Building 3D Representations and Generating Motions From a Single Image via Video-Generation

Weiming Zhi [1,2,3] **Ziyong Ma** [3] **Tianyi Zhang** [3,4]

Matthew Johnson-Roberson [2,3]
[1] School of Computer Science, The University of Sydney, Australia.
[2] College of Connected Computing, Vanderbilt University, TN, USA.
[3] Robotics Institute, Carnegie Mellon University, PA, USA.
[4] Aurora, USA.
Correspondence to W. Zhi (`Weiming.zhi@sydney.edu.au`).

Abstract

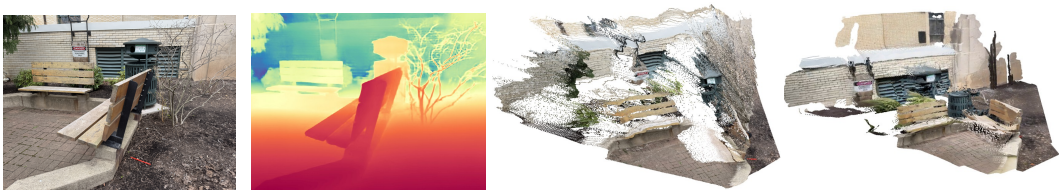
Autonomous robots typically need to construct representations of their surroundings and adapt their motions to the geometry of their environment. Here, we tackle the problem of constructing a policy model for collision-free motion generation, consistent with the environment, from a single input RGB image. Extracting 3D structures from a single image often involves monocular depth estimation. Developments in depth estimation have given rise to large pre-trained models such as *DepthAnything*. However, using outputs of these models for downstream motion generation is challenging due to frustum-shaped errors that arise. Instead, we propose a framework known as Video-Generation Environment Representation (VGER), which leverages the advances of large-scale video generation models to generate a moving camera video conditioned on the input image. Frames of this video, which form a multiview dataset, are then input into a pre-trained 3D foundation model to produce a dense point cloud. We then introduce a multi-scale noise approach to train an implicit representation of the environment structure and build a motion generation model that complies with the geometry of the representation. We extensively evaluate VGER over a diverse set of indoor and outdoor environments. We demonstrate its ability to produce smooth motions that account for the captured geometry of a scene, all from a single RGB input image.

1 Introduction

Autonomous robots operating in unstructured environments maintain a representation of their surroundings to enable reliable motion planning. Building complex environment representations generally requires large sets of images [1–3], and often depth measurements [4, 5]. In this work, we push the boundaries of data efficiency by constructing both a scene representation for collision avoidance and a downstream motion generation model from a single RGB image, by leveraging recent advances in pre-trained video generation and 3D foundation models.

A common strategy for recovering 3D structure from a single image is to estimate per-pixel depth. Recent advances have yielded powerful monocular depth estimators, most notably *DepthAnything*. However, the depth maps they produce often exhibit frustum-shaped artifacts at object boundaries, rendering them unsuitable for collision avoidance. To overcome these limitations, we introduce Video-generation Environment Representation (VGER), which sidesteps direct monocular depth prediction by conditioning a large, pre-trained video model on the input image to generate a se-

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Input image (b) DepthAnything-V2 image (l) and structure (r) (c) VGER

Figure 1: (a): Single original image used to construct the environment; (b): Predictions by DepthAnything-V2 [9]. Predicted depth image of the left, and extracted structure on the right. We observed errors that are frustum-shaped "tails" in the extracted structure; (c): Our proposed VGER does not suffer from these errors, and completes regions blocked from the original view. These artifacts in free space make it impossible to use the representation for motion generation.

quence of consistent novel views. We then apply a 3D foundation model such as DUSt3R [6] to fuse these views into a dense representation that faithfully captures scene geometry.

Beyond representation, our primary contribution lies in coupling the structure extracted from the single input image with a structured motion-generation framework. We integrate our reconstruction into motion generation by deriving an implicit unsigned distance field from the 3D foundation model's output. We do so efficiently via multi-scale perturbations. By sampling perturbations of point positions at varying noise scales and contrasting them against the dense output, we recover an implicit, unsigned distance field that defines a continuous distance function over the workspace. From this, we construct a smoothly varying metric field that encodes proximity to obstacles. Finally, we embed this environment-dependent metric field into a motion policy framework, in a similar manner to Riemannian Motion Policies [7] and Diffeomorphic Templates [8], in which the environment-induced metric modulates a nominal dynamical system. The result is a policy that generates motion and is modelled by a non-linear dynamical system that adapts fluidly to the reconstructed geometry, producing smooth, collision-free trajectories directly from a single input image.

Concretely, our technical contributions within VGER include,

1. A pipeline that harnesses pretrained video synthesis and 3D foundation models to generate multi-view videos from a single RGB image, producing dense, artifact-free reconstructions that overcome the limitations of monocular depth estimators;
2. A multi-scale noise-contrastive denoising approach to extract a globally detailed implicit unsigned distance field directly from the generated output;
3. A novel metric-modulated motion generation framework that embeds the implicit representation into a metric field that is dependent on the environment. This ensures that any nominal dynamical system coupled with the obstacle-induced curvature metric field will yield smooth, collision-free trajectories in real time.

2 Related Work

Pre-trained Models to Recover 3D Structure RGB Images: Estimating the structure within a single RGB image typically requires the estimation of depth, i.e. monocular depth estimation. Advances in deep learning have lead to large pre-trained models capable of predicting a depth image given an RGB input. Well-known models include Marigold [10], DepthAnything [11], and its follow-up model DepthAnything-V2 [9]. However, extracting structure from learned depth estimates can often lead to artifacts. Our proposed VGER differs in that it seeks to generate an image sequence from the single input image, and extract the structure that jointly appears.

