

Detection of Background Stars over an Artificial Satellite Pass using Blob Detection Algorithms

André Gaudin, Richard Clare, Vishnu Anand Muruganandan, Stephen Weddell
Department of Electrical & Computer Engineering, University of Canterbury, New Zealand

ABSTRACT

Due to the large number of unresponsive satellites and space debris it can be dangerous to launch additional satellites due to the increased likelihood of collisions. It has therefore become increasingly important to identify these objects to minimise the risk to functioning satellites. Telescopic imaging of artificial satellites through the Earth's atmosphere typically results in images with faint or distorted outlines of objects. To reduce this distortion, either artificial or natural beacons can be used. The natural method uses the Point Spread Function (PSF) of natural background stars to restore images of artificial satellites in the foreground, while the artificial method uses one or more laser guide stars.

This paper considered the Laplacian of Gaussian (LoG) and Difference of Gaussian (DoG) blob detection algorithms on a central processing unit (CPU) to perform natural guide star (NGS) searches over a wide field-of-view (FoV). It was found that the DoG has improved performance over the LoG algorithm, as the DoG was able to consistently detect both focused and defocused stars, as well as a low-Earth orbit (LEO) satellite. As is reported in this paper, the DoG performed approximately 10 times faster than the LoG in all trials and had a high accuracy rate.

1. INTRODUCTION

Due to the rising number of artificial satellites in Earth's orbit, there is an increased risk of in-orbit objects colliding with each other [1]. As of July 2021, there are over 7,500 man-made satellites in space, but only around 4,500 of these are still functioning [2]. A satellite colliding with another satellite or orbiting debris results in a number of fragments, many of which might themselves collide with other orbiting satellites, resulting in further fragmentation. This cascaded fragmentation process may eventually result in a debris 'belt' around the Earth, where large clusters of debris orbit our planet making additional satellite launches dangerous. This exponential process is referred to as "Kessler Syndrome" [3] [4], and is considered a major threat to future satellite launches.

Detecting potential hazards to operational satellites is an important step in mitigating the potential damage of a collision or avoiding one altogether [5] [6]. However, an object in space viewed from Earth is distorted due to the effects of Earth's atmosphere. Therefore, it is necessary to digitally correct for the loss of quality in captured images [7]. However, a fast detection algorithm is required to accurately locate objects of interest. Our method uses a fast blob detection algorithm to find k number of stars across a wide field-of-view (FoV), in addition to satellite locations. Once the stars have been detected, adaptive optics methods can be used to correct the optical path of a telescope in real time from an individual star and determine the distortion functions for image correction [8] [9].

This paper considers two approaches for blob detection: the *Laplacian of Gaussian* and the *Difference of Gaussian* methods. Both are commonly used in a variety of applications, such as processing images from satellites, and aim to detect either convex or concave structures in a given image [10]. The algorithms used to implement these two approaches are from *scikit-image*, which provides a collection of image processing algorithms for the Python programming language [11].

2. THEORY

2.1 Background

Images of astronomical objects taken by ground-based telescopes are distorted by atmospheric turbulence [9]. To improve these images, the Point Spread Function (PSF) can be estimated using a Wavefront Sensor (WFS) [12]. A control system operating with the WFS can then calculate and apply a correction to the wavefront by using a Deformable Mirror (DM) [8]. This system is called Adaptive Optics (AO).

As an alternative to AO, deconvolution techniques can correct for the effects of Earth's atmosphere once an image has been captured. However, these methods are dependent on, in our case, defocused stars to estimate distortion functions [7]. The low probability of imaging one or more background stars within an isoplanatic patch of a low-Earth orbit (LEO) satellite makes using current techniques challenging [7]. However, wide-field tomographic images have been tested in simulation [13]. Another deconvolution method has been developed using the structural similarity metric to restore LEO images. Instead of using a WFS, the spread of a single background star is used to estimate the PSF [7], and therefore restore the image of a LEO satellite.

To improve the probability of imaging multiple background stars, a wide FoV was used to better estimate atmospheric turbulence [14]. These background stars need to be detected and located, using either a fast blob detection or a similar method, so their PSF can be approximated. It is expected that this system will be implemented on the University of Canterbury Mt John University Observatory (MJUO) Boller and Chivens (B&C) optical telescope. This would aid satellite imaging and tracking at MJUO and therefore improve the space situational awareness (SSA) capabilities of New Zealand [12].

2.2 Laplacian of Gaussian

The *Laplacian of Gaussian* (LoG) characterises bright and dark localised regions, commonly referred to as blobs in terms of local minima or maxima [15]. The derivation of the LoG operator applied to an input image $f(x,y)$, given in [16], starts with the 2-D Gaussian function

$$G(x, y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \quad (1)$$

where x and y are pixel intensity values, and σ is the standard deviation. The Laplacian operator ∇^2 can then be applied to the 2-D Gaussian function, where the Laplacian operator is defined as

$$\nabla^2 = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \quad (2)$$

The application of the Laplacian operator to the 2-D Gaussian function results in

$$\nabla^2 G(x, y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \quad (3)$$

which is the LoG operator applied to the input image [16].

The Laplacian portion highlights areas of abrupt intensity change, which is useful in edge detection applications [17]. Because the Laplacian is a second-derivative operator, it is invariant to rotation, meaning the use of multiple kernels is avoided. The LoG is commonly used to implement low-pass filters [16].

