

CLOWN: a new tool for cloud detection with an All-Sky camera for optimization of space-debris surveys.
Luís Gonçalves^{a*}, Ana Sousa^b

^a *Departamento de Física, Instituto de Telecomunicações, Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*

^b *Departamento de Eletrónica, Telecomunicações e Informática, Instituto de Telecomunicações, Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*

* Corresponding Author

Abstract

The ever-growing competition in space exploration has led to the accumulation of a great number of space debris in orbit, which due to high speeds can cause great damage to active satellites or manned missions. As such, there has been a growing effort in the worldwide monitoring and surveying capacity of this debris to protect present and future missions through the use of radar and optical telescopes. The optical telescopes typically do acquire thousands of images per night and their operation time has to be optimized for space surveillance and tracking operations. This project aims to create a real-time automatic detection of clouds in the sky, to optimize the operations of the new large field-of-view double telescope, which is being installed at the Pampilhosa da Serra Space Observatory (PASO), Portugal, as a future sensor of the Portuguese node of the EU Space Surveillance and Tracking (SST) program. The telescope platform is designed to conduct automatic space debris surveillance and tracking operations. The project's main goal aims to avoid dark images (cloudy) taken while surveying the sky, optimize the telescope time, reduce unnecessary processing, and increase the telescope's efficiency. The method uses an all-sky camera to obtain images of the sky, analyses the images to map the presence of clouds, and relays that information to assist the automatic operation of the telescope.

The CLOud Watcher at Night CLOWN tool was developed to work with any type of all-sky camera. CLOWN requires minimal information like the camera location and acquired images' timestamps. The tool app can be used with any camera since it is not model or brand-specific enabling its integration with any already installed sensor equipment. CLOWN uses the presence or absence of stars in the image to confirm cloud detection even in difficult detection conditions. The software tool was also developed to have a low computational cost, making it possible to use local resources already installed in any observatory without major investment. The tool results are very good: it correctly identifies and maps clouds in a variety of sky and meteorological conditions and improves space debris survey operations. This tool can be implemented at any observatory to increase the efficiency of telescope time, not only for space-debris surveillance but also for astronomical surveys. In the future, CLOWN will also be used to assist other robotic equipment at PASO and may be integrated into a visual assistance tool to aid operations of other observatory sensors. **Keywords:** (space debris, cloud, telescope, all-sky camera)

Acronyms/Abbreviations

Space Surveillance & Tracking – Portugal (SST-PT), Pampilhosa da Serra Space Observatory (PASO), Medium/Geosynchronous/Low Earth Orbit (MEO, GEO, LEO), Field of View (FoV), Observatório Pico dos Dias (OPD), Laboratório Nacional de Astronomia (LNA), Laplacian of Gaussian (LoG), False Positives (FP)

1. Introduction

With the growing demand and interest for space exploration and the development of new technologies, there has been an increase in satellite launches in the last few decades with many more planned for the coming years. As far back as 1978, there has been a worry about the problem of space debris accumulation, designed as the “Kessler Syndrome” [1]. This problem is expected to quickly escalate, as there has been a great surge of interest in companies to create their mega-constellation of satellites, such as SpaceX, OneWeb,

and Amazon with the possibility of others to follow. Just in November of last year, the International Space Station (ISS) feared a possible impact from space debris, caused by the explosion of a Russian satellite. To prevent impacts, NASA alone watches over more than 27000 space debris in orbit, with sizes bigger than 10 cm in diameter, but with the possible rapid increase in their number, it becomes imperative to develop more tools to follow these objects in an automated manner [2,3].

As part of the SST-PT, PASO is equipped with a set of optical sensors and a radar system. One of the systems, installed this year (2022), is a large FoV double telescope, able to observe a field of 4.3° x 2.3°. It will incorporate a set of filters, to allow chromatic characterization throughout the visible spectrum. It is primarily aimed to track objects in LEO, however, it will also be able to track and survey objects in MEO and GEO regimes [4]. Since survey observation will be

conducted in an automated way, it is necessary to consider a multitude of factors, such as the moon's position and phase, the earth's shadow, and weather conditions. The presence of clouds may lead to the production of useless images. To optimize operations during surveys, it was developed an automatic and real-time cloud detection tool, CLOWN.

