Acta Astronautica 204 (2023) 358–369

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

Dataset generation and validation for spacecraft pose estimation via monocular images processing

Michele Bechini *, Michèle Lavagna, Paolo Lunghi

Politecnico di Milano, Dept. of Aerospace Science & Technology, via La Masa 34, 20156 Milano, Italy

ARTICLE INFO

Keywords:
Image generation
Synthetic image datasets
Relative pose
Vision-based navigation

ABSTRACT

Recent studies demonstrate the possibility of navigating in proximity of uncooperative space resident objects by using only monocular images. Despite the results achieved, the development and testing of new algorithms are strongly constrained by the availability of spaceborne image datasets. To overcome this, a new algorithm embedded in a tool to generate synthetic high-fidelity spaceborne image datasets is presented here. The architecture developed can be tailored to a wide range of scenarios and it is based on an open-source ray-tracing software. All assumptions and simplifications adopted are discussed in detail for the different models considered, including a trade-off between accuracy and rendering time. The new method described is subsequently adapted to a baseline scenario where the optical properties of a reference spacecraft model are tuned. Both qualitative and quantitative validations are detailed and successfully carried out for the baseline case, demonstrating the high photo-realism achievable with the proposed method. As a consequence of this main outcome, the paper details the generation of labeled spaceborne image datasets publicly available and reports the analyses that confirm the high level of representativeness, making them suitable for training and testing image-based navigation algorithms. As another outcome, the most comprehensive multi-purpose labeled dataset of validated spaceborne synthetic images currently publicly available is presented.

1. Introduction

New classes of missions that envision a prominent role for autonomous close proximity operations, with a particular interest in cooperative and non-cooperative artificial objects, like formation flying missions (FF) with fractionated scientific payloads, on-orbit servicing demonstrators (OOS), and active debris removal, gained increasing attention in the last few years, aiming to perform regular in-orbit services [1]. Significant technology development is still needed to make these missions feasible. The proximity operations and maneuvering impose a high level of reactivity of the chaser, leading to the need for guidance, navigation, and control (GNC) chains solved autonomously onboard to ensure timeliness, reactivity, effectiveness, and robustness in both nominal and off-nominal operations. The first ring of that chain is the autonomous relative navigation, hence the estimation of the relative pose (position and attitude) of the chaser with respect to the target. Dealing with an uncooperative target represents the most challenging scenario for the relative pose estimation task since no information is shared between the two spacecraft (except for the a priori knowledge of geometry in the case of a known target), reducing the possibilities to only those that exploit only the capabilities of the chaser. Among others, sensor suites that include monocular cameras are widely considered as one of the most attractive solutions to acquire measurements for the onboard GNC chain [2,3]. In fact, solutions that include cameras operating in the visible spectrum have been extensively studied [4] and applied in the context of both cooperative [5] and uncooperative [6] rendezvous.

"Classical" solutions to deal with the pose estimation via monocular images of known uncooperative targets include the extraction of handcrafted features of the target (corner [7], edges [8], etc.) via dedicated image processing algorithms [9]. The relative pose of the chaser camera with respect to the target is hence retrieved by solving the Perspective-n-Points (PnP) problem between the 2D features extracted from the image and the features of the 3D CAD model of the target available onboard. Among the algorithms proposed, the Sharma–Ventura–D'Amico (SVD) [10] achieved state-of-the-art (SOTA) performance by exploiting the edges of the target extracted via Hough Transform [11]. More recently, Artificial Neural Networks (ANNs) revealed to be a valuable tool to process acquired images in many different contexts [12,13]. Hence, the wide range of "modern" approaches for relative pose estimation involving the usage of ANNs, mostly Deep Convolutional Neural Networks (CNNs), can be classified into two branches: direct CNN regression [14,15], and PnP solvers aided by CNN [16,17]. The former

* Corresponding author.
E-mail addresses: michele.bechini@polimi.it (M. Bechini), michelle.lavagna@polimi.it (M. Lavagna), paolo.lunghi@polimi.it (P. Lunghi).

https://doi.org/10.1016/j.actaastro.2023.01.012
Received 2 December 2022; Received in revised form 1 January 2023; Accepted 10 January 2023
Available online 11 January 2023

approach aims to recover the relative pose via directly convolving the input image with a CNN (usually split into two branches) that outputs the relative position and the relative quaternion. The latter approach includes a first step in which the CNN takes the image as an input and outputs the 2D features that subsequently are fed to the PnP solver.

The results of the 2019 ESA's Kelvins Pose Estimation Challenge [18] demonstrate that the most effective approach to deal with relative pose estimation via monocular images is to use CNNs-aided PnP solvers. The robustness and effectiveness of this approach rely on the parameters learned by the CNN during the training phase hence, labeled image datasets are needed to correctly train CNNs. Despite that, currently, there is a lack of publicly available spaceborne image datasets, strongly limiting the applicability of CNNs-aided PnP solvers to a small range of possible scenarios, acting also as a bottleneck for the development of new algorithms. A spaceborne image dataset generation tool is here proposed to overcome this limitation. The method adopted allows the generation of synthetic noiseless images that have been validated successfully by taking the Spacecraft Pose Estimation Dataset (SPEED) Dataset [19] as reference. The tool developed has been used to generate fully-synthetic spaceborne noiseless image datasets using a simplified 3D CAD model of Tango from PRISMA mission [20] reconstructed from a previous work [10], since the official 3D CAD model is not publicly available. The datasets generated are made publicly available and comprise a Pose Estimation Dataset [21], a dataset tailored to bounding box extraction and semantic segmentation [22], and a dataset labeled with Tango edges in view at the current relative pose [23].

The main contributions of the proposed work are threefold. First, a pipeline to generate spaceborne synthetic noiseless image datasets via physically-based ray tracing is defined. It can be applied to a wide range of scenarios including different spacecraft models, space resident objects as background (airless or with an atmosphere layer), and multiple camera models. Then, a qualitative and quantitative validation scheme is defined and applied successfully to the images generated by the proposed tool using Tango as spacecraft and the SPEED dataset as reference. Lastly, spaceborne synthetic image datasets have been generated and made publicly available. Note that the datasets generated can be combined among them (through the different labels associated with the same images common to all the datasets) to constitute a multi-purpose dataset that can be applied to a wide range of tasks exploitable via training of CNNs. Notice that the image noise level can be dependent on the application scenario (e.g. for spacecraft orbiting airless bodies, the electric field generated by the interaction between the plasma and the body can be not negligible [24] and this can increase the electronic noise in the images). Thus, to increase the flexibility of the tool, the images generated by the tool are noiseless. Notice that the noises that usually affect images can be easily added to the synthetic images in post-processing.

The remainder of the paper is organized as follows: Section 2 reports an overview of the available software solutions for image generation via ray-tracing and the already publicly available datasets. The tool proposed is then described in detail in Section 3, while the validation of the images generated is discussed in Section 4. The analysis of the datasets generated and an overview of their labels are provided in Section 5. Conclusions resuming the main outcomes of this work and hints for possible future developments are finally reported in Section 6.

2. Available rendering software and image datasets

Although of a strong interest in spaceborne images for training and testing algorithms for relative GNC, the availability of real spacecraft images captured during space missions and properly labeled is limited. Hence, the need to generate realistic images synthetically arises. Generating realistic synthetic images is a well-studied task that usually is achieved via ray-tracing. Ray-tracing is a rendering technique that relies on the evaluation and simulation of the paths of view lines that start from the observer camera and interact with generic virtual objects in the scene. Coupling this with the simulated light rays from the light sources to the virtual object enables the evaluation of the shadows and the calculation of the color intensity of the associated pixels. Physically based ray-tracers, which simulate the physics of the light, allow the generation of synthetic images with a high degree of accuracy [25], and also to fully control the scene to be rendered. Currently, there are several tools, mostly based on OpenGL, ray-tracing, or path-tracing, that offer the possibility to create 3D scenes to be rendered with various degrees of photo-realism. Among them, the available commercial solutions comprise PANGU (Planet and Asteroid Natural Scene Generation Utility) [26], and Airbus SurRender [27]. The former was developed to create synthetic planetary surface images, then the possibility of generating also images of artificial objects was implemented in the software. The latter can handle various space objects such as planets, asteroids, satellites, and spacecraft by providing both a fast real-time and a more accurate but slower rendering scheme. Some well-known simulators widely adopted for video games and animations, like Blender [28] and Unreal Engine [29], have been used to render space-related scenes but, despite the visually impressive images generated, these simulators lack the realism needed, being devoted to cope with human vision [27]. Among the open-source software, POV-Ray (Persistence of Vision Raytracer) [30] and PBRT (Physically Based Ray-Tracing) [31] are the most promising, offering the possibility of directly accessing the code and eventually patching it to tailor the software to the user's needs.

