2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

GenLaM: Generative Layered Mesh for Multi-modal Sensor Emulation in Robotics

Aakash Singh Bais, Akash Patel, Christoforos Kanellakis, George Nikolakopoulos

***Abstract*— Accurate environment perception is fundamental for robust robot navigation, mapping, and interaction. Traditional perception pipelines rely on multiple sensors, including stereo cameras and LiDAR, which impose constraints on cost, payload, and system integration. In this paper, we propose a novel single-image perception framework that unifies novel view synthesis and RGB/segmented LiDAR emulation into a single pipeline. Leveraging monocular depth estimation and camera intrinsics recovery, our approach projects image pixels into 3D space and performs mesh reconstruction to generate dense geometric representations. This enables high-fidelity sensor emulation, including transparent surface reconstruction such as glass - an element often missed by conventional LiDAR. By enriching synthetic LiDAR scans with otherwise unavailable geometry, our method enhances downstream tasks such as robot path planning and obstacle avoidance. This work opens up new possibilities for resource-efficient robotic perception by reducing sensor dependency while improving geometric reasoning.**

I. INTRODUCTION

Perception is a cornerstone of modern robotics, providing the foundation for navigation, manipulation, and interaction in unstructured environments. Mobile robots, autonomous vehicles, and service robots often rely on multi-sensor fusion, combining RGB cameras with LiDAR to achieve robust scene understanding [1], [2]. While effective, this dependency on multiple high-cost sensors introduces challenges in terms of system design, payload capacity, and failure modes. A key research question is whether a single RGB image can be leveraged to emulate multi-modal sensing capabilities, thus enabling lighter, cheaper, and more flexible robotic systems.

Recent advances in monocular depth estimation [3], [4] and camera self-calibration [5] provide a foundation for recovering 3D scene structure from a single image. When combined with geometric projection and mesh reconstruction, these methods enable a dense and structured 3D representation that can be used for novel view synthesis [6], [7] and sensor modality emulation. Unlike traditional depth-only pipelines, our approach integrates mesh-based reconstruction and explicit modeling of transparent surfaces such as glass, a long-standing challenge for LiDAR-based perception [8]. Transparent surfaces are particularly problematic in robotics since their absence from point clouds can lead to unsafe path planning or collisions.

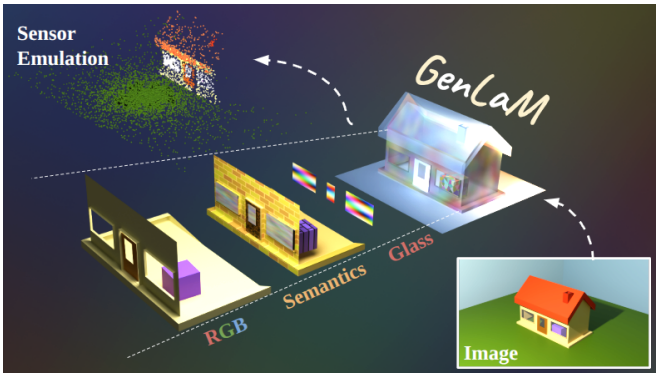

Fig. 1. GenLaM

In this paper, we introduce a pipeline that begins with monocular depth estimation and camera intrinsics recovery from a single RGB image. The estimated depths are projected into 3D space, forming a point cloud which is subsequently meshed to generate a continuous 3D surface model. From this mesh, we emulate LiDAR scans including segmented LiDAR point clouds as well as generate novel RGB views from arbitrary camera poses. A key contribution of our work is the ability to explicitly reconstruct and project transparent surfaces, enriching the emulated LiDAR scans with critical geometric cues absent in conventional data.

The contributions of this work are fourfold:

1) A unified single-image perception pipeline for simultaneous novel view synthesis and RGB/segmented LiDAR emulation. GenLaM itself is extendable and can be adapted for generating richer mesh representations from a single image in order to simulate various other sensors and simulating electro-magnetic / audio wave propagation.
2) Enhanced depth estimation specifically for regions behind the transparent glass surfaces to get a holistic 3D representation of the scene.
3) A mesh-based reconstruction framework that incorporates flat transparent surfaces like windows, doors and glass panes for enhanced safety in path planning.
4) Finally, we test the pipeline for reduced visibility scenarios - low light conditions, rain, fog and snow.

Our approach achieves both higher-quality reconstructions and improved navigational performance, suggesting a new direction for sensor-efficient robotics perception.

All authors are with the Robotics and AI Group, Luleå University of Technology, Luleå, Sweden.
aakash.singh.bais@ltu.se, akash.patel@ltu.se, christoforos.kanellakis@ltu.se, geonik@ltu.se

This work was supported by the European Union's Horizon Europe Research and Innovation Program, under the Grant Agreement No. 101138330, CIRCULess.

