Unifying Scene Representation and Hand-Eye Calibration with 3D Foundation Models

Weiming Zhi Haozhan Tang Tianyi Zhang Matthew Johnson-Roberson

I. Introduction

The manipulator-mounted camera setup, where a camera is rigidly attached onto the manipulator, enables the robot to actively perceive its environment and is a common setup for robot manipulation. The camera must be calibrated before collecting data from the environment. Specifically, to obtain representations that the robot can plan within, it is important to transform them to the frame of the robot. Finding the camera pose with respect to the end of the manipulator arm, or *end-effector*, is known as *hand-eye calibration* [1]. This is generally an elaborate procedure that requires the camera to move to a set of poses while recording images of an external marker, usually a checkerboard or an AprilTag [2].

Recent advances within the computer vision community has been driven by deep learning approaches. This has led to the emergence of large pre-trained models, which greatly outperform classical approaches at estimating correspondences between a set of images of the scene, and can thereby be used for many multi-view problems. These models are trained on prohibitively large datasets, and intended as *plug-and-play* modules to facilitate a wide variety of downstream tasks. We therefore describe these models as *3D Foundation Models* [3], and advocate for their integration in robot camera calibration and scene representation.

In this paper, we contribute the *Joint Calibration and Representation (JCR)* method. JCR leverages 3D foundation models to enable the construction of a scene representation in the coordinate frame of the manipulator's base, from a small set of images, collected from a RGB camera mounted on the manipulator. To the best of our knowledge, **our approach is the first to simultaneously calibrate the camera and build a scene representation from the same set of images captured by a manipulator-mounted camera.** We obtain a model of our environment in the robot's coordinate frame, while not requiring any *a priori* calibration, nor external markers, nor depth measurements. The constructed scene representation is an continuous model which can be used for collision-checking in subsequent motion planning.

II. Related Work

Scene Representation: Early work in representing environments, notably Occupancy Grid Maps [4], recorded properties in discretized cells. Distance representations has also been applied [5], [6] to check for collisions. Continuous representations emerged thereafter: e.g. by Gaussian Processes [7], kernel regression [8], [9], Bayesian methods [10], [11], and neural networks [12]. Deep learning methods have also operated directly on point clouds [13], [14]. Concurrently, there has been an effort to create *photorealistic representations*, including Neural Radiance Fields (NeRFs) [15] and variants [16]. These rely on obtaining an initial solution from the Structure-from-Motion method COLMAP [17]. **Hand-eye Calibration:** Hand-eye calibration is a well-studied problem with many solutions [1], [18], [19] developed, when some external calibration marker is provided. A recent learning approach for hand-eye calibration is presented in [20], but requires the robot's gripper to be partially visible in the camera view. Additionally, methods in end-to-end policy learning [21], [22] directly produce actions from the camera images with no calibration. However, unlike our method, these methods are unable to construct an environment which can then be used for collision-checking and motion planning [23], [24]. **Pre-trained Models:** Considerable effort has been undertaken to create large models, for natural language processing [25] and computer vision [26]. In particular, [27] introduces a pre-trained model specifically for 3D tasks. These large pre-trained models are known as *foundation models* [3], and are typically treated as back-boxes whose outputs are used in downstream tasks.

III. Prelminaries: Foundation Models for 3D Vision

Traditional methods for 3D vision tasks such as Structure-from-Motion [17] or multi-view stereo [28] depend on identifying visual features over a set of images to infer the 3D structure. On the other hand, pre-trained models, such as *Dense Unconstrained Stereo 3D Reconstruction* (Dust3r), have been trained on large datasets and that can identify correspondences over a set of images without engineered features. Throughout this work, we use Dust3r [27] as the foundation model and follow its conventions.

