2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Tracking Snake-Like Robots in the Wild Using Only a Single Camera

Jingpei Lu[1], Florian Richter[1], Shan Lin[1], and Michael C. Yip[1]

***Abstract*— Robot navigation within complex environments requires precise state estimation and localization to ensure robust and safe operations. For ambulating mobile robots like robot snakes, traditional methods for sensing require multiple embedded sensors or markers, leading to increased complexity, cost, and increased points of failure. Alternatively, deploying an external camera in the environment is very easy to do, and marker-less state estimation of the robot from this camera's images is an ideal solution: both simple and cost-effective. However, the challenge in this process is in tracking the robot under larger environments where the cameras may be moved around without extrinsic calibration, or maybe when in motion (e.g., a drone following the robot). The scenario itself presents a complex challenge: single-image reconstruction of robot poses under noisy observations. In this paper, we address the problem of tracking ambulatory mobile robots from a single camera. The method combines differentiable rendering with the Kalman filter. This synergy allows for simultaneous estimation of the robot's joint angle and pose while also providing state uncertainty which could be used later on for robust control. We demonstrate the efficacy of our approach on a snake-like robot in both stationary and non-stationary (moving) cameras, validating its performance in both structured and unstructured scenarios. The results achieved show an average error of 0.05 m in localizing the robot's base position and 6 degrees in joint state estimation. We believe this novel technique opens up possibilities for enhanced robot mobility and navigation in future exploratory and search-and-rescue missions.**

I. Introduction

Unlike their stationary counterparts, mobile robots are designed to navigate through the physical world in environments that are often too treacherous for humans such as the deep sea [1] and even other planets [2]. With mobile robots acting as surrogates for humans, exploration for research and search and rescue missions in extreme environments are conducted without risking human lives [3]. A growing class of mobile robots involves ambulatory systems. These ambulatory mobile robots (AMRs) have specialized articulated robotic designs for enhanced mobility and stability on uneven ground techniques in order to navigate broader terrains. AMRs include but are not limited to quadruped robots [4], flying drones [5], and snake-like and serpentine robots [6], [7].

To ensure the safe operation of AMRs in complex environments, various sensors are integrated into their systems. These sensors aid in localizing the robot and understanding its surroundings, though this can introduce increased complexity in real-world deployments. A more streamlined approach involves tracking AMRs using cameras. Cameras, given their ease of installation and portability, are better for navigating challenging terrains. For example, in the Mars 2020 NASA mission, where the Mars Helicopter utilized onboard cameras to scout the landscape and guide the Perseverance rover's exploration. As we look to the future, exploratory and search-and-rescue missions likely involve collaborative efforts between multiple robots, and the ability to track one robot using a camera mounted on another will be crucial.

Fig. 1: A snake-like robot, Arcsnake [7], is tracked on camera in the outdoor environment by a hovering drone.

In this paper, we address the problem of tracking snake-like robots from a single camera. Along the lines of the Mars Helicopter's mission, we aim to bring robot state estimation from camera data to snake-like robots, and by extension, other AMRs, to aid in future exploratory missions. By estimating the pose and state of an AMR, drones can provide more detailed guidance when providing mapping of the environment [8]. Our focus is on snake robots that draw inspiration from biological snakes [9] and are currently funded by NASA for exploration on extraterrestrial planetary bodies [10]. Toward this end, we recognize a fundamental need for being able to track AMRs using only a monocular camera. These techniques will also become foundational in the future to deploying robots in search-and-rescue missions or leveraging autonomous robot teams for work in the remote wilderness.

The overall tracking approach involves first a method for automatic robot mask generation. Leveraging this mask, we present a tracking technique that seamlessly integrates

[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA.{jil360, frichter, shl102, yip}@ucsd.edu

differentiable rendering with the Kalman filter, ensuring precise online state estimation. We conduct experiments in both laboratory and outdoor environments (Fig. 1). Through both qualitative and quantitative evaluations, we demonstrate the effectiveness of our method in different scenarios. Our contributions are threefold:

- We present the first work on marker-less state estimation for a snake robot from a single monocular camera.
- Our method combines differentiable rendering with a Kalman filter, and simultaneously estimates the joint angle and the pose of a snake robot.
- Validation of the effectiveness of the algorithm on a snake robot in both structured and unstructured environments, achieving a localization accuracy of 0.05 m for the robot base position and 0.11 rad on the robot's joint states.

II. Previous Work

A. Robot Localization from Single Camera

Localizing the robot is crucial for a wide range of robotic applications, especially when relying on a single camera, which presents unique challenges. One popular approach to address this is using the fiducial markers as 2D point features [11], [12]. For articulated robots like a snake robot, the 3D position of the markers can be calculated using robot kinematics and the robot pose can be derived by solving a Perspective-n-Point problem [13], [14], [15], [16].

