

Geometry and Texture Completion of Partially Scanned 3D Objects Through Material Segmentation

Jelle Vermandere^a, Maarten Bassier^b and Maarten Vergauwen^c

KU Leuven, Belgium

{jelle.vermandere, maarten.bassier, maarten.vergauwen}@kuleuven.be

Keywords: Indoor 3D, Mesh Geometry Models, Texturing.

Abstract: This work aims to improve the geometry and texture completion of partially scanned 3D objects in indoor environments through the integration of a novel material prediction step. Completing segmented objects from these environments remains a significant challenge due to high occlusion levels and texture variance. State-of-the-art techniques in this field typically follow a two-step process, addressing geometry completion first, followed by texture completion. Although recent advancements have significantly improved geometry completion, texture completion continues to focus primarily on correcting minor defects or generating textures from scratch. This work highlights key limitations in existing completion techniques, such as the lack of material awareness, inadequate methods for fine detailing, and the limited availability of textured 3D object datasets. To address these gaps, a novel completion pipeline is proposed, enhancing both the geometry and texture completion processes. Experimental results demonstrate that the proposed method produces clearer material boundaries, particularly on scanned objects, and generalizes effectively even with synthetic training data.

1 INTRODUCTION


Dynamic 3D scanned indoor environments are increasingly in demand within the gaming industry and the Architecture, Engineering, Construction, and Operations (AECO) sectors (Vermandere et al., 2022). Like digitally created assets, these environments consist of collections of digital objects that can be interacted with (e.g., by modifying or removing objects) or utilized in computations (e.g., volumetric analysis).


3D scanned environments are typically captured as a whole, not only for efficiency but also for cost-effectiveness. However, when isolating an object from a scene, it is often incomplete due to occlusions and contact with other objects. This missing information presents a significant bottleneck, as the aforementioned applications require complete object data for both geometry and texture (Vermandere et al., 2023). As a result, there is an urgent need for completion methods. Traditionally, geometry and texture completion, typically represented as polygonal meshes, is performed through interpolation. With recent advancements in machine learning and neural


networks, it is now possible to probabilistically predict these outputs (Mittal et al., 2022). Meshes provide a lightweight and scalable representation of 3D scene data, making them effective for scanned environments as they can achieve a similar level of detail to point clouds, while retaining highly detailed texture representations.

Current state-of-the-art (SOTA) techniques typically divide the completion process into two stages, beginning with geometry completion, followed by texture completion. In recent years, significant research has focused on geometry completion (Liu et al., 2023; Lin et al., 2022; Gao et al., 2020; Zhou et al., 2021; Chibane et al., 2020), while research on texture completion has also gained increasing popularity (Cheng et al., 2022; Oechsle et al., 2019; Siddiqui et al., 2022; Lugmayr et al., 2022). However, existing texture completion methods are primarily focused on either restoring minor defects (Maggiordomo et al., 2023) or generating complete textures from scratch (Siddiqui et al., 2022; Richardson et al., 2023).

Texture completion is currently achieved through texture inpainting techniques. However, these methods are often limited to filling very small missing regions (Maggiordomo et al., 2023), which results in

^a  <https://orcid.org/0000-0002-7809-9798>

^b  <https://orcid.org/0000-0001-8526-8847>

^c  <https://orcid.org/0000-0003-3465-9033>

blurry outputs when applied to larger gaps, or they lack fine details due to the limited spatial resolution of 3D inpainting techniques (Chibane et al., 2020). Additionally, deploying trained models on real-world captured data frequently leads to lower-quality results, as many machine learning models are predominantly trained on synthetic data. This discrepancy creates a gap between training data and real-world inputs.

The goal of this work is to improve both the geometry and texture completion on partially scanned meshes. Specifically, the proposed method predicts both the missing polygonal mesh faces and textures of objects segmented from 3D scanned environments. The procedure still treats geometry and texture separately. By splitting the texture completion process into a material prediction and texture inpainting step, as shown in Figure 1, the material boundaries can be more clearly defined. This gives the texture inpainting module a clear inpainting and reference area, which improves the final results. This also allows the usage of synthetic training data for the 3D material prediction step, as no real textures are needed. The realistic textures can then be inpainted on the 2D texture map of the object.

The insertion of the novel material prediction step in the object completion pipeline abstracts the texture inpainting process allowing better real-world results while still using synthetic material datasets.

2 BACKGROUND AND RELATED WORK

In this section, the state-of-the-art of the three main steps in the completion pipeline are discussed.

2.1 Geometry Completion

Mesh completion is a challenging task because meshes have no fixed input size, which is a requirement of machine learning networks. Some models aim to overcome this by using a retrieval-based method (Gao et al., 2023; Siddiqui et al., 2021) which aims to replace the partial data with existing models from a library. However, this limits the generality of the objects that can be completed. This is why most works convert the meshes to either point clouds or Signed Distance Fields (SDFs).

Point-based geometry completion like Point-Voxel Diffusion (Zhou et al., 2021) uses a normalized point cloud as input to predict the final shape through 3D diffusion. On the other hand, IF-Nets (Chibane et al., 2020) use implicit features generated from the

point cloud to predict the missing points. While these models can provide good results, the point cloud sampling can lead to a loss of detail in very dense areas and struggles with large missing parts, something which is very common in incomplete scanned objects.

