RotoSLAM: Improving SLAM Scale Consistency for Rotational Motion

Erich Liang
erliang@princeton.edu

Hang Pham
hang.pham@princeton.edu

Sabrina Van
sabrinavan@princeton.edu

Arjun Menon
am9755@princeton.edu

Abstract

Today, many technologies such as autonomous vehicles, aerial mapping, and AR/VR rely on Simultaneous Localization and Mapping (SLAM) to track device location and create 3D maps of their surroundings using video input. However, SLAM can catastrophically fail in the monocular camera setting when the camera undergoes near-perfect rotational motion. Specifically, this type of camera motion prevents SLAM's bundle adjustment algorithm from propagating non-zero gradient information to scene depth, resulting in inconsistent scaling of different parts of the 3D reconstruction. In this project, we introduce RotoSLAM, which address rotational scale drift by supplying SLAM with additional supervisory signals from a diffusion-based depth estimator to help recover more accurate scene geometry when camera rotational motion is detected. Our diffusion-based depth estimator is conditioned on multiple previous frames' RGB images, thus leveraging the sequential nature of video frames. To finetune our diffusion depth estimator, we procedurally generate 3D ground truth training data via Infinigen. Through this depth supervision, our method reduces the amount of rotational scale drift observed in both real and synthetic SLAM video sequences.

1 Introduction

SLAM is an important problem in robotics and 3D vision, where the goal is to extract camera trajectory motion as well as a 3D map of the environment from just an input RGB video. Given how easy it is to place cameras on devices such as cars, phones, and other robots, SLAM algorithms serve as a cheap and popular way to obtain 3D information in settings such as robotics, autonomous vehicles, and AR/VR.

However, in the monocular SLAM setting, all state of the art (SOTA) SLAM methods catastrophically fail during camera rotational motion. This occurs because there is very little parallax between neighboring frames during rotation, causing near-zero gradient flow to the depth of the scene during SLAM's bundle adjustment algorithm. This results in different portions of the reconstructed scene to have different scales, thus incurring large trajectory inaccuracies. See Fig. 1 for an example. This is an important issue, especially in cases where we need to apply SLAM in environments that make it difficult to safely deploy multiple cameras, such as narrow caves or dense forests.

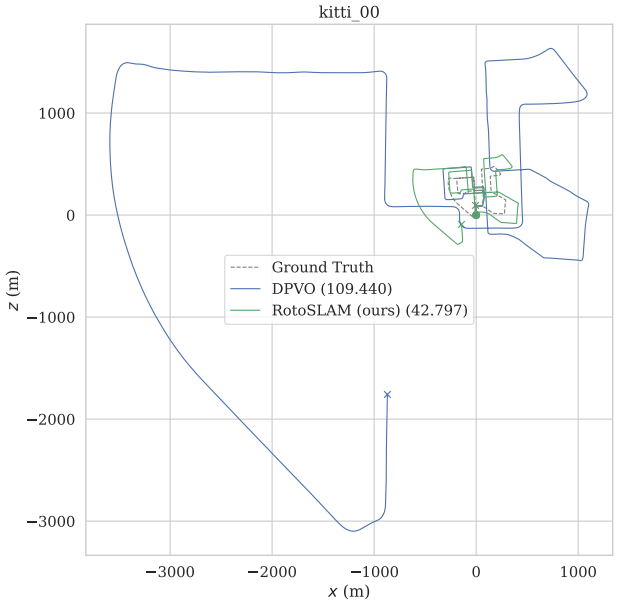

Fig. 1. An example of rotation scale drift from SLAM method DPVO [15]. The ground truth trajectory and predicted trajectory are aligned on their first 20 frames to showcase the effects of rotational scale drift over time. Our method RotoSLAM reduces the effect of rotational scale drift.

RotoSLAM addresses this issue by getting additional supervisory signal from external depth predictors during rotational motion, thus avoiding the zero-gradient flow issue

inherent to bundle adjustment. This is done in two main steps. First, we detect when rotational motion occurs by utilizing heuristics based on the reprojection locations between neighboring frames. When rotational motion is detected, we invoke our finetuned diffusion-based depth predictor module. Unlike traditional monocular depth predictors which only take in a single RGB image, our predictor is conditioned on multiple previous frames' RGB images. This takes advantage of the fact that consecutive frames in a video have similar structures that can help improve depth prediction accuracy. We utilize the predicted depth as additional supervisory signal during bundle adjustment, thereby reducing the amount of scale drift that occurs during rotational motion.

Training such a diffusion model requires a large amount of ground truth depth and camera pose data, which is hard to accurately obtain from real life video datasets. As a result, we choose to utilize procedurally generated scenes from Infinigen [11] to serve as our ground truth data. Because these scenes are procedurally generated, we can directly compute perfectly accurate depth maps and camera locations. By training on Infinigen data, our diffusion model is able to outperform other depth prediction models in the SLAM context, and ultimately achieve more accurate camera trajectories.

In summary, our contributions are as follows:

- We build a rotation detection module based on reprojection heuristics between consecutive frames
- We utilize Infinigen to produce large amounts of ground truth 3D data to train a diffusion model to predict depth from consecutive RGB frames
- We utilize our diffusion model's depth predictions as additional supervisory pressure during bundle adjustment to obtain more accurate scene depth during rotational motion, and ultimately more accurate camera trajectory reconstruction

Fig. 2. An example of depth ambiguity in the presence of camera rotational motion. In this example, let I_1^* and I_2^* denote the ground truth camera locations for two neighboring views, and let the two green Xs denote a pair of corresponding points between the images. When minimizing reprojection error, typically computed as the L2 distance between the projection of the 3D point responsible for the correspondence pair and the actual projection locations on each images, note that any point along the dashed line will result in 0 reprojection loss. As a result, the depth of the point is ambiguous, and the gradient flow to the scene depth during bundle adjustment will be zero.

