AAS 22-132

IMPROVEMENTS AND APPLICATIONS OF THE DART VISION-BASED NAVIGATION TEST-BENCH TINYV3RSE

Paolo Panicucci[*], Mattia Pugliatti[†], Vittorio Franzese[‡], and Francesco Topputo[§]

Image processing and vision-based navigation algorithms require images for design, testing, and validation. For space exploration purposes, it is complex if not impossible to retrieve realistic images. To mitigate this, two approaches can be used: high-fidelity rendering of celestial bodies or hardware-in-the-loop testing. In this work, we focus on the latter by elaborating on the design, implementation, validation, and calibration of a vision-based navigation test bench called TinyV3RSE. The design of such facility has been a collaborative effort at the Deep-space Astrodynamics Research & Technology (DART) group, which will benefit from its usage in various projects and missions in which is involved. In this work the facility design, the current calibration procedure, and also some preliminary results are presented. These are focused on the image processing in a small-body mission and on the performance of an optical navigation algorithm about the Moon.

INTRODUCTION

Solar System exploration is booming. The recent discovery of ocean worlds, the scientific and economic interest for small bodies, and the future human and robotic exploration of the main ones are pushing for numerous mission concepts aimed at the massive exploration and exploitation of the Solar System.[1] The increasing number of space missions, also promoted by deep-space small satellites, raises issues in how to operate them. First, traditional ground-based control centers and tracking facilities (e.g., the Deep Space Network – DSN) have limited number of communications slots. Second, the impetus for compelling science implies riskier operations, such as touch-and-go and landing on small bodies, or real-time and accurate state estimation, which is currently limited by the delayed DSN communications and cut-off time. Third, the current approach to spacecraft navigation implies high costs due to human-in-the-loop intervention that could become a critical driver with the increase of mission number.

Autonomous vision-based navigation (VBN) and robust Image Processing (IP) are crucial to overcome these limitations. This is because of the capability of such algorithms to provide near real-time information to navigation filters on-board. Among all the navigation sensors available on the market, cameras are usually preferred as they are light, compact and low power demanding when compared to other advanced navigation sensors, such as LIDARs. For these reasons the use of passive cameras, in combination with IP algorithms, provides compelling navigation performances with light and cost-effective hardware.

[*]Postodctoral Research Fellow, Department of Aerospace Science and Technology, paolo.panicucci@polimi.it
[†]PhD Candidate, Department of Aerospace Science and Technology, mattia.pugliatti@polimi.it
[‡]Postodctoral Research Fellow, Department of Aerospace Science and Technology, vittorio.franzese@polimi.it
[§]Full Professor, Department of Aerospace Science and Technology, topputo.francesco@polimi.it

IP and VBN require images for design, testing, and validation. In exploration missions, it is complex if not impossible to gather images to perform such studies during the design phase of the algorithms. This is due to several reasons. First, even though the number of interplanetary missions is increasing, few missions are flown every year. This implies limited publicly available databases to be exploited, which could lead to limited testing capabilities. Second, archived images are associated with a spacecraft state estimate from the orbit determination solution. This, however, is affected by navigation errors that make the true spacecraft state not available if not with uncertainty associated with it. Thus, the VBN state estimate cannot be compared against the true spacecraft state and image renderings could lead to rendering errors.[2] Furthermore, available datasets are associated with specific missions. This implies that images are directly constrained to the flown trajectory and designed spacecraft. Given these constraints, it is not possible to change illumination conditions, camera parameters, and the dynamical context.

As space missions data are affected by navigation errors, two approaches are usually used to address the testing of VBN algorithms: high-fidelity rendering simulations or hardware-in-the-loop (HIL) testing. High-fidelity rendering engines, such as ESA's PANGU[3] or Airbus Defence & Space's SurRender,[4] enable reproducing the data received at the processors by simulating not only the light propagation from the source to the sensor, but also the camera electronics.[5] On the contrary, HIL test benches rely on the manufactured sensor or a flight model in a laboratory environment to characterize the performances without any sensor simulation.

The former solution is generally more convenient as a broad series of sensors can be implemented without any additional expense in terms of equipment. Moreover, it enables the true spacecraft trajectory to be simulated without any constraints on the camera position, as is the case with robotic arms. A major drawback of this approach is the validation of the rendering simulator in reproducing the real world. Validation procedures could last years, if not decades, which lead to costs growth. Moreover, additional issues of this approach are the software licenses. Some of them, as PANGU and SurRender, are not open-source, while others like Blender* or POVray† are not specifically designed for the rendering of celestial objects nor for engineering applications. The latter solution requires reproducing the mission dynamical environment and a trustworthy representation of the observed scene to generate images affected by error. The HIL test bench is generally more expensive - at least in terms of equipment - as the representative dynamics or the observed scene requires costly hardware as robotic arms or terrains analogs. Moreover, the use of real sensors and simulation hardware implies efforts in calibration which are not needed for high-fidelity rendering engines since the sensor is simulated numerically. Despite these limitations, HIL test benches are generally preferred as the hardware used within the simulation loop is more representative of the mission scenario.

Historically VBN test benches have been designed on optical tables to characterize star trackers performances[6–11] or to simulate close proximity or landing operations in highly-non-linear dynamics exploiting robotic arms.[12–15] Only recently, work has been done to design and build test benches that aim to validate VBN algorithms and their integration with cameras.[16] This work presents one of these facilities: TinyV3RSE, the vision-based navigation test bench designed and built at the Deep-space Astrodynamics Research & Technology ‡ (DART) group. TinyV3RSE stands for Tiny Versatile 3D Reality Simulation Environment and it is used to support the design, validation, and testing of IP and VBN algorithms. This facility has been designed as a collaborative effort at the

*https://www.blender.org/, last accessed 20th November 2021.

†http://www.povray.org/, last accessed 20th November 2021.

‡https://dart.polimi.it/, last time accessed: November 5th 2021

DART group and will play a pivotal role in the current and future projects in which the group is involved.
The rest of the paper is organized as follows. Section provides a description of the facility design. Then, Section illustrates the TinyV3RSE calibration procedure. Section reports some preliminary results that have been obtained through imaging in the facility. Finally, Section summarizes the conclusions of this work.

HARDWARE TINYV3RSE DESIGN

Design drivers

TinyV3RSE is composed of three main elements: a screen, a collimator, and a camera, positioned as in Figure 4. When the light emitted by the screen passes through the collimating lens, it respects the thin lens equation. Thus:

$$\frac{1}{f_{\text{coll}}} = \frac{1}{d_{\text{r}}} + \frac{1}{d_{\text{i}}} \tag{1}$$

where f_{coll} is the collimating lens focal length, d_{i} is the distance between the collimating lens and the image, and d_{r} is the distance between the collimating lens and the object, i.e., the screen. Recall that f_{coll} is positive for converging lenses and negative for diverging ones. Moreover, note that d_{r} is positive when it is placed on the left side of the lens and negative otherwise. Finally d_{i} is positive when the image is generated on the right side of the lens, i.e. a real image is formed, and negative otherwise, i.e., a virtual image is formed. Figure 1 shows the geometrical configuration under study, which in this case generates a virtual image.
Equation 1 can be rewritten to explicitly compute the image distance as:

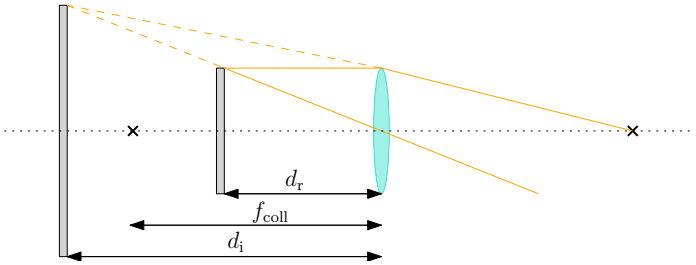


Figure 1: Geometrical configuration for the lens equation of the collimator in the case considered in TinyV3RSE.

$$d_{\text{i}} = \left(\frac{d_{\text{r}}}{d_{\text{r}} - f_{\text{coll}}} \right) f_{\text{coll}} \tag{2}$$

This shows that the object to be observed, i.e., the screen, must be placed at the focal length distance if it must be placed at infinity. The perfect design choice would be that the screen image fully fits the camera field of view (FOV). To simplify the design procedure, the vertical FOV is considered since it is smaller than the horizontal one. Under the assumption of perfect components' alignment, the problem can be studied as outlined in Figure 2. Thanks to basic geometrical relationships, it is easy to show that:

$$\tan\left(\frac{\theta}{2}\right) = \frac{h_{\text{s}}}{2 f_{\text{coll}}} \tag{3}$$

where θ is the camera FOV, h_s is the vertical screen size, and $\theta_1 = \theta_2 = \frac{\theta}{2}$. Equation 3 links the three components of TinyV3RSE, showing that their design choice is not arbitrary.
Note that, because of the collimation, the distance between the camera and the collimating lens d_{cam} is not a design parameter that depends on the collimator focal length. This parameter is important to determine the diameter of the collimating lens.[6] To avoid that the camera observes outside of the collimating lens, the following relation has to be satisfied:

$$R_{coll} \leq R_{cam} + d_{cam} \tan\left(\frac{\theta}{2}\right) \tag{4}$$

where R_{coll} is the collimating lens radius and R_{cam} is the camera lens objective radius. Moreover, to ensure to work in paraxial area of the collimating lens, i.e., where the thin lens equation hypothesis holds, d_{cam} must be chosen as small as possible.