SDFs are an implicit representation of a 3D shape. They define a function which represents the distance to the boundary of the object from any point in space. They are signed because they also define whether a point is inside or outside the object. An SDF can be voxelised to create a fixed amount of distances. These have become a popular input type due to their clear boundary definition. Models like AutoSDF (Mittal et al., 2022) are able to encode the SDFs and, by highlighting the voxels in the incomplete regions, can predict the missing geometry. SD Fusion (Cheng et al., 2022) builds upon this by allowing multiple input types to guide the generation simultaneously like text prompts or images. Because of the encoding, these can be used to generate a complete shape based on a very small existing part. Models like PatchComplete (Rao et al., 2022) split the SDF into multiple smaller parts to increase its generalizability, while DiffComplete (Chu et al., 2023) uses a diffusion-based approach to allow for a higher flexibility of inputs. While SDFs result in decreased resolution due to the voxelisation, they are much better at retaining the surface definition of the object compared to point clouds. Non-watertight meshes can be difficult to convert to an SDF due to the ambiguity of what is inside and what is not. Jacobson et al. (Jacobson et al., 2013) aims to solve this by generalizing the winding number for arbitrary meshes, however, this method lacks when large parts of the mesh are missing. To handle real-world incomplete scanned objects, Unsigned Distance Fields (UDFs) can be used as a more generalized representation which only defines the absolute distance to the object. These can be generated for arbitrary meshes and are compatible with geometry completion networks like AutoSDF. Therefore, we use the UDF representation of incomplete meshes and complete them using AutoSDF.

2.2 Texture Completion

A challenge when trying to complete the texture of a 3D mesh using its 2D texture map is the inconsistent layout of the UV texture map, where adjacent 3D faces are not always adjacent in 2D. Despite works like (Maggiordomo et al., 2021) trying to improve this, this is still a field of ongoing research. TUVF (Cheng et al., 2023) aims to create a standard UV layout for each object class, making it much easier to generate consistent textures for an ob-

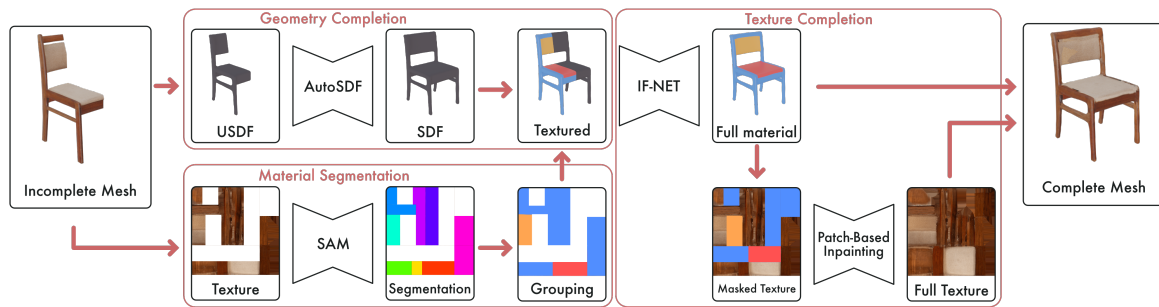


Figure 1: Overview of the object completion pipeline, starting with an incomplete mesh (left), featuring the parallel geometry completion and material segmentation (center-left), followed by the texture completion (center-right) to result in a completed, textured mesh.

ject. This creates a much more predictable inpainting region but severely limits the geometric variation in the objects. Texture Inpainting for Photogrammetric Models (Maggiordomo et al., 2023) aims to overcome this by focusing on smaller patches that are dynamically unwrapped on the texture map. This minimizes distortion and ensures that the surrounding reference area is consistent. For larger areas, works like Image quilting for texture synthesis and transfer (Efros and Freeman, 2001) use an input sample to learn how to inpaint the missing parts leading to very consistent results in distinct materials. When provided with a clear reference area of a single material, these models perform very well. Instead of inpainting directly on the uv map, TEXTure (Richardson et al., 2023) generates 2D textured renders of the object from different viewpoints using diffusion and projects them on the object. Circumventing the need for a clean uv map.

Recent works like Texture Fields (Oechsle et al., 2019) have tried to tackle the texture generation in a similar way compared to the geometry generation, by encoding the texture in 3D space instead of on the 2D plane. This has led to a number of other works like Texturify (Siddiqui et al., 2022) uses texture fields to generate plausible textures for certain object classes. SD Fusion (Cheng et al., 2022) is able to directly colorize the generated geometry by using text prompts. While these models do not take the existing partial textures into account, the introduction of Texture Fields has led to networks like IF-Net Texture (Chibane and Pons-Moll, 2020), which uses partially colored point clouds to predict the remaining, uncolored points. The point-wise structure limits the spatial resolution which can be too low for fine details, leading to unclear boundaries of the different materials. Similar to Image Quilting (Efros and Freeman, 2001), this method can greatly benefit from a clear material boundary and is therefore implemented in our framework as texture completion network.