As the field evolved, there was a shift towards marker-less pose estimation. Initial efforts in this direction utilized depth cameras to localize articulated robots [17], [18], [19], [20]. With the rise of Deep Neural Networks (DNNs), a new paradigm emerged. DNNs, with their advantages of extracting point features without the need for markers, have significantly enhanced the performance of marker-less pose estimation for articulated robots [21], [22], [23], [24], [25]. Beyond keypoint-based methods, recent works [26], [27] have demonstrated the potential of rendering-based methods. Benefiting from the dense correspondence provided by robot masks, rendering-based methods achieve state-of-the-art performance on robot pose estimation. However, they suffer from processing speed.

In this work, we adopt a rendering-based approach for robot state estimation. Instead of purely relying on the rendering, we integrate image moments with a Kalman Filter, aiming to utilize temporal information to achieve precise and fast online inference using a single camera.

B. Snake Robot State Estimation

For a broader category of mobile robots, the primary focus of state estimation has been on localizing the robot within its surroundings. For instance, Milella et al. [28] utilizes visually distinctive features on stereo images for localization. Several other works [29], [30], [31] have proposed methods that take into account the environment dynamics and potential measurement errors to enhance localization accuracy.

However, in the realm of snake robots, state estimation becomes even more intricate due to the need to consider joint angles for accurate 3D space modeling. Historically, state estimation for snake robots has relied on the robot's internal proprioceptive sensors, as highlighted by works like Rollinson et al. [32], [33]. Then, the filtering methods, like the Unscented Kalman Filter and Extended Kalman Filter [34], [35], have been employed to account for the measurement error for real-time estimation.

In this work, we seek to estimate both the position and joint angle of the snake robot using only images. This approach not only simplifies the estimation process but also enhances the robot's adaptability in outdoor scenarios.

Algorithm 1: Online State Estimation

Input : Initialized robot state $\mathbf{x}_{0|0}, \Sigma_{0|0}$
Output: Estimated robot state $\mathbf{x}_{t|t}, \Sigma_{t|t}$

while *receive new image* $\mathbb{I}_t$ **do**
- `// Motion Model`
- $\mathbf{x}_{t|t-1}, \Sigma_{t|t-1} \leftarrow motionModel(\mathbf{x}_{t-1|t-1}, \mathbf{v}_{t-1}, \Sigma_{t-1|t-1})$
- `// Observation from Image`
- $\mathbb{M}_t^{ref} \leftarrow f_{seg}(\mathbb{I}_t)$
- $\mathbf{m}_t \leftarrow computeMoments(\mathbb{M}_t^{ref})$
- `// Observation Model`
- $\mathcal{M}_{t|t-1} \leftarrow reconstructMesh(\mathbf{x}_{t|t-1})$
- $\mathbb{M}_{t|t-1}^{pred} \leftarrow renderPrediction(\mathbf{x}_{t|t-1}, \mathcal{M}_{t|t-1})$
- $\hat{\mathbf{m}}_t \leftarrow computeMoments(\mathbb{M}_{t|t-1}^{pred})$
- $H_t = \frac{\partial \hat{\mathbf{m}}_t}{\partial \mathbf{x}_{t|t-1}}$
- `// Compute the Residual`
- $\mathbf{y}_t = \mathbf{m}_t - \hat{\mathbf{m}}_t$
- `// Update Belief`
- $K_t = \Sigma_{t|t-1} H_t^\top (H_t \Sigma_{t|t-1} H_t^\top)^{-1}$
- $\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + K_t \mathbf{y}_t$
- $\Sigma_{t|t} = (I - K_t H_t)\Sigma_{t|t-1}$
- `// Refine with Image Loss`
- **for** *number of refinement steps* **do**
  - $\mathcal{M}_{t|t} \leftarrow reconstructMesh(\mathbf{x}_{t|t})$
  - $\mathbb{M}_{t|t}^{pred} \leftarrow renderPrediction(\mathbf{x}_{t|t}, \mathcal{M}_{t|t})$
  - $\mathcal{L}_t \leftarrow computeLoss(\mathbb{M}_{t|t}^{pred}, \mathbb{M}_t^{ref})$
  - $\mathbf{x}_{t|t} = \mathbf{x}_{t|t} - \lambda \frac{\partial \mathcal{L}_t}{\partial \mathbf{x}_{t|t}}$
- `// Update Velocity`
- $\mathbf{v}_t \leftarrow computeVelocity(\mathbf{x}_{t|t}, \mathbf{x}_{t-1|t-1})$

III. Methodology

The overall proposed approach follows an online state estimation method combining differentiable rendering of a robot mask, with image moment prediction, a robot motion model, and a Kalman filter to estimate the joint angle and the pose of a mobile robot from a single camera. The method includes, additionally, refinement steps and velocity update steps to enhance the accuracy of the estimation, as well as model transfer techniques to reduce computation and memory costs so that the method can run on modest hardware. The details

follow in the next section, and Algorithm 1 outlines the main steps of the method.

