Stroboscopic Method for Verification and Adjustment of the Timestamp in Occultations

Javier de Elías · Agrupación Astronómica de Madrid (AAM) · Madrid · Spain · javierdeelias@gmail.com

ABSTRACT: This article shows a method for determining the timestamp accuracy of astronomical image captures. The method analyses the deviation between the timestamp associated with the images and the time given by a GPS PPS signal, achieving a sub-millisecond accuracy. The method has also been used to characterise the delays introduced by the readout time per line in most CMOS cameras due to the rolling shutter, which means that the position of the object to be measured in the image must be considered in order to determine the correct timestamp. By applying the method, we can characterise which offset to apply to our occultations and improve the accuracy of our measurements.

Introduction

Certain astronomical observations require a precise timestamp to be associated with them, such as the photometry of occultations, whether of stars by the moon or by asteroids. For some measurements, for example, when dealing with fast asteroids, it may be necessary to have an accuracy of better than one millisecond at the timestamp [1].

Basically, there are two options to mark the timestamp in each image: either the camera itself can do it, as it has an integrated GPS, e.g. QHY174M-GPS [2], or the camera control program will do it from the computer time, once the image has been received. In the first case, very precise timestamps are obtained, but in the case of cameras without GPS, important errors can be made, for example due to the time synchronisation method of the computer or the download time of the image from the camera to the computer. A special case is the delay introduced by CMOS sensors with rolling shutter in which the position of the object to be measured in the image must be considered and a correction related to the readout time per line has to be applied. It is precisely the presence of these errors, which can be tens of milliseconds, that justifies the need for a verification and correction method such as the one described in this article.

One of the most used time references today is the PPS (Pulse Per Second) signal obtained from the GPS (Global Positioning System) signal. This time reference is highly accurate, in the order of microseconds [3] and can serve as the basis for the generation of light pulses with very precise timing. One example is the SEXTA (Southern EXposure Timing Array) device [4] which uses a panel of LEDs that are activated sequentially so that when an image is recorded, the last LED to come on allows the time to be determined with an accuracy of 2 ms. Another option is the NEXTA (New EXposure Timing Analyser) device [5] which also uses LEDs but implements the improvement of encoding the time with an accuracy of 0.1 ms. Both devices have the disadvantage that they require focusing of the image of the LEDs on the sensor, so they are a good option to verify the precision of the setup, but it is complicated and impractical to use them to know the precision at the moment of the astronomical observation, since it would be necessary to point the panel of LEDs and adapt the focus, or use mirrors and optical elements to avoid having to manipulate the telescope in the moments before and after the astronomical observation. Another difficulty with these methods is that the precise verification process requires very short exposure times that are not possible with certain cameras.

The method described in this article also uses light pulses synchronised with the PPS signal, but with a simpler approach, as it uses a single LED and its use requires only placing the LED at the entrance of the telescope. Construction is much simpler, and there are even commercial devices that generate the necessary pulses. Moreover, it is not only a verification system, but can also be used just before and after the astronomical observation to check the deviation and correct the timestamp. The verification and correction method works with both fast and slow cameras and does not require short exposures to perform the verification. Sub-millisecond accuracy is achieved in the measurement of the deviation from the time reference set by a PPS signal.

Instrumentation and Method

Instrumentation

Two cameras, the ASI183MM Pro [6] and ASI120MM mini [7], have been used for testing the verification and correction method.

Both are monochrome cameras with CMOS sensors, but the method is also valid for cameras with CCD sensors. The ASI183MM Pro camera has a USB 3.0 interface, and the ASI120MM mini camera has a USB 2.0 interface, and they have been selected for