Point-UV Diffusion (Yu et al., 2023) aims to com-

bine the two texture generation methods by working in a two-step process. First, a coarse 3D point-wise texture is generated. Second, a fine 2D texture map inpainting is performed based on the point colors. TSCom-Net (Karadeniz et al., 2022) uses the same method, but also focuses on texture completion. The completed 3D texture is projected back on a texture map and the coarse color is used to inpaint finer detail directly on the texture. While this improves the detail in the texture, the fuzzy edges of the materials still lead to inconsistent results.

2.3 Material Detection

The different materials of an object can be detected, both in 2D and 3D. In 2D, material differences can be detected on the texture map using image segmentation models like Segment Anything Model (SAM) (Kirillov et al., 2023) which can detect distinct objects or textures by determining large similar areas in the image. Materialistic (Sharma et al., 2023) specializes in detecting similar materials in a single image, however, does not allow for much granularity in the matching process. Other works aim to segment the object based on an image of their 3D appearance using Material-Based Segmentation of Objects (Stets et al., 2019) creating a view-based segmentation. Some reflective materials can be hard to segment because they reflect the environment, resulting in visually confusing images. Multimodal Material Segmentation (Liang et al., 2022) uses multiple camera types like RGB, near-infrared and polarized images to further improve the detection rate of these materials.

In 3D, models like TextureNet (Huang et al., 2018) leverage the color of the feature points in a 3D scene to create more distinct feature vectors, improving the segmentation results. This also allows the model to segment the different materials in a single object. These material segmentation techniques

are underused to aid the texture inpainting process. Therefore, we aim to use a similar technique to segment the objects on a sub-material level.

3 METHODOLOGY

The presented method (Figure 1) follows the SOTA approach to separate the geometry and texture completion. First, the missing geometry is predicted from the geometric inputs by utilizing implicit shape representations (Mittal et al., 2022). In addition, the mesh textures are analyzed, and material information for the meshes is computed based on image segmentation. Second, these results are integrated, and the missing materials are predicted using the texture generation network, IF-Net (Chibane and Pons-Moll, 2020). In the final step, a detailed inpainting of the missing regions is conducted, utilizing both shape and material information to complete the mesh representation. This process results in a comprehensive prediction of the object’s shape and appearance, that can be rendered in photorealistic detail.

3.1 Geometry Completion

The first step in the geometry prediction is the pre-processing of the mesh geometry to a suitable implicit shape representation i.e. a continuous volumetric field. In the literature, SDFs are typically used as it can be easily discretised into a voxel raster with a fixed number of distances, which is compatible with CNN architectures (Mittal et al., 2022). However, as explained in the related work, conventional SDF assume the shape to be watertight, which is not the case for our geometry prediction. Instead, we employ a UDF to voxelise the mesh geometries. Concretely, we employ a dual octree graph as proposed by (Wang et al., 2022) with 128^3 resolution to represent the geometry (Figure 2). Then, we use the open edges of the incomplete mesh surface to indicate the voxels for which a prediction must be computed. As shape completion networks currently only operate on geometries that are positioned symmetrically and centered, we also perform a grounding and symmetrisation step to optimize the objects’ position based on (Sipiran et al., 2014).

Next, the UDF is fed to a shape prediction model that samples vertices in the highlighted voxels. Specifically, we adjust the VQ-VAE autoregressive model proposed in AutoSDF (Mittal et al., 2022) to predict the distribution over the latent representation of 3D shapes and solve it for shape completion (Eq. 1). The voxel selection process has been further



Figure 2: The incomplete mesh (left) and the meshed UDF (right).

refined, to allow for a more granular selection. This allows us to better define the correct parts of the partially scanned object. This can be formulated as the conditional probability optimisation of k number of possible solutions of the 3D shape \mathbf{X} given the partially observed shape \mathbf{X}_p , which are expressed as a set of latent variables $\mathbf{O} = \{z_{g_1}, z_{g_2}, \dots, z_{g_k}\}$ that are factorized to model the distribution over the latent variables \mathbf{Z} (see VQ-VAE and AutoSDF for more details).

$$P(\mathbf{X}|\mathbf{X}_p) \approx p(\mathbf{Z}|\mathbf{O}) = \prod_{j>k} p_{\theta}(z_{g_j}|z_{g_{<j}}, \mathbf{O}) \quad (1)$$

The network returns a number of possible solutions. The best option is selected based on the closest fitting geometry to the original incomplete edges based on the Euclidean distance of the vertices. Note that due to the encoding, the SDF representation was compressed and thus the overlap between the original edges and the sampled edges can be evaluated. To convert the object back into a mesh, we employ marching cubes (Lorensen and Cline, 1998). The result is a watertight polygonal mesh geometry with a topological correct fit between the original and the predicted geometries (Figure 3).

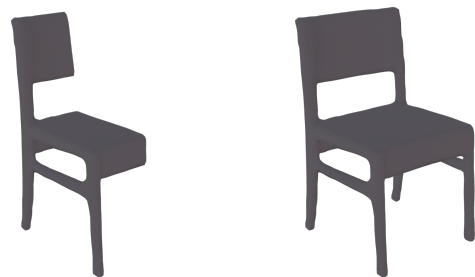


Figure 3: Meshed representations of the incomplete UDF (left) and the completed SDF (right).