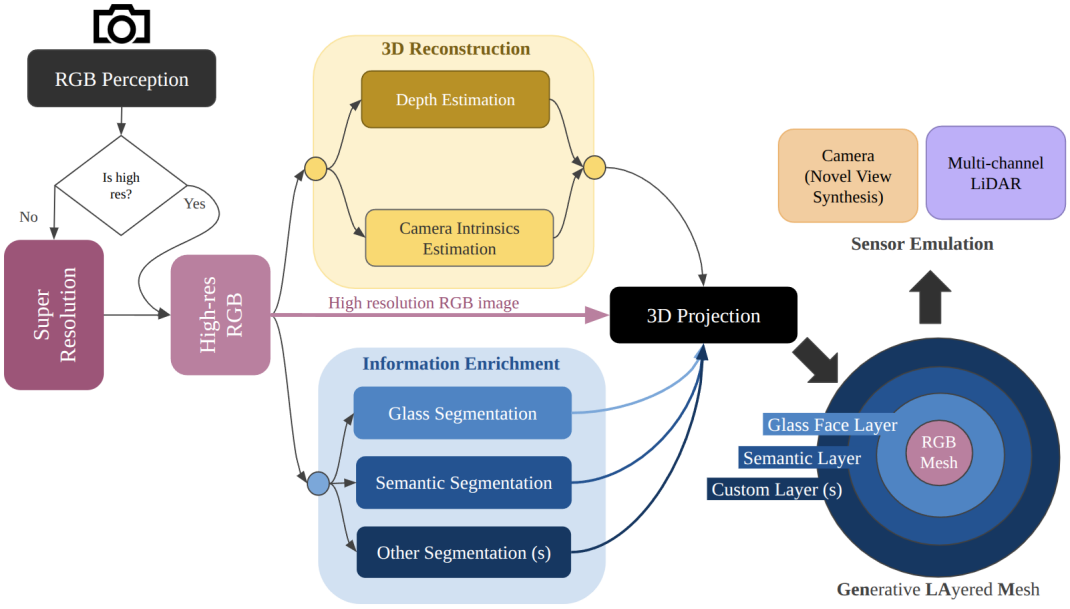

Fig. 2. Overview of proposed framework

II. RELATED WORKS

Robotic perception has historically relied on the integration of multiple sensing modalities to achieve robustness in diverse environments. LiDAR has been a cornerstone for mapping and navigation due to its high accuracy depth sensing [1], [2]. However, LiDAR struggles with transparent or reflective materials [8], leading to incomplete representations that can compromise navigation safety. Complementary use of RGB sensors helps fill semantic gaps [9], but fusion often increases system complexity.

Monocular depth estimation has made significant progress with deep learning methods [3], [4], [10], [11], [12], [13], [14], enabling accurate dense depth predictions from a single image. These approaches have been widely explored in robotics for low-cost mapping and SLAM, particularly in resource-constrained platforms such as UAVs and micro-robots [15]. Recent monocular depth estimation architectures like MoGe [14] are able to provide accurate metric depth estimations which are usable for robotics applications.

In parallel, novel view synthesis has become a vibrant research direction with neural radiance fields (NeRFs) [6], gaussian splatting [16] and related methods [7]. While most work targets computer vision applications such as free-viewpoint rendering, their potential for robotics has been explored in the context of active perception, SLAM, and digital twin construction [17], [18]. Our work differs in that we leverage single-view synthesis not primarily for visual realism or global mapping but for multi-modal sensor emulation for safe navigation.

Sensor emulation and augmentation have also attracted attention in robotics. Works on depth completion [19] and LiDAR simulation [20] have sought to enhance perception pipelines by filling missing or sparse depth data. Transparent object reconstruction [8] has highlighted the importance of addressing materials often invisible to LiDAR. Our pipeline builds on this line of research by explicitly incorporating transparent surface reconstruction into synthetic LiDAR emulation, thereby addressing a critical safety gap.

In summary, while prior work in monocular depth estimation, novel view synthesis, and LiDAR augmentation has shown promise individually, our contribution lies in unifying these advances into a single-image pipeline tailored for robotics. By bridging vision-based reconstruction with sensor emulation, we provide a resource-efficient framework that directly benefits robot navigation and path planning.

III. OUR APPROACH

We try to solve the problem of generating a reliable and information-rich representation of the environment with least amount of sensors and sensor data. To enable this, we develop a pipeline that is able to generate such a rich representation of a large part of the immediate environment of the robot with just a single RGB image. While developing the pipeline, attention is paid to making the representation especially useful for path planning in robotics. Our pipeline consists of three sequential stages. The first involves 3D reconstruction from the RGB image with special attention to transparent surfaces and the objects behind such surfaces. The second stage is enrichment of the data to get the generative layered mesh (GenLaM). We superimpose useful layers of information on the reconstructed mesh like glass-mask and semantic-mask. This is followed by sensor emulation. Sensor emulation allows our pipeline to be hot-pluggable into existing systems. It can feed rich, simulated sensor data to path planners that would otherwise depend on real sensor data. Figure 2 shows a schematic of our pipeline. Figure 3 and Figure 4 depict a practical overview of the pipeline and the sensor emulation respectively.

The following sections explain each of the stages of our pipeline in detail.

A. 3D Reconstruction

3D reconstruction from RGB camera images usually requires monocular depth estimation for the RGB image followed by projection of the depth and RGB information to a mesh using estimates of camera intrinsics like field of view (FOV) and focal length [14], [21]. This may be followed by post-processing to correct the mesh in occluded areas. Various architectures exist for monocular depth estimation [14], [21], [10], [11], [12], [22], some of which also generate the final mesh.

