Robust Tube Localization for Mars Sample Return: Lightweight YOLO-Segmentation with Angle-Guided PnP

Daniel Posada[a,*], Tu-Hoa Pham[a], Nikos Mavrakis[b], Philip Bailey[a]

[a] *Jet Propulsion Laboratory California Institute of Technology, 4800 Oak Grove Dr, Pasadena, 91101, California, United States*
[b] *University of Birmingham, Birmingham Research Park, 97 Vincent Drive, Birmingham, B15 2SQ, United Kingdom*

Abstract

One considered approach in the planned Mars Sample Return (MSR) campaign involves accurately identifying and retrieving sample tubes from the Martian surface. This paper presents an innovative approach that utilises lightweight computer-vision techniques to enhance the efficiency and accuracy of the Sample Transfer Arm (STA) aboard the MSR lander. Our methodology employs the YOLOv8 deep learning model for image segmentation, and centroid detection of tubes in the challenging dusty Martian environment. These detected masks and centroids provide the foundation for constructing an outlined representation of the tubes, which is critical for precise spatial orientation. We exploit the knowledge of the object geometry to find key points and match them using their relative positions with respect to the geometry. Subsequently, a Perspective-n-Point (PnP) algorithm with RANSAC utilizes this outline and pre-computed 3D coordinates to ascertain the tube's pose. This enables the STA's camera-equipped gripper to locate and retrieve the samples accurately. This process is meticulously tailored for the constrained computational resources available on Martian missions, addressing limitations in processing speed and lack of parallelization capabilities. Extensive simulations under Martian-like conditions demonstrate the robustness and reliability of our approach, which would be a necessary technology to enable a backup tube retrieval concept for a MSR campaign using a robotic arm by

*Corresponding author
Email address: dposadac88@hotmail.com (Daniel Posada)

ensuring precise and efficient sample collection. This method can achieve sub-degree and sub-centimeter accuracy with a single image.
Keywords: Mars sample return, YOLOv8 image segmentation, PnP pose estimation, robotic manipulation.

---

1. Introduction

The quest for understanding Mars has been a pivotal aspect of space exploration in the 21st century. Among the various missions aimed at unraveling the mysteries of the Red Planet, the Mars Sample Return[1] (MSR) campaign stands out as a landmark endeavor, explicitly targeting the study of potential past life on Mars. This campaign, designed to collect and return samples from the Martian surface to Earth, promises to provide unprecedented insights into Martian geology and potential bio-signatures.

The MSR campaign is composed of several critical stages. Each one is integral to the successful return of Martian samples. The first stage of the multi-part campaign was the Perseverance rover which launched in 2020 and as of the writing of this paper has collected a number of samples, some of which were deposited on the Martian surface. The second part of this campaign is the Sample Return Lander (SRL). The lander is tasked with receiving Returnable Sample Tube Assemblies (RSTAs) and inserting them into the Orbital Sample (OS) canister. The design approach considered here utilizes a Sample Transfer Arm (STA), guided by an end-effector mounted STA Camera. The RSTAs are provided to SRL with a handling glove attached as an “RSTA Glove Assembly" or Returnable Glove Assembly (RGA) as it will be referred to in this paper. The primary method of transfer of RGAs to SRL would be via M2020, however a backup campaign concept is defined for RGAs to be delivered to the surface to be picked up by the STA. This backup campaign, whose Concep of Operations (conops)

[1]The decision to implement Mars Sample Return will not be finalized until NASA's completion of the National Environmental Policy Act (NEPA) process. This document is being made available for information purposes only. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

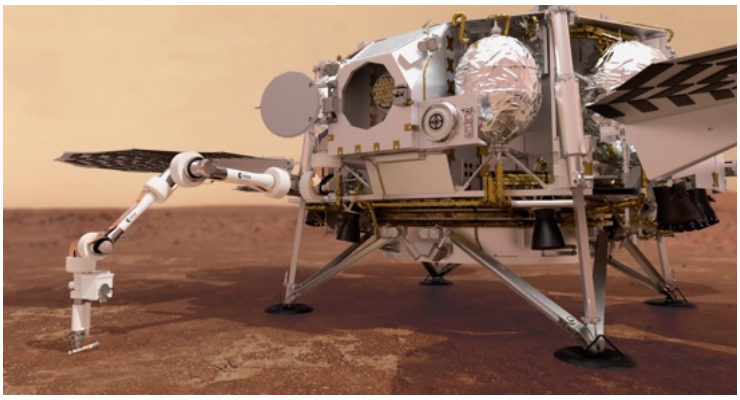


Figure 1: Image concept of the Sample Return Arm picking the RGA from the Martian surface. Courtesy of ESA/NASA.

and method of sample delivery is outside the scope of this paper, would interface with the lander by depositing an RGA on the surface of Mars in the STA's reachable area. This paper is primarily concerned with a proposed tentative set of conops for this transfer, which assumed the RGA will be dropped in a position such that the RGA will be visible in an STA Camera image. The notional requirements for the grasping are 1 cm lateral error, 1 cm of normal error, 5 degrees of tilt out of the ground plane, and 2 degrees of clocking in the ground plane. The third stage of the proposed campaign is a rocket that is tasked with launching the sample tubes into orbit called the Mars Ascent Vehicle (MAV), which has the filled OS at the tip of the rocket. In the fourth and final stage, these samples are transferred to an orbiter, which then positions itself to launch them back toward Earth on a carefully calculated trajectory.

As mentioned, a critical component of the proposed MSR campaign concept involves the precise and efficient retrieval of sample tubes by the STA aboard the MSR lander from the surface in case Perseverance fails, either fully or partially, due to wear over time. Figure 1 illustrates a concept of the STA robotic arm performing the pickup operation after identifying where the tube is on the surface. Operating in Mars's harsh and unpredictable environment, the STA must accurately locate, identify, and handle geological samples, primarily enabled by its dual redundant monocular greyscale 4-MegaPixel cameras (STACams), which are the primary method for aligning the STA to

targets for tube transfer. Traditional robotic manipulation and object retrieval methods are hindered due to the unique Martian terrain, variable lighting conditions, and the stringent payload and computational constraints of space missions. While the eventual conops are not fully defined, this work assumes that this localization will be required to be executed on-board the spacecraft in order to execute the campaign timeline. The on-board processor is assumed to provide $\approx$ 100 MHz of CPU available and $\approx$ 500 MB of RAM.

The objective of this research is to develop and implement an advanced state-of-the-art computer vision system that enhances the capabilities of the STA. By employing a recent lightweight deep-learning detector, specifically the YOLOv8 [1] network (which was the latest at the time when the testing was conducted) for image segmentation and object detection, our goal is to accurately determine the position and orientation of sample tubes on the Martian soil. This information is crucial for enabling the STA's robotic arm to retrieve these samples successfully.

We introduce introduces a novel geometric approach that integrates state-of-the-art computer vision techniques with the practical constraints of space exploration. We have created a new pipeline that exploits the knowledge of the localizable objects' geometry to reliably extract and associate key points to successfully find the pose of the RGA on the Martian surface with a single monocular image. We extract 2D key points from the perimeter of a baseline rendered image of the RGA, for which we know the associated 3D coordinates, and match those 3D key points with those extracted from a monocular test image. The key element for the association of key point mappings from the rendered and test image is a proposed angular mapping of the features to optimize the point correspondences and ensure a correct estimation. The resulting 2D-3D correspondences from this mapping are then passed through a RANSAC [2] based Perspective-n-Point (PnP) solver to extract the final pose of the tube [3].

The paper is structured as follows: Section 2 describes some related work on feature matching and learning-based computer vision and applications on different industries including space. Section 3 outlines the pipeline and algorithms. Section 4 describes the simulator, datasets, and experimental process. Section 4.3 provides results from the pipeline execution. Finally, sections 5 and 6 provide commentary on the results and

future applications.

Contributions:

Here is a summary of the contributions of this work:

- The development of a new pose estimation pipeline that leverages known geometry for improved accuracy in object localization and detection.
- A novel approach to cluttered texture rejection that enhances the robustness of template matching techniques in unstructured environments, a common challenge in our field.
- A systematic evaluation of our algorithm on an extensive dataset comprising a total of 1500 images acquired in varying conditions such as pose, lighting, and terrain.

2. Related Work

Finding objects in 3D space has been a common task for robotic arms in multiple domains. On-orbit satellite-servicing missions place similar—and often stricter—demands on robotic manipulators: a capture tool mounted at the arm's end-effector must be localised to within millimetres and a few degrees with respect to a non-co-operative client vehicle before docking or component exchange can proceed as illustrated in Figure 2.

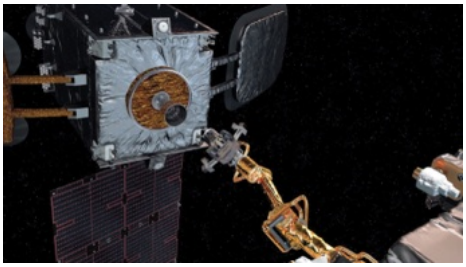


Figure 2: NASA's conceptual servicing spacecraft, extends its robotic arm to grab and refuel a satellite. [4]

Unlike factory pick-and-place systems that operate in structured, repeatable environments, space servicers must perform real-time pose estimation under uncontrolled

illumination, specular surfaces, and constrained compute budgets. Recent servicing demonstrators therefore integrate vision-based relative-navigation modules that identify the client spacecraft, segment key structural features (antenna booms, solar-panel corners, nozzle rims), and solve for 6-DoF pose using geometric or learning-based techniques. For example [5] introduces a transformer based architecture to estimate the pose using different probabilistic distributions to reduce and understand the pose uncertainty and correlated to a specific navigation state. Another example, in [6] exploits the overall geometry of the spacecraft by fitting bounding boxes using convolutional neural networks to extract a 3D model and respective states. This is key if another satellite with an end-effector is to provide a service and wants to interact with a specific component of the target spacecraft. Identifying the compoent itself is its own task that can be solved by training the end-effector as a pick-and-place machine using reinforcement learning to distinguish the object and optimize the trajectory [7, 8].