3.2 Material Segmentation

Similar to the geometry completion, the different materials of the objects are identified to produce the inputs for the final texture prediction. Specifically, we

compute indices for each distinct material in the object. First, we segment the different texture regions from the texture images of the objects. However, UV maps generated from scanned objects are not ideal for this purpose as these are typically optimised to minimize the texture footprint and maximise each triangle separately. As a result, there is no topological relationship between the adjacent pixels in the texture image compared to the 3D geometry. To counteract this, we re-unwrap each object’s texture to preserve this topology while keeping connected parts together (Figure 4). Building on previous works (Vermandere et al., 2024), this is done by performing a part-wise semantic segmentation (Sun et al., 2022), which splits the object into smaller geometrically more basic parts using 3D semantic instance segmentation. Each part is then unwrapped using Blender’s unwrapping API (Flavell, 2010) with the Angle Based Flattening (ABF) (Chen et al., 2007) algorithm.

The resulting unwrapped texture images are then processed by an image segmentation network. Specifically, we transfer the zero-shot segmentation of the Segment Anything Model (SAM) (Kirillov et al., 2023) to our dataset. SAM is a powerful encoder-decoder network trained on over 1.1 billion masks and shows promising results for zero-shot generalization. The result is set of patches containing a large number of disjoint instances of the different materials.

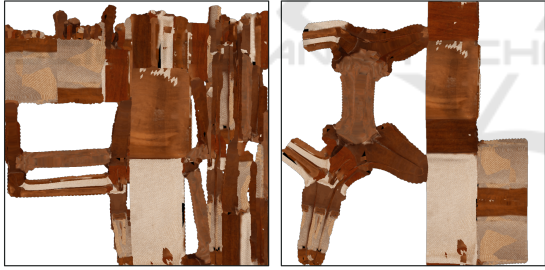


Figure 4: Overview texture preprocessing: (left) The original UV layout and (right) the re-unwrapped UV layout optimised for geometric topology.

Second, these patches P_{set} are grouped per distinct material set S_{set} . To this end, the cosine similarity is evaluated between the image feature vectors f_{P_i} of each patch, which are derived from the EfficientNet (Tan and Le, 2019) network, given a matching threshold t_c . The unique set $\bigcup_{i=1}^n S_i$ of the grouped patches are then used to assign a unique material index to each S_i (Figure 5) as shown in Eq. 2.

$$S_{set} = \bigcup_{i=1}^n \left\{ S_i = \{P_i, P_j\} \mid \forall P_i, P_j \in P_{set} : \frac{f_{P_i} \cdot f_{P_j}}{\|f_{P_i}\| \|f_{P_j}\|} \geq t_c \right\} \quad (2)$$

Next, the material indices are assigned to the partial mesh. However, because the 2D boundaries of the material patches do not necessarily align with the 3D mesh edges, an additional mesh refinement step is performed. Based on previous work (Vermandere et al., 2023), the boundaries S_{set} between texture materials are baked as new edges in the mesh, and duplicate the involved vertices, so that each face shares the same material in its three vertices. This ensures that each material can be completely isolated in 3D with only a face selection.

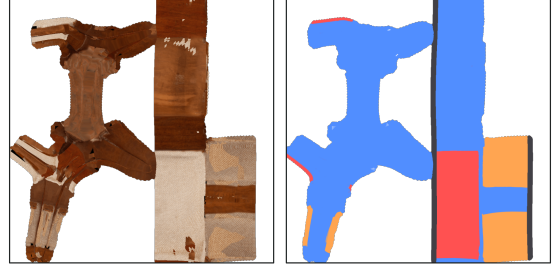


Figure 5: Overview of the texture segmentation using the Segment Anything Model (SAM) and subsequent clustering of the different patches through cosine similarity.

3.3 Texture Completion

For the texture completion, we again employ an implicit representation that can be trained and decoded to predict color information of the missing parts. As we want the texture prediction to be shape sensitive, we retain the spatial encoding of the 3D geometry and expand it with additional color channels. Specifically, our work expands upon IF-Net (Chibane and Pons-Moll, 2020) which extracts a learnable multi-scale tensor of deep features from a spatial encoding of both shape and appearance. Concretely, we first assign the generic segmented material index labels to the completed mesh geometry. Each segment is given a unique color based on its index. Each index is encoded as a combination in three binary channels. Enabling the network to generate up to 8 materials per object. This ensures maximum separation between different materials to minimize potential confusion in the network.