For 3D reconstruction, we have chosen MoGe [14] since it is a recent, state-of-art architecture and directly provides very accurate 3D reconstructions of scenes. Compared to the other available reconstruction methods, in addition to being accurate, this method has the advantage that it estimates camera FOV and takes it into account to directly render the meshes. It also omits areas of the mesh where the depth information is unreliable. The architecture uses ViT encoder to encode an RGB image followed by a convolutional decoder predicting an affine-invariant point map P and a mask to exclude areas with indefinite geometry (like sky). This is followed by a post-processing step to infer the depth, camera-shift and focal length. For an image $I \in \mathbb{R}^{H \times W \times 3}$, the model M predicts the 3D coordinates (x_i, y_i, z_i) of the image pixels (u_i, v_i) using P which is agnostic to global scale ($s \in \mathbb{R}$) and offset ($t \in \mathbb{R}^3$). Minimization of the following projection error helps to predict the focal length (f) and Z-axis shift ($t'_z = t_z/s$):

$$\min_{f,t'_z} \sum_{i=1}^{N} \left(\frac{fx_i}{z_i + t'_z} - u_i \right)^2 + \left(\frac{fy_i}{z_i + t'_z} - v_i \right)^2 \quad (1)$$

The network is trained with supervisions coming from a loss function which aggregates a global point-map loss (L_G), multi-scale local geometry losses (L_{S1}, L_{S2}, L_{S3}), normal loss (L_N) and mask loss (L_M). Here, $L_G = \sum_{i \in \eta} \frac{1}{z_i} ||s\hat{p}_i + t - p_i||_1$, where η is set of all points in the predicted point-map with definite geometry. For calculating multiscale loss at a point p_j, the set of points centered around p_j as $\Phi_j = \{i \mid ||p_i - p_j|| \leq r_j, i \in \eta\}$ are defined where $r_j = \alpha . z_j . \frac{D}{2.f}$ is the radius of the sphere and z_j is the z-coordinate of the point p_j, f is the ground truth focal length and D is the diagonal length of the image. Additionally, normal loss (L_N) is used to supervise the normals computed from the point-map predictions with respect to ground truth normals, such that $L_N = \sum_{i \in \eta} \angle(\hat{n}_i, n_i)$ where $\hat{n}_i$ and n_i are predicted and ground truth normals at pixel i and $\angle$ measures the difference in angles. The combined loss term becomes the following:

$$L = \lambda_G L_G + (\lambda_{S1} L_{S1} + \lambda_{S2} L_{S2} + \lambda_{S3} L_{S3}) + \lambda_N L_N + \lambda_M L_M \quad (2)$$

The above combined loss function helps supervise the architecture well for accurate, metric depth estimation and 3D reconstruction. For our use-case, we use the pretrained ViT-L based model having 331 million parameters for depth and normal estimation.

Effect of image quality - the need for super resolution: Since the global and local scales are predicted from the image, it is also essential to analyze the effect of input image quality on the 3D reconstructions. A well-trained model can understand the relative depths between points in the image. These can be biased by image resolution of the training set. But for use in robotics, the inferred global depth scale needs to be grounded in reality irrespective of resolution. While lower quality images are very fast to process and run inferences on, higher quality images can generate high-quality reconstructions. We compared the 3D reconstructions obtained from different resolution versions of the same image. These are shown in Figure 5. It can be seen that 3D reconstructions from images below a 1 mega-pixel resolution tend to have a variable global scale. On the other hand increasing the resolution beyond 1 mega-pixel has no visible improvement in scale. It can also be observed that each of the 3D reconstructions tend to lose scale at points farther away from the position of the camera. As a result the reconstructed tunnel tapers, while the ground truth point cloud shows the tunnel to be equally wide throughout.

Super-resolution architectures exist which aim to upscale low-resolution images using diffusion models [23], generative adversarial networks [24], transformers [25] and other techniques. We conducted experiments to upscale lower quality images using super-resolution architectures and then using them for 3D reconstruction. We found that super-resolving the images had the same stabilizing impact on the predicted depth from monocular depth estimation as a high-resolution camera image has. Hence, we incorporated Real-ESRGAN [24] based super-resolution of low-res images as part of our framework. We have chosen this architecture because it was fast enough and produced good-quality inferences. In theory any super-resolution architecture should work well in place of Real-ESRGAN. But care should be taken to not simply increase the resolution by replicating the pixels, as this does not improve the depth inference.

B. Data Enrichment

Depth perception for flat transparent surfaces: Transparent glass surfaces are ubiquitous in our daily lives. These could be glass windows, or doors or other transparent surfaces in general. LiDAR and depth sensors often fail to infer the depth of transparent glass surfaces. LiDAR scans simply do not respect transparent surfaces as the laser passes through them, and the surface is not represented by any LiDAR points. A monocular depth perception architecture like the one we have used (MoGe) can also get confused on how to treat transparent glass surfaces - whether to project them as flat or to ignore them and project the objects visible through them. Proper representation of glass or other transparent surfaces in sensor data is important for safe path planning and