2. Material and methods

Since any optical observatory has an all-sky camera, so operators can monitor sky conditions, CLOWN software was developed to work with any type of all-sky camera. It was developed in Python and does not require heavy processing capabilities in terms of hardware, so it can be run on any laptop/desktop computer which receives the images from the camera. The tool will be available later this year on GitHub.

3. Principles of all-sky cameras

An all-sky camera is a type of camera that can observe the entire visible sky in a single image in 360° vision above the horizon. This is achieved by using a special type of lens, the fish-eye lens. There are different variations for this lens, which deform and record the image in different ways. Knowing the type of fish-eye lens is important to configure the software with the best setting possible and correctly analyse the sky images.

Firstly, since the sensor is usually rectangular, and a fish-eye lens projects a circular image, depending on the camera design, the recorded image may fall in one of three situations or types: i) circular type, where the image circle is inscribed into the sensor area. This type has the advantage of capturing the entire sky in a single image but uses only a portion of the sensor area; ii) the second type or full-frame, where the sensor area is inscribed in the image circle. This type has the advantage of using every pixel in the sensor, increasing the resolution of the image; iii) The third type or cropped circle, in which the image circle diameter is aligned with the largest side of the sensor. Secondly, the fish-eye lens has 4 main types of projections: stereographical, equidistant, equisolid angle, and orthographic. Each of them distorts the image differently, according to the expressions given in table 1 [5].

The stereographical projection has the property of preserving angles between curves and allows for 3D reconstructions. The equidistant projection implies a linear relation between r and θ , which makes it the most desirable for astronomical applications. In the equisolid angle projection, each pixel in the image represents the same amount of area in the sky. The orthographic projection has the advantage of retrieving the same FoV with a smaller sensor when compared to the other projections [5].

Although this information is important to calibrate the image, even in cases where it is unavailable, the software still provides good results, as will be explained further in section 4.5.

Table 1. Distortion expression for each type of projection, relating the focal distance (f) with the distance to the center of the sensor (r) and the corresponding angle θ .

Projection	Formula
Stereographical	$f = \frac{r}{2 \tan\left(\frac{\theta}{2}\right)}$
Equidistant	$f = \frac{r}{\theta}$
Equisolid Angle	$f = \frac{r}{2 \sin\left(\frac{\theta}{2}\right)}$
Orthographic	$f = \frac{r}{\sin(\theta)}$

4. CLOWN

In this section, we explain the CLOWN tool as well as other software used in the process that can be used to complement.

4.1 Image Acquisition Software

The camera used for PASO is the Oculus All-Sky Camera 180° [6]. It has an equidistant projection with a focal distance of $f=1.55$ mm and a pixel size of 4.65 μm . It comes with image acquisition software. We found that this software, while good enough for short time observations, is not the best choice for a prolonged automatic survey. Additionally, it is only available with the camera and can not be used for a camera already implemented. Instead, we used the software AllSkEye [7] for image acquisition, which is publicly available, works with a large range of cameras, and comes with a multitude of features.

The algorithm developed can be used with any all-sky camera, even if we lack detailed datasheet information (like it is the case for old cameras). To confirm, we also tested the software on images from the all-sky camera installed at OPD of LNA in Brazil, which are publicly available at [8].

4.2 Star Detection

We firstly identify which visible stars are in the field image and check which stars are missing in the image within the same magnitude range. Stars in an all-sky image will appear as bright blobs on a dark background. There is a multitude of ways to calculate pixel variations in an image. One way to do so is to apply the

Laplacian filter, which approximates the second derivative. A disadvantage to this method is the fact that the image noise variations may trigger the filter. To solve this problem, it is usually applied a Gaussian blurring step, to smooth these noise variations. Due to the associative property of the convolution operation, we can apply these two steps simultaneously, significantly decreasing the computational cost. This filter is usually called LoG and it is the norm for blob detection in an image with a gaussian shape, making it ideal for this work. Another advantage of the gaussian step is that it can be used to detect the sizes of the blobs, as it will originate a stronger response only when the radius of the blobs is close to $\sqrt{2}\sigma$, where σ is the standard deviation used for the gaussian filter. To apply this filter, we used the “blob_log” method from the “astropy” library [9,10]. The presence of objects in the image, such as buildings in the horizon, can create false detections which may slow down the algorithm significantly. As these objects are constant in time, they can easily be removed using a mask. One example of a mask can be seen in figure 1.