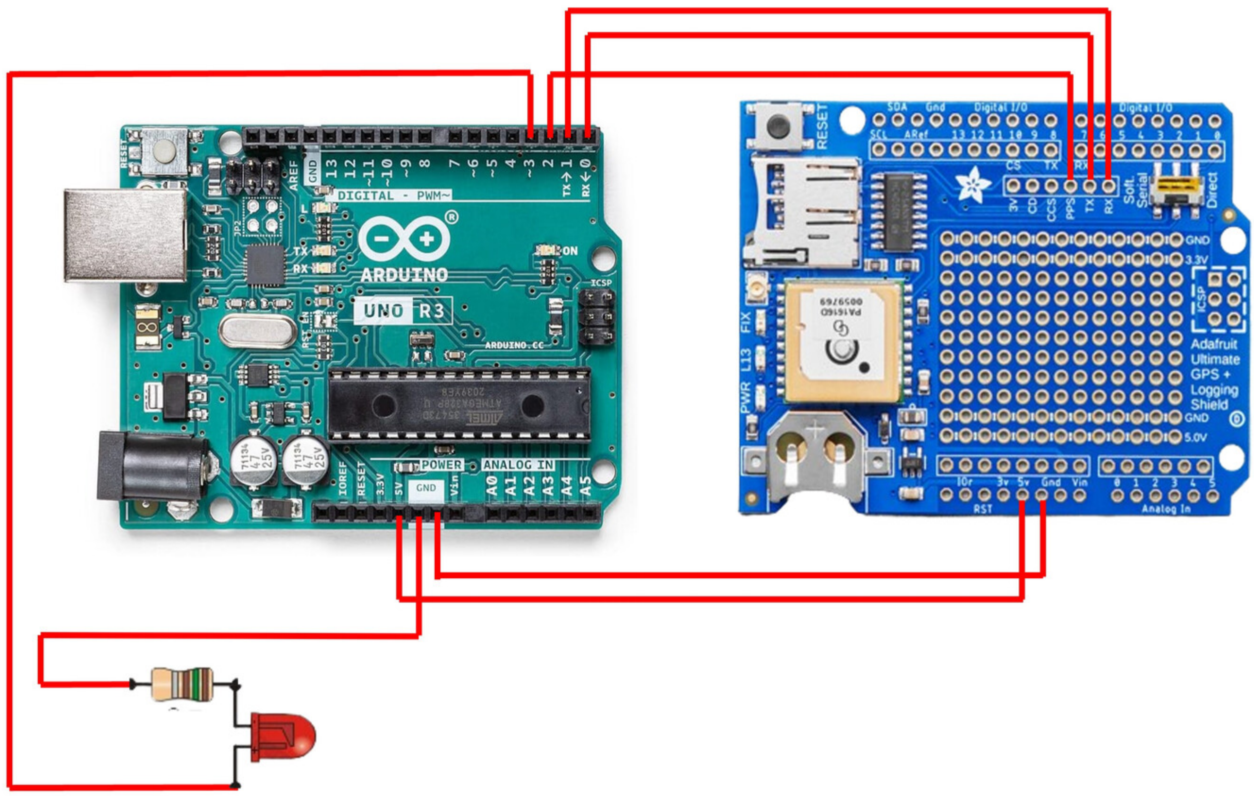

Figure 1. Connection diagram between the Arduino Uno controller and the Arduino Ultimate GPS Logger.

testing precisely because they are cameras with very different fps (frames per second).

For the generation of the light pulses synchronised with the GPS PPS, the following elements have been used:

- Adafruit Ultimate GPS Logger Shield [8]. This GPS receiver incorporates a PPS digital output necessary for the implementation of the method.
- Arduino Uno [9]. This micro controller receives the PPS pulse via a digital input and by programming (the source is included in the Annexes) an interrupt triggered by the rising edge of the PPS signal, a pulse of 500 ms duration is generated. Additionally, the RX and TX signals of the GPS receiver can be connected to receive NMEA 0183 type messages, but this is not necessary for the purpose of the application.
- An LED connected to a digital output of the Arduino controller with a resistor in series.

In practical applications of this system, the LED can be integrated into a translucent methacrylate ball and positioned at the entrance of the telescope. In the system tests no telescope was required and the LED was placed directly in front of the camera.

The device fabricated to generate the light pulses is shown in the Figure 2. To use the method described with this device, it is not necessary to implement the display and the box shown in the picture, it is sufficient to implement the schematic in Figure 1.

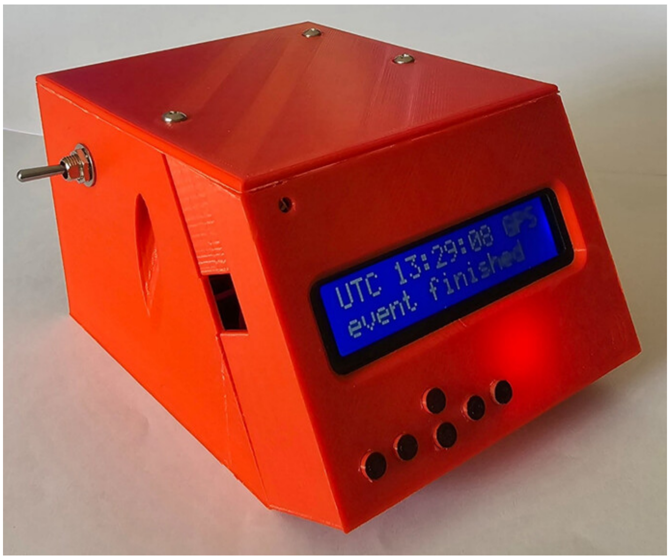

Figure 2. Device to generate light pulses.

Instead of manufacturing a device for the generation of the 500 ms pulse synchronised with PPS, it is possible to use other devices such as the GARMIN 18xLVC [10] which allows the pulse generation to be configured with its software.

Since the method aims to verify the timestamp accurately, it was decided to measure the delay that the assembly generates between the PPS pulse and the flash of light. For this measurement, a Thorlabs photometer model PDA10A [11] and a Keysight oscilloscope model DSOX3024T [12] were used, both with sufficient bandwidth to be able to carry out the measurement in the microsecond range. Two probes were used in the oscilloscope, one connected to the PPS output of the GPS and the other to the output of the photometer.