Second, the partially textured mesh is sampled as a point cloud due to IF-Nets point-based encoding. The same voxel grid is employed as during the shape geometry completion. To generate the deep features grids F_k , it subsequently convolutes the point cloud with learned 3D convolutions while decreasing the resolution. These features are then passed to the decoder $f(\cdot)$, which predicts the point and material index values at the grid intervals (Eq. 3) (Chibane and

Pons-Moll, 2020).

$$f(F_k) : F_1 \times \dots \times F_n \rightarrow [0, 1] \quad (3)$$

Given the material indices, the final step is to compute the detailed textures for the complete mesh. To this end, we leverage patch-based inpainting (Efros and Freeman, 2001). Because it only uses the surrounding image for reference, the results can be more faithful to the original data compared to more recent generative approaches as it does not suffer from hallucinations. First, the UV layout of the original partial mesh is aligned with the newly created UV layout of the completed geometry. To achieve this, we project the original textures onto the completed geometry and unwrap it together with the material indices. Iteratively, all the patches in a material set $P \in S_i$ are used as reference samples to compute the average texture for the new regions (Figure 6), while $P \notin S$ are masked out. For every new patch, arbitrary square blocks $\{B_1, B_2, \dots, B_n\}$ from S_i are merged together with overlap to synthesize a new texture sample P' . The best fit cut between each two overlapping blocks is retrieved by minimizing neighboring contrasts $e_{ij} = f(B_i, B_j)$. The minimal cut is then obtained by traversing all cuts and computing the cumulative minimum error E for each block (Eq. 4).

$$E_{ij} = e_{ij} + \min(E_{i-1, j-1}, E_{i-1, j}, E_{i-1, j+1}). \quad (4)$$

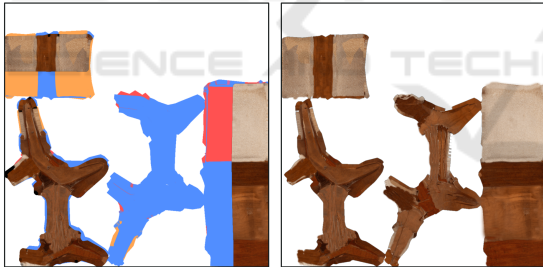


Figure 6: The inpainting process where each material is inpainted separately.

4 EXPERIMENTS

In this section, we will first discuss the dataset used, then the training of our models and finally discuss the results of our experiments.

4.1 Dataset Preprocessing

Two datasets are used for the experiments. ShapeNet-Core (Chang et al., 2015) is a synthetic object library that we use for training and validation. It contains 55 common object categories like chairs, benches and tables, with about 51.300 unique 3D models. It is a

good training dataset since it both contains the completed geometries and also the material indices for the textures so both the AutoSDF and IF-Net have correct ground truth data.

On the other hand, Matterport (Chang et al., 2017) is a scanned dataset that we use for the evaluation. It consists of 90 fully textured building-scale scenes, with each between 15-30 objects that can be segmented and completed (Figure 7). It is ideally suited to investigate the domain-transfer capabilities of the network to deal with realistic textures and incomplete geometries. No ground truth is available for this dataset so a visual study is made of the resulting reconstructions.

A relevant subselection is made from both datasets for the experiments. The AutoSDF training dataset is generated by converting the meshes to a normalised, aligned 128^3 SDF grid as discussed in section 3.1. The IF-Net input data is created by separating each submesh and giving it a material index. Each mesh is sampled to a colored point cloud from which 4 incomplete variations are created by randomly removing parts of the point cloud. These four incomplete colored point clouds, along with the uncolored complete point cloud are used as the training input.



Figure 7: Examples of isolated objects from the Matterport dataset with varying occlusions and shapes.

4.2 Training

For AutoSDF and SAM, we use the pre-trained models available to the public because they are trained on relevant datasets. However, the IF-Net was retrained using the Adam optimizer with a learning rate of 10^{-4} for 1000 epochs with a minimal loss of 65.81 using our custom dataset. The query points for the training data are obtained by sampling the ground truth for 100,000 points. The partial scans are voxelised by sampling 100,000 points from the partial surface and setting the occupancy value in the nearest voxel grid to 1. Similarly, for the colored voxelisation, the value of the nearest voxel is set to the three-channel value of the corresponding material index.

4.3 Results

4.3.1 Geometry Completion

For the geometry completion using AutoSDF, the validation is performed by completing the objects from the ShapeNet dataset (Figure 8) at different levels of completeness. Table 1 shows the resulting average MIOU and Chamfer distance of the dataset. Each object is completed with 25%, 50% and 75% of the original mesh remaining. These results show that the MIOU and Chamfer distance increase when more of the original mesh is present. There is, however, still a loss in accuracy due to the voxel-based SDF conversion, leading to lower MIOU.

Table 1: The MIOU of and Chamfer Distance on the ShapeNet Core v2 dataset, completed at 25, 50 and 75% respectively.

Shapenet Core v2	25% completion	50% completion	75% completion
MIOU	25.26%	54.14%	62.17%
Chamfer Distance	0.09	0.06	0.06



Figure 8: The results from the ShapeNet Core dataset. The rows show the partial inputs, the completed geometry, the final output and the the ground truth.

For the Matterport data, we focus on the visual fidelity and accuracy. Since the completion uses a VQ-VAE network, multiple probable solutions are generated, as seen in Fig 9. There is a large variety in proposed solutions due to the large and detailed voxel selection necessary because the UDFs can be hollow in the missing areas. The best option is determined based on the largest overlap with the original incomplete mesh.



Figure 9: Examples of the multiple results returned from AutoSDF geometry completion with the input UDF (left) and four possible outputs (right).

4.3.2 Material Prediction

The material prediction is validated by calculating the percentage of correctly predicted points as seen in Table 2. The IF-Net can accurately predict the correct material index if the materials are all present in the partial scan. It does not introduce new materials, leading to lower correctness percentages at the lower completion levels. For smaller defects or missing parts, the material mostly stays consistent.