A. Motion Model with Belief Propagation

For AMR navigation, the robot state, denoted by $\mathbf{x}_t$, can encapsulate various attributes such as joint angles, camera-to-robot transformations, and other necessary parameters at time t. In this work, we define the robot state as $\mathbf{x} := [\theta, \mathbf{q}, \mathbf{b}]$, where $\theta \in \mathbb{R}^N$ is the robot joint angle (N is the number of joints), $\mathbf{q}$ is the quaternion, and $\mathbf{b}$ is the translational vector for the first link of the robot. The quaternion and the translational vector are parametrizations of the $\mathbf{T}_b^c \in SE(3)$, which is the robot pose in the camera frame.

The next state of the robot is predicted with a motion model, based on its previous state and velocity. This prediction phase provides a rough direction for belief propagation. We will model the robot's motion using a simple linear relationship:

$$\mathbf{b}_{t|t-1} = \mathbf{b}_{t-1|t-1} + \mathbf{v}_{t-1}\Delta t \tag{1}$$

where we try to predict the position of the robot $\mathbf{b}_{t|t-1}$ at time t by considering the previous robot position $\mathbf{b}_{t-1|t-1}$, the velocity $\mathbf{v}_{t-1}$, and the time step Δt. We will make the assumption that there is negligible process noise (i.e., imperfections in the system's motion model are negligible as compared to observation noise), leading to the following expression for the propagation of the covariance matrix:

$$\Sigma_{t|t-1} = F_t \Sigma_{t-1|t-1} F_t^\top \tag{2}$$

In this case, F_t is the identity matrix, reflecting our assumption that the motion model follows a linear relationship without any non-linear or stochastic effects.

B. Automatic Mask Generation for Segmentation

The proposed state estimation algorithm requires segmenting the robot from images, but manually labeling the robot masks can be highly time-consuming. Recently, the zero-shot generalizable segmentation model, Segment Anything Model (SAM) [36], allows automatic robot mask generation with simple bounding box prompts.

Given the binary robot mask of the previous frame, $\mathbb{M}_{t-1} \in \mathbb{R}^{H \times W}$, the bounding box prompt for the current frame, $\mathcal{B}_t := (u_{min}, v_{min}, u_{max}, v_{max})$, is estimated by a mask-to-box operation,

$$(u_{min}, v_{min}) = \min\{(u, v) \,|\, \mathbb{M}_{t-1}[u, v] \neq 0\} \tag{3}$$

$$(u_{max}, v_{max}) = \max\{(u, v) \,|\, \mathbb{M}_{t-1}[u, v] \neq 0\} \tag{4}$$

Then, the SAM is utilized to generate the robot mask of the current frame, given the bounding box prompt $\mathcal{B}_t$, as shown in Fig. 2. To ensure the robustness of the bounding box prompt, the robot mask is dilated before performing the mask-to-box operation.

Using SAM for robot mask generation can, however, be slow as SAM is not optimized for real-time application (around 0.5 seconds per frame using a single Nvidia GeForce RTX 4090 GPU). To achieve real-time performance, we utilize the robot masks generated from SAM to train a lightweight neural network for segmentation. Specifically, we employ DeepLabV3+ [37], a popular semantic segmentation architecture, to segment the robot from RGB images during the online estimation process. By training DeepLabV3+ with the generated masks, we ensure that our system can segment the robot in real-time with modest memory and computation requirements, effectively enabling realistic deployment in the wild.

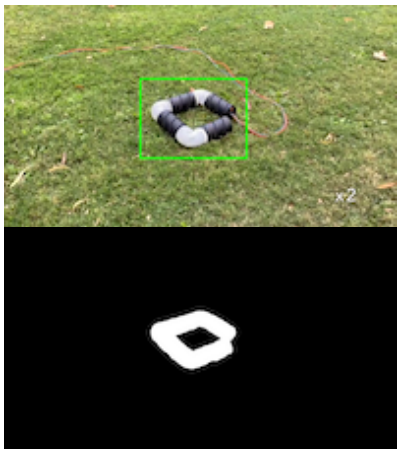

Fig. 2: Example of the bounding box prompt generated by mask-to-box operation (top) and the corresponding robot mask generated using SAM (bottom).

C. Observation Model for Belief Propagation

In this section, we introduce the mapping from the predicted robot states $\mathbf{x}_{t|t-1}$ to the observation of image moment [38] $\hat{\mathbf{m}}_t$ in the proposed algorithm 1.