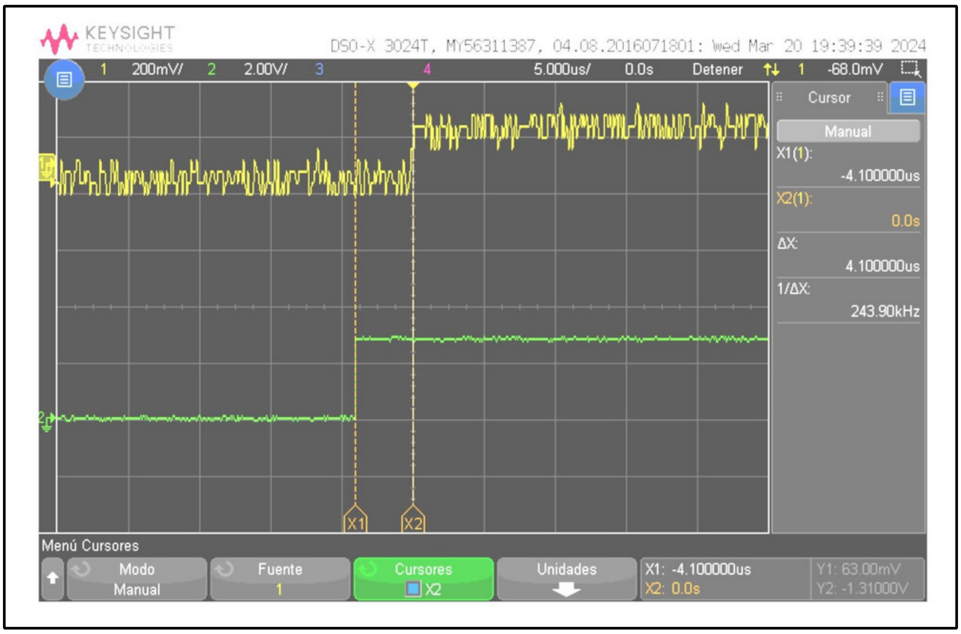

Figure 3. Measured delay between PPS signal and LED flashes. Green shows the pulse generated by the GPS and yellow shows the signal recorded by the photometer measuring the LED flashes.

Figure 3 shows the delay of approximately 4 microseconds, so it can be concluded that the setup provides a light pulse well synchronised with the PPS signal.

The computer used for testing is a laptop with 12th Gen Intel(R) Core (TM) i7-12700H 2.70 GHz processor and 32GB RAM, with *Windows 11*. For synchronisation the NTP service installed with the *Meinberg NTP* package [13] has been used. In the local network a Raspberry-based Stratum 1 time server with GPS PPS [14] has been used verifying throughout the tests that this host is selected for synchronisation and the PPS signal is in use.

The PC's time, synchronised with NTP and using the Stratum 1 server, will be used as a time reference to compare the timestamp measurements. The offset from the time server applied by the NTP service has been monitored during the test and is shown in Figure 4 for the interval in which the measurements were made with the camera ASI183MM Pro and Figure 5 with the camera ASI120MM mini. The average offset is 0.15 ms and 0.09 ms respectively, so during the tests the equipment was synchronised with an accuracy of better than 1 ms.

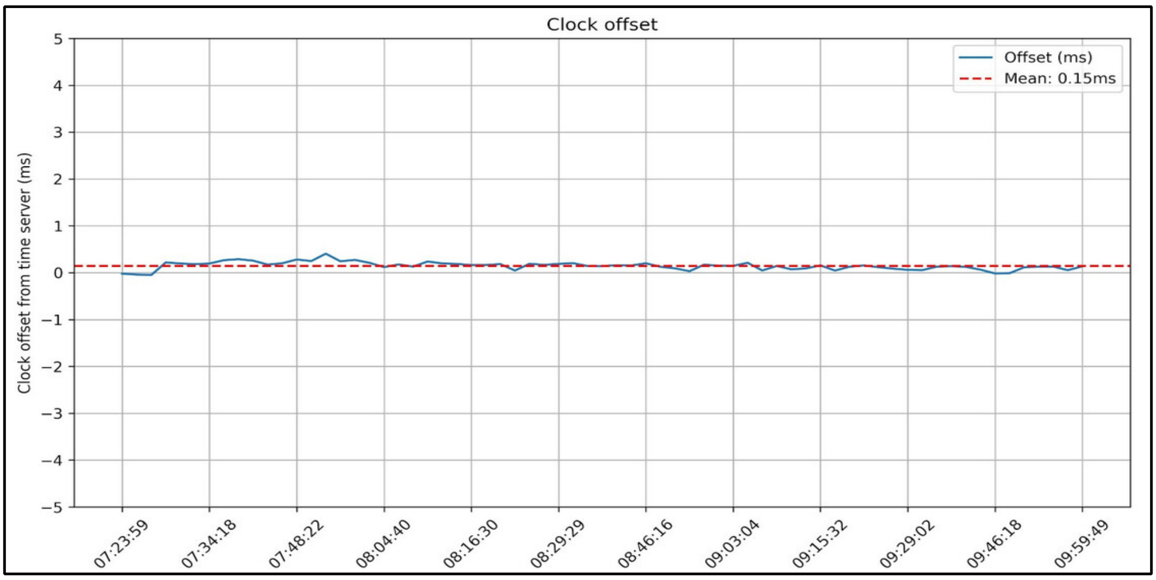

Figure 4. Offset from time server in the interval in which the tests with the camera ASI183MM Pro were carried out.