2 Related Works

SOTA SLAM methods such as DPVO [15], DROID-SLAM [14], and ORB-SLAMv3 [2] achieve good accuracy on SLAM benchmarks when utilizing stereo video input. However, when limited to only monocular video, all three methods suffer from rotational scale drift despite the differences in their algorithmic design. Key to all three algorithms is bundle adjustment, which attempts to minimize the reprojection error for each pair of corresponding points on neighboring images. Fundamentally, there is a natural ambiguity of scene depth in the case of perfect rotational camera motion, making it theoretically impossible to deduce a correct scene depth in such cases. The gradient flow to the depth of each pixel becomes 0, thus allowing scenes to be arbitrarily rescaled during rotation motion. See Fig. 2 for an illustration of this principle. As a result, scenes are scale consistent during straight line motion, but can suddenly rescale during rotation motion, thereby leading rotational scale drift. RotoSLAM is designed to mitigate this issue by obtaining non-zero depth supervision from other sources when vanilla bundle adjustment runs into this theoretical bottleneck.

Existing SLAM benchmarks such as KITTI [9] and Waymo Open Dataset [13] utilize sensor fusion from stereo cameras, GPS, IMU, LiDAR, radar, and more to create ground truth camera trajectories for videos recorded. However, modalities like GPS and IMU can be error prone, especially when integrated over long periods of time. In addition, other modalities such as LiDAR and radar return sparse and noisy signal in the presence of special surfaces such as metallic and highly reflective objects. As a result, the generated camera trajectories and depth maps from these datasets are sparse and noisy. In contrast, Infinigen provides ground truth 3D geometry information without fail, as the entire scene is procedurally generated via raytracing. Furthermore, most existing SLAM datasets are obtained from driving and drone flying datasets. In contrast, Infinigen's scene diversity is much greater, with the ability to render both indoor and outdoor scenes.

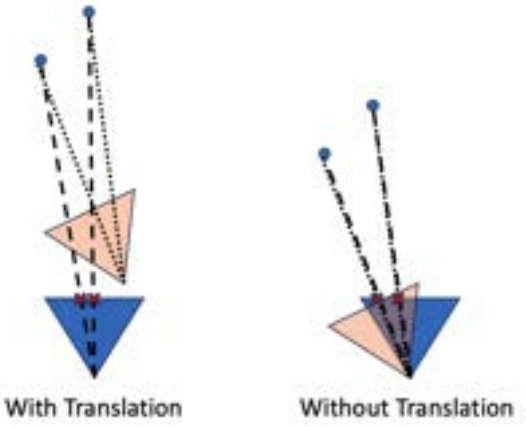

Fig. 3. An illustration of the reprojection difference heuristic used to determin the presence of rotational motion. By computing the reprojection of points on one image onto the other with and without the relative translation between them, we can get a sense of the reprojection's "sensitivity" to translation between the camera poses. If this sensitivity is low, this suggests that the relative motion between the cameras is rotaional in nature.

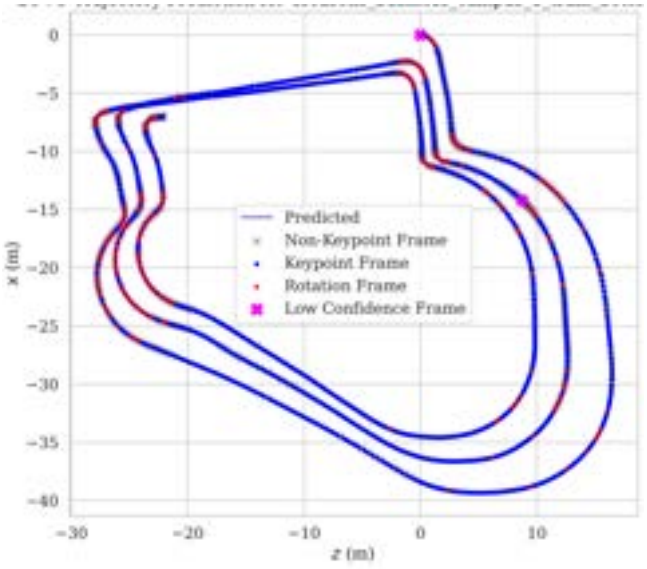

Fig. 4. An example of our rotation detection heuristic on a sample SLAM trajectory prediction. By changing the heuristic's thresholding, it is possible to tune the sensitivity of the rotation detection module.

3 Methodology

3.1 Rotation Detection

To detect when rotational motion is occurring within SLAM, we utilize a heuristic based on reprojections of corresponding points on consecutive image frames. Consider two consecutive images I_1 and I_2 in the input video. After several iterations of bundle adjustment have passed, SLAM will produce an estimate of the relative translation and rotation between the camera poses, as well as the depths of the points for some pixels on image I_1. Utilizing this information, we can first reproject the points on I_1 out into 3D space, and project these points back onto I_2 and record these reprojection locations. We can also repeat this process again, but this time ignoring the relative translation between the poses of I_1 and I_2. If the new reprojection locations on I_2 are close in position compared to the old reprojection locations, this means that the reprojection location is insensitive to the translational motion between I_1 and I_2, suggesting the presence of rotational motion. See Fig. 3 for an illustration of this reprojection heuristic. By thresholding on this reprojection difference, we are able to successfully detect rotational motion segments during trajectories, as shown in Fig. 4.

3.2 Fitting Monocular Depth Predictions to Existing Data

When training models for depth prediction, a common challenge is the mismatch in the color representation of depth maps between predicted outputs and ground truth data, as seen by Fig. 5. This may arise as the ground truth comes in different forms, such as absolute depth (from stereo cameras with known calibrations), disparity maps (from stereo cameras with unknown calibration), or depth up to an unknown scale (from structure from motion) [8]. Though these depth maps are conceptually equivalent, their numerical scales or baseline shifts may differ, leading to inaccuracies when using standard loss functions like L2 norm.

Problem Description In a typical training setup, the model outputs a predicted depth map, and the loss is calculated based on the difference between this prediction and the ground truth depth map. The loss function used is generally the L2 norm:

$$L = \|\text{GT} - \text{pred}\|_2 = \sqrt{\sum_{i=1}^{n} (\text{pred}_i - \text{GT}_i)^2}$$

However, this approach assumes that the predicted depth values (pred) and the ground truth (GT) are directly comparable. In reality, the predicted values may need to be adjusted by a scale and a shift to effectively match the ground truth values so the model does not overfit to particular numerical ranges or specific characteristics of the training data.

