Synchronized RGB-D Inpainting for Privacy-Aware 3D Scene Graph Construction

Gawtam Chithra Ramesh[1,5], Püren Güler[1]✉, Hiba Alqaysi[1], Marcus Valtonen Örnhag[1], Héctor Caltenco[1], Erdal Akin[2,3,4], and Kayode Sakariyah Adewole[2,3]

[1] Ericsson Research, Lund, Sweden
[2] Department of Computer Science and Media Technology, Malmö University, Malmö, Sweden
[3] Sustainable Digitization Research Center, Malmö University, Malmö, Sweden
[4] Computer Engineering Department, Bitlis Eren University, Bitlis, Türkiye
[5] Royal Institute of Technology, Stockholm, Sweden

Abstract. 3D Scene Graphs (3DSGs) are hierarchical representations of environments that integrate semantic labels and structural information among objects, places and rooms with the geometric reconstruction. Such representations are efficient and compact making them well suited for many tasks, such as robotic navigation and extended reality (XR) interactions. A recent work, Privacy-aware Hydra (PA-Hydra), demonstrated that detecting and occluding privacy-sensitive objects (e.g., computer screens) with a black-box mask prevent their reconstruction in the mesh, mitigating sensitive objects exposure in a 3DSG framework called Hydra. However, such masking degrades scene verisimilitude, as the geometric reconstruction still contains traces from the privacy-sensitive objects erased from the scene (e.g., dark regions in the mesh). In this paper, we propose a synchronized RGB-D inpainting pipeline that is integrated to the 3DSG construction to replace masked sensitive objects with realistic scene inpainting. We incorporate state-of-the-art inpainting models into the Hydra framework to blend regions where the privacy-sensitive objects are into their surroundings. This ensures that the reconstructed regions do not trace back to the original sensitive objects. Our qualitative experiments on the uHumans2 dataset show that the proposed method improves visual verisimilitude and geometric continuity of the reconstruction compared to black-box masking, while reducing residual traces of removed objects. We position this pipeline as a building block for privacy-aware 3DSG construction in robotic and XR deployments.

Keywords: 3D scene graph · Inpainting · Privacy · Segmentation · Scene understanding · 3D Mapping · eXtended Reality

1 Introduction

3D Scene Graphs (3DSGs) are used to create hierarchical representations of 3D geometric information (meshes, voxel grids, point clouds, etc.) with semantic labels and relational connections between objects and spaces. Such a hierarchical

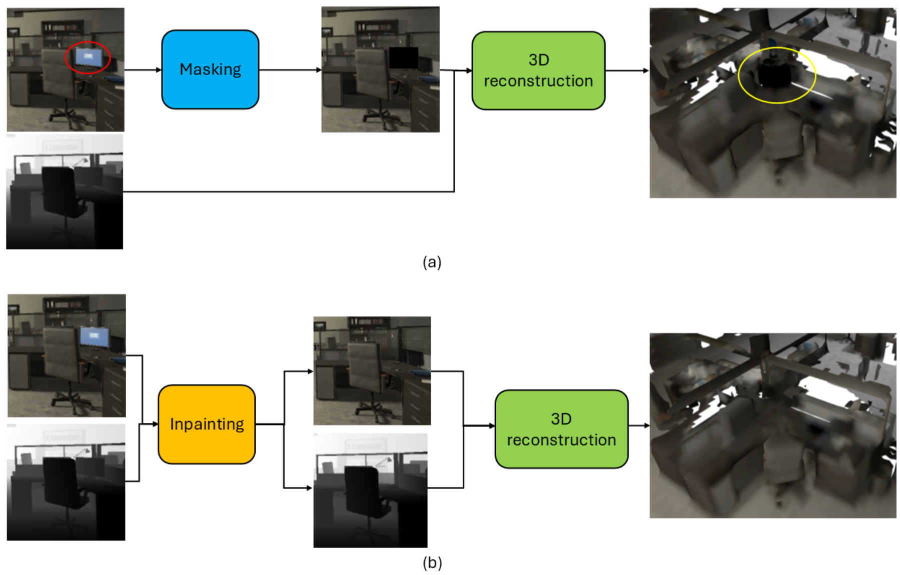


Fig. 1: (a) Privacy-aware Hydra [1] improves privacy in a 3D Scene Graph (3DSG) by masking the privacy-sensitive object (red circle) in the input RGB image before feeding that to the 3D reconstruction pipeline in Hydra framework [8] which generates a 3D reconstructed environment with embedded semantic information. The reconstruction leaves a dark region (yellow circle) in the mesh, indicating that an object was removed. (b) The proposed method that inpaints the privacy-sensitive object in both RGB and depth images which removes all traces of it within the reconstructed mesh.

structure allows an agent or application to understand the environment at multiple levels (mesh level, object level, room level, etc.) by integrating geometry and semantics into a single representation. Through such unified representation, they can enable tasks like path planning that takes into consideration both geometric structures and semantics, e.g., a robot responding to a query like "bring the book on the nearest shelf to the TV" or context-aware interactions, e.g. an extended reality (XR) game application suggesting which place in an apartment would be suitable to set up the playing area.

