Guided object completion with interactive voxel editing

Jelle Vermandere[1], Maarten Bassier[1], Maarten Vergauwen[1]

[1] KU Leuven, Department of Civil Engineering, Ghent, Belgium
(jelle.vermandere, maarten.bassier, maarten.vergauwen)@kuleuven.be

Keywords: GSW 2025, Voxel, Object completion, Generative modeling, interaction

Abstract

Object completion in 3D scanned indoor scenes remains a challenging problem, as most current approaches either focus on completing entire scenes or isolated objects. Completing objects within their scene context is still an area of active research. A key limitation of existing methods is their disregard for the scene's environmental cues—such as walls and floors—which could provide valuable information for defining the boundaries of incomplete objects. Additionally, object completion models are often trained on synthetic datasets, where objects are neatly aligned and centred, unlike real-world scanned data that is typically unaligned. This misalignment hinders the practical application of existing models, although some approaches have attempted to address this by estimating symmetry planes. State-of-the-art (SOTA) methods also face challenges in guiding object completion, often relying on a range of potential outputs with minimal user interaction. In this work, we aim to improve the completion of objects from partially scanned indoor scenes by leveraging environmental cues to better inform the boundaries of incomplete objects. Furthermore, we introduce an interactive voxel editor that allows users to guide the object completion process toward more accurate results. Our contributions are twofold: (1) a novel boundary-defining and object-alignment method that integrates with existing object completion pipelines, and (2) the development of an interactive voxel editing tool that enhances user control over the completion process. Experimental results demonstrate the effectiveness of our approach in improving object completion in complex, real-world scanned scenes.

1. Introduction

Dynamic indoor environments originating from 3D scans are increasingly in demand within the gaming industry and the Architecture, Engineering, Construction, and Operations (AECO) sectors (Vermandere et al., 2022). Similar to digitally created scenes, these environments consist of collections of digital objects that can be interacted with independently from static walls and floors (e.g., by modifying or removing objects).

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). Consequently, there is an urgent need for effective completion methods.

Current methods for object completion are either focused on completing the scene as a whole (Dai et al., 2018) or on completing already isolated objects (Mittal et al., 2022). When completing the scene as a whole, missing regions due to sensor occlusions can be filled in; however, occlusions between contacting objects often remain unresolved. This is why many models first attempt to detect the objects in the scene, after which they are isolated from their context. This approach results in the environment being ignored in the final completion process. The fact that scene structures such as walls, floors, and other objects can provide key insights into the boundaries of partial objects remains largely unexplored in the state of the art (SOTA).

Object completion models are often trained on synthetic data (Mittal et al., 2022, Cheng et al., 2023, Zhou et al., 2021), primarily due to the limited availability of real-world data and the ease of use of properly aligned and clean 3D models. While synthetic data simplifies the training process, it also limits the models' effectiveness for real scanned objects, which are rarely aligned or centred around their approximate centres. Works such as (Sipiran et al., 2014, Mitra et al., 2006, Shi et al., 2020, Gao et al., 2019) attempt to address this problem by estimating symmetry planes of incomplete objects. Together with environmental structures like floors and walls, these symmetry planes can help match partial objects to synthetic input formats by aligning them according to their principal axes of symmetry and physical boundaries.

State-of-the-art (SOTA) object completion models (Mittal et al., 2022, Cheng et al., 2023) can provide a range of possible completion results due to the encoding-decoding process of the VAE model, increasing the likelihood that one of the provided options will be a good fit. However, the guiding capabilities of these models are currently limited to sub-bounding boxes within the voxel grid, as determined by their training methods. By incorporating a voxel editor tool, objects can be more precisely guided to the desired shape. These guiding voxels can be generated based on environmental alignment cues and further refined by the end user.

The main goal of this research is to improve object completion for objects originating from partially scanned indoor scenes. This goal is achieved by leveraging environmental cues and symmetry axes to more accurately define the physical boundaries for the completion network. Additionally, by enabling users to guide the object completion with a voxel editor, the final object completion can be significantly enhanced.

The main contributions of this work are twofold. First, we introduce a novel boundary-defining and object alignment method to better fit partial objects into existing object completion pipelines.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-901-2025 | © Author(s) 2025. CC BY 4.0 License.