Concerning the spaceborne synthetic image datasets, the only ones publicly available are URSO (Unreal Rendered Spacecraft On-Orbit) [14], SPEED (Spacecraft Pose Estimation Dataset) [19], and SPEED+ [32] (improved version of SPEED). URSO is a synthetic dataset of 5000 RGB images of both Soyuz and Dragon spacecraft models with Earth as background, rendered with a simulator based on Unreal Engine 4 with a resolution of 1080 × 960 pixels. Game engines have been criticized for the lack of photometric accuracy of the camera sensors [27], but the author of URSO claims that efforts have been put to implement physically-based shading models and cameras in such engines (e.g. by using path-tracing) and that the custom simulator used to synthesize URSO obtains photorealistic images. Despite that, no validation of the images in URSO is provided. The SPEED dataset and its improved version, the SPEED+, are used for the ESA's Kelvins Pose Estimation Challenges, with SPEED being the first publicly available dataset of spaceborne labeled images ever released. SPEED comprises 15,000 synthetic grayscale images of Tango obtained using the Optical Simulator [33], based on an OpenGL rendering pipeline, and 300 grayscale images obtained using a mock-up. The SPEED+ contains 60,000 synthetic grayscale images of Tango generated like in SPEED, 6740 grayscale mock-up images obtained by introducing hardware-in-the-loop to simulate the Earth albedo, and 2791 grayscale mock-up images with sun-lamps used to simulate effects like flares and saturation [34]. All the images have a resolution of 1920 × 1200 pixels and are validated against real images of Tango from the PRISMA mission by employing histogram comparison [18]. Further details regarding SPEED and SPEED+ are available in the related articles [18,34] for the interested reader.

3. Image dataset generation via ray-tracing

The possibility of using PBRT as a ray-tracer, the core of the image generation algorithm implemented, has been evaluated. The main advantage offered by PBRT is the possibility of directly simulating the entire lens set of real cameras [31] in a physically-based way, making it possible to model more realistically the image acquisition device and the sensor characteristics. Despite that, the main issue identified in PBRT is the simulation of the Earth's atmosphere and the scattering properties. The density of the atmosphere layer can be defined in PBRT only on a regular grid that is then interpolated to retrieve the scattering effects on the light rays. This leads to artifacts in the image,

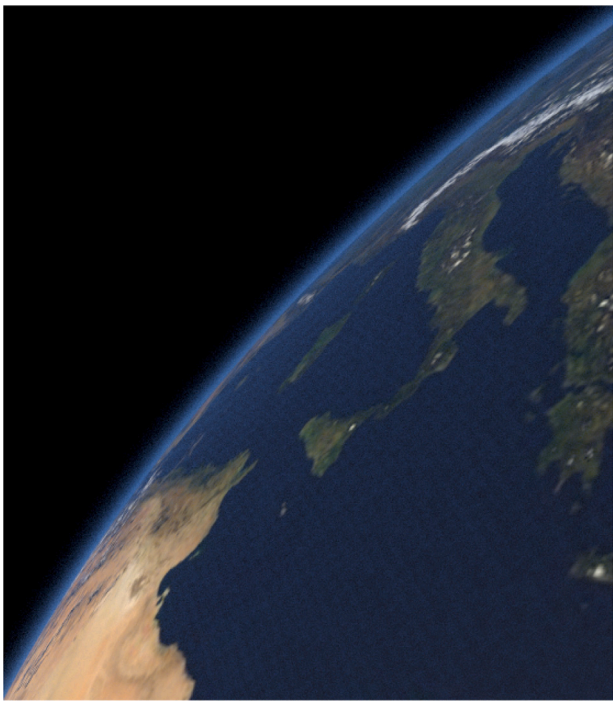


Fig. 1. Example of artifacts in atmosphere layer rendered with PBRT.

as shown in Fig. 1 (notice the brighter regular dots, more evident over the seas), also for a highly refined mesh with a grid resolution of 1 km. It is acknowledged that this issue could be overcome by patching the source code, but here it has been preferred to use a ray-tracing software that more easily handles the atmosphere scattering evaluation, due to its crucial role in achieving realistic and representative images. Due to this issue, and although PBRT offers in its latest release the possibility of performing ray-tracing via GPU acceleration, here it has been decided to adopt POV-Ray as ray-tracer, despite a few well-known limitations [35] (no GPU acceleration, long parsing time, etc.). POV-Ray has been already successfully employed to render via ray-tracing both spaceborne natural landscapes, like the Moon's craters [35,36], and spacecraft [37,38]. The POV-Ray release used for this tool is a patched version of POV-Ray 3.7 made available in [38] that restores the possibility to exclude the computation of the bounding box around each object, avoiding errors in the rendering of scenes with space-scale distances (notice that the unit of measure used here is the km).

The remainder of this section is devoted to the modeling assumptions used to render spaceborne synthetic images, with a particular focus on Earth modeling, including the atmosphere simulation, in Section 3.1, and on the spacecraft model adopted and the pipeline to feed the mesh to POV-Ray in Section 3.2. It is worth pointing out that the assumptions introduced in the following sections can be easily generalized to several celestial bodies and target spacecraft by simply changing a few hyperparameters or the "include file" for the 3D model of the target spacecraft but keeping the core of the tool unchanged. Hence, the image generation pipeline is well suited to simulate several scenarios of interest as Mars and its atmosphere as background, airless celestial bodies, and multiple target spacecraft with different geometries.

3.1. Earth and atmosphere modeling

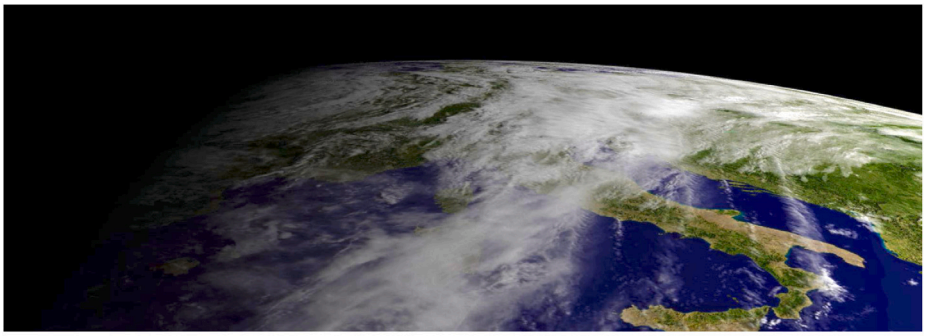
The first step in modeling a celestial body is to retrieve its shape. Concerning objects like the planets of the Solar System and most of their moons, they could be represented in first approximation as spheres. It is possible to recover the correct ellipsoidal shape of a planet by modeling it as a sphere and then applying a scale factor, to improve the realism of the images. The scale factor used to simulate the Earth's oblateness is equal to 0.9967 applied along the z-axis of the Earth-centered inertial (ECI) reference frame. For more complex bodies such as asteroids, the ellipsoidal approximations could be excessively rough hence, to properly handle the complex geometries, it is here preferred to import into the scene a dedicated external file, named "include file", that contains the refined mesh of the object. Please refer to Section 3.2 for a step-by-step discussion on the generation of mesh files compatible with POV-Ray. The next step toward photo-realistic synthetic images is the definition of textures. For airless bodies with regular shape, like the Moon, the only step needed is to apply a realistic surface texture by using in POV-Ray the option "*map_type 1*" that directly attaches the 2D texture over a 3D sphere, with the relative latitude-dependent shrinking. This process could apply even to the more complex case of celestial bodies with atmosphere and clouds since texture maps that include both the surface and the clouds map in a single image are available such as for the Earth case. This approach leads to good-quality images, but this approach is still not well suited to build datasets for CNN training. The following main issues have been observed: the clouds are stuck over the same regions of the Earth for all images, the optical properties of clouds, seas, and terrains are all the same, and the diffusion of sunlight through the atmosphere is not considered.

In the image generation pipeline adopted here, a clouds-only texture is used here and applied over a shell defined to have a random thickness between 625 m and 1250 m and an altitude from 650 m to 3150 m at most. Since the cloud map is a grayscale image, with black associated with regions where there are no clouds and white where there are thick and dense clouds, it is possible to tune the channel related to the transparency of the texture in POV-Ray such that it is mapped following the density of the cloud layer. This makes it possible to have the Earth's surface still visible under low-density clouds (see Fig. 2 where the semi-transparent clouds are evident). Notice that the cloud texture is detached and independent from the Earth, allowing the user to independently tune their optical properties and orientation in the scene, overcoming the issue related to the fixed clouds and common optical properties. A random rotation angle with respect to the z-axis of the ECI frame is defined at each image generation step and applied to the cloud texture to make the clouds rotate with respect to the ground. Concerning the Earth's surface, a differentiation of the optical parameters of the lands with respect to the seas has been included in the Earth's model, similarly to [35,38]. A binary map of the Earth is adopted to split the waters from the lands. The splitting allows the definition of different optical properties (especially for the reflection of light over the seas) making the images more photorealistic.