To construct a 3DSG, most frameworks such as Hydra [8] create a dense 3D reconstruction of the environment built using visual simultaneous localization and mapping (SLAM) or structure-from-motion (SfM). These algorithms construct the environment using RGB data that may contain sensitive personal data (personal documents, work computer screens, etc.). In real-world deployments, such frameworks can cause a violation of privacy laws or user consent preferences.

In Privacy-aware Hydra (PA-Hydra) [1], the authors address this problem by masking privacy-sensitive objects in RGB data before feeding them to the

3DSG pipeline, called Hydra [8]. However, masking with a black box can still reveal the locations of privacy-sensitive objects in the generated 3DSG (e.g., dark regions in the mesh, as seen in Fig. 1), compromising privacy. More realistic scene completion can enhance privacy preservation in 3DSGs.

In this work, we present a method that extends PA-Hydra by replacing masked privacy-sensitive regions with realistic, privacy-preserving scene completion using inpainting models integrated into the 3DSG pipeline. Our proposed method detects privacy-sensitive objects in a scene using object detection, masks the objects in the image, and inpaints the masked regions in both RGB and depth images before sending them for 3DSG generation. Therefore, the masked region blends seamlessly into the reconstructed mesh of the environment without any trace of the privacy-sensitive object.

We evaluated our method using the uHumans2 dataset, office scene [14]. We performed a qualitative evaluation, as no ground truth is available for the meshes in the uHumans2 dataset. We emphasize that our contribution is an integration and systems contribution rather than a new inpainting algorithm: we adapt off-the-shelf inpainting backbones and integrate them, via a shared occlusion mask across synchronized RGB and depth streams, into the Hydra 3DSG pipeline, and we empirically characterize which backbone is best suited to this privacy-preserving use case under real-time constraints.

Our main contributions are:

1. Enhancing the PA-Hydra framework by proposing a synchronized RGB-D inpainting method to blend masked privacy-sensitive regions more realistically with the environment.
2. Our approach increases the robustness of reducing privacy-sensitive object exposure in 3DSG creation and improves privacy awareness.
3. We evaluated our method using various inpainting models and demonstrated which method is the most effective through qualitative experiments.
4. We analyzed the proposed method across different setups, e.g., objects of different types and sizes, to evaluate it in different conditions.

In the following sections, we introduce relevant work in privacy-aware scene graphs (Section 2), the details of our framework (Section 3), experiments (Section 4), discussion of results and limitations (Section 5), and conclude with Section 6.

2 Related work

2.1 3D Scene Graphs

Early work by Armeni et al. [2] introduced 3DSG by converting metric-semantic 3D meshes into a graph representation, and Kim et al. [11] extended this to explicitly reconstruct objects and their relations. Rosinol et al. [14] proposed a 3DSG framework that is built directly from sensor data, includes place subgraphs that are useful for navigation, and represents objects, spaces, and dynamic objects. Hydra [8] advanced real-time incremental reconstruction of scene graph layers and demonstrated strong performance on various datasets, like uHuman2 introduced in [14].

Recent works improved the field in different directions, such as integrating open-vocabulary object detection into 3DSGs for better generalization as in [4], generating 3DSGs from multi-modal inputs (e.g., textual, visual, point cloud data) [7] or adapting 3DSGs to dynamic and complex environments using Gaussian splatting [6], [18].

However, none of the above works address the privacy issues when deploying such frameworks in real-world environments like IoT systems to control infrastructure of a workplace or mobile robots with helping house chores, which can capture privacy-sensitive data while constructing 3DSGs from visual-inertial sensory data (e.g., RGB images, IMU data).

2.2 Privacy preserving in robotics and XR

Overall, existing work provides strong building blocks but leaves several gaps for privacy-preserving construction of 3D scene graphs across different application areas. PA-Hydra [1] examined the privacy implications of 3DSGs, particularly in smart buildings and IoT systems. To prevent the 3DSG from reconstructing sensitive objects, the method integrated an object detection model alongside a filter to black-box and eliminate sensitive objects, such as people and laptops, before the mesh construction. This approach causes visual artifacts that may lead to identifying the location of sensitive objects in the environment (Fig. 1). Hence, it does not entirely prevent sensitive data leakage during 3DSG construction.

In robotics, privacy-aware scene graphs have also been explored through task-driven path planning. In [20], the authors discussed privacy aspects that arise from integrating 3DSGs and intelligent sensor-based systems in smart buildings in the context of robot path planning. They proposed specific algorithms, such as data minimization and obfuscation, within the path-planning framework to handle obstacles, such as people and restricted areas on the map.

Focusing on privacy at the level of 3D point clouds in visual localization and SLAM, the work in [16] introduced privacy-preserving image-based localization by encoding the 3D map in a way that prevents sensitive information from being extracted. The method replaces the conventional 3D structure from motion points with 3D line clouds, where each 3D point is replaced by a randomly oriented 3D line. However, subsequent work [3] has shown that line clouds still retain enough structure to approximately reconstruct scene details, indicating that this representation alone is not a complete privacy solution.

