missing parts of an image has evolved from algorithm-based methods like Gaussian inpainting [GL17] to machine learning-based approaches like Inpaint Anything [YFF*23] and Mask-Based Inpainting [LLZ*22]. These approaches have the advantage of being able to predict the missing pixels based on the surrounding data instead of solely extrapolating a pattern from the image.

The shift towards diffusion-based inpainting has enabled works like No Shadow Left Behind [ZMBKC21] to remove masked objects completely from a picture, including the object's shadows. PanoDR [GSZ*21] and the work that builds on it [SGC*24] take this a step further by training the diffusion model on spherical panoramic images to enable direct object removal on 360° images. Because these models operate purely in 2D, they do not contain any 3D representation of the scene.

Diffusion-based inpainting has also been used to remove objects from a scene in 3D. Clutter Detection and Removal [WFR23] inpaints both RGB and depth images from multiple viewpoints of a single object and reconstructs the 3D mesh in those missing parts. NeRFiller [WHJ*23] uses a similar approach but creates a Neural Radiance Field (NeRF) instead. These view-based models are limited by what is visible to the camera in a single view. Instead of using a camera view of the missing region, Texture Inpainting for Photographic Models [MCT23] uses dynamic UV mapping to ensure the missing region is always centered and surrounded by reference pixels to perform the inpainting, but it is limited to small areas.

The missing color can also be predicted directly on the mesh. STINet [FN22] directly predicts the vertex colors of the missing regions, while Texture Fields [CYF22] creates an implicit neural field to generate the missing regions. However, these methods are limited by the resolution of the geometry and struggle to generate fine details.

2.2. Scene Segmentation

When trying to segment a scene, the different objects need to be detected. Works like Votenet [DHN20] and V detr [SGY*23] use a point-transformer model [WJW*23] to create bounding boxes for each distinct object. While these work well, they only detect objects.

Full scene instance segmentation takes this a step further by labeling every face. Works like Unscene3D [RLD23] can perform class-agnostic segmentation completely unsupervised. Sai3D [YLX*23] also enables CLIP-based embedding to search for specific objects in the scene.

2.3. UV mapping

The biggest obstacle in using 2D inpainting methods on 3D meshes is the lack of a UV map that is both efficient and retains the face-adjacent relationships of objects in a scene. Graphseam [TRC*20] uses a Graph Neural Network (GNN) to automate the UV mapping process while retaining semantic seams, while Flatten Anything [ZHWH24] uses point-wise mappings between the 3D points and UV coordinates. However, these methods are difficult to generalize to a large scene. Nuvo [SGV*23] aims to address this by optimizing the UV layout for the visible parts using a neural field. This largely overcomes the challenges posed by the complex geometry of reconstructed scenes.

3. Methodology

The proposed method as shown in Figure 1 consists of multiple steps: First the input mesh is segmented, and the segmentation masks are used for both element separation and UV seam creation. Second, The loose objects are removed and the segmented structural elements are all completed geometrically. Third, the UV map is unwrapped following the semantic seams. Fourth, the texture is inpainted in the newly created geometry. Finally, the texture is reprojected on the empty mesh.

3.1. Scene segmentation

The first step is segmenting the full scene as seen in Figure 1a. Before the segmentation is performed, due to the limited resolution of the mesh, we cannot guarantee that each face is exclusive to a single object. This is why we first perform a Geometry refinement step [VBV23] to split the faces according to their texture. Both the loose objects and the structural elements are detected using UnScene3D [RLD23] Which uses geometric and colour features to generate pseudo masks, these masks are then refined using a self-trained model. Since the model is optimised for object detection, structural elements like walls can sometimes remain clustered. We also perform a RANSAC plane segmentation [KL18] to refine the walls.

3.2. Geometric Reconstruction

The loose objects, detected in the previous step, are removed from the scene as illustrated in Figure 1b. This results in large holes that need to be reconstructed. Before each segmented structural element is reconstructed one by one, the RANSAC planes, detected in the previous step, are used to determine the intersection edges between the elements. These edges form the boundary conditions for the Delaunay reconstruction [BG14].