By adjusting for scale and shift to map between the ground truth images and output provided by our model, we ensure that the model learns to capture the underlying depth structures rather than memorizing noise-specific artifacts or distributional idiosyncrasies present in the training set. This approach

Fig. 5. Illustration of depth map generation and the challenges associated with color representation mismatches. **Top**: Original input image depicting a natural landscape with trees. **Middle**: Ground truth depth map generated by Infinigen. **Bottom**: Depth map generated by the Marigold model.

promotes better generalization to unseen data, which is crucial for deploying the model in diverse real-world scenarios.

Proposed Solution To address this scale and shift discrepancy, we propose the following two-step solution:

1. **Depth Prediction:** The model predicts the depth map.
2. **Scale and Shift Adjustment:** Post-prediction, we apply a linear transformation to the predicted depth values to align them with the ground truth. The transformation is defined by two parameters, α and β, representing the scale and shift, respectively. These parameters are computed by solving the following equation:

$$\min_{\alpha,\beta} \|\alpha \cdot \text{pred} + \beta - \text{GT}\|^2$$

where α and β are determined by optimizing the minimization problem above for each data batch without backpropagation of gradients. This is to ensure they are fixed values optimized to best fit the current predictions to the ground truth. Squared L2 norm was used as the square operation provides a smooth and continuous function that is easier to differentiate.

$$\min_{\alpha,\beta} \sum (\alpha \cdot \text{pred}_i + \beta - \text{GT}_i)^2$$

This optimization problem is transformed into a linear system to find the optimal values of α and β.

To solve this, we define the objective function:

$$S = \sum (\alpha \cdot \text{pred}_i + \beta - \text{GT}_i)^2$$

Afterward, we take the partial derivatives of S with respect to α and β, set them to zero, and solve for α and β.

- Derivative with respect to α:

$$\frac{\partial S}{\partial \alpha} = 2 \cdot \sum_{i=1}^{n} (\alpha \cdot \text{pred}_i + \beta - \text{GT}_i) \cdot \text{pred}_i = 0$$

$$2 \cdot \sum_{i=1}^{n} (\alpha \cdot \text{pred}_i + \beta - \text{GT}_i) \cdot \text{pred}_i = 0$$

$$\alpha \cdot \sum_{i=1}^{n} \text{pred}_i^2 + \beta \cdot \sum_{i=1}^{n} \text{pred}_i = \sum_{i=1}^{n} (\text{pred}_i \cdot \text{GT}_i)$$

- Derivative with respect to β:

$$\frac{\partial S}{\partial \beta} = 2 \cdot \sum_{i=1}^{n} (\alpha \cdot \text{pred}_i + \beta - \text{GT}_i) = 0$$

$$\alpha \cdot \sum_{i=1}^{n} \text{pred}_i + \beta \cdot n = \sum_{i=1}^{n} \text{GT}_i$$

These equations can be arranged into a matrix form, leading to the linear system:

$$\begin{bmatrix} \sum \text{pred}_i^2 & \sum \text{pred}_i \\ \sum \text{pred}_i & n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sum (\text{pred}_i \cdot \text{GT}_i) \\ \sum \text{GT}_i \end{bmatrix}$$

This linear system can be solved using standard techniques in linear algebra, specifically through the use of the `np.linalg.solve` function available in NumPy's library.

3.3 Data Acquisition Methodology

In order to train a model of this nature, we needed large-scale accurate ground truth data. In particular, we required ground truth data that was accurate on a pixelwise basis, as any lesser degree of specificity would further risk of producing the scale inconsistency issues our model was built to address.

Now this condition produces a challenge very early into the data acquisition process. It is simply impossible to

get pixelwise ground truth values for real videos. There certainly are many high quality video datasets available with LiDAR-produced depth maps, but these do not have a guarantee of pixelwise accuracy. LiDAR works well for individual, closer-range objects, but when the task broadens to mapping a full scene, the depth maps are simply estimates extrapolated from a smaller set of certifiably accurate depth measurements. True pixelwise accuracy can only be achieved from synthetic data. Thus, we turned our data acquisition efforts to a large-scale procedural 3D image generator out of Princeton's Vision & Learning Lab: Infinigen.

Infinigen We chose Infinigen because it guaranteed three different traits that were absolutely necessary for our data.

1. **Variance:** Infinigen generates all scenes procedurally, meaning that each scene is generated and composed from scratch, offering an infinitely large set of potential images. As Infinigen can generate a diverse array of entities, weather patterns, and locations, the effective randomness of the generation process produces the variance we need to ensure adequate training range.

2. **Photorealism:** Various studies, like the one described in "How Transferable are Video Representations Based on Synthetic Data?" [7] have confirmed the applicability and often increased accuracy of synthetic data-based models to real-world applications. Still, cautionary steps must be taken to ensure accuracy, with the best way to ensure that a synthetically-trained vision model works on real-world data being the utilization of synthetic data that mimics real data as closely as possible. The photorealism of the video frames produced by Infinigen allows for this.

3. **Accurate Ground Truth:** Our need for accurate ground truth values brings us to our primary reason for choosing Infinigen: it offers a pixelwise ground truth depth map for every frame of each generated video scene. In fact it goes even further, as by utilizing real geometry for scene models as opposed to the more common method of noise-induced artificial complexity generation, Infinigen is able to guarantee each depth value is to be completely accurate.

After finding a procedural generation tool that could create viable training data with the unique addition of perfect ground truth depth per pixel, we moved onto the process of scene generation itself. As Infinigen allows customizability of camera movement within scenes, we were able to generate various video scenes, each of approximately 200 frames. In each scene, we utilized a monocular camera rig undergoing simultaneous inward radial motion and rotational motion around a fixed central region. The specific parametric motion path differed on a scene-to-scene basis while still following the patterns of rotational motion and scale variation that tend to produce scale inconsistencies in traditional SLAM.

Fig. 6. Comparison of input images and their corresponding depth maps generated by Marigold. **Top**: Original images of a flower garden, a Ferris wheel, and a mountain house. **Bottom**: Output depth maps showcasing the model's ability to visually encode depth information based on color gradients.

3.4 Depth Prediction Methodology

Marigold To predict depth maps, we fine-tune Marigold, a diffusion model designed for image generation to predict monocular depth maps. The model, detailed in "Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation" [6], transforms the approach to depth estimation by leveraging generative capabilities.