Table 2: The average material prediction accuracy on the ShapeNet Core v2 dataset, completed at 25, 50 and 75% respectively.

ShapeNet Core V2	25% completion	50% completion	75% completion
Material Correctness	60.56%	80.78%	92.07%

The material completion returns good results on the Matterport data for large uniform areas as seen in Figure 10, where the highlighted areas A get a much better result due to a larger reference area. The highlighted B areas have a limited amount of reference area, so they have a very clear repeating pattern. SAM often overly segments because of color artifacts in the original scans, leading to a very high number of patches. Its drawback is that it is impossible to define a single cosine similarity threshold to group the different patches. Therefore, we adapt the threshold to fit the 8-materials constraint.

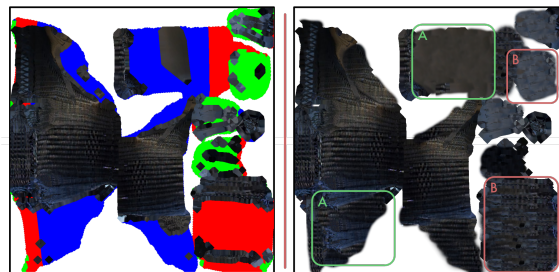


Figure 10: The inpainting results (right) with the partial texture (left). highlighted areas "A" and "B" indicate good and poor results respectively.

4.3.3 Texture Inpainting

The performance of the texture inpainting is measured with the cosine similarity of the predicted patches compared to the ground truth. Table 3 shows the results at 3 different completion levels. Due to the material prediction step, the inpainter only relies on one type of material as the training area, leading to high results across the board.

Table 3: The average texture inpainting similarity on the ShapeNet Core v2 dataset, completed at 25, 50 and 75% respectively.

ShapeNet Core V2	25% completion	50% completion	75% completion
Cosine Similarity	86.34%	90.00%	91.40%

The patched-based inpainting model inpaints the textures as seen in Figure 11 with a patch size of 8 and an overlap size of 2. To increase the rotational invariance, we use rotations of $[0, 45, 90, 135, 180]$ degrees.

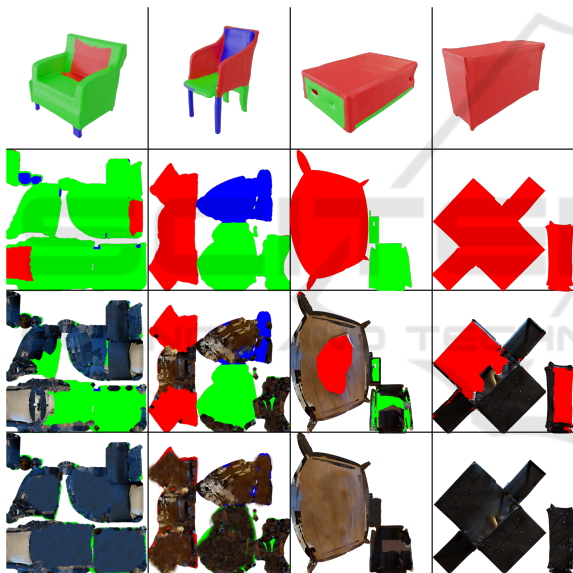


Figure 11: The inpainting results. The first row shows the predicted material indexes, the second row shows the UV texture map, the third row the mapped incomplete original texture and the final row the texture inpainting.

4.3.4 Full Completion

Figure 12 illustrates the completion results on the Matterport dataset compared to the state of the art. Objects with near-complete scans, such as the sofa and stool, yield consistent geometry and texture completions. In contrast, less-scanned objects still produce plausible geometries but face challenges in texture inpainting. Extensive missing areas result in insufficient reference patches, leading to repetitive textures.

5 DISCUSSION

The AutoSDF network demonstrates robust geometry completion, effectively predicting large missing areas even for meshes with limited ground truth. However, it sacrifices fine details, and the VAE encoding alters originally observed parts. Despite this, AutoSDF outperforms the SOTA Multimodal Point-cloud Completion method (MPC) in preserving existing parts, as shown in (Mittal et al., 2022).

Material segmentation with SAM excels at accurately segmenting basic materials like wood, fabric, and plastic, regardless of orientation. However, it struggles with very small patches due to 2D resolution limits and is affected by lighting conditions, as shadows and reflections are baked into the object during capture, similar to (Siddiqui et al., 2022).

Material prediction enhances boundary definitions between materials, improving representation. Challenges remain with extensive missing areas or objects featuring numerous distinct materials. Patch-based image inpainting struggles with irregular patterns, such as printed illustrations or intricate details, underscoring the need for better handling of nonuniform textures. Unlike (Stets et al., 2019), which limits segmentation to predefined material classes, our method assigns generic material labels to patches, abstracting actual materials and shifting the challenge to 2D inpainting.

Incorporating the material mask—a map indicating patches with the same material index—into the completion process ensures cleaner reference areas and facilitates effective inpainting for repeating textures. However, difficulties in inpainting orientation and the reliance on UV layout pose challenges, potentially limiting the method’s broader applicability.