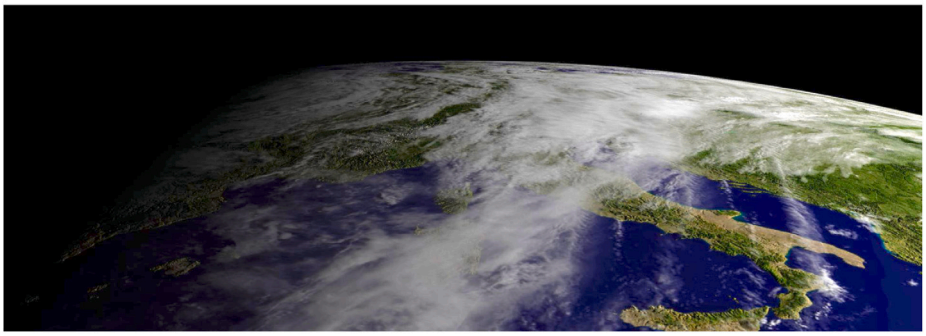
The possibility of including the simulation of the topography of the terrain has been evaluated to further improve the accuracy of the images. There are two options available to deal with topography in POV-Ray:

- Bump-map approximation: the irregularities of the surface are emulated by modifying the normal vector of each surface, hence by altering the light reflection. Notice that this approach generates irregularities (i.e. bumps) by modifying only the reflection of the light, while the geometry is not affected. Hence, the original shape is preserved.
- Isosurface function: the isosurface object mathematically describes the topography of the terrain in POV-Ray. In this case, the geometry of the object is modified according to a model defined by the user. Hence, the accuracy is high also in terms of shadows in the image.

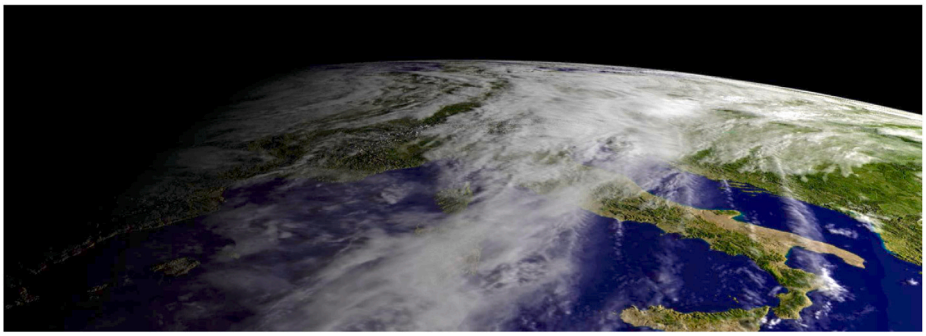
Both the bump-map approximation and the isosurface can be defined starting from a grayscale texture. The texture has the altitude mapped into the intensity of each pixel. The output map is here scaled to have a maximum altitude of about 9 km over the sea level, corresponding to the altitude of Mount Everest. A comparison between the same scene



(a) Image rendered without topography.



(b) Image rendered with bumpmap approximation.



(c) Image rendered with isosurface function.

Fig. 2. Comparison of images rendered with and without topography information.

rendered without topography (Fig. 2(a)), with bumpmap approximation (Fig. 2(b)), and with isosurface function (Fig. 2(c)) is reported in Fig. 2. All images in Fig. 2 are overexposed and rendered without considering the atmosphere to make more evident the differences in the topography. The reported images show that excluding the topography leads to unrealistic images and poor quality. The image obtained using the bumpmap approximation has a higher quality than the "flat" image, with mountains visible (due to the correct shadowing). The image achieved using the isosurface is almost identical to the one rendered by including the bumpmap approximation. The only exception is the region close to the terminator, where the shadows are better evaluated and defined in the case of the isosurface option.

From this, it is clear that it is mandatory to include topography information in the image generation pipeline to achieve highly photorealistic images. This result holds for any celestial body used as background for spaceborne images. Moreover, the improvement in photorealism is as high as the relative distance between the camera and the celestial body is reduced (e.g. for Low Earth Orbit missions). Despite the comparable results in terms of quality of the images rendered with topography taken into account during the rendering process, it should be noticed that using the bumpmap approximation gives a computational burden almost null compared to the case without topography. Instead, by using the isosurface, the rendering time is highly increased. Hence, due to the reduced computational time and the high quality of the output images, the bumpmap approximation is clearly selected as the default method in the image generation pipeline.

The last part considered in the background modeling is the atmosphere. The colors assumed by the atmosphere (from dark blue to red close to the terminator) are due to light scattering. The scattering depends on the spectrum of the light emitted by the Sun (here modeled as a white light), the density of the particles in the atmosphere, their kind, and on the length of the light path through the atmosphere. Concerning the interaction between the particles and the light, it is well known that the incident light over particles scatters in all directions by following given laws. The Rayleigh scattering well approximates the interaction between the sunlight and the atmospheric particles. Following this model, the quantity of scattered light depends on the angle of the incident light. The scattering is higher when the incident light is parallel or antiparallel to the direction of observation and smaller when the incident light is perpendicular to the direction of observation. To implement a high-fidelity model, it must be included that the Rayleigh scattering in the atmosphere is also proportional to $1/\lambda^4$, where λ is the light wavelength. Hence, the scattering is higher for short wavelengths (blue color) than for long wavelengths (red color).

(a) True image from Apollo 17 mission (credit NASA).

(b) Image rendered with simplified atmosphere model.

(c) Image rendered with high-fidelity atmosphere model.

Fig. 3. Comparison of images rendered with different atmospheric models against a true Earth picture.

The Rayleigh scattering model already available in POV-Ray includes the dependency from the direction of the incoming light, while it does not consider the wavelengths. As a consequence, two different models are considered here: the first is a simple but computationally efficient model based on the simplified Rayleigh scattering available in POV-Ray with scattering color hand-tuned to achieve photorealistic results, while the second is a high-fidelity model that uses a custom function to restore the dependency also from the wavelength. The high-fidelity model uses the exponential law for the variation of the atmospheric density of the atmosphere with the height: $\rho = \rho_0 \exp(-h/H)$, where $\rho_0 = 1.225\,\mathrm{kg/m^3}$ is the density at sea level for the International Standard Atmosphere (ISA) [39], h is the altitude above sea level of the point considered, and H is the scale height. The scale height is the height at which the density falls to a value of 1/e of the surface density [40]. The scale height can be defined as $H = kT/mg$, where k is the Boltzmann's constant, T is the atmospheric temperature, m is the mean molecular mass, and g is the gravitational acceleration [40,41]. The average composition of the atmosphere can be assumed to be almost constant up to about 90 km [39]. Thus, the scale height is affected only by the temperature as a function of the altitude above sea level. Therefore, by using the given equation for the computation of the scale height, it is possible to compute H at sea level for ISA ($T = 288\,\mathrm{K}$), resulting in $H = 8.4\,\mathrm{km}$. In the same way, the temperature at h = 90 km above sea level is about 190 K [40,41] hence, it is possible to get $H = 5.5\,\mathrm{km}$ at h = 90 km. The scale height value adopted in the high-fidelity model is the mean value between 5.5 km and 8.4 km. We acknowledge that more complex models for the variations of the scale height and the atmospheric density as a function of the altitude above sea level could be adopted, but the improvements in the quality of the rendered images will be small despite the computational overhead introduced. A comparison of images rendered with the Earth fully in view, with a simplified and accurate atmosphere model respectively against a true picture of the Earth from the Apollo 17 mission [42], is provided in Fig. 3. Notice that, although both the synthetic images in Figs 3(b)–3(c) cannot perfectly replicate the ground truth image in Fig. 3(a) due to a difference in image resolution, optical parameters of the actual camera not modeled, and different exposure or light conditions, the synthetic images have both a high degree of similarity with the real one. It should be also taken into account that the iconic real image used in Fig. 3 just for comparison was extensively post-processed on ground to optimize the color and contrast. As for the case of topography mapping, also for the atmosphere, the high-fidelity method has an extremely high computational burden compared to the simpler model. Despite that, the results are comparable, as it can be noticed from Fig. 3, for altitudes above 1000 km. This outcome also confirms that the simple exponential model adopted here for the atmosphere density is well suited, confirming that more complex atmospheric models are not needed for the range of scenarios considered. For lower altitudes, the results achieved by the low-fidelity model are poor, mainly in the transition region between the atmosphere and deep space. As a consequence, here it is applied the accurate model only for camera altitudes below 1000 km, while the simplified model is the default choice. It is worth remarking that the models discussed above are all included in the image generation algorithm, and they can be selected and combined by the user. By changing a few parameters in the atmospheric model (i.e. the density model and the wavelength dependency), it is possible to simulate different atmospheres from different celestial bodies. The same holds for the texture and the topography map applied to more complex bodies such as asteroids. Therefore, the tool can generate images for a wide range of natural space resident objects ranging from regularly shaped objects with an atmosphere layer (e.g. the Earth) to highly irregularly shaped airless bodies, such as asteroids. Fig. 4 reports an example of 162173 Ryugu rendered in POV-Ray with the proposed pipeline. Notice that, in contrast to the SPEED and SPEED+ datasets, where the background is not rendered with the spacecraft in the same scene but is added in post-processing, here the background is included in the scene and rendered together with the spacecraft, making it possible to directly evaluate in the images the effect of the albedo on the spacecraft itself, without using other light sources to simulate it.