When it comes to privacy preservation in visual SLAM, recent work extends it to both the input images and the 3D map outputs. Yang et al. [19] introduced a dual component privacy preserving SLAM approach that inpaints sensitive image regions using object detection then encrypts point clouds into sparse line clouds. The inpainted image (with privacy-sensitive content removed/replaced) is then fed to the ORB-SLAM3 pipeline for feature extraction, matching, and pose estimation. It is reported that ORB-SLAM3 with inpainted images maintains pose accuracy close to the baseline. The work does not cover 3DSG integration.

In our work, to further improve the privacy-aware construction in 3DSG creation, we introduce a method to blend the masked region with the generated texture, blending with the environment seamlessly. In the next section (Section 3), we introduce the details of our framework.

3 Method

We propose a synchronized RGB-D inpainting method for privacy-preserving 3DSG construction that detects privacy-sensitive objects with semantic segmentation and removes the objects via inpainting from synchronized RGB-D data streams. The pipeline ensures that removed objects are erased not only visually (in 2D image) but also geometrically (in 3D scene), without the traces of the object location (e.g., a dark region in the reconstructed mesh) in downstream 3D mapping tasks. The system comprises three stages: sensitive instance segmentation, visual anonymization, and geometric structural restoration. As shown in Fig. 2, RGB and depth images are masked based on the output masks of the privacy-sensitive objects detected by YOLO-Seg [10]. The masked RGB image is sent directly to the inpainting module. The depth image is first normalized before sent to the inpainting module and is de-normalized after the inpainting. The inpainted RGB and depth images are fed to the Hydra framework to reconstruct a privacy-aware 3DSG. The pseudocode for our proposed method is presented in Algorithm 1. We detail each component of the algorithm in the following sections.

RGB and depth are inpainted in two independent passes by the same pretrained backbone; they are not jointly or cross-modally conditioned. The coupling between the two modalities is enforced solely through (i) the approximate time synchronizer that pairs each $\mathbf{I}_{\text{rgb}}$ with its corresponding depth frame $\mathbf{D}$, and (ii) the shared binary mask $\mathbf{M}$, which is computed once from the RGB detections and applied identically to both streams. We impose no explicit temporal or multi-view consistency constraints; cross-frame consistency instead emerges downstream from Hydra's TSDF volumetric fusion (Section 4.3).

3.1 Sensitive Instance Segmentation

The first stage identifies privacy-sensitive instances within the visual scene (Algorithm 1, line 3-10). Let $\mathbf{I}_{\text{rgb}} \in \mathbb{R}^{H \times W \times 3}$ denote the input RGB frame. We employ the YOLO11-Seg [10] large model with a 0.25 confidence threshold to predict a set of instance masks $\{m_i \,|\, i = 1, 2, \ldots, N\}$ where N presents number of detected instances and associated class labels $\{c_i\}$ (e.g., TV, laptop). We chose a relatively low confidence threshold to allow detection of displays in the scene (labeled as TV by YOLO11-Seg), which introduced mislabeling of other objects.

Let $\mathcal{S}$ be the set of pre-defined sensitive semantic classes. The binary occlusion mask $\mathbf{M}$ is constructed by the union of all pixel-wise predictions belonging to $\mathcal{S}$

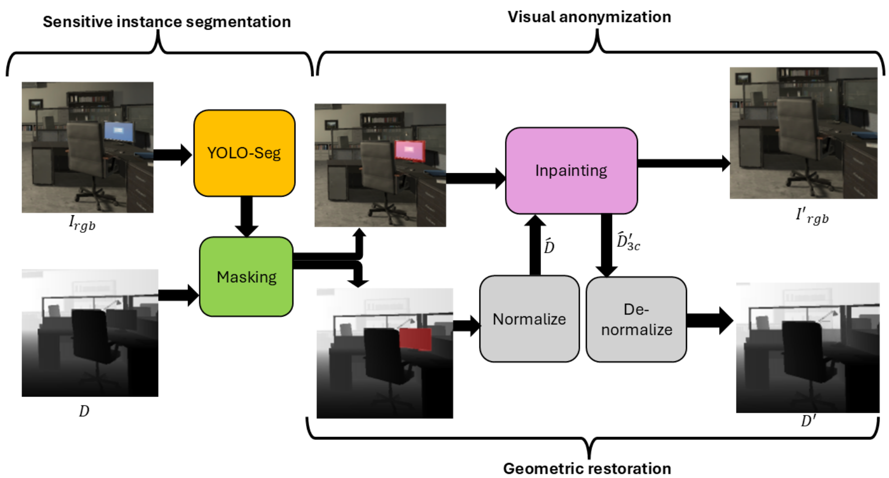


Fig. 2: Overview of our synchronized RGB-D inpainting pipeline. The original input images are cropped for visualization purposes.

and $(u, v) \in \{0, \ldots, W-1\} \times \{0, \ldots, H-1\}$:

$$\mathbf{M}(u, v) = \bigvee_{i \,|\, c_i \in \mathcal{S}} m_i(u, v). \tag{1}$$

After computing $\mathbf{M}$, we dilate the segmented regions with a kernel size of 5 to improve the blending of the masked regions with the surrounding environment during inpainting.