Given the predicted robot states $\mathbf{x}_{t|t-1}$, which includes joint angle and robot pose, we first reconstruct the robot mesh by interconnecting individual robot body parts through forward kinematics. For a snake-like (serpentine) robot, we approximate each individual robot body part as a cylinder with the dimension mentioned in [39], [7]. Given a mesh vertex $\mathbf{r}^n \in \mathbb{R}^3$ on the n-th robot link, this vertex undergoes a transformation into the robot base frame considering the joint angle:

$$\bar{\mathbf{r}}^b = \mathbf{T}_n^b(\theta)\bar{\mathbf{r}}^n \tag{5}$$

where $\bar{\cdot}$ represents the homogeneous representation of a point (i.e. $\bar{\mathbf{r}} = [\mathbf{r}, 1]^T$), and $\mathbf{T}_n^b(\theta)$ is the coordinate frame transformation obtained from the forward kinematics [40].

Having the reconstructed robot mesh and the predicted robot base-to-camera transformation, $\mathbf{T}_b^c$, the PyTorch3D differentiable renderer [41] comes into play to produce a virtual-model-derived, or rendered robot mask. By referencing techniques similar to those in [27], a differentiable silhouette renderer paired with a perspective camera is employed. The *SoftSilhouetteShader* is specifically leveraged to compute pixel values that form the robot mask.

With the rendered robot mask, $\mathbb{M}$, the image moments

become computable as:

$$M_{ij} = \sum_u \sum_v u^i v^j \mathbb{M}(u, v) \tag{6}$$

Then, we derive the centroid, which is our observation for belief propagation, by:

$$\hat{\mathbf{m}} = \begin{bmatrix} \frac{M_{10}}{M_{00}} & \frac{M_{01}}{M_{00}} \end{bmatrix}^\top \tag{7}$$

We employ pytorch autograd [42] to track the gradient of each step and compute the observation matrix H by collecting the derivatives of the image moment $\hat{\mathbf{m}}$ with respect to the robot states $\mathbf{x}_{t|t-1}$.

Finally, an Extended Kalman Filter (EKF) [34] is employed to update the belief of the robot states (lines 9-12 in Algorithm 1), which ensures that our belief about the robot states is continually refined as more observations come in.

D. Image Loss Refinement and Velocity Estimation

While image moments have historically proven useful in object tracking [38], [43], their efficacy diminishes in the complex arena of robot state estimation. This is because they encapsulate only limited details of the robot mask. Consequently, a direct method that compares the estimated and reference robot masks provides an enhancement to state estimation accuracy.

We predict the robot mask from estimated robot states using the same differentiable rendering pipeline as described in Section III-C. To measure the difference between this prediction and the reference mask, we employ an image loss function, which sums the squared differences between the predicted mask $\mathbb{M}^{pred}$ and the reference mask $\mathbb{M}^{ref}$ across the image dimensions:

$$\mathcal{L} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \left(\mathbb{M}^{pred}(i,j) - \mathbb{M}^{ref}(i,j) \right)^2. \tag{8}$$

We refine the mean of the robot states by applying back-propagation on this image loss (line 17 in Algorithm 1), bringing the estimation closer to the true state.

As a final step, in service of the next belief propagation timestep, we derive the velocity from the updated position:

$$\mathbf{v}_t = \frac{\mathbf{b}_{t|t} - \mathbf{b}_{t-1|t-1}}{\Delta t} \tag{9}$$

This velocity is used for the motion model in forthcoming iterations, as it feeds into predictions for the robot's future states.

IV. Experiments and Results

To comprehensively assess the efficacy of our proposed state estimation algorithm, we collected datasets of a snake robot operating in both structured and unstructured environments. These datasets facilitated both qualitative and quantitative evaluations of the state estimation method.

The snake robot hardware is described in [39], [7] and is the evolutionary precursor to the NASA Extant Exobiology Life Surveyor (EELS) robot [10] that is anticipated to serve a science research vehicle for both earth science missions as well as extraterrestrial planetary exploration on Saturn's moon, Enceladus, or Jupiter's moon, Europa.

Snake-Lab Dataset: We introduced the Snake-Lab Dataset for evaluating the accuracy of the joint angle estimation and robot pose estimation. This dataset was acquired in a lab setting using an Intel® Realsense™ camera at a resolution of (1280, 720). The robot's joint angles were recorded using electromagnetic sensors and were synchronized with the captured images. Additionally, the robot's spatial position was determined using the depth capabilities of the camera. For evaluation metrics, we employed the Euclidean distance for position estimation and the L_1 norm for joint angle estimation.