Fig. 4. 162173 Ryugu rendered in POV-Ray.

3.2. Spacecraft modeling

The spacecraft (or any other object) can be included in the rendered scene by modeling it directly in POV-Ray with primitive shapes. This method is simple but not efficient, especially for spacecraft with complex geometries. The pipeline adopted here uses input files named "include files" compatible with POV-Ray that already contain all the information about the target, from its geometry to all the needed textures.

The input files can be exported in a POV-Ray compatible format directly from Blender, as already used in [38]. The input to the entire pipeline is a 3D CAD model representative of the target spacecraft. The 3D CAD model is exported in Standard Triangle Language (STL) and directly imported into Blender, where the optical properties and the textures can be easily defined and applied. Notice that, to assign to each surface different optical properties or textures, it is needed to import the STL generated by a 3D CAD assembly, hence composed of parts bonded in a final assembly, that are interpreted in Blender as children objects. An equivalent but more time-consuming process consists in splitting the parent object into children objects in Blender (by selecting the faces of the mesh belonging to each child) and then defining for each child the optical properties. After that, by using the POV-Ray add-on available for Blender, it is possible to both add custom code to assign textures and optical properties to the objects and to render the scene in Blender with POV-Ray as the rendering engine. Optionally, it is possible to export also the file used to render the scene in Scene Descriptor Language (SDL). The SDL exported contains indeed all the information about the textures and the custom code previously added, together with all the children objects, but treated as split entities. To ease working with the exported spacecraft model, the SDL file is hand-modified by merging all children objects in a single general parent object representative of the entire spacecraft, which is then converted into a POV-Ray include file. Note that all children's optical properties are preserved also once they are merged into a single parent object. The "include file" is then used as an external object to be added to the scene description file when needed at the desired location and attitude with respect to the POV-Ray reference frame. It is important to point out that the process previously detailed must be accomplished only once per spacecraft model when the target object is modeled for the first time. Once the "include file" is defined, it can be called and added to each scene in POV-Ray without any modification needed, for all positions and attitudes wanted. This procedure will be tailored to a simplified model of the Tango spacecraft from the PRISMA mission in Section 4, to validate the tool, but it can be generalized to all spacecraft models available, as it can be noticed from Fig. 5 where the tool has been successfully adopted to render an artificial space resident object. The optical properties of the surfaces of the simplified Tango model are tuned during the validation phase.

Fig. 5. Vega Secondary Payload Adapter rendered in POV-Ray.

4. Synthetic images validation

The SPEED dataset has been used as a reference to validate the images generated with the method previously described. The optical properties of the spacecraft surfaces have been tuned during the validation phase. The validation of the SPEED dataset with respect to actual images from the PRISMA mission has been achieved by comparing the pixel intensity histograms of synthetic images with respect to the real ones [18]. The validation of the synthetic rendered scenes is carried out here by considering only the case without background. This constraint for the validation is set because the information about the rotation of the Earth with respect to the z-axis of the ECI frame and the position of the light source is missing for the SPEED images annotations. Moreover, the clouds in the SPEED images are not reproduced by textures since they are taken from actual pictures of the Earth [18], making the exact reproduction of the SPEED images with background not possible, jeopardizing the validation procedure. Concerning the spacecraft model, the official CAD used for the SPEED dataset is not publicly available, hence a simplified 3D CAD model has been reconstructed from the dimensions of the main components available in [10]: the single body-mounted solar panel is a polygon of 560 × 750 mm, the main body is a polyhedron of 560 × 550 × 300 mm, and all appendages have a length of 204 mm. Concerning the camera parameters used for the SPEED dataset [18], they are reported in Table 1. These parameters are defined to tailor the perspective pinhole camera model used in POV-Ray to generate the images for validation. Notice that in POV-Ray, it is also needed to specify the aspect ratio, equal to $AR = 1920/1200 = 1.6$ for the SPEED images, and the angular field of view of the camera that is computed for a pinhole camera with squared pixels as:

Table 1
Parameters of the camera used for the SPEED dataset.

Parameter	Value
Number of horizontal pixels	1920 px
Number of vertical pixels	1200 px
Focal length	17.6 mm
Pixel length	5.86×10^{-3} mm

$$\alpha = 2 \arctan\left(\frac{CCD_{size}}{2f}\right) \tag{1}$$

where CCD_{size} is the size of the sensor and f is the focal length. For the SPEED camera $\alpha \approx 35.45°$. To properly recover the relative pose associated with each image of the SPEED dataset, the left-handed reference frame in POV-Ray is shifted to a right-handed reference frame by defining in the camera specifications a *right* vector (the vector used in POV-Ray to describe the direction to the right of the camera) equal to $[-AR, 0, 0]^T$ and the *sky* vector to $[0, 0, 1]^T$. In doing so, the POV-Ray global reference frame coincides with the ECI frame [38]. The last parameters to be specified to reproduce the images in the SPEED dataset are the location and pointing direction (boresight axis) for the camera, the position of the *arealight* used to approximate the Sun, and the location and attitude of the spacecraft. Having available only the relative position between the spacecraft and the camera per each SPEED image, the camera has been placed in the origin of the POV-Ray global reference frame, with pointing direction coincident with the z-axis in the ECI frame, the relative pose annotated has been then used to locate and rotate the Tango model properly in the scene. The position of the Sun has been hand-tuned to have shadows in the rendered image as close as possible to the SPEED image considered. Notice that to compare the SPEED images with those generated by the proposed tool, the noiseless images are processed to add the same noise level as the SPEED images before performing the validation. In agreement with [18], the images are blurred by applying a Gaussian blurring with $\sigma = 1$ and then an additive white Gaussian noise with variance $\sigma^2 = 0.0022$ is added to the blurred grayscale image.

Since the size of the spacecraft reported in [10] and used here for the simplified model is not perfectly the one of the Tango spacecraft (whose size is not publicly available, up to our knowledge) and since the 3D model is a simplified version that lacks the details of the official mock-up used for the SPEED dataset, the images are firstly preliminarily validated by performing the histogram comparison as in [18] by using the SPEED images as ground truth. Then, to have quantitative indexes,

the validation is carried out also by considering the shadow index and the feature quality index [43]. The shadow index J_s is computed as:

$$J_s = 1 - \frac{D_s}{S_{real}} \tag{2}$$

where D_s is the sum of non-zero pixels in the disparity map and S_{real} is the number of pixels classified as "in shadow" in the reference SPEED image. The disparity map is defined by subtracting the binary images obtained by applying the Otsu thresholding [44] to both the tested and the reference image in all validation pairs. The threshold level automatically identified to discriminate between shadowed and illuminated pixels in the SPEED image is used to define S_{real}. Therefore, the index J_s represents the number of pixels in shadow correctly reproduced in the rendered image. The feature quality index FQI is computed as:

$$FQI = 1 - \frac{\mu(H_d)}{H_{d,max}} \tag{3}$$

where H_d is the Hamming distance between two corresponding feature descriptors extracted and matched in both images and $H_{d,max}$ is the maximum possible Hamming distance. The mean value $\mu(H_d)$ is computed from 10 matched features extracted using ORB [7]. This index is adopted since several algorithms rely on feature extraction. Thus, comparing the features extracted in the reference image with those coming from the synthetic rendered image offers an evaluation of the similarity of the images at the features level. Both J_s and FQI are such that the closer to 1, the higher the similarities between images. In agreement with [43], the validation is successful if $J_s > 0.90$ and $FQI > 0.80$.

The validation is performed by replicating 20 random images from the SPEED dataset with the proposed method. Table 2 reports the coefficients of the optical parameters used for the spacecraft model and tuned during the validation steps. Fig. 6 gives an example of the comparison of the histograms computed for a couple of images during the validation phase, where Fig. 6(a) shows the reference SPEED image, Fig. 6(b) shows the noised image reproduced with the presented tool, and Fig. 6(c) shows the two histograms compared. By comparing Fig. 6(a) and Fig. 6(b), it can be noticed that, visually, the two images are close, even if the dimensions given in [18] and used here only approximate the actual shape of the Tango spacecraft, and despite that the simplified CAD model used in Fig. 6(b) does not include all the details included in the official CAD, like the support structures for the antennas.