Second, we develop an interactive voxel editor to more effectively guide object completion toward its desired shape.

The remainder of this work is structured as follows. The background and related work is presented in Section 2. Following is the explanation of the proposed method in Section 3. In Section 4, an overview of the used datasets and their results is presented. Finally, the conclusions are presented in Section 5

2. Background and related work

2.1 Object Detection

Object detection in a 3D scene is performed by clustering points or voxels belonging to a given object. VoteNet (Qi et al., 2019) achieves this by using deep Hough voting to cluster each point and identify clusters that could form a single object. This approach is improved in MLCVNet (Xie et al., 2020), which introduces a multi-level context to enhance clustering accuracy. V-DETR (Shen et al., 2023), on the other hand, applies DETR (Detection Transformer) in 3D with Vertex Relative Position Encoding to improve locality. These models produce a range of bounding boxes that encapsulate potential objects.

2.2 Object Completion

3D scanned environments are typically captured as unstructured pointclouds or meshes. Both of these formats are irregular and are not easily used in machine learning networks. This is why the models are often converter to either standardized pointclouds or Signed Distance Fields (SDF) Both of which, can be mapped to a fixed input size.

Point-based geometry completion like Point-Voxel-diffusion (Zhou et al., 2021) uses a fixed-size pointcloud as input to predict the final shape through 3D diffusion. IF-Nets (Chibane et al., 2020) also use points, but employs implicit features generated from those points to predict the missing regions. These models provide good results for general shapes, but lack in fine detail completion due to the amount of noise and lack of a clear surface definition typically present in point clouds.

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 (Mittal et al., 2022). An SDF can be discretised into a voxelgrid to standardize the input size. These have become a popular input type because they retain the shape of the object while using less data points. Models like PatchComplete (Rao et al., 2022) and WSSC (Wu et al., 2024) use a coarse-to-fine approach by first predicting the general shape and then refining each sub-grid using multi-resolution priors. Shapeformer (Yan et al., 2023) is able to leverage the Transformer architecture by using a vector quantized deep implicit function (VQDIF) to represent an incomplete shape.

Models like AutoSDF (Mittal et al., 2022) are able to encode the SDFs and, by spliting the SDF in sub-grids during training, can predict the missing geometry. SD Fusion (Cheng et al., 2023) builds upon this by allowing multi-modal input types to guide the generation. XCube (Ren et al., 2023) expands upon the object completion by employing a hierarchical voxel latent diffusion model which generates progressively higher resolution grids in a coarse-to-fine manner using a custom framework built on the highly efficient VDB data structure. This enables the model to generate much larger scenes.

These works all rely on a voxelised Truncated Signed Distance Field(TSDF) as input, allowing the user to highlight the parts which need to be completed by highlighting certain voxels, however, most of these works are limited to range selections for this purpose. XCube (Ren et al., 2023) has stated a potential voxel guiding workflow using an off-the-shelf voxel editor to define the guiding voxels. This is however practically limited because there is no feedback loop between the generation and guiding due to the lack of software integration. Works like Interactive Voxel Editing (Wegen et al., 2022) made it possible to edit the full scene by providing boolean tools to isolate and delete voxels from the scene. However, a true interactive voxel editor build for object completion does not exist yet.

2.3 Environment aided Completion

The main cause of incomplete 3D scans often come from occlusions, be it from the object self, or the surrounding environment. Instead of ignoring the environment, some works like tracking partially-occluded objects (Wang et al., 2021) try to look past these occlusions and use the environment estimate the hidden objects complete shape. Co-Section (Strecke and Stückler, 2020) on the other hand uses the object detection from EM-Fusion (Strecke and Stückler, 2019) and leverages intersection constraints from walls and floors to infer hidden shape information. This clearly defines object boundaries creates physically plausible 3D objects.

2.4 Object Alignment

Most object completion networks are trained on normalised training data, for object completion, this means the objects are scaled and aligned before they enter the network. This is trivial for most synthetic data, but aligning partially scanned 3D objects is still a field of ongoing research. Finding symmetry in complete objects is possible by works like PRS-Net (Gao et al., 2019), which employs a novel learning framework to automatically discover global planar reflective symmetry of a 3D shape by training an unsupervised 3D convolutional neural network to extract global model features.