Recovering 3D structure from a set of images is another well-known problem known as *structure-from-motion* [12]. Classical methods, such as COLMAP [13], produce relatively sparse representations. Modern advances in deep learning have led to transformer models capable of efficiently constructing highly dense 3D representations in a single feed-forward pass, including DUSt3R [6], MASt3R [14] and Visual Geometry Grounded Transformers (VGGT) [15]. These models typically act as *3D foundation models*, which are large pre-trained models, intended as plug-and-play modules

(a) Input Image (b) Frames of generated input-conditioned video (c) 3D structure

Figure 3: Examples of extracting a 3D structure of an outdoor bench (top) and indoor office environment (bottom) from input images. Input images are shown in subfigure (a). We leverage a video generator, conditional on the images, to generate videos, with frames shown in subfigure (b). These are then subsequently used to construct 3D structures via foundation models, without any frustum-shaped artifacts, shown in subfigure (c).

for downstream tasks. These 3D foundation models have found use in a variety of robot perception and calibration problems [16].

Video Generation Models: A recent wave of deep learning–based general video generators spans unconditional, text-to-video, and image-to-video synthesis. Early GAN-based methods (StyleGAN-V [17]) gave way to diffusion and transformer systems: Video Diffusion Models [18] produce high-quality clips, and MagicVideo [19] extends diffusion priors for text-driven generation. Many recent advances in this field have focused on the controllability of the camera motions in the video generator, these include GEN3C [20] and ViewCrafter[21], and SEVA [22]. Particularly, in this work, we leverage a recent video generator Stable Virtual Camera (Seva) [22], which can condition on an image as well as a camera trajectory to generate a video that follows the trajectory in the scene of the image.

Reactive Motion Generation: Our work is the first to apply structures and representations produced by 3D foundation models for reactive motion generation. Relevant frameworks, including Riemannian Motion Policies [7] and Geometric Fabrics [23, 24], to generate motion creatively generally approach the generation from a planning perspective and require the assumption of a pre-defined surface, or a completely known environment constructed from simple shape primitives. Other robotics focused representations may require depth [25] or velocity measurements [26]. There has been some effort to extend from representing environments to meshes instead of pre-designed primitives [27]. However, there is a clear disconnect between the literature on scene representation and motion generation. To bridge this gap, Our method is unique in extracting both a representation of the structure and a subsequent motion generation model from a single RGB image.

3 Video generation Environment Representation: Structure Extraction to Motion Generation From a Single Image

Here, we elaborate on the technical details of the VGER method. This includes: leveraging pre-trained video generators to extract a structure conditional on the input image (section 3.1); the construction of an implicit model via multi-scale noise contrastive samples (section 3.2); generating motion from an environment-dependent metric field (section 3.3), giving collision-free motion. An overview of the workflow is shown in fig. 2.

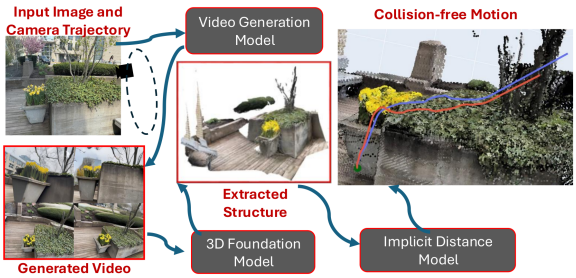


Figure 2: Pipeline of VGER.

3.1 Structure Extraction from Single Image with Pre-trained Video Generators

We circumvent the direct prediction of any depth information from the input image, thereby avoiding frustum-shaped artifacts that are characteristic. Instead, we leverage modern video generators to synthesize a sequence of images of the camera moving, conditional on the input image. These images are then passed into a 3D foundation model, such as DUSt3R [6], to produce a dense point cloud. We use the pre-trained video generation model, Stable Virtual Camera (Seva) [22]. Seva is a 1.3 billion parameter diffusion model, using a stable diffusion 2.1 backbone [28]. It produces a sequence of output images $\{I_1, I_2, \ldots, I_n\}$, conditional on the input image I_0 and a trajectory of camera poses $\{T_1, \ldots, T_n\}$, where n is a pre-determined image sequence length. Despite the video generator not constructing an explicit 3D representation, Seva is trained on videos that maintain both photometric and geometric consistency across frames. As a result, the generated image sequences maintain geometry and temporal consistency.

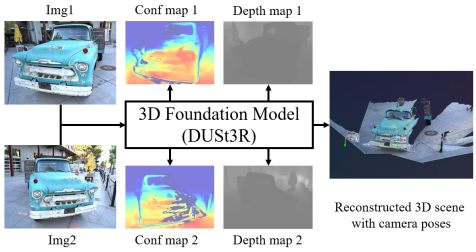


Figure 4: We leverage DUSt3R [6], which can produce 3D structures from sets of 2D images, and filter based on confidence maps.

The generated images, along with the conditioned input image, are then all inputted into the 3D foundation model, DUSt3R [6]. DUSt3R is a fully data-driven system built around a large vision transformer [29]. It produces dense 3D point-maps along with per-pixel confidence and depth estimates, and then uses these to recover relative camera poses and a fully consistent 3D reconstruction. Because it learns correspondences end-to-end rather than relying on hand-crafted features, it can accurately estimate pose even when the two views have very little visual overlap. After filtering away the points with low confidence, we recover a dataset of dense cloud of N 3D points $\{p_i, c_i\}_{i=1}^N$, where $p_i \in \mathbb{R}^3$ gives the 3D coordinates and c_i gives the RGB colors of the point. Next, we seek to construct a continuous implicit representation of the environment from the point cloud. This implicit model enables smooth geometry reconstruction and supports downstream tasks such as collision avoidance and motion generation.