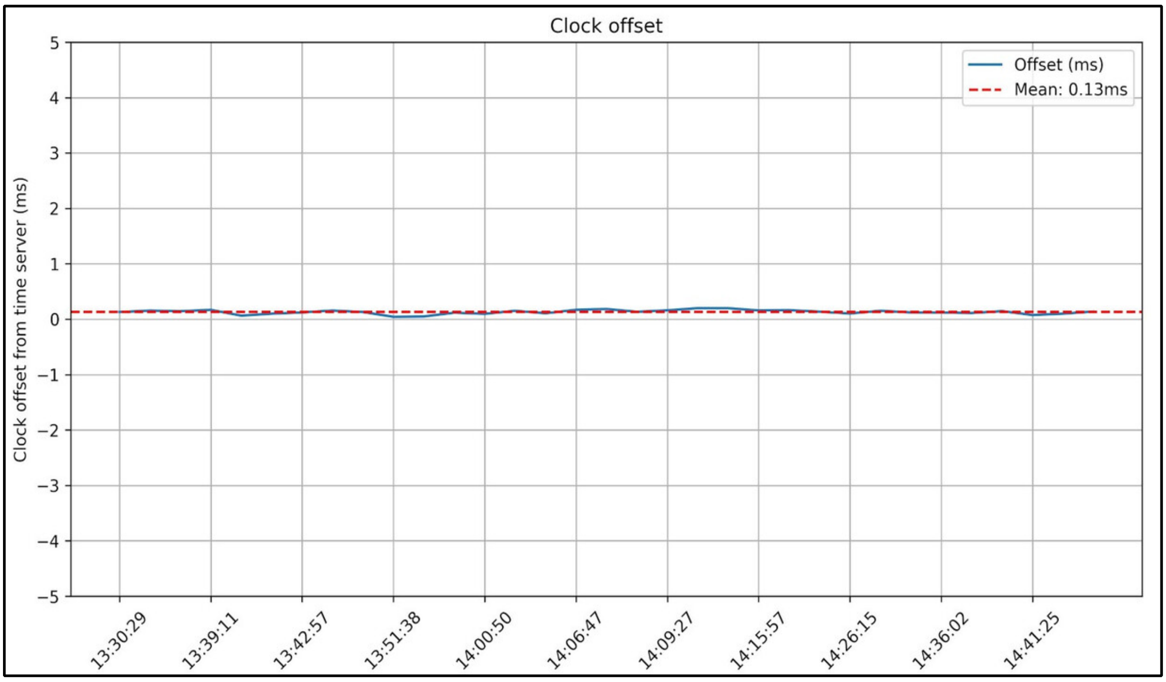

Figure 5. Offset from time server in the interval in which the tests with the camera ASI120MM mini were carried out.

The *SharpCap* software [15] in version V4.1.11824 was used to control and download the images. This software performs the time stamping of the images using the time of the computer as a reference when receiving the images and subtracting the exposure time. The tests were carried out by configuring *SharpCap* to generate a sequence of .fit files.

For the analysis of the generated images, the *Tangra* software [16] in its V3.7.5 version was used. This software allows the generation of a light curve in CSV format (comma separated values) from which the difference between the time according to the PPS signal and the timestamp of the images is calculated with the procedure explained in the following section. Another recommended tool for such an analysis is the software *PyMovie/PyOTE* [17].

Method

Principle of operation (stroboscopic effect)

The developed method consists of generating and recording light flashes with a duration of 500 ms, whose rising edge is synchronised with the PPS signal obtained by a GPS receiver. Using the setup described in the instrumentation section, the Arduino controller activates an output connected to an LED every time it detects the rising edge of the PPS signal. The LED therefore generates an illumination pattern so that every second the first half of 500 ms is on and the second half of 500 ms is off. The light signal thus described is recorded by the camera with an exposure time slightly longer than that of the light pulse. An exposure time of 505 ms, i.e. a time offset of 5 ms, has been selected to generate a stroboscopic effect.

The camera will start recording at an arbitrary time which, in general, will not be synchronised with the on/off pattern that occurs every 500 ms. It is precisely the difference between the flash duration and the exposure time that is decisive for the intensity level of each frame to vary from shot to shot. Figure 6 shows how this works.

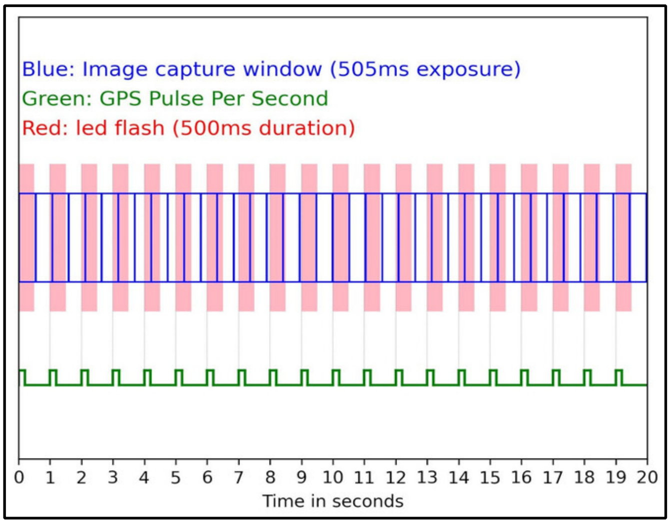

Figure 6. Principle of operation. The 5 ms offset is not scaled.