3.3. Semantic UV Mapping

After the geometry has been reconstructed, the newly generated faces are given the same label as the original element. The boundaries between the different segmentation labels are marked as UV seams. These seams serve as the basis for the semantic UV mapping as seen in Figure 1c. Since the end goal is to inpaint the missing areas, we optimize the UV map to minimize distortion by using Least Squares Conformal Maps [LPRM02], while still aiming to keep as many faces as possible from the same label joined. This is largely possible due to the flat nature of the structural elements in indoor scenes. Furthermore, when inpainting textures the orientation of the

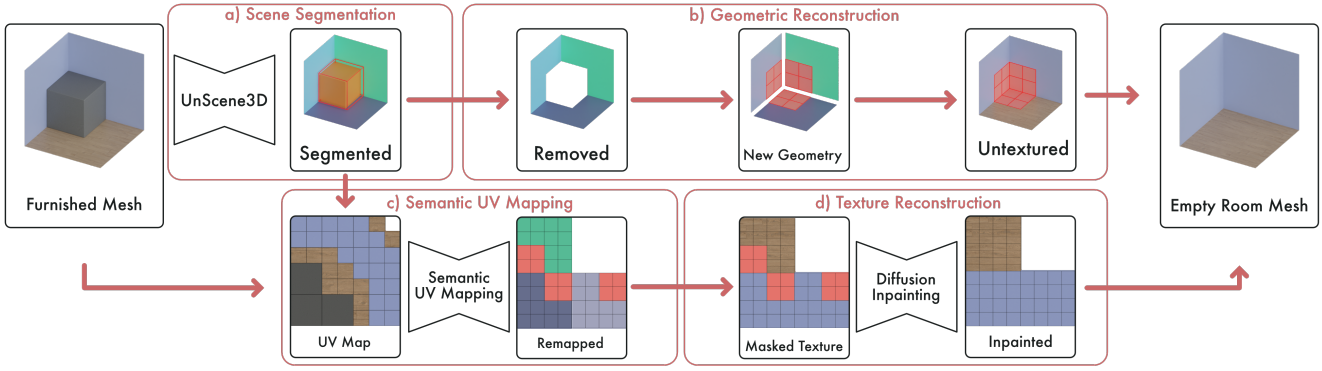

Figure 1: *Overview of the proposed pipeline, starting with a furnished mesh (left), featuring the parallel scene segmentation and geometric reconstruction (top), and the semantic UV mapping and texture reconstruction (bottom) to result in an empty room mesh (right).*

image is also relevant. By introducing a Y/Z up consistency in our unwrapping method, each element is oriented consistently. where vertical elements like walls and beams are always oriented with the up-direction facing up on the image, flat elements like floors and ceilings have their forward direction facing up.

3.4. Texture reconstruction

The final step after the mesh has been semantically UV mapped is painting in the missing regions. This is performed on the 2D texture of each element. The newly generated geometry serves as the inpainting mask, this ensures that only the new parts are altered. The rest of the UV island serves as a reference for the diffusion-based inpainting [LDF*22]. Because each element is inpainted separately, there is no confusion from other adjacent materials possible. After the texture has been inpainted completely, the texture is reprojected on the 3D mesh. Because the UV map was optimized for inpainting, and not for efficiency, the resulting UV maps can be very large. This is why, for a final step we repack the UV layout for optimized space efficiency while keeping the semantic islands intact.

4. Experiments

4.1. Dataset

For our experiments, we used the ScanNet++ [YLND23] and Matterport 3D [CDF*17] Datasets as seen in Figure 2. The ScanNet++ dataset contains 460 high-resolution 3D reconstructions of indoor scenes with dense semantic and instance annotations. The Matterport 3D Dataset is a scanned dataset that consists of 90 fully textured building-scale scenes, including semantic labels of the whole dataset. We focused on single-room scenes with moderately dense furniture, pre-labelled. These labels serve as the baseline for both the loose object removal and the semantic UV mapping.

4.2. Object detection and removal

For our experiments, the instance masks from the Matterport3D dataset are used to separate the mesh as seen in Figure 3. The labels do not always align perfectly with the objects, this is why we removed all the faces inside the bounding box of the objects. this

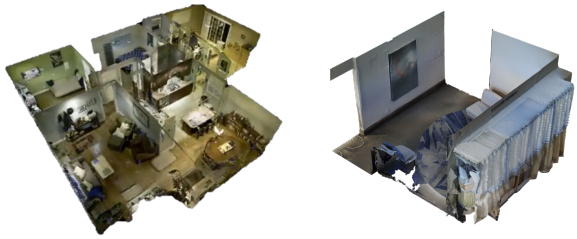

Figure 2: *A scene from the ScanNet++ dataset (left) and Matterport3D dataset (right)*

ensured a clean-cut line. The RANSAC plane segmentation performed well on the walls and floors but had difficulty with more complex geometry.