Partial symmetry completion is more complex, because there is no guarantee to find symmetrical features. Early works like Partial approximate symmetry detection (Mitra et al., 2006) developed an algorithm that processes geometric models and efficiently discovers and extracts a compact representation of their Euclidean symmetries. This was improved by Approximate symmetry detection (Sipiran et al., 2014), which uses local extrema to find corresponding symmetry points and aligns the partial mesh. SymmetryNet (Shi et al., 2020) Only requires a single RGB-D input to discover global planar reflective symmetry by using an unsupervised 3D convolutional network to extract global model features.

3. Methodology

The presented method (Figure 1) illustrates the full workflow. First, scene detection is performed by detecting the objects using Votenet (Qi et al., 2019), these are removed from the scene and the planes are detected in the remaining geometry using a RANSAC algorithm. Second, we perform the object alignment step where each object is processed to find its primary symmetry axis using Approximate symmetry detection (Sipiran et al., 2014) and aligned accordingly. The scene planes are used to

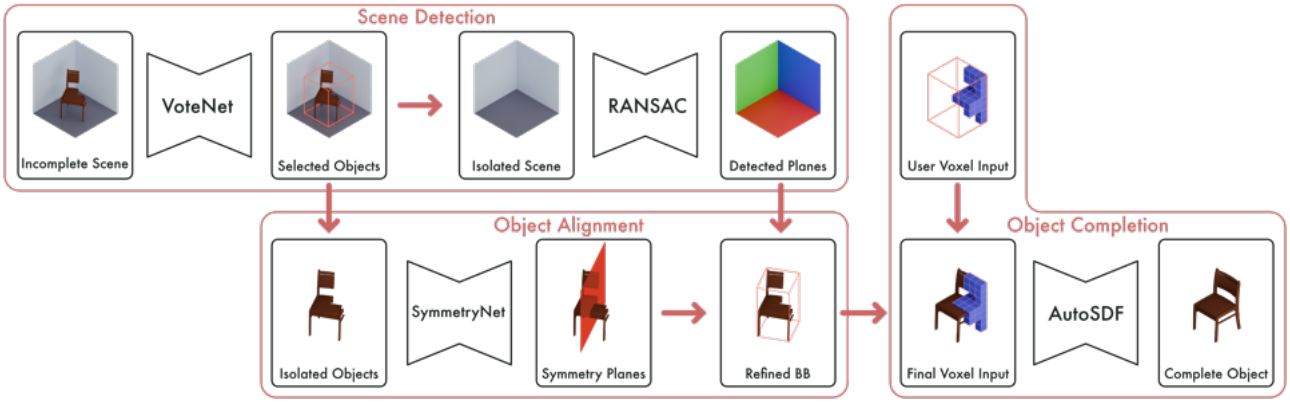

Figure 1. Overview of the object completion pipeline, starting with an incomplete scene (left), followed by a object and plane detection (top-centre), and a symmetry detection to combine into Bounding box refinement (bottom-centre). The user-input (top-right) is combined with the predicted voxel input to result in a completed, textured mesh using AutoSDF (right).

limit the object's bounding box. Finally, the user is able to refine the completion voxels using the voxel editor. Those voxels are used to predict the missing geometry from the incomplete inputs by utilizing implicit shape representations with AutoSDF (Mittal et al., 2022). This results in a list of complete objects.

3.1 Scene Detection

To isolate the objects from the scene, we perform an object detection using VoteNet(Qi et al., 2019) on the whole scene as seen in Figure 2. Since the network requires a point cloud as an input, the incomplete scene is sampled to a pointcloud with a $5cm$ resolution. This provides a good balance between detail and execution speed. AutoSDF has a tendency to over-detect a scene, this is why overlapping boxes are combined into a larger boxes if the Intersection over Union (IOU) is larger then 80%. After the bounding boxes are cleaned up, they are used to remove the objects from the scene. This results in a list of unaligned objects and a remaining scene that can be further segmented.

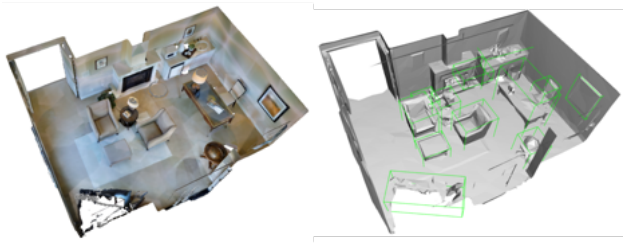

Figure 2. Left, an indoor scene filled with furniture. Right, The resulting detected bounding boxes of the objects highlighted in green.