Table 2
Optical parameters scaling coefficients of the spacecraft main components.

Parameter	Solar panel	Antennas	Main body
Ambient	0.0001	0.2	0.2
Roughness	0.15	0.0005	0.0005
Brillance	0	3.15	3.15
Diffuse	0.3	0.95	0.99
Reflection	0.25 – 0.5	0.65 – 1.0	0.65 – 1.0
Specular	0.005	0.95	0.95
Phong	0	0.43	0.43
Phong size	0	0.25	0.25

The histograms in Fig. 6(c) show that, even if they are not perfectly matching, the trends are close. The differences between the two histograms are given mainly by a bias in the black region that is due to a slightly different illumination condition between the two images and a lack of details that are darker in the original SPEED image, explaining why this bias has been obtained in all the images. Notice that the light-source position has been hand-tuned by taking the SPEED images as references. Table 3 contains the J_s and FQI indexes for the 20 random images compared during the validation process. The results show that with an average value of $J_s = 0.98$, the rendered images have a shadow distribution extremely close to the reference SPEED images. The same holds for the quality of the features extracted, with an average score of $FQI = 0.83$. Notice that scoring a high FQI means that the ORB descriptors extracted in the two images are similar

Table 3
J_s and FQI indexes evaluated for the tested images.

SPEED image number	$J_s > 0.90$	$FQI > 0.80$
img000012	0.93	0.83
img000043	0.99	0.83
img000059	0.99	0.85
img000114	0.99	0.81
img000138	0.99	0.84
img000324	0.97	0.86
img000388	0.99	0.84
img000468	0.96	0.81
img000486	0.97	0.81
img002122	0.99	0.81
img002156	0.99	0.87
img002350	0.99	0.85
img003271	0.99	0.82
img003849	0.99	0.83
img004386	0.99	0.86
img005964	0.98	0.81
img006967	0.99	0.82
img007408	0.99	0.83
img007461	0.99	0.83
img007485	0.92	0.84
Average value	**0.98**	**0.83**

hence, both the illumination and the material of the original image are reproduced correctly in the rendered image. From the results obtained, since both indexes satisfy the requirements $J_s > 0.90$ and $FQI > 0.80$ for all random images taken into account, the validation campaign is considered concluded and performed successfully. The successful validation demonstrates that with the proposed image generation pipeline, it is possible to generate photo-realistic images, paving the way to the generation of validated synthetic image datasets with the simplified Tango model as the target.

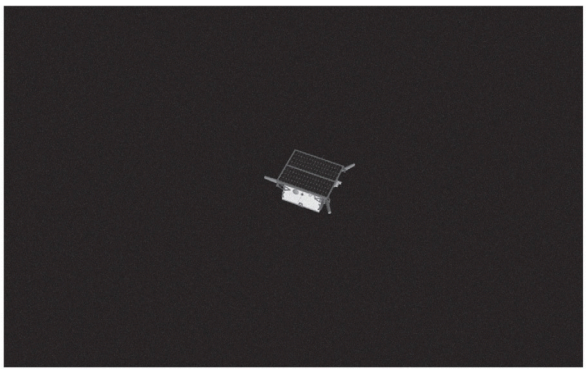
5. Multi-purpose labeled spacecraft dataset

The image generation pipeline described and validated above has been used as the main core in an algorithm to produce sets of images of the simplified and validated Tango model with the Earth in the background. The parameters of the camera model adopted for the dataset generation are reported in Table 4. The camera is modeled from the parameters of the Chameleon 3 by FLIR, with a reduced array size. The aspect ratio is equal to 1 and, from Eq. (1), the angular field of view results to be $\alpha \approx 45.55°$. The dataset produced contains about 30,000 synthetic noiseless images for training and 3000 for testing, partitioned with a 10:1 ratio between the train and test set. About 1/6 of the images for each split show the Earth in the background, while the others have the background perfectly black, representing deep space. All images are 16-bit grayscale in PNG format with 1024×1024 pixels resolution. The scenes are such that the target spacecraft altitude ranges from 1500 km to 6500 km with respect to the Earth's surface. The relative distance between the target and the camera is uniformly distributed in the range from 5 m to 30 m. Fig. 7 reports examples of random images with Earth in the background extracted from the dataset.

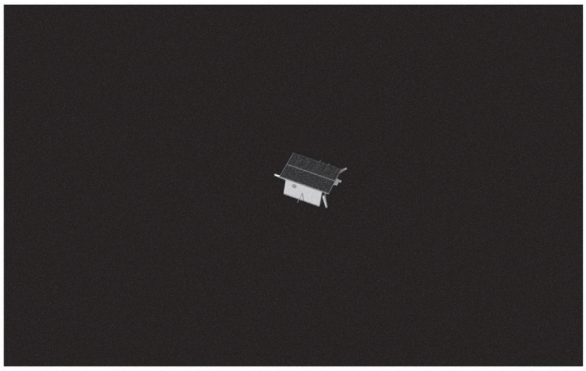
Table 4
Parameters of the camera used for datasets generation.

Parameter	Value
Number of horizontal pixels	1024 px
Number of vertical pixels	1024 px
Focal length	6 mm
Pixel length	4.8×10^{-3} mm

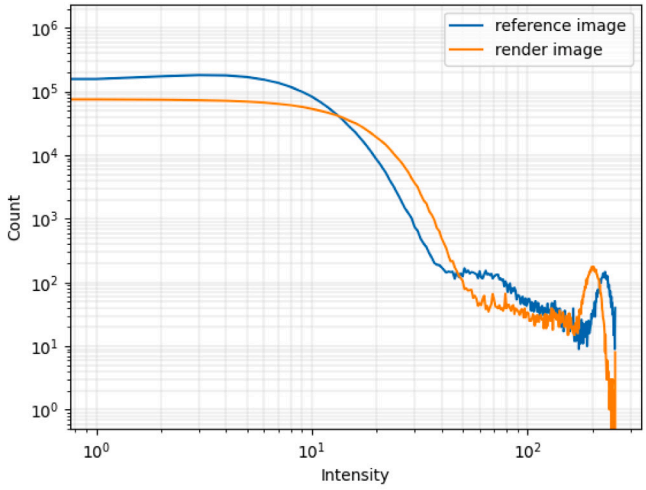
The main algorithm places the Earth in the scene at the origin of the ECI frame. After that, the position of the target around the Earth is prescribed by randomly picking the altitude in the previously defined range. Then the 3D coordinates in ECI of the target are selected by following a uniform distribution over a shell around the Earth of thickness equal to the altitude randomly picked in the previously defined



(a) SPEED image *img002350.jpg*.



(b) Rendered *img002350.jpg*.



(c) Histogram comparison between Fig. 6a (reference - blue line) and Fig. 6b (render - orange line).

Fig. 6. Histogram comparison between SPEED image "*img002350.jpg*" and the reproduced noised image.

range (see Fig. 8(a)). The attitude of the target, i.e. the orientation of the target reference frame (TRG), with respect to the ECI frame, named $R_{ECI \to TRG}$, is randomly defined and used to correctly place the Tango model in the POV-Ray scene. As it can be noticed from Fig. 8(b), the attitude of the target, with respect to the ECI frame, is uniformly distributed in roll, pitch, and yaw angles.