Pairwise Pixel Correspondence: Suppose we have a pair of RGB images with width W and height H, i.e. $I_1, I_2 \in \mathbb{R}^{W\times H\times 3}$, our foundation model can produces *pointmaps*, $X^{1,1}, X^{1,2} \in \mathbb{R}^{W\times H\times 3}$. These assign each pixel in the 2D image to its predicted 3D coordinates and are **critically in the same coordinate frame of** I_1. Confidence maps, $C^{1,1}, C^{1,2} \in \mathbb{R}^{W\times H}$, for the pointmaps are also produced. By finding the nearest predicted coordinates of each pixel in the pointmap with those of the other pointmap, we can find dense correspondences between pixels in the image pair.

Recovering Relative Camera Poses: We optimize to globally align the pairwise pointmaps predicted by the

The authors are with the Robotics Institute, Carnegie Mellon University. Correspondence to wzhi@andrew.cmu.edu.

foundation model to recover the relative camera poses corresponding to *a set of images*. For a set of N images, we have the cameras $n = 1, \ldots, N$ and possible image pairs with indices $(n, m) \in \varepsilon$, where $m = 1, \ldots, N$ and $m \neq n$. For each pair, the foundation model gives us: (1) Pointmaps in $X^{n,n}, X^{n,m} \in \mathbb{R}^{W \times H \times 3}$ in the frame of I_n; (2) Corresponding confidence maps $C^{n,n}, C^{n,m} \in \mathbb{R}^{W \times H}$. With these, we seek to optimize to find: (1) For each of the N images, a pointmap in global coordinates $\bar{X}^n$; (2) A rigid transformation described $P_n \in \mathbb{R}^{3\times4}$ and scale factor $\sigma_n > 0$. Intuitively, the same transformation should be able to align both images in each pair to their equivalents in the global coordinate. We then minimize the distance between the transformed and predicted pointmaps in global coordinates:

$$\min_{\hat{X}, P, \sigma} \sum_{(n,m)} \sum_{i \in (n,m)} \sum_{(w,h)} C^{n,i}_{w,h} ||\hat{X}_i - \sigma_e P_e X^{n,e}_{w,h}||_2. \quad (1)$$

Here, $(n, m) \in \varepsilon$ denotes the pairs, i iterates through the two images in each pair, and (w, h) iterates through each pixel in the image. After alignment, we extract the set of camera poses as well as the aligned pointmaps over the set of images. Next, we need to transform the outputs to be in the robot's frame and recover physically-accurate scales.

IV. Joint Calibration and Representation

We tackle the problem setup of a manipulator with an uncalibrated inexpensive RGB camera rigidly mounted on the manipulator. The pose of the camera relative to the end-effector is unknown. We control the end-effector manipulator to go to a small set of N poses, $\{E_1, \ldots, E_N\}$, and capture an image at each pose. This gives us a set of images N of the environment, $\{I_1, \ldots, I_N\}$, which can we inputted into the foundation model to obtain a set of aligned relative camera poses $\{P_1, \ldots, P_N\}$ and pointmaps $\{X_1, \ldots, X_N\}$ with respect to an arbitrary coordinate system and scale. Here, we seek: (1) The rigid transformation, T^e_c, from the frame of the mounted camera to that of the end-effector. (2) An environment representation in the robot's frame.

Calibration With Foundation Model Outputs: Here, we seek to solve for T^e_c with the end-effector poses $\{E_1, \ldots, E_N\}$ the predicted unscaled relative camera poses $\{P_1, \ldots, P_N\}$. We shall consider transformations between subsequent end-effector poses $T^{E_{i+1}}_{E_i}$ and transformations between camera poses $T^{P_{i+1}}_{P_i}$, where $E_{i+1} = T^{E_{i+1}}_{E_i} E_i$ and $P_{i+1} = T^{P_{i+1}}_{P_i} P_i$ for $i = 1, \ldots, N-1$.

As the foundation model does not recover absolute scale, we introduce a *scale factor* λ. The transformation between scaled estimated camera poses as $T^{P_{i+1}}_{P_i}(\lambda) = \begin{bmatrix} R^{P_{i+1}}_{P_i} & \lambda \mathbf{t}^{P_{i+1}}_{P_i} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \in \mathbf{SE}3$, where $R^{P_{i+1}}_{P_i} \in \mathbf{SO}3$ denotes the rotation component of $T^{P_{i+1}}_{P_i}$ and $\mathbf{t}^{P_{i+1}}_{P_i} \in \mathbb{R}^3$ denotes the translation. Scaling the distances between camera poses does not affect the rotation, only the translation.