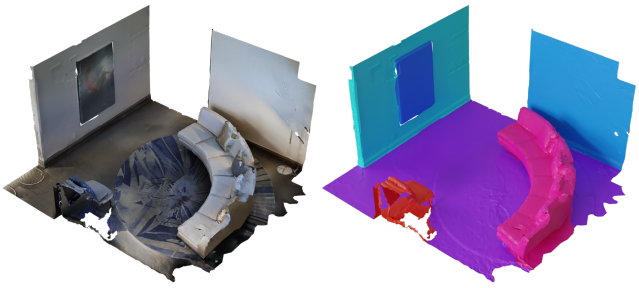

Figure 3: *A room(left) and its segmented labels (right)*

The experiments have shown that the geometry completion performs better when each object is removed sequentially, rather than in parallel. Furthermore, overlapping objects can disrupt the planar detection, so they should be removed together, while this leads to a higher amount of existing data that is removed, the reconstruction results, illustrated in Figure 4, will be better.

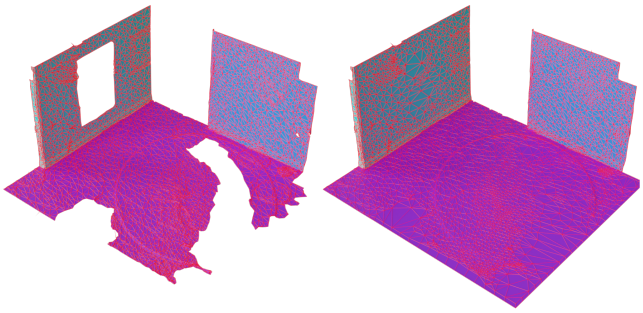

Figure 4: *The object-removed room (left) and the reconstructed room (right)*

4.3. Semantic UV Mapping

The semantic segmentation has created the UV seams at not just geometrically distinct edges, but also texturally (Figure 3). Due to the simple geometry of the structural elements the uv maps can be created without too much distortion using the segmentation seams as seen in Figure 5. We do see, however, that due to the orientation constraint, the UV maps are laid out separately for each object. This means the texture size is larger than the original texture map from the dataset.

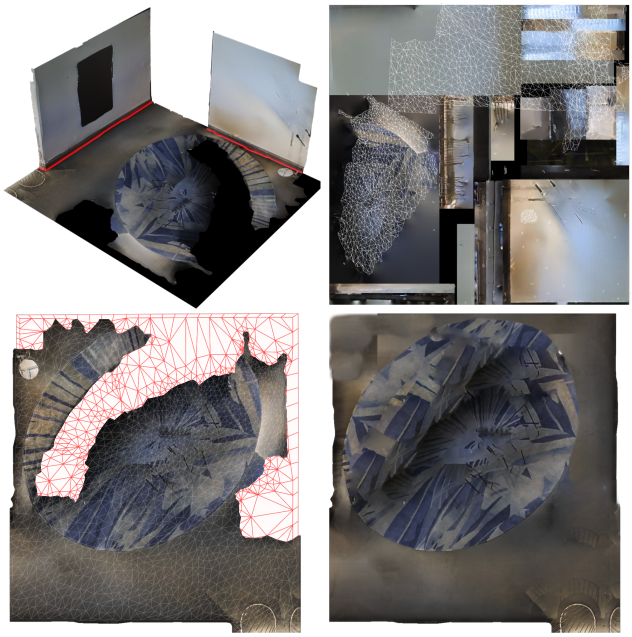

Figure 5: *The new seams in the un-texture-completed scene(top-left), the original texture map (top-right) the new semantic UV layout for the floor (bottom-left) and the inpainted texture (bottom-right)*

4.4. Texture reconstruction

The texture inpainting process is able to use the existing texture as an example which creates mostly indistinguishable textures for the more basic surfaces (5). The faces of the new geometry provide a clear bounding mask, allowing the inpainting to only affect the required area. The final result can be seen in Figure 6

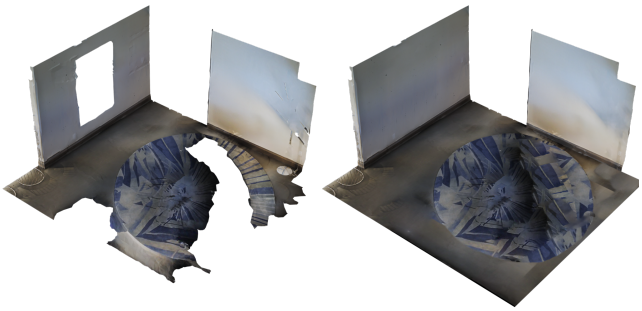

Figure 6: *The object removed textured room (left) final unfurnished room (right)*