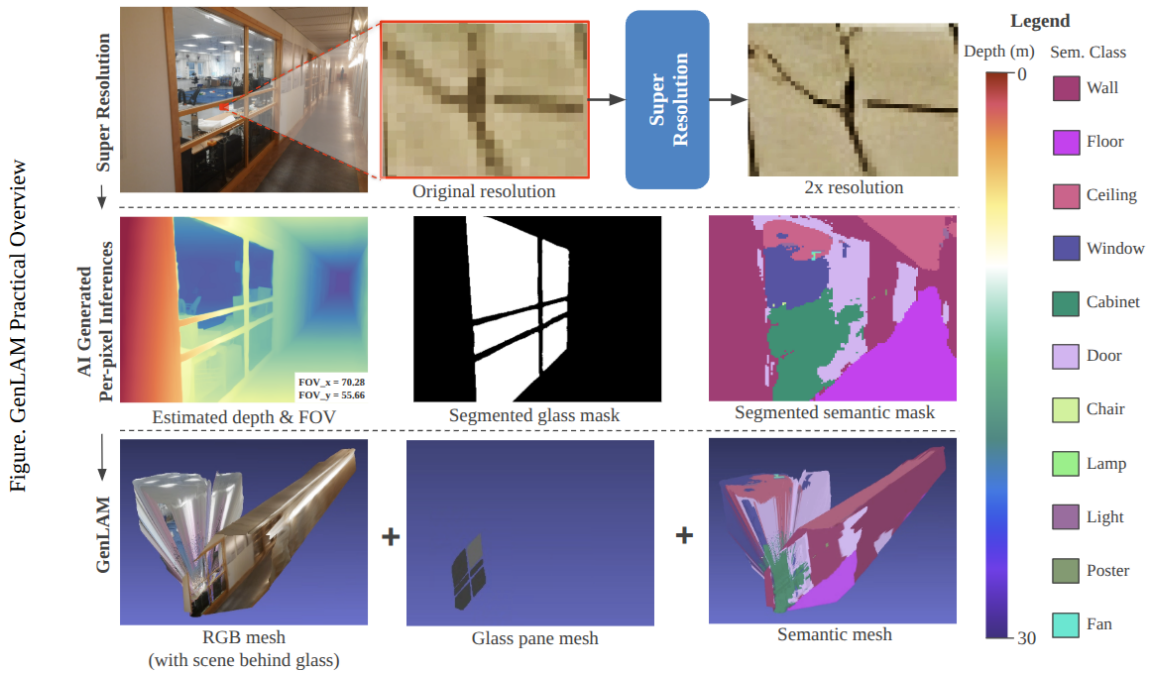

Fig. 3. Practical overview of GenLaM

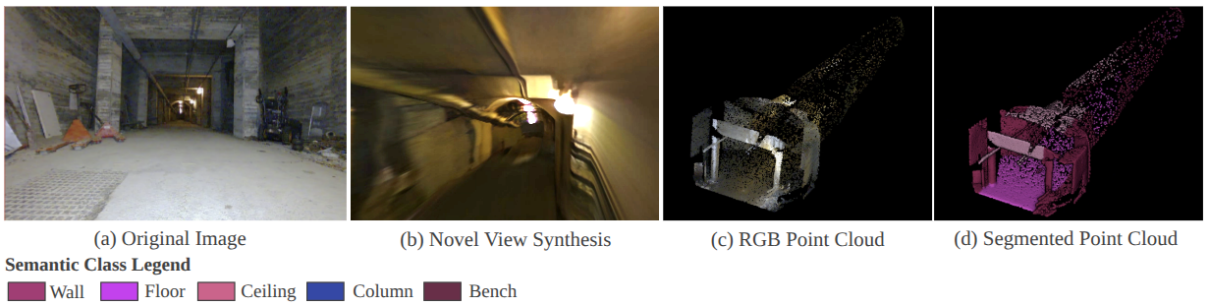

Fig. 4. Sensor emulation using GenLaM

robot navigation. Without them, the robot can unintentionally plan a path through the glass and end up breaking it. Solving this problem using our framework becomes quite straightforward. Glass segmentation neural networks [26], [27], [28] exist which specialize in segmenting glass points in input images. We use the GlassSemNet [28] for this job, but any alternate efficient architecture can be used for this purpose. We obtain the segmentation mask from GlassSemNet. To extract flat glass panes, we back-project depth-estimated pixels using the camera intrinsics derived from the field-of-view predicted from MoGe, segment connected regions via the mask, and fit planes from perimeter points before triangulating them into convex polygonal meshes. (Algorithm 1) gives detailed pseudo-code for this process.

While reliably adding glass surfaces is important, it is also important not to neglect the information about the objects visible through the transparent surface. As mentioned, depth estimation architectures do not treat transparent surfaces and objects visible through them reliably. Hence, we use a strategy to project such objects placed behind the glass surface using a trick shown in Figure 6. The trick involves predicting the depth and normal map for the full image that has glass. Then, we force MoGe depth estimator (M) to focus only on the region visible (V_R) through the glass, while the rest of the scene is provided as a normal map (N) to represent the perspective. This makes M to use the scale from the visible area and the perspective from N to be able to predict the correct combined depth and normal map for the full image.

Overlaying semantic information: In addition to information about transparent surfaces, we add information about object semantics to the mesh. We do this by first performing semantic segmentation on the original image. We employed the SegFormer-B4 model (47M parameters), a transformer-CNN hybrid architecture designed for efficient semantic segmentation. A pretrained checkpoint fine-tuned on the ADE20k dataset (150 classes, ∼20k training images) was used via HuggingFace's `nvidia/segformer-b4-finetuned-ade-512-512`. Given an input RGB image, the model predicts dense per-