The camera is located in the scene by randomly selecting a position in a sphere centered at the target position with a radius ranging from 5 m to 30 m. The pointing direction (i.e. the "*look_at*" parameter in POV-Ray) is needed to fully define the attitude of the camera with respect to the ECI frame. The pointing vector has been computed by adding a random displacement vector to the target position vector in the ECI frame to avoid having the target always located at the center of the image. The displacement vector magnitude is constrained to keep the target in the scene inside the field of view of the selected camera. The components of the attitude of the camera, i.e. the orientation of the camera reference frame (CAM), with respect to the ECI frame, named $R_{ECI \to CAM}$, can be defined as:

$$\hat{z}_{ECI \to CAM} = \frac{p_{cam} - l_{cam}}{\|p_{cam} - l_{cam}\|} \tag{4}$$

$$\hat{x}_{ECI \to CAM} = \frac{\hat{z}_{ECI \to CAM} \times s_{cam}}{\|\hat{z}_{ECI \to CAM} \times s_{cam}\|} \tag{5}$$

$$\hat{y}_{ECI \to CAM} = \frac{\hat{z}_{ECI \to CAM} \times \hat{x}_{ECI \to CAM}}{\|\hat{z}_{ECI \to CAM} \times \hat{x}_{ECI \to CAM}\|} \tag{6}$$

where p_{cam} is the camera pointing vector in the ECI frame, l_{cam} is the camera location vector in the ECI frame, and $s_{cam} = [0, 0, 1]^T$ is the "*sky*" vector in POV-Ray. Note that now it is possible to define the

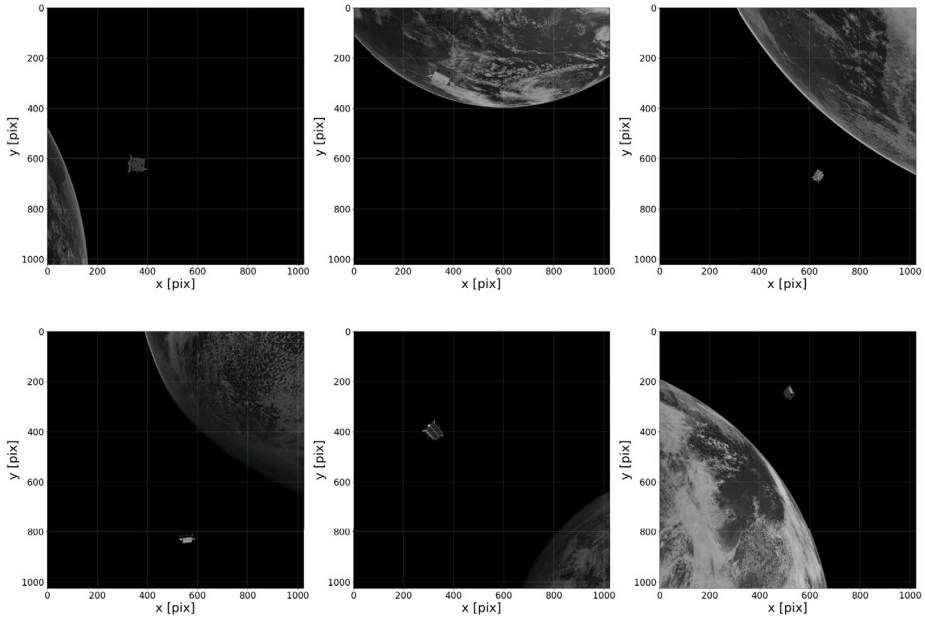
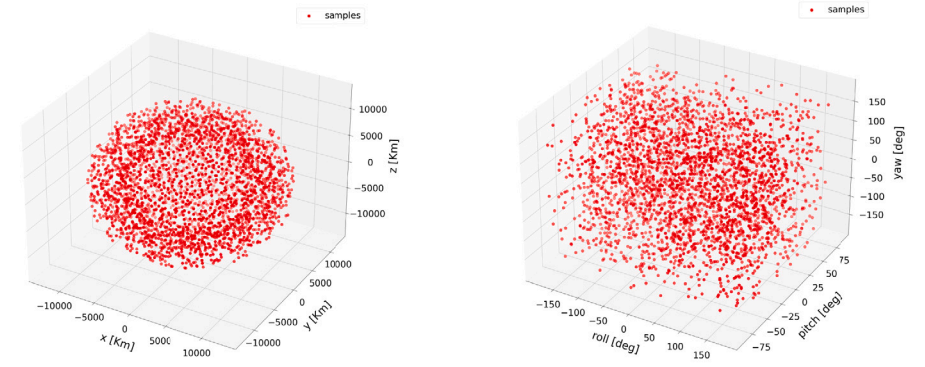


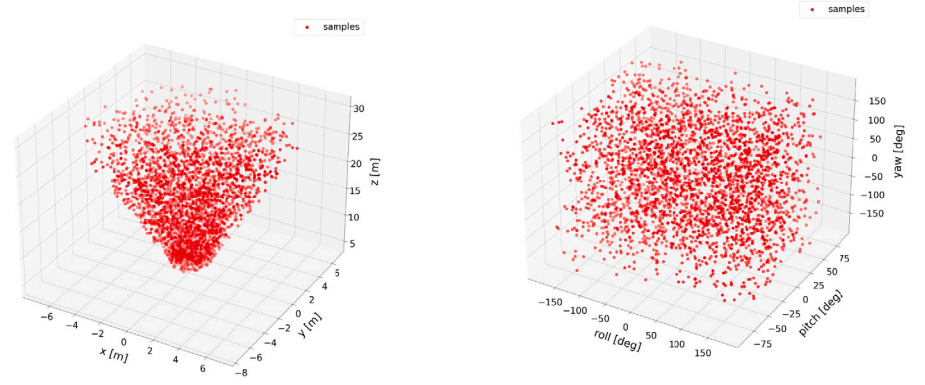
Fig. 7. Random images with background from the generated dataset.



(a) Target position distribution in the test set in ECI reference frame.

(b) Target attitude distribution in the test set with respect to the ECI reference frame.

Fig. 8. Target position and attitude distribution with respect to ECI reference frame in test set.



(a) Target relative position distribution in the test set in camera reference frame.

(b) Target relative attitude distribution in the test set with respect to the camera reference frame.

Fig. 9. Target relative position and attitude distribution with respect to camera reference frame in test set.

ground truth relative pose, i.e. the attitude of the target with respect to the camera in camera reference frame $R_{TRG\rightarrow CAM}$ and the relative position of the target with respect to the camera in camera reference frame $t_{CAM\rightarrow TRG}$ as:

$$R_{TRG\rightarrow CAM} = R_{ECI\rightarrow CAM}\ \left(R_{ECI\rightarrow TRG}\right)^T \tag{7}$$

$$t_{CAM\rightarrow TRG} = R_{ECI\rightarrow CAM}\ \left(l_{trg} - l_{cam}\right) \tag{8}$$

where l_{trg} is the location vector of the target in the ECI frame. By following the steps formulated above, it is possible to obtain a uniform random distribution of the relative attitude and position of the target with respect to the camera. Fig. 9 shows the relative position distribution (Fig. 9(a)) and the relative attitude distribution (Fig. 9(b)) in the test set. Notice that the distribution in Fig. 9(a) has a conical shape because the target is constrained to be inside the field of view of the selected camera. Finally, to correctly render the image, the light source (i.e. the Sun model) is placed in the scene on the ecliptic plane by following a random uniform distribution to ensure a complete range of illumination conditions inside the dataset. Notice that the position of the Sun is constrained to be out of the field of view of the camera. Moreover, imposing that neither the camera nor the target can be in eclipse avoids having black images in the dataset. From all previous information, by exploiting simple geometric relations, it is possible to define before the rendering if the Earth is in the scene and if it is in light or shadow. To speed up the image generation, the Earth is modeled as in Section 3.1 only if it is visible and in light. Otherwise, it is substituted by an equivalent sphere with equivalent reflection properties to take always into account the albedo. Note that this step is highly advantageous in terms of computational time because both the parsing time and the rendering time strongly increase if the full Earth model has to be rendered. With the current implementation, the parsing time and rendering time for an image without background are in the order of 0.5 s and 0.4 s, respectively, while in the case the Earth is visible the parsing time is about 5 s per image, while the rendering time has a high variance, but on average it takes about 2 min per image. The parsing time is evaluated on a single-thread single-CPU process, while the rendering time is on a 16-threads 8-CPUs process. The CPUs used are Intel® Core™ i7-9700 CPU, clocked at 3.00 GHz.

The plots shown in Figs. 8 and 9, together with the normalized Sun position distribution with respect to the target reference frame shown in Fig. 10, confirm the uniform distribution of all main parameters of the image dataset generated and confirm the representativeness of a wide range of orbital scenarios and illumination conditions for the given target, making the dataset itself well suited for GNC algorithms testing and development. Notice that the images in Figs. 8–10 are reported only for the test set but also hold for the training set since it has been obtained with the same tool just by increasing the number of the desired output images. The output pose labels coupled with the images have been collected in a dataset named *Tango Spacecraft Dataset for Monocular Pose Estimation* that has been made publicly available [21].

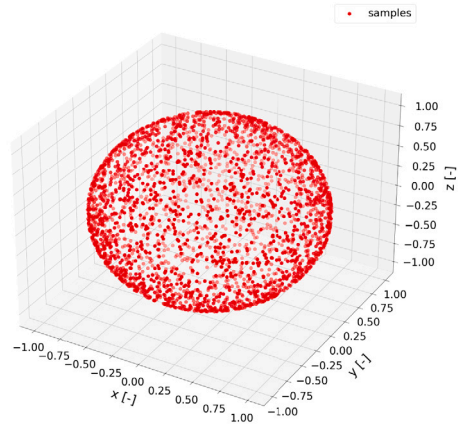


Fig. 10. Normalized Sun position distribution in the test set in Target reference frame.