5. Discussion

The resulting empty scenes after the loose objects have been removed show believable results. This is helped by the fact that the structural elements in indoor scenes are generally straightforward. By reducing the geometric reconstruction to a planar triangulation, the problem becomes much less complex and manageable. This is however not possible for all types of structural elements. More organic shapes require a more complex Reconstruction like AUTOSDF [MCST22]. The advantage of our pre-processing pipeline is that the methods are interchangeable, while still retaining the advantages of the semantic segmentation. The inpainted textures show very good results for repetitive and basic materials. However, more graphic elements that are not properly segmented can lead to artifacting in the final results. The effectiveness of this method is however difficult to quantify due to the lack of real ground truth data. This is why we visually evaluated each scene, checking for visual consistency and believability.

6. Conclusion

This paper introduced a novel pre-processing step in the object removal pipeline for indoor scanned environments. By semantically labelling the different elements in the scene, both the geometry completion and texture reconstruction can be improved due to clearer boundaries between the different elements. The holes resulting from the removal of the detected loose objects can be better completed element-wise, rather than for the whole scene. Using the predicted intersection lines between the different elements, we can clearly define the boundary conditions for the geometric Reconstruction. The semantic UV mapping also ensures each element is mapped as close as possible to its 3D representation, making the inpainting process much more straightforward. The existing, textured parts of the elements serve as a reference for the newly created geometry.

References

[BG14] BOISSONNAT J. D., GHOSH A.: Manifold reconstruction using tangential delaunay complexes. *Discrete and Computational Geometry 51* (2014), 221–267. doi:10.1007/s00454-013-9557-2. 1, 2

[BVGW24] BASSIER M., VERMANDERE J., GEYTER S. D., WINTER H. D.: Geomapi: Processing close-range sensing data of construction scenes with semantic web technologies. *Automation in Construction 164* (2024), 105454. URL: https://doi.org/10.1016/j.autcon.2024.105454, doi:10.1016/j.autcon.2024.105454. 1

[CDF*17] CHANG A., DAI A., FUNKHOUSER T., HALBER M., NIESSNER M., SAVVA M., SONG S., ZENG A., ZHANG Y.: Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)* (2017). 3

[CYF22] CHEN Z., YIN K., FIDLER S.: Auv-net: Learning aligned uv maps for texture transfer and synthesis, 2022. URL: https://nv-tlabs.github.io/AUV-NET. 2

[DHN20] DING Z., HAN X., NIETHAMMER M.: Votenet +: An improved deep learning label fusion method for multi-atlas segmentation. *Proceedings - International Symposium on Biomedical Imaging 2020-April* (2020), 363–367. doi:10.1109/ISBI45749.2020.9098493. 2

[DRB*18] DAI A., RITCHIE D., BOKELOH M., REED S., STURM J., NIEBNER M.: Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), 4578–4587. doi:10.1109/CVPR.2018.00481. 1

[FN22] FLYNN J. P., NIESSNER M.: Free-form surface texture inpainting using graph neural networks. Visual Computing Group. 1, 2

[GL17] GALERNE B., LECLAIRE A.: Texture inpainting using efficient gaussian conditional simulation. *SIAM Journal on Imaging Sciences 10* (2017), 1446–1474. URL: https://doi.org/10.1137/16M1109047, doi:10.1137/16M1109047. 2

[GSZ*21] GKITSAS V., STERZENTSENKO V., ZIOULIS N., ALBANIS G., ZARPALAS D.: Panodr: Spherical panorama diminished reality for indoor scenes. URL: http://arxiv.org/abs/2106.00446. 1, 2

[KL18] KORMAN S., LITMAN R.: Latent ransac. URL: http://arxiv.org/abs/1802.07045. 2

[LDF*22] LUGMAYR A., DANELLJAN M., FISHER A. R., RADU Y., LUC T., GOOL V.: Repaint: Inpainting using denoising diffusion probabilistic models, 2022. 1, 3

[LLZ*22] LI W., LIN Z., ZHOU K., QI L., WANG Y., JIA J.: Mat: Mask-aware transformer for large hole image inpainting, 2022. URL: https://github.com/fenglinglwb/MAT. 2