2.3 Difference of Gaussian

The *Difference of Gaussians* (DoG) method approximates the LoG by applying the Gaussian function to the image twice with slightly different standard deviations and subtracting the result of one from the other. As σ_1 and σ_2 approach each other, the closer the approximation is to the LoG. The equation for the DoG applied to an input image, as in [16],

$$G(x, y) = \frac{1}{2\pi\sigma_1^2} \exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2} \exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right), \quad (4)$$

where σ_1 and σ_2 are the minimum and maximum standard deviations respectively.

The primary advantage of the DoG over the LoG is its computational efficiency and noise reduction at the risk of information loss due to approximation [17]. Because the subtraction of one image from another preserves the information between the frequencies of the images, the DoG is essentially a bandpass filter [16]. Despite the similarities of the DoG and LoG, examples by *scikit-image* suggest the DoG may be better than the LoG for larger and dimmer blobs [11].

2.4 Methodology

The *scikit-image* Python library was used to compare the LoG and DoG algorithms. While *scikit-image* provides source code for a third blob detection method, the *Determinant of Hessian* (DoH), it was discounted as a viable option due to the length of time the algorithm takes to complete. This is presented in Section 3.1.

Both the LoG and DoG algorithms from *scikit-image* contain similar input variables. These variables were adjusted to find the optimal heuristic values for comparison. The most important input variables for this study are as follows [11]:

- *min_sigma*: the minimum standard deviation of the Gaussian kernel. This number is lowered to detect smaller blobs. Set to 20.
- *max_sigma*: the maximum standard deviation of the Gaussian kernel. This number is raised to detect larger blobs. Set to 30.
- *threshold*: the lower bound for scale-space maxima. The lower this number is, the less intense the detected blobs can be. Set to 0.05.

The LoG algorithm also contains the *num_sigma* variable which corresponds to the number of intermediate standard deviation values that are considered between *min_sigma* and *max_sigma*. DoG contains the *sigma_ratio* variable which is the ratio between σ_1 and σ_2 . The best values for *num_sigma* and *sigma_ratio* were found to be 10 and 1.6 respectively.

Both algorithms were tested on the same sets of astronomical and satellite images using an AMD Ryzen 5 3600 central processing unit (CPU). All images were captured at MJUO using a B&C optical telescope. This telescope has a diameter of 0.61m, and a focal ratio of 6.25 [7]. All images are 2048x2048 pixels, with a pixel size of 5.5 μ m, and captured with a FoV of 10 arcmins [6]. The image sets are, in order of presentation:

1. 200 defocused images of the Orion Trapezium (HIP 26220), showing three faint background stars.
2. 18 images of a pass of the International Space Station (ISS), showing just the ISS.
3. 21 images of the inactive European Space Agency (ESA) Envisat, with two background stars.

2.5 Performance Metric

Given that only a small sample of 239 on-sky images have been used to test the algorithms, the Matthews correlation coefficient (MCC) was selected to determine the accuracy of the algorithms. MCC, also known as Pearson's phi coefficient, considers four result categories to determine overall accuracy [18]. The four metrics are:

1. True positives (TP): actual positives correctly predicted by the algorithms as positives
2. False positives (FP): actual negatives incorrectly predicted by the algorithms as positives
3. True negatives (TN): actual negatives correctly predicted by the algorithms as negatives
4. False negatives (FN): actual positives incorrectly predicted by the algorithms as negatives.

In all cases of the above, a human expert was used to verify the result of each tested image.

In this study:

1. TP is when the algorithm correctly detects that a star or satellite exists in an image
2. FP is when there is no star or satellite in an image area, but the algorithm returns a detection in that area.
3. TN is when there is no star or satellite in a given area of an image, and the algorithm does not return a detection for that image portion
4. FN is when the algorithm does not detect a star or satellite in an image, when one exists.

TP and FN are calculated in terms of the number of objects across all images in a set, while FP and TN are calculated in terms of the number of images in that set. This is because returning FP results will potentially mean the B&C telescope is moved to study an object that does not actually exist. In contrast, while the return of FN is the best outcome, the failure to detect one object does not mean the algorithm failed to find the other objects in an image and does not mean the object will remain undetected in all images in the set.

The MCC is defined as follows [18]:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+TN)(TN+TP)(TN+FN)}} \quad (5)$$

The results range from -1 to +1, with -1 being the worst value (perfect misclassification) and +1 being the best value (perfect classification) [18]. This means that a value of -1 is 0% accuracy, +1 is 100% accuracy, and 0 is 50% accuracy.

3. RESULTS

3.1 Defocused Orion Trapezium Cluster

The first test case was three defocused stars of the Orion Trapezium with example results presented in Fig. 1 and Fig. 2. Additional images are presented in Appendix A. These images contain three stars that require detection. The stars as shown in Fig. 1 and Fig. 2 are, from top to bottom: Theta¹ Orionis A, C, and D, with Theta¹ Orionis C (HIP 26221) the brightest star in the image. Because there are 200 images, this means there are 600 stars that require detection.

After each algorithm has finished processing the image, the x and y position of the centre of the detected blob along with its radius r are returned. A different algorithm provided by *scikit-image* uses these values to draw a circle around the centre point of the blob.