The same pipeline discussed above has also been exploited to create RGB masks for each scene. The RGB images are such that the R-channel (red) corresponds to the spacecraft, the G-channel (green) to the Earth (if present), and the B-channel (blue) to the background (deep space). Per each channel, the pixels have non-zero values only in correspondence with the object they represent (Tango, Earth, Deep Space). The scene is rendered by adopting a properly tuned single-color texture for both the spacecraft and the total volume occupied by the Earth and the atmosphere. Fig. 11(a) shows an example of a RGB mask. Notice that the RGB masks can be exploited for semantic segmentation tasks. In addition, by using the spacecraft-related channel, it is possible to extract the associated bounding box for each image. This task has been accomplished, and both the RGB masks and the bounding box annotations have been collected together with the related grayscale images (shared with the dataset in [21]) in a dataset that has been made publicly available [22], named *Tango Spacecraft Dataset for Region of Interest Estimation and Semantic Segmentation*. Concerning the bounding box annotations, they are taken in the image reference frame, with the origin located in the top-left corner of the image. The bounding box is labeled through the top-left corner, the bottom-right corner, and the center coordinates in pixels . To prove the correctness of the relative pose annotations, they have been exploited to project a simplified wireframe model of the Tango spacecraft in the image space, by using the following transformation:

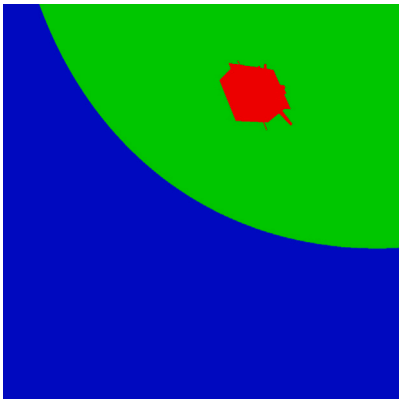
$$x_{px} = K\ [\ R_{TRG\rightarrow CAM}\ |\ t_{CAM\rightarrow TRG}\ ]\ x^H_{TRG} \tag{9}$$

where $x_{px} = [x_{px}/z_{px},\ y_{px}/z_{px},\ 1]^T$ is the position vector in the image frame in pixels of the projected point, $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, and $x^H_{TRG} = [x_{TRG},\ y_{TRG},\ z_{TRG},\ 1]^T$ is the position homogeneous vector in the target reference frame of a specific point. By using a properly defined simplified mesh of the target spacecraft and exploiting the Möller–Trumbore ray-triangle intersection algorithm [45], it is possible to extract the line-segment portions that are in view per each image (see Fig. 11(b) for an example). Another dataset has been created to include this additional information. Hence, the same images of the two datasets previously discussed (both in full-scale and in a bounding-box cropped version) have been labeled with the line segments in view per each image. This dataset, named *Tango Spacecraft Wireframe Dataset Model for Line Segments Detection*, has been made publicly available [23].

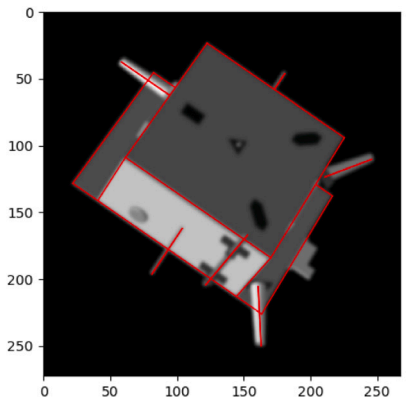
It is here remarked that these datasets [21–23] contain the same fullscale images, hence they can be used together by combining the annotations of the relative pose, the reprojected wireframe model of Tango, and also the region of interest. These three datasets constitute together the most comprehensive multi-purpose labeled dataset of validated spaceborne synthetic images publicly available, at least up to our knowledge.

6. Conclusions and future works

The paper presented here highlighted the necessity of a viable method to generate high-quality spaceborne synthetic images to support the development of new algorithms for relative navigation and to serve as a starting point through new missions with a high level of autonomy required. This need is addressed with a new image generation tool based on POV-Ray, an open-source ray-tracing software, exploited here as the core of the dataset rendering chain developed. The main assumptions and simplifications adopted for creating representative space-related images in POV-Ray are discussed in detail, leading to models for both the target spacecraft and the celestial body (used here as background) with a high degree of photo-realism. The simplifications



(a) Example of RGB mask image from dataset.



(b) Example of in-view wireframe reprojection in a cropped image of the dataset.

Fig. 11. Examples of additional annotations generated. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

used to model a celestial body with an atmospheric layer and a well-defined topography have been analyzed both in terms of accuracy and rendering time, leading to a trade-off selection of the default functionality implemented in the tool. The implementation for the background object, which can be chosen by the user to be included or not in the scene, can be used to model several celestial bodies by tuning a few hyperparameters, making the tool proposed highly flexible. The full pipeline to generate a spacecraft model compatible with POV-Ray has been explained in detail, from the 3D CAD model to the textured POV-Ray readable "include file" that can be handled easily by the image generation tool. It is here acknowledged that having a highly detailed CAD model is mandatory for some applications such as image segmentation and neural networks training since a strong drift between the training set and the actual application scenario, commonly named domain gap, can jeopardize the entire image processing pipeline. The pipeline discussed has been applied to a simplified model of the Tango spacecraft from PRISMA mission. The optical properties of the spacecraft have been tuned during the validation phase. The validation of the proposed tool tailored to the generation of spaceborne synthetic noiseless image dataset via physically-based ray-tracing has been carried out qualitatively via histogram comparison, and quantitatively by evaluating two different indexes with respect to the SPEED dataset images. Despite the assumptions made and the simplified CAD model used, the validation phase highlighted the high level of accuracy of the synthetic images rendered with the proposed tool. In conclusion, the successful validation phase carried out by using the SPEED dataset as ground truth demonstrates that the proposed image generation chain can generate high-fidelity spaceborne images. The last part of the paper delves into the details of the datasets that have been created by using the previously discussed image generation pipeline and further delving into the details of the image dataset generation algorithm. In particular, the distributions of the relative pose and the illumination conditions of the target through the datasets show that the images included are representative of a wide range of scenarios. It is worth remarking that the images are noiseless to increase the flexibility of the datasets since the noises can be easily added in the postprocessing phase to properly tailor the images to the final application scenario. We acknowledge here that some disturbances in the images, such as flares and stray lights, are due to the refraction and reflection of the light inside the lens set of a real camera. Hence, algorithms to simulate the physics of the light inside a complete set of lenses during ray-tracing still need to be developed to include these effects in the synthetic images. The annotations of the images are extracted automatically during the image generation and, in particular, the equations and algorithms needed to extract the relative pose annotations, as well as additional labels as the RGB masks for each scene and the in-view wireframe reprojections in the image reference frame, have been discussed. The datasets created have been made publicly available. Since the datasets share the same fullscale images, but with different annotations, they can be combined to constitute the most comprehensive multi-purpose labeled dataset of validated spaceborne synthetic images publicly available, at least up to our knowledge.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors want to acknowledge Francescodario Cuzzocrea for his extremely appreciated support in the development of the synthetic image generation tool presented here.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] D.K. Geller, Orbital rendezvous: When is autonomy required? J. Guid. Control Dyn. 30 (4) (2007) 974–981, http://dx.doi.org/10.2514/1.27052.

[2] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations, Prog. Aerosp. Sci. 93 (2017) 53–72, http://dx.doi.org/10.1016/j.paerosci.2017.07.001.

[3] R. Volpe, C. Circi, M. Sabatini, G.B. Palmerini, GNC architecture for an optimal rendezvous to an uncooperative tumbling target using passive monocular camera, Acta Astronaut. 196 (2022) 380–393, http://dx.doi.org/10.1016/j.actaastro.2020.10.038.

[4] V. Capuano, K. Kim, A. Harvard, S.-J. Chung, Monocular-based pose determination of uncooperative space objects, Acta Astronaut. 166 (2020) 493–506, http://dx.doi.org/10.1016/j.actaastro.2019.09.027.

[5] M.R. Leinz, C.T. Chen, M.W. Beaven, T.P. Weismuller, D.L. Caballero, W.B. Gaumer, P.W. Sabasteanski, P.A. Scott, M.A. Lundgren, Orbital express autonomous rendezvous and capture sensor system (ARCSS) flight test results, in: R.T. Howard, P. Motaghedi (Eds.), Sensors and Systems for Space Applications II, in: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6958, 2008, p. 69580A, http://dx.doi.org/10.1117/12.779595.

[6] F. Castellini, D. Antal-Wokes, R.P. de Santayana, K. Vantournhout, Far approach optical navigation and comet photometry for the Rosetta mission, in: Proceedings of 25^{Th} International Symposium on Space Flight Dynamics, 25^{Th} ISSFD, 2015.