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation, 2002. 2

[MCST22] MITTAL P., CHENG Y.-C., SINGH M., TULSIANI S.: Autosdf: Shape priors for 3d completion, reconstruction and generation. URL: http://arxiv.org/abs/2203.09516. 1, 4

[MCT23] MAGGIORDOMO A., CIGNONI P., TARINI M.: Texture inpainting for photogrammetric models. *Computer Graphics Forum 42* (9 2023). URL: https://onlinelibrary.wiley.com/doi/10.1111/cgf.14735, doi:10.1111/cgf.14735. 2

[OMN*19] OECHSLE M., MESCHEDER L., NIEMEYER M., STRAUSS T., GEIGER A.: Texture fields: Learning texture representations in function space. URL: http://arxiv.org/abs/1905.07259. 1

[RLD23] ROZENBERSZKI D., LITANY O., DAI A.: Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. URL: http://arxiv.org/abs/2303.14541. 2

[SGC*24] SLAVCHEVA M., GAUSEBECK D., CHEN K., BUCHHOFER D., SABIK A., MA C., DHILLON S., BRANDT O., DOLHASZ A.: An empty room is all we want: Automatic defurnishing of indoor panoramas. URL: http://arxiv.org/abs/2405.03682. 1, 2

[SGV*23] SRINIVASAN P. P., GARBIN S. J., VERBIN D., BARRON J. T., MILDENHALL B.: Nuvo: Neural uv mapping for unruly 3d representations. URL: http://arxiv.org/abs/2312.05283. 2

[SGY*23] SHEN Y., GENG Z., YUAN Y., LIN Y., LIU Z., WANG C., HU H., ZHENG N., GUO B.: V-detr: Detr with vertex relative position encoding for 3d object detection. URL: http://arxiv.org/abs/2308.04409. 2

[TRC*20] TEIMURY F., ROY B., CASALLAS J. S., MACDONALD D., COATES M.: Graphseam: Supervised graph learning framework for semantic uv mapping. URL: http://arxiv.org/abs/2011.13748. 2

[VBV22] VERMANDERE J., BASSIER M., VERGAUWEN M.: Automatic alignment and completion of point cloud environments using xr data. In *Proceedings - SIGGRAPH 2022 Posters* (2022). doi:10.1145/3532719.3543258. 1

[VBV23] VERMANDERE J., BASSIER M., VERGAUWEN M.: Texture-based separation to refine building meshes. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences X-1/W1-2023* (12 2023), 479–485. doi:10.5194/isprs-annals-x-1-w1-2023-479-2023. 1, 2

[WFR23] WEI F., FUNKHOUSER T., RUSINKIEWICZ S.: Clutter detection and removal in 3d scenes with view-consistent inpainting. URL: http://arxiv.org/abs/2304.03763. 1, 2

[WHJ*23] WEBER E., HOŁYŃSKI A., JAMPANI V., SAXENA S., SNAVELY N., KAR A., KANAZAWA A.: Nerfiller: Completing scenes via generative 3d inpainting. URL: http://arxiv.org/abs/2312.04560. 2

[WJW*23] WU X., JIANG L., WANG P.-S., LIU Z., LIU X., QIAO Y., OUYANG W., HE T., ZHAO H.: Point transformer v3: Simpler, faster, stronger. URL: http://arxiv.org/abs/2312.10035. 2

[YFF*23] YU T., FENG R., FENG R., LIU J., JIN X., ZENG W., CHEN Z.: Inpaint anything: Segment anything meets image inpainting. URL: http://arxiv.org/abs/2304.06790. 2

[YLND23] YESHWANTH C., LIU Y.-C., NIESSNER M., DAI A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. URL: http://arxiv.org/abs/2308.11417. 3

[YLX*23] YIN Y., LIU Y., XIAO Y., COHEN-OR D., HUANG J., CHEN B.: Sai3d: Segment any instance in 3d scenes. URL: http://arxiv.org/abs/2312.11557. 2

[ZHWH24] ZHANG Q., HOU J., WANG W., HE Y.: Flatten anything: Unsupervised neural surface parameterization. URL: http://arxiv.org/abs/2405.14633. 2

[ZMBKC21] ZHANG E., MARTIN-BRUALLA R., KONTKANEN J., CURLESS B.: No shadow left behind: Removing objects and their shadows using approximate lighting and geometry, 2021. URL: https://hdrihaven.com/. 2