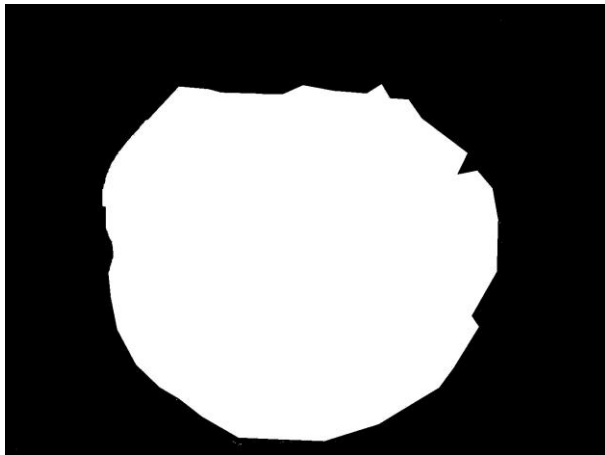


Fig. 1 PASO camera mask, in black the pixels to ignore; and in white the pixels of the zone of interest.

4.3 Star Matching

The second step is to compare the positions of the detected objects with a catalog. We chose the Hipparcos catalog [11] which contains enough objects for the application, as the all-sky camera does not reach very high magnitudes (~6). To compare the two sets, we need to find a relation between the (x,y) position in the image and the (Ra, Dec) coordinates of the catalog

To do this we transform the (x,y) cartesian coordinates into (r,θ) polar coordinates using the center¹ of the image as the origin, as can be seen in figure 2 in green and red respectively. After this transformation, it is easy to relate to (Alt, Az), which is related to (Ra, Dec) by known equations.

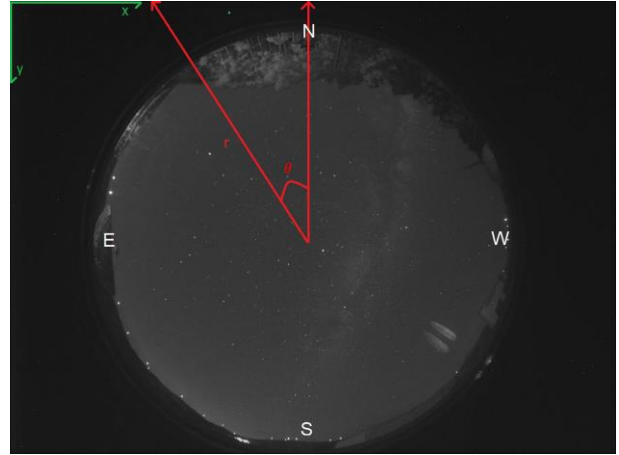


Fig. 2. Polar coordinate system

Assuming that the North is as shown in figure 2, the Azimuth angle will perfectly coincide with the θ value of the polar coordinates. Practically, this will never be the case, and the image will be rotated by some degrees creating an offset between θ and Azimuth. This effect is permanent and can be easily calculated in the calibration and fixed.

If the origin coincides with the zenith of the image, the Altitude angle can be calculated using the value of r of the polar coordinates and the corresponding camera transformation. In this way, we can calculate the (Alt, Az) coordinates of any pixel in the image. In case the camera information (focal distance, pixel size, and transformation type) is not available, it can be approximated using the linear transformation (equidistant). In this case, it is easier to estimate the relation between r and θ , as it is known that the length/width/diagonal of the sky image will correspond to 180°. Another advantage of this transformation is that it produces intermediate solutions compared with the other projections, decreasing the average error when the projection is unknown. The zenith may not be perfectly positioned in the center of the image, but the pixel can be determined in the calibration.

Using this process, we were able to obtain a median error of 0.42° for the position of stars, with our all-sky camera installed in PASO, and 0.61° for the OPD camera. Knowing the camera characteristics slightly decreases the error but even without camera information, it is still precise enough for the chosen application [8].

We used the library “astropy” to transform the Alt/Az coordinates into Ra/Dec [9,10].

4.4 Cloud Mask Creation

The cloud mask was thought to be a binary image, to remove ambiguity about the cloud level, forcing a decision for the threshold value.