Marigold operates on a single input image to generate depth maps, as demonstrated in Fig. 6. To enhance the model's accuracy and adapt it to dynamic scenes typical in video footage, we extended the input mechanism to integrate contextual information from multiple frames. Our method leverages the temporal continuity inherent in videos, a departure from traditional single-image depth estimation approaches.

Data Split and Performance Metrics For the evaluation of our model's performance, we utilized a dataset of videos generated by Infinigen, each accompanied by corresponding depth maps. This dataset comprises 30 labelled videos with corresponding depth maps, each consisting of 192 frames (5760 frames total), providing a robust basis for testing the enhancements made to Marigold. An example of the first 8 frames of one of the videos can be seen in Fig. 7. In organizing our dataset for a comprehensive evaluation, we divided the videos into three subsets: 19 videos were allocated for training

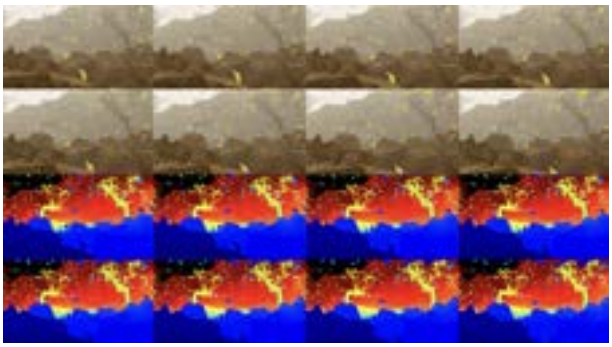

Fig. 7. Visualization of the first eight frames from a video generated by Infinigen, alongside their corresponding depth maps. The top two rows display the original video frames depicting a natural landscape scene with detailed foliage and rocky terrain. The bottom two rows show the depth maps for each frame, colored from red (indicating greater distance) to blue (indicating closer proximity).

($\approx 64\%$), 5 for validation ($\approx 16\%$), and 6 for testing (20%). To maintain the integrity of the video sequences due to the dependency on temporal information from previous frames, we randomized the allocation of whole videos across the training, validation, and test sets for each training iteration. However, the frames within each video were not randomized individually to preserve the sequential context, crucial for accurate depth estimation by our model.

To evaluate and quantify the improvements our methodology provides, we established a baseline performance measure where each frame is input independently into the Marigold model without utilizing the enhancements of incorporating historical frame data. The performance of the model—both the baseline and the enhanced version—is assessed using the Root Mean Squared Error (RMSE) between the depth maps predicted by the model and the ground truth depth maps provided with the dataset. To evaluate and quantify the improvements our methodology provides, we established a baseline performance measure where each frame is input independently into the Marigold model without utilizing the enhancements of incorporating historical frame data. The performance of the model—both the baseline and the enhanced version—is assessed using the Root Mean Squared Error (RMSE) between the depth maps predicted by the model and the ground truth depth maps provided with the dataset:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{pred}_i - \text{GT}_i)^2}$$

where pred_i represents the depth value predicted by the model for the i-th pixel, GT_i is the actual depth value for the i-th pixel, and n is the total number of pixels in the depth map.

Despite utilizing scale and shift factors to align the predicted depth maps more closely with the ground truth, the color variance across different videos might still be present. Consequently, a lower RMSE in one video compared to another does not necessarily indicate superior model performance on that specific video. Regardless, we use RMSE as a comparative metric within individual videos to discern the improvements our feature-enhanced model offers over the baseline. This internal comparison helps isolate the effect of our enhancements, ensuring any observed improvements in RMSE are attributable to our methodology rather than external factors.

Initial Approach Our initial approach was to extract features and compute optical flow from the preceding $N \in \{5, 8, 10\}$ frames relative to the current frame. This integration of historical data allows the model to account for motion and changes in the scene, enriching the context for depth prediction. The optical flow is calculated using the Farneback method [1], which provides a dense flow field capturing the motion between two consecutive grayscale frames. On the other hand, the feature extraction step is conducted using the DINO-ViT-S/8 model [10], a pre-trained vision transformer by Meta known for its ability to capture high-level contextual information from images. In theory, the extracted features would provide a semantically enhanced basis for adjusting the input images, ensuring the depth prediction model not only captures the geometric but also the dynamic semantic nuances of the scene. An example of the features extracted from one frame can be seen in Fig. 8.

Considering the differences in output characteristics between the feature extractions, optical flow data, and the original input images, we developed a method to adjust the input images based on these extracted data. These adjustments involve modifying the brightness and contrast of the input image. The rationale is to harmonize the input with transient scene elements, thereby enhancing the model's accuracy in perceiving depth. Specifically, the brightness is adjusted based on the mean and standard deviation of the extracted features, while the contrast is adjusted according to the normalized magnitude of the optical flow.

$$\text{Brightness Factor}(\beta) = 1 + k_\beta \left(\frac{\mu_F}{\sigma_F}\right)$$

$$\text{Contrast Factor}(\gamma) = 1 + k_\gamma \left(\frac{\text{norm}(\mathbf{F})}{\max(\mathbf{F})}\right)$$

Here, μ_F represents the mean of the extracted features, σ_F their standard deviation, and $\mathbf{F}$ denotes the optical flow matrix. The hyperparameters k_β and k_γ are scaling factors that modulate the influence of features and flow on the image adjustments, respectively. To optimize these, we employed a training strategy using a subset of our dataset specifically

