

# NeRFoot: Robot-Footprint Estimation for Image-Based Visual Servoing

Daoxin Zhong, Luke Robinson and Daniele De Martini  
Mobile Robotics Group (MRG), University of Oxford

{daoxin.zhong, luke, danielle}@oxfordrobotics.institute

**Abstract**—This paper investigates the utility of Neural Radiance Fields (NeRF) models in extending the regions of operation of a mobile robot, controlled by Image-Based Visual Servoing (IBVS) via static CCTV cameras. Using NeRF as a 3D-representation prior, the robot’s footprint may be extrapolated geometrically and used to train a CNN-based network to extract it online from the robot’s appearance alone. The resulting footprint results in a tighter bound than a robot-wide bounding box, allowing the robot’s controller to prescribe more optimal trajectories and expand its safe operational floor area.

## I. INTRODUCTION

Visual servoing is a robotics technique that provides control based on visual feedback from external cameras. Since [1], the field has evolved to encompass various methodologies and approaches. Here we focus on Image-Based Visual Servoing (IBVS) [2], [3], which is typically about the specific 3D information of the robot and scene and performs state estimation and control of the robot in the image space.

Following [4], [5], we aim for a generalised pipeline for controlling a mobile robot indoors (e.g., a restaurant or a warehouse). The mobile robot is designed to be a simple, drive-by-wire apparatus capable of only receiving actuator commands from a Information and Communications Technology (ICT) infrastructure. The robot’s position is captured via various CCTV cameras placed within the environment and is processed to supply a control signal to the mobile robot, directing it towards some navigational target. Critically, no 3D information about the robot or the environment is supplied to the system, leading to suboptimal use of the driving surfaces, as shown in fig. 1.

Drawing inspiration from previous works leveraging CAD as known prior [6], we overcome these issues by first building a Neural Radiance Fields (NeRF)-based [7] 3D representation of the robot using images captured from our CCTV cameras. The robot’s footprint, i.e. its contour when projected into the ground plane, is then estimated using an image of the robot synthesised from a downward perspective. Its outline is then reprojected into the camera plane of the original images to generate masks and train a YOLO [8] segmenter for online footprint estimation at operation time.

## II. METHODOLOGY

Our proposed methodology consists of two sequential steps: NeRF model training, and YOLO footprint estimation.

### A. NeRF Model Training

We start from the same calibration data collected from [5], consisting of a video of the robot spinning in place



Fig. 1: [4] controls the robot based on its bounding box (yellow) and orientation (green). When checking if a trajectory is safe, its box must stay within the drivable region (blue). However, this precludes a huge area, which is still safe but intersects with the wall (red) in the image plane.

with its bounding boxes. To train a NeRF model, a mask for the robot is first generated *at every frame* from Segment Anything Model (SAM) [9] by seeding it with the bounding boxes. This mask serves two purposes. It segments the robot from the static background, allowing the relative pose of the camera to the robot to be estimated via traditional Structure-from-Motion (SfM) methods or more recent transformer models such as DUST3R[10]. Additionally, it selects the relevant ray bundles to be passed onto Nerfstudio [11] that informs the NeRF about the colour and density functions of the robot. Critically, the iNeRF [12] correction mechanism helps correct the estimated camera poses and improves the overall render quality. This is further improved by optimising the camera poses along a plane, as constrained by the physical dimensions of the experimental setup.

### B. YOLO Footprint Estimation

We first synthesise a downward view of the robot and extract its mask via SAM [9]. The coordinates of the footprint contours in the world frame are calculated by solving for the points of intersection between the projection rays and the ground plane.

To estimate the robot’s footprint during operation, we train a segmentation network using a synthetic dataset of the robot rendered via NeRF. By projecting the footprint contours into the image plane, we generate a corresponding footprint mask that varies with the robot’s appearance. This is used to train a YOLOv8 segmentation network, which is fast enough to provide real-time segmentation during deployment. Figure 2 shows examples of the footprint taken from a CAD model and the estimates from YOLO trained on the CAD or the

# Spins	Frames ps	PSNR ( $\uparrow$ )	SSIM ( $\uparrow$ )	LPIPS ( $\downarrow$ )
6	50	41.0	0.993	0.00256
	25	43.2	0.995	0.00188
	10	34.0	0.985	0.01556
3	50	40.4	0.993	0.00479
	25	33.6	0.985	0.01595
	10	33.3	0.984	0.01709
1	50	33.6	0.984	0.01714
	25	33.5	0.984	0.01708
	10	33.3	0.984	0.01672

TABLE I: Average metrics by spins count and number of sampled images per spin.

	Avg Error	PSNR ( $\uparrow$ )	SSIM ( $\uparrow$ )	LPIPS ( $\downarrow$ )
Baseline	0	56.0	0.999	0.00037
Default Opt.	1	34.8	0.987	0.0135
	2	34.8	0.987	0.0139
	5	34.3	0.986	0.0151
	10	34.2	0.985	0.0164
Plane Opt.	1	41.0	0.993	0.00256
	2	41.8	0.994	0.00259
	5	34.4	0.986	0.00152
	10	34.2	0.985	0.00165

TABLE II: Average metrics by varying error magnitude.

synthetic dataset, showing how YOLO can segment the footprint reliably from the constructed data.

### III. EXPERIMENTAL RESULTS

Here, we report the results obtained in simulation on a Clearpath Jackal robot model using Blender, where we apply the same calibration approach as in [5].