In the captured image sequence, the odd and even images are analysed independently, and it is observed that in each of them the average intensity will go up from a minimum in which the image coincides with the part of the second in which there is no flash, to a maximum in which the image coincides with the opposite case, that is, synchronised with the flash. The odd and even images are complementary, when one has a maximum, the other has a minimum and the slopes of the curves showing the evolution of the intensity have opposite signs. As the time offset progresses during the capture, in the general case there will not be perfect synchronization at any time, but the method will allow the maximum and minimum to be calculated accurately.

Figure 7 shows a typical light curve obtained by this method (the image is a result of the *Python* program described in this document), in which the light curves of the odd and even images have been differentiated. The curves will consist of straight sections as long as the LED intensity stays within the linearity zone of the sensor (no saturation). Both curves are treated identically in the processing, but independently of each other. The objective is to determine the maximum and minimum for each of the curves. We will see that each maximum or minimum allows us to calculate a deviation in the timestamp and then we will average all the calculated deviations.

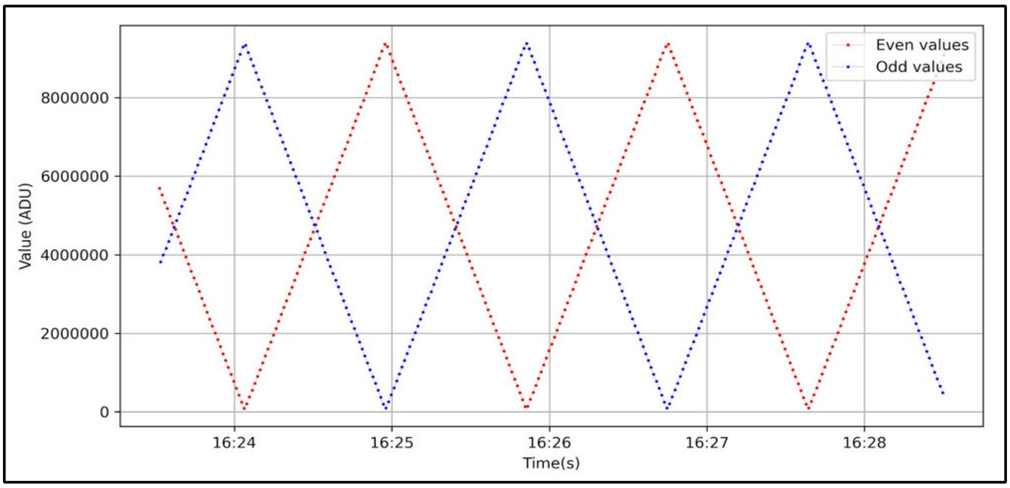

Figure 7. Example of a light curve obtained.

To estimate the time between maxima or minima of an even or odd curve, we must take into account that for it to produce a maximum or minimum again, we have to generate a displacement of the observation window of 1 s. In the example given for an exposure duration of 505 ms, as each second produces a displacement of 10ms (corresponding to 2 images), we need 1000/10=100 s to elapse. If the exposure time is 5 minutes, we will obtain 3 maximums and 3 minimums for the curve of even and the same for the curve of odd images, so we will be able to average with 12 measurements.

To accurately calculate the maxima and minima of the light curves, N points are determined on either side of each approximate maximum or minimum found. For each set of N points, a regression line is calculated, and the maximum or minimum is recorded as the intersection of the two lines.

The value of N, the number of points for the linear regressions, has been set to a minimum of 10 during the tests. The minimum of 10 means that in some cases the calculation cannot be performed (depending on the distance of the maximum or minimum from the beginning or end of the capture), so that the total number of values to be averaged in 5 minutes and with 505 ms exposure can sometimes be one or two units lower than 12. Although 5 minutes is considered an adequate interval to obtain sufficient measurements, it can be extended to obtain more measurements to improve accuracy.

Figures 8 and 9 (the images are a result of the *Python* program described in this document) show graphically the procedure of obtaining a maximum and a minimum respectively by calculating the intersection point.

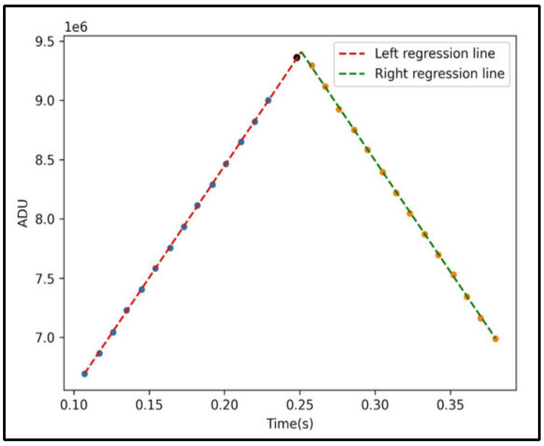

Figure 8. Obtaining a maximum as the intersection of the regression lines.

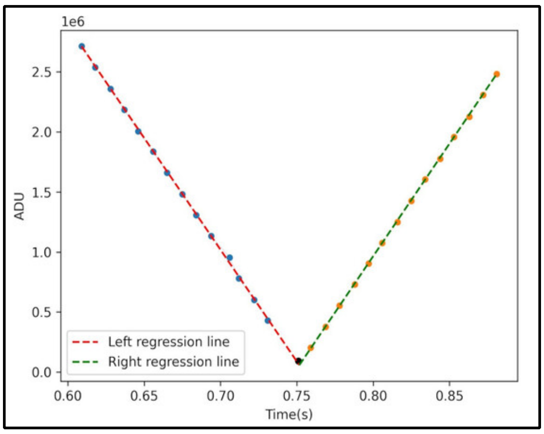

Figure 9. Obtaining a minimum as the intersection of the regression lines.