The relationship between $T^{E_{i+1}}_{E_i}$, $T^{P_{i+1}}_{P_i}(\lambda)$ and the desired T^e_c follows the matrix equation from classical hand-eye calibration [1]:

$$T^{E_{i+1}}_{E_i} T^e_c = T^e_c T^{P_{i+1}}_{P_i}(\lambda), \quad (2)$$

and we shall solve for the best fit T^e_c and λ. We begin by solving for the rotational term R^e_c by following [19], and considering the log map of $\mathbf{SO}3$ to its lie algebra ($\mathfrak{so}3$) where for some $R \in \mathbf{SO}3$, $\omega = \arccos(\frac{\mathrm{Tr}(R)-1}{2})$, $\mathrm{LogMap}(R) := \frac{\omega}{2\sin(\omega)} \begin{bmatrix} R_{3,2} - R_{2,3} \\ R_{1,3} - R_{3,1} \\ R_{2,1} - R_{1,2} \end{bmatrix}$ Here, the subscripts indicate the elements in R, and $\mathrm{Tr}(\cdot)$ indicates the trace operator. Then, the best fit rotation R^{e*}_c can be found via: $R^{e*}_c = (M^\top M)^{-\frac{1}{2}} M^\top$, where $M = \sum_{i=1}^{N-1} \mathrm{LogMap}(R^{E_{i+1}}_{E_i}) \otimes \mathrm{LogMap}(R^{P_{i+1}}_{P_i})$, and where $\otimes$ denotes the outer product, and the matrix inverse square root can be found via singular value decomposition.

Next, we formulate a residual optimization problem to find the best-fit translation $\mathbf{t}^{e*}_c$ and scale λ^*, known as the *Scale Recovery Problem* (SRP):

$$\text{SRP:} \quad \arg\min_{\mathbf{t}^e_c, \lambda} \sum_{i=1}^{N-1} ||C_i \mathbf{t}^e_c - \mathbf{d}_i||^2_2, \quad (3)$$

where $C_i = I - R^{E_{i+1}}_{E_i}$, $\mathbf{d}_i = \mathbf{t}^{E_{i+1}}_{E_i} - R^{e*}_c(\lambda \mathbf{t}^e_c)$. After solving the SRP, we can obtain the entire camera to end-effector transformation T^{e*}_c along with λ^*.

Map Construction with Foundation Model Outputs: Next, we seek to build representations of the environment with the output of the foundation model: a set of aligned pointmaps $\{X_1, \ldots, X_N\}$ with associated confidence maps $\{C_1, \ldots, C_N\}$. From these, we can set a confidence threshold and filter out a 3D point cloud $\{\mathbf{x}_i\}_{i=1}^{N_{pc}}$. For the camera pose P and at end-effector pose E, we transform the point cloud to the frame of the robot and adjust the scale to match the real-world via $\bar{\mathbf{x}}_i = E^{-1} T^{e*}_c (\lambda^* \mathbf{x}_i)$.

The *occupancy* is useful for planning tasks. Here, we use a small neural network f_θ to learn a continuous *implicit* model of occupancy, which assigns a probability of being occupied to each sptial coordinate. We take a Noise Contrastive Estimate (NCE) [29] approach and minimize the binary cross-entropy loss [30], with $\bar{\mathbf{x}}_i$ as positive examples and uniformly drawing negative examples $\bar{\mathbf{x}}^{neg}_i$. Similar to NeRF models [15], we also apply position embedding ϕ on the inputs. Our loss function is,

$$L(\theta) = BCELoss(\{f_\theta(\phi(\bar{\mathbf{x}}_i))\}_{i=1}^{N_{pc}}, \{f_\theta(\phi(\bar{\mathbf{x}}^{neg}_i))\}_{i=1}^{N_{pc}}),$$

where $\bar{\mathbf{x}}^{neg}_i \sim U(\bar{\mathbf{x}}^{neg}_{min}, \bar{\mathbf{x}}^{neg}_{max})$ draws from a uniform distribution between $\bar{\mathbf{x}}^{neg}_{min}$, $\bar{\mathbf{x}}^{neg}_{max}$. We can then train the neural network by optimizing f_θ with respect to parameters θ, and query the trained neural network to predict the occupancy.