To create the cloud mask, we drew a blob in every position for a missing object. This would not create a smooth cloud image and, due to FP, would create some isolated blobs. To resolve these issues, it was implemented the second step of image analysis. In this step, it was applied a local binarization, where for each pixel it was analysed the values of pixels in a selected region and compared with a threshold value. These 3 values can be controlled by the user, in the configuration file, to better improve results. The value of the blob size used for the PASO and OPD all-sky images was obtained by dividing the number of useful pixels in the image and the average number of stars detected in an image. The size of the region and the threshold value were tested for each camera.

In the case of the moon illuminating a big cloud, the program would falsely identify a very big set of small objects in the cloud itself, far exceeding the number of expected objects in the image, and it would not detect the stars, as they would not be present in the image due to pixel saturation. To use it to our advantage, it was created a condition that if the number of detected objects far exceeded the expected objects, to draw the cloud using the detected points, skipping the star matching step.

4.5 Calibration

The calibration of the zenith pixel and the offset in the Azimuth coordinate will be done only once, and the values are then stored since the all-sky camera installation is permanent.

To do this, we used the program “Stellarium” to simulate the sky observed at the time of a certain image. Then we correspond some stars in the image to stars observed in “Stellarium”, and save both the pixel coordinates (x,y) from the image and the horizontal coordinates (Alt, Az) from “Stellarium”. Then, we can estimate the values for these coordinates with minimal error.

In figures 3 and 4, we can see an example of an image and the simulated sky from “Stellarium”.

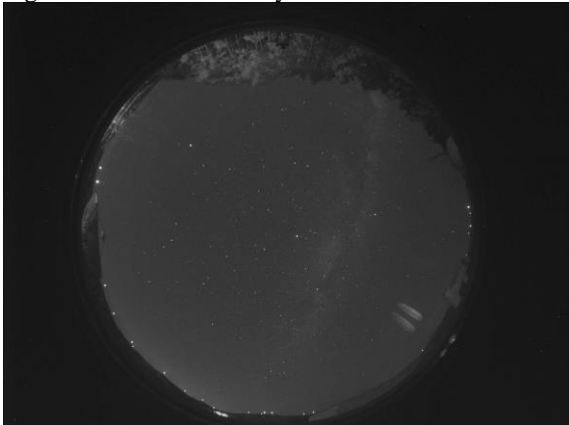


Fig. 3. Clear sky image from PASO



Fig. 4. Simulated sky from “Stellarium”, already oriented according to figure 3 [12]

5. Results

In this section, we show some sky images and respective cloud masks, both for our images and the LNA images. There are “clouds” detected in the outskirts of the image, which are due to the horizon of the image both in the clear sky example and the cloudy example. This does not create a problem, as the telescope does not observe in this area and so they can be ignored. In the cloud maps, (figures 6 and 8), we can observe that there are some holes in the clouds. The size of these holes is not significant enough to compromise the assessment of clouds, for our optical telescope so they can also be ignored.

In figure 9 we can see an example of a possible application. If the planned surveys for a certain night were the areas drawn on the image, a normal plan would start on the left-hand side where the image would be blocked by a cloud. With this software, the telescope can immediately skip the first few regions until it finds a clear sky. Later in the night, the skipped regions could be analysed when the cloud passed through.