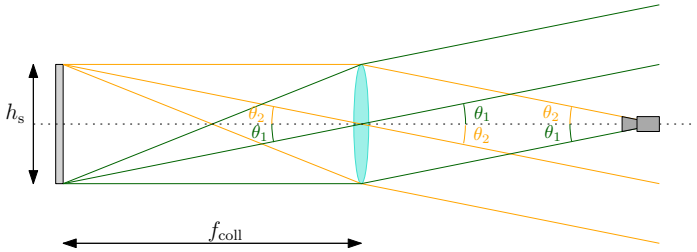


Figure 2: Optical configuration of the components of the TinyV3RSE facility.

Components description

TinyV3RSE is composed of three main modules mounted on an optical table:

1. **The camera**, rigidly mounted on its mechanical support which enables vertical translation, pitch, and yaw mechanical adjustments;

2. **The high-resolution screen**, whose orientation is set to ensure that the screen and the optical plane of the camera are parallel;

3. **The collimator**, which ensures that the light coming from the screen and entering the camera is simulated as coming from infinity (or from a very high distance). The collimator is mounted on an optical support that can rotate, change in elevation, and can be finely adjusted laterally and transversely.

These three modules are visible in the CAD model in Figure 3 and from the top view of the Tiny-V3RSE facility in Figure 4, while their relationships and functional connections are illustrated in Figure 5. Each module is now described in detail.
The camera currently in use is a Basler acA1300-22gm (CS-Mount)[§] with a 12mm C series fixed focal length lens[¶]. The key data-sheet characteristics of the camera assembly are a focal length of 12 mm, a resolution of 1280 pixels $\times$ 960 pixels, a pixel size of 3.75 μm $\times$ 3.75 μm, and a sensor size of 4.9 mm $\times$ 3.6 mm. The camera FOV is 22.6° $\times$ 17°. The camera is mounted on a dedicated

[§]Camera data-sheet, last time accessed: November 5th 2021
[¶]Lens data-sheet, last time accessed: November 5th 2021

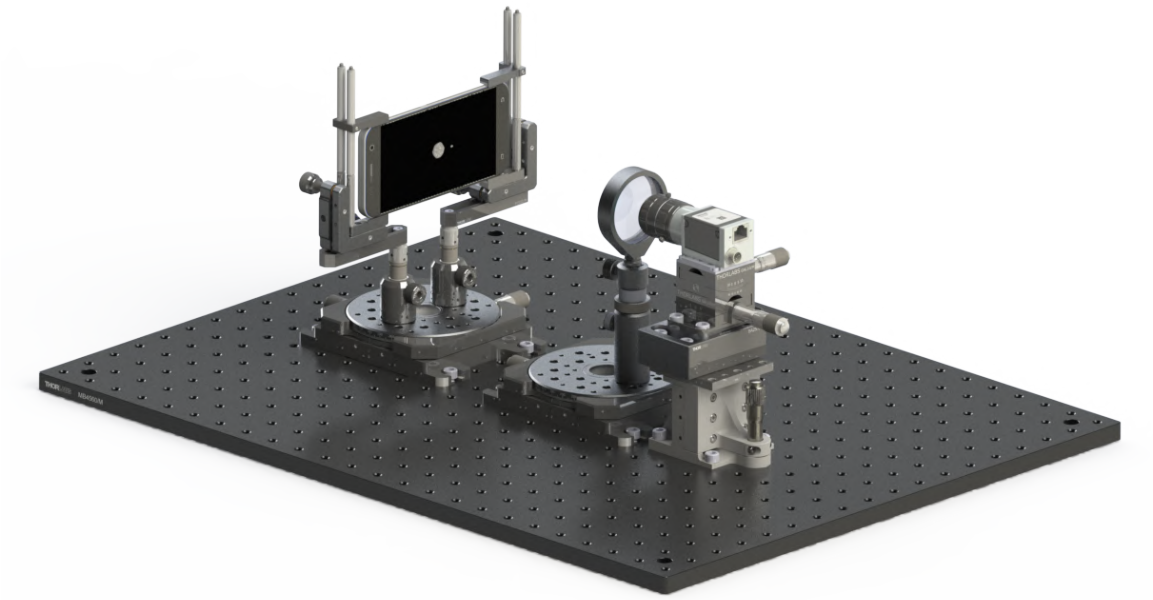


Figure 3: CAD model of TinyV3RSE

assembly which is composed of three parts. The first one is a vertical translation stage ensuring vertical control of the assembly. The second one is a goniometer enabling pitch and roll. The third and last part is a custom mounting adapter to interface between the camera and the optical assembly. In Figure 6 it is possible to see a close-up view of the camera assembly pointed towards the collimator. As explained in Section , the distance between the collimator and the camera is kept as small as possible.

The screen is represented by a Galaxy S7 smartphone[‖] with a resolution of 2560 pixels $\times$ 1440 pixels, a pixel size of 44.1 μm $\times$ 44.1 μm and a screen size of 112.9 mm $\times$ 63.5 mm. As for the camera and collimator, the screen is mounted on a dedicated assembly composed of two parts. The first one is a translational stage enabling movements of the screen on a plane parallel to the optical one. The second one is a screen holding mechanism that enables re-orientation by changing 4 pins disposed close to the screen's corners. The choice to use a commercial smartphone as a screen presents several advantages. First, in terms of the sizing of the facility, having a compact high-resolution screen makes it possible to position it within a limited distance from the camera-collimator assemblies, thus ultimately ensuring a compact facility. This is an advantage both in terms of laboratory space but also in terms of portability and eventual external testing since the optical test bench could easily be moved as carry-on luggage to a different location. Having a smartphone as a screen is also simple to set up the interfaces with the server. The smartphone is therefore a commercial, hence low-cost solution, which also possesses interesting properties in terms of image contrast. The screen used is an OLED one, which does not suffer from screen bleeding phenomena typical of Liquid Crystal Displays (LCDs) and exhibits a high contrast between inactive and active pixels. This is of particular interest in the rendering of the pitch-black background of a celestial scene, before considering camera noise. The smartphone as a screen solution exhibits also a drawback, which is given by

[‖]Screen data-sheet, last time accessed: November 5th 2021

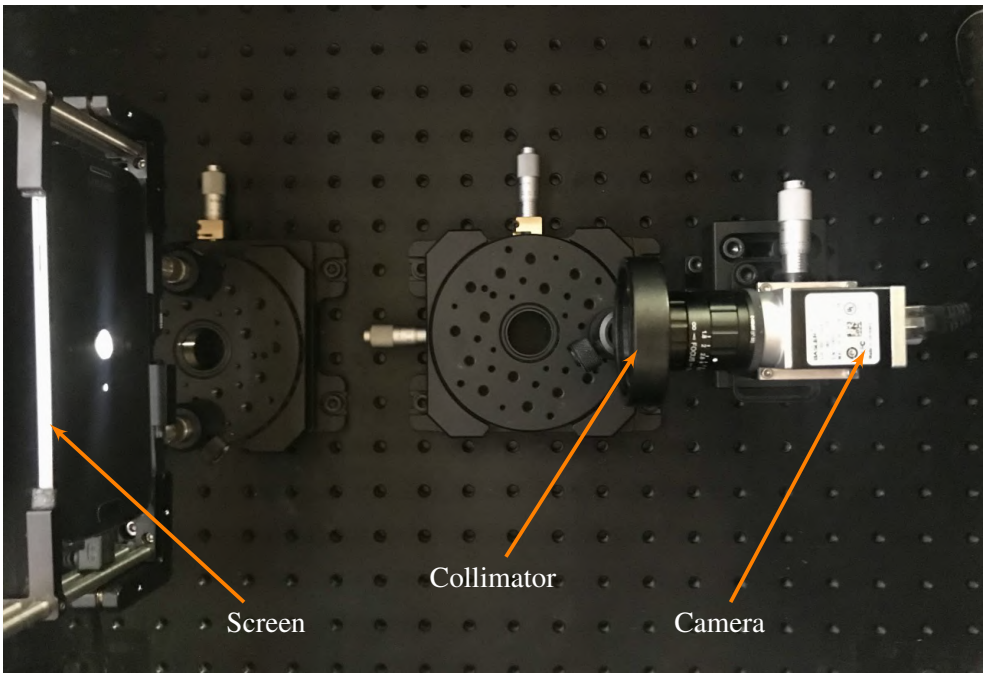


Figure 4: Top view of the TinyV3RSE facility.