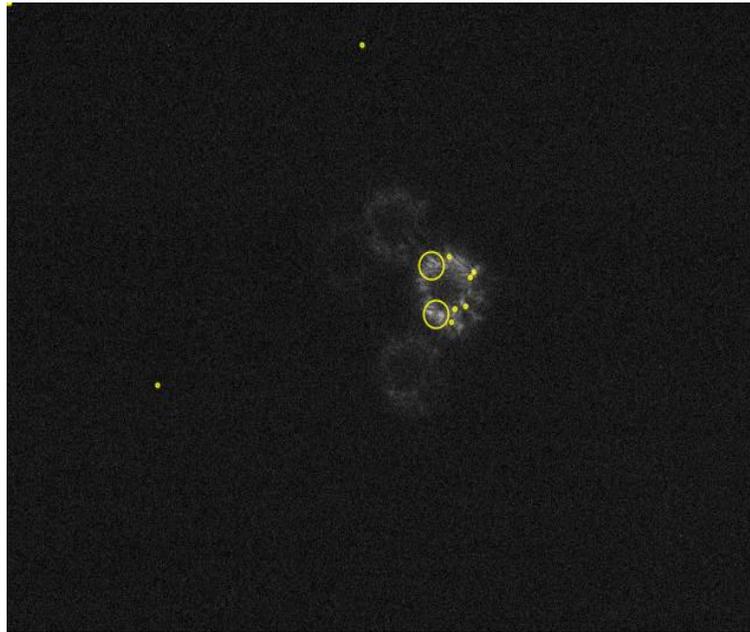


Fig. 1: LoG performance for the Trapezium case, falsely highlighting the wavefront regions, whilst the location of each encircled annular is required for a true positive classification.

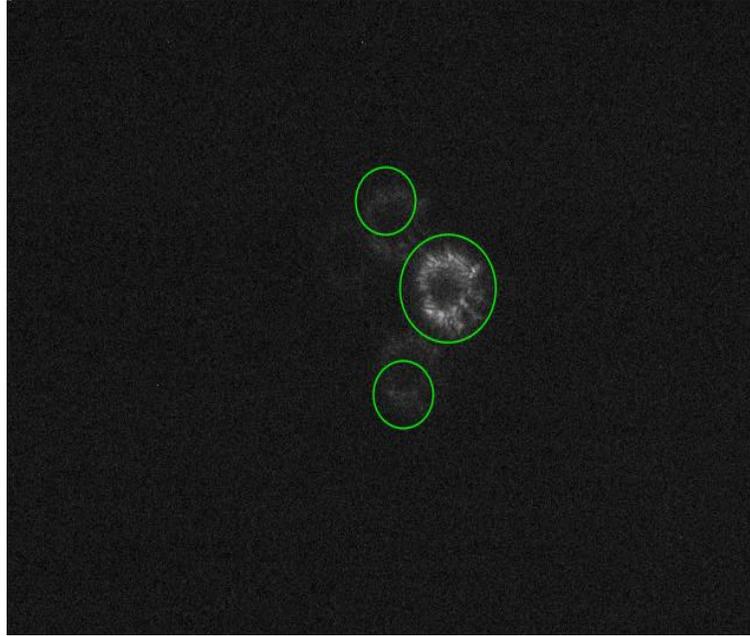


Fig. 2: DoG performance for the Trapezium case, where the defocused regions encircling each defocused source location is clearly showed

Fig. 1 shows the performance of the LoG algorithm. It was unable to consistently find HIP 26221 and was never able to detect the two other bright stars in the trapezium. The algorithm instead focused on smaller bright regions, such as wavefront regions and annular image noise. This resulted in several false positives, which is reflected in the MCC score in Table 1.

As highlighted by Fig. 2, the DoG algorithm had far better performance than the LoG. The DoG algorithm was able to detect the three defocused stars. The best DoG performance was on the defocused Theta¹ Orionis C, as the circle encapsulates the star in its entirety. The other two stars are partially circled, which is sufficient for the DoG algorithm to be used with a multi-object wavefront sensor to estimate and remove low order aberrations for corrections using a tip-tilt mirror [9], and with post deconvolution from wavefront sensing processing [12] [14].

The average time it took for the algorithms to complete each image is shown in Table 1. The DoG algorithm was 10 times faster than the LoG algorithm. Subsequentlym that the DoH algorithm mention in Section 2.4 was discounted as a viable alternative, since the time taken to process even one image was far too slow. The FP value is not recorded in our results as there were simply too many FP to count, especially considering the large run time.

Out of the 200 images tested, there were 56 cases were the DoG algorithm failed to find Theta¹ Orionis A (HIP 26220). This means that of the 600 stars detected, there were 544 TP cases. There were no FP cases. Based on Eq. 5, this gave an MCC value of 0.84. Examples of failures are presented in Appendix A. The detection failure is a result of the *threshold* value being set sufficiently high to filter out the star. Lowering this value however results in more FP cases.

Table 1: Time taken in seconds to complete 200 images and MCC for each algorithm.

Algorithm	Time per image (s)	TP	FP	TN	FN	MCC Value
LoG	0.99	128	492	0	472	-0.79
DoG	0.11	544	0	200	56	0.84
DoH	242.72	0	-	0	600	-1

3.2 International Space Station Pass

The second test case was a set of images of a pass of the International Space Station (ISS). Because the MJUO does not have a satellite tracking system, the B&C telescope was centrally pointed at a star (HIP 58905) where $M_v = 5.0$, and the ISS was imaged as it moved over this FoV [7]. Due to the short exposure times used to minimise motion blur this star is not clearly visible in the test images so the performance of the algorithms on an artificial satellite by itself could be tested. Example frames from the test are shown in Figs. 3 and 4, which show the ISS and the circles drawn by the LoG and DoG algorithms.