Snake-Outdoor Dataset: To examine the robustness of our algorithm in less structured environments, we collected the Snake-Outdoor dataset. This dataset comprises three videos: the first two were recorded using a hand-held camera at a resolution of (1280, 720), while the third was captured via a drone camera, which has no direct connection to the snake robot system. Given the absence of ground truth for the robot's state in this setting, we adopted the Intersection-over-Union metric (IoU):

$$\text{IoU} = \frac{|\mathbb{M}^{ref} \cap \mathbb{M}^{pred}|}{|\mathbb{M}^{ref} \cup \mathbb{M}^{pred}|} \tag{10}$$

to compare the ground-truth robot mask $\mathbb{M}^{ref}$ with our algorithm's estimated mask $\mathbb{M}^{pred}$.

A. Implementation Details

To train DeepLabV3+, we collected around 1500 images, captured at a resolution of (1280, 720) and the ground truth segmentation masks were generated using Segment Anything Model [36]. We used the Adam optimizer [44] for gradient descent with 20 epochs and 8 batch size. The initial learning rate was set to 0.0001 and was decayed by a factor of 0.1 at the 10th epoch.

During the online estimation, we resize the raw image to a resolution of (640, 360). Both the observed robot mask and the rendered robot mask are processed at this resolution. For the refinement step, we set the learning rate to 0.005 and also used the Adam optimizer for gradient descent. All computational experiments were executed on a system equipped with an Intel® Core™ i9-11900F Processor and NVIDIA GeForce RTX 4090. To strike a balance between accuracy and processing speed, we perform 10 refinement iterations for each incoming image, ensuring optimal performance while sustaining an estimation speed of 1 FPS.

TABLE I: Average Position and State Estimation Error on Snake-Lab Dataset

	Position error (m)	Joint state error (rad)
Static	0.0278	0.0605
Moving camera	0.0647	0.0849
Moving robot	0.0587	0.1352
Overall	0.0540	0.1125

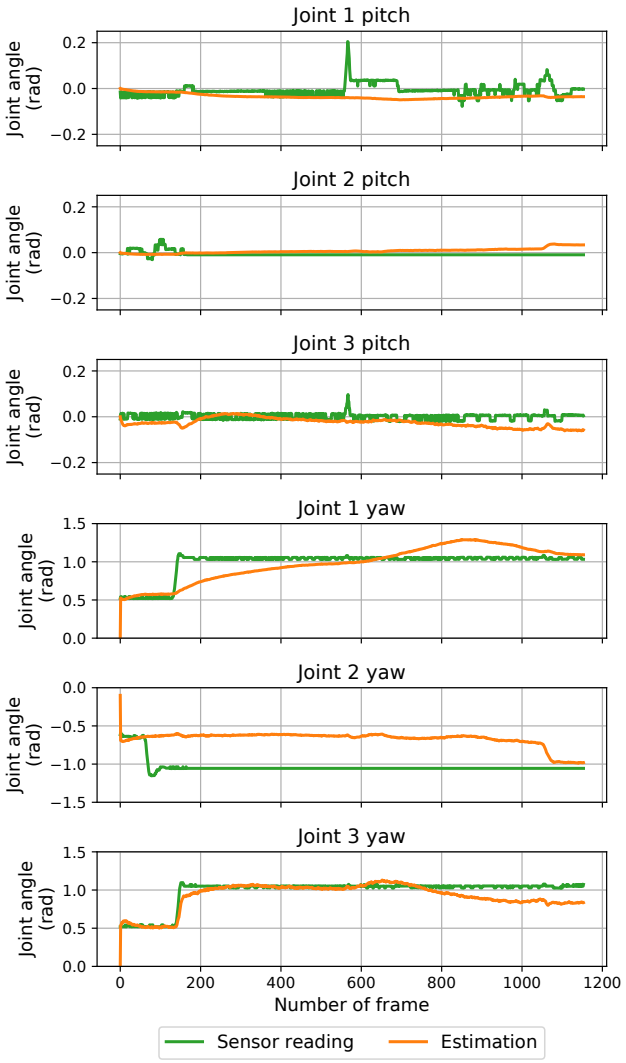

Fig. 3: Plots of estimated joint trajectory vs. sensor reading for the Snake-Lab dataset in the moving robot scenario. For each joint, we plot the pitch and yaw angle separately. Note that the snake robot uses magnetic encoders for sensor readings and are slightly noisy due misalignment between the encoder and magnet from vibrations during the experiment.

B. Experiment on Snake-Lab dataset

We present the qualitative results on the Snake-Lab dataset in Fig. 4, and the quantitative evaluation of our state estimation algorithm is presented in Table I. We also plot the estimated joint trajectory with sensor readings in Fig. 3.

