VISION-BASED LOCALIZATION FOR THE MSR SAMPLE TRANSFER ARM

Marcos Avilés[1], David Savary[2], Augusto Gómez[3], Marco Mammarella[4], Andrea Rusconi[5], Francesco Villa[6], Guido Sangiovanni[7], Davide Nicolis[8]

[1] GMV Aerospace and Defence SAU, Spain, Email: maaviles@gmv.com
[2] GMV Aerospace and Defence SAU, Spain, Email: dsavary@gmv.com
[3] GMV Aerospace and Defence SAU, Spain, Email: augusto.gomez.e@gmv.com
[4] GMV Aerospace and Defence SAU, Spain, Email: mmammarella@gmv.com
[5] Leonardo SpA, Italy, Email: andrea.rusconi@leonardo.com
[6] Leonardo SpA, Italy, Email: francesco.villa01@leonardo.com
[7] Leonardo SpA, Italy, Email: guido.sangiovanni@leonardo.com
[8]ESA/ESTEC, The Netherlands, Email: Davide.Nicolis@ext.esa.int

ABSTRACT

This paper presents the design, development and initial test results of the vision algorithms involved in the autonomous operations of the Sample Transfer Arm, responsible of transferring the sample tubes from the Perseverance and Mars terrain to the Mars Ascent System, as part of the Mars Sample Return campaign.

These algorithms, integrated in a dedicated EGSE replicating the lander processor and operating the arm, will be used for its validation in Europe before delivery to NASA/JPL for its integration in the lander.

1. INTRODUCTION

The NASA/JPL led Mars Sample Return Campaign (MSR) is a response to the long-running scientific objective to better understand Mars. By acquiring and returning to Earth a rigorously documented set of Mars samples, scientists will have access to the full breadth and depth of analytical science instruments available in terrestrial laboratories.

The Mars Sample Return - Sample Transfer Arm (STA) is one of the ESA contributions to the MSR Campaign. The STA will transfer the sample tubes from the Perseverance rover and from Mars terrain. Once the rover has returned to its parking location near the lander, the STA on-board the lander will then transfer the sample tubes from the parked rover into the Orbiting Sample container (OS) located inside the Mars Ascent System (MAS).

The STA will also be used to close and secure the OS lid after the completion of the tube transfer operations. Due to mission constraints, all these functions need to be performed within a limited period of time, which requires a high level of autonomy. The vision algorithms that will be running on the lander on-board processor are key elements to this autonomy, since they will allow an accurate localization of the target elements to be manipulated by the STA.

The overall mission involving the STA is summarized in Fig. 1. Grey tasks are those that involve Vision-related functions.

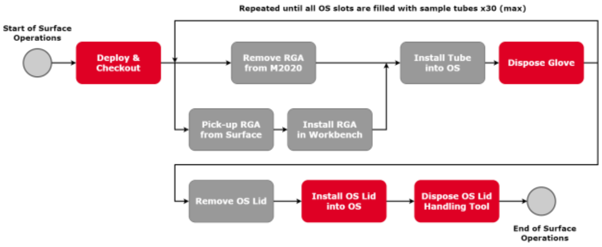

Figure 1. STA Mission operations

This paper focuses on the design, development and initial test results of the various vision algorithms involved in the process. More specifically, vision-based solutions for the following scenarios are presented and described:

- Localization of the Perseverance Bit Carousel, from where the STA will collect the samples collected by the rover.
- Localization of sample tubes on the terrain left by the Sample Recovery Helicopters in the proximity of the lander.
- Localization of the OS, where the sample tubes will be inserted by the STA.
- Localization of the OS lid, to be placed onto the OS.
- Localization of the Workbench, where the tubes are temporarily placed to switch the STA end effector grip type.

2. VISION-BASED LOCALIZATION

Due to the uncertainties in the positions of the different elements to be operated by the STA (the OS might have moved slightly during the landing, the position of the Perseverance is computed from cameras in the lander, etc), vision-based localization is performed incrementally. This allows a safe approach of the STA to the target minimizing the risk of collision while also maximizing the accuracy at the point of operation. More specifically, localization is performed at three different points (see Fig. 2):

- Far Viewpoint. Based on a (ground-based) teach point location. Used to compute a more accurate estimation that allows a safer approach to the target.
- Intermediate Viewpoint. Based on the previous estimate, the arm can safely get closer to the target within the limits of the accuracy of the previously computed pose.
- Close Viewpoint. Based on the already accurate estimate obtained at the medium point, the arm moves as close as possible to the target to compute the final and most accurate estimate which will guide the last movement before switching to active compliance.