3.2 Multi-scale Noise Contrastive Implicit Model

Reactive motions generated in the environment are influenced by distances to the nearest surface in the environment. Here, VGER converts the point-based output from the 3D foundation model into a smoothly varying function that captures the distance to the nearest surface in the scene. This implicit function is modelled by a neural network $f_\theta : \mathbb{R}^3 \to \mathbb{R}$ that approximates the unsigned distance to a surface captured in our point cloud $\mathcal{P} = \{p_i\}_{i=1}^N$. Our training procedure optimizes a loss:

$$\mathcal{L} = \mathcal{L}_{\text{fit}} + \alpha_{\text{surf}}\mathcal{L}_{\text{surf}} + \alpha_{\text{eik}}\mathcal{L}_{\text{eik}}. \tag{1}$$

This loss combines: (i) regression to ground-truth distances at query points, (ii) a surface consistency term enforcing a zero level-set at the surface, and (iii) an Eikonal regularizer to encourage unit gradient norm. Linear weightings are denoted as α_{surf} and α_{eik}. Next, we elaborate on this construct of each loss term.

Multi-Scale Noise Negative Querying: To train a model to recover distances both on and near the surface, we generate negative samples by perturbing points drawn from the surface with Gaussian noise at multiple scales. Concretely, at each iteration we first sample a batch of raw surface points $\{p_i\}_{i=1}^N$ uniformly from $\mathcal{P}$. Around each p_i, we add a zero-mean Gaussian noise ε_j whose standard deviation σ_j is itself drawn from a log-uniform distribution:

$$u_j \sim \text{Uniform}\big(\ln(\sigma_{\min}), \ln(\sigma_{\max})\big), \quad \sigma_j = \exp(u_j), \quad \varepsilon_j \sim \mathcal{N}(0, \sigma_j^2 I), \quad \hat{p}_{i,j} = p_i + \varepsilon_j.$$

Here, $\hat{p}_{i,j}$ denotes a *query point* that is perturbed off the surface point p_i with a noise ε_j, and $\sigma_{\min}$ and $\sigma_{\max}$ give boundaries of the standard deviation of the noise. Sampling σ_j on a log scale ensures that the network sees perturbations ranging from extremely fine to coarse (one-tenth of the bounding radius) distances. This multi-scale scheme serves two purposes: *Local surface fidelity:* Small σ_j focuses the loss on points extremely close to the true surface, improving pointwise accuracy and helping the zero level set align precisely with $\mathcal{P}$; *Robust gradient learning:* Larger σ_j forces the model to estimate distances further from the surface, ensuring that the predicted gradient field ∇f_θ

remains informative and prevents collapse to trivial solutions. Next, we compute the true unsigned distance efficiently via a KD-tree, and construct a mean squared error loss to learn a continuous distance representation:

$$d(\hat{p}) = \min_{p \in \mathcal{P}} \|\hat{p} - p\|, \qquad \mathcal{L}_{\text{fit}} = MSE(\, f_\theta(\hat{p}_{i,j}), d(\hat{p}_{i,j})).$$

The implicit model f_θ is represented by a neural network, that incorporates SIREN activation layers [30]. That is, each hidden layer applies a periodic activation:

$$h^{(l)} = \sin\big(\omega_0 \, (W^{(l)} \, h^{(l-1)} + b^{(l)})\big), \quad l = 1, \ldots, L.$$

Here, $h^{(l-1)}$ is the input to layer l, $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b^{(l)} \in \mathbb{R}^{n_l}$ are the learned weights and biases, and $\omega_0 > 0$ is a fixed frequency scale controlling the periodicity. By fitting parameters in the frequency domain, these activation layers enable f_θ to capture finer details with higher frequency.

Surface Consistency and Eikonal Regularization: To ensure that our representation has *a priori* structure of $f_\theta(p_i) = 0$ and $\|\nabla f_\theta(x_i)\| = 1$, we provide additional regularization. For each training iteration, let $\mathcal{B}_s$ denote the minibatch of surface points and $\mathcal{B}_q$ the full minibatch of query points. To align the network's zero level-set with the input surface, and to encourage $\|\nabla f_\theta(x)\| = 1$ almost everywhere. We add a surface loss $\mathcal{L}_{\text{surf}}$ and a Eikonal loss $\mathcal{L}_{\text{eik}}$, given by

$$\mathcal{L}_{\text{surf}} = \frac{1}{|\mathcal{B}_s|} \sum_{p \in \mathcal{B}_s} \big|f_\theta(p)\big|, \qquad \mathcal{L}_{\text{eik}} = \frac{1}{|\mathcal{B}_q|} \sum_{\hat{p} \in \mathcal{B}_q} \big(\|\nabla_x f_\theta(\hat{p})\|_2 - 1\big)^2. \tag{2}$$

Together, these terms ensure exact surface interpolation and a well-conditioned gradient field, which promotes accurate normal estimation and stable mesh extraction.

Connection to Diffusion Models: Multi-scale denoising has proven crucial for sharpening implicit energy-based representations by teaching models to recover fine details at varying noise levels. In particular, frameworks such as denoising score matching [31] and score-based diffusion models [32] denoise data corrupted at multiple noise scales to learn a noise-conditional score

$$s_\theta(x, \sigma) \approx \nabla_x \log p_\sigma(x), \tag{3}$$

where p_σ is the data distribution corrupted by Gaussian noise of scale σ. By sampling σ log-uniformly over several orders of magnitude, these methods capture both high-frequency surface details (small σ) and global structure (large σ). Analogously, the multi-scale noise in VGER trains the implicit distance field f_θ to denoise points at various distances from the surface, effectively learning a landscape whose gradient aligns with the true distance gradient. This mirrors the reverse diffusion process, where successive denoising steps refine a sample towards the data manifold. Our incorporation of an Eikonal term $\mathcal{L}_{eik}$ further ensures that $\|\nabla f_\theta\| = 1$, enforcing the learned field into a distance metric and yielding sharp, artifact-free reconstructions.