Fig. 3: LoG performance on an ISS pass, showing the detection of the ISS.



Fig. 4: DoG performance on an ISS pass, showing the detection of the ISS

Both algorithms were able to detect the ISS as it passed through the FoV with 100% accuracy. Similar to the Orion Nebula case, the DoG was approximately 10 times faster than the LoG algorithm, which is presented in Table 2. Additional images, including the images of the start and end of the pass, are presented in Appendix B.

Table 2: Time taken in seconds and MCC for the ISS test case

Algorithm	Time per image (s)	TP	FP	TN	FN	MCC Value
LoG	14.105	18	0	18	0	1
DoG	1.475	18	0	18	0	1

3.3 Satellite with Background Stars

The final set of images tested were of the ESA Envisat with two (one much brighter) background stars. Maximum detection of stars and satellite would give 63 total objects across the 21 images. To capture these images, the B&C telescope was positioned to capture ν -Centauri (HIP 70069) where $M_v = 4.3$. This was to test the effectiveness of the algorithms when detecting background stars for image restoration. Examples images are presented in Fig. 5 and Fig. 6, with additional images available in Appendix C.

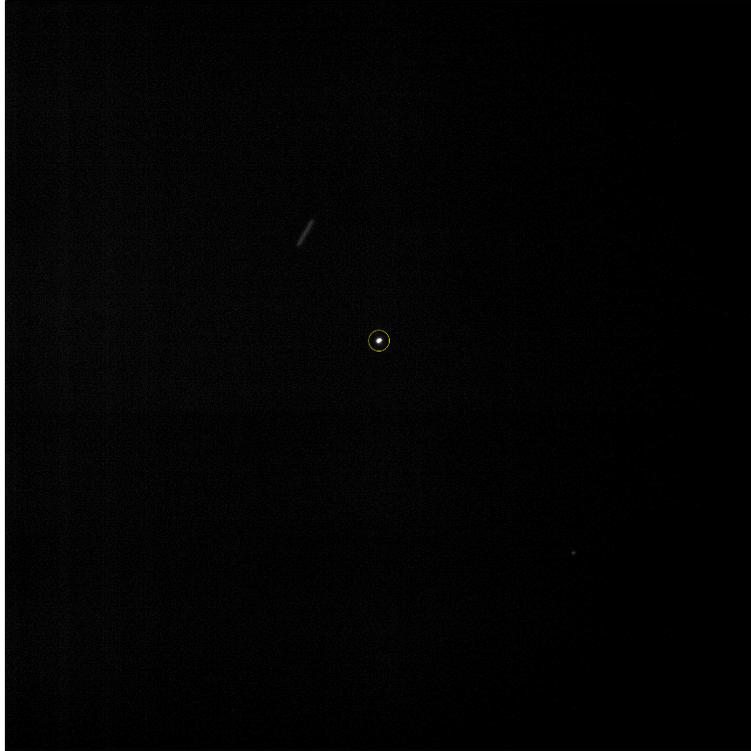


Fig. 5: LoG performance on a satellite with two background stars, where the brightest star in the image is detected but the satellite (blurred) region and second fainter star region were undetected.

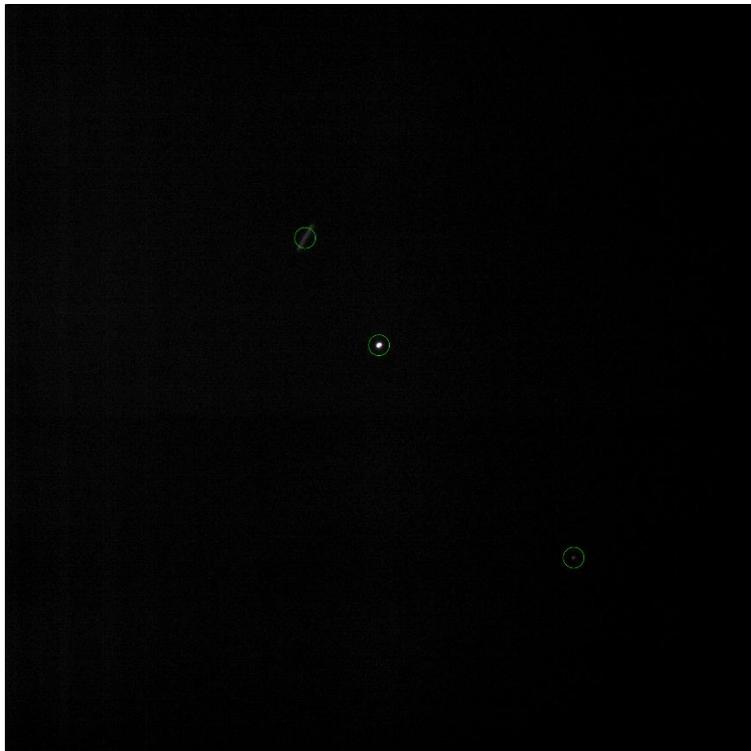


Fig. 6: DoG performance on a satellite with two background stars, showing a true positive result for both stars, varying by at least two magnitudes, and blurred satellite detection.

While the LoG algorithm returned no false positives it was only able to consistently detect the centre bright star while only occasionally detecting the satellite in the top-left position. Of the 21 images tested, the satellite was detected twice. The faint star in the bottom-right quadrant was never detected by the LoG. This is reflected in the LoG MCC value shown in Table 3.