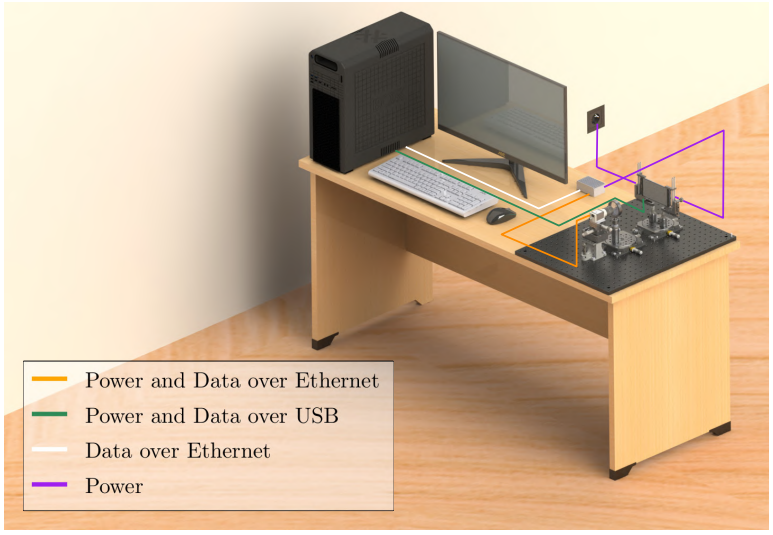


Figure 5: Functional architecture of the TinyV3RSE facility.

the screen resolution. Some facilities designed in the past seems to abide by an empirical sampling law for which each pixel of the sensor is to be stimulated at least by 4 pixels of the screen (or 1:2 if considered linear).[17,18] This is to ensure a continuous representation of the environment to the sensor and to satisfy Nyquist sampling theorem. However, at the time of the facility design, a higher resolution screen than the one considered has not been identified from existing commercial smartphones. The current setup has roughly a 1:1.43 ratio between sensor and screen pixels. It is also observed that this phenomenon did not seem to have played an important disturbance on the

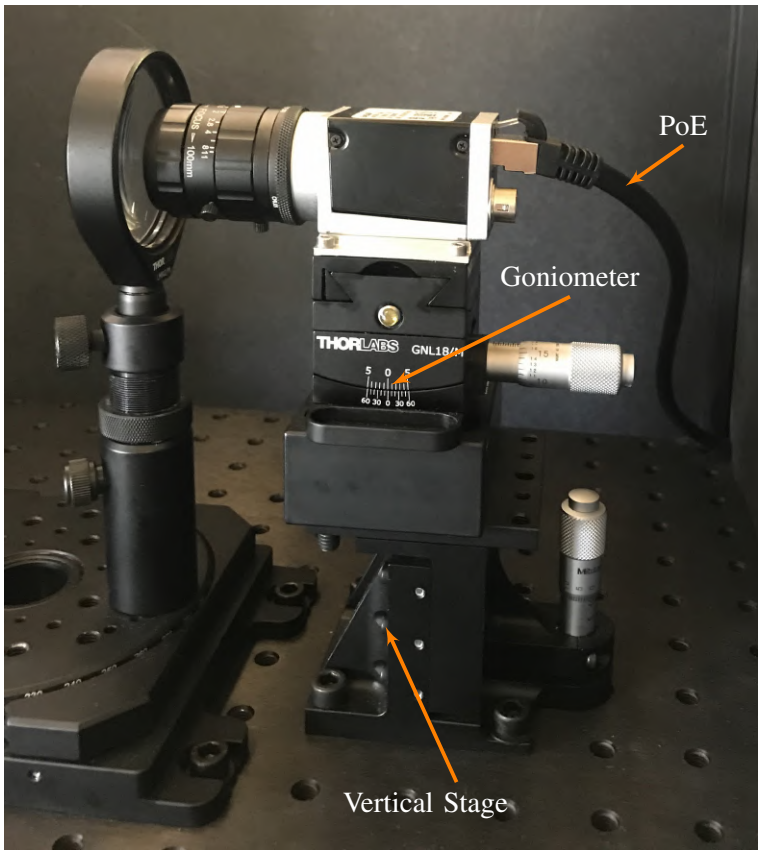


Figure 6: Close-up view of the camera and collimator assemblies.

performance of the algorithms tested so far.
Finally, the collimator used is a 2" diameter N-BK7 plano-convex lens (AR Coating: 350 - 700 nm)[**] with a focal length of 200 mm. The collimator is mounted on a dedicated assembly which is composed of a roto-translational stage and a post holder which is used to gain vertical alignment between the collimator and the camera.
By considering the screen and camera characteristics and using Equation 3, the camera should be placed at 211.7 mm in order to perfectly fit the screen vertical dimension with the vertical length of the camera's FOV. Because of that, the collimator has been chosen with a trade-off study among the plano-convex lenses available as off-the-shelf components. The selected one has been chosen to maximize the observed portion of the screen while avoiding vignetting. Note that, when an image is displayed on the screen, it has the size of 112.9 mm $\times$ 63.5 mm. By taking out the calculation with the camera FOV and a collimating distance of 200 mm under the hypothesis of perfectly aligned optical components, a coarse estimation give that only 80 mm $\times$ 60 mm of the screen is covered by the facility camera FOV. Thus the image taken by the facility camera is just a portion of the image displayed on the screen. This is important to be considered when operating and calibrating TinyV3RSE.
The functional architecture of the facility is illustrated in Figure 5. The screen and server are directly connected to a power outlet while the camera is exchanging data and power via a Power over

[**]Collimator data-sheet, last time accessed: November 5th 2021

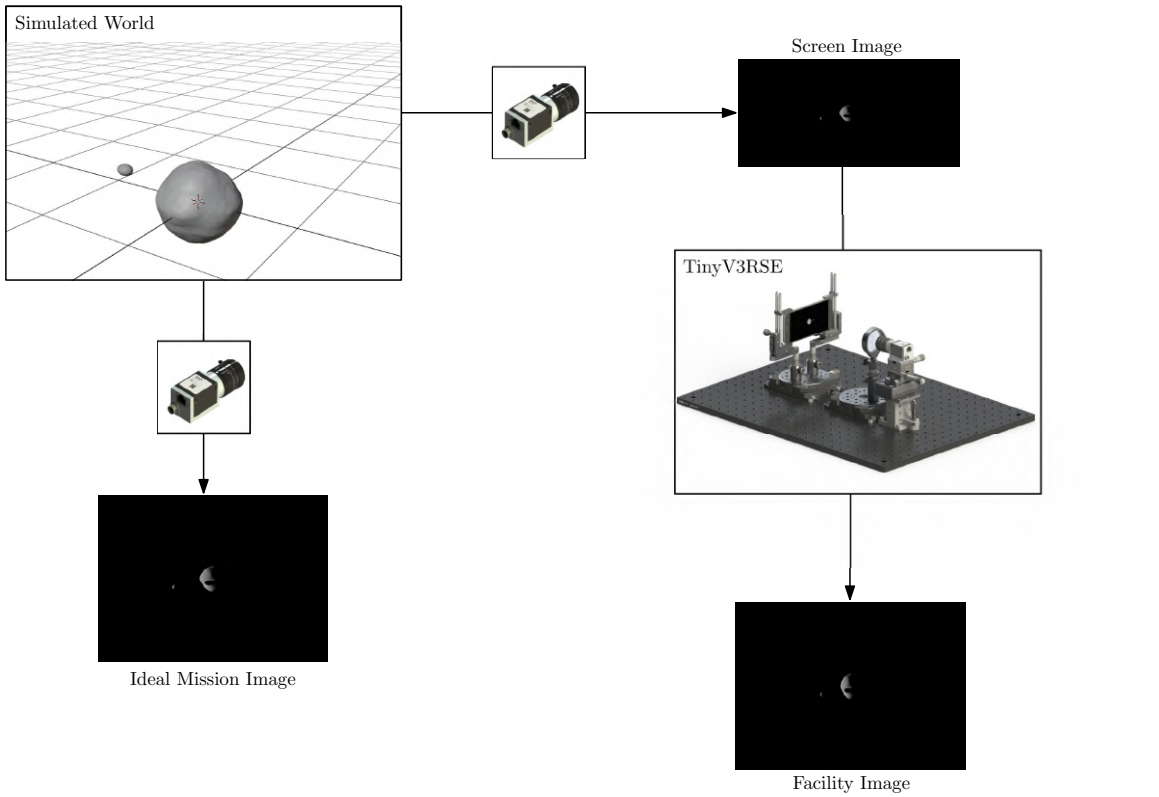


Figure 7: Functional work-flow in TinyV3RSE

Ethernet cable (PoE). Virtual scenes are rendered on the server with Blender or PovRay (depending on the user preference) and then sent to the screen via a Wi-Fi or USB connection. The server is also responsible for activating the camera, receiving and eventually processing the images obtained by the same. The screen, collimator, and camera are all enclosed in a box that is closed during the collection of the images. This is done to ensure that proper illumination conditions are met and that no light artifacts are generated on the screen due to the external conditions (such as reflections, external lighting, shadows of personnel working next to the facility).