3.2 Visual Anonymization of RGB image

We utilize inpainting algorithms to generate an image where all the privacy-sensitive objects have been anonymized (Algorithm 1, line 11). The inpainted image $\mathbf{I}'_{rgb}$ is generated via

$$\mathbf{I}'_{\text{rgb}} = f_{\text{inpainting}}(\mathbf{I}_{\text{rgb}}, \mathbf{M}), \tag{2}$$

where the network blends background texture into the region defined by $\mathbf{M}$.

For inpainting models, we employ several state-of-the-art models based on generative adversarial networks (GAN) and convolutional neural networks (CNN) architectures: MIGAN[15], LaMa [17], FcF [9], ZITS [5], MAT [12]. A qualitative evaluation of their effect on the final 3DSG mesh reconstruction is provided in Section 4.

3.3 Geometric Restoration of depth image

A naive removal of objects from RGB data using black-box masking without addressing the depth channel results in geometric inconsistencies, where a "phan-

Algorithm 1 Synchronized RGB-D Inpainting

1: **Input:** RGB $\mathbf{I}_{rgb}$, Depth $\mathbf{D}$, Class Set $\mathcal{S}$
2: **Output:** Anonymized $\mathbf{I}'_{rgb}, \mathbf{D}'$
3: $results \leftarrow f_{\text{YOLO}}(\mathbf{I}_{rgb})$
4: $\mathbf{M} \leftarrow \mathbf{0}_{H\times W}$ ▷ initialize with blank mask
5: **for** $d \in results$ **do**
6:     **if** d.class $\in \mathcal{S}$ **then**
7:         $\mathbf{M} \leftarrow \mathbf{M} \vee d.\text{mask}$
8:     **end if**
9: **end for**
10: $\mathbf{M} \leftarrow \text{Dilate}(\mathbf{M}, \text{kernel})$
11: $\mathbf{I}'_{rgb} \leftarrow f_{\text{inpainting}}(\mathbf{I}_{rgb}, \mathbf{M})$ ▷ RGB Inpainting
12: $\hat{\mathbf{D}} \leftarrow \text{Normalize}(\mathbf{D})$ ▷ Eq. 3
13: $\hat{\mathbf{D}}'_{3c} \leftarrow f_{\text{inpainting}}(\text{Stack}(\hat{\mathbf{D}}), \mathbf{M})$ ▷ Depth Inpainting
14: $\mathbf{D}' \leftarrow \text{Denormalize}(\hat{\mathbf{D}}'_{3c})$ ▷ Eq. 5
15: **return** $\mathbf{I}'_{rgb}, \mathbf{D}'$

tom" object appears as a black region in the texture but remains visible in the geometry. To generate a consistent depth map with the inpainted image and to improve the robustness of our approach, we apply structural inpainting to the depth map (Algorithm 1, line 12-14). Since standard inpainting models are trained on 3-channel 8-bit natural images, we map the single-channel floating-point depth map $\mathbf{D} \in \mathbb{R}^{H\times W}$ into a compatible feature space.

Depth Normalization We first normalize the dynamic range of the current depth frame $[d_{\min}, d_{\max}]$ to the uint8 range $[0, 255]$:

$$\hat{\mathbf{D}} = \left\lfloor 255 \cdot \frac{\mathbf{D} - d_{\min}}{d_{\max} - d_{\min}} \right\rfloor. \tag{3}$$

The normalized map $\hat{\mathbf{D}}$ is then stacked to form a pseudo-3-channel tensor $\mathbf{D}_{3c}$ to match the input dimensions of the pre-trained inpainting model.

Structural Inpainting We leverage the observation that geometric discontinuities in depth maps often correlate with high-frequency edges in natural images. Consequently, we employ the RGB-trained inpainting model to inpaint the depth structure:

$$\hat{\mathbf{D}}'_{3c} = f_{\text{inpainting}}(\mathbf{D}_{3c}, \mathbf{M}). \tag{4}$$

This step effectively reconstructs the background geometry (e.g., a wall behind a TV) by generating smooth surfaces or edges consistent with the surrounding unmasked depth.

Denormalization The inpainted output is converted back to a single channel and projected to the original metric depth space:

$$\mathbf{D}' = \left(\frac{\hat{\mathbf{D}}'}{255} \cdot (d_{\max} - d_{\min})\right) + d_{\min}. \tag{5}$$

This yields the anonymized depth map $\mathbf{D}'$, ensuring geometric consistency with $\mathbf{I}'_{\text{rgb}}$.

3.4 System Implementation

The framework is implemented as a ROS node that employs an approximate time synchronizer to pair incoming $\mathbf{I}_{\text{rgb}}$ and $\mathbf{D}$ messages based on timestamp similarity. Although the pipeline includes a 'drop-if-busy' safeguard that would skip intermediate frames if the processing time T_{proc} were to exceed the sensor period, no frames were skipped in any of our experiments. The mechanism serves solely as a fail-safe to notify the user when the input stream rate should be adjusted to prevent potential overload. The computational complexity is dominated by the two inpainting passes, scaling as $\mathcal{O}(2f_{\text{inpainting}} + f_{\text{YOLO}})$, with inference accelerated using CUDA.

4 Experiments

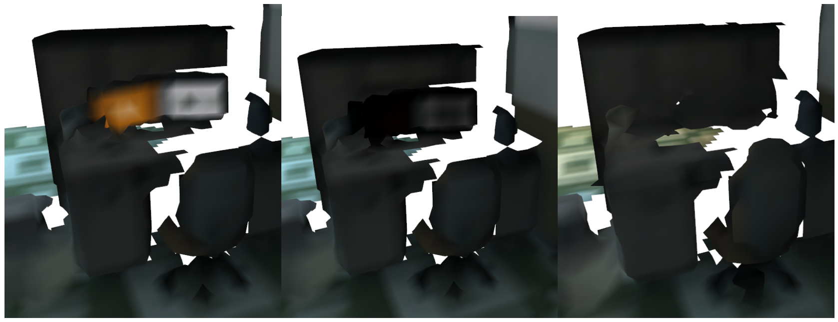
4.1 Experimental Setup

We evaluated our pipeline on the MIT SPARK uHumans2 dataset [14], a photorealistic Unity-based simulator providing synchronized RGB-D streams and semantic segmentation ground truth in an office environment. This choice provides a controlled digital-twin evaluation environment: synchronized RGB-D ground truth and semantic labels isolate the effect of the inpainting pipeline on 3DSG construction from confounders such as sensor noise and calibration drift, while sim-to-real transfer to physical RGB-D streams remains the natural next step in the deployment pipeline. The pipeline is implemented in ROS Noetic on a workstation equipped with an NVIDIA RTX3090 GPU. We primarily target detecting and removing privacy-sensitive objects commonly found in office settings: specifically, Class 62 (TVs/Monitors) from the COCO dataset [13]; however, this choice is arbitrary and could be replaced by any other class depending on the user's requirements.

4.2 Impact on 3D Scene Reconstruction

The primary limitation of prior work, such as PA-Hydra [1], is the creation of structural voids, such as dark regions in the mesh, where the object is masked. These voids not only degrade visual verisimilitude but can also cause topological errors in the scene graph.

As shown in Fig. 3, we compare three approaches:



(a) (b) (c)

Fig. 3: Impact of anonymization of sensitive objects with inpainting on 3D mapping. (a) Original reconstruction with privacy-sensitive object. (b) PA-Hydra's [1] reconstruction using black mask, (c) Our method's reconstruction inpainting RGB and depth

- **Original Map:** The standard reconstruction includes the privacy-sensitive monitor (Fig. 3a).
- **Naive Removal (PA-Hydra):** Masking the object in RGB results in a visually inconsistent region with dark artifacts and a faded planar surface that does not blend with the surrounding geometry naturally (Fig. 3b).
- **Geometric Restoration (Ours):** By inpainting the depth map prior to integration, our method reconstructs the wall behind the monitor. The resulting mesh is qualitatively watertight and continuous (Fig. 3c).

4.3 Qualitative Analysis: RGB-D Inpainting

Due to lack of ground truth inpainted images, we visually evaluated several state-of-the-art inpainting backbones using side-by-side comparison against reconstruction with unaltered images. Visual evaluation is based on artifact-free reconstruction, prioritizing geometric completeness and coherence (e.g., black patches in depth, non-smooth surface transition in RGB) with cross-modal RGB-depth consistency. The evaluated backbone models are MIGAN, LaMa, FcF, ZITS, and MAT. Fig. 4 illustrates the visual and geometric restoration results.

Visually, LaMa produced the most texture-consistent results across individual sequential frames. However, perfect per-frame consistency is not strictly necessary for high-quality 3D Scene Graph construction. While faster models like MIGAN occasionally produce artifacts (approximately 1 in 10 frames), the downstream TSDF volume integration used by Hydra acts as a temporal filter. The fusion process averages out transient inpainting errors, resulting in a stable and consistent final mesh as seen in the third column of Fig. 4. This observation allows us to prioritize low-latency models without compromising the final map quality.

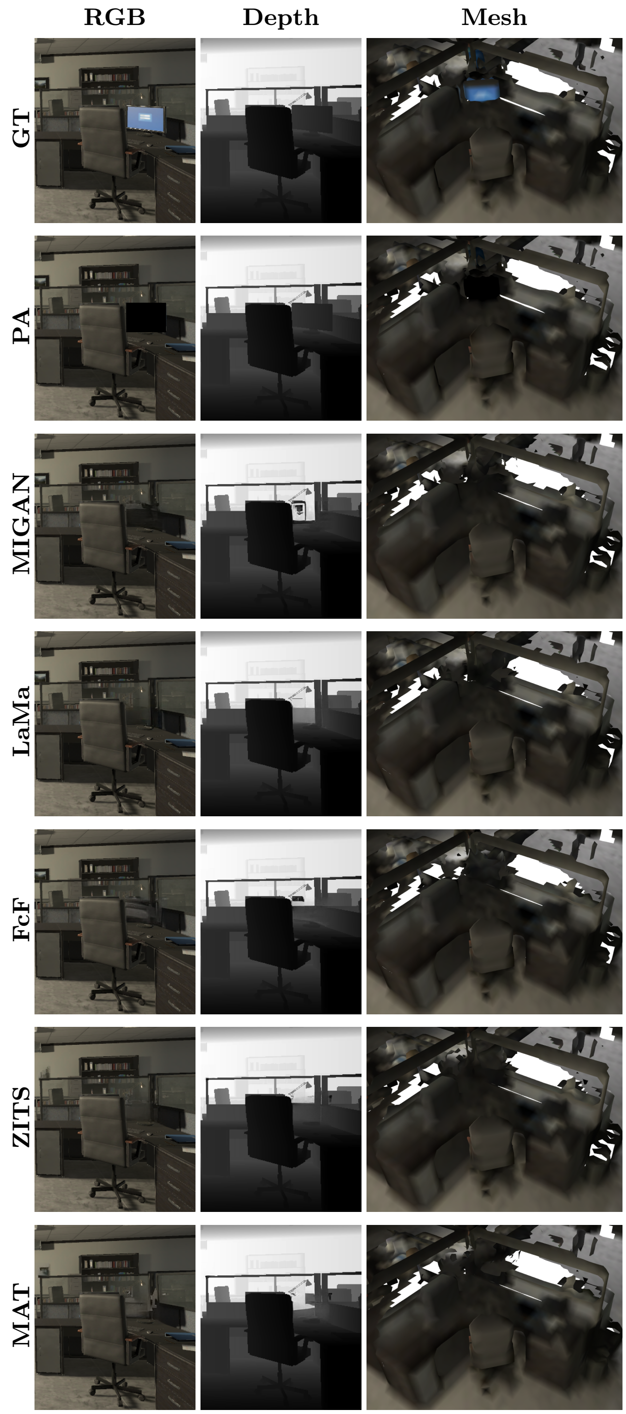


Fig. 4: Qualitative results comparing different inpainting backbones. While LaMa offers high texture consistency in both RGB and depth images, MIGAN provides the best balance of structure and speed, with mesh integration smoothing out minor artifacts.

4.4 Runtime Performance

Table 1: Runtime Analysis. (Left) Breakdown of latency per pipeline stage using MIGAN. (Right) Comparison of inference speeds for different inpainting backbones.

Pipeline Stage	**Average Time(ms)**
YOLO11l-Seg Inference	134.51
Masking	0.45
RGB Inpainting (MIGAN)	57.51
Depth Inpainting (MIGAN)	59.08
Total Latency	251.56

Model	**Inpainting Latency (ms)**
MIGAN[15]	**92** $\pm$ 15
LaMa[17]	509 $\pm$ 7
FcF[9]	305 $\pm$ 34
ZITS[5]	530 $\pm$ 17
MAT[12]	604 $\pm$ 13

To verify the real-time feasibility of our system for robotic platforms, we measured the average latency of each pipeline stage. As shown in Table 1, the total processing time per frame using MIGAN is approximately 250ms ($\sim$4 Hz).

While this is lower than the raw sensor rate, it is sufficient for dense mapping threads, which typically operate at lower frequencies (1–5 Hz). As shown in the comparison table 1, while models like ZITS and MAT provide high fidelity, their high latency (350–400 ms) renders them unsuitable for online operation. MIGAN emerged as the optimal choice, offering the fastest inference while maintaining sufficient visual quality. Although our benchmarks were collected on a desktop-class GPU, the drop-if-busy synchronizer ensures graceful degradation on slower embedded accelerators (e.g., Jetson Orin) when the inpainting stage cannot keep up with the sensor stream. Notably, detection rather than inpainting dominates the per-frame budget (134.5 ms of the 251.6 ms total, Table 1). This indicates that the highest-leverage path to higher throughput is detector-side optimization, such as a lighter YOLO11 variant (n/s), TensorRT export, or INT8 quantization. These are complementary to, and independent of, our low-latency inpainting choice. We therefore regard the reported $\sim$4 Hz as a conservative lower bound on the achievable throughput.

4.5 Sensitivity to Occlusion for Large Objects

We tested the system on varying object scales to assess robustness. Standard privacy targets like monitors represent medium-sized occlusions. To stress-test the pipeline, we targeted larger objects such as Office Chairs (Class 56 in Coco dataset [13]).

As shown in Fig. 5, large masks increase the difficulty for the generative model to inpaint consistent background geometry. However, despite the complex occlusion, the system successfully removes the bulk of the object volume. Although minor geometric smoothing occurs where the chair legs meet the floor, the semantic and physical traces of the object are effectively removed.

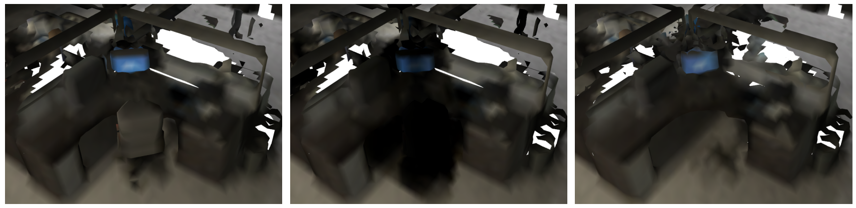


Fig. 5: Robustness to large occlusions. (Left) Ground Truth. (Middle) Naive masking of a chair using PA-Hydra approach leaves a large geometric void. (Right) Our method with MIGAN successfully removes the chair volume.

In Fig. 6, we demonstrate 3DSG construction using our method. In the 3DSG constructed from the ground truth data, a privacy sensitive object (computer screen) is detected and moved to the object layer (purple bounding box in cropped area in Fig. 6(a)). In contrast, when the 3DSG is constructed using our method from inpainted data, the privacy sensitive object is not detected and not moved to the object-layer.

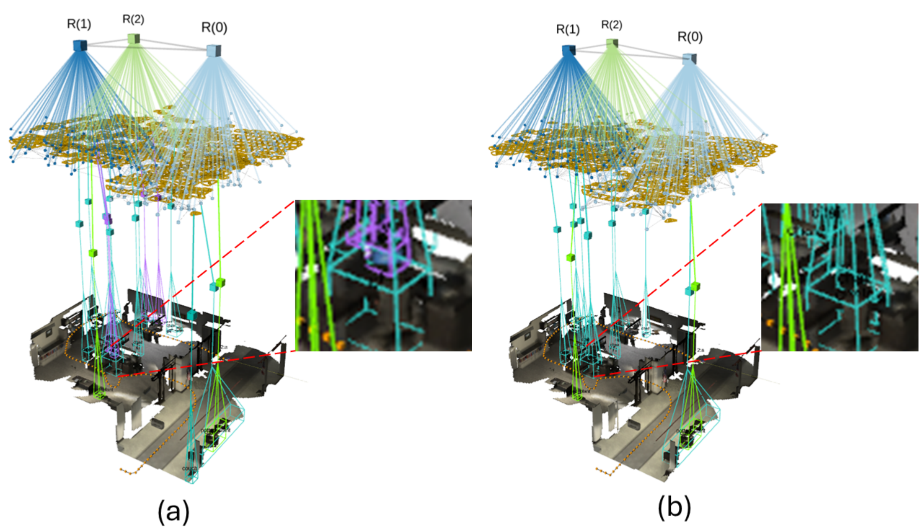


Fig. 6: (a) 3DSG constructed from ground truth data, in which a privacy-sensitive object (computer screen) is detected and moved to the object layer. (b) 3DSG constructed using our inpainted-data method. Cropped areas show the zoomed regions of the 3DSGs.

5 Analysis and Discussion

According to the experiments in Section 4, our proposed synchronized RGB-D inpainting approach produces results that blend naturally with the surrounding environment, reducing privacy-sensitive object leakage into the 3DSG relative to the prior approach, PA-Hydra (Fig. 3). In Fig. 4, we compared several state-of-the-art inpainting models to assess their effect on the final mesh construction. While LaMa produced the most visually consistent texture for both RGB and depth inpainting, temporal smoothing during 3DSG creation reduced the differences across sequential frames, yielding similar final meshes across methods. Among the evaluated models, MIGAN achieved the lowest runtime (Table 1), making it the most suitable choice for downstream tasks with real-time constraints such as robotic navigation. Using MIGAN, we further examined the sensitivity of our method to large objects. As shown in Fig. 5, although some traces of the chair remained, such as its legs, the method removed the bulk of the object and blended the generated texture more naturally with the surroundings than PA-Hydra.

A second design point concerns the choice of object detector. We used YOLO11-Seg with a confidence threshold of 0.25, which is lower than common settings. This was a deliberate, privacy-conservative choice: a false positive (inpainting a non-sensitive object) carries far lower cost than a false negative (leaving a sensitive object visible in the mesh). The trade-off is that whiteboards were occasionally inpainted, since YOLO11-Seg confused them with the TV class. Practical mitigations include class-specific confidence thresholds, restricting inpainting to detections that exceed a minimum mask-area threshold, and fine-tuning the detector on the deployment domain, none of which require changes to the pipeline itself.

Beyond detection, a further concern is whether the inpainter introduces visually plausible but geometrically or semantically false content into the graph. By design, the masked region is replaced with a plausible background rather than a faithful reconstruction of what was removed, so fidelity to the original content is intentionally traded for privacy. Three factors keep this trade contained. First, depth inpainting is constrained by the surrounding unmasked depth and tends to produce smooth background surfaces (Section 3.3). Second, the TSDF fusion in Hydra averages out transient per-frame artifacts. Third, the conservative masking threshold confines any fabricated geometry to the masked region. As shown in Fig. 6, this fabricated content does not propagate into spurious nodes: the non-sensitive layers of the 3DSG are preserved, and the only graph-level change is the removal of the sensitive object from the object layer. A systematic graph-level study of false-node generation is left to future work.

6 Conclusions

This work integrates a synchronized RGB-D inpainting method into Hydra [8], a state-of-the-art 3DSG framework, to strengthen privacy preservation during

3DSG construction. Prior black-box masking removes objects only in texture and leaves dark traces in the reconstructed mesh, which can support privacy attacks such as object linkage from the resulting 3DSG. Our method addresses this by inpainting the masked region in both the RGB and depth streams so that it blends realistically with the surrounding environment, removing those residual traces while keeping the rest of the scene graph intact.

Scope and future work. This study is intentionally scoped to a controlled, photorealistic setting (uHumans2), which isolates the effect of inpainting on 3DSG construction from confounders such as sensor noise and calibration drift. As the simulator provides no ground-truth mesh for the anonymized scene, our evaluation here is qualitative. A quantitative evaluation is the natural continuation of this work and can be carried out directly on the proposed pipeline. Inpainting fidelity can be measured with SSIM and FID for the RGB stream and with mean depth-reconstruction error against synthetic ground truth. Privacy can be assessed through the sensitive-object re-detection rate and object-inference success on the anonymized mesh. The effect on the scene graph itself can be quantified using node and relationship accuracy, graph edit distance against a clean graph, the rate of false nodes, and the preservation of non-sensitive spatial relations. Fig. 6 already offers preliminary evidence on the privacy and graph-level axes, since the sensitive object is neither detected nor promoted to the object layer after inpainting while the surrounding graph is preserved. Because the pipeline is class-agnostic, with the sensitive set $\mathcal{S}$ given as a configurable input, extending beyond monitors and chairs to richer environments containing laptops, documents, and mobile devices, together with transfer to consumer RGB-D sensors (e.g., RealSense, Kinect) under real noise and illumination, follows naturally without modifying the method.

References

1. Akin, E., Adewole, K.S., Caltenco, H., Malekian, R., Persson, J.A.: Privacy-aware Hydra (PA-Hydra) for 3D Scene Graph Construction. In: 2024 IEEE 10th World Forum on Internet of Things (WF-IoT). pp. 822–827 (Nov 2024). https://doi.org/10.1109/WF-IoT62078.2024.10811446, https://ieeexplore.ieee.org/document/10811446/, iSSN: 2768-1734
2. Armeni, I., He, Z.Y., Gwak, J., Zamir, A.R., Fischer, M., Malik, J., Savarese, S.: 3d scene graph: A structure for unified semantics, 3d space, and camera. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5664–5673 (2019)
3. Chelani, K., Kahl, F., Sattler, T.: How privacy-preserving are line clouds? recovering scene details from 3d lines. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15663–15673 (2021)
4. Chen, L., Wang, X., Lu, J., Lin, S., Wang, C., He, G.: Clip-driven open-vocabulary 3d scene graph generation via cross-modality contrastive learning. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
5. Dong, Q., Cao, C., Fu, Y.: Incremental transformer structure enhanced image inpainting with masking positional encoding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)

6. Ge, L., Zhu, X., Yang, Z., Li, X.: Dynamicgsg: Dynamic 3d gaussian scene graphs for environment adaptation. In: International Conference on Intelligent Robots and Systems (IROS) (2025)
7. Hu, Y., Zhou, X., Wang, C., Meng, W., He, G.: 3d scene graph generation with cross-modal alignment and adversarial learning. In: International Conference on Multimedia Retrieval (ICMR). p. 487–496 (2025)
8. Hughes, N., Chang, Y., Carlone, L.: Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. In: Robotics: Science and Systems XVIII. Robotics: Science and Systems Foundation (Jun 2022). https://doi.org/10.15607/RSS.2022.XVIII.050
9. Jain, J., Zhou, Y., Yu, N., Shi, H.: Keys to better image inpainting: Structure and texture go hand in hand. In: WACV (2023)
10. Jocher, G., Qiu, J.: Ultralytics yolo11 (2024), https://github.com/ultralytics/ultralytics
11. Kim, U.H., Park, J.M., Song, T.J., Kim, J.H.: 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. IEEE transactions on cybernetics **50**(12), 4921–4933 (2019)
12. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
14. Rosinol, A., Violette, A., Abate, M., Hughes, N., Chang, Y., Shi, J., Gupta, A., Carlone, L.: Kimera: from SLAM to spatial perception with 3D dynamic scene graphs. Intl. J. of Robotics Research **40**(12–14), 1510–1546 (2021)
15. Sargsyan, A., Navasardyan, S., Xu, X., Shi, H.: MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7301–7311. IEEE, Paris, France (Oct 2023). https://doi.org/10.1109/ICCV51070.2023.00674, https://ieeexplore.ieee.org/document/10377435/
16. Speciale, P., Schönberger, J.L., Kang, S.B., Sinha, S.N., Pollefeys, M.: Privacy preserving image-based localization (2019), https://arxiv.org/abs/1903.05572
17. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
18. Wang, X., Yang, D., Gao, Y., Yue, Y., Yang, Y., Fu, M.: Gaussiangraph: 3d gaussian-based scene graph generation for open-world scene understanding. In: International Conference on Intelligent Robots and Systems (IROS) (2025)
19. Yang, M., Huang, C., Huang, X., Hou, S.: Privacy-preserved visual simultaneous localization and mapping based on a dual-component approach. Applied Sciences **15**(5) (2025). https://doi.org/10.3390/app15052583, https://www.mdpi.com/2076-3417/15/5/2583
20. Zhang, B.: Using Sensors to Improve Path-planning Efficiency While Satisfying Privacy in 3DSG Scenarios. Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS) (2024)