Light Curve and Processing

In the tests carried out, *SharpCap* has been configured to record a sequence of files in FITS format. The analysis has been carried out with the *Tangra* software, oriented to the processing of occultation captures. For this purpose, the "no tracking" mode was selected choosing an area near the centre of the image for the photometry and using only one aperture. Once the light curve is obtained, it will be exported to a csv file. The appearance of the light curve provided by *Tangra* may not show the pattern of maxima and minima as *Tangra* subtracts the background which will not be done when processing, as only the signal of the photometric circle will be used (the csv file generated by *Tangra* offers signal and background in differentiated columns).

The parameters in Figure 10 have been configured for reductions with *Tangra.*

The stroboscopic method generates maxima and minima of the light curve which in perfect synchronisation should coincide with the first half of the second (i.e. a timestamp value of the type xx.250 s) or the next half of the second (i.e. a timestamp value of the type xx.750 s). The deviation is measured by subtracting the value of the maximum or minimum of the time closest to xx.250 s or xx.750 s, i.e. we only subtract the milliseconds ignoring the seconds. This allows us to detect variations of at most 500 ms, so this method can only be applied for fine-tuned detections of timestamp deviations.

For the automatic analysis of the light curve, a *Python* program (Included in the Annexes) has been developed that separates the measurements into two dataframes: one for the even and one for the odd frames. For each dataframe, the maximum and minimum values are obtained and the regression lines around each value are calculated to obtain the time associated with the intersection of the lines. From the data obtained, the associated delay or advance in milliseconds is extracted and the mean is calculated, as well as the standard error.

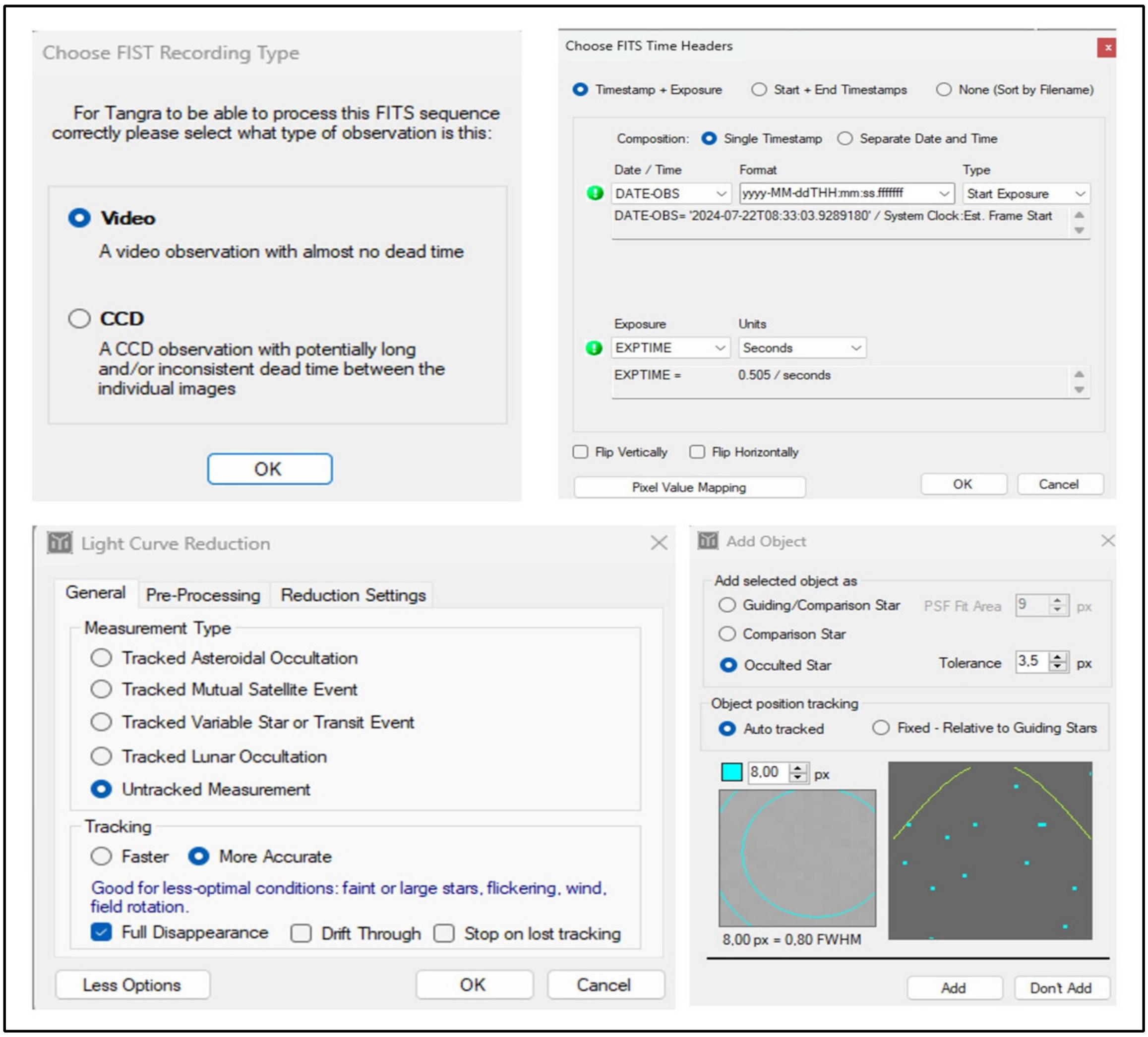

Figure 10. Screenshots to indicate the configuration used in Tangra.