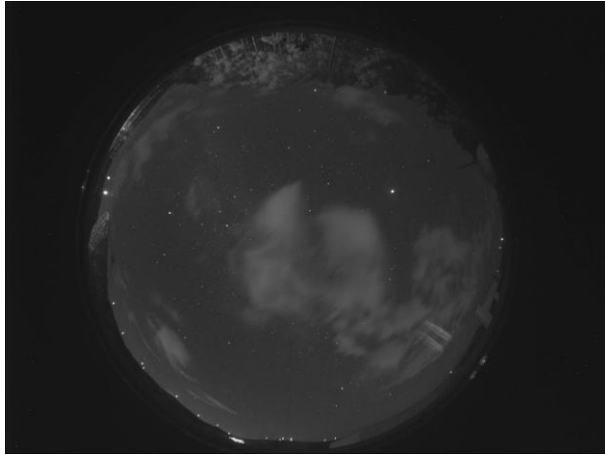


Fig. 5. Cloudy sky image from PASO

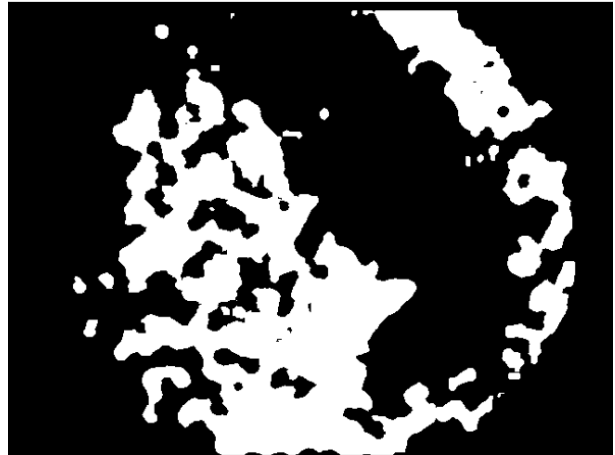


Fig. 8. Cloud mask obtained from figure 8

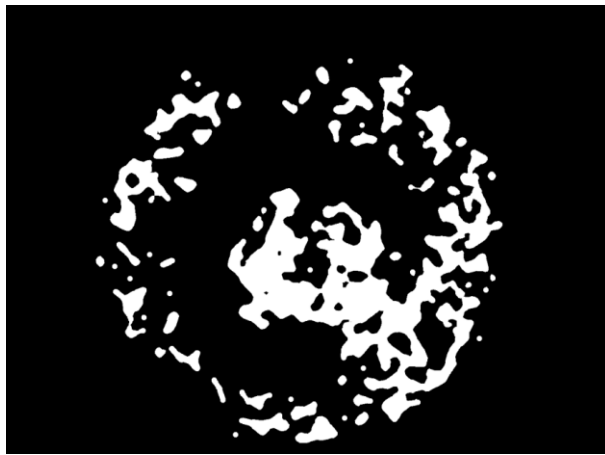


Fig. 6. Cloud mask obtained from figure 6

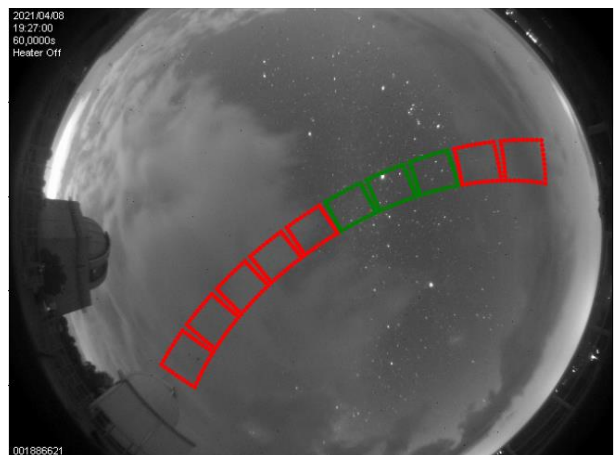


Fig. 9. Example of application of the software, in red are the zones of the sky in which clouds are detected and in green are the ones where the clear sky is detected [5].

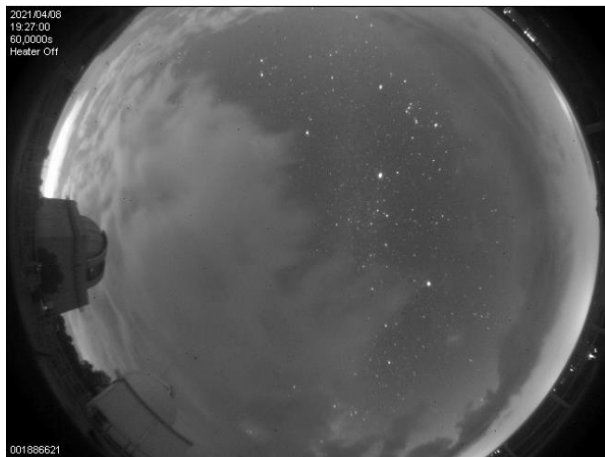


Fig. 7. Cloudy sky image from OPD [8]

6. Conclusions

This work developed CLOWN, a new tool to aid the automatic operation of a new optical double telescope installed in PASO, by removing the problem of the presence of clouds in the sky. This software has the ability to automatically, and in real-time, detect clouds' presence in the entire visible sky with the use of an all-sky camera, producing one image every 30 seconds. The software can be used to assist the double telescope to increase its efficiency in clouded conditions, as seen in figure 9, as well as other possible equipment present in the observatory or even to create a visual assistance tool for the observatory in case of manual operations.