The resulting Empty scene is further segmented using a RANSAC plane detection as illustrated in 3. Since the scene was sampled in the previous step, we can perform a point-wise RANSAC plane detection. To define proper boundaries, the intersections between the planes are computed and used to define boundaries. Those boundaries, together with the planes normal, these are used to compute the final quads used to limit the bounding box in the next step.

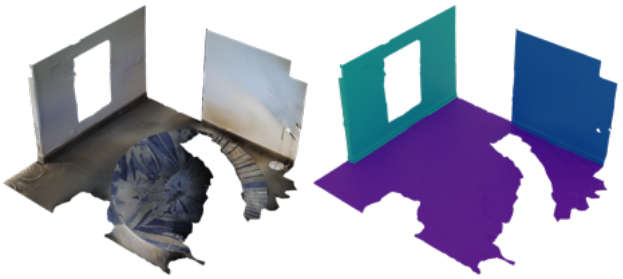

Figure 3. Left, an indoor scene where the detected objects have been removed. Right, the 3 detected planes from the RANSAC algorithm.

3.2 Object alignment

The detected incomplete objects have an initial axis aligned bounding box. However, this does not necessarily align with the objects orientation, which is needed to get a proper completion in the next step. Since most indoor objects have some sort of symmetry or orthogonal construction due to manufacturing constraints, we can estimate the object alignment by finding the principal symmetry axis. This is done by finding pointwise symmetry pairs in the incomplete mesh using Approximate symmetry detection (Sipiran et al., 2014) as illustrated in Figure 4.

Together with the planes computed in the previous step, the aligned bounding box is computed. This is done by first checking if the object is grounded or mounted to a wall by computing the adjacency to each plane. This is used to fix the first axis of rotation labelled as ”up” as the normal to the detected plane. Then the symmetry axis is used to find the second rotation axis defining the ”right” vector as the normal of the symmetry plane. Those two vectors define a unique rotation that is used to orient the incomplete object.

In a final step, the planes are used to refine the boundaries of the bounding box. limiting the range at which the object could theoretically extend. This is further enhanced by using the symmetry axis as the centre of the bounding box. When an object has multiple axis, the alignment can be further refined.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-901-2025 | © Author(s) 2025. CC BY 4.0 License.

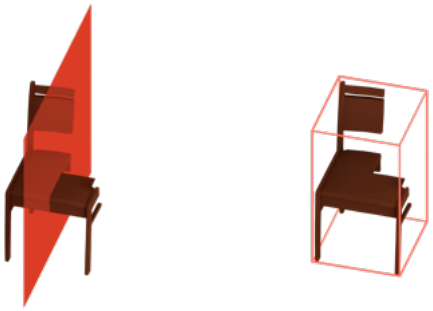

Figure 4. The detected symmetry plane of the partial object on the left and the resulting refined bounding box on the right.

3.3 Object Completion

The object completion network needs the incomplete TSDF and a list of voxels that represent the missing parts (Mittal et al., 2022). The TSDF is created from the incomplete mesh and is discretised using the refined bounding box divided into a 64^3 voxel grid.

The next step is defining the voxels to be completed, this is done using a user interface where the incomplete mesh is positioned within an empty voxel grid as seen in Figure 5. However, to provide an initial guess of the to-be-completed voxels, the symmetry axis is used to mirror the occupied voxels.

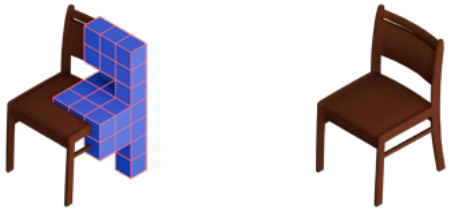

Figure 5. The proposed voxels used to complete the partial object on the left and the resulting completed object on the right.

The predicted voxels, together with the incomplete UDF form the basis for the user interface. The user is able to edit the initial guess by adding and removing new voxels using the left and right mouse buttons respectively. This is all performed in a intuitive 3D environment as seen in Figure 6.