We can also regress a neural network f_θ to assign multi-dimensional continuous properties to spatial coordinates. For example, we can assign colour values: The pointmaps from the foundation model, $\{X_1, \ldots, X_N\}$, correspond pixel-wise to input images. We can therefore obtain an RGB colour label for each point, giving us a dataset $\{\bar{\mathbf{x}}_i, \mathbf{y}^{rgb}_i\}_{i=1}^{N_{pc}}$. Then, we can apply a positional encoding ϕ and optimize:

$$L(\theta) = MSELoss\Big(\{f_\theta(\phi(\bar{\mathbf{x}}_i))\}_{i=1}^{N_{pc}}, \{\mathbf{y}^{rgb}_i\}_{i=1}^{N_{pc}}\Big). \quad (4)$$

After training, we can check for occupied regions and predict the colour assigned to them via a forward pass of f_θ.

		Light Tabletop (8 items)			Light Tabletop (7 items)			Dark Tabletop		
	Images Provided	10 images	12 images	15 images	10 images	12 images	15 images	10 images	12 images	15 images
Ours	Converged	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Residual $\delta_{\mathbf{t}}$	0.0420	0.0419	0.0396	0.0208	0.0317	0.0357	0.0310	0.0536	0.0414
	Residual δ_R	0.0655	0.0657	0.0513	0.0519	0.0623	0.0701	0.0732	0.0742	0.0818
	No. of Poses	10	12	15	10	12	15	10	12	15
COLMAP [17]	Converged	×	×	×	✓	✓	✓	×	×	✓
	Residual $\delta_{\mathbf{t}}$	NA	NA	NA	0.0412	0.0412	0.0469	NA	NA	0.0454
	Residual δ_R	NA	NA	NA	1.27	1.27	0.0662	NA	NA	0.0503
	No. of Poses	2	2	2	5	5	10	4	4	10

TABLE I: We evaluate our JCR against estimating camera poses with COLMAP and then run calibration. We observe that, especially when the number of images is low, COLMAP can only estimate a very few number of camera poses, which results in divergence and large residuals. Our method is able to accurately run hand-eye calibration even when a low number of images are provided.

(a) Light Tabletop 8 items (b) Light Tabletop 7 items (c) Dark Tabletop

Fig. 1: Top Row: Examples of images taken by our manipulated-mounted camera. Bottom Row: Environment representations built with JCR. We visualize by sampling points at regions with predicted high occupancy and assign the colours predicted by the representation at these points. The representation is in the coordinate frame of the robot with physically-accurate scales recovered.

V. Empirical Evaluations

In this section, we evaluate the quality of the our Joint calibration and Representation (JCR) method to both calibrate the camera with respect to the manipulator end-effector as well as to build environment representations. We attach an inexpensive USB webcam, which captures low-resolution RGB images, onto a Unitree Z1 manipulator. Here, the questions we seek to answer are (1) Can JCR, with foundation models, enable *image efficient* hand-eye calibration? (2) Can high-quality representations be learned with JCR?

Hand-Eye Calibration with JFR: Hand-eye calibration requires the determination of relative camera poses. Historically, this has been done via artificial external markers such as checkerboards or Apriltags [2], which are highly feature-rich and easy to identify. In the absence of such markers, Structure-from-Motion (SfM) methods, such as COLMAP [17], are typical alternative approaches to estimate relative camera poses. We take images in 3 different environments, two of these are table-top scenes on a light-coloured table with 8 and 7 items respectively, along with a scene on a dark table. We evaluate with an increasing number of input images, then check the calibration convergence and the residual values of eq. (2). We report the $L2$ norm of the translation-term residuals $\delta_{\mathbf{t}}$ and the Frobenius norm of the rotation-term residuals δ_R.