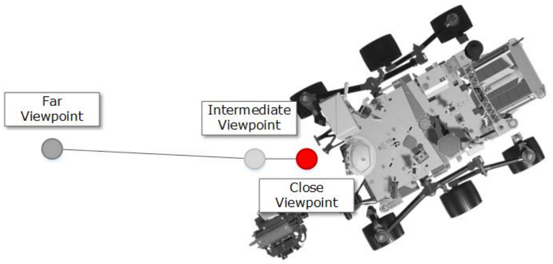

Figure 2. Progressive localization approach to the target.

We now describe the algorithms for the different scenarios, starting from the general and initial stage of preprocessing, which is executed before the actual localization algorithms are called.

2.1. Preprocessing

The preprocessing stage is responsible of doing the necessary steps to provide the different vision algorithms with suitable images. This includes the evaluation of the exposure time, the acquisition of multiple images with different integration times, and finally, the computation of a High Dynamic Range (HDR) image which is then tonemapped to fit it into the lower dynamic range of the vision algorithms while also retaining local contrast. After that, the lens distortion is corrected from the tonemapped image.

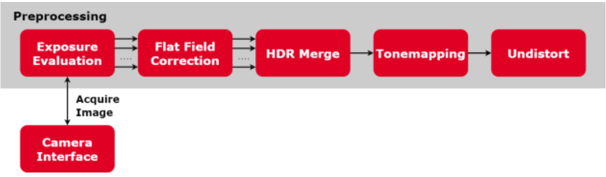

Figure 3. Preprocessing steps

The sequence of steps is summarized in Fig. 3 and described below:

- Exposure Evaluation: Before any of the different localization algorithms is executed, the images acquired by the camera are first evaluated for a proper exposure that enables an optimal processing of the vision algorithms. An automatic auto exposure step is first performed, estimating an exposure time $t_{highlight}$, to ensure that no clipping exists in the highlights (or, more specifically, that only small fraction of pixels is clipped). The same image is then evaluated to check if no clipping is happening in the shadows (or that only a small fraction of pixel is clipped). If significant shadow clipping is happening, then a new exposure time t_{shadow} is computed to prevent this issue. Then a sequence of intermediate exposures going from t_{shadow} to $t_{highlight}$ with exposure times progressively incremented in steps of x4 (which corresponds to +2EV) are taken.
- Flat Field Correction: Flat Field Correction (FCC) is the process where irregularities in pixel values are corrected by multiplying each pixel value by a factor that uniformizes the brightness across the image. In our case, FFC is used to correct the vignetting caused by the camera lens.
- HDR Merge: By using different exposure parameters on the same scene, a wider dynamic range can be represented and then merged into an image with better dynamic range. The stability of the arm and the static environment also simplifies the process as no alignment step between the images acquired at different exposure times is required. HDR merging is performed by taking the well-exposed pixels (neither saturated nor clipped) of a set of images with different exposures and combining them to obtain a single HDR output image.
- Tonemapping: Once the HDR image has been produced by combining multiple exposures, a tone mapping step is executed. Tone mapping reduces the dynamic range of the entire image while retaining local contrast. This allows a more natural graduation that fits into the more limited range of the vision algorithms.
- Lens Distortion Correction: The lens distortion correction component compensates the deformation introduced by the camera lens that makes the projection no longer being rectilinear.

2.2. Localization of the Perseverance Bit Carousel

Fig. 4 shows the functions involved in the localization of the Perseverance Bit Carousel. The method detects concentric circles, corresponding to the Bit Carousel fasteners, and uses them as features to solve the Perspective-N-Point problem and obtain the 6D pose of the Bit Carousel with respect to the camera.

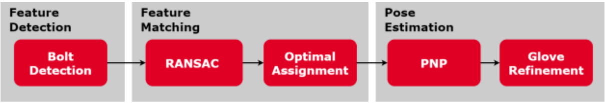

Figure 4. Steps of the Bit Carousel localization.

The different functions in which the method is divided are described below.

2.2.1. Feature Detection

The proposed feature detection method for the Bit Carousel scenario looks for concentric circles on the image corresponding to the bolts fixing the structure to the Perseverance. The method binarizes grayscale images using a locally adaptive threshold where the size of the mask (i.e. local neighbourhood) is set according to the radius of the circle. Then, the centroids of black and white regions are computed and matched according to their Euclidean distance. A concentric circle is detected if the radii of a pair of black-white regions, which is proportional to the square root of the area, fulfils a set of constraints given by the expected size of the bolt in pixels. Fig. 5 shows an example of the proposed method where the centre of the bolts is found.