4. Experiments

To evaluate the effectiveness of our method, we compared the object completion results from a set of objects detected in a real scanned dataset. This evaluation begins by performing object completion on the partial object as it is detected in the scene, without any refinement. Next, we apply our method, which aligns the object to its principal axis and constrains the bounding box according to the physical boundaries. The results of our experiments are shown in Figure 7

4.1 Dataset

For the experiments, we used the Matterport (Chang et al., 2017) dataset, a scanned dataset consisting of 90 fully textured building-scale scenes, each containing between 15-30 objects that can be

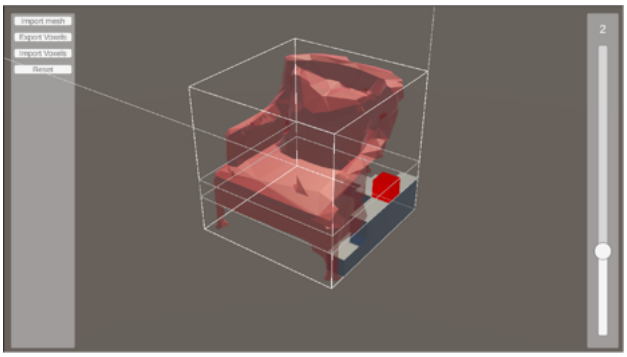

Figure 6. The interface for the voxel editor. The left side provides an intuitive set of buttons to manage data. The right slider limits the voxel drawing to a fixed height so the user can draw the voxel in the centre.

detected and completed. Because no ground truth is available for this dataset, the completion is evaluated on a visual basis. We selected a few objects to highlight, which are shown in context in Column 1 of Figure 7.

4.2 Plane detection

The effectiveness of the plane detection for bounding box refinement is evaluated on a freestanding object and an object against both an axis-aligned and a non-axis-aligned wall. The detected planes are shown in Column 3 of Figure 7. When the object is not positioned against a wall, the impact is minimal, as the only relevant plane is the floor. Since the detection bounding boxes are aligned with the global axis, and the floor is typically level, the bounding box aligns naturally with the floor. However, for objects positioned against a wall, especially when the walls do not align with the global orthogonal axis, the additional plane provides a clear initial alignment and physical boundary for the object. This advantage is illustrated in Row 1 of Figure 7, where the detected wall behind the cupboard provides a clear depth limit.

4.3 Object alignment

Proper alignment of the object has the greatest impact on the completion results. As shown in Column 2 of Figure 7, the completion network struggles to interpret unaligned objects and attempts to create orthogonal shapes from them. This issue is most evident in Row 1 of Figure 7, where the cupboard is rendered as a triangular shape instead of its original rectangular form. Even for objects with known boundaries, as in Row 3 of Figure 7, the network fails to complete the object in a meaningful way.

4.4 Object completion

The final results show that the overall shape is largely preserved. However, due to the encoding-decoding process of the VAE model, not all details are retained. While the results appear plausible, we observe that some objects are slightly altered, even in regions of the object that were already known. The voxel editor offers a quick way to refine the voxels needed for the completion network.

5. Conclusions

The evaluation of our proposed object completion method demonstrates its effectiveness, with key factors influencing the quality of results. Among these, object alignment proves to be the