We compare against running hand-eye calibration on camera poses estimated from COLMAP, across three different scene. We are interested to investigate the behaviour of both methods when the number of images in low, and run the methods on image sets of size 10, 12 and 15. We tabulate the results in table I. As COLMAP relies on matching consistent features, when the number of provided images is low many camera poses cannot be found, resulting in divergence during calibration. JCR leverages foundation models to predict the correspondence and can consistently estimate all camera poses. As a result JCR is more *image-efficient*, allowing for calibration even with a small set of images.

Environment Representation: We model the occupancy and colour of each environment using neural networks with one hidden layer of size 256. Each representation can be trained to convergence within 15 seconds on a laptop with NVIDIA RTX 3070 GPU. We sampled points at where occupancy is high and visualize the points with their predicted colours. We observe that JCR can construct accurate and dense representations from small sets of RGB images, in the absences of depth information. Example images taken by the manipulator-mounted camera, along with visualizations of our representations are in fig. 1.

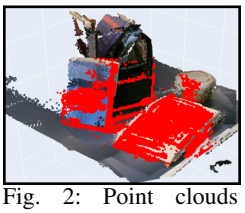

Fig. 2: Point clouds used to train JCR (in colour), produced by the upstream foundation model, is much denser than that by COLMAP, even after densification (overlaid in red).

JCR leverages foundation models to extract much denser correspondences than traditional SfM methods, which rely on visual feature-matching. In fig. 2, we overlay the point clouds produced by COLMAP, after its built-in densification, onto points produced by the foundation model. We observe that COLMAP cannot produce dense points over smooth surfaces such as the tabletop. It can only identify regions that correspond to edges with contrast, such as the text in the open book. The dense outputs of the foundation model enables us to jointly calibrate the camera and map the environment.

REFERENCES

[1] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Trans. Robotics Autom.*, 1988.

[2] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *IEEE International Conference on Robotics and Automation*, 2011.

[3] R. Bommasani and et al., "On the opportunities and risks of foundation models," *CoRR*, 2021.

[4] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal on Robotics and Automation*, 1987.

[5] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[6] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[7] S. O'Callaghan, F. T. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *2009 IEEE International Conference on Robotics and Automation*, 2009.

[8] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *International Journal Robotics Research*, 2016.

[9] W. Zhi, R. Senanayake, L. Ott, and F. Ramos, "Spatiotemporal learning of directional uncertainty in urban environments with kernel recurrent mixture density networks," *IEEE Robotics and Automation Letters*, 2019.

[10] W. Zhi, L. Ott, R. Senanayake, and F. Ramos, "Continuous occupancy map fusion with fast bayesian hilbert maps," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[11] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning (CoRL)*, 2017.

[12] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[15] 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," in *ECCV*, 2020.

[16] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, 2022.

[17] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] R. Horaud and F. Dornaika, "Hand-eye calibration," *I. J. Robotic Res.*, 1995.

[19] F. Park and B. Martin, "Robot Sensor Calibration: Solving AX = XB on the Euclidean Group," *IEEE Transactions on Robotics and Automation*, 1994.

[20] E. Valassakis, K. Dreczkowski, and E. Johns, "Learning eye-in-hand camera calibration from a single image," in *Proceedings of the 5th Conference on Robot Learning*, 2022.

[21] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems*, 2021.

[22] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning (CoRL)*, 2021.

[23] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning,"

[24] T. Lai, W. Zhi, T. Hermans, and F. Ramos, "Parallelised diffeomorphic sampling-based motion planning," in *Conference on Robot Learning (CoRL)*, 2021.

[25] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *CoRR*, 2023.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *CoRR*, 2021.

[27] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," *CoRR*, 2023.

[28] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," 2015.

[29] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Conference on Artificial Intelligence and Statistics*, 2010.

[30] C. M. Bishop, *Pattern recognition and machine learning, 5th Edition*. Information science and statistics, Springer, 2007.