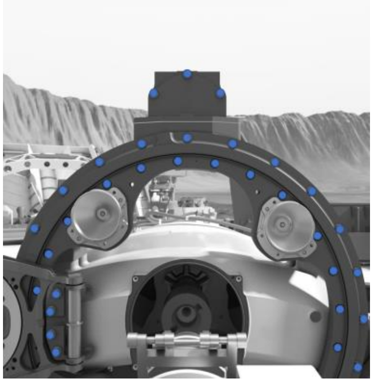

Figure 5. Detected bolts in the Bit Carousel

2.2.2. Feature Matching

The feature matching method used for the Bit Carousel scenario consists of three stages: *Modelling*, *Model Fitting* and *Feature Assignment*.

In the *Modelling* stage, a 2D model of the Bit Carousel features is computed. A model consisting of the Bit Carousel bolt centres in an arbitrary reference frame set at its central point is obtained from a 3D model of the Perseverance. An image reprojection of the model is computed for the (inaccurate) initial estimation of the Perseverance location. The resultant points are transformed into polar coordinates and used to find a best-fit circle by solving for the circle equation through least squares minimization. The reprojected point that best fits the circle (i.e. with a smaller residual error) is chosen as a reference point and used to normalise the magnitude of the polar coordinates for all points. The resulting Bit Carousel feature model includes the polar coordinates of all reprojected points corresponding to the bolts with normalised magnitude respect to a reference point.

In the *Model Fitting* stage, a suitable image reprojection of the Bit Carousel feature model is found from a set of observed data, i.e. the detected features. A Random Sample Consensus (RANSAC) approach was adopted to cope with the feature detection method missing some features or detecting others that were not considered in the model, i.e. outliers.

In the first step, a sample subset is randomly selected from the detected features. A circle fit is performed similarly as described above (i.e. through Least Squares minimisation) using only the elements of this sample subset. Then, the estimated model parameters, centre and radius, are used to instantiate a Bit Carousel 2D model from the Bit Carousel feature model obtained in the Modelling stage. In the second step, the algorithm checks consistency between the instantiated Bit Carousel 2D model and the rest of the detected features. Detected features are considered outliers if they do not fit the instantiated 2D model within some error threshold. The procedure repeats iteratively these two steps until the mean absolute error is below a threshold or a maximum number of iterations is reached, keeping the best solution that satisfies a minimum number of inliers criteria.

In the *Feature Assignment* stage, the proposed method finds correspondences between the detected features and the points of the instantiated Bit Carousel 2D model obtained in the Model fitting stage. The optimal assignment problem is solved using the Kuhn–Munkres algorithm [8] where the tasks are the detected features, the agents are the points of the instantiated Bit Carousel 2D model obtained, and the cost is their Euclidean distance. The method minimises the cost by assigning one agent to each task and, thus, matching all detected features.

2.2.3. Pose Estimation

The pose estimation method solves the Perspective-N-Point (PnP) problem using the method in [5]. This method is able to estimate the pose of the camera given a relation between a set of 3D points in an arbitrary coordinate frame and their respective location in the projected 2D plane of the image. The method exploits a projective imaging model and automatic mechanisms for pose initialization and convergence. Thus, the relative pose between the Camera and the Bit Carousel is obtained.

This method is heavily dependent on the precision of the 3D points, as these 3D points are the reference for the estimated camera pose. In the case of the Bit Carousel, this means that the positions of the bolts used as the 3D model to solve the PnP should be precisely known. However, only the yellow parts shown in Fig. 6 are static while the rest of them can rotate around the longitudinal axis at the centre of the Bit Carousel, resulting in a possible uncertainty in the position of the bolts located in the inner ring of the Bit Carousel. For this reason and to improve the accuracy of the estimated pose, only the static subset of bolts located on the outer ring of the Bit Carousel is used to solve the PnP problem. However, the non-static bolts are still used in the feature matching step, as they provide valuable information to avoid errors while matching detected bolts to the projected 2D model.

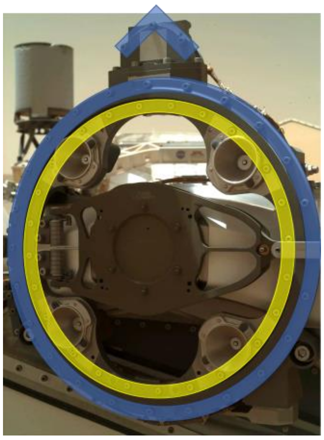

Figure 6. Bit Carousel diagram representing the static (blue) and dynamic (yellow) parts. Perseverance image courtesy of NASA/JPL.