Accurate 6-DoF pose estimation is essential for any manipulator that must grasp or service a target object. The task has been central to computer-vision research since the field's inception, with early work addressing head-tracked virtual-reality displays [9]. A comprehensive survey traces the evolution of template-, feature-, and learning-based pipelines and formalises their algorithmic taxonomy [10]. A comparative analysis of PnP solvers under sparse key-point noise offers guidance for hardware-constrained manipulators such as the Sample Transfer Arm (STA) [11]. An analytic treatment of the efficient PnP algorithm quantifies the stability limits that arise when only four non-coplanar correspondences are available [12]. Real-time detectors that regress 3D bounding boxes are reviewed with an emphasis on latency–accuracy trade-offs [13]. A survey of open challenges highlights failure modes in texture-less or highly specular environments that resemble Martian sample tubes [14]. A separate overview benchmarks end-to-end deep networks against classical pipelines and discusses their memory footprints [15]. These studies converge on three canonical method families—template-based, feature-based, and learning-based—and collectively emphasise the need for solutions that operate with a single monocular camera, because the STA's dual imagers are mission-redundant and therefore unavailable for stereo processing.

Alternative depth sources are likewise impractical. Structure-from-motion would

require a stereo baseline generated by arm motion, but the baseline attainable within the STA workspace is too small to yield reliable depth, and error grows rapidly with target distance. Active-depth sensors such as LiDAR or RGB-D cameras are excluded by mass, and power constraints. Consequently, stereo-based or depth-cloud methods lie outside the scope of this study; the remainder of the paper focuses exclusively on monocular techniques that satisfy the STA's 100 MHz, 500 MB processing envelope.

2.1. Template-based pose estimation

This class of methods utilizes a template to detect a unique or specific pattern in an image. The template is usually a rendered silhouette of the CAD mesh, and it is accompanied by pre-computed 2D keypoints that serve as matching anchors. As new features are detected on a new image, a matching algorithm compares them to the closest externally computed key points and returns an estimation of the corresponding pose. However, template pipelines are sensitive to scale because they require thousands of rendered views. This overhead can be mitigated by using lower-resolution templates or by applying a multi-resolution search pyramid. In the publication [16] a technique for matching CAD models is explained. The method involves using edge detection on the model's image, creating hierarchical projected viewpoints, and then comparing them using an edge-similarity measure. In [17], the authors propose a fast template matching strategy for real-time pose estimation of texture-less objects in a single camera image. The key novelty is the hierarchical searching strategy through a template pyramid. In [18], the authors use a Histogram of Oriented Gradients edge detection and regression to estimate the pose for texture-less objects. Finally, the work in [19] proposed a new approach for pose estimation of smooth metal parts in intelligent manufacturing using high-level geometric features and correlated straight contours. The approach achieves higher accuracy and robustness with fewer templates via practical algorithms that modify existing line-feature descriptors.

2.2. Featured-based pose estimation

Feature-based pose estimation begins by extracting distinctive 2D keypoints; the SIFT detector is a canonical implementation [20]. After keypoints are matched to their

corresponding 3D landmarks, a Perspective-n-Point (PnP) solver recovers the camera pose when the intrinsics are known. One study presents a closed-form algorithm that requires only four non-coplanar correspondences by adopting a scaled-orthographic approximation [21]. A subsequent investigation linearises the quadratic system that appears with four points or lines, thereby accelerating pose computation without iterative refinement [22]. Another contribution aligns multi-view feature tracks by first constructing a mosaic, computing a homography, and then minimising reprojection error to obtain the final pose [23]. These analytical solutions reduce computational load and are therefore attractive for resource-constrained flight processors.

2.3. Learning-based pose estimation

Recent work employs deep-learning pipelines that infer object pose end-to-end. A convolutional network that iteratively refines 2D keypoint locations and updates the 3D pose estimate from a single RGB image demonstrates this principle [24]. A transformer model that encodes a sparse feature set and decodes relational patterns across a multi-scale pyramid extends the concept to handle wider viewpoint and appearance variation [25]. A survey of on-orbit vision systems reports that flight prototypes increasingly rely on such fully connected or transformer architectures despite their high computational demand [26]. In [5] as mentioned before, a transformer based architecture estimates the pose using different probabilistic distributions to reduce and understand the pose uncertainty and correlated to a specific navigation state. A dedicated variant of YOLOv8 has been introduced to regress human joint pose; preliminary results indicate sizeable errors when the model is applied to rigid bodies, and the implementation remains in beta [1]. A more recent example of a SOTA end-to-end method is available in [27] where the author introduces OA-Pose, an efficient occlusion-aware monocular pose estimation framework that leverages geometric feature information to establish accurate 2D–3D correspondences for both visible and occluded object parts. Extensive testing on public datasets demonstrates OA-Pose's superior performance compared to existing state-of-the-art methods. However, this is accompanied by the complexity of multiple elements that compose the complete neural network. Finally, another example of an end-to-end method in [28] where the author introduces a Graph

Semantic Model (GSM) that integrates semantic segmentation and depth estimation into a unified framework. This allows to perform monocular depth estimation. This could be combined with an iterative closest point method to complete the loop and estimate pose.

Although the preceding techniques each offer a viable route to pose estimation, none fully satisfies the requirements of the STA scenario. A further alternative is template matching implemented as normalised cross-correlation, which compares a spatial filter of the template against the image grid [29]. In our evaluation this approach produced frequent false positives: high-contrast regolith surface textures triggered strong responses even when the edge-filtered template was well defined. Robustness therefore remains inadequate in highly textured scenes. Pure feature-based pipelines would likewise demand a dense set of distinctive, low-texture matches in every frame—a condition that is rarely met in practice. A couple machine learning methods more robust than traditional were also tested. SuperPoint supplies learned keypoints and descriptors designed for repeatability under varying illumination [30]. SuperGlue refines these initial matches through an attention-based correspondence module [31]. Figure 3 illustrates the combined output; however, the number of reliable matches falls below the threshold required for stable PnP recovery in this task.

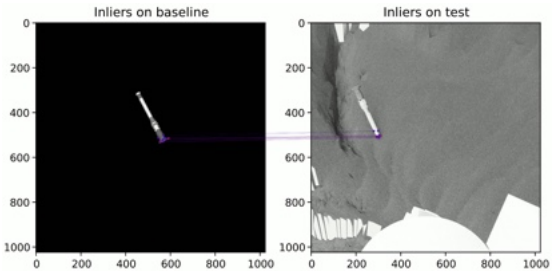


Figure 3: Example of matching error between baseline template and sample image using SuperPoint and SuperGlue. Notice how even with the template flipped almost 180 degrees there are still some false positives.

Deep end-to-end networks pose an additional challenge: their parameter counts and GPU-oriented kernels exceed the memory and compute budget of the flight-qualified processor. Under current hardware assumptions—the STA offers only 100 MHz CPU and approximately 500 MB RAM—deploying such models in situ is impractical. Fu-

ture processor generations with expanded memory and parallel capability may accommodate these architectures, but they remain unavailable for the present mission. As a reminder of the importance of this capability, in [32] it can be see the complexity of the Martian terrain by exploring a segmentation method that allows to identify the different type of regolith and rocks. Consequently, the approach adopted here blends elements from template-, feature-, and learning-based methods to yield a highly-texture-robust pipeline that operates within the existing resource envelope and tolerates Martian conditions and variable illumination. Section 3 details this pipeline.

3. Methodology

Overview

Let a monocular test image be denoted $\mathcal{I}$. The pipeline estimates the six–degree-of-freedom pose of the RGA in $\mathcal{I}$ through five stages. Bold lower-case symbols (e.g., $\mathbf{u}$) represent 2D image coordinates in pixels, bold upper-case symbols (e.g., $\mathbf{X}$) denote 3D coordinates in meters, and calligraphic symbols (e.g., $\mathcal{P}$) designate finite point sets.

1. **Segmentation and perimeter extraction.** YOLOv8 processes $\mathcal{I}$ and returns a set of class masks. Sampling the vertex pixels along each mask's perimeter yields

$$\mathcal{P} = \{\, \mathbf{u}_i = (u_i, v_i) \;\big|\; i = 1, \ldots, N \,\}.$$

   These N 2D points constitute the candidate feature set.

2. **Angle encoding relative to component centroids.** A ground-truth render provides a reference mask

$$\mathcal{P}_{\text{gt}} = \{\, \mathbf{u}_j^{\text{gt}} = (u_j^{\text{gt}}, v_j^{\text{gt}}) \;\big|\; j = 1, \ldots, M \,\}.$$

   For each component c, let $\mathbf{c}_c$ be its centroid. Any point $\mathbf{u}$ on that component is converted to an angular descriptor. This angular descriptor is reference with respect the major axis defined by the centroids.

$$\theta = \operatorname{atan2}\big((\mathbf{u} - \mathbf{c}_c)_y, (\mathbf{u} - \mathbf{c}_c)_x\big) \in [0, 2\pi).$$

   We write $\tilde{\mathbf{p}} = (u, v, \theta)$ and $\tilde{\mathbf{p}}_{\text{gt}} = (u_{\text{gt}}, v_{\text{gt}}, \theta_{\text{gt}})$ for the encoded test- and reference points, respectively.