It can be easily implemented as it only requires an all-sky camera, which is usually present in a usual observatory, and one calibration procedure. It can be used with older all-sky cameras already installed even if the camera characteristics are unknown. It has a median difference of 0.61° , improving to 0.42° if the camera

specifications are known, between found and expected coordinates. It also does not require great computing power, so it does not require the installation of additional hardware systems.

The results were very good as the software can identify the clouds in the sky in a great variety of shapes, for both visible and invisible cases, as can be seen in figures 6 and 8, and even with the interference of the moon on a big cloud. It has incorporated a mask option to minimize errors caused by permanent obstructions such as buildings on the horizon.

For future work, the images processed by this software can then be used for the training of a machine learning algorithm, which can lead to even better and faster results. Another future possibility is the use of these images for a meteorological study about cloud conditions at night in the area, or the prediction of their movement in real time.

The software was written in Python and uses many libraries available publicly [9,10].

Acknowledgments

This work is funded by FCT/MEC through national funds under the project PCIF/SSI/0103/2018. The team acknowledges further support from ENGAGE-SKA Research Infrastructure, ref. POCI-01-0145-FEDER-022217, funded by COMPETE 2020 and FCT, Portugal; exploratory project of reference IF/00498/2015 project funded by FCT, through Instituto de Telecomunicações PCIF/SSI/0103/2018; support from the Space Surveillance & Tracking through the European Commission H2020 Programme under the grant agreement 2-3SST2018-20. The team acknowledges support from the Aga Khan Development Network and the FCT for the Science and Technology Cooperation through DOPPLER - Development of PaloP knowLEdge in Radioastronomy, project number 333197717.

References

[1] Kessler, D. J., and Cour-Palais, B. G. (1978), Collision frequency of artificial satellites: The creation of a debris belt, *J. Geophys. Res.*, 83(A6), 2637-2646, doi:10.1029/JA083iA06p02637

- [2] M. Garcia, Space Debris and Human Spacecraft. May 27, 2021, https://www.nasa.gov/mission_pages/station/news/orbital_debris.html (accessed 19.08.22)
- [3] I. Sample, Mind that satellite! The mission to clean up dangerous space junk, 21 April 2022, <https://www.theguardian.com/science/2022/apr/21/mind-satellite-mission-clean-up-dangerous-space-junk-astronauts-debris>, (accessed 19.08.22)
- [4] Bruno Coelho et al. New SST Optical Sensor of Pampilhosa da Serra: studies on image processing algorithms and multi-filter characterization of Space Debris. 2021. arXiv: 2107.02315 [astro-ph.IM].
- [5] Margaret Fleck. Perspective Projection: The Wrong Imaging Model. Rel. téc. 1995
- [6] Oculus All-Sky Camera 180, Starlight Xpress CDD (2022). <https://www.sxccd.com/product/oculus-all-sky-camera-180/> (accessed 21.08.22)
- [7] Michael Poelzl. "AllSkEye". In: AllSkEye (2022). URL: [HTTP://www.allskEye.com/](http://www.allskEye.com/)
- [8] "All-sky II from "Central Climatica do Observatorio do Pico dos Dias". In: Pico dos Dias Observatory (2021). url: <http://200.131.64.207/allsky/centralclima>. HTML.
- [9] Astropy Collaboration et al. "Astropy: A community Python package for astronomy". In: 558, A33 (out. de 2013), A33. DOI: 10.1051/0004-6361/201322068. arXiv: 1307.6212 [astro-ph.IM].
- [10] Astropy Collaboration et al. "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". *Em*: 156.3, 123 (Sept. 2018), p. 123. DOI: 10.3847/1538-3881/aabc4f. arXiv: 1801.02634 [astro-ph.IM].
- [11] M. A. C. Perryman et al. "The Hipparcos Catalogue." *Em*: Astronomy and Astrophysics 500 (Jul. de 1997), pp. 501–504.
- [12] Georg Zotti et al. "The Simulated Sky: Stellarium for Cultural Astronomy Research". In: *Journal of Skyscape Archaeology* 6.2 (Mar. 2021), pp. 221–258. doi:10.1558/jsa.17822. URL: <https://journal.equinoxpub.com/JSA/article/view/17822>. <https://stellarium.org/>