The position of the glove within the Bit Carousel is also subject to additional mechanical errors. This means that on top of the positioning error with respect to the Bit Carousel, the error of the glove with respect to the Bit Carousel shall be added. To minimize this impact, a final stage of refinement in the tangential plane (there is little observability in the normal direction) is added. The algorithm relies on detecting the circles that can be seen on the glove of the RGA. In a first step, the RGA model is reprojected for the (inaccurate) pose of the Perseverance Bit Carousel. In a second step, each one of the reprojected circles of the model is matched in an iterative process with the closest detected feature that is yet to be matched. All matched circles whose centres exceed a certain distance are discarded.

After matching the reprojected model with the detected features, the centre of the glove is located. The pixel coordinates of the detected RGA centre are then subtracted from the previous estimation of the Bit Carousel centre to compute the pixel shift needed to alight the STA with the RGA. This pixel shift is then translated to distance using the intrinsic parameters of the camera and the depth estimation of the STA pose with respect to the Perseverance Bit Carousel.

2.3. Localization of the OS Container

The main functions involving the localization of the OS Container are depicted in Fig. 7. It follows a similar strategy to the localization of the Bit Carousel. In this case, rather than detecting the fasteners, the goal is to detect the OS slots.

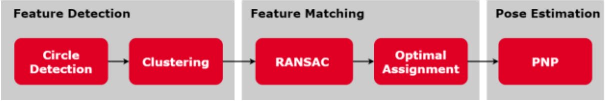

Figure 7. Steps of the OS localization.

The detection of the slots allows estimating the relative pose of the camera with respect to the OS. However, inserting the RGA in certain slots may cause the OS container to be occluded as the STA approaches the target. For instance, when inserting an RGA in the slots at the top of the OS container multiple slots can be occluded by the STA. In these cases, the STA orientation is chosen to minimise occlusions. For instance, for the slots at the top if the OS, the STA will approach the OS container upside-down in a way that no slots become occluded (see Fig. 8)

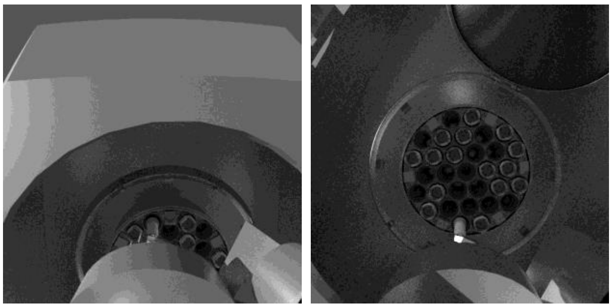

Figure 8. Insertion in the first slot. Left, without rotation of the arm; right, rotating the arm to maximize visibility of the OS.

2.3.1. Feature Detection

The feature detection method looks for circular features corresponding to the OS Container slots and the inserted RSTAs. Interestingly, the use of features corresponding to the inserted RSTAs was found to help the OS Container calibration, especially in cases where most of the slots contained RSTAs.

A batch of circular features is obtained using the ED Circles algorithm [1], which relies on a contiguous set of edge segments detected using [12]. The resulting batch often produces more features than needed as the OS container itself has a range of circular parts such as bolts or plates, and the inserted RSTAs can have a circular appearance when inserted.

To cope with these issues, the circular features in the batch are grouped using Density-based spatial clustering of applications with noise (DBSCAN) method [3]. It is a non-parametric clustering which allows coping with the variable number of slots, e.g., caused by occlusions or missed detections.

However, the cluster centroids are not always a good representation of the slot centre. The cluster feature that corresponds to the circle with a greater radius was found to be the most robust solution to the corresponding slot centre and, therefore, that criterion was used to choose the resulting features.

Fig. 9 shows the result of the slot detection. Blue lines correspond to the circles which have been detected but were discarded as slots. Yellow lines correspond to the circles which have assigned as slots (after clustering)

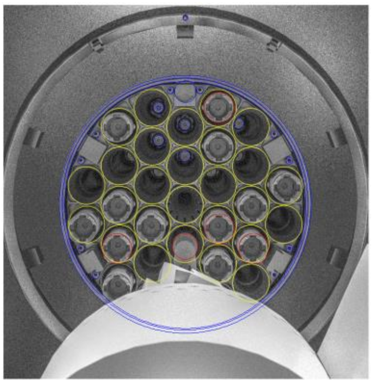

Figure 9. Detected (yellow) and discarded (blue) slots

2.3.2. Feature Matching

The Feature matching method used in the OS Container scenario is similar to the method presented in Section 2.2.2 for the Bit Carousel scenario. The main variations of this method are in the *Model Fitting* stage, where additional constraints are imposed to resolve possible indeterminations that can arise due to the OS Container geometry and expected camera perspective.