6 CONCLUSION

This study presents a novel material prediction step in the geometry and texture completion pipeline for partially scanned 3D objects. The process begins with geometry prediction to establish the structure, followed by a three-step texture completion. First, the partial UV map undergoes material segmentation using SAM to abstract the object for alignment with training data. Next, the IF-Net network predicts the material for missing areas. Finally, a 2D inpainting refines the texture on the UV map for visual detail.

Our method delivers promising results, particularly with real scans, achieving clearer material boundaries and advancing the state of the art. However, areas for improvement remain: enhancing UV

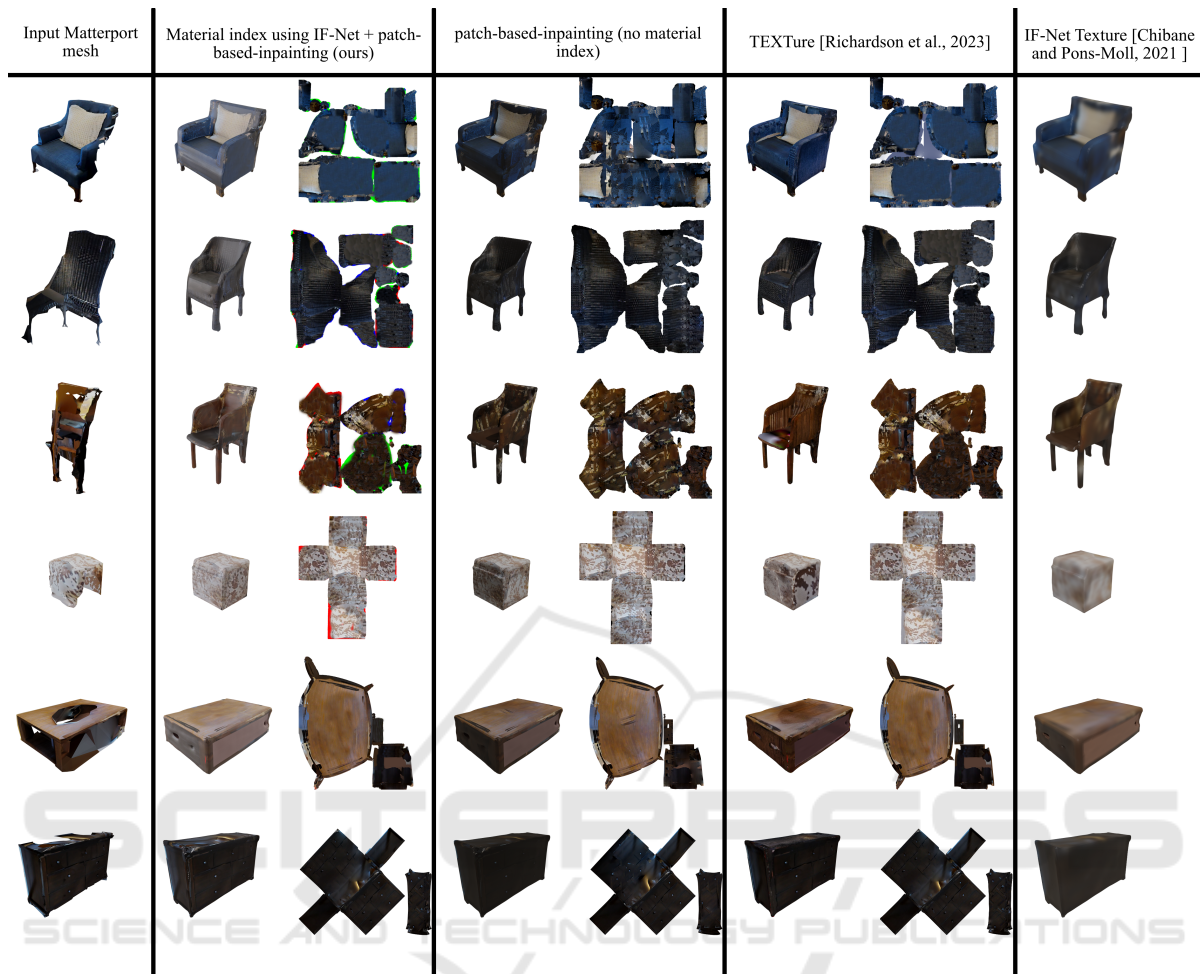


Figure 12: The texture inpainting results from the Matterport dataset using our method compared against: patch-based inpainting, TEXTure, and IF-net Texture.

unwrapping could refine texture mapping, and better occlusion detection would improve scene accuracy. Future work may explore applying this approach to full-scene reconstructions.