In Figure 7 the functional workflow used to generate images with the facility is illustrated. The starting point is the simulated world, in which the physical and geometrical properties of the celestial bodies of interest are simulated in a virtual environment. Choosing a rendering software to do so is convenient since it enables sampling of such a virtual environment assuming a certain camera model positioned from a specific point of view. A rendering of a scene can thus be seen as a sampling of this simulated synthetic environment through the physical model of the camera. As it is possible to see from Figure 7, this is done twice for any given camera position: The first time to generate an image representative of what the mission camera would be seeing and the second time to get the scene to be projected on the screen at the proper resolution. These are respectively the ”Ideal Mission Image” and ”Screen Image” illustrated in Figure 7. Once the screen image is projected, it will then stimulate the collimator and camera, which will capture this scene with the real sensor properly positioned, given that a successful calibration ensures the correct alignment of all the components of TinyV3RSE. A final step is required to transform the image captured with the sensor in the facility to an equivalent version of the one captured in the virtual environment. This

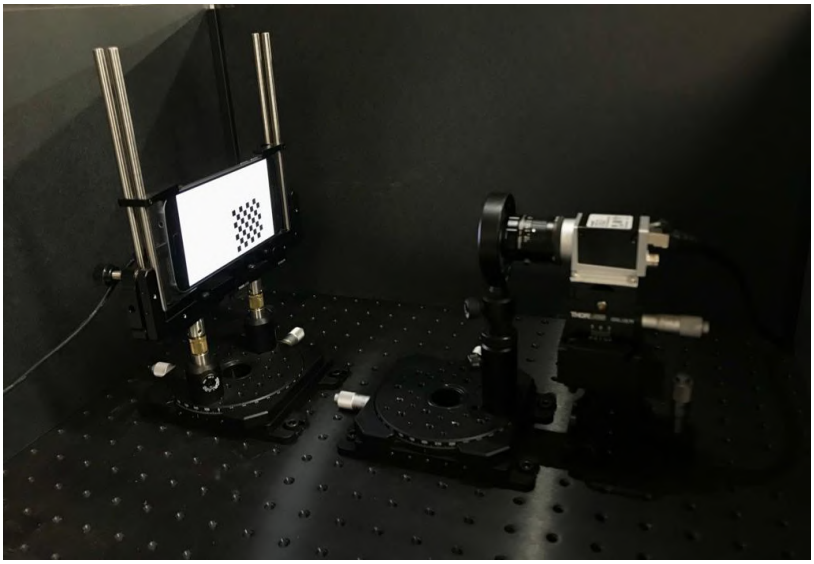
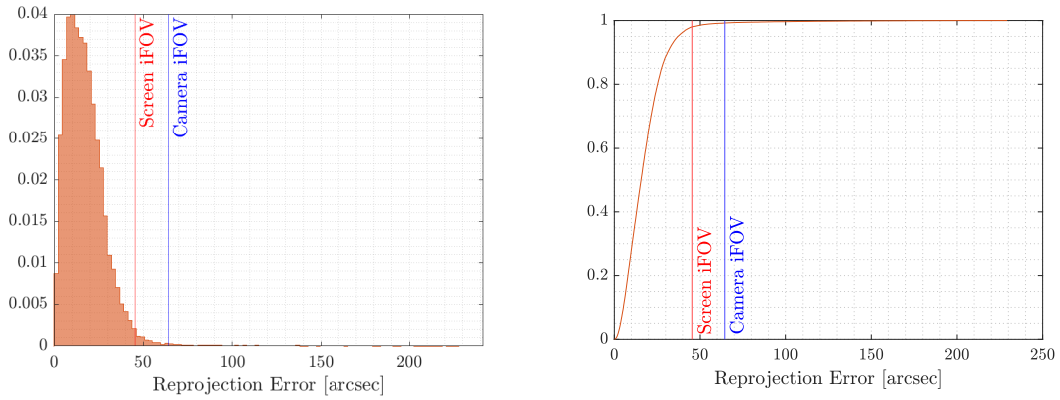


Figure 8: A picture of the TinyV3RSE test bench during calibration while using a checkerboard pattern.



(a) Calibration points reprojection error PDF

(b) Calibration points reprojection error CDF

Figure 9: Pixel reprojection error after the calibration procedure.

step is fundamental since, apart from calibration errors, these two images should be geometrically equivalent, yet photometrically different. The synthetic one has been generated with an ideal camera model, with no noise and perfect environmental conditions while the image from the facility encompasses noise and all phenomena typical of a sensor reading. The difference between these two images also represents the same domain gap between real and synthetic images which TinyV3RSE aims to reproduce for validation. A previous version of the facility presented in this work is also illustrated in Reference 19. In such a version, all the design choices are illustrated in detail, the interested reader is directed towards it.

TINYV3RSE SOFTWARE FUNCTIONALITIES

Geometrical Calibration

Before using TinyV3RSE, a geometrical calibration procedure is necessary. This section describes such procedure. First, the calibration is required to find the intrinsic camera matrix of the equivalent pinhole camera model for the camera mounted in the facility.[20] Second, the calibration is necessary to take into account the lenses distortion. When the light emitted by the display passes through the collimator and the optical head lenses, it is distorted by the lenses. Thus, the image projected into the screen is warped by the lenses apparatus, therefore changing the observed image. The distortion is a major factor when a series of lenses is used. A previous work[21] introduces various mathematical representations to model the optical distortion introduced by lenses. TinyV3RSE's calibration procedure allows for two different models to be used as distortion model: the tangential-radial distortion model[22] and the polynomial distortion model.[7,16] Lastly, the geometrical calibration is necessary to estimate the misalignment among the components in the facility. Large angular errors would cause wrong functioning of IP and VBN algorithms to be tested, which could invalidate their performances and comparability. These problems are solved in a sequential procedure. First, the camera mounted on the facility is calibrated with the algorithm proposed in Reference 23 to find the equivalent pinhole camera model. Then, the alignment of the screen with respect to the camera assembly is estimated by displaying on the screen a series of checkerboards with different orientations, as shown in Figure 8. The procedure followed is similar to the one presented in Reference 7.

At the end of the procedure, the camera intrinsic matrix, the calibration distortion model, and the facility misalignment are all estimated. This helps in converting the IP output to a consistent measurement as it would have been seen from the real camera without alignment errors from the correct view. To shown the possible output of the calibration procedure, the PDF and the CDF of the calibration points reprojection error are shown in Figures 9a and 9b. In these figures, the calibration points reprojection error distribution is compared with the camera iFOV (istantaneous FOV) and the screen iFOV. The screen iFOV is defined as the screen pixel angular size at camera position. Note that the screen iFOV is the angular size of the smallest point that can be displayed on the screen. The estimated calibration places 98% of the calibration points under the screen iFOV which implies that celestial objects are correctly displayed on the screen.

Camera PSF Estimation

An important step to be performed before the use of the facility is the characterization of the optical response of the camera in TinyV3RSE.

The radiometric optical response of a camera is represented by the camera Point Spread Function (PSF).[24] The PSF is the camera radiometric response to a point source stimulating it. In other words, it is the radiometric mathematical model from a black-box system perspective that link the stimulation and the image. It is thus of central importance to quantify it not only to model the camera under study, but also to quantify optical response, such as blurring, which could influence VBN performances.