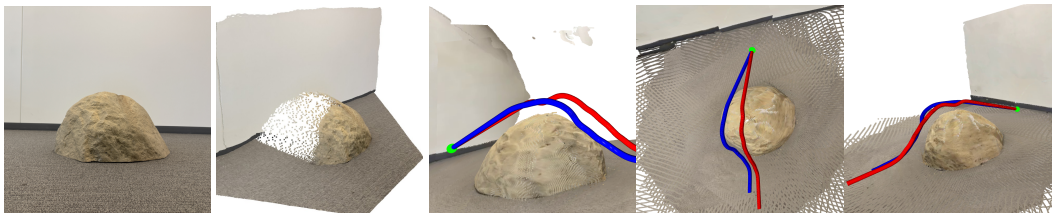
3.3 Motion Generation in Constructed Environment via Continuous Metric Fields

Generating motion is central to enabling robots to interact with their surroundings. Here we outline how we can leverage the environment representations extracted from a single image to shape robot motions. We model robot policies as a dynamical system that is influenced by an environment-dependent metric field.

Robot Policies as a Dynamical System: We take a reactive dynamical systems approach, enabling motion at arbitrary initial positions in the environment. Similar to many of the reactive methods introduced in [7, 8], we model the task-space motion of a robot as an autonomous dynamical system, where trajectories can be obtained via numerical integration,

$$\dot{\mathbf{x}}(t) = g(\mathbf{x}(t)), \qquad \mathbf{x}(t) = \mathbf{x}_0 + \int_0^t g(\mathbf{x}(s))ds, \tag{4}$$

where $\mathbf{x}(t)$ give the task-space coordinates of the motion, $g : \mathbb{R}^3 \to \mathbb{R}^3$ is the dynamics, $\mathbf{x}_0$ is the initial condition for the trajectory, integrated for time t. For robot manipulators, where the Jacobian of the robot forward kinematics is known, we can further pull the dynamical system into the robot's configuration space by multiplying with the pseudoinverse of the Jacobian, as is done in [8].



(a) input (b) DA-V2 (c) Our reconstructed 3D representation and trajectories

Figure 5: With a single example input image, shown in (a), of a stone model in an indoor environment, VGER can build a 3D representation of the scene. It does not suffer from incomplete surfaces like the results from DepthAnything-V2, shown in (b)). It can also facilitate downstream motion trajectory generation. Two trajectories, colored in red and blue, smoothly avoiding the obstacle, are shown in (c).

Environment Influence as a Metric Field: We propose a principled framework for shaping motion generation based on task-driven Riemannian metrics derived from the upstream distance function f_θ, such that the reactive behaviours like collision-avoidance and surface following emerge. Given an nominal base dynamical system $g_{base}(\mathbf{x})$, we define a metric-modulated motion as

$$\dot{\mathbf{x}} = G(\mathbf{x})^{-1} g_{base}(\mathbf{x}), \qquad \text{and} \quad G(\mathbf{x}) \in \mathbb{S}^{++}. \tag{5}$$

Here, g_{base} is the dynamics that governs the initial motion pattern without considering the environment. This could be defined as a simple attractor to a goal coordinate or a dynamical system learned from data [33, 34]. $G(\mathbf{x})$ is a Riemannian metric that is positive definite ($\mathbb{S}^{++}$) and varies smoothly throughout the environment, constructed from f_θ. This metric-based modulation can be interpreted as solving, at every point $\mathbf{x}$, a local quadratic program,

$$\dot{\mathbf{x}} = \arg\min_{\mathbf{v}\in\mathbb{R}^3} \left\{ \frac{1}{2}\mathbf{v}^\top G(\mathbf{x})\mathbf{v} - \mathbf{v}^\top g_{base}(\mathbf{x}) \right\}. \tag{6}$$

Thus, the metric shaping procedure can be viewed equivalently as solving a structured QP at every point in state space, where the metric $G(\mathbf{x})$ defines the local cost geometry. Importantly, by modulating $G(\mathbf{x})$, we influence both the preferred directions of motion and the associated effort, enabling smooth reactive collision avoidance.

Our goal is now to construct a Riemannian metric field to induce smooth collision avoidance behavior around surfaces represented by a learned unsigned distance function $f_\theta : \mathbb{R}^3 \to \mathbb{R}_{\geq 0}$, where $f_\theta(\mathbf{x})$ approximates the distance from a point $x \in \mathbb{R}^3$ to the closest point on the surface. To penalize motion toward the surface, we construct the collision avoidance metric as

$$G(\mathbf{x}) = I + f_{\text{blow}}(\mathbf{x})\, \mathbf{u}(\mathbf{x})\mathbf{u}(\mathbf{x})^\top, \tag{7}$$

where I is the 3×3 identity matrix. Here, define a blow-up scaling factor f_{blow} and denote a unit surface normal $u(\mathbf{x})$, where,

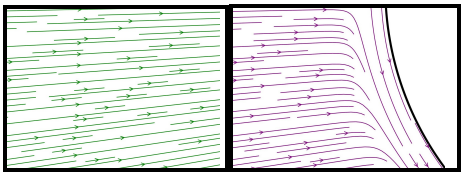


Figure 6: Constructed metric field stretches and warps a base dynamical system $g_{base}(\mathbf{x})$ with flows (left), to produce the flows $G(\mathbf{x})^{-1} g_{base}(\mathbf{x})$ which avoid colliding into the black surface (right).

$$f_{blow}(\mathbf{x}) = \frac{k}{(f_\theta(\mathbf{x}) + \epsilon)^4} \exp(-\beta f_\theta(\mathbf{x})), \qquad \mathbf{u}(\mathbf{x}) = \frac{\nabla f_\theta(\mathbf{x})}{\|\nabla f_\theta(\mathbf{x})\|}. \tag{8}$$