In comparison, the DoG was able to consistently detect all three objects with 100% accuracy; an example is shown in Fig 6. The DoG was approximately 10 times faster than the LoG, similar to the other cases.

Table 3: Time taken in seconds and MCC for the joint case

Algorithm	Time per image (s)	TP	FP	TN	FN	MCC Value
LoG	13.74	23	0	21	40	0.35
DoG	1.479	63	0	63	0	1

4. CONCLUSION AND FUTURE WORK

Of the two algorithms tested, only the DoG was able to consistently detect the stars and satellites. In contrast, the LoG was able to find the brightest areas in the FoV, which would not always be the entire star. This can be seen in Fig. 1, where the algorithm returned the brightest portions of the central star rather than the entire star itself. In Fig. 5, the LoG algorithm successfully detects the brightest star, which is approximately the same size as the ISS in Fig. 3 and the brightest blobs in Fig. 1. This shows that the LoG is well suited to only detecting small, bright blobs in a given image, and therefore the LoG is unsuitable for general detection of stars.

In contrast to the LoG, the DoG was able to consistently detect both faint and bright objects, including the defocused stars in Fig. 2. This means the DoG may be a viable candidate for NGS detection. However, while the DoG consistently performed approximately 10 times faster than the LoG, the time taken for the DoG to complete a single image must be drastically reduced if it is to be used as a real-time detection method. Results may improve when using a graphics processing unit (GPU) instead of a CPU.

Future work will investigate the DoG algorithm's performance on a graphics processing unit (GPU) and compare its performance to other detection algorithms, such as solving the maximum sum problem [19]. An example of a DoG false negative is presented in Appendix A. It is expected that this method will improve the results from images of satellites by employing a fast and accurate detection algorithm for classification of natural source beacons.

ACKNOWLEDGMENTS

The authors would like to acknowledge and thank the staff at Mt. John University Observatory. This research was supported by the Marsden Fund Contract (MFP UOC1803), administered by the Royal Society of New Zealand.

5. REFERENCES

- [1] P. Bernat, "Orbital Satellite Constellations and the Growing Threat of Kessler Syndrome in the Lower Earth Orbit," *Safety of Engineering and Astropogenic Objects*, vol. 4, 2020.
- [2] European Space Agency, "Space debris by the numbers," 9 July 2021. [Online]. Available: https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers. [Accessed 12 August 2021].
- [3] D. J. Kessler and B. G. Cour-Palais, "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt," *Journal of Geophysical Research*, vol. 83, no. A6, pp. 2637-2646, 1978.
- [4] D. J. Kessler, N. L. Johnson, J. C. Liou and M. Matney, "The Kessler Syndrome: Implications to Future Space Operations," *Rocky Mountain Guidance and Control Conference*, Breckenridge, 2010.
- [5] D. Mehrholz, L. Leushacke, W. Flury, R. Jehn, H. Klinkrad and M. Landgraf, "Detecting, Tracking, and Imaging Space Debris," *ESA Bulletin*, vol. 109, pp. 128-134, 2002.
- [6] V. A. Muruganandan, R. Clare, A. Lambert and S. Weddell, "Spatial Characteristics of Functional and Non-functional Satellites in Low Earth Orbit," *Proc. 8th European Conference on Space Debris*, 2021.
- [7] S. Hickman, V. A. Muruganandan, S. Weddell and R. Clare, "Image Metrics for Deconvolution of Satellites in Low Earth Orbit," *35th International Conference on Image and Vision Computing New Zealand (ICVNZ)*, pp. 1-6, 2020.
- [8] R. K. Tyson, *Introduction to Adaptive Optics*, Bellingham: SPIE, 2000, pp. 63-80.
- [9] J. Liu, V. A. Muruganandan, R. Clare, M. C. R. Trujillo and S. J. Weddell, "A Tip-Tilt Mirror Control System for Partial Image Correction at UC Mount John Observatory," *2020 35th International Conference on Image and Vision Computing New Zealand (ICVNZ)*, pp. 1-6, 2020.
- [10] C. D. Athanassas, A. Vaiopoulos, P. Kolokoussis and D. Argialas, "Remote Sensing of Mars: Detection of Impact Craters on the Mars Global Surveyor DTM by Integrating Edge- and Region-Based Algorithms," *Earth Moon Planets*, vol. 121, pp. 59-72, 2018.
- [11] S. v. d. Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart and T. Yu, *scikit-image: Image processing in Python.*, 2014.
- [12] S. Hickman, S. J. Weddell and R. M. Clare, "Image Correction with Curvature and Geometric Wavefront Sensors in Simulation and On-sky," *ICVNZ*, 2019.
- [13] S. J. Weddell and P. J. Bones, "Wavefront prediction with reservoir computing for minimizing the effects of angular anisoplanatism," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 7140-7151, 2008.
- [14] S. J. Weddell, R. M. Clare and A. Lambert, "Near Earth Object Image Restoration with Multi-Object Adaptive Optics," *Proc. 1st NEO and Debris Detection Conference*, pp. 22-24, 2019.
- [15] T. Lindeberg, "Scale Selection Properties of Generalized Scale-Space Interest Point Detectors.," *J Math Imaging Vis*, vol. 46, pp. 177-210, 2013.
- [16] R. C. Gonzalez and R. E. Woods, "Image Segmentation," in *Digital Image Processing*, Pearson Education Limited, 2018, pp. 724-729.
- [17] H. Kong, H. C. Akakin and S. E. Sarma, "A Generalized Laplacian of Gaussian Filter for Blob Detection and Its Applications," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1719-1733, 2013.
- [18] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation.," *BMC Genomics*, vol. 21, no. 6, 2020.
- [19] S. E. Bae, "Sequential and Parallel Algorithms for the Generalized Maximum Subarray Problem," University of Canterbury, 2007.