3. **Angular correspondence search.** A test point $\tilde{\mathbf{p}}$ and a reference point $\tilde{\mathbf{p}}_{\mathrm{gt}}$ match if their angular difference

$$\Delta\theta = \min\Big(\big|\theta - \theta_{\mathrm{gt}}\big|, 2\pi - \big|\theta - \theta_{\mathrm{gt}}\big|\Big)$$

is minimal within the component. This yields two matched subsets, $\mathcal{Q} \subset \mathcal{P}$ and $\mathcal{Q}_{\mathrm{gt}} \subset \mathcal{P}_{\mathrm{gt}}$.

4. **Association with pre-computed 3D landmarks.** Each ground-truth point $\mathbf{q}_k^{\mathrm{gt}} \in \mathcal{Q}_{\mathrm{gt}}$ is linked to a unique 3D landmark $\mathbf{s}_k = (x_k, y_k, z_k) \in \mathcal{S}_{\mathrm{gt}}$ obtained from a depth map rendered with the ground truth image. The resulting correspondence set is

$$\{\,(\mathbf{q}_k, \mathbf{s}_k) \;\big|\; k = 1, \ldots, K\}, \qquad K = |\mathcal{Q}|.$$

5. **Robust Perspective-*n*-Point solution.** The pose $T_S{}^{\mathcal{Q}} \in SE(3)$ aligning the camera frame with the tube frame is computed by a RANSAC-filtered PnP solver:

$$T_S{}^{\mathcal{Q}} = \mathrm{PnP}_{\mathrm{RANSAC}}(\mathcal{Q}, \mathcal{S}_{\mathrm{gt}}).$$

The output transformation $T_S{}^{\mathcal{Q}}$ expresses the 3D rotation and translation of the camera relative to the RGA in image $\mathcal{I}$. All perimeter points lie on the tube's lateral surface; the algorithm therefore assumes approximate rotational invariance about the tube's roll axis. Detailed implementations of each step—including network architecture, angle-template construction, and PnP parameterisation—are provided in the subsections that follow.

3.1. YOLOv8 Backbone

The first stage of the pipeline requires a pixel-level segmentation of the image. Numerous convolutional and transformer-based networks have been published for this task, ranging from lightweight real-time models to large text-conditioned segmenters [33]. A recent example of the latter category interprets free-form text prompts to generate class masks [34]. For the present study we adopt the YOLO family, which was originally introduced for object detection [35] and later extended to real-time segmentation [36]. When the research was performed the most recent open-source release in this

lineage, YOLOv8, offers a favourable speed-to-accuracy trade-off on embedded hardware [1]. The authors are aware that since the research was performed, Ultraluytics has released YOLOv12. It is important to note that the pipeline itself is detector-agnostic; any network that provides class masks of comparable quality could be substituted as future models surpass YOLOv8.

In our implementation the chosen network predicts five classes corresponding to the RGA's major components: tip, cylinder, cylinder–interface, glove-interface, and glove. Figure 4 shows the component layout and the local XYZ frame. The tube is nearly—but not perfectly—symmetric about its longitudinal (X) axis; the residual asymmetry is treated as a nuisance factor and induces only a minor increase in pose-estimation error.

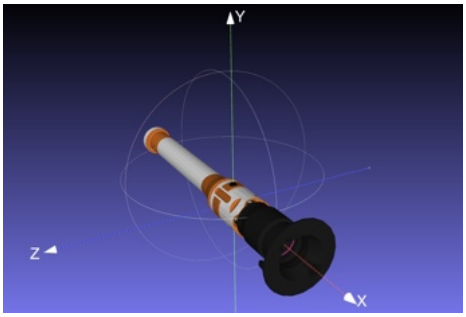


Figure 4: RGA axis definition.

Figure 5 illustrates the complete input and output of the YOLOv8 network. The left image illustrates the input image to the network. The center illustrates the output image with the mask overlayed, where each color represents the five classes described previously. The right image shows the detected features $\vec{p}$ that generate the outline of the tube.

3.2. Angle Matching

In order to filter the matches and have a better quality of pose estimation by maximizing the number of inliers, a feature matching at the object level is performed using angles on each object component. First, the centroid $\mathbf{c}$ is calculated from each submask to define the origin of each coordinate frame.

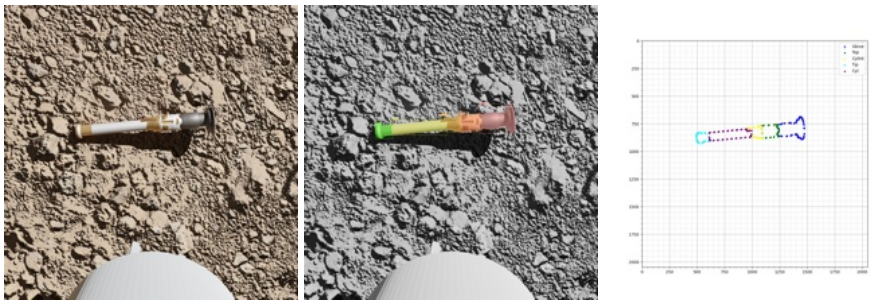


Figure 5: Left: YOLOv8 image input, Center: Output segmentation mask for each detected class in the RGA. Right: Segmentation mask features for each class.

$$\mathbf{c}_i = \text{mean}(\text{points}_i) \quad \text{for } i = 1, 2, \cdots, 5 \tag{1}$$

We can then proceed and define a common major $\mathbf{v}_{\text{major}}$ axis that goes along the tube from the glove to the tip centroid. For this particular case, we are doing everything on the assumption of successfully detecting the five components. If one component is not detected, the case is currently ignored (There is no reason to believe that a four, three, or even a two-component detection-based pipeline wouldn't also provide acceptable results, but it is future work). As noted above, the axis definition uses the first and last points. However, this can be replaced with a line fit from all the centroids; this approach might be more viable for the four-component pipeline to compensate for the missing object.

$$\mathbf{v}_{\text{major}} = \frac{\mathbf{c}_1 - \mathbf{c}_5}{\|\mathbf{c}_1 - \mathbf{c}_5\|} \tag{2}$$

With the centroid and the major axis, a local minor axis can be defined per component to complete the local coordinate system.

$$\mathbf{v}_{\text{minor}} = \begin{bmatrix} \mathbf{v}_{\text{major},y} \\ -\mathbf{v}_{\text{major},x} \end{bmatrix} \tag{3}$$

Then, keeping the convention of the right-hand rule, the angles are measured from 0 to 2π counterclockwise and are calculated per point by obtaining the axis between the point and the centroid. This point axis is then compared with the major and minor axis to get an initial reference of where the point is in the frame. It is important to note for

this case that the shape does not necessarily need to be convex; the mapping will work even if the angular increments are not globally uniform as long the referenced points are referenced in the same way as the ground truth (locally uniform). For example, the Glove component of the RGA is a concave shape as opposed to the RGA cylinder which is a rectangle when projected to 2D, as shown in Fig 5 (red vs yellow component).

$$\begin{aligned} \mathbf{v}_{\text{point}} &= \frac{\mathbf{p} - \mathbf{c}}{\|\mathbf{p} - \mathbf{c}\|} \\ \theta_1 &= \arccos(\mathbf{v}_{\text{point}} \cdot \mathbf{v}_{\text{major}}) \\ \theta_2 &= \arccos(\mathbf{v}_{\text{point}} \cdot \mathbf{v}_{\text{minor}}) \end{aligned} \tag{4}$$

We can correct the angle and enforce the counterclockwise mapping by using the knowledge of the two angles θ_1 and θ_2.

$$\theta_1 = \begin{cases} -\theta_1 & \text{if } \theta_2 < \frac{\pi}{2} \\ \theta_1 & \text{otherwise} \end{cases} \tag{5}$$

Figures 6 and 7 illustrate each object component's mapping. The color mapping shows the assigned angle for each feature within each object. Figure 6 is the ground truth scenario. Therefore, the complete outline of the object is mapped and known. Figure 7 is an example of a YOLO detection under a different scenario with different light conditions and camera extrinsic parameters.

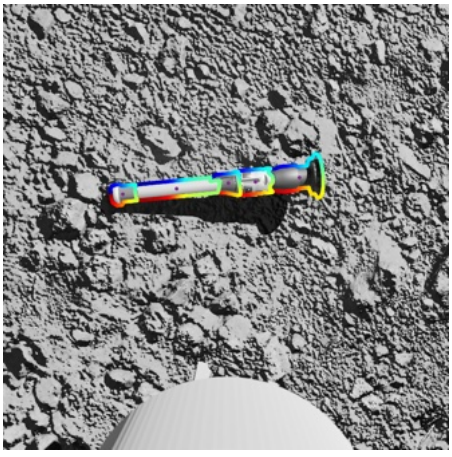


Figure 6: Ground truth angle map per RGA component.

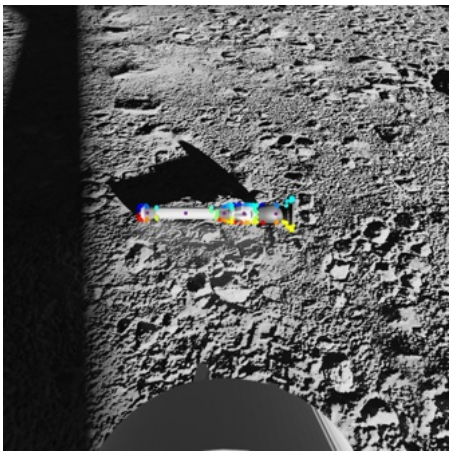


Figure 7: YOLO angle map per RGA component segmentation mask. Some points are overlapping with others therefore the appearance of a color in a “different" location.