The results are segmented based on different scenarios: static conditions, moving camera, and moving robot. Under static conditions, where both the camera and the robot remain stationary, both the joint angle error and position error are the lowest, indicating that the algorithm performs exceptionally well in stable environments. Moving the camera or robot slightly affects the algorithm's accuracy. This could be attributed to the dynamic nature of the camera and the robot's movements, which might introduce complexities in state estimation. The overall average position error and joint angle error across all scenarios are 0.0540 m and 0.1125 rad, respectively. These results affirm the robustness of our state estimation algorithm, even in varying conditions. However, it's evident that dynamic factors, such as camera or robot movement, introduce some challenges, leading to increased errors.

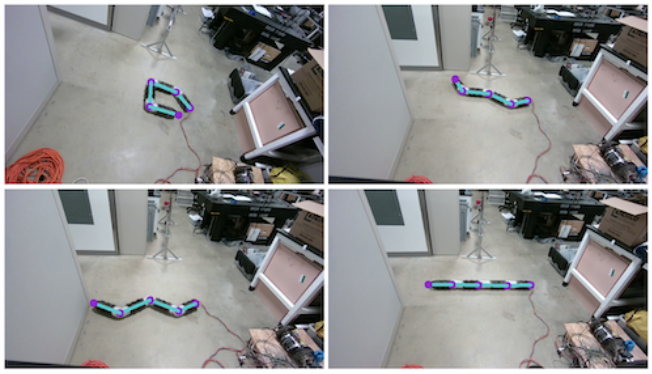

Fig. 4: Qualitative results on Snake-Lab dataset. We derive the skeleton from the estimated robot pose and joint angle, and visualize it by projecting the skeleton on images.

C. Experiment on Snake-Outdoor dataset

Table II presents the quantitative evaluation of our state estimation algorithm on the Snake-Outdoor dataset. The results are organized based on the number of refinement steps taken, which are 1, 5, and 10. The performance metric used is the Intersection-over-Union (IoU) for each video, and the speed of the algorithm in frames per second (FPS) is also provided. From the Table II, we can see a clear trade-off between accuracy and speed. As the number of refinement steps increases, there is a noticeable improvement in the Mean IoU, but the speed decreases. With 10 refinement steps, the algorithm operates at 1 FPS, which might be a limiting factor for real-time applications. However, the significant boost in accuracy might justify this trade-off in scenarios where precision is critical.

We also present qualitative results in Fig 5, showing the estimated skeleton and the predicted robot mask overlaid on the images. We can observe the estimated skeleton aligns

TABLE II: Quantitative evaluation on Snake-Outdoor dataset. We compute the IoU between the estimated robot mask and the ground-truth robot mask. We also report the processing speed under different settings.

	Number of refinement steps		
	1	5	10
Video 1 (Mean IoU)	0.4659	0.7632	0.8665
Video 2 (Mean IoU)	0.2456	0.3584	0.7690
Video 3 (Mean IoU)	0.3088	0.4394	0.8210
Speed (FPS)	3.5	1.5	1