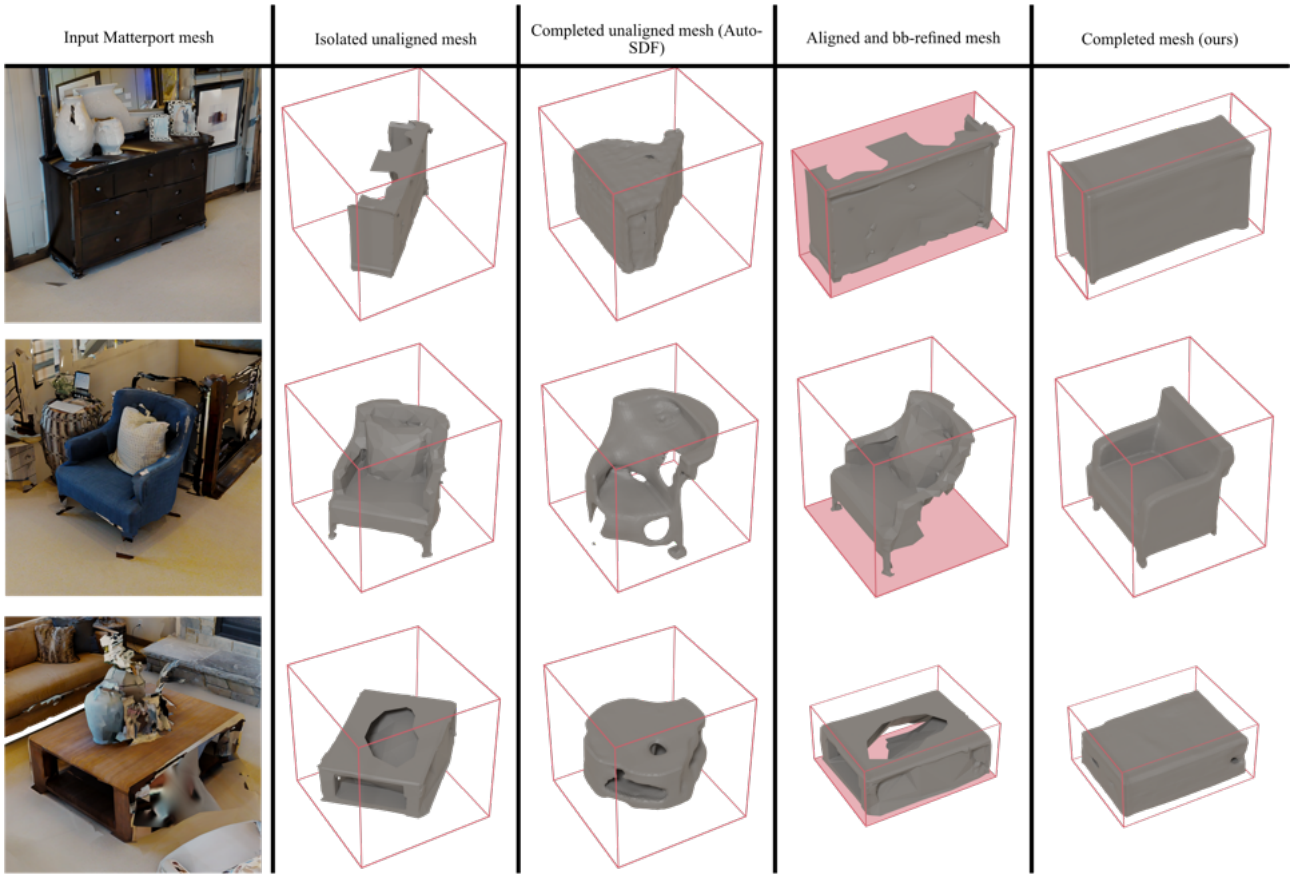

Figure 7. Experimental results: The first column shows the objects in their context, the second the unaligned isolated objects, the third the unaligned completed objects using Auto-SDF, the fourth the aligned objects with refined bounding boxes, highlighting the detected planes used for refinement, and the final column shows our results.

most crucial for achieving accurate and realistic completion results. Objects positioned against other surfaces or objects benefit from better boundary definitions, which enhance bounding box refinement and lead to more reliable completions. Aligned objects generally yield good completion outcomes; however, some fine details are inevitably lost due to the SDF conversion process. Additionally, the voxel editor serves as a quick and effective tool for specifying voxels for completion, allowing users to refine the final output further.

Despite these strengths, there are limitations to our approach. The detection model struggles to accurately segment very small details and objects with limited geometric information. Freestanding objects, in particular, have less environmental context to guide alignment and therefore rely more on symmetry, which is not always sufficient. Moreover, not all objects possess inherent symmetry, which can lead to alignment errors and reduce the precision of the completion process.

Overall, our method shows promise in refining object boundaries and improving completion in complex scenes, especially when alignment and boundary context are available. Future work will focus on addressing limitations related to asymmetry and enhancing segmentation for detailed structures.

ACKNOWLEDGEMENTS

This project has received funding from the FWO-SB grant (grant agreement 1S16923N) and the Geomatics research group of the Department of Civil Engineering, TC Construction at the KU Leuven in Belgium.

References

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y., 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)*.