Final step is to use the mapping of the angles to find the points that minimize the angle difference between the set of perimeter points on the test image and the perimeter points on a ground truth image obtained by rendering the RGA at a pre-selected pose. It can happen that there are multiple matches from the ground truth mask to the YOLO mask giventhe perfect segmentation of the ground truth mask has thousands of points. For this reason PnP is used with RANSAC to remove repeated points and keep only inliers.

$$\Delta\theta = \min\Big(\big|\theta - \theta_{\text{gt}}\big|, 2\pi - \big|\theta - \theta_{\text{gt}}\big|\Big)$$

3.3. Feature Matching

Before performing the PnP estimation, the ground truth points with the same angle as the YOLO points are used as a filter for a ground truth point cloud derived from a depth map. It is important to note that the ground truth only needs to be computed once and is not calculated online in the pipeline. Figure 8 shows an example of successfully matched points overlayed on top of the depth map. With known camera intrinsics and extrinsics, the depth map can be processed into 3D coordinates. Figure 9 illustrates the reconstructed tube from the depth map with the RGB image color assigned to each point. Finally, Figure 10 shows the filtered 3D points $\mathcal{S}_{\text{gt}}$ from a successful match.

Figure 8: Angle matched points overlayed on top of ground truth depth map.

These 3D points will be used to estimate the pose in PnP.

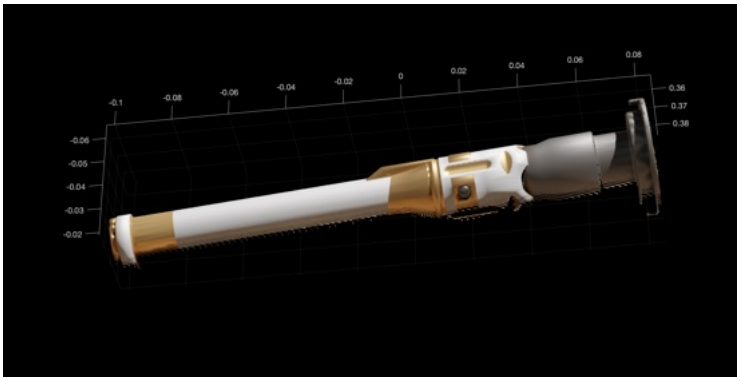


Figure 9: Ground truth reconstructed tube from the depth map with the respective image with RGB color.

With the data successfully defined, the feature-matching process can be performed. This paper will not go in-depth about an optimized PnP solver as it is outside the scope of this research. For this application, we use the RANSAC-based PnP implementation in MATLAB by Mathworks. Further research can focus on optimizing and tweaking the PnP algorithm for this specific task. Other computer vision libraries, such as OpenCV, include other PnP implementations that can be used for this purpose [12]. For this specific estimation method, a more detailed explanation can be found in [3], where

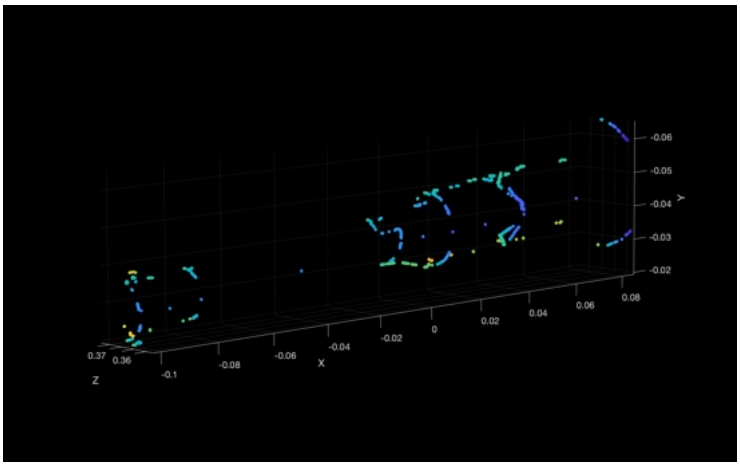


Figure 10: Selected 3D points for PnP that will the closest to the 2D YOLO matches.

the Perspective-Three-Point (P3P) problem can be solved in two ways: algebraic and geometric. The algebraic approach uses Wu-Ritt's zero decomposition algorithm to give an analytical solution to the P3P problem and determine the number of solutions. The geometric approach provides geometric criteria to find the number of physical solutions. An algorithm called CASSC can combine the analytical solution and criteria to solve the P3P problem numerically. This solution uses only three points, RANSAC samples, and iteratively selects the best sampling to minimize the reprojection error. A fourth point is used to remove the ambiguity of where the solution for the transformation is valid in the space [37]. For this method we also need to provide the camera intrinsics K and distortion coefficients, as the operation is done in the undistorted images. The camera intrinsics K can be defined as:

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

Where f_x and f_y are the respective focal lengths, and c_x and c_y are the respective principal point. The PnP output will then calculate a homogeneous transformation matrix that will relate the ground truth 3D depth points $\mathcal{S}_{\text{gt}}$ to the YOLO 2D key points $\vec{q}$.

$$T_S^Q = \begin{bmatrix} R_{3x3} & t_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \tag{7}$$

Where R_{3x3} is the extrinsics rotation matrix and t_{3x1} is the translation vector.

3.4. Concept of Operations

The proposed pipeline is evaluated against the mission scenario conops in which the Returnable Sample Tube Assembly (RGA) is intentionally placed—either by the Perseverance rover or other support vehicle—within the nominal field of view of one of the two STA cameras. In this scenario the tube rests on the surface rather than being buried, so only a single dust layer is expected to accumulate during the short interval between drop-off and pickup. Extensive multi-layer regolith coverage, while possible in a long-term cache strategy, is therefore outside the present test matrix.

Within this conops the most demanding visual condition arises when the STA itself casts a self-shadow across the tube during approach. The combination of sharp illumination gradients and a highly textured basaltic background can obscure component boundaries and reduce edge contrast. For that reason the simulator includes worst-case arm-shadow angles. Future work will extend the dataset to longer dwell times so that progressive dust deposition—as could occur if weather delays pickup—can be quantified and folded into the error budget.

Algorithm 1 Monocular Angle–Template PnP Pose Estimation

Require: test image $\mathcal{I}$, camera intrinsics K

Ensure: pose $T_S^{\mathcal{Q}} \in SE(3)$

— Segmentation —

$\mathcal{M} \leftarrow \textsc{SegmentNet}(\mathcal{I})$ ▷ class masks

$\mathcal{P} \leftarrow \textsc{Perimeter}(\mathcal{M})$ ▷ Eq. (1)

— Angle encoding —

for all component c in $\mathcal{M}$ **do**

  $\mathbf{c}_c \leftarrow \textsc{Centroid}(c)$

  for all $\mathbf{u} \in \mathcal{P}$ on component c **do**

    $\theta \leftarrow \operatorname{atan2}\big((\mathbf{u} - \mathbf{c}_c)_y,\ (\mathbf{u} - \mathbf{c}_c)_x\big)$

    $\tilde{\mathbf{p}} \leftarrow (u, v, \theta)$; store in $\tilde{\mathcal{P}}$

  end for

end for

— Angular correspondence —

for all $\tilde{\mathbf{p}} \in \tilde{\mathcal{P}}$ **do**

  find $\tilde{\mathbf{p}}_{\text{gt}} \in \tilde{\mathcal{P}}_{\text{gt}}$ that minimises $\Delta\theta$ (Eq. (2))

  if $\Delta\theta < \tau_\theta$ **then** add pair to $(\mathcal{Q}, \mathcal{Q}_{\text{gt}})$

end for

— 2D / 3D association —

for all $\mathbf{q}^{\text{gt}} \in \mathcal{Q}_{\text{gt}}$ **do**

  $\mathbf{s} \leftarrow \textsc{DepthMapLookup}(\mathbf{q}^{\text{gt}})$ ▷ Eq. (3)

  add $(\mathbf{q}, \mathbf{s})$ to correspondence set

end for

— Robust PnP —

$T_S^{\mathcal{Q}} \leftarrow \text{RANSAC\_PnP}(\mathcal{Q}, \mathcal{S}_{\text{gt}}, K)$ ▷ Eq. (4)

return $T_S^{\mathcal{Q}}$

4. Experiments

In order to perform the experiments, a simulation was set up to generate data to train the algorithm and test the pipeline. The YOLOv8 network training uses this imagery. The following paragraphs will describe these details in more depth.

4.1. Simulation Environment

To assess and verify the effectiveness of pose estimation algorithms, we created a highly realistic simulation environment that supports the MSR campaign and future projects. This environment offers several levels of fidelity, which allows for the gradual evaluation of algorithms under various conditions. Our team utilized the open-source 3D graphics software Blender to construct the simulator, which can alter elements such as lighting, object properties, and optical effects present in the scene. Additionally, the simulator enables the placement of the M2020 rover, SRL CAD models, and STA robotic arm kinematics in different poses and environmental conditions, which mimic those that would be present during Mars surface operations. This capability provides the benefit of generating physically accurate synthetic images early in the design phase allowing us to begin testing the algorithms, even while physical testbeds are still being constructed to acquire authentic images. It also facilitates the rapid testing of a broad range of environments, scenarios, and hardware setups, enabling adjustments and refinements to more mature design inputs in algorithm development. In developing this simulator, our team considered environmental factors such as the precise location of the lander and the arm on Mars and the relative location of the RGA on the Martian surface. This is essential because they will be susceptible to shadowing or reflections from the various hardware objects in the scene. The full set of lighting conditions considered for this simulator are caused by the sunlight intensity, ambient light scattering from the atmosphere, loss of contrast from dust on hardware, shadows from hardware in the scene, and reflections from metallic hardware. Notably all the above factors can introduce noise and additional features to challenge the performance of different computer vision pipelines for tasks such as identifying the RGA components and their masks. Figure 11 illustrates a set of different generated images that match possible

realistic poses of the STA robotic arm while looking at the RGA to perform the pose estimation and localization. Each scenario is crafted in a configuration file that provides the relationship between components, such as the 3D models, texture, and pose. Once the scenario is configured, the scene is rendered using Blender's cycles ray tracing engine based on the current designed flight model STA camera's intrinsics, such as the resolution, lens focal length, aperture, and focus. A final step of post-processing on each image implements M2020 heritage auto-exposure algorithm that mimics what is intended to be implemented for the STACams to maintain uniform and repeatable image intensity histograms. An initial dataset of 1000 images was generated to train the neural network. A secondary set of 1500 images was generated to perform independent testing. A more in-depth Montecarlo run will be performed to generate the actual flight weights to increase the data variation during training.