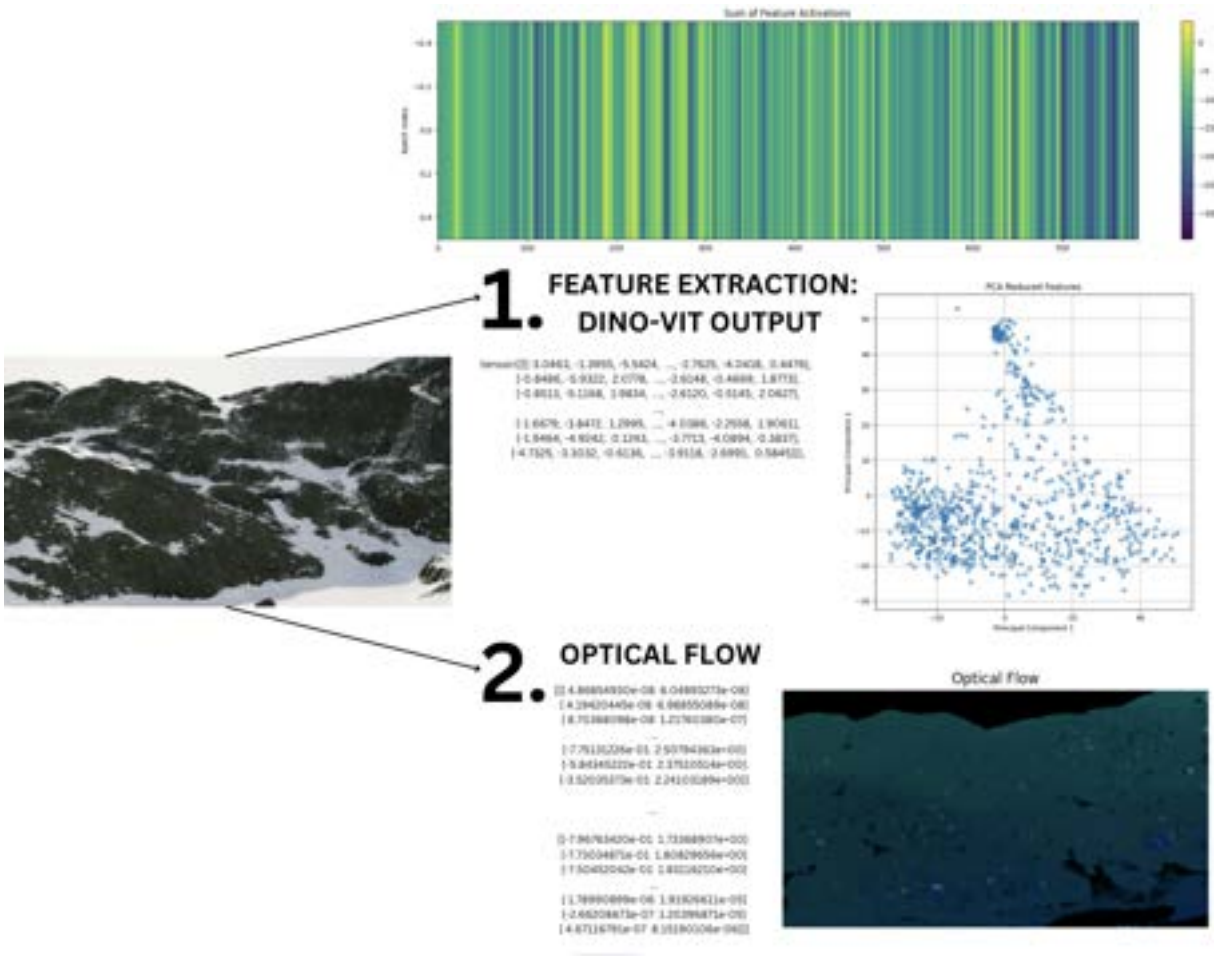

Fig. 8. Comprehensive analysis of feature extraction and optical flow visualization for a mountainous snowy landscape. **(1) Feature Extraction:** The leftmost image depicts the input scene used for feature extraction. The top right image displays a heatmap of summed feature activations across different channels, illustrating the intensity and distribution of activations throughout the scene. The middle right scatter plot represents the PCA-reduced features, highlighting the primary components of data variance which simplifies understanding the multidimensional data. **(2) Optical Flow:** The bottom right image visualizes the optical flow between consecutive frames, encoded as vector fields, where the color intensity and direction represent the motion dynamics within the scene.

held out for validation purposes. This training involved a systematic grid search over a range of potential values for k_β and k_γ that would reduce the overall RMSE between the predicted depth maps and the ground truth across the dataset.

Upon deploying the model in broader testing scenarios with videos not used in the initial grid search, the optimized values of $k_\beta = 0.025$ and $k_\gamma = 0.010$ were overfitting to the grid search videos, as the loss was larger for multiple test videos compared to baseline loss (see Table 1). This lack of generalization was attributed to the hyperparameters being finely tuned to the specific characteristics—such as lighting conditions and motion dynamics—of the training set.

Dynamic Frame Weight Approach We decide to use a dynamic frame weighting approach, which assigns different weights to each frame and combines them with the input. This aims to capitalize on the varying significance of each frame in contributing to depth perception. Importantly, this approach is inherently designed to enhance generalization across different videos. By focusing on the contextual relevance of each frame within a given video, rather than relying on fixed parameters or features extracted across disparate videos, the model adapts dynamically to the specific characteristics of each video sequence. This context-sensitive weighting mechanism ensures that our model can more effectively handle the diverse scenarios encountered in new videos, making it robust to variations in motion dynamics, scene composition, and camera handling that typically challenge fixed-parameter models.

Video #	Loss value (RMSE)						
	Baseline	N = 5		N =8		N = 10	
		FT	+ OF	FT	+ OF	FT	+ OF
3	0.564	0.561	0.560	0.563	0.559	0.546	0.445
9	0.386	0.389	0.386	0.401	0.387	0.413	0.406
14	0.798	0.783	0.769	0.801	0.812	0.803	0.802
17	0.437	0.428	0.425	0.435	0.434	0.439	0.438
25	0.481	0.482	0.479	0.491	0.487	0.501	0.492
28	0.613	0.623	0.612	0.623	0.621	0.632	0.626

Table 1. Summary of RMSE values for different video samples under baseline conditions and with enhanced feature extraction configurations. The table records the RMSE when the model runs with only feature extraction (FE) and with both feature extraction and optical flow (+ OF) for $N = 5$, 8, and 10 previous frames.

- **Process:** We process each frame using our depth prediction pipeline and calculate a weighted average of the depth maps based on optimized weights. These weights are adjusted iteratively during training to minimize the RMSE between the weighted depth prediction and ground truth.
- **Optimization:** The frame weights are optimized through a gradient descent method using the Adam optimizer, renowned for its efficiency in handling sparse gradients and adaptively tuning the learning rates for different weights. The optimization objective is to minimize the mean squared error between the weighted average of the predicted depth maps and the ground truth depth maps. This process is repeated for each video in the training set, ensuring that the final weights are robustly tuned across a diverse array of conditions.