REFERENCES

- Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*.
- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). Shapenet: An information-rich 3d model repository.
- Chen, Z., Liu, L., Zhang, Z., and Wang, G. (2007). Surface parameterization via aligning optimal local flattening. In *Proceedings - SPM 2007: ACM Symposium on Solid and Physical Modeling*, pages 291–296.
- Cheng, A.-C., Li, X., Liu, S., and Wang, X. (2023). Tuvf: Learning generalizable texture uv radiance fields.
- Cheng, Y.-C., Lee, H.-Y., Tulyakov, S., Schwing, A., and Gui, L. (2022). Sdfusion: Multimodal 3d shape completion, reconstruction, and generation.
- Chibane, J., Alldieck, T., and Pons-Moll, G. (2020). Implicit functions in feature space for 3d shape reconstruction and completion.
- Chibane, J. and Pons-Moll, G. (2020). Implicit feature networks for texture completion from partial 3d data.
- Chu, R., Xie, E., Mo, S., Li, Z., Nießner, M., Fu, C.-W., and Jia, J. (2023). Diffcomplete: Diffusion-based generative 3d shape completion.
- Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346. Association for Computing Machinery.
- Flavell, L. (2010). *Beginning Blender : open source 3D modeling, animation, and game design*. Apress.

- Gao, D., Rozenberszki, D., Leutenegger, S., and Dai, A. (2023). Diffcad: Weakly-supervised probabilistic cad model retrieval and alignment from an rgb image.
- Gao, L., Wu, T., Yuan, Y.-J., Lin, M.-X., Lai, Y.-K., and Zhang, H. (2020). Tm-net: Deep generative networks for textured meshes.
- Huang, J., Zhang, H., Yi, L., Funkhouser, T., Nießner, M., and Guibas, L. (2018). Texturennet: Consistent local parametrizations for learning from high-resolution signals on meshes.
- Jacobson, A., Kavan, L., and Sorkine-Hornung, O. (2013). Robust inside-outside segmentation using generalized winding numbers.
- Karadeniz, A. S., Ali, S. A., Kacem, A., Dupont, E., and Aouada, D. (2022). Tscm-net: Coarse-to-fine 3d textured shape completion network.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment anything.
- Liang, Y., Wakaki, R., Nobuhara, S., and Nishino, K. (2022). Multimodal material segmentation rgb near infra-red angle of polarization degree of polarization material segmentation annotation.
- Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.-Y., and Lin, T.-Y. (2022). Magic3d: High-resolution text-to-3d content creation.
- Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., and Su, H. (2023). One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion.
- Lorenson, W. E. and Cline, H. E. (1998). Marching cubes: A high resolution 3d surface construction algorithm.
- Lugmayr, A., Danelljan, M., Fisher, A. R., Radu, Y., Luc, T., and Gool, V. (2022). Repaint: Inpainting using denoising diffusion probabilistic models.
- Maggiordomo, A., Cignoni, P., and Tarini, M. (2021). Texture defragmentation for photo-reconstructed 3d models. *Computer Graphics Forum*, 40:65–78.
- Maggiordomo, A., Cignoni, P., and Tarini, M. (2023). Texture inpainting for photogrammetric models. *Computer Graphics Forum*, 42.
- Mittal, P., Cheng, Y.-C., Singh, M., and Tulsiani, S. (2022). Autosdf: Shape priors for 3d completion, reconstruction and generation.
- Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., and Geiger, A. (2019). Texture fields: Learning texture representations in function space.
- Rao, Y., Nie, Y., and Dai, A. (2022). Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories.
- Richardson, E., Metzger, G., Alaluf, Y., Giryas, R., and Cohen-Or, D. (2023). Texture: Text-guided texturing of 3d shapes.
- Sharma, P., Philip, J., Gharbi, M., Freeman, B., Durand, F., and Deschaintre, V. (2023). Materialistic: Selecting similar materials in images. *ACM Transactions on Graphics*, 42.
- Siddiqui, Y., Thies, J., Ma, F., Shan, Q., Nießner, M., and Dai, A. (2021). Retrievalfuse: Neural 3d scene reconstruction with a database.
- Siddiqui, Y., Thies, J., Ma, F., Shan, Q., Nießner, M., and Dai, A. (2022). Texturify: Generating textures on 3d shape surfaces.
- Sipiran, I., Gregor, R., and Schreck, T. (2014). Approximate symmetry detection in partial 3d meshes. *Computer Graphics Forum*, 33:131–140.
- Stets, J. D., Lyngby, R. A., Frisvad, J. R., and Dahl, A. B. (2019). Material-based segmentation of objects. In *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11482 LNCS, pages 152–163. Springer Verlag.
- Sun, C., Tong, X., and Liu, Y. (2022). Semantic segmentation-assisted instance feature fusion for multi-level 3d part instance segmentation. *Computational Visual Media*.
- Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks.
- Vermandere, J., Bassier, M., and Vergauwen, M. (2022). Two-step alignment of mixed reality devices to existing building data. *Remote Sensing*, 14.
- Vermandere, J., Bassier, M., and Vergauwen, M. (2023). Texture-based separation to refine building meshes. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-1/W1-2023:479–485.
- Vermandere, J., Bassier, M., and Vergauwen, M. (2024). Semantic uv mapping to improve texture inpainting for indoor scenes.
- Wang, P.-S., Liu, Y., and Tong, X. (2022). Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Transactions on Graphics (SIGGRAPH)*, 41.
- Yu, X., Dai, P., Li, W., Ma, L., Liu, Z., and Qi, X. (2023). Texture generation on 3d meshes with point-uv diffusion.
- Zhou, L., Du, Y., and Wu, J. (2021). 3d shape generation and completion through point-voxel diffusion.