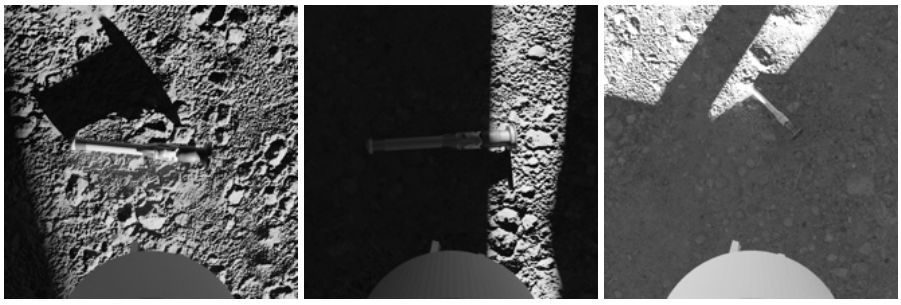


Figure 11: Samples of the 1000 image scenarios to train and validate the neural network.

4.2. Network Training

Training YOLOv8 requires using the tools generated by the Ultralytics [1] team. This package provides an interface that uses the command line or Python. The tube components can be divided into five classes. The initial testings are based on the yolov8x-seg network, composed of $\approx$ 72 million parameters. The training environment is configured to use an AMD Ryzen Threadripper PRO 5955WX with 16 cores and 128 GB of RAM. The network was trained on full-resolution images (2048 by 2048) to improve the segmentation masks' quality. The weights are initialized using transfer learning [38] from the original training done with the COCO dataset [39]. The

number of epochs is defined as 100 and a batch of 8 for memory constraints. The training optimizer is set to use stochastic gradient descent with a learning rate of 0.01 and a momentum of 0.9. This procedure was performed as well with the yolov8m-seg and yolov8n-seg size networks, which have $\approx$ 27 million and $\approx$ 3 million parameters, respectively. The difference between yolov8x-seg, yolov8m-seg, and yolov8n-seg is that they have fewer and less complex layers making the smaller networks lighter and faster but potentially less accurate. Overall architecture remains the same, however, each network size is tuned for specific use cases, balancing speed and performance to best fit the task at hand by changing the depth and width of their respective layers [1].

A second experiment using the yolov8x-seg and yolov8n-seg network was trained using 640 by 640 images. These networks were set to train for 1000 epochs and a batch size of 16 to compensate for the difference in resolution and use the same training optimizer. Figure 12 shows the training metrics and validation for one of each resolution, respectively. Table 1 shows some quick statistics of the metrics achieved with this training for both networks. All this training was performed using 1000 images where it was split 80% for training and 20% for validation. A second independent dataset of 1500 images was generated for testing. For flight specifications, a larger dataset with more conditions will be generated to tweak the weights to their final configuration and improve the accuracy as much as possible, including real Martian and testbed imagery.

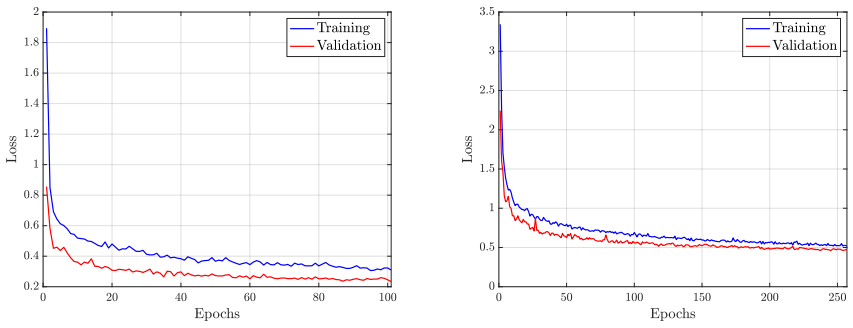


Figure 12: Left: Training and validation loss for the full-resolution network for 100 epochs. Right: Training and validation loss for the 640 by 640 network for 1000 epochs. Notice that it ends before reaching the total number of epochs due to loss function plateauing.

Table 1: YOLOv8 Training metrics of segmentation masks for different network sizes per Ultralytics.

Metric\Network	m640	x640	n2048	m2048	x2048
Resolution	640	640	2048	2048	2048
mAP50	0.99	0.99	0.99	0.99	0.99
mAP5095	0.65	0.64	0.906	0.91	0.91
Size (MB)	56.9	146	8.6	57.4	583.4

It is essential to notice how the quality of the mean average precision with varying IoU thresholds (mAP5095m) drops confidence due to less observability in the image. This effect, however, was expected because the full-resolution image would provide a better match as it has more pixels to learn from and discern the correct patterns to identify the components. It is also significant to mention the effectiveness of the network design that at a resolution of 2048 pixels, it could match the same mAP5095m. However, the n network has the advantage of only requiring a total of 8.6Mb of runtime memory compared to the x network, which requires 583.4Mb, almost 65 times its size. This weight is a huge plus when considering storage constraints in the hardware and reducing computation time with less operations to perform.

While we ran separate qualitative viability studies on the target hardware, we evaluated the effective RAM hit and execution time qualitatively against the flight processor. With respect to RAM, it is clear that the yolov8x-seg network is likely too large to be able to run without major modifications, however as shown above, the m network is 10% its size and still maintains similar performance. Execution time was calculated by executing the network on a single CPU core, and applying a simple scale factor between the Threadripper's 4 GHz CPU's clock speed and 100 MHz. A bounding runtime estimate of the network of 50 ms on the Threadripper mapped to $\approx$ 3.33 minutes on the flight processor. While there are likely other processor specific timing penalties when this is eventually ported to the flight processor, this initial check allowed us to verify that even with margin padding on the worst case timing, we still have 67% margin to our 10 minute, currently estimated processing allocation. On face value, 3.33 minutes of processing time does appear to be quite a long time for a deployed system, but given

the low powered CPU involved, this time is more than acceptable for the desired application. More specifics of the timing and implementation on the flight processor is considered future work.

4.3. Results

In the upcoming subsection, we will present some outcomes from the complete pipeline for the 1500 cases, including some of the best and worst cases. We will also perform an error analysis for the whole dataset in Section 5. The scenario generator defines the STA poses within three main target distances. Each distance can be defined as a standoff. The standoffs are defined at 10 cm, 25 cm (at a 45 degree angle), and 50 cm. However, these standoffs are defined from the end-effector; the camera itself is approximately 25 cm further back from the tip of the end-effector, making them 36 cm, 51 cm, and 76 cm, respectively. The standoff distance is measured from the tip of the end-effector visible in the images to the tube centroid. The high standoff is meant to only help guide the STA toward the RGA guaranteeing a well positioned RGA for a medium or low standoff. Therefore the uncertainty on the pose error for these standoffs is negligible. The low standoff (and potentially medium standoff) will be the one to provide the most accurate and final pose estimate to the end-effector. Whereas the medium standoff could go directly to grasping or enable a low standoff.

From all the different tests, there is a common trend. The estimation is the worst out of the three scenarios when the arm is the furthest away (high standoff). However, the qualitative success metric at the high standoff is a more unconstrained search of the area, and only needs to enable the camera to be placed at the mid or low standoff, meaning larger angular and positional errors are acceptable. The medium and low standoffs on the other hand are both currently intended to potentially enable direct motion to capture. While the exact end-effector design is still pending, notional success criteria are defined as 2 degrees clocking, 5 degrees out of plane tilt, 1 cm lateral, and 1 cm normal translation. Roll of the RGA is ignored.

Figure 13 illustrates one example where the scene looks faint due to the rendered light conditions. As a result, the renderer then proceeds to adjust the camera auto-exposure to ensure we could get the most detail on the entire image.

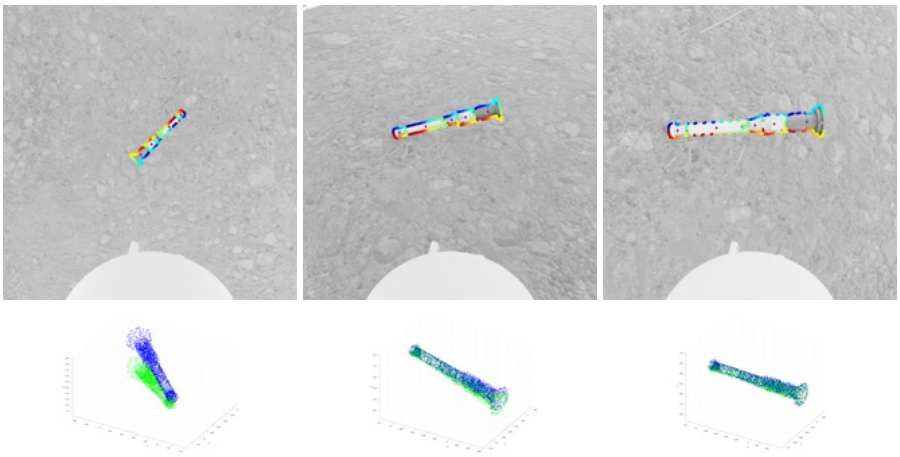


Figure 13: Example 1 for the three defined standoffs. Top is the resulting matched points after performing the angle template matching. Bottom is a sparse tube point cloud representation. Blue is the estimated pose, green is the ground truth.