[7] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: 2011 International Conference on Computer Vision, Vol. 1, 2011, pp. 2564–2571, http://dx.doi.org/10.1109/ICCV.2011.6126544.

[8] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679–698, http://dx.doi.org/10.1109/TPAMI.1986.4767851.

[9] S. Sharma, S. D'Amico, Comparative assessment of techniques for initial pose estimation using monocular vision, Acta Astronaut. 123 (2016) 435–445, http://dx.doi.org/10.1016/j.actaastro.2015.12.032.

[10] S. Sharma, J. Ventura, S. D'Amico, Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous, AIAA J. Spacecr. Rockets 55 (6) (2018) 1414–1429, http://dx.doi.org/10.2514/1.A34124.

[11] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1) (1972) 11–15, http://dx.doi.org/10.1145/361237.361242.

[12] L. Pasqualetto Cassinis, R. Fonod, E. Gill, I. Ahrns, J. Gil-Fernández, Evaluation of tightly- and loosely-coupled approaches in CNN-based pose estimation systems for uncooperative spacecraft, Acta Astronaut. 182 (2021) 189–202, http://dx.doi.org/10.1016/j.actaastro.2021.01.035.

[13] J. Song, D. Rondao, N. Aouf, Deep learning-based spacecraft relative navigation methods: A survey, Acta Astronaut. 191 (2022) 22–40, http://dx.doi.org/10.1016/j.actaastro.2021.10.025.

[14] P.F. Proença, Y. Gao, Deep learning for spacecraft pose estimation from photorealistic rendering, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 6007–6013, http://dx.doi.org/10.1109/ICRA40945.2020.9197244.

[15] S. Sonawani, R. Alimo, R. Detry, D. Jeong, A. Hess, H.B. Amor, Assistive relative pose estimation for on-orbit assembly using convolutional neural networks, in: AIAA Scitech Forum, 2020, American Institute of Aeronautics and Astronautics Inc, AIAA, 2020, pp. 1–11, http://dx.doi.org/10.2514/6.2020-2096.

[16] B. Chen, J. Cao, A. Parra, T.-J. Chin, Satellite pose estimation with deep landmark regression and nonlinear pose refinement, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, IEEE Computer Society, 2019, pp. 2816–2824, http://dx.doi.org/10.1109/ICCVW.2019.00343.

[17] M. Piazza, M. Maestrini, P. Di Lizia, Monocular relative pose estimation pipeline for uncooperative resident space objects, J. Aerosp. Inf. Syst. 19 (9) (2022) 613–632, http://dx.doi.org/10.2514/1.I011064.

[18] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Satellite pose estimation challenge: Dataset, competition design, and results, IEEE Trans. Aerosp. Electron. Syst. 56 (5) (2020) 4083–4098, http://dx.doi.org/10.1109/TAES.2020.2989063.

[19] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Spacecraft Pose Estimation Dataset (SPEED), 2019, http://dx.doi.org/10.5281/zenodo.6327546, Zenodo.

[20] S. D'Amico, P. Bodin, M. Delpech, R. Noteborn, PRISMA, in: M. D'Errico (Ed.), Distributed Space Missions for Earth System Monitoring, Springer New York, New York, NY, 2013, pp. 599–637, http://dx.doi.org/10.1007/978-1-4614-4541-8-21.

[21] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Dataset for Monocular Pose Estimation, 2022, http://dx.doi.org/10.5281/zenodo.6499008, Zenodo.

[22] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Dataset for Region of Interest Estimation and Semantic Segmentation, 2022, http://dx.doi.org/10.5281/zenodo.6507863, Zenodo.

[23] M. Bechini, P. Lunghi, M. Lavagna, Tango Spacecraft Wireframe Dataset Model for Line Segments Detection, 2022, http://dx.doi.org/10.5281/zenodo.6372848, Zenodo.

[24] M. Bechini, M.B. Quadrelli, M. Lavagna, J.J. Wang, Hovering of an electrically actuated spacecraft in a small-body plasma field, AIAA J. Spacecr. Rockets 58 (5) (2021) 1461–1476, http://dx.doi.org/10.2514/1.A34954.

[25] P. Shirley, R.K. Morley, Realistic Ray Tracing, second ed., Taylor & Francis, 2008.

[26] S. Parkes, I. Martin, M. Dunstan, D. Matthews, Planet surface simulation with PANGU, in: Space OPS 2004 Conference, 2004, p. 389, http://dx.doi.org/10.2514/6.2004-592-389.

[27] R. Brochard, J. Lebreton, C. Robin, K. Kanani, G. Jonniaux, A. Masson, N. Despré, A. Berjaoui, Scientific image rendering for space scenes with the SurRender software, 2018, arXiv preprint arXiv:1810.01423.

[28] B.O. Community, Blender - a 3D Modelling and Rendering Package, Blender Foundation, 2018.

[29] B. Karis, Epic Games, Real shading in Unreal Engine 4, in: Proc. Physically Based Shading Theory Practice, Vol. 4, No. 3, 2013, p. 1.

[30] T. Plachetka, POV Ray: Persistence of vision parallel raytracer, in: Proc. of Spring Conf. on Computer Graphics, Budmerice, Slovakia, Vol. 123, 1998, p. 129.

[31] M. Pharr, W. Jakob, G. Humphreys, Physically Based Rendering: From Theory to Implementation, Morgan Kaufmann, 2016.

[32] T.H. Park, M. Märtens, G. Lecuyer, D. Izzo, S. D'Amico, Next Generation Spacecraft Pose Estimation Dataset (SPEED+), 2021, http://dx.doi.org/10.25740/wv398fc4383, Zenodo.

[33] C. Beierle, S. D'Amico, Variable-magnification optical stimulator for training and validation of spaceborne vision-based navigation, AIAA J. Spacecr. Rockets 56 (4) (2019) 1060–1072, http://dx.doi.org/10.2514/1.A34337.

[34] T.H. Park, M. Märtens, G. Lecuyer, D. Izzo, S. D'Amico, SPEED+: Next generation dataset for spacecraft pose estimation across domain gap, in: 2022 IEEE Aerospace Conference, AERO, 2022, http://dx.doi.org/10.1109/AERO53065.2022.9843439.

[35] J. Guarneri, Autonomous Optical Navigation and Attitude Estimation System Tested on Artificially Generated Images (MSc Thesis), Politecnico Di Milano, 2020.

[36] P. Lunghi, M. Ciarambino, M. Lavagna, A multilayer perceptron hazard detector for vision-based autonomous planetary landing, Adv. Space Res. 58 (1) (2016) 131–144, http://dx.doi.org/10.1016/j.asr.2016.04.012.

[37] A. Rivolta, P. Lunghi, M. Lavagna, GNC & robotics for on orbit servicing with simulated vision in the loop, Acta Astronaut. 162 (2019) 327–335, http://dx.doi.org/10.1016/j.actaastro.2019.06.005.

[38] F. Cuzzocrea, Analysis and Validation of Spaceborne Synthetic Imagery Using a Vision-Based Pose Initialization Algorithm for Non-Cooperative Spacecrafts (MSc Thesis), Politecnico Di Milano, 2020.

[39] N.E. Daidzic, et al., Efficient general computational method for estimation of standard atmosphere parameters, Int. J. Aviat. Aeronaut. Aerosp. 2 (1) (2015) 3, http://dx.doi.org/10.15394/ijaaa.2015.1053.

[40] A.S. Jursa, Handbook of Geophysics and the Space Environment, fourth ed., Final Report Air Force Geophysics Lab, Hanscom AFB, MA, 1985.

[41] United States Committee on Extension to the Standard Atmosphere and United States. Environmental Science Services Administration, US Standard Atmosphere, 1962: ICAO Standard Atmosphere to 20 kilometers; Proposed ICAO Extension to 32 kilometers; Tables and Data to 700 kilometers, US Government Printing Office, 1962.

[42] NASA, Earth from space - Apollo 17. this image or video was catalogued by Langley Research Center of the United States National Aeronautics and Space Administration (NASA) under photo ID: EL-1996-00155 and alternate ID: L86-5815, 1996, https://web.archive.org/web/20080617152411/http://lisar.larc.nasa.gov/UTILS/info.cgi?id=EL-1996-00155. (Accessed 18 January 2022).

[43] M. Piccinin, S. Silvestrini, G. Zanotti, A. Brandonisio, P. Lunghi, M. Lavagna, ARGOS: calibrated facility for image based relative navigation technologies on ground verification and testing, in: 72^{Nd} International Astronautical Congress, IAC 2021, International Astronautical Federation, IAF, Dubai, United Arab Emirates, 2021, pp. 1–12.

[44] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66, http://dx.doi.org/10.1109/TSMC.1979.4310076.

[45] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, J. Graph. Tools 2 (1) (1997) 21–28, http://dx.doi.org/10.1080/10867651.1997.10487468.