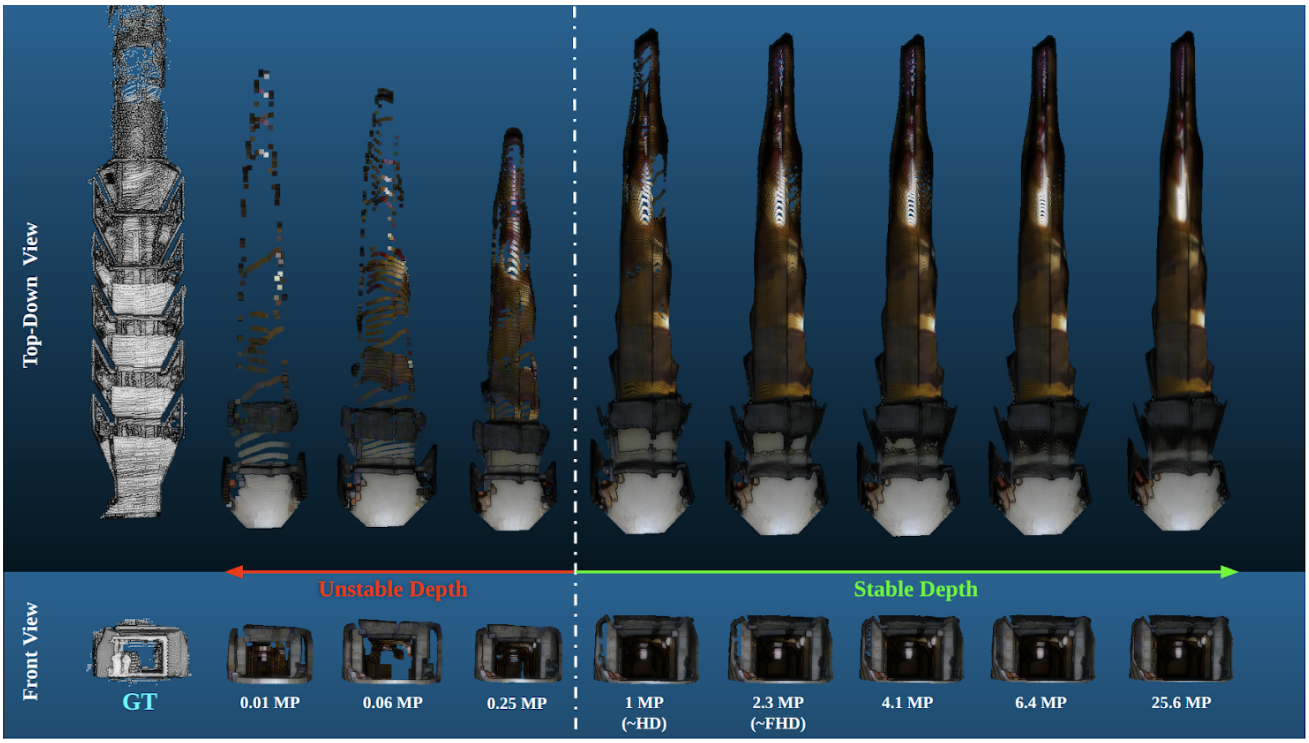

Fig. 5. Effect of image resolution on 3D reconstruction

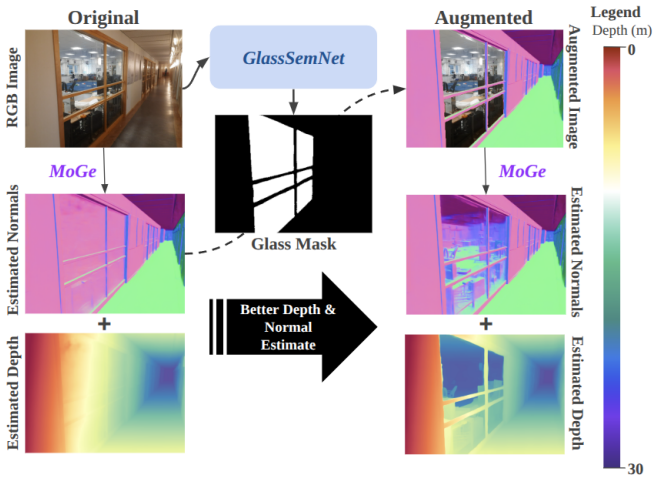

Fig. 6. Depth perception for flat transparent surfaces

pixel class labels which are subsequently mapped to a fixed colormap. This is then projected to the faces of the mesh using the pre-estimated depth and camera instrinsics from MoGe.

C. CUDA Accelerated Sensor Emulation:

Once the 3D reconstruction is performed on an RGB image captured by the agent, we get a realistic colored 3D mesh of the scene. Next, we emulate the LiDAR sensor within the reconstructed 3D mesh. For sensor emulation, we first localize the robot with respect to the reconstructed mesh. This gives us the location of the agent's LiDAR sensor with respect to the mesh. We emulate a LiDAR sensor at this location of the mesh. To generate LiDAR data, we use GPU-based ray-casting. We define a bunch of rays starting at the virtual LiDAR sensor and determine where each of them intersect the mesh for the first time, if they do. For each ray we check intersection of the ray with each of the triangular faces of the mesh. For a ray and a triangular face, the intersection is determined using the Möller–Trumbore intersection algorithm [29]. To simulate colored LiDAR scans, we uniformly sample rays on the unit sphere and perform parallel ray-mesh intersections on the GPU. Each hit point is assigned the mean color of its intersected face, producing a colored point cloud (Algorithm 2). This sensor emulation allows our pipeline to be pluggable in any existing robotic system where the real LiDAR data feed can be replaced by the synthetic LiDAR feed generated by our pipeline.

IV. EXPERIMENTAL RESULTS

In the following sections we first discuss the hardware and processing requirements. We also show the results of testing our pipeline in important edge-case scenarios to prove the robustness of the proposed pipeline.