Under the hypothesis of isotropic optical response, the PSF can be approximated by the Line Spread Function (LSF) which is the camera optical response to a line stimulus. The LSF is the marginalization of the PSF to the direction defined by the line.[24] This hypothesis is generally done to simply the PSF estimation to a easier problem which can be solved by the slant-edge method.[25]

This procedure is implemented in TinyV3RSE's software suite which enable to retrieve the best

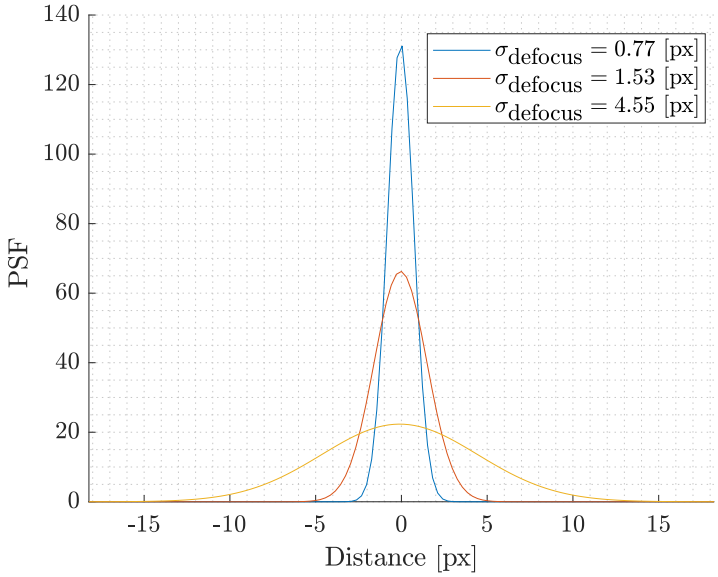


Figure 10: An example of the Gaussian LSFs obtained from the PSF estimation process.

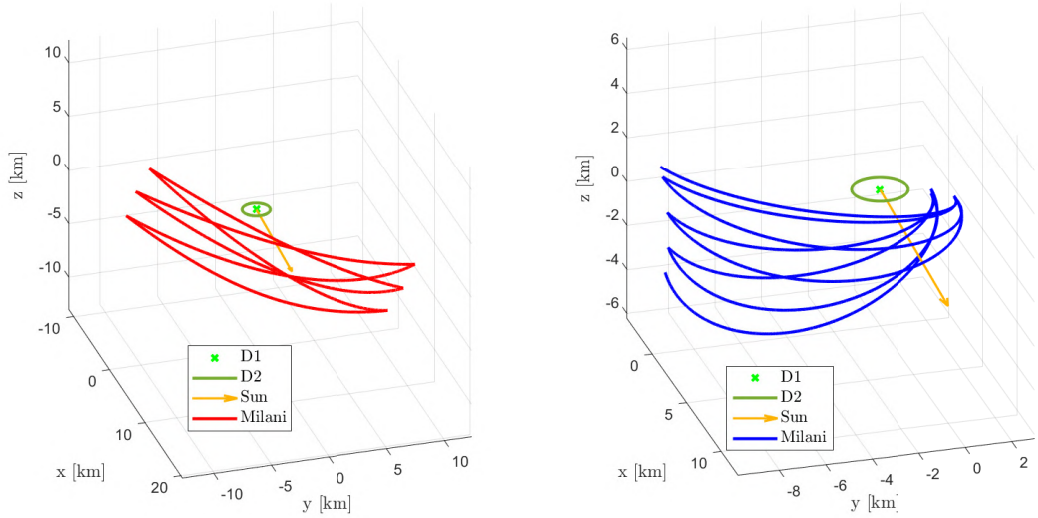
Gaussian fit of the LSF and its Fourier transform, the Modulation Transfer Function (MTF). This fully characterize the camera in the pixel and frequency domains. The output of this procedure is presented in Figure 10 for different values of blur. In the presented results the same calibrated camera takes several images of the same pattern with different blur level and the PSF is estimated from the blurred image. This enables the operator to quantify the blur and to perform a quantitative study on the VBN algorithm performances with changing blur.

PRELIMINARY RESULTS

In this Section, preliminary results that have been obtained by the DART group using the TinyV3RSE facility are illustrated. These results demonstrate the type of analysis that can be performed with the use of such facility within the framework of verification and validation (V&V) of IP and VBN algorithms.

Small bodies case

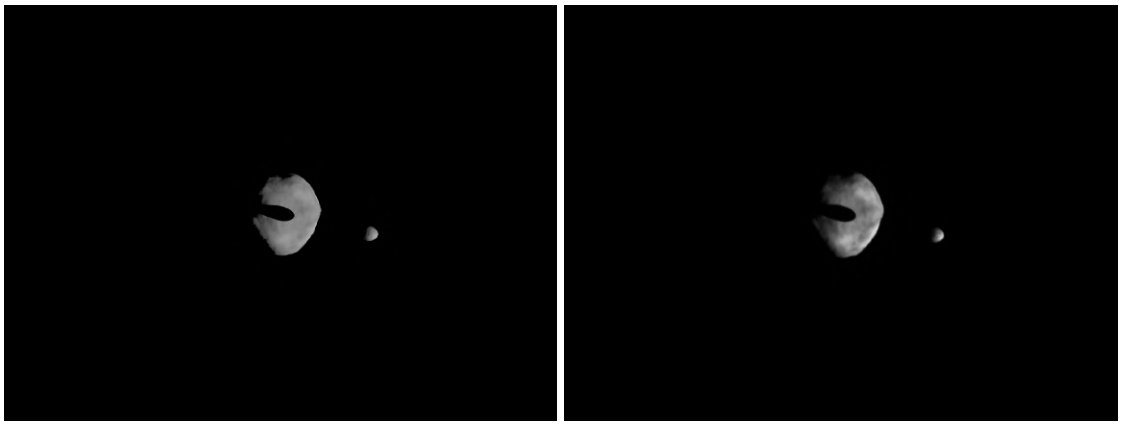
In this section an glance on the validation and verification of the Milani's IP is reported to show the applicability of TinyV3RSE within DART Lab's project.
Milani is a 6U CubeSat, part of ESA's Hera mission,[26] which will visit and characterize the Didymos binary system. Milani is a semi-autonomous spacecraft that combines innovative IP and autonomous navigation components with traditional guidance and control approaches. Its mission profile is characterized by two main phases: a Far Range Phase (FRP) lasting 21 days and a Close Range Phase (CRP) lasting 28 days. The nominal trajectory of the spacecraft during these phases is illustrated in Figures 11a and 11b. During both phases, the autonomous navigation mostly relies on observables produced by its IP algorithm. The IP algorithm analyzed hereunder works as follows. First, the raw image is manipulated to identify the primary of the binary system and extract key geometrical properties associated with its blob of pixels, e.g. centroid and best-ellipse fitting parameters. Second, the so-called WCOB (Weighted Center Of Brightness) is applied to determine a



(a) Far Range Phase (FRP)

(b) Close Range Phase (CRP)

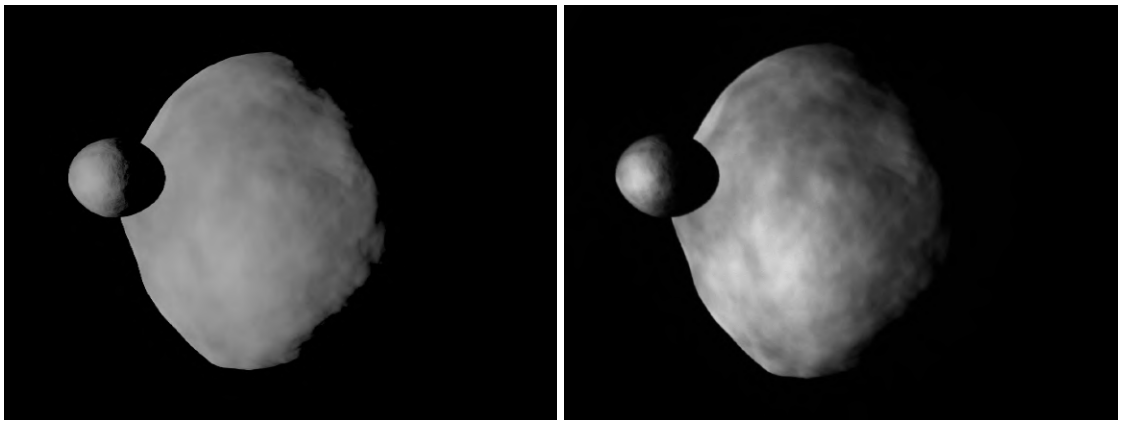
Figure 11: Two mission phases under study for the Milani mission (see Reference 27)