Here, $k > 0$ controls the penalty strength, $\epsilon > 0$ ensures numerical stability near the surface, and $\beta > 0$ induces exponential decay away from the surface. Direct inversion of the metric G at every control step would incur a full 3×3 matrix solve. Instead, we exploit the fact that G is a rank-one update of the identity and apply the Sherman–Morrison formula [35] to eq. (5):

$$\left(I + \alpha\, \mathbf{u}\,\mathbf{u}^\top\right)^{-1} = I - \frac{\alpha}{1+\alpha}\, \mathbf{u}\,\mathbf{u}^\top \iff \dot{\mathbf{x}} = \left[I - \frac{f_{\text{blow}}(\mathbf{x})}{1 + f_{\text{blow}}(\mathbf{x})} \mathbf{u}(\mathbf{x})\mathbf{u}(\mathbf{x})^\top \right] g_{\text{base}}(\mathbf{x}). \tag{9}$$

As the trajectory approaches a surface, that is $f_\theta(\mathbf{x}) \to 0$, the metric sharply penalizes motion aligned with the surface normal, naturally steering trajectories tangentially to the surface without

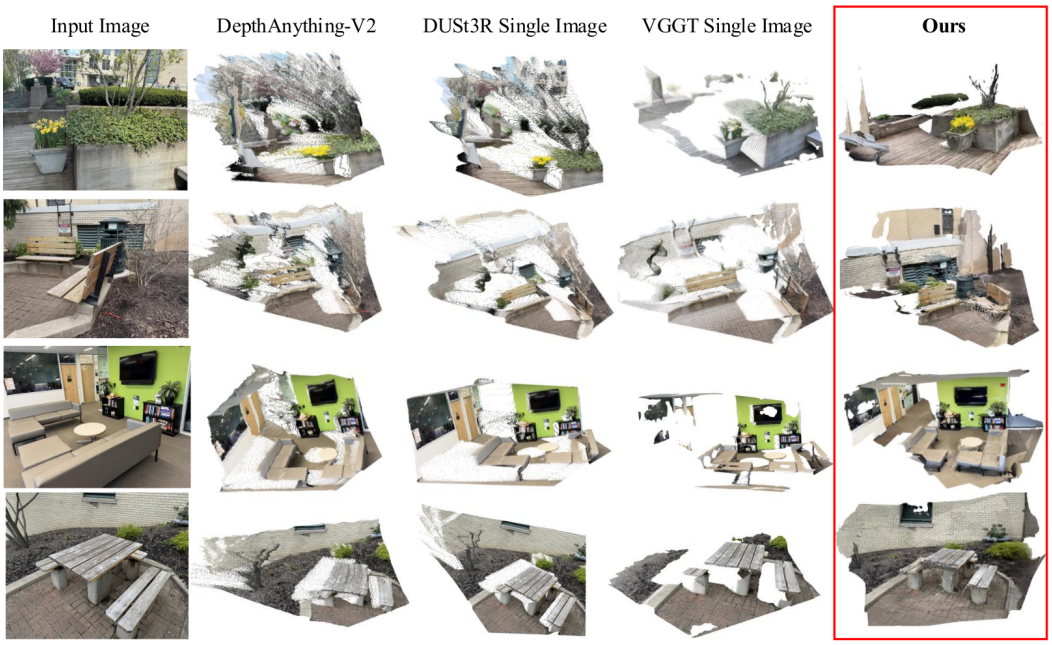


Figure 7: Qualitative evaluations of the complex 3D environment scenes (from top to bottom: garden, bench, indoor, table respectively) each extracted from a single image with our VGER relative to the baseline, DepthAnything-V2 and DUSt3R/ VGGT models with a single image input. We observe that not only does VGER fill in gaps occluded in the single image, but also altogether avoids the frustum-shaped noise artifacts, which are particularly prominent in the garden and bench images.

requiring hard constraints. The construction is smooth and fully differentiable, making it well-suited for reactive online motion generation. We illustrate the effect of a metric field steering the vector field away from the normal of the surface in fig. 6, on a toy problem. Here, we observe that the illustrated base vector field (shown in the left), after applying the effect of the metric field, smoothly warps around the black surface.

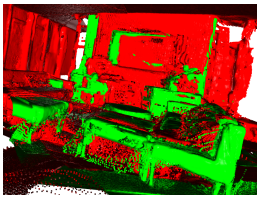
4 Empirical Evaluations

To evaluate the robustness and quality of our proposed Video-generation Environment Representation (VGER) along with the produced motion trajectories, we collect sets of 10 images in both outdoor and indoor scenes. These include the complex multi-object scenes **Garden**, **Bench**, **Table**, and **Indoor**, along with the more object-centric **Stone** and **Cabinet** scenes. For the evaluation of the quality of the performance of VGER and benchmark models, we extract the structure from a single image. We use the entire set of ten images, passed to DUSt3R, to construct a representation that we then consider to be the ground truth. Implementation details in our experiments are provided in section A1.

4.1 VGER Avoids Frustum Noise and Extracts Structure Accurately

We compare VGER with a single input image against a suite of strong baselines. In our setup, we uniformly sample 10 frames from the generated video to extract out our structure for VGER. These include: **DepthAnything-v2** [9]: a state-of-the-art monocular depth estimation foundation model; **VGGT with a single image** [15]: a recent 3D foundation model which can handle a single input image. We only pass a single image to generate the 3D representation; **DUSt3R with a single image** [6]: a widely used 3D foundation model, we again only input a single image to generate the 3D representation; **GEN3C** [20] and **ViewCrafter** [21]: These are contemporary controllable video generators, which we then extract 3D structure from to obtain the representation.