#### A. Synthetic Image Generation

NeRF is notoriously data-hungry. As such, we first analyse how many data points we need to create a usable render in table I. Unsurprisingly, we notice how more images produce a better model for rendering; however, this incremental improvement is not uniform, and it becomes marginal after 150 images. In a qualitative assessment, 166 of 200 generated images are deemed sufficiently photorealistic, closely resembling their counterparts in the original dataset.

Similarly, as the synthetic camera positions will be inherently noisy, we show ablations on an initial error in NeRF’s rendering quality. The results are reported in table II. Here, we analyse the utility of our constrained camera-pose optimisation. We can see how ours (plane opt.) always outperforms the default optimizer.

#### B. Footprint Estimation with YOLO

To show the validity of our approach, we train a YOLOv8 model on the labels created through the method described in section II-B. Here, we show two separate approaches, where we train YOLO on the real images used for training NeRF or fully synthetic ones. We use 300 images in both cases and train a `yolov8m-seg` model for 100 epochs. The resulting YOLO models predict the footprint of the 200 images in the evaluation dataset, which is separate from the training one. A qualitative example can be seen in fig. 2.

Calculating the Cross-Correlation Ratio (CCR) of the predicted footprint against that found via ray projection, the

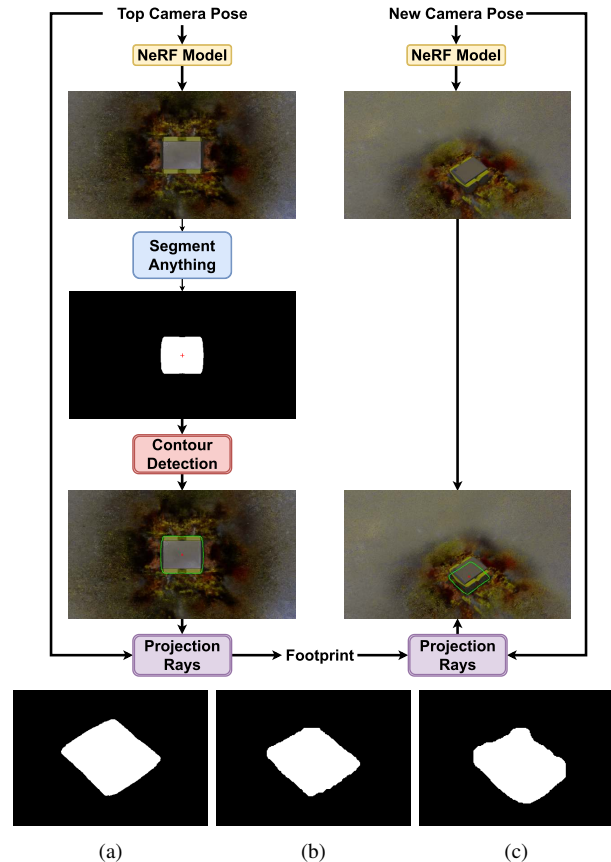


Fig. 2: System diagram of the footprint-estimation training process via the projection ray transform method (top) and excerpt of the footprint (bottom) as from the CAD ground truth (a), YOLO trained from the real-world ground truth (b) and from the synthetic ground truth (c).

first YOLO model achieves an average CCR score of 0.849 but fails to segment a footprint in 14 out of the 200 images. The second YOLO model is always successful at segmenting a footprint while achieving a higher CCR score of 0.873. However, a qualitative survey of the footprints generated reveals that its predicted footprint tends to be blobbier, lacking the straight edges calculated from the projection ray transform method. Critically for this application, though, blobbier images result in a conservative estimate, which may be favourable.

### IV. CONCLUSION

This work demonstrates a simple training procedure for extracting 3D priors to extend the operational areas of a IBVS system. By estimating the relative pose of the fixed CCTV to the moving robot, we can construct its NeRF model through the CCTV footage with no additional priors. This model can then extrapolate the robot’s footprint and render a synthetic, labelled dataset to train a YOLO model to segment the robot’s footprint from real data online at deployment time. We demonstrated its viability through simulation data, highlighting its simplicity and limitations in training data.

## REFERENCES

- [1] J. Hill, "Real time control of a robot with a mobile camera," in *Proc. 9th Int. Symp. on Industrial Robots*, 1979, pp. 233–245.
- [2] N. Garcia-Aracil, C. Perez-Vidal, J. M. Sabater, R. Morales, and F. J. Badesa, "Robust and cooperative image-based visual servoing system using a redundant architecture," *Sensors*, vol. 11, no. 12, p. 11885–11900, Dec. 2011. [Online]. Available: <http://dx.doi.org/10.3390/s111211885>
- [3] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.
- [4] L. Robinson, M. Gadd, P. Newman, and D. De Martini, "Robot-relay: Building-wide, calibration-less visual servoing with learned sensor handover network," *arXiv preprint arXiv:2310.15677*, 2023.
- [5] L. Robinson, D. De Martini, M. Gadd, and P. Newman, "Visual servoing on wheels: Robust robot orientation estimation in remote viewpoint control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 6364–6370.
- [6] J. Feddema and O. Mitchell, "Vision-guided servoing with feature-based trajectory generation (for robots)," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, pp. 691–700, 1989.
- [7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," 2020. [Online]. Available: <https://arxiv.org/abs/2003.08934>
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.
- [10] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.
- [11] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," 2023. [Online]. Available: <https://arxiv.org/abs/2302.04264>
- [12] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting neural radiance fields for pose estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.