APPENDIX A

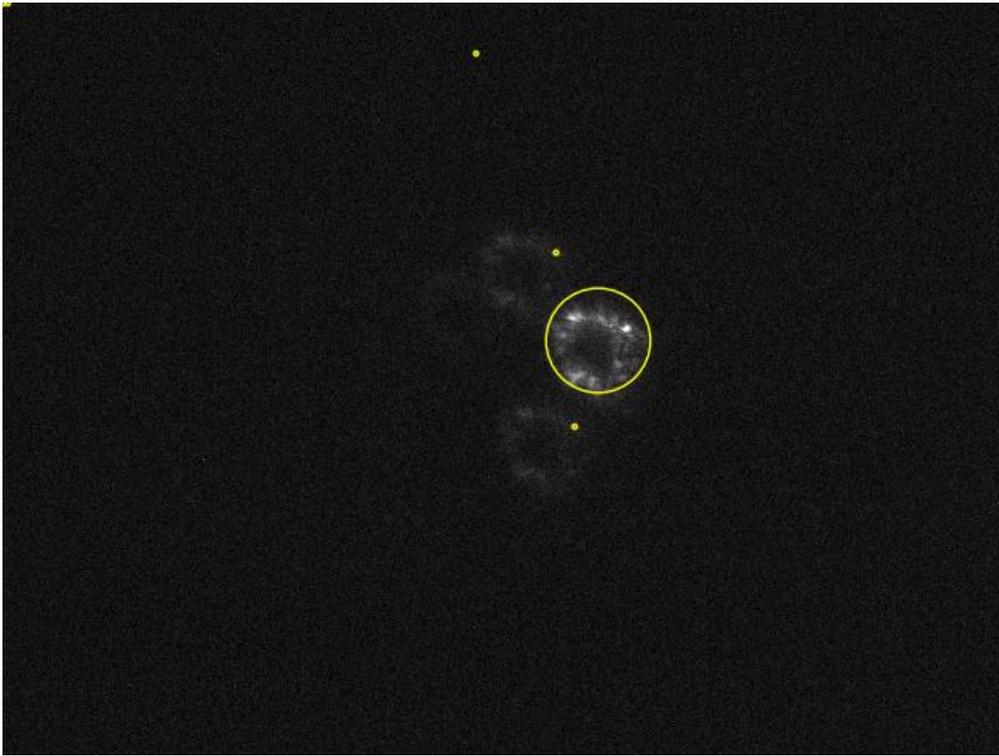


Figure 7: The LoG algorithm can detect the brightest defocused star, but also returns false positives