2.3.3. Pose Estimation

The Pose estimation method used in the OS container scenario is also similar to the one described in 2.3.3. The method again solves the PnP using the method in [5]. A 3D model of the OS carousel slot centres and the detected/matched features are used to obtain the relative pose between the camera and the reference frame in which the model is defined. In this case all matches between the 3D model and the detected features obtained during the matching step are used to estimate the camera pose.

2.4. Localization of the Sample Tubes in the Martian Terrain

Computing the RGA's pose from a single image is a challenging problem and could not allow fulfilling the accuracy requirements due to a lack of observability. The STA is mounting a monocular camera, which imposes observability limitations in depth perception, particularly for the inclination over the terrain and the Z distance.

Furthermore, no clear distinct features can be determined which could be detected in the image and matched against corresponding 3D features in the model. Previous works on a similar problem [6][11] showed that the pose determination based on keypoints extracted from the RGA might not be robust enough.

A multi view capture from different camera orientations is proposed to overcome these limitations and to provide a feasible solution for the 5DOF pose estimation of the RGA (the sixth degree of freedom, the rotation of the RGA along its longitudinal axis, is not required for its grasping and later operations). The set of images is captured as rotations around the projection over the terrain of an approximate RGA main axis. The accuracy of this knowledge does not influence the accuracy of the pose estimation (only a rough indication is required to acquire the images from meaningful viewpoints).

The number of views required and the angle between them is linked to the accuracy of the arm telemetry.

The algorithm pipeline is shown in Fig. 10.

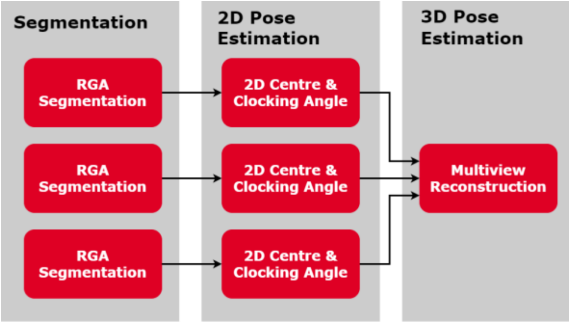

Figure 10. Steps of the RGA localization.

2.4.1. RGA Segmentation

The first step to compute the 5DOF pose is to segment the RGA in each one of the different acquired views. The segmentation step is performed by a custom deep learning model, which can cope with a wide range of poses and illumination conditions. The developed model is based on the popular U-Net convolutional neural network and consists in an encoder-decoder solution that produces a binary mask from a single 1-channel grayscale image. The main difference between the original U-Net [9] architecture and the developed model is that the encoder section of the network was replaced by an image classification encoder pretrained with the ImageNet [10] dataset.

The model was trained with real images of RGAs on simulated Mars environments, as well as computer generated scenes. This hybrid training approach has the big advantage of allowing the model to learn the shape of the RGA with many synthetic images while also ensuring a good performance with real images that are much more time-consuming to acquire. To further reduce the time invested in creating the dataset of real images, the ground truth mask generation was semi-automated using the Segment Anything [4] model, which requires an initial input and is much more computationally expensive than the developed solution.

The hybrid training approach, together with a data augmentation pipeline containing random flips, rotations, and perspective transformations achieves good performances with only hundreds of real images and a relatively compact model of 5.5 million parameters.

Fig. 11 shows an example of the segmentation function applied on an image of a real RGA placed on a sandbox. Note how the projected shadows (recreating those projected by the STA) are correctly handled and the RGA is accurately segmented.

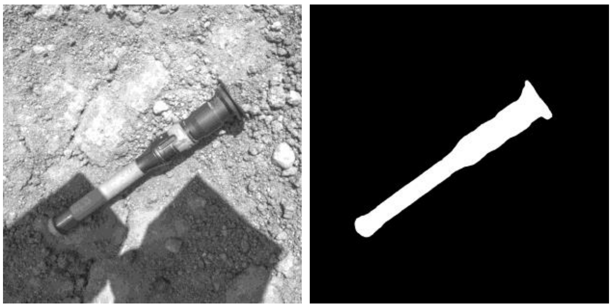

Figure 11. Example of the RGA segmentation function.