When finding the optimized weights, we evaluate the performance of the dynamic frame weighting approach using different numbers of previous frames. Specifically, we conducted tests with 5, 8, and 10 previous frames to assess how the temporal depth influenced the accuracy and robustness of our depth estimation model.

Through these tests, with results recorded in Fig. 9, we discovered that utilizing 5 previous frames consistently yielded the best performance (lowest average RMSE). This provided a balance between capturing sufficient temporal information and avoiding the diminishing returns or potential overfitting associated with longer frame sequences. Note that while the RMSE experience fluctuations when transitioning between different videos, indicating sensitivity to scene-specific characteristics, the average RMSE across all test videos remained comparatively stable.

	N = 5	N = 8	N = 10
Average RMSE	0.384	0.410	0.465

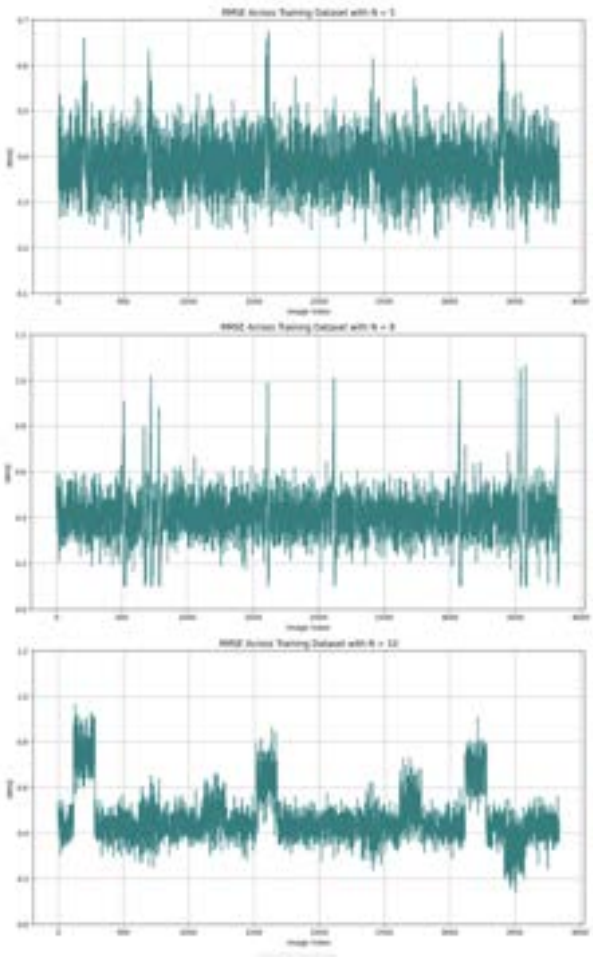

Fig. 9. Data obtained when optimizing weights for the diffusion models. **Top:** Summary of average RMSE values when optimizing weights for $N \in \{5, 8, 10\}$ frames. **Bottom:** Graphs demonstrating RMSE values across the training dataset for different frame counts (top is $N = 5$, middle is $N = 8$, bottom is $N = 10$). Significant fluctuations in RMSE values correspond to transitions between different video sequences within the training set.

Video #	Loss value (RMSE)			
	Baseline	Trained Weight	Exponential	Average
1	0.427	0.403	0.404	0.423
6	0.354	0.347	0.351	0.351
12	0.623	0.520	0.527	0.636
18	0.357	0.356	0.354	0.356
21	0.462	0.458	0.469	0.470
24	0.354	0.311	0.315	0.354

Table 2. Comparative RMSE results for depth estimation across test video sequences using baseline and modified models. This table highlights the performance improvement with the Dynamic Frame Weights over the standard Baseline and Average weight models.

When evaluating the performance of this model (**Trained Weight**) and further validate the efficacy of the dynamic frame weighting approach, we conducted a comparative analysis against several baseline models during our testing phase.

- **Baseline**: The baseline model (out-of-the-box Marigold) where each frame is processed independently without temporal integration.
- **Exponential**: A model where the weights for the frames exponentially decay with a factor of 0.8, assigning more significance to more recent frames.
- **Average**: A model that simply averages the weights across all considered previous frames.

As seen in Table 2, the dynamic frame weighting approach consistently outperformed the baseline model by at least 0.28%. This highlights the effectiveness of incorporating temporal dynamics into depth estimation. Moreover, it generally performed better than both the model with exponential decay and the average weighting model, emphasizing its superior ability to leverage relevant information from multiple frames. Furthermore, in the evaluation of depth maps on Infinigen data depicted in Fig. 10, our model successfully captured finer details, such as the stratification of mountains in the left image and the delineation of leaves in the right image.

Regarding the assessment of real-world scenarios, our model was qualitatively evaluated using a selection of images from both the DDAD dataset [5] and the KITTI dataset [3]. The model effectively captured essential structural details and maintained a high level of depth accuracy across various environmental conditions. Notably, in urban settings, the depth estimation preserved the geometric integrity of the scene, such as the accurate rendering of building outlines and road structures. For instance, as depicted in Fig. 11, our depth map more distinctly clarifies the building depths on the right side of the left photo. Additionally, in the right photo, our model more smoothly delineates the building in the top left corner, unlike the out-of-the-box model, which rendered it with a staircased appearance.

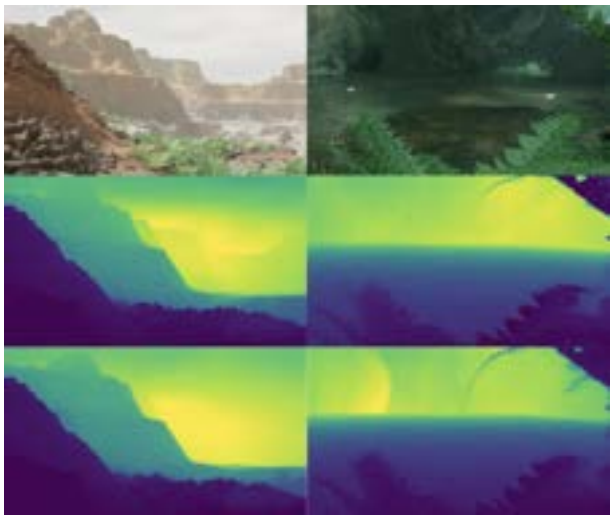

Fig. 10. Comparative visualization of depth map generation for a scene from generated footage with Infinigen. **Top:** Original image captured in an urban environment. **Middle:** Depth map generated by our model. **Bottom:** Depth map produced by the out-of-the-box Marigold model.