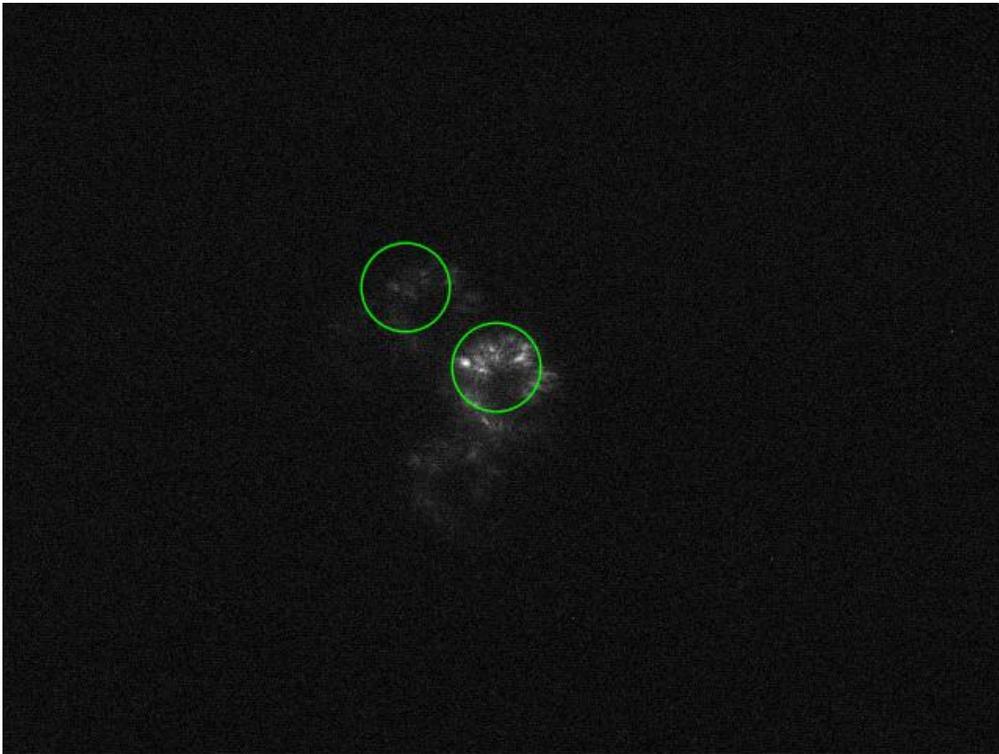


Figure 8: The DoG algorithm occasionally returns a false negative

APPENDIX B

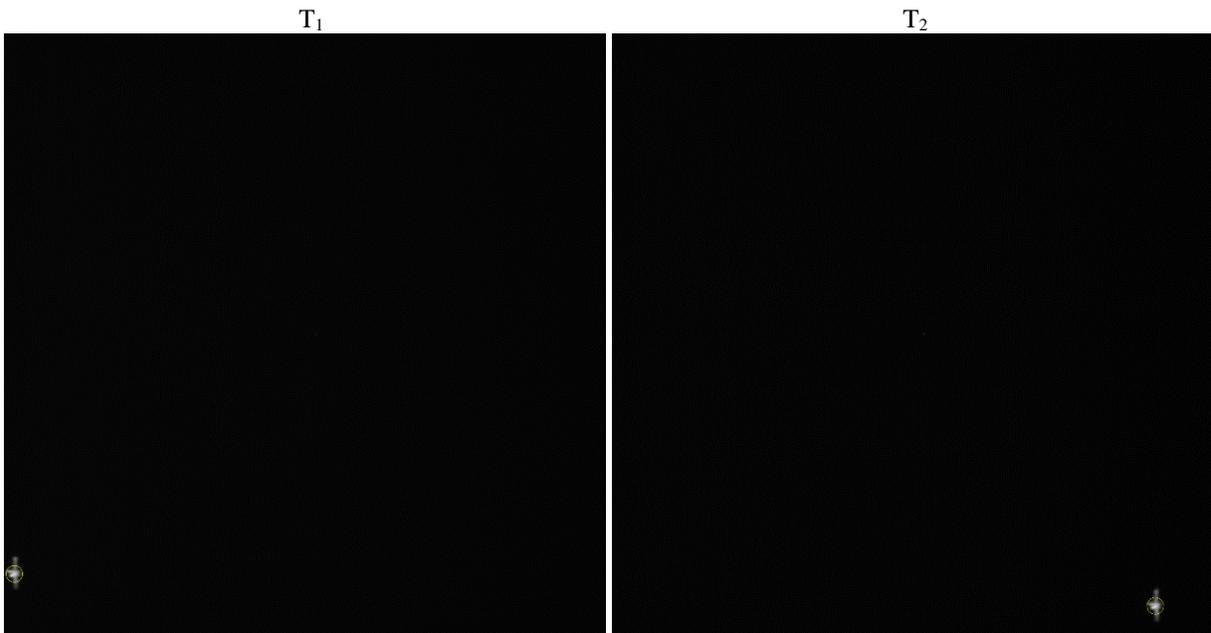


Figure 9: The beginning (T_1) and end (T_2) of the ISS pass, showing detection by the LoG algorithm

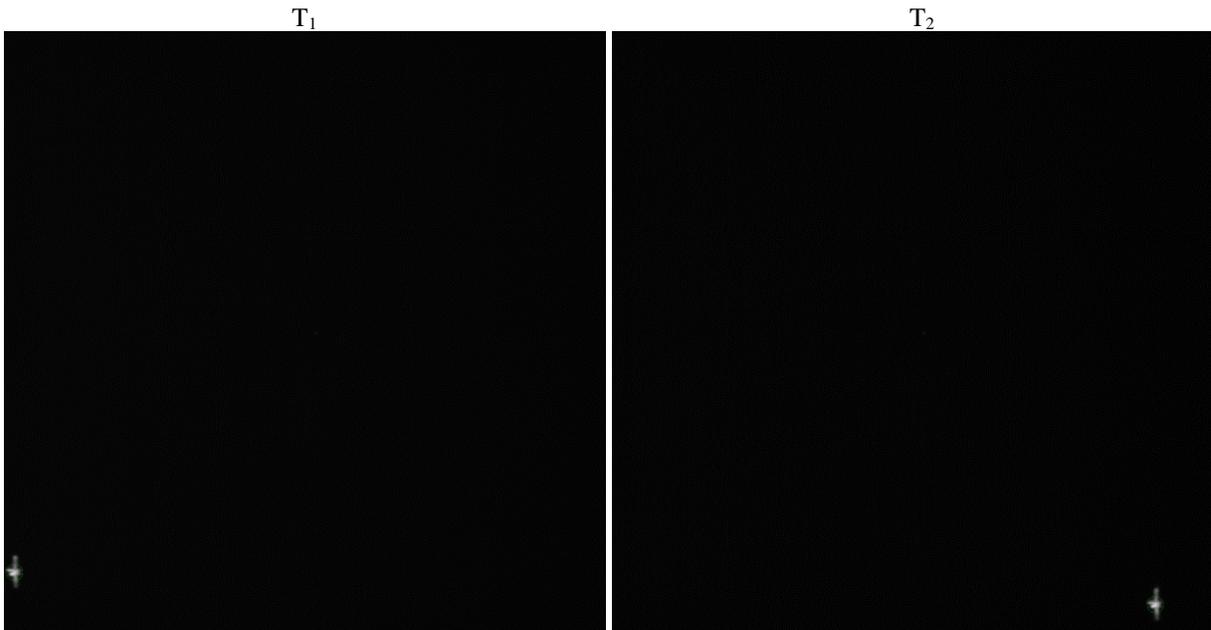


Figure 10: Beginning (T_1) and end (T_2) of the ISS pass, with detection by the DoG algorithm shown