2.4.2. 2D Pose Estimation

Once a segmentation mask has been obtained, the next step is to estimate its centre (or another arbitrary point at a known distance from the RGA endpoints) and clocking angle.

The approach followed is to register the extracted segmentation mask into a single object with the shape of an RGA. It is based on Iterative Closest Point (ICP) [2] algorithm with a binary model of the RGA.

This method minimizes the distance between corresponding cloud points, computing the best alignment between the extracted mask and the object position. This also allows for removing false positive blobs and segmentation mask defects while preserving the shape of the RGA.

The reference mask of the RGA is computed based on an initial guess of the distance from the camera. To cope with the error of this guess, the ICP is iterated with at slightly scaled versions of this distance to get the best fit with the model.

Figure 12 shows the result of the registration process and how the centre and clocking angle is derived. The left image depicts the generated reference image based on the knowledge of the RGA shape and the distance to it. The centre (or more specifically, the grasping point) is also shown, as well as the main RGA direction. The right image shows the result of registering this reference mask with the mask obtained during the segmentation step described before. The result of this registration is the estimated grasping point and direction (clocking angle) of the RGA in the image acquired by the camera.

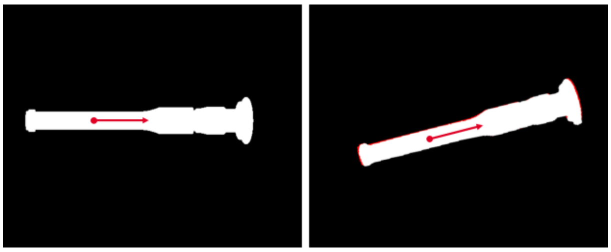

Figure 12. Left, reference mask with grasping point and main direction; right, result of registering the segmented mask with the reference.

2.4.3. 3D Pose Estimation

Coupling the RGA position and heading on the image plane for the different images (and the camera intrinsic calibration) with the telemetry from the robotic arm at the different captures it is possible to perform a multi-view reconstruction and compute the 5DOF RGA pose. The RGA centre point is computed by intersecting the vectors from each of the camera positions to the centre point. Note that these two lines do not necessarily cross exactly at a 3D location, hence it is possible to perform a sanity check. If the minimum distance between two lines exceeds a certain distance, an alarm can be raised meaning that either the RSTA detection might not be good, or the telemetry of the robotic arm is not correct. On the contrary, if the minimum distance between the lines is small enough, the 3D point that minimizes the distance between the lines is chosen as the RGA centre point.

The 3D axis of the RGA is computed by intersecting the planes that contain the RGA centre point, its main direction and each of the camera locations (provided by the robotic arm telemetry).

Fig. 13 illustrates the process of estimating the 5DOF pose based on 2D estimates.

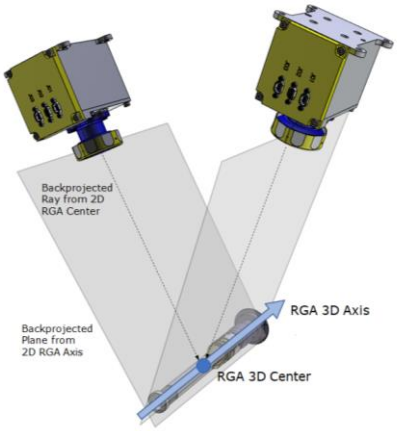
Backprojected Ray from 2D RGA Center
Backprojected Plane from 2D RGA Axis
RGA 3D Axis
RGA 3D Center

Figure 13. 5DOF Pose Estimation

2.5. Localization of the OS Lid and Workbench

The localization of both the OS Lid and Workbench will be performed using visual markers. The type and disposition of these elements is still being defined by NASA/JPL, taking into account the available space and the accuracy requirements.

3. RESULTS

The testing and evaluation of the algorithms have been performed using synthetic images. This allows to more easily recreate different environmental conditions (such as illumination), OS occupancy, camera configurations, etc. The potential drawback of using synthetic imagery is related to the quality and representativeness.

To reduce this gap, we opted for using computer graphics software implementing accurate global illumination algorithms. Special effort was also put to properly define the materials of the 3D models. A commercial 3D model of the Perseverance was used to provide a good starting point for the entire environment and for the rover. Then, the Bit Carousel of this model was replaced with the CAD model provided by NASA/JPL. The materials of the Bit Carousel were then adjusted (the CAD model did not have any appearance attributes) to match the sample images taken in the clean room provided by NASA/JPL. Fig. 14 shows a sample rendering of the rover in a Mars-like environment and already with the realistic Bit Carousel model integrated.