Similarly, in more complex and dynamic scenes, the model adeptly managed to delineate between objects at varying distances, showcasing its robustness in handling real-world variability. As illustrated in Fig. 12, our model demonstrates a profound capacity for capturing intricate details at both near and distant ranges, exemplified by the clarity of the electricity pole in the left image and the leaves on the distantly positioned tree in the center of the right image. This qualitative analysis underlines the model's potential in practical applications, suggesting its utility in navigational assistance systems and autonomous vehicle technologies.

4 Experiments

For our experiments, we connect our rotation detection module, diffusion-based depth predictor, and the DPVO SLAM algorithm connector for an augmented end-to-end SLAM system forming RotoSLAM. To evaluate the effectiveness of our method, we evaluate RotoSLAM on the KITTI [9] dataset, which none of our trained components have seen during training. Despite having never seen any KITTI trajectories or image data, our model is able to outperform DPVO and DPVO + Marigold methods. Qualitatively, the amount of scale drift is greatly reduced by RotoSLAM, and when utilizing Absolute Trajectory Error (ATE), our method performs the best.

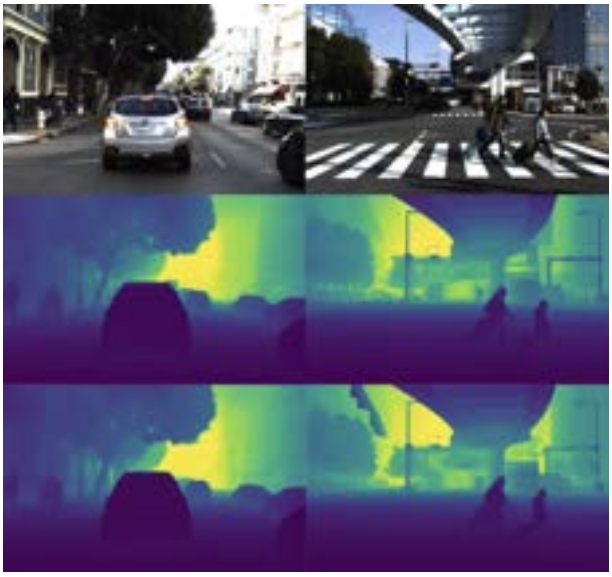

Fig. 11. Comparative visualization of depth map generation for a scene from the DDAD dataset. **Top:** Original image captured in an urban environment. **Middle:** Depth map generated by our model. **Bottom:** Depth map produced by the out-of-the-box Marigold model.

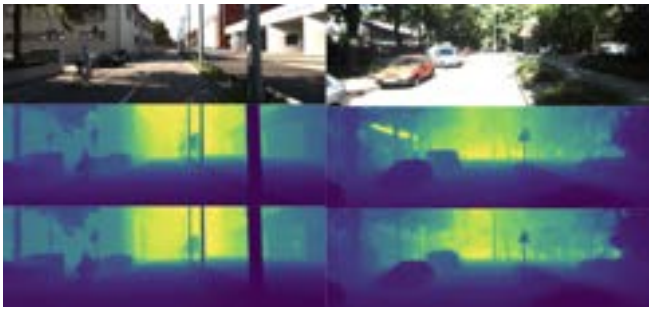

Fig. 12. Comparative visualization of depth map generation for a scene from the KITTI dataset. **Top:** Original image captured in an urban environment. **Middle:** Depth map generated by our model. **Bottom:** Depth map produced by the out-of-the-box Marigold model.

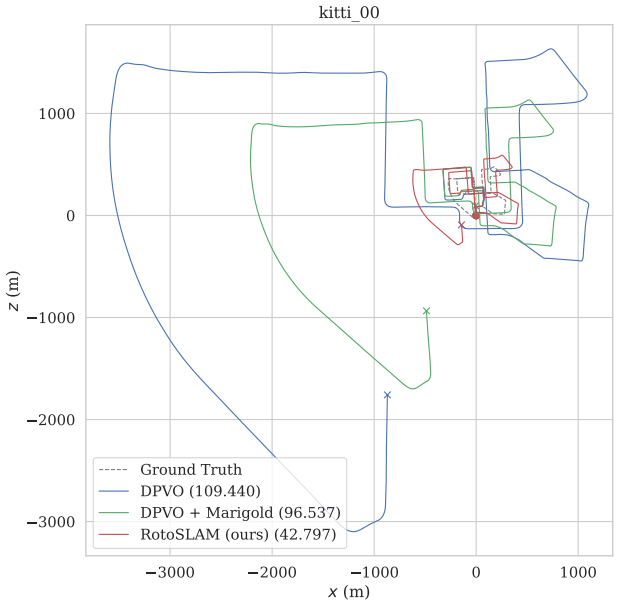

Fig. 13. Plot of trajectories of ground truth, DPVO, DPVO + Marigold, and RotoSLAM based on data from KITTI dataset's 00 trajectory.

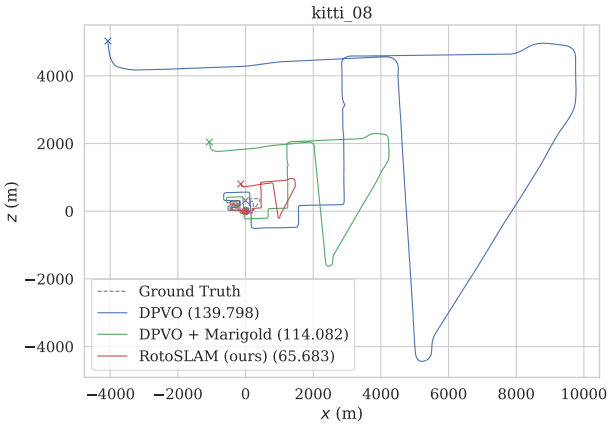

Fig. 14. Plot of trajectories of ground truth, DPVO, DPVO + Marigold, and RotoSLAM, based on more data from KITTI dataset's 08 trajectory.