To evaluate the quality of the extracted scene representation, we rescale the environments to the interval $[-1, 1]$ and align the extracted scene representations with the ground truth representation built with the entire set of images using iterative closest point [36]. Then, we compute the Chamfer



	Garden	Bench	Table	Office	Stone	Cabinet
DepthAnythingV2	0.222	0.033	0.065	0.065	0.259	0.148
VGGT (single image)	0.109	0.068	0.094	0.062	0.296	0.121
DUSt3R (single image)	0.266	0.049	0.045	0.054	0.283	0.090
GEN3C [20]	**0.060**	**0.016**	0.060	0.049	0.104	0.066
ViewCrafter [21]	0.086	0.028	0.066	0.057	0.178	0.075
VGER (Ours)	0.075	0.029	**0.030**	**0.047**	**0.096**	**0.044**

Figure 8: (Left) Before computing distances, structure from one image by VGER (green) vs. ground-truth (red) are aligned. (Right) We report normalized Chamfer distances on different scene categories. Lower is better.

Distance [37] between the point clouds of the ground truth and extracted structure. For two point clouds P_1 and P_2, this is defined as,

$$D_{\text{Chamfer}}(P_1, P_2) = \frac{1}{|P_1|} \sum_{x \in P_1} \min_{y \in P_2} \|x - y\|_2 + \frac{1}{|P_2|} \sum_{y \in P_2} \min_{x \in P_1} \|y - x\|_2. \tag{10}$$

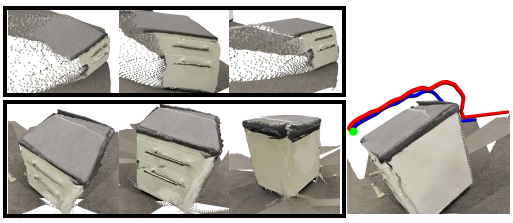


Figure 9: Single-image extraction leaves occluded regions incomplete, VGER reliably reconstructs them. Left: Top: DepthAnything-V2; bottom: VGER.; Right: Trajectories approaching the cabinet from the occluded backside, avoiding collision to reach the green goal.

The qualitative results are presented in fig. 8. Additionally, we illustrate an example alignment of the structure of the indoor environment extracted from a single image with VGER overlaid with a structure extracted via a 3D foundation model using all of the images collected. The Chamfer distances are subsequently computed between the aligned structures to obtain our quantitative results. We observe that VGER outperforms all the baseline methods consistently. Qualitatively, we observed that by using video generators, conditional on the input image, VGER subjects our structures to greater multi-view consistency. As a result, the frustum-shaped noise artifacts that appear in free space no longer appear in our obtained structure. Additionally, the video generator successfully estimates much of the geometry occluded from a single image, resulting in more geometrically accurate representations. This is particularly prominent in the complex Bench and Table scenes shown in rows 2 and 4 in fig. 7.

Here, we also highlight VGER's ability to deal with obstructions in fig. 9, where VGER can reconstruct the occluded backside of a cabinet. This enables motion trajectories and the correct collision-avoidance behaviour of even when approaching the cabinet from the back, which was not observed from the input image. A more intricate level of detail in regions that are hidden from the input image can also be seen in fig. 10. Here, by leveraging the large amounts of video data used to train the video generator model, VGER is able to accurately capture the geometry of the reverse side of the table surface, as well as the structure under the table, hidden from the input image view. We observe that this cannot be captured by the alternative comparison methods, and are left with gaps in obstructed regions.

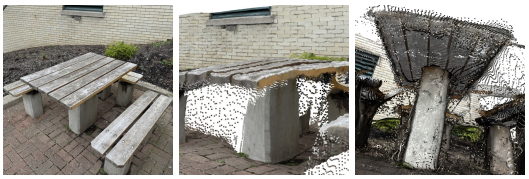


Figure 10: Left: Input image; Middle: Comparisons do not reconstruct unseen areas; Right: VGER reconstructs geometrically-consistent details in unseen regions, such as under the table.

4.2 Trajectories from Motion Policies Built on Constructed Representations

Next, we assess how VGER improves downstream motion generation. Using the approach outlined in section 3.3, we generate trajectories with implicit distance models derived from VGER, DepthAnything-V2 (DA-V2), and our ground-truth geometry. Our metric-field-based motion policy yields smooth trajectories, exemplified in the table scene (fig. 11), and we can compute per-step velocities in under 1ms. To quantify alignment with true motion, we integrate collision-free paths

Figure 11: VGER enables smooth collision-free motion. Here, two trajectories (red and blue) warp over benches and around the leg of the table to reach the goal (green).

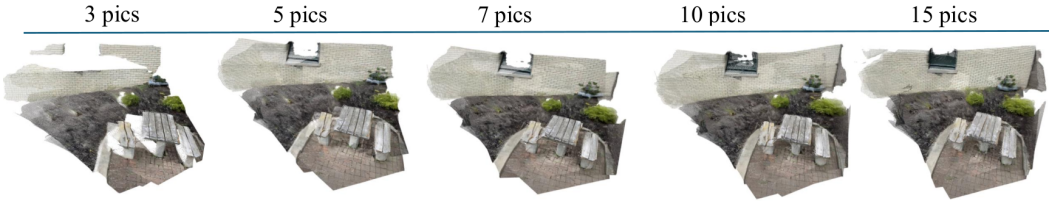


Figure 12: The extracted structure provided by VGER, sampling different numbers of frames from the generated video.

to a designated goal and measure the Discrete Fréchet Distance (DFD) [38], normalized by the Euclidean distance from start to goal. As shown in fig. 13c, lower DFD values indicate closer agreement with the ground-truth trajectories. Across all test scenes, VGER-based structures consistently outperform DA-V2, with particularly large gains in indoor, cabinet, and stone environments. This advantage arises because alternative methods introduce frustum-shaped noise artifacts in free space, which "funnel" the motion into occluded regions (see fig. 13a). By contrast, VGER completes obstructed areas without spurious artifacts, guiding trajectories correctly around the backside of the cabinet. Further examples in the complex indoor and garden scenes (fig. 13b) demonstrate that VGER-driven policies smoothly avoid collisions with the table surface and elegantly skirt around the plant and curb, respectively.