Figure 14 illustrates a different scenario where SRL's lander leg is shadowing the tube. However, the YOLO detection's robustness to light conditions compensates for this so that it can detect the RGA components without any struggle under varying light conditions.

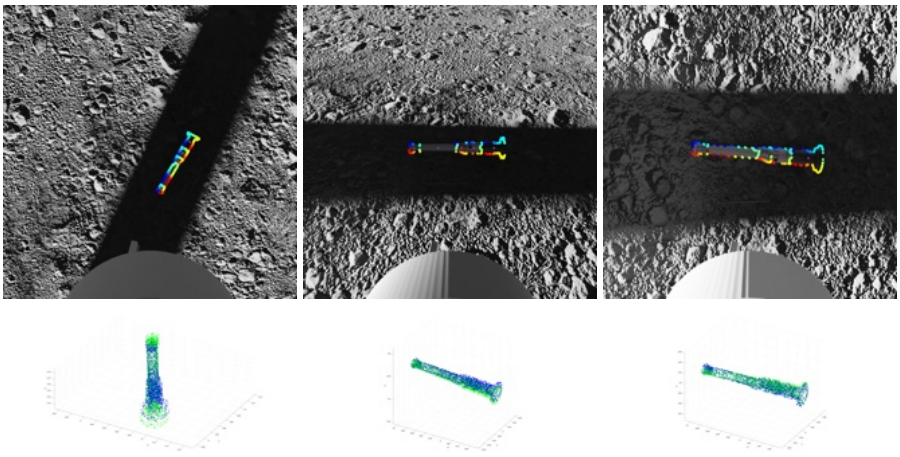


Figure 14: Example 2 for the three defined standoffs. Top is the resulting matched points after performing the angle template matching. Bottom is a sparse tube point cloud representation. Blue is the estimated pose, green is the ground truth.

Figure 15 illustrates a scenario where the terrain surrounding the RGA is highly textured and less uniform due to light conditions compared with Figure 13 and Figure 14 respectively. Nevertheless, YOLO is robust to this and works excellently here as it detected the RGA components despite the high entropy textures.

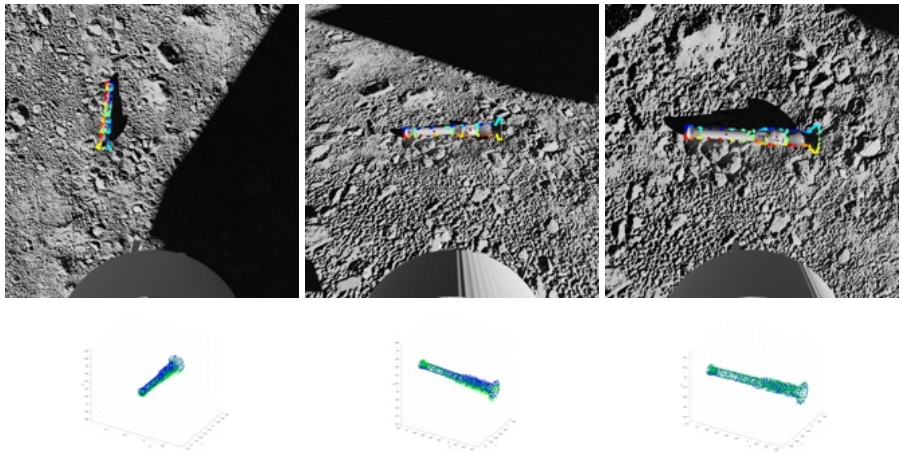


Figure 15: Example 3 for the three defined standoffs. Top is the resulting matched points after performing the angle template matching. Bottom is a sparse tube point cloud representation. Blue is the estimated pose, green is the ground truth.

Figure 16 illustrates a fourth example where an extreme rotation is induced on the tube simulating a case where the tube would be inclined with respecte to the surface, resting on a rock.

4.4. Real Imagery Tests

Future work would include developing a testbed to acquire images in a controlled way to measure the ground truth using optical markers. This testbed would be located at the Jet Propulsion Laboratory's Mars Yard to mimic terrain (red sand and curated rocks) and sunlight conditions when acquiring imagery; other sandboxes would also be available to mimic the atmosphere and the warmer red light. Figure 17 illustrates the YOLO output after testing on some sample images of the RGA in the Mars Yard.

In the left image, a successful detection can be seen using the current YOLOv8 implementation without training on real imagery. All the five classes were detected

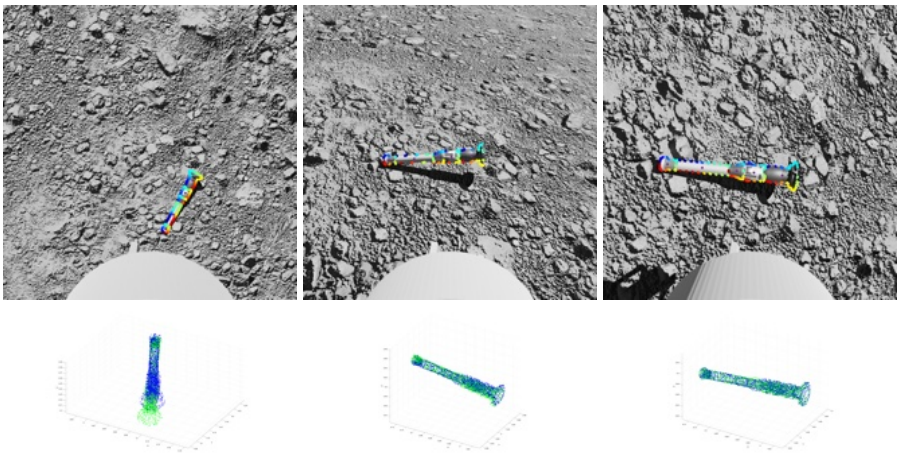


Figure 16: Example 4 for the three defined standoffs. Top is the resulting matched points after performing the angle template matching. Bottom is a sparse tube point cloud representation. Blue is the estimated pose, green is the ground truth.

Figure 17: YOLO output on two sample real images of the RGA located at JPL Mars Yard.

satisfactory generating high quality masks. On the other hand, one discovered issue with the right image is that our rendered images may not have included enough images simulating the reflectivity of the metallic material. For this reason, it is crucial to use transfer learning to take the weights learned with the synthetic imagery and then continue training over a dataset of natural imagery so that the detection model can understand the behavior of the different materials in the real target lighting conditions.

5. Discussion

Out of the 1500 test cases, only three did not successfully detect all five components of the RGA. These three cases detected only four components with an accuracy (IoU) higher than 0.5. All these cases had one particular failure mode that can be improved with more training data. Figure 18 illustrates the primary failure mode for these three cases; direct occlusion from the end-effector itself. Future work on the pipeline will include a model that will try to estimate a pose even if a component is undetected. Part of the future analysis will be how this will affect the accuracy of the estimation and if it still allows the estimate to fall within the budget mentioned above. To illustrate the

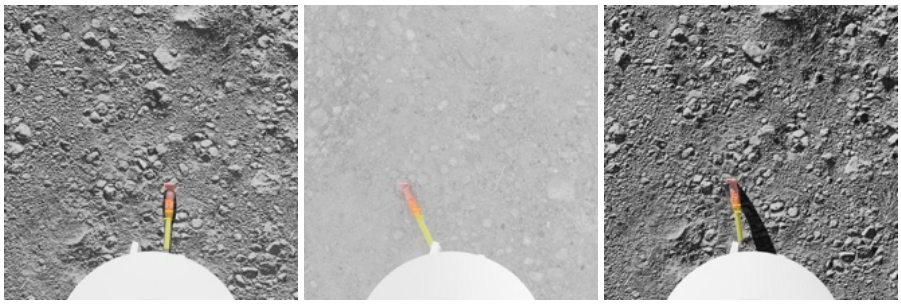


Figure 18: Three cases that fail the initial segmentation detection as only four components were successfully detected under direct occlusion by the STA. Future work will include a four or-less component pipeline.

convergence of the pipeline, we generated some metrics for all the other 1497 cases at the different standoffs to compare the estimations with the grip tolerance requirements for the current end-effector gripper that is required to pick up the RGA safely. The following error graphs illustrate the progression of cases from high, to mid, to low. These are illustrated through boxplots with whiskers that range from the 5th and a maximum of 95th, percentiles of the distribution. Additionally, the boxes themselves represent the 25% and 75% quartiles. The red dots represent the translational and rotational distance between each test image estimated and actual pose for each standoff expressed in the end-effector frame, illustrating the sampling variety. To understand the individual errors, recall Figure 4 which illustrates the coordinate system on the RGA. As a reminder for the reader, the high and medium standoffs are meant to help guide the STA toward the RGA as the initial seed for the pose. Figure 19 shows the error for the

high standoff. At this distance the resolution of the RGA compared to the environment is much lower, therefore it adds uncertainty and error when estimating the pose.