5 Conclusion / Future Directions

5.1 Infinigen

As the process of training our model requires perfect pixel-wise ground truth data, we are limited by the scope of our only source of such data, Infinigen. As of the date this paper was written, Infinigen only offers full access for the generation of outdoor natural 3D images, which gives us adequate variation for a functional model but doesn't account for the full set of potential SLAM use cases. Infinigen is currently in the process of expanding to indoor and mixed-location scene generation as well, so retraining the model with a wider dataset that includes these scenes as well could increase the scope of its accuracy.

5.2 Diffusion Model

Our current experiment developed 3 diffusion models (based on averaging past frames, weights based on exponential decay, and weight optimization) in addition to the baseline model. Our models have shown improvement to the RMSE compared to the baseline model. However, there are approaches we have considered that could theoretically improve the RMSE even further.

One such approach would be to use Flow Matching in the diffusion model. Flow matching, which is often used for generative modelling, establishes a probability distribution $p_t : [0,1] \times \mathbb{R}^d \mapsto \mathbb{R}_+$ for the vector vector $u_t : [0,1] \times \mathbb{R}^d \mapsto \mathbb{R}^d$ [12]. Current research has shown that $u_t(x|z)$ can be properly sampled via $p_t(x|z)$ by transporting the probability path along the trajectory during training. This has only been trained and tested on straight trajectories, but flow matching has been shown to have significantly lower RMSE than Marigold [12].

Theoretically, if given enough data to train (which would be immensely huge to consider all possible trajectories), flow matching has the capability to use sampling and conditioning to learn the probability distributions of each 'object'/pixel and be able to accurately create depth maps along the correct trajectory.

Furthermore, according to "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer" [4], there are some limitations to using RMSE. Because RMSE considers the loss of all points, it is prone to outliers. Since most large-scale datasets can only provide imperfect ground truths, the paper suggests that the loss function should be modeled using the median. While we used RMSE since it is a common practice, our accuracy could increase by using a median-based loss function.

6 Author Contribution Statement

Erich focused on rotation detection, connecting depth prediction module to the DPVO backend, and evaluation scripts. Hang focused on the fine-tuning, training, and testing of the depth prediction model, leveraging the capabilities of the existing Marigold diffusion model for monocular depth prediction. Sabrina sourced real world datasets, trained multiple diffusion models, connected the diffusion model to Erich's backend, and analyzed possible further directions for enhancing the diffusion model. Arjun researched Infinigen, wrote software to control synthetic camera trajectories, generated all synthetic data, sourced real world data, and worked on the model theory. All authors contributed equally to the writing and revising of the paper, including citations and reference figures.

References

[1] BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[2] CAMPOS, C., ELVIRA, R., RODRIGUEZ, J. J. G., M. MONTIEL, J. M., AND D. TARDOS, J. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics 37*, 6 (Dec. 2021), 1874–1890.

[3] GEIGER, A., LENZ, P., STILLER, C., AND URTASUN, R. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)* (2013).

[4] GUI, M., FISCHER, J. S., PRESTEL, U., MA, P., KOTOVENKO, D., GREBENKOVA, O., BAUMANN, S. A., HU, V. T., AND OMMER, B. Depthfm: Fast monocular depth estimation with flow matching, 2024.

[5] GUIZILINI, V., AMBRUS, R., PILLAI, S., RAVENTOS, A., AND GAIDON, A. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[6] KE, B., OBUKHOV, A., HUANG, S., METZGER, N., DAUDT, R. C., AND SCHINDLER, K. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).

[7] KIM, Y.-W., MISHRA, S., JIN, S., PANDA, R., KUEHNE, H., KARLINSKY, L., SALIGRAMA, V., SAENKO, K., OLIVA, A., AND FERIS, R. How transferable are video representations based on synthetic data? In *Advances in Neural Information Processing Systems* (2022), S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., pp. 35710–35723.

[8] LASINGER, K., RANFTL, R., SCHINDLER, K., AND KOLTUN, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *CoRR abs/1907.01341* (2019).

[9] LIAO, Y., XIE, J., AND GEIGER, A. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d, 2022.

[10] OQUAB, M., DARCET, T., MOUTAKANNI, T., VO, H., SZAFRANIEC, M., KHALIDOV, V., FERNANDEZ, P., HAZIZA, D., MASSA, F., EL-NOUBY, A., ASSRAN, M., BALLAS, N., GALUBA, W., HOWES, R., HUANG, P.-Y., LI, S.-W., MISRA, I., RABBAT, M., SHARMA, V., SYNNAEVE, G., XU, H., JEGOU, H., MAIRAL, J., LABATUT, P., JOULIN, A., AND BOJANOWSKI, P. Dinov2: Learning robust visual features without supervision, 2024.

[11] RAISTRICK, A., LIPSON, L., MA, Z., MEI, L., WANG, M., ZUO, Y., KAYAN, K., WEN, H., HAN, B., WANG, Y., NEWELL, A., LAW, H., GOYAL, A., YANG, K., AND DENG, J. Infinite photorealistic worlds using procedural generation, 2023.

[12] RANFTL, R., LASINGER, K., HAFNER, D., SCHINDLER, K., AND KOLTUN, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2020.

[13] SUN, P., KRETZSCHMAR, H., DOTIWALLA, X., CHOUARD, A., PATNAIK, V., TSUI, P., GUO, J., ZHOU, Y., CHAI, Y., CAINE, B., VASUDEVAN, V., HAN, W., NGIAM, J., ZHAO, H., TIMOFEEV, A., ETTINGER, S., KRIVOKON, M., GAO, A., JOSHI, A., ZHAO, S., CHENG, S., ZHANG, Y., SHLENS, J., CHEN, Z., AND ANGUELOV, D. Scalability in perception for autonomous driving: Waymo open dataset, 2020.

[14] TEED, Z., AND DENG, J. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras, 2022.

[15] TEED, Z., LIPSON, L., AND DENG, J. Deep patch visual odometry, 2023.