4.3 Ablation: Frames of Generated Video Used for Structure Extraction

The key to VGER's performance is its ability to leverage frames of videos generated from a large pre-trained model. When extracting the structure from the video, it can be unnecessarily resource-intensive to use every frame in the video, and we can uniformly subsample frames instead. In our results reported, we subsample 10 images. Here, we conduct an ablation study over the number images sampled from the video. We report results on the *table* scene using 3, 5, 7, 10, 15 images, computing the normalized Chamfer distance between the extracted structure and the ground truth after alignment. We observe that, beyond 5 images, the performance of VGER is robust to the number of frames extracted from the generated video. Images of the structures extracted for number of images sampled are in fig. 12.

Table 1: Performance of VGER with various images sampled on the *table* scene.

# Images	3	5	7	10	15
Dist. ($\times 10^{-2}$)	4.2	3.1	3.3	3.0	2.9

5 Limitations

The VGER pipeline exploits pre-trained video generators as foundation models for reconstruction and motion generation. In practice, these models synthesize geometry well within the camera's field of view but may "hallucinate" when extrapolating beyond the observed frustum. As the spatial distance from the original image capture region increases, so too does the uncertainty. An example of hallucinations in the generated video is illustrated in fig. 14. Future avenues of research to extend VGER can incorporate mechanisms to track extrapolated regions far outside the spatial re-

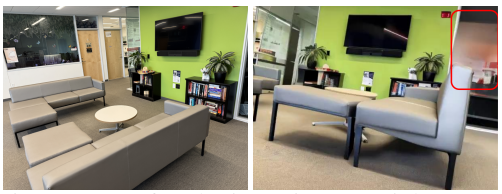
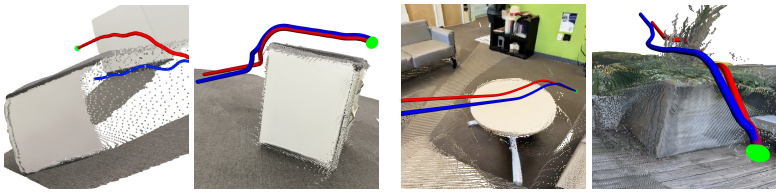


Figure 14: Left: The input image; Right: Generated frame contains hallucinated structure outlined in red that does not match the setting.



(a) Cabinet, Left: DA-V2; Right: VGER. (b) VGER trajectory examples.

	Ours	DA-V2
Bench	0.087	0.098
Garden	0.232	0.245
Indoor	0.111	0.203
Table	0.211	0.245
Cabinet	0.127	0.322
Stone	0.109	0.190

(c) Normalized DFD.

Figure 13: (a) Frustum-shaped noise artifacts often introduce local minima in the motion objective. We compare trajectories generated using DepthAnything-V2 (DA-V2) versus VGER: because VGER completes the cabinet's backside without spurious artifacts, it successfully navigates to the goal. (b) Sample paths on VGER-derived structures smoothly traverse along the table surface (indoor) and curve around a plant (garden), reaching their targets without collision. (c) The normalized Discrete Fréchet distances between trajectories generated with VGER and DA-V2 structures.

gion captured by the input image, assign uncertainty estimates and integrate the confidence-levels within the downstream motion policies.

6 Conclusion

In this paper, we tackle the problem of building an environment representation that enables motion generation from a single image. Here, we have present Video-generation Environment Representation (VGER), a framework that leverages pre-trained video synthesis and 3D foundation models to recover dense scene geometry from a single RGB image, without frustum-shaped artifacts. By applying a multi-scale noise contrastive sampling procedure to the 3D reconstruction, we extract an implicit unsigned distance field that smoothly encodes obstacle proximity across the workspace. Embedding this field into a metric-modulated motion framework yields a motion policy, modelled by a non-linear dynamical system capable of producing collision-free trajectories directly from a single viewpoint. Extensive evaluations demonstrate that VGER consistently outperforms state-of-the-art baselines, avoiding frustum-shaped artifacts and smoothly navigating complex indoor and outdoor scenes. This drives the broader impact of a future where robots can be actively deployed in open environments, enabling a safer workplace in the future.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009.

[2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[4] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[5] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[6] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[7] Nathan D. Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies, 2018.

[8] Weiming Zhi, Tin Lai, Lionel Ott, and Fabio Ramos. Diffeomorphic transforms for generalised imitation learning. In *Learning for Dynamics and Control Conference, L4DC*, 2022.

[9] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[10] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[11] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[12] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 1979.

[13] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024.

[15] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[16] Weiming Zhi, Haozhan Tang, Tianyi Zhang, and Matthew Johnson-Roberson. Unifying representation and calibration with 3d foundation models. *IEEE Robotics and Automation Letters*, 2024.

[17] Ilya Skorokhodov, Stefano Sabatini, and Guang Wang. Stylegan-v: A continuous video generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] Jonathan Ho, Jascha Sohl-Dickstein, and Pieter Abbeel. Video diffusion models. arXiv preprint arXiv:2209.14792, 2022.

[19] Hu Zhou, Liang Lin, and Shih-Fu Chang. Magicvideo: Aligning latent diffusion with transformers for text-guided video generation. arXiv preprint arXiv:2212.12367, 2022.

[20] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[21] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2025.

[22] Jinghao Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025.

[23] Karl Van Wyk, Mandy Xie, Anqi Li, Muhammad Asif Rana, Buck Babich, Bryan Peele, Qian Wan, Iretiayo Akinola, Balakumar Sundaralingam, Dieter Fox, Byron Boots, and Nathan D. Ratliff. Geometric fabrics: Generalizing classical mechanics to capture the physics of behavior. *IEEE Robotics and Automation Letters*, 2022.