Results

Accuracy of Time Stamp

Table 1 shows the results measured with the AS183MM Pro camera, with an exposure of 505 ms, binning 1×1, no gain, 5-minute image capture. All the recordings have been made with a frame size of only 32×32 in the centre of the image. The column samples indicate the total number of maxima and minima obtained and used to average the result and calculate the standard error between the samples. The mean of the measurement series is 0.48 ms, with a standard error (between the averages) of 17 µs. According to Figure 4, the average offset measured by the NTP during the tests was 0.15 ms, so we obtain a difference of 0.33 ms, which we will discuss in the next section.

Delay (ms)	Error (µs)	Samples
0.46	89	12
0.49	57	10
0.59	107	12
0.48	80	12
0.45	76	10
0.47	86	12
0.41	73	10
0.51	63	12
0.52	87	10
0.43	86	10

Table 1. Results with frame size (32×32), AS183MM camera, exposure 505 ms, duration each 5-minute measurement. Mean value of 0.48 ms, standard error of 17 µs.

Table 2 shows the results of a test under the same conditions but with the ASI120MM mini camera. According to Figure 5, the average offset measured by the NTP during the tests was 0.13 ms, so we obtain a difference of 1.2 ms, which we will discuss in the next section. There is an important difference between this result and that obtained for the ASI183MM camera which will also be explained also in the next section.

Delay (ms)	Error (µs)	Samples
1,20	61	10
0.92	136	12
1.06	34	10
1.05	47	10
1.11	51	10
1.07	46	10
1.26	82	10
1.27	53	12
1.09	44	10
1.27	69	10

Table 2. Results with frame size 32×32, AS1120MM minicamera, exposure 505 ms, duration each 5-minute measurement. Mean value of 1.33 ms, standard error of 51 µs.

Row Dependence of the Time Stamp in Sensors with Rolling Shutter

An interesting result of the tests with the stroboscopic method was the fact that measuring the timestamp in different rows for the same sequence of images generates different results. Figure 11 show the measured values in three different rows of the images: one as close as Tangra allows to the top edge, another in the centre of the image and the third at the top edge. With the camera ASI183MM Pro, an 800×600 ROI and a 1280×1024 ROI sequence were used. In both cases, a linear response with a slope of 0.0127 in one case, and 0.0126 in the other is observed. The explanation is simple: the time stamp of the image corresponds to what happens in the first row, while in the following rows what happens in the image takes places at a later time. The reason for this effect is the rolling shutter of the tested camera, i.e. the image picks up what happens in each row as the camera unloads the corresponding row from the sensor. According to this explanation, a timestamp must be associated with each row, which is the sum of the image timestamp (which can be associated with the first row) plus the readout time per row (sensor download time of a row) multiplied by the row number. In the tests carried out with the two ROI's indicated, the reading time per row is 12.7 µs and 12.6 µs (slope of the lines), so an approximate value of 13 µs is estimated.

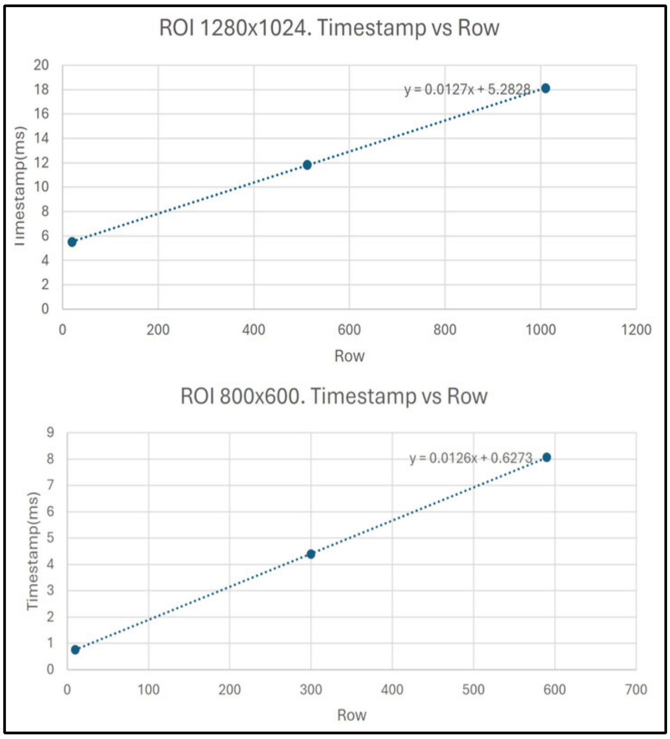

Figure 11. Timestamp measurements in different rows for two different ROIs, using the ASI183MM Pro camera.