A. Hardware, processing time and memory requirements

For our experiments, we use NVIDIA RTX 4090 GPU, with 24GB VRAM for running the monocular depth estimation, 3D reconstruction and sensor emulation. Even though we used a high-end GPU, the VRAM requirements of the algorithms are not that high. We found that the generation of GenLaM took about 1 second when run without super-resolution and about 4 minutes when run

Algorithm 1: Glass Pane Generation from Depth and Mask

Input: $P \in \mathbb{R}^{H\times W\times 3}$: per-pixel 3D points,
$M \in \{0,1\}^{H\times W}$: binary mask of glass regions,
f_x, f_y: horizontal and vertical FOV (radians),
τ: minimum perimeter points,
$\mathcal{O}$: output path.

Output: Polygonal mesh $\mathcal{G}$ of glass panes saved at $\mathcal{O}$.

Camera intrinsics:

$c_x \leftarrow (W-1)/2, \quad c_y \leftarrow (H-1)/2$
$f'_x \leftarrow \frac{W}{2}/\tan(f_x/2), \quad f'_y \leftarrow \frac{H}{2}/\tan(f_y/2)$

Find connected components:

$\{M_\ell\}_{\ell=1}^{L} \leftarrow \textsc{ConnectedComponents}(M)$

for $\ell = 1$ **to** L **do**

Extract perimeter contour $\mathcal{C}_\ell$ of M_ℓ;
Sample perimeter pixels $\{(x_i, y_i)\}$ from $\mathcal{C}_\ell$;
Collect 3D perimeter points $\mathcal{P}_\ell = \{P[y_i, x_i]\}$;
Filter invalid points; if $|\mathcal{P}_\ell| < \tau$, continue;
Plane fitting:
$\bar{p} \leftarrow \frac{1}{|\mathcal{P}_\ell|}\sum_{p\in\mathcal{P}_\ell} p$
$\hat{P} \leftarrow \{p - \bar{p} \mid p \in \mathcal{P}_\ell\}$
$[U, \Sigma, V^\top] \leftarrow \text{SVD}(\hat{P})$
$n \leftarrow V_{3,:}$ (plane normal), normalized, flipped if $n_z > 0$
$u \leftarrow V_{1,:}$, $v \leftarrow V_{2,:}$ (in-plane axes)
2D projection and polygon:
$Q \leftarrow \{[u^\top(p-\bar{p}), v^\top(p-\bar{p})] \mid p \in \mathcal{P}_\ell\}$
$H \leftarrow \textsc{ConvexHull}(Q)$
$V_\ell \leftarrow \{\bar{p} + q_x u + q_y v \mid (q_x, q_y) \in H\}$
Triangulate convex polygon:
$F_\ell \leftarrow \{(0, i, i+1) \mid i = 1, \ldots, |V_\ell| - 2\}$
Store mesh $\mathcal{M}_\ell = (V_\ell, F_\ell)$;

Combine and export:

$\mathcal{G} \leftarrow \textsc{Concatenate}(\{\mathcal{M}_\ell\}_{\ell=1}^{L})$
$\mathcal{G}.V \leftarrow \mathcal{G}.V \odot [1, -1, -1]$ // flip to OpenGL convention
$\textsc{Export}(\mathcal{G}, \mathcal{O})$

Algorithm 2: Colored LiDAR Simulation via CUDA Ray–Mesh Intersection

Input: Mesh $\mathcal{M} = (V, F, C)$ with vertices V, faces F, and face colors C; scan origin $p \in \mathbb{R}^3$; number of rays N

Output: Point cloud $\mathcal{P} = \{(x_i, c_i)\}$ with 3D points x_i and colors c_i

Sample N ray directions $\{d_i\}$ uniformly on the unit sphere;
for *each batch of rays* (p, d_i) **do**
Compute intersections $t_{i,f}$ with all faces $f \in F$ in parallel using CUDA;
Select nearest hit:
$f^* \leftarrow \arg\min_f \{t_{i,f} \mid t_{i,f} > 0\}$;
if *hit exists* **then**
$x_i \leftarrow p + t_{i,f^*} d_i$;
$c_i \leftarrow C_{f^*}$; `// mean vertex color of face`
Append (x_i, c_i) to $\mathcal{P}$;
return $\mathcal{P}$;

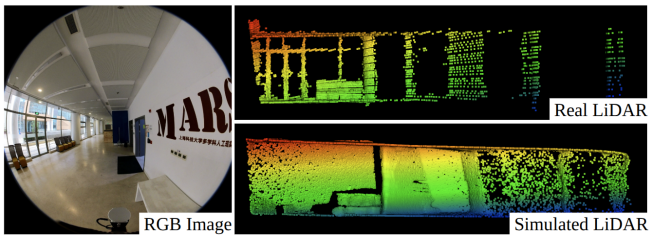

Fig. 7. Comparison between a LiDAR scan simulated using our pipeline and a real LiDAR scan.

with super-resolution. But once the GenLaM is generated, sensor emulation takes 0.1-0.2 seconds when not set to very high resolution settings. In this 3D reconstruction by MoGe takes about 0.7 seconds. The timings were found to be good enough for applications in robotics given that once the GenLaM is generated, many succeeding sensor-emulation cycles will not require a new GenLaM to be generated. The maximum GPU VRAM required for the 3D reconstruction and LiDAR emulation was found to be less than 10 GB for an HD image.