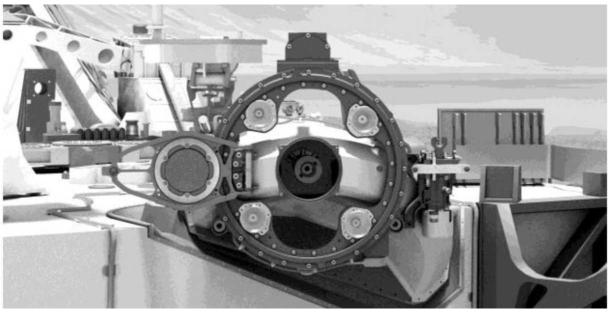

Figure 14. Rendering of the Perseverance rover already integrating the Bit Carousel CAD.

Lens distortion was also simulated following the profile provided by camera manufacturer based on their lens design. All possible illumination directions have been considered, covering the entire upper hemisphere to ensure any eventual impact of the Sun direction and shadows was analyzed.

3.1. Bit Carousel Localization

Table 1 summarizes the accuracy of the Bit Carousel localization in camera frame and obtained at different distances (corresponding to the Far, Intermediate and Close Viewpoints). Note that the distances are given from the arm end-effector to better understand how close or far is the arm from the target. From a vision point of view, the camera is placed backwards approximate 27 cm from the end effector.

Table 1. Localization errors at different distances from the STA End Effector

Component	50 cm	10 cm	3 cm
Trans X [mm]	0.266	0.065	0.049
Trans Y [mm]	0.412	0.155	0.14
Trans Z [mm]	0.754	0.328	0.412
Trans Mag [mm]	0.830	0.331	0.409
Rot X [deg]	0.234	0.104	0.067
Rot Y [deg]	0.121	0.049	0.189
Rot Z [deg]	0.043	0.024	0.022
Rot Mag [deg]	0.414	0.287	0.201

The largest translation errors are obtained in the Z component (the depth) whereas the largest rotation errors are obtained in the X and Y components. This is a typical behaviour of 3D from 2D problems where there is much higher observability in the camera plane (XY translations and Z rotations) than in the depth (Z translations and XY rotations).

The translation error at 3 cm is slightly higher than at 10 cm. The disappearance from the FOV of a few fasteners in the bottom part introduced a small bias in the depth estimation. Rotation error is however better at 3 cm. It is also important to remark that above values are under a perfectly calibrated camera. Due to thermal variations (and even if the STA camera is planned to be calibrated over the temperature), the final accuracy of the vision will be slightly lower.

3.2. OS Localization

Table 2 shows the error of OS localization algorithm at the different distances. In this case, they are given from the tip of the grasped sample tube to the OS. The camera is placed approximately 47 cm from the tube tip. The impact of targeting at different slots of the OS is negligible due to the rotation applied to the arm (see Section 2.3) to maximize observability. The occupancy level of the OS (whether is empty or full of RSTAs) also has little effect. The results, nevertheless, are provided for the worst-case configuration.

Table 2. Localization errors at different distances from the STA End Effector

Component	50 cm	10 cm	1 cm
Trans X [mm]	0.234	0.042	0.035
Trans Y [mm]	0.330	0.200	0.089
Trans Z [mm]	6.557	1.261	0.532
Trans Mag [mm]	6.501	1.240	1.263
Rot X [deg]	2.835	0.733	0.690
Rot Y [deg]	2.246	1.547	0.775
Rot Z [deg]	0.173	0.092	0.109
Rot Mag [deg]	4.861	1.659	1.164

As in the Bit Carousel calibration, the largest translation errors are obtained in the Z component, whereas the largest rotation errors are obtained in the X and Y components. Again, this is a typical behaviour of 3D from 2D problems.

Errors are larger than in the Bit Carousel calibration. This is because both to the camera is further from the target (approximately 20 cm further) and because the OS is smaller than the Bit Carousel, which means that the features are more concentrated in the centre of the FOV and not spread over the entire image, which reduces the observability.

Sensitivity to the accuracy of the initial guess has also been studied, showing that the algorithm is robust to several degrees (15 deg) in orientation and several centimetres (~15 cm) in translation.