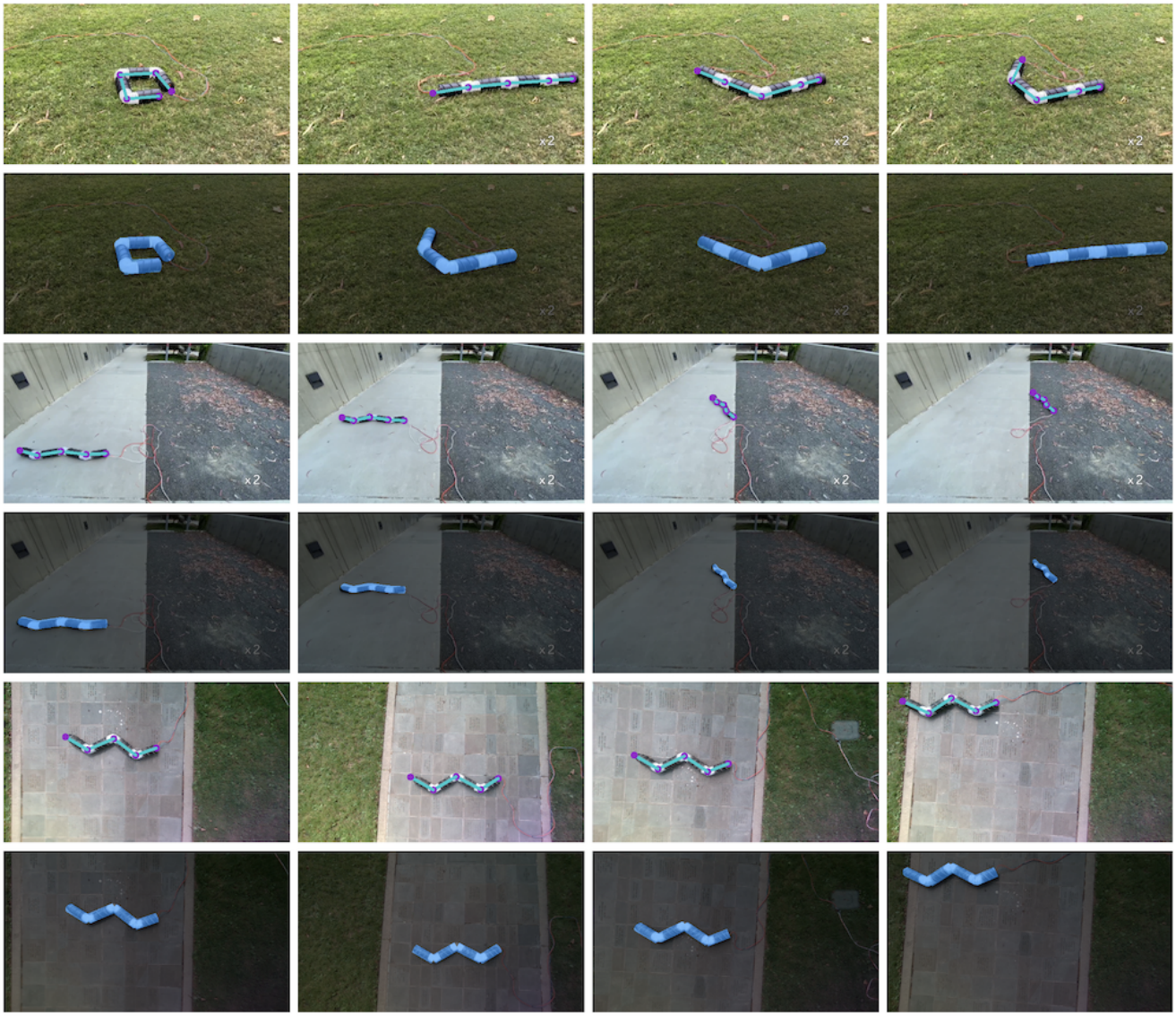

Fig. 5: Qualitative results on Snake-Outdoor dataset. We show the estimated skeleton and predicted robot mask overlaid on images. Rows 1-2 correspond to video 1, rows 3-4 correspond to video 2, and rows 5-6 correspond to video 3. Notably, there's a precise alignment of the skeleton and mask with the robot as shown in the images.

with the robot's actual structure, providing a clear and intuitive understanding of the algorithm's performance in real-world, outdoor settings.

V. Conclusion

In this work, we present a novel method for state estimation of snake robots using a single camera. The proposed approach combines differentiable rendering with the Kalman filter, fusing temporal information with a rendering-based optimization technique to improve the estimation process, which enhances the method's adaptability in outdoor scenarios. The results demonstrate the efficacy of our approach on a snake robot, validating its performance in both structured and unstructured environments. We believe this technique opens up possibilities for expanded capabilities for ambulatory mobile robot deployment and navigation in complex environments, making it a promising solution for future mobile robot applications.

For future works, an exciting avenue is the exploration of how our method can be adapted for collaborative robotics, where multiple robots work in tandem. This could involve state estimation in scenarios where robots share sensory data to navigate or perform tasks (e.g. drone-assisted routing in different landscapes).

Acknowledgement

We thank Professor Nikolay Atanasov and Jason Stanley from the Existential Robotics Laboratory at UCSD for his assistance with the drone experiments, and NASA Jet Propulsion Laboratory for their continued mission guidance.

REFERENCES

[1] C. Kunz *et al.*, "Deep sea underwater robotic exploration in the ice-covered arctic ocean with auvs," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3654–3660, IEEE, 2008.

[2] S. W. Squyres *et al.*, "Athena mars rover science investigation," *Journal of Geophysical Research: Planets*, vol. 108, no. E12, 2003.

[3] J. Whitman, N. Zevallos, M. Travers, and H. Choset, "Snake robot urban search after the 2017 mexico city earthquake," in *2018 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pp. 1–6, IEEE, 2018.

[4] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252, IEEE, 2018.

[5] A. Aabid, B. Parveez, N. Parveen, S. A. Khan, J. Zayan, and O. Shabbir, "Reviews on design and development of unmanned aerial vehicle (drone) for different applications," *J. Mech. Eng. Res. Dev*, vol. 45, no. 2, pp. 53–69, 2022.

[6] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, "Design of a modular snake robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2609–2614, IEEE, 2007.