(a) Image taken by the camera in the virtual world

(b) Image taken by the camera in TinyV3RSE

Figure 12: Example of the image taken during the FRP



(a) Image taken by the camera in the virtual world

(b) Image taken by the camera in TinyV3RSE

Figure 13: Example of the image taken during the CRP

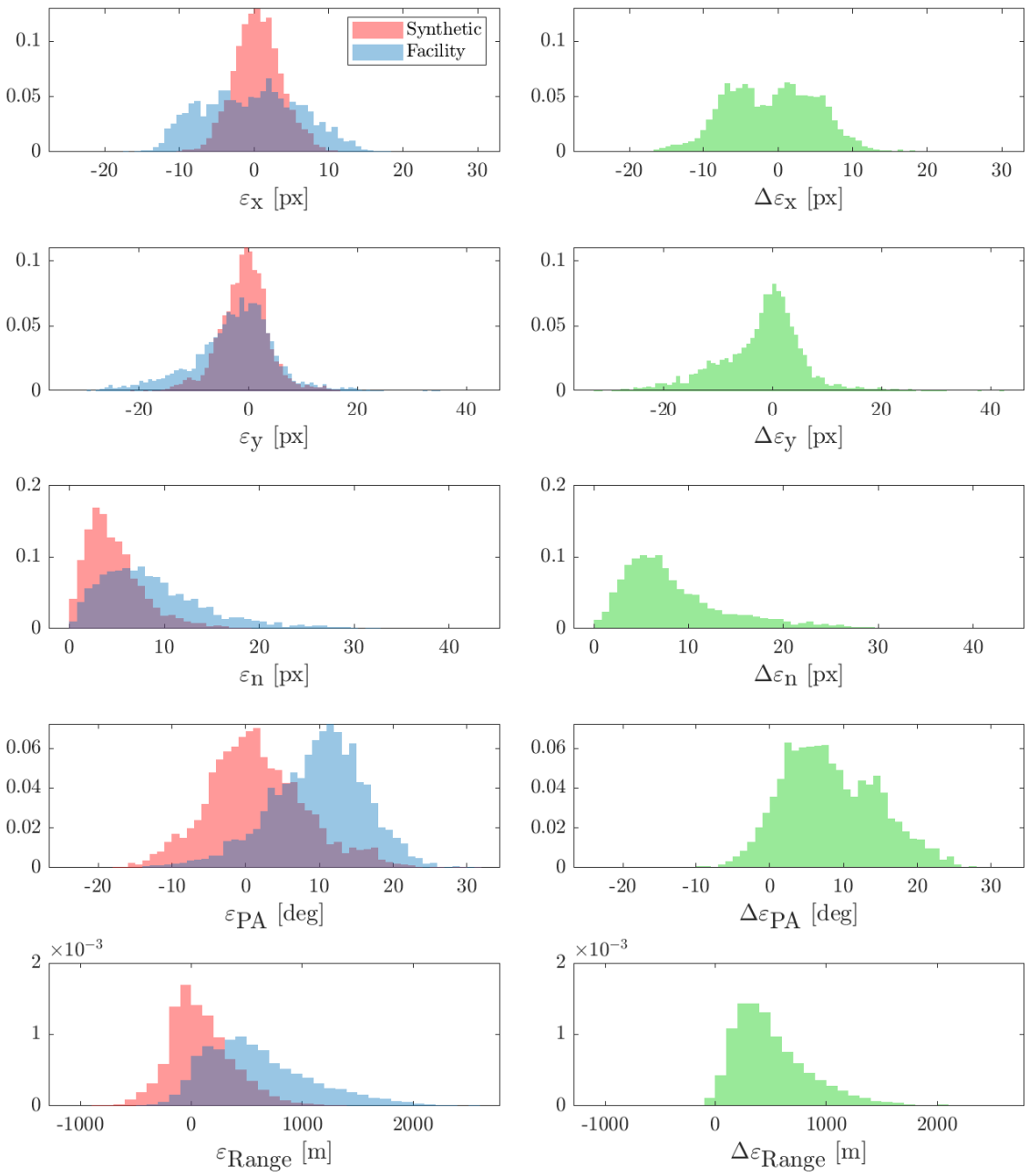


Figure 14: Results for the FRP dataset, shown as normalized histograms. Left: Estimation errors statistics. Right: Difference between the IP output on facility images and the IP output on synthetic images.

data-driven weighted correction on the centroid to approximate the primary Center of Figure (CoF). In the process, the phase angle (PA) is determined with data-driven functions too. Finally, the range is gathered with a simple apparent diameter formula. To summarize the output of the IP algorithm are estimates of the CoF, of the PA, and of the range. A more detailed discussion of the Milani's IP can be found in details in References 27 and 28.

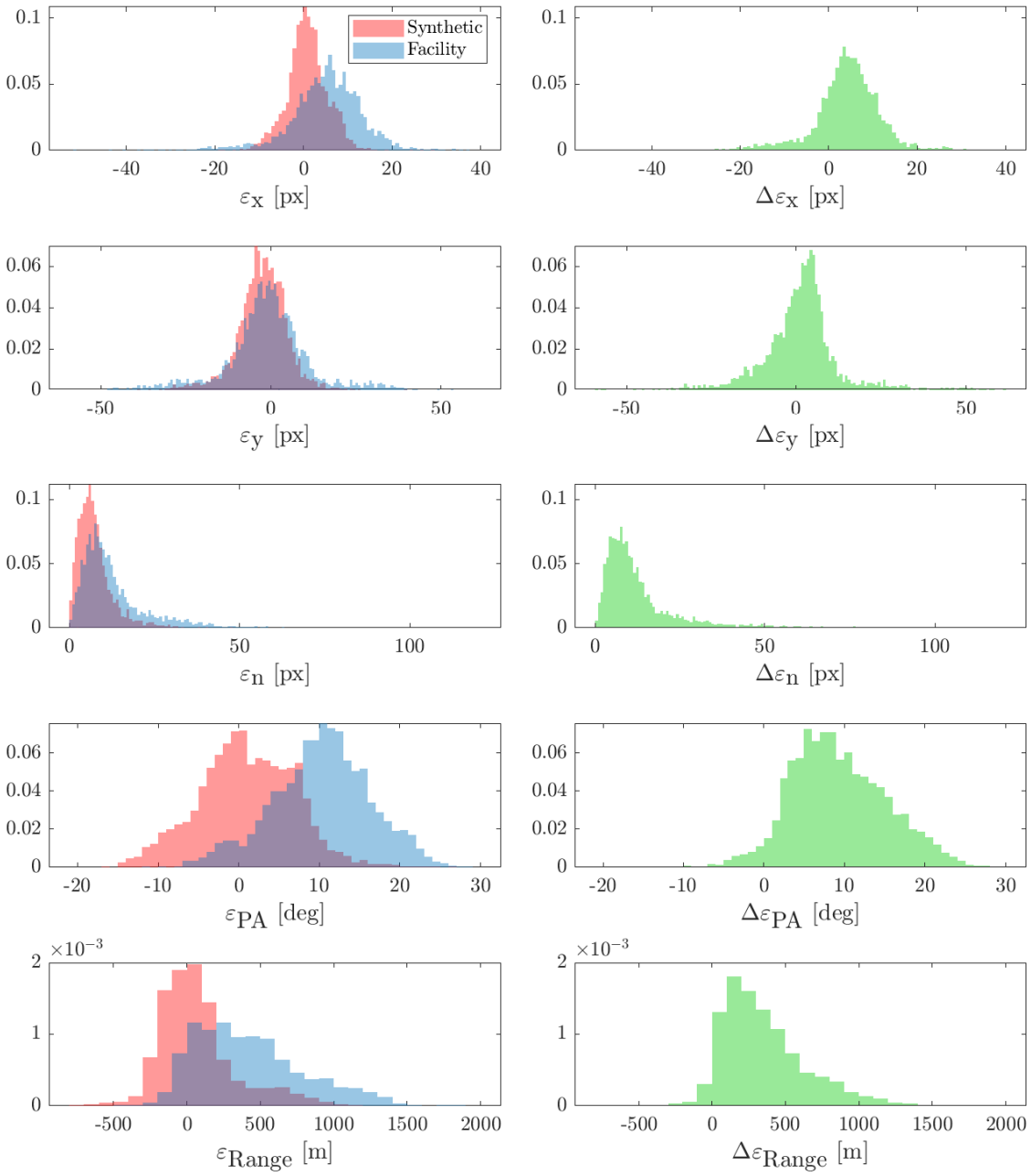


Figure 15: Results for the CRP dataset, shown as normalized histograms. Left: Estimation errors statistics. Right: Difference between the IP output on facility images and the IP output on synthetic images.