3.3. RGA Localization

Testing of the RGA Localization is still an on-going activity at the time of writing. The segmentation step has been widely tested using both synthetic and real imagery of an almost flight model RGA provided by NASA/JPL (physically and visually equal to the flight unit and only missing some of the internal mechanisms). A COTS camera with equivalent characteristics (field of view and sensor) to the camera to be mounted STA was used and different specific sandboxes replicating the appearance of the Mars terrain were also utilized. Furthermore, to confirm the robustness of the trained model, many more images taken with a mobile phone (and, naturally, never used in the training) were also tested. In all cases, the quality of the estimated mask was very high, leading to very accurate segmentations.

The registration of these mask with the reference model resulted in accuracies better than 1mm in the estimation of the RGA centre in the tangential plane and less than 0.4 degrees in the clocking angle when observing the RGA from approximately 40 cm from the camera.

As already stated in 2.4, the accuracy of the 3D reconstructed pose of the RGA is dependent on the accuracy of the arm telemetry, but also on the number of views and angle span of such views (for instance, an angle span of 45 degrees results in a better estimation of the depth than an angle of 15 degrees). The final expected accuracy of STA is still being determined and therefore, the definition number of required viewpoints and angles between them is still pending to maintain the resulting accuracy of the RGA estimate below the system requirements.

4. CONCLUSIONS AND FUTURE WORK

The design, development and testing of the vision-based localization algorithms involved in the operations of the MSR-STA have been presented.

The algorithms have been specifically developed to take advantage of the unique characteristics of each of the elements to be localized. For the Perseverance Bit Carousel, the fasteners were used as relevant features to be matched against a reference model and solve the PnP problem. A similar approach was followed for the OS container, but in this case, the circular shape of the RSTA slots was used. On the other side, the localization of the RGAs in the terrain needed to follow a multi-view strategy to cope with the lower observability of the depth and inclination of a monocular observation and particular shape of the sample tube. Achieved accuracies are in all cases compatible with the requirements of autonomy for the STA operations.

Operations involving the workbench and OS lid are still pending its final definition as well as the type and disposition of the visual markers. Once this is performed by NASA/JPL, the integration of the corresponding marker detection algorithm will be performed.

Validation with real mock-ups and a COTS camera with similar characteristics to the one mounted on the STA is already foreseen in the short future. Both mock-ups of the Bit Carousel and OS will be provided by NASA/JPL to be visually almost equivalent to the flight models, to ensure the representativeness of the tests.

5. REFERENCES

[1] C. Akinlar, and C. Topal. “EDCircles: A real-time circle detector with a false detection control.” *Pattern Recognition*, **46**(3), 725–740 (2013).

[2] P. J. Besl, and H. D. McKay. “A method for registration of 3-D shapes”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(2): 239–256 (1992).

[3] M. Ester, H-P. Kriegel, J. Sander, and X. Xu. “A density-based algorithm for discovering clusters in large spatial databases with noise.” *In Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 226–231 (1996).

[4] A. Kirillov et al, “Segment Anything”, arXiv: 2304.02643, 2023.

[5] M. Lourakis, and X. Zabulis. “Model-Based Pose Estimation for Rigid Objects.” *International Conference on Computer Vision Systems*. ICVS 2013.

[6] L. M. Mantoani, R. Castilla, G. J. Paz, C. J. Pérez del Pulgar, and M. Azkárate, “Samples Detection and Retrieval for a Sample Fetch Rover.” *16th Symposium on Advanced Space Technologies in Robotics and Automation* (ASTRA), 2022.

[7] R. C. Moeller et al. “The Sampling and Caching Subsystem (SCS) for the Scientific Exploration of Jezero Crater by the Mars 2020 Perseverance Rover.” *Space Sci Rev* **217**, 5 (2021).

[8] J. Munkres. “Algorithms for the assignment and transportation problems.” *Journal of the society for industrial and applied mathematics*, **5**(1), pp. 32-38 (1957).

[9] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, arXiv:1505.04597, 2015.

[10] O. Russakovsky et al, “ImageNet Large Scale Visual Recognition Challenge”, arXiv: 1409.0575, 2015.

[11] I. R. Tiñini, I. Amat, T. Wiese, L. Bielenberg, and R. Detry. “Sample-Tube Pose Estimation Based on Two-Stage Approach for Fetching on Mars”. *16th Symposium on Advanced Space Technologies in Robotics and Automation* (ASTRA), 2022.

[12] C. Topal, C. Akınlar, and Y. Genç, “Edge Drawing: A Heuristic Approach to Robust Real-Time Edge Detection.” *2010 20th International Conference on Pattern Recognition*, 2424–2427 (2010).