B. Sensor emulation with glass surfaces

We tested our pipeline with transparent glass surfaces. LiDAR sensors and most RGB-D sensors have trouble reliably measuring the depth of clear glass. But our hand-crafted pipeline allows us to detect glass and emulate synthetic LiDAR scans which respect glass surfaces. This can be very helpful for safe path planning in robotics. To test this, we took an image with glass windows and doors from the 3DRef benchmark dataset specially designed for glass reflection detection. We simulated a LiDAR scan using our pipeline and compared it with the real LiDAR scan in 3DRef. The results in Figure 7 show clearly how our pipeline detects points on transparent surfaces.

C. Performance in low visibility conditions

Camera-based perception can have issues in low visiblity conditions like poor lighting or rainy, foggy and snowy weather conditions. On the other hand, LiDAR is generally considered to be more robust. Since LiDAR is the most important sensor in robotics and GenLaM proposes to simulate LiDAR using only the image data. It is important to test if our pipeline can simulate clear LiDAR point clouds under such conditions. In Figure 4 we show the performance of GenLaM in low artificial lighting conditions of a tunnel, and we compare it with ground truth in Figure 5. We also tested our sensor emulation pipeline on street images taken

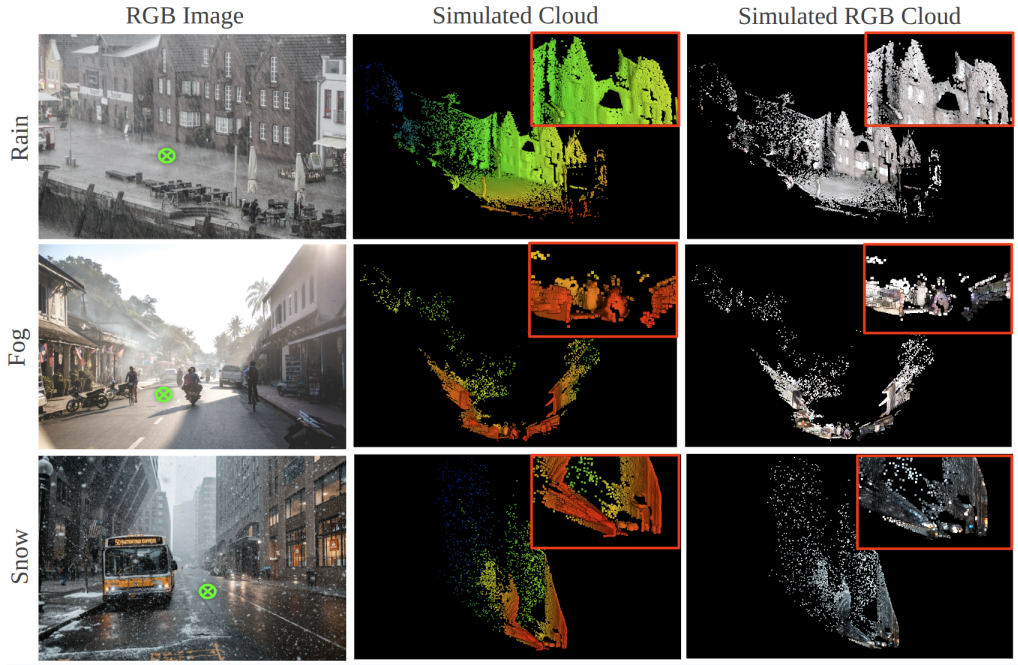

Fig. 8. Sensor emulation in low visibility conditions. The arbitrarily chosen location for LiDAR scan emulation is depicted as a green cross on the corresponding RGB images.

in rainy, foggy and snowy conditions. Then we simulated a mobile laser scan at an arbitrarily selected point on the road. These are visualized in Figure 8. Our pipeline can be seen to perform well enough even in harsh environmental conditions.

V. CONCLUSIONS

In this paper, we propose GenLaM - a framework to simulate synthetic multi-modal sensor data for a real environment with the least amount of real sensor data. We test our pipeline in various environments including low-visibility tunnels. We show that the pipeline robustly simulates sensor data using accurate 3D reconstruction and fast sensor emulation. We also show that our pipeline can solve the issue of safe navigation around transparent glass surfaces by detecting the surfaces, placing them in the GenLaM mesh and generating LiDAR data that includes points for glass surfaces. Our pipeline can also generate multichannel LiDAR directly without the need for computationally expensive 3D post-processing. The LiDAR data generated has both RGB and segmentation labels for every point. Our pipeline can be plugged into any path planner. The pipeline is also extendable and other layers of information can be added to the GenLaM to emulate various other sensors. This work intends to open a new frontier of research using generative AI-based simulated sensor data to enable efficient utilization of available information about the environment. It does come with its share of limitations which can be improved in future works.

A. *Limitations*

The foremost limitation of GenLaM is the speed at which new data can be meshed. This is mainly due to the time it takes for the 3D reconstruction. Secondly, GenLaM is dependent on GPU compute which may not be available on small, light-weight mobile robots aiming to navigate without remote compute nodes. Running the current pipeline on CPU does reduce the frequency at which new scenes can be reconstructed. Thirdly, since the 3D reconstruction is being performed generatively, the system requires some error allowances for its effective use. This can be improved to some extent by building more robust and specialized monocular depth estimation models. Another limitation of the current work is that the pipeline does not completely eliminate the use of sensors even after the reconstruction is generated. This is because even though a robot may use the mesh to perform path planning, but the robot may depend on the perception for localization.