[24] Weiming Zhi, Iretiayo Akinola, Karl van Wyk, Nathan Ratliff, and Fabio Ramos. Global and reactive motion generation with geometric fabric command sequences. In *IEEE International Conference on Robotics and Automation, ICRA*, 2023.

[25] Weiming Zhi, Lionel Ott, Ransalu Senanayake, and Fabio Ramos. Continuous occupancy map fusion with fast bayesian hilbert maps. In *International Conference on Robotics and Automation (ICRA)*, 2019.

[26] Weiming Zhi, Ransalu Senanayake, Lionel Ott, and Fabio Ramos. Spatiotemporal learning of directional uncertainty in urban environments with kernel recurrent mixture density networks. *IEEE Robotics and Automation Letters*, 2019.

[27] Michael Pantic, Lionel Ott, Cesar Cadena, Roland Siegwart, and Juan Nieto. Mesh manifold based riemannian motion planning for omnidirectional micro aerial vehicles. *IEEE Robotics and Automation Letters*, 2021.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[30] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[31] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 2011.

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[33] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

[34] Weiming Zhi, Tin Lai, Lionel Ott, Edwin V. Bonilla, and Fabio Ramos. Learning efficient and robust ordinary differential equations via invertible neural networks. In *International Conference on Machine Learning, ICML*, 2022.

[35] Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 21:124–127, 1950.

[36] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987.

[37] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 1977.

[38] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. *Technical Report CD-TR 94/64*, 1994.

Appendix

A1 Implementation Details

We run our experiments on a standard desktop with an Intel i9 CPU and an NVIDIA RTX 4090 GPU with 24GB VRAM. We use all the standard hyper-parameters in the Seva and DUSt3R foundation models used in our pipeline. Here list all the hyper-parameters used for the reconstruction and motion policies construction of our VGER method in table A1.

Implicit Model Multi-Scale Sampling		Network for Implicit Model		Motion Policy Blow-up	
α_{surf}	0.5	Layers	3	k	20
α_{eik}	0.1	Width	256	β	100
$\sigma_{\min}$	0.0025	ω_0	25	ϵ	10^{-8}
$\sigma_{\max}$	0.1	Training Epochs per LR	2000		
$\lvert\mathcal{P}\rvert$	10000	Learning rates (LRs)	3e-4, 1e-4, 5e-5, 1e-5		
$\lvert\mathcal{B}_s\rvert$	5000				

Table A1: VGER hyperparameters at a glance.

A2 Additional Visualizations of Reconstructions

Here we provide additional visualizations of the structures extracted from VGER along with baselines. These are provided in fig. A1. Additionally, here we provide visualizations for our ablation

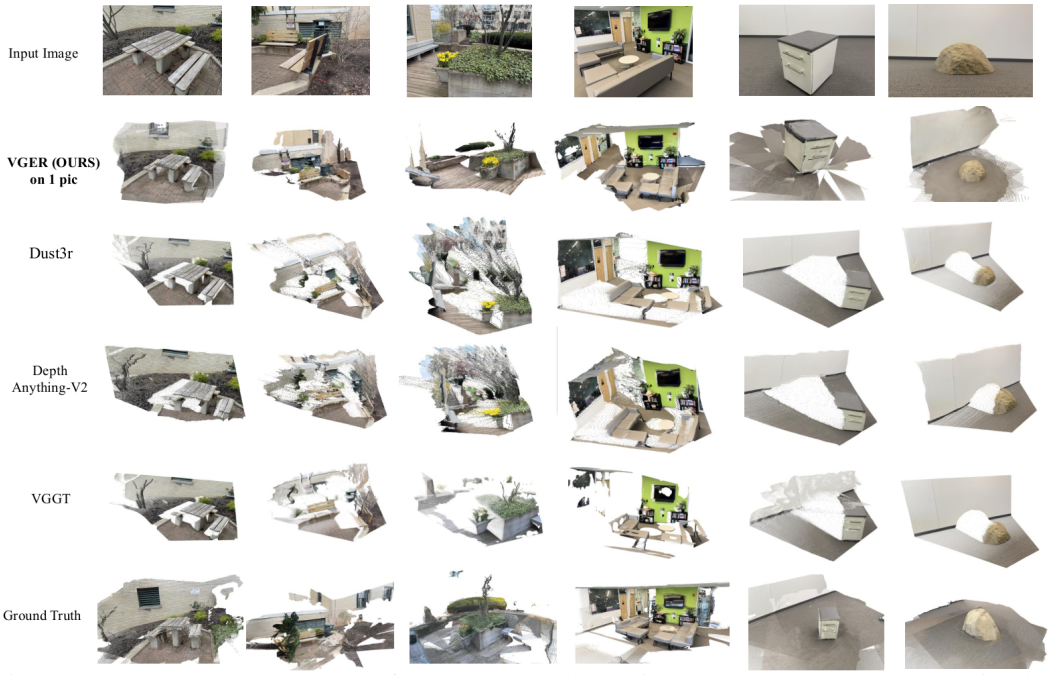


Figure A1: The extracted structures from our method, baselines, and ground truth, along with the single input image, for the table, bench, garden, indoor, cabinet, and stone environments.

experiment, and observe the structure extracted via VGER when the number of frames that we use from the generated video differ.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions of our proposed task and method are introduced in both abstract and introduction sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a limitations section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results is presented in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we detail the implementation details in both the experimental section, and more in detail in the appendices.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Though code has not been released during the review, the authors commit to releasing code with a camera-ready version of this manuscript.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All hyper-parameters used to reproduce the results are provided in the appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The metrics used are clearly defined in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The resources used in our experiments are clearly stated in the implementation details in the appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors reviewed the code and have no concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We highlight the application of VGER to robotics, and emphasize how this will lead to a more automated future with safer workplaces.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The authors see no potential risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: They are properly cited and licenses are respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new asset is released at the review stage.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing experiments and research with human subjects are involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.