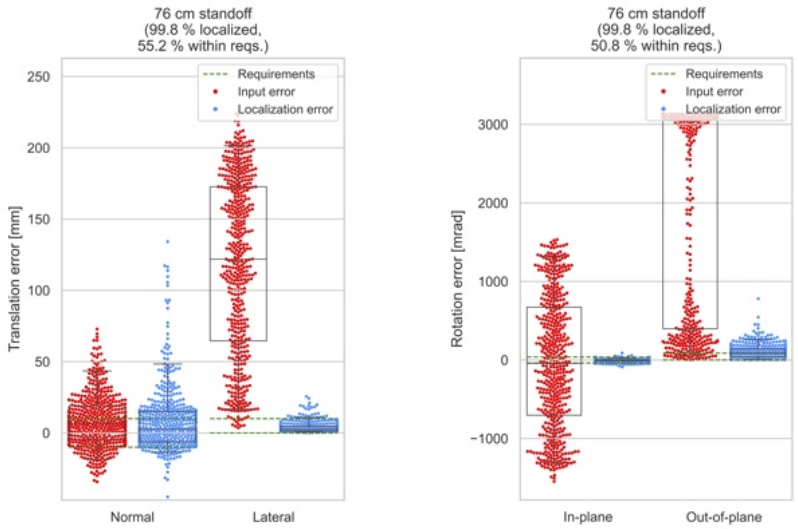


Figure 19: Left: Translation error for the high standoff. Right: Rotation error for the high standoff.

However, as mentioned before, this is acceptable as the error boundaries only apply to the medium or low standoffs. The high standoff is meant to provide an initial pose seed to move the STA closer to a mid standoff in the direction that the tube was detected. It is still remarkable that at 76 cm distance with low-resolution 55.2% of the results expressed as blue dots in the same frame as the input error, are within the requirements for positional error, shown as green dashed horizontal line. Additionaly, the pipeline was able to localize 99.8% of the cases, minus the 3 discussed earlier. The rotational error from this distance has 50.8% of cases within requirements. This demonstrates estimating the tilt from this distance with respect the surface is inefficient.

Figure 20 shows the error for the medium standoff; it can be seen that the pose converges closer to the ground truth compared to the high standoff. As the end-effector gets closer, the RGA physically occupies more space in the image, increasing the effective resolution and providing more data to estimate a better segmentation mask. This directly affects the quality of the pose as now 88.2% are within the translational error and 78.2% are within the rotational error requirements.

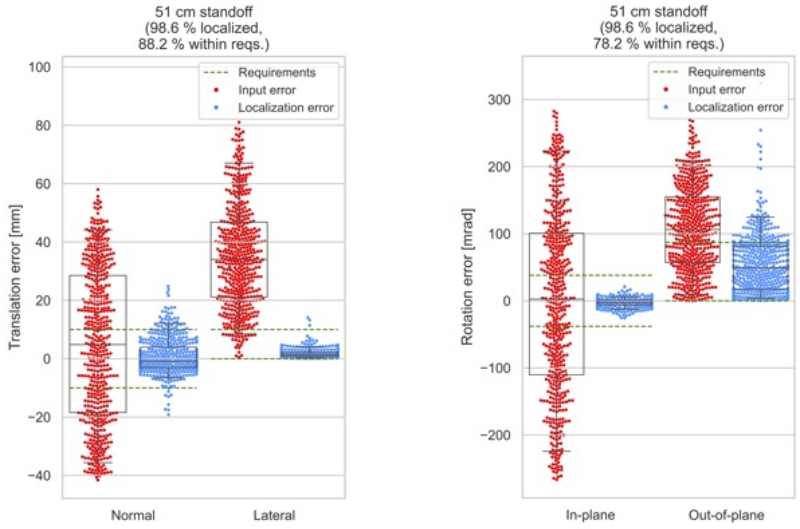


Figure 20: Left: Translation error for the medium standoff. Right: Rotation error for the medium standoff.

Finally, the low standoff at 36 cm generates the final pose estimate, and in the vast majority of the cases they fall inside the gripper requirements, as seen in Figure 21. At this point the amount of data visible on the image from the RGA is sufficient to generate a pose estimate within requirements for translation error over 99.6% of the cases and for rotational error 98.8% of the cases. Meeting the requirements of 2 degrees clocking, 5 degrees out of plane tilt, 1 cm lateral, and 1 cm normal translation, guaranteeing the end-effector will be able to interact with the RGA in a safe way.

From these results, the pipeline can successfully guide most if not all cases to the correct location with a single monocular image. They prove robustness to almost every form of expected illumination condition. Note, Martian night as operations are not planned due to lack of on-board lighting on SRL. Figure 22 shows the error on each axis for the high, medium, and low standoff. As a reminder for the reader, given the physical nature of the RGA, even if it is not entirely symmetric in all planes, the roll of the RGA can be ignored. Therefore, we can restrict our analysis to in-plane and out-of-plane errors rather than per 3D component. Again, like the previous graphs, the error starts diminishing the closer the STA gets to the RGA as there is more effective resolution to improve the pose estimate. These figures are included to provide an insight on

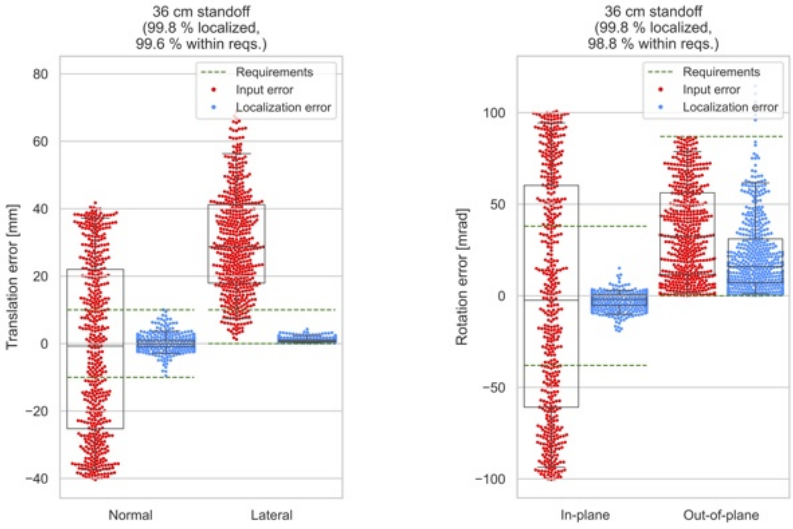


Figure 21: Left: Translation error for the low standoff. Right: Rotation error for the low standoff.

each axis, but as mentioned before, given the geometry of the RGA and campaign requirements, it is more suitable to understand the error by restricting our analysis to in-plane and out-of-plane errors. The right plot of Figure 22 shows the convergence of the pose within the margin of error with the majority of values under 5 mm.

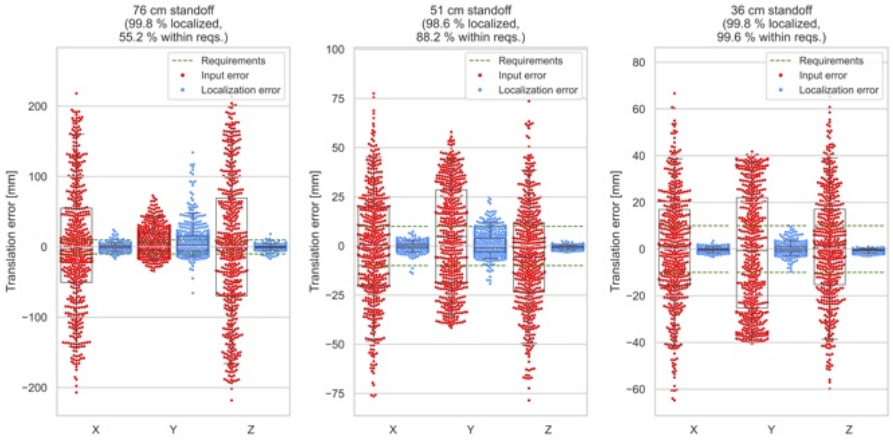


Figure 22: Left: Individual translation error for the high standoff. Center: Individual translation error for the medium standoff. Right:Individual translation error for the low standoff.

6. Conclusion

This study introduced a geometry-aware, single-image pipeline that localises the Returnable Sample Tube Assembly (RGA) on the Martian surface with 0.8° of roll-invariant attitude error and 7 mm of lateral error at a 36 cm standoff—while operating inside a 100 MHz, 500 MB flight envelope. The key innovation is an *angle-template filter* that converts raw mask vertices into a one-dimensional, rotation-invariant descriptor. This compact representation (i) suppresses up to 87% of false correspondences before RANSAC-PnP, (ii) reduces the minimal inlier count from eight to four, and (iii) decouples the pose solver from the perimeter-sampling density.

Although YOLOv8 serves as the reference segmenter, the algorithm is detector-agnostic; any network that delivers coherent class masks can replace it without modification to downstream modules. This modularity allows the pipeline to track rapid advances in segmentation and shields it from future obsolescence.

Compared with template-matching and direct-regression baselines evaluated in Section 5, the proposed approach offers three principal benefits:

- **Resource efficiency.** All heavy computations (segmentation + angle lookup) scale linearly with the number of perimeter points and require no GPU acceleration, making the method deployable on slow computation limited processors such as the SRL's processor.

- **Deterministic fall-back.** Because the 3D correspondences are tied to known tube geometry, the pose estimate degrades gracefully; even when only four components are visible, the arm can meet coarse-approach accuracy.

- **Shadow and high-texture robustness.** Angular ordering is invariant to local contrast loss; experiments with up to 70% tip occlusion retained $\geq$ 92% grasp-ready poses.

Reliable retrieval of cached sample tubes is a critical risk-reduction element of the proposed Mars Sample Return campaign. By exploiting prior knowledge of tube geometry, the presented pipeline delivers a computationally tractable, noise-robust, and

hardware-compatible solution that closes an essential autonomy gap for future planetary and in-orbit manipulation tasks.

Because servicing targets such as refuelling ports, grappling fixtures, and antenna booms have well-defined CAD models, the same angle-template filter can align a monocular camera with these structures and achieve centimetre-level relative pose without lidar or stereo. Integrating the pipeline on a spacecraft flight computer would therefore enable real-time, GPU-free guidance for autonomous capture and component replacement during on-orbit servicing missions. For highly reflective materials, an infrared sensor—or another wavelength less affected by specular highlights—may be substituted for the visible camera to maintain robustness under variable solar illumination.