APPENDIX C

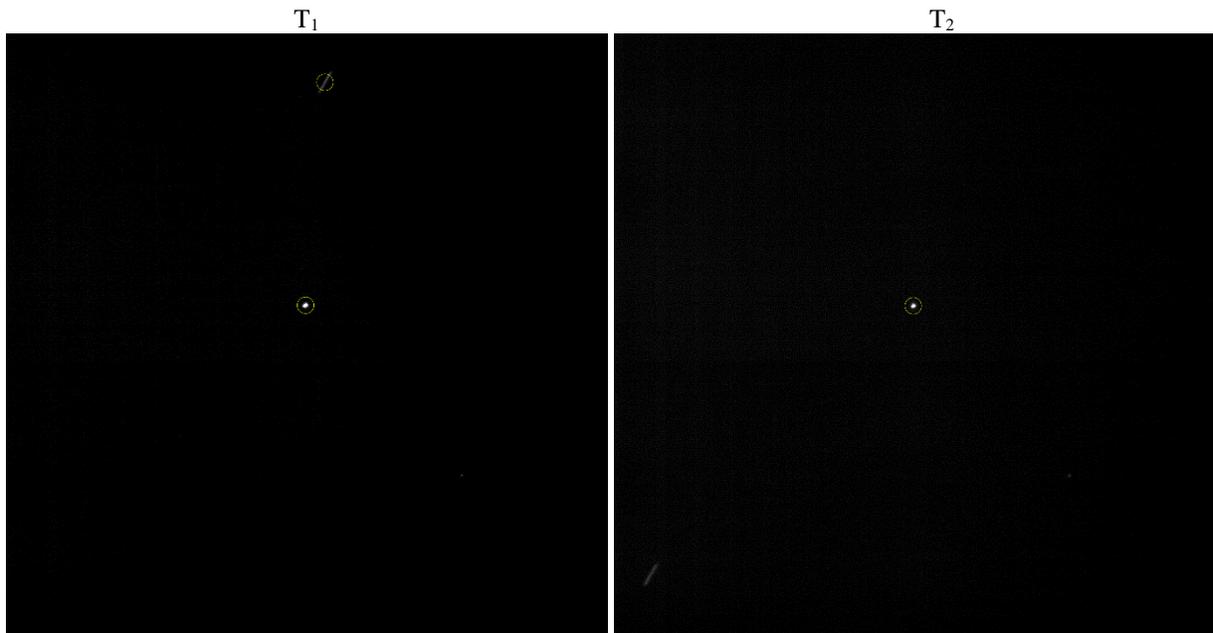


Figure 11: Beginning and end of the Envisat satellite pass with LoG algorithm results. The Envisat has been detected in the left image, taken at time T_1 , but not the right, taken at time T_2 .

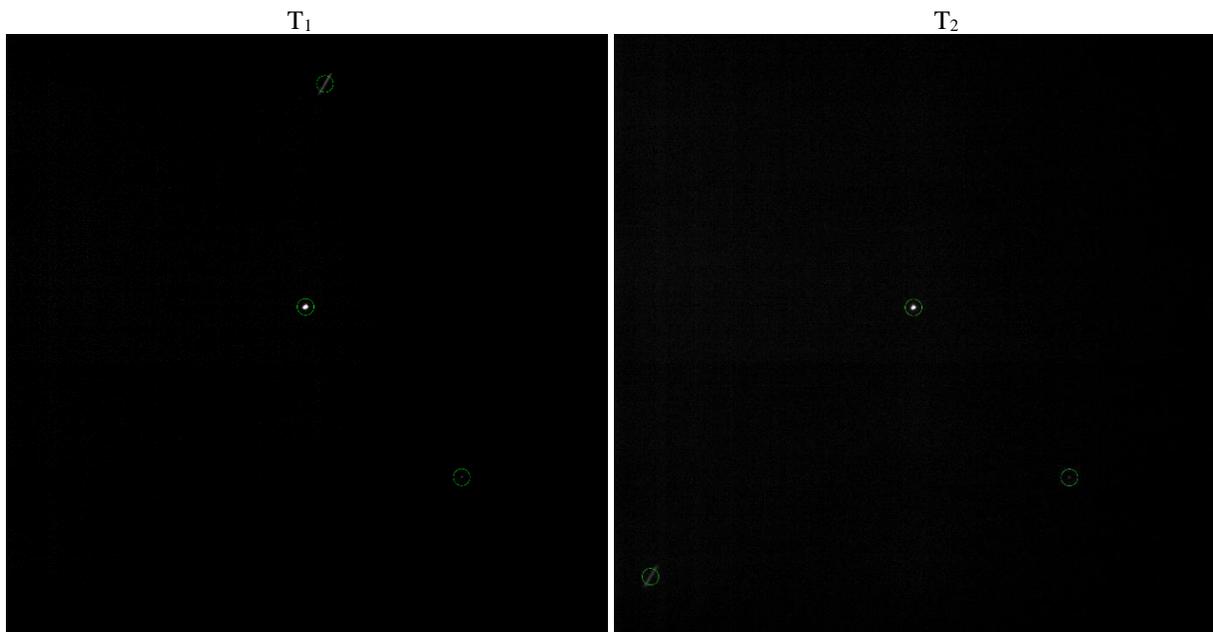


Figure 12: Beginning at time T_1 and end at T_2 of the Envisat satellite pass with DoG algorithm results. All three objects have been detected in both image sequences.