As the data-driven functions coefficients are estimated using a dataset of ideal mission images rendered from the simulated world, it is of fundamental importance to understand the behavior of the WCOB algorithm when errors are present in the image. As a consequence, in this section the Milani IP is run on images on the CRP and FRP taken in TinyV3RSE to account for camera and calibration errors. A comparison of ideal mission images and facility images are shown in Figures 12a and 12b

for the FRP and in Figures 13a and 13b for the CRP. A more detailed analysis of the V&V study of the WCOB can be found in Reference 29 where the algorithm is challenged with different levels of blur, exposure times, asteroid shapes, and observation geometries.
To analyze the results, the WCOB algorithm is run on the ideal mission image and on the facility image and the IP estimates are compared with their true values. The error in the CoF estimate is the most interesting quantity, and it is analyzed by considering both the error in the x and y components of the image and the error norm, computed as:

$$\varepsilon_{\mathrm{n}} = \sqrt{\varepsilon_{\mathrm{x}}^2 + \varepsilon_{\mathrm{y}}^2} \tag{5}$$

where ε_{x} and ε_{y} are the errors in the x and y components in the image plane. The errors on the phase angle and range are indicated respectively as $\varepsilon_{\mathrm{PA}}$ and $\varepsilon_{\mathrm{Range}}$. The difference between the errors obtained from facility images and the ones obtained from synthetic images, instead, is indicated by placing a Δ before the corresponding variable (e.g. $\Delta\varepsilon_{\mathrm{x}}$ and $\Delta\varepsilon_{\mathrm{y}}$).
The results obtained with this procedure are outlined in Figures 14 and 15. The total error is mostly below 20 pixels which is the 1-σ measurement noise provided to the Milani navigation filter.[30] For the CRP case the CoF absolute errors increase, which is expected since the spacecraft is closer to the system and thus the primary appears bigger in the images. Moreover, for the same reason, the errors due to the HIL, i.e. $\Delta\varepsilon_{\mathrm{x}}$, $\Delta\varepsilon_{\mathrm{y}}$, and $\Delta\varepsilon_{\mathrm{n}}$, is higher in the CRP case due to the difference in pixel intensity associated with each pixel with respect to the ideal mission image (see Figures 12a, 12b, 13a and 13b). Furthermore, this difference in pixels intensity with respect to the ideal mission image causes the PA and the range to slightly be biased with respect to IP estimates obtained from the ideal mission image. Similarly, the x component of the CoF is slightly biased as the correction performs slightly worse. Recall that the only observable which is used in the Milani navigation filter is the CoF estimate whose HIL errors are consistent with the measurement covariance used in the GNC analysis.

Moon case

In this section another example of the use of TinyV3RSE within the DART Lab's project is presented. Preliminary results of the V&V of the full-disk navigation method around the Moon are reported. This work has been previously presented in details in Reference 19 and interested readers can consult that work for further details. In the simulated world set-up, a spacecraft is observing the same face of the Moon at varying distances from 5000 km to 100000 km, every 500 km, and images are rendered to be then projected on the screen. The VNB algorithm works as follows. First, the raw image is analyzed to extract, after denoising, the light direction projected on the image and some points laying on the Moon limb. Second the best ellipse fitting the points is estimated from these points. Finally, the spacecraft position relative to the Moon is computed by knowing the Moon shape and its projection in the image. The steps of full-disk navigation method are illustrated in Figure 16. Moreover, in the same figure it is shown the ideal camera mission image of the Moon as well as the facility image of the same target. It is also possible to see an example of horizon points fiting and the ellipse fit. Interested readers can refer to Reference 31 for further details.
Given that the previously-explained algorithm has been designed with ideal mission images, it is important to address the effects of hardware components and other errors on the whole VBN chain. Thus, TinyV3RSE has been used to test and assess hardware errors on the VBN algortihm. Images of the Moon have been rendered and displayed on the screen to stimulate the camera. Facility images have then been used to test the full-disk navigation method.

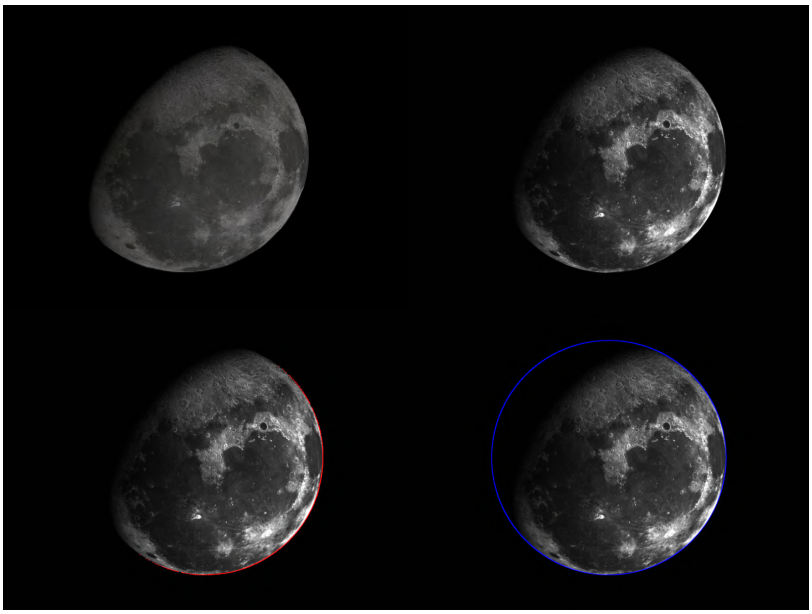


Figure 16: From left to right, top to bottom: original image of the moon projected to screen, image captured by the camera in the TinyV3RSE facility, identification of the lit horizon (red), identification of the ellipse fit (blue).

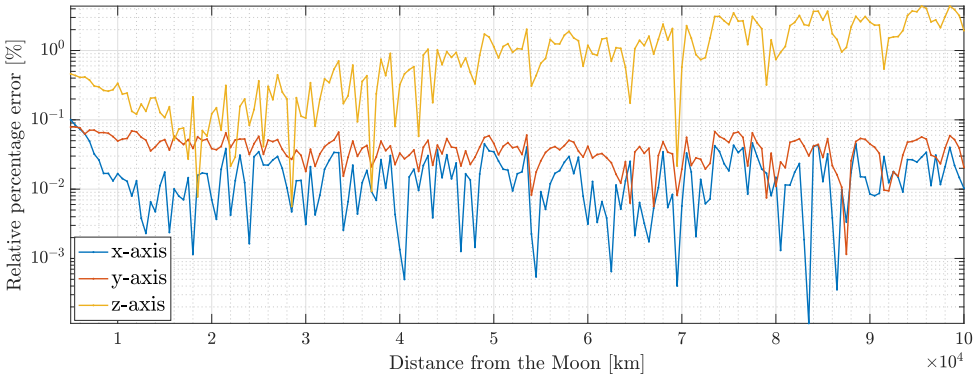


Figure 17: Relative positioning error per axis of the VBN algorithm used in the moon case at varying distances.

The error on the position estimate obtained with this procedure is illustrated in Figure 17, decomposed in the 3 axes of the camera frame. The x and y are the ones co-planar with the image plane, while the z-axis represents the bore-sight direction. The error is computed as relative percentage error, as the ratio between the absolute positioning error and the true range from the Moon, expressed as a percentage error. By inspection of Figure 17, it is possible to see that that error remains small in the x and y axes, never exceeding 0.1%, while it grows in the radial direction (z), as expected in optical navigation. This gradually increases from roughly 18000 km onward, reaching a maximum value of about 4% roughly at 100000 km.
This effect is caused by the natural behavior of optical navigation and uncertainties in calibration procedures. In general, it is natural for optical navigation algorithms to be less precise in the depth

axis with respect to the other ones because the optical information does not lie on the radial direction. To compensate for this, an altimeter could be used to provide more precise estimates on the z-axis. However, analysis of the VBN algorithm applied to the synthetic images has shown this effect should be smaller than the one recorded on real images. The second contribution is due to the calibration. The convergence of the calibration algorithm to the principal point of the camera can not be exact. Thus, some errors in the calibration procedure are affecting the intrinsic camera matrix and thus the solution provided by the VBN algorithm.

CONCLUSIONS AND FUTURE WORKS

In this work we have presented for the first time the design of the DART's vision-based navigation test bench, TinyV3RSE, a hardware-in-the-loop test bench used by the DART group for the development, validation, and testing of IP and VBN algorithms. The design and calibration procedures adopted within TinyV3RSE have been briefly illustrated as well its exploitation for IP and VBN algorithms applied to small bodies and in the lunar environment. The results presented here are preliminary, future works are planned to further extend the analysis performed with TinyV3RSE as well as to further refine the design of the optomechanical components (especially relative to the screen-support mounting) and overall calibration procedure. The facility will be exploited to assess the VBN and IP performances of on-going space missions[32–34] and projects of the DART's group.