[7] F. Richter, P. V. Gavrilov, H. M. Lam, A. Degani, and M. C. Yip, "Arcsnake: Reconfigurable snakelike robot with archimedean screw propulsion for multidomain mobility," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 797–809, 2021.

[8] L. von Stumberg, V. Usenko, J. Engel, J. Stückler, and D. Cremers, "From monocular slam to autonomous drone exploration," in *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–8, IEEE, 2017.

[9] K. Y. Pettersen, "Snake robots," *Annual Reviews in Control*, vol. 44, pp. 19–44, 2017.

[10] K. Carpenter, A. Thoesen, D. Mick, J. Martia, M. Cable, K. Mitchell, S. Hovsepian, J. Jasper, N. Georgiev, R. Thakker, A. Kourchians, B. Wilcox, M. Yip, and H. Marvi, *Exobiology Extant Life Surveyor (EELS)*, pp. 328–338. ASCE Library, 2021.

[11] S. Garrido-Jurado *et al.*, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[12] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*, pp. 3400–3407, IEEE, 2011.

[13] F. C. Park and B. J. Martin, "Robot sensor calibration: solving AX= XB on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[14] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation ax= xb," *Journal of Robotic Systems*, vol. 22, no. 9, pp. 497–506, 2005.

[15] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 67–74, IEEE, 2011.

[16] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

[17] T. Schmidt, R. A. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking.," in *Robotics: Science and Systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014.

[18] K. Pauwels, L. Rubio, and E. Ros, "Real-time model-based articulated object pose detection and tracking with variable rigidity constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4001, 2014.

[19] F. Michel, A. Krull, E. Brachmann, M. Y. Yang, S. Gumhold, and C. Rother, "Pose estimation of kinematic chain instances via object coordinate regression.," in *BMVC*, pp. 181–1, 2015.

[20] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Factored pose estimation of articulated objects using efficient nonparametric belief propagation," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7221–7227, IEEE, 2019.

[21] J. Lambrecht and L. Kästner, "Towards the usage of synthetic data for marker-less pose estimation of articulated robots in rgb images," in *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 240–247, IEEE, 2019.

[22] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9426–9432, IEEE, 2020.

[23] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4622–4629, 2022.

[24] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille, "Craves: Controlling robotic arm with a vision-based economic system," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4214–4223, 2019.

[25] J. Lu, F. Richter, and M. C. Yip, "Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21296–21306, 2023.

[26] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render & compare," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1654–1663, 2021.

[27] J. Lu, F. Liu, C. Girerd, and M. C. Yip, "Image-based pose estimation and shape reconstruction for robot manipulators and soft, continuum robots via differentiable rendering," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 560–567, 2023.

[28] A. Milella, G. Reina, R. Siegwart, *et al.*, "Computer vision methods for improved mobile robot state estimation in challenging terrains.," *J. Multim.*, vol. 1, no. 7, pp. 49–61, 2006.

[29] D. Goldberg and M. J. Matarić, "Maximizing reward in a non-stationary mobile robot environment," *Autonomous Agents and Multi-Agent Systems*, vol. 6, pp. 287–316, 2003.

[30] L. Zouaghi, A. Alexopoulos, A. Wagner, and E. Badreddin, "Probabilistic online-generated monitoring models for mobile robot navigation using modified petri net," in *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 594–599, IEEE, 2011.

[31] E. Colle and S. Galerne, "A robust set approach for mobile robot localization in ambient environment," *Autonomous Robots*, vol. 43, pp. 557–573, 2019.

[32] D. Rollinson, A. Buchan, and H. Choset, "State estimation for snake robots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1075–1080, IEEE, 2011.

[33] D. Rollinson, H. Choset, and S. Tully, "Robust state estimation with redundant proprioceptive sensors," in *Dynamic Systems and Control Conference*, vol. 56147, p. V003T40A005, American Society of Mechanical Engineers, 2013.

[34] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," 1961.

[35] R. Van Der Merwe, E. A. Wan, S. Julier, *et al.*, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," in *Proceedings of the AIAA guidance, navigation & control conference*, vol. 3, p. 08, Providence, RI Providence, RI, 2004.

[36] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[37] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Europ. Conf. Comput. Vis.*, pp. 801–818, 2018.

[38] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.

[39] D. A. Schreiber, F. Richter, A. Bilan, P. V. Gavrilov, H. M. Lam, C. H. Price, K. C. Carpenter, and M. C. Yip, "Arcsnake: An archimedes' screw-propelled, reconfigurable serpentine robot for complex environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7029–7034, IEEE, 2020.

[40] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[41] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv preprint arXiv:2007.08501*, 2020.

[42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[43] G. Yao, R. Saltus, and A. Dani, "Image moment-based extended object tracking for complex motions," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6560–6572, 2020.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.