Cheng, Y.-C., Lee, H.-Y., Tulyakov, S., Schwing, A., Gui, L., 2023. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. *CVPR*.

Chibane, J., Alldieck, T., Pons-Moll, G., 2020. Implicit functions in feature space for 3d shape reconstruction and completion. *CVPR*.

Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., Niebner, M., 2018. ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4578-4587.

Gao, L., Zhang, L.-X., Meng, H.-Y., Ren, Y.-H., Lai, Y.-K., Kobbelt, L., 2019. PRS-Net: Planar Reflective Symmetry Detection Net for 3D Models. *IEEE Transactions on Visualization and Computer Graphics*, 27, 3007-3018. http://arxiv.org/abs/1910.06511 http://dx.doi.org/10.1109/TVCG.2020.3003823.

Mitra, N. J., Guibas, L. J., Pauly, M., Zürich, E., 2006. Partial and Approximate Symmetry Detection for 3D Geometry. *ACM SIGGRAPH 2006 Papers*. https://doi.org/10.1145/1179352.1141924.

Mittal, P., Cheng, Y.-C., Singh, M., Tulsiani, S., 2022. Autosdf: Shape priors for 3d completion, reconstruction and generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 306–315.

Qi, C. R., Litany, O., He, K., Guibas, L., 2019. Deep hough voting for 3d object detection in point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob, 9276–9285.

Rao, Y., Nie, Y., Dai, A., 2022. Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories. *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 34436 – 34450.

Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F., 2023. XCube ($\{X\}^3$): Large-Scale 3D Generative Modeling using Sparse Voxel Hierarchies. http://arxiv.org/abs/2312.03806.

Shen, Y., Geng, Z., Yuan, Y., Lin, Y., Liu, Z., Wang, C., Hu, H., Zheng, N., Guo, B., 2023. V-DETR: DETR with Vertex Relative Position Encoding for 3D Object Detection. http://arxiv.org/abs/2308.04409.

Shi, Y., Huang, J., Zhang, H., Xu, X., Rusinkiewicz, S., Xu, K., 2020. SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images. *ACM Transactions on Graphics (SIGGRAPH Asia 2020)*, 39. https://doi.org/10.1145/3414685.3417775.

Sipiran, I., Gregor, R., Schreck, T., 2014. Approximate Symmetry Detection in Partial 3D Meshes. *Computer Graphics Forum*, 131-140. https://doi.org/10.1111/cgf.12481.

Strecke, M., Stückler, J., 2019. Em-fusion: Dynamic object-level slam with probabilistic data association. *Proceedings IEEE/CVF International Conference on Computer Vision 2019 (ICCV)*, IEEE.

Strecke, M., Stückler, J., 2020. Where does it end?-reasoning about hidden surfaces by object intersection constraints. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Vermandere, J., Bassier, M., Vergauwen, M., 2022. Two-Step Alignment of Mixed Reality Devices to Existing Building Data. *Remote Sensing*, 14. https://www.mdpi.com/2072-4292/14/11/2680.

Vermandere, J., Bassier, M., 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.

Wang, Y., McConachie, D., Berenson, D., 2021. Tracking partially-occluded deformable objects while enforcing geometric constraints. *Proceedings - IEEE International Conference on Robotics and Automation*, 2021-May, Institute of Electrical and Electronics Engineers Inc., 14199–14205.

Wegen, O., Döllner, J., Trapp, M., 2022. Interactive Editing of Voxel-Based Signed Distance Fields. *Journal of WSCG*, 30, 72-81.

Wu, L., Hou, J., Song, L., Xu, Y., 2024. 3D Shape Completion on Unseen Categories:A Weakly-supervised Approach. http://arxiv.org/abs/2401.10578.

Xie, Q., Lai, Y.-K., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J., 2020. MLCVNet: Multi-Level Context VoteNet for 3D Object Detection. *CVPR2020*. https://doi.org/10.48550/arXiv.2004.05679.

Yan, X., Lin, L., Mitra, N. J., Lischinski, D., Cohen-Or, D., Huang, H., 2023. ShapeFormer: Transformer-based Shape Completion via Sparse Representation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16, 2681-2689. https://shapeformer.github.io.

Zhou, L., Du, Y., Wu, J., 2021. 3d shape generation and completion through point-voxel diffusion. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5806–5815.