ACKNOWLEDGMENTS

The authors would like to acknowledge the funding within EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864697). M.P. and F.T. would like to acknowledge the funding received from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie (Grant agreement No. 813644). The authors would also like to acknowledge Nicolò Vattai for the preliminary design of the test bench and Felice Piccolo for the Milani IP study.

REFERENCES

[1] M. B. Quadrelli, L. J. Wood, J. E. Riedel, M. C. McHenry, M. Aung, L. A. Cangahuala, R. A. Volpe, P. M. Beauchamp, and J. A. Cutts, "Guidance, Navigation, and Control Technology Assessment for Future Planetary Science Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 38, July 2015, pp. 1165–1186, 10.2514/1.g000525.

[2] I. Belgacem, G. Jonniaux, and F. Schmidt, "Image processing for precise geometry determination," *Planetary and Space Science*, Vol. 193, 2020, p. 105081, 10.1016/j.pss.2020.105081.

[3] N. Rowell, S. Parkes, M. Dunstan, and O. Dubois-Matra, "PANGU: Virtual spacecraft image generation," *5th Int. Conf. on Astrodynamics Tools and Techniques, ICATT*, 2012.

[4] J. Lebreton, R. Brochard, M. Baudry, G. Jonniaux, A. H. Salah, K. Kanani, M. Le Goff, A. Masson, N. Ollagnier, A. Panicucci, Paolo ad Proag, and C. Robin, "Image Simulation for Space Applications with the SurRender Software," *Proceedings of the 11th International ESA Conference on Guidance, Navigation & Control Systems, 22 - 25 June 2021, Virtual*, 2021.

[5] G. Jonniaux and D. Gherardi, "Development, tests and results of onboard image processing for JUICE," *Proceedings of the 11th International ESA Conference on Guidance, Navigation & Control Systems, 22 - 25 June 2021, Virtual*, 2021.

[6] G. Rufino and A. Moccia, "Laboratory test system for performance evaluation of advanced star sensors," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 200–208, 10.2514/2.4888.

[7] M. A. Samaan, S. R. Steffes, and S. Theil, "Star tracker real-time hardware in the loop testing using optical star simulator," *Spaceflight Mechanics*, Vol. 140, 2011.

[8] B. Boone, J. Bruzzi, W. Dellinger, B. Kluga, and K. Strobehn, "Optical simulator and testbed for spacecraft star tracker development," *Optical Modeling and Performance Predictions II*, Vol. 5867, International Society for Optics and Photonics, 2005, p. 586711, 10.1117/12.619133.

[9] V. Nardino, D. Guzzi, M. Burresi, M. Cecchi, T. Cecchi, F. Corti, M. Corti, E. Franci, G. Guidotti, I. Pippi, *et al.*, "MINISTAR: a miniaturized device for the test of star trackers," *International Conference on Space Optics—ICSO 2018*, Vol. 11180, International Society for Optics and Photonics, 2019, p. 111807V.

[10] G. Rufino, D. Accardo, M. Grassi, G. Fasano, A. Renga, and U. Tancredi, "Real-time hardware-in-the-loop tests of star tracker algorithms," *International Journal of Aerospace Engineering*, Vol. 2013, 2013, pp. 1–13, 10.1155/2013/505720.

[11] N. Filipe, L. Jones-Wilson, S. Mohan, K. Lo, and W. Jones-Wilson, "Miniaturized star tracker stimulator for closed-loop testing of cubesats," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3239–3246, 10.2514/1.g002794.

[12] R. Volpe, M. Sabatini, G. Palmerini, and D. Mora, "Testing and Validation of an Image-Based, Pose and Shape Reconstruction Algorithm for Didymos Mission," *Aerotecnica Missili & Spazio*, 2020, pp. 17–32, 10.1007/s42496-020-00034-6.

[13] A. Petit, E. Marchand, and K. Kanani, "Vision-based space autonomous rendezvous: A case study," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 619–624.

[14] M. Zwick, I. Huertas, L. Gerdes, and G. Ortega, "ORGL–ESA'S test facility for approach and contact operations in orbital and planetary environments," 2018.

[15] J. Brannan, N. Scott, and C. Carignan, "Robot Servicer Interaction with a Satellite During Capture," *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, 2018.

[16] C. Beierle and S. D'Amico, "Variable-magnification optical stimulator for training and validation of spaceborne vision-based navigation," *Journal of Spacecraft and Rockets*, Vol. 56, No. 4, 2019, pp. 1060–1072, 10.2514/1.a34337.

[17] A. Pellacani, M. Graziano, M. Fittock, J. Gil, and I. Carnelli, "HERA vision based GNC and autonomy," 2019, 10.13009/EUCASS2019-39.

[18] G. N. Holt, C. N. D'Souza, and D. W. Saley, "Orion optical navigation progress toward exploration mission 1," *2018 Space Flight Mechanics Meeting*, 2018, p. 1978.

[19] N. Vattai, "Development and validation of a horizon-based optical navigation test facility," Master's thesis, Politecnico di Milano, 12 2019. Accessible from: `https://www.politesi.polimi.it/handle/10589/151672`.

[20] A. M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, 2001.

[21] Z. Tang, R. G. v. Gioi, P. Monasse, and J.-M. Morel, "A precision analysis of camera distortion models," *IEEE Transactions on Image Processing*, Vol. 26, No. 6, 2017, pp. 2694–2704.

[22] J. Heikkila and O. Silvén, "A four-step camera calibration procedure with implicit image correction," *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, IEEE, 1997, pp. 1106–1112.

[23] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 22, No. 11, 2000, pp. 1330–1334, 10.1109/34.888718.

[24] K. Rossmann, "Point spread-function, line spread-function, and modulation transfer function: tools for the study of imaging systems," *Radiology*, Vol. 93, No. 2, 1969, pp. 257–272.

[25] F. Viallefont-Robinet, D. Helder, R. Fraisse, A. Newbury, F. v. d. Bergh, D. Lee, and S. Saunier, "Comparison of MTF measurements using edge method: towards reference data set," *Optics express*, Vol. 26, No. 26, 2018, pp. 33625–33648.

[26] P. Michel, M. Küppers, and I. Carnelli, "The Hera mission: European component of the ESA-NASA AIDA mission to a binary asteroid," *42nd COSPAR Scientific Assembly*, Vol. 42, 2018, pp. B1–1.

[27] M. Pugliatti, V. Franzese, A. Rizza, F. Piccolo, C. Bottiglieri, C. Giordano, F. Ferrari, and F. Topputo, "Design of the on-board image processing of the Milani mission," *AAS GN&C conference, 3 - 9 February 2022*, 2022.

[28] M. Pugliatti, V. Franzese, and F. Topputo, "Data-Driven Image Processing for Onboard Optical Navigation Around a Binary Asteroid," *Journal of Spacecraft and Rockets*, Jan. 2022. (Article in Advance), 10.2514/1.A35213.

[29] F. Piccolo, M. Pugliatti, P. Panicucci, and F. Topputo, "Toward verification and validation of the Milani Image Processing pipeline in the hardware-in-the-loop testbench TinyV3RSE," *AAS GN&C conference, 3 - 9 February 2022*, 2022.

[30] M. Pugliatti, A. Rizza, F. Piccolo, V. Franzese, C. Bottiglieri, C. Giordano, F. Ferrari, and F. Topputo, "The Milani mission: overview and architecture of the optical-based GNC system," *AIAA Scitech 2022 Forum*, 2022, p. 2381.

[31] V. Franzese, P. Di Lizia, and F. Topputo, "Autonomous optical navigation for lumio mission," *2018 Space Flight Mechanics Meeting*, 2018, p. 1977.

[32] F. Ferrari, V. Franzese, M. Pugliatti, C. Giordano, and F. Topputo, "Preliminary mission profile of Hera's Milani CubeSat," *Advances in Space Research*, Vol. 67, No. 6, 2021, pp. 2010–2029, 10.1016/j.asr.2020.12.034.

[33] F. Topputo, Y. Wang, C. Giordano, V. Franzese, H. Goldberg, F. Perez-Lissi, and R. Walker, "Envelop of reachable asteroids by M-ARGO CubeSat," *Advances in Space Research*, 2021, pp. 4193–4221, 10.1016/j.asr.2021.02.031.

[34] V. Franzese, P. Di Lizia, and F. Topputo, "Autonomous Optical Navigation for the Lunar Meteoroid Impacts Observer," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 7, 2019, pp. 1579–1586, 10.2514/1.g003999.