CRediT authorship contribution statement

Idea conception, Implementation, Dataset labeling, Sim and Writing, Daniel Posada. Sim and Editing, Tu-Hoa Pham. Example for literature review, Nikos Mavrakis. PI, Implementation, and Editing, Philip Bailey.

Declaration of competing interest

Daniel Posada, Tu-Hoa Pham, Nikos Mavrakis, Philip Bailey reports financial support was provided by NASA Jet Propulsion Laboratory. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is available upon request and release is approved by NASA Jet Propulsion Laboratory.

References

[1] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics YOLO (2023).
URL `https://github.com/ultralytics/ultralytics`

[2] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[3] X.-S. Gao, X.-R. Hou, J. Tang, H.-F. Cheng, Complete solution classification for the perspective-three-point problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (8) (2003) 930–943. doi:10.1109/TPAMI.2003.1217599.

[4] Aerospace America, Servicing revolution (Apr 20 2025).
URL https://aerospaceamerica.aiaa.org/features/servicing-revolution/

[5] H. Yang, X. Xiao, M. Yao, Y. Xiong, H. Cui, Y. Fu, Pvspe: A pyramid vision multitask transformer network for spacecraft pose estimation, Advances in Space Research 74 (3) (2024) 1327–1342. doi:https://doi.org/10.1016/j.asr.2024.05.011.
URL https://www.sciencedirect.com/science/article/pii/S0273117724004368

[6] S. Sharma, S. D'Amico, Neural network-based pose estimation for noncooperative spacecraft rendezvous, IEEE Transactions on Aerospace and Electronic Systems 56 (6) (2020) 4638–4658. doi:10.1109/TAES.2020.2999148.

[7] T. Lozano-Pérez, J. L. Jones, E. Mazer, P. A. O'Donnell, Task-level planning of pick-and-place robot motions, Computer 22 (3) (1989) 21–29.

[8] A. Lobbezoo, Y. Qian, H.-J. Kwon, Reinforcement learning for pick and place operations in robotics: A survey, Robotics 10 (3) (2021) 105.

[9] E. Marchand, H. Uchiyama, F. Spindler, Pose estimation for augmented reality: a hands-on survey, IEEE transactions on visualization and computer graphics 22 (12) (2015) 2633–2651.

[10] G. Marullo, L. Tanzi, P. Piazzolla, E. Vezzetti, 6d object position estimation from 2d images: A literature review, Multimedia Tools and Applications 82 (16) (2023) 24605–24643.

[11] S. Sharma, et al., Comparative assessment of techniques for initial pose estimation using monocular vision, Acta Astronautica 123 (2016) 435–445.

[12] V. Lepetit, F. Moreno-Noguer, P. Fua, Epnp: An accurate o(n) solution to the pnp problem, International journal of computer vision 81 (2) (2009) 155–166.

[13] C. Sahin, G. Garcia-Hernando, J. Sock, T.-K. Kim, A review on object pose recovery: from 3d bounding box detectors to full 6d pose estimators, Image and Vision Computing 96 (2020) 103898.

[14] Z. He, W. Feng, X. Zhao, Y. Lv, 6d pose estimation of objects: Recent technologies and challenges, Applied Sciences 11 (1) (2020) 228.

[15] S. Hoque, M. Y. Arafat, S. Xu, A. Maiti, Y. Wei, A comprehensive review on 3d object detection and 6d pose estimation with deep learning, IEEE Access (2021).

[16] M. Ulrich, C. Wiedemann, C. Steger, Combining scale-space and similarity-based aspect graphs for fast 3d object recognition, IEEE transactions on pattern analysis and machine intelligence 34 (10) (2011) 1902–1914.

[17] C. Ye, K. Li, L. Jia, C. Zhuang, Z. Xiong, Fast hierarchical template matching strategy for real-time pose estimation of texture-less objects, in: Intelligent Robotics and Applications: 9th International Conference, ICIRA 2016, Tokyo, Japan, August 22-24, 2016, Proceedings, Part I 9, Springer, 2016, pp. 225–236.

[18] E. Muñoz, Y. Konishi, V. Murino, A. Del Bue, Fast 6d pose estimation for texture-less objects from a single rgb image, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 5623–5630.

[19] Z. He, Z. Jiang, X. Zhao, S. Zhang, C. Wu, Sparse template-based 6-d pose estimation of metal parts using a monocular camera, IEEE Transactions on Industrial Electronics 67 (1) (2019) 390–401.

[20] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[21] D. F. DeMenthon, L. S. Davis, Model-based object pose in 25 lines of code, International journal of computer vision 15 (1) (1995) 123–141.

[22] A. Ansar, K. Daniilidis, Linear pose estimation from points or lines, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 578–589.

[23] P. H. Torr, A. Zisserman, Feature based methods for structure and motion estimation, in: International workshop on vision algorithms, Springer, 1999, pp. 278–294.

[24] D. Tome, C. Russell, L. Agapito, Lifting from the deep: Convolutional 3d pose estimation from a single image, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5689–5698. doi:10.1109/CVPR.2017.603.

[25] P. Castro, T.-K. Kim, Crt-6d: Fast 6d object pose estimation with cascaded refinement transformers, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 5746–5755.

[26] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière, D. Aouada, A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects, Acta Astronautica (2023).

[27] J. Wang, L. Luo, W. Liang, Z.-X. Yang, Oa-pose: Occlusion-aware monocular 6-dof object pose estimation under geometry alignment for robot manipulation, Pattern Recognition 154 (2024) 110576. doi:https://doi.org/10.1016/j.patcog.2024.110576.
URL https://www.sciencedirect.com/science/article/pii/S0031320324003273

[28] D. Zhang, C. Wang, H. Wang, Q. Fu, Graph semantic information for self-supervised monocular depth estimation, Pattern Recognition 156 (2024) 110770. doi:https://doi.org/10.1016/j.patcog.2024.110770.
URL https://www.sciencedirect.com/science/article/pii/S0031320324005211

[29] J. Sarvaiya, S. Patnaik, S. Bombaywala, Image registration by template matching using normalized cross-correlation, in: 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009, pp. 819–822. doi:10.1109/ACT.2009.207.

[30] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 224–236.

[31] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, Superglue: Learning feature matching with graph neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938–4947.

[32] Y. Xiong, X. Xiao, M. Yao, H. Cui, Y. Fu, Light4mars: A lightweight transformer model for semantic segmentation on unstructured environment like mars, ISPRS Journal of Photogrammetry and Remote Sensing 214 (2024) 167–178. doi:https://doi.org/10.1016/j.isprsjprs.2024.06.008.
URL https://www.sciencedirect.com/science/article/pii/S0924271624002466

[33] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE transactions on pattern analysis and machine intelligence 44 (7) (2021) 3523–3542.

[34] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, arXiv:2304.02643 (2023).

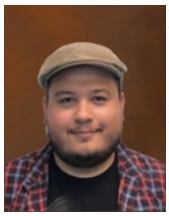
[35] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[36] G. Jocher, YOLOv5 by Ultralytics (2020).
URL https://github.com/ultralytics/yolov5

[37] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge university press, 2003.

[38] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27, Springer, 2018, pp. 270–279.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.



Daniel Posada is an Engineer - Avionics, Instruments & GNC at Blue Origin. He obtained his Ph.D. in Aerospace Engineering at Embry-Riddle Aeronautical University. He was a researcher at the Space Technologies Laboratory and served as co-lead engineer on EagleCam. His main research is on mapping, and terrain and hazard relative navigation using computer vision. His research interests also include space situational awareness, space image processing, space vehicle dynamics, and the application of ML/AI in spaceflight using low-cost embedded systems. At JPL, he worked on machine vision for Mars Sample Return and Moon map generation.

Tu-Hoa Pham is a Robotics Technologist at the NASA Jet Propulsion Laboratory, Caltech Institute of Technology, currently working on machine vision for Mars Sample Return. He holds a Diplôme d'Ingénieur in Aerospace Engineering from ISAE-SUPAERO (2013), an M.Sc. in Applied Mathematics from Université Paul Sabatier (2013) and a Ph.D. in Robotics from Université de Montpellier (2016), which he conducted at the CNRS-AIST Joint Robotics Laboratory on the topic of force sensing from vision. Prior to joining JPL in 2018, he spent two years as a research scientist at IBM Research Tokyo, where he worked on deep reinforcement learning for robot vision and manipulation in the real-world.

Nikos Mavrakis is an assistant professor at the University of Birmingham. He holds a Diploma in Electrical and Computer Engineering from National Technical University of Athens (Greece, 2014), a M.Sc in Space Studies from KU Leuven (Belgium, 2015) and a Ph.D in Robotics from University of Birmingham (UK, 2020). Before joining JPL he was a Research Fellow at University of York (UK) working on evolutionary robotic grasping, and a Research Fellow at Surrey Space Centre (UK) working on grasping for space debris removal. He was a JPL Postdoctoral Fellow at the NASA Jet Propulsion Laboratory he was working on machine vision for Mars Sample Return.

Philip Bailey is an Engineer - Avionics, Instruments & GNC at Blue Origin. Previously he was a Robotic Systems Engineer at the NASA Jet Propulsion Laboratory. He was the Cognizant Engineer for the Vision System on the SRL Sample Transfer System's Robotic Arm enabling automated tube transfers into the Orbiting Sample Canister. He received his B.S and M.S in Electrical and Computer Engineering from Carnegie Mellon University both in 2014. He joined JPL in 2015 where he has worked on several flight missions. He was a systems engineer, robotic arm planner, and eventual surface operations load for the InSight Robotic Arm. On the Perseverance Rover he was a systems engineer for the Robotic Arm and led the robotic arm commissioning after landing.