To support this result, an additional experiment has been carried out, which also allows the readout time per row to be measured in a very simple way, albeit with somewhat less precision. The idea is to place a light source from the camera powered by AC power source. For the test I used a Neon pilot. Simply capture an image (exposure time is not relevant), just be careful not to saturate, and measure the number of lines between two maxima. If we divide 100 ms (how long the Neon takes between two maxima) by the number of lines separating two consecutive maxima in the image we get the readout time. I have obtained a separation of 269 lines which gives a readout time of 13.3 µs, less than 1 µs difference with the value obtained with the stroboscopic method.

Figure 12. Image captured with Neon light and camera ASI183MM Pro.

The same tests have been carried out for the ASI120MM mini camera with a readout time of 66 µs.

Let us now return to the results obtained for the 32×32 ROIs. In the case of the ASI183 camera, the average result was 0.48 ms. If we subtract from this value the redout time corresponding to row 16 where the measurements were taken (centre of the image), we have 0.48-16*0.013=0.27 ms, when with the NTP we obtained an average time lag with respect to the server of 0.15 ms (Figure 4). In the case of the ASI120MM mini camera, the average result was 1.33 ms. If we subtract from this value the redout time corresponding to row 16 where the measurements were taken, we have 1.33-16*0.066=0.27 ms, and with NTP we obtained an average time offset of 0.13 ms (Figure 5).

In both cases, the difference between the measured time and the expected time according to the PC offset monitoring with the NTC service is of the order of 0.1 ms, a difference that can be attributed to both the processing of the cameras and the software downloading the camera images. This is, however, a very small time and within the sub-millisecond that I consider the stroboscopic method allows us to measure.

An additional test he performed was to measure the timestamp of the central row of different ROIs. Table 3 shows the results. Without going into a detailed analysis that would require a deeper knowledge of how the camera firmware manages the timestamp, we can conclude some interesting results:

- The timestamp associated with the centre line of the image sequences increases as the row on which we measure increases.
- Up to an ROI of 320×240, the increase corresponds to the readout time quite well, but as the ROI size increases, the curve loses linearity and it is necessary to consider other additional factors to explain the higher timestamp, such as the download time between the camera and the PC.

ROI	Measured Row	Time (ms)
32×32	16	0.4271
48×64	32	0.6212
192×192	96	1.521
320×240	120	1.6693
496×350	175	2.6816
800×600	300	4.3892
1024×768	384	6.4449
1280×1024	512	11.8434
1600×1200	600	19.1985
1832×1836	918	34.8295

Table 3. Results in different rows, with ASI183MM Pro, exposure 505 ms, 5-minute measurement.

All measurements were made with gain and offset 0, 16-bit colour space and 1×1 binning. It has been found that the readout time per row does not vary if the binning, gain or offset are changed, but it does change if the colour space is changed: 16-bit gives twice the readout time per row as 8-bit. However, this may depend on the camera used and it is recommended to measure the readout time per row with the same configuration with which the occultations are going to be made.

Measurementof PC Time Drift

The stroboscopic method has also been used to measure the time drift of the PC when it deactivates the NTC synchronisation. To perform this test, the NTC service was forced to stop for one hour and the time stamp was measured with 5-minute series, using a 32×32 pixel ROI and the ASI183MM Pro camera. Figure 13 shows the result, which turns out to be very linear with a drift reaching 100ms in one hour.

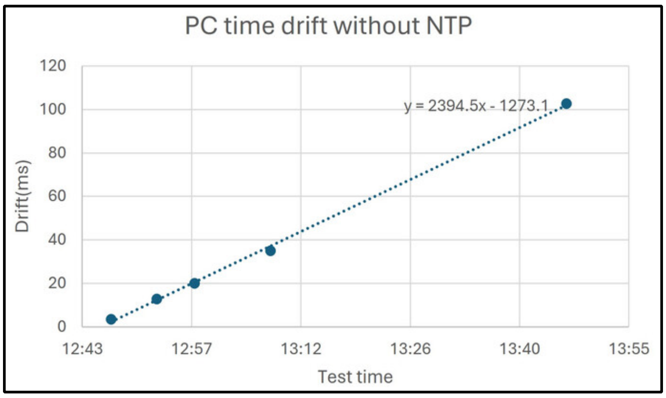

Figure 13. PC time drift when stopping NTP synchronisation.

This result highlights the importance of keeping the PC synchronised to obtain correct results in occultation measurements.

Discussion

Flash Pulse Duration and Offset Setting

In applying this method, a flash pulse of 500 ms duration and an observation window of 505 ms generates the stroboscopic effect. The question arises as to whether this is the correct selection and why.

Regarding the duration of the flashes, we studied the alternatives of 250 ms and 1000 ms. In the case of 250 ms, we would have to generate, between two rising edges of the PPS signal, a first 250 ms pulse, wait 250 ms, and generate another 250 ms pulse. This poses a technical problem in that the second 250 ms pulse cannot be triggered by an interrupt, so the high accuracy obtained from only 4-microsecond difference between the rising edge of the PPS signal and the light pulse cannot be guaranteed. In the case of 1000 ms pulses, the above-mentioned problem does not occur, because it is only necessary to prepare the code so that, with the interruption of the rising edge of the PPS, the pulse of 1 s is alternately activated or deactivated. However, this pulse of twice the duration does not bring any advantage, it only makes the time between maxima and minima longer and consequently the measurement procedure longer. Based on the above, it is considered that light pulses of 