B. *Future Research Directions*

The limitations of the current pipeline can be improved upon mostly by improving the monocular depth estimation model itself. The pipeline can be optimized with more efficient depth perception models that can run on CPU, detect glass surfaces and generate more accurate depth maps. A novel way to localize the robot within the mesh without the need for frequent real sensor data updates can help to further reduce the need of sensors after the reconstruction is generated. This research work also opens the direction for future research into data-efficient robotics.

ACKNOWLEDGMENT

We acknowledge the use of ChatGPT, for assisting in correcting grammar and syntax of this manuscript.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7747236

[2] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 4372–4378, iSSN: 1050-4729. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5509700

[3] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision Transformers for Dense Prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 12 159–12 168. [Online]. Available: https://ieeexplore.ieee.org/document/9711226/

[4] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/hash/91c56ce4a249fae5419b90cba831e303-Abstract.html

[5] O. Bogdan, V. Eckstein, F. Rameau, and J.-C. Bazin, "DeepCalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras," in *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production*, ser. CVMP '18. New York, NY, USA: Association for Computing Machinery, Dec. 2018, pp. 1–10. [Online]. Available: https://dl.acm.org/doi/10.1145/3278471.3278479

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Dec. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3503250

[7] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 257:1–257:15, Dec. 2018. [Online]. Available: https://dl.acm.org/doi/10.1145/3272127.3275084

[8] Y. Zhou, W. Peng, Z. Yang, H. Liu, and Y. Sun, "Transparent Object Depth Completion," May 2024, arXiv:2405.15299 [cs]. [Online]. Available: http://arxiv.org/abs/2405.15299

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364913491297

[10] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth Anything V2," *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, Dec. 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/26cfdcd8fe6fd75cc53e92963a656c58-Abstract-Conference.html

[11] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2024, pp. 10 371–10 381. [Online]. Available: https://ieeexplore.ieee.org/document/10657693/

[12] W. Hu, X. Gao, X. Li, S. Zhao, X. Cun, Y. Zhang, L. Quan, and Y. Shan, "DepthCrafter: Generating Consistent Long Depth Sequences for Open-world Videos." [Online]. Available: https://openaccess.thecvf.com/content/CVPR2025/papers/Hu_DepthCrafter_Generating_Consistent_Long_Depth_Sequences_for_Open-world_Videos_CVPR_2025_paper.pdf

[13] R. Birkl, D. Wofk, and M. Müller, "MiDaS v3.1 – A Model Zoo for Robust Monocular Relative Depth Estimation," Jul. 2023, arXiv:2307.14460 [cs]. [Online]. Available: http://arxiv.org/abs/2307.14460

[14] R. Wang, S. Xu, C. Dai, J. Xiang, Y. Deng, X. Tong, and J. Yang, "MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision."

[15] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-10605-2_54

[16] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3D Gaussian Splatting as a New Era: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 8, pp. 4429–4449, Aug. 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10521791

[17] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, Dec. 2016, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364916669237

[18] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6565–6574, iSSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/document/8100178

[19] F. Ma and S. Karaman, "Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 4796–4803, iSSN: 2577-087X. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8460184

[20] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," Jan. 2020, arXiv:2001.10773 [cs]. [Online]. Available: http://arxiv.org/abs/2001.10773

[21] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. V. Gool, and F. Yu, "UniDepth: Universal Monocular Metric Depth Estimation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2024, pp. 10 106–10 116. [Online]. Available: https://ieeexplore.ieee.org/document/10657139/

[22] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang, "Video Depth Anything: Consistent Depth Estimation for Super-Long Videos," Jun. 2025, arXiv:2501.12375 [cs]. [Online]. Available: http://arxiv.org/abs/2501.12375

[23] Y. Wang, W. Yang, X. Chen, Y. Wang, L. Guo, L.-P. Chau, Z. Liu, Y. Qiao, A. C. Kot, and B. Wen, "SinSR: Diffusion-Based Image Super-Resolution in a Single Step," Nov. 2023, arXiv:2311.14760 [cs]. [Online]. Available: http://arxiv.org/abs/2311.14760

[24] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data," Aug. 2021, arXiv:2107.10833 [eess]. [Online]. Available: http://arxiv.org/abs/2107.10833

[25] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image Restoration Using Swin Transformer," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Montreal, BC, Canada: IEEE, Oct. 2021, pp. 1833–1844. [Online]. Available: https://ieeexplore.ieee.org/document/9607618/

[26] J. Hao, M. Liu, and K. F. Hung, "GEM: Boost Simple Network for Glass Surface Segmentation via Segment Anything Model and Data Synthesis," Jan. 2024, arXiv:2401.15282 [cs]. [Online]. Available: http://arxiv.org/abs/2401.15282

[27] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting Transparent Objects in the Wild," Aug. 2020, arXiv:2003.13948 [cs]. [Online]. Available: http://arxiv.org/abs/2003.13948

[28] J. Lin, Y.-H. Yeung, and R. Lau, "Exploiting semantic relations for glass surface detection," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 22 490–22 504. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/8d162f48c816af5f8c114eb437e8b28b-Paper-Conference.pdf

[29] T. Möller and B. Trumbore, "Fast, Minimum Storage Ray-Triangle Intersection," *Journal of Graphics Tools*, vol. 2, no. 1, pp. 21–28, Jan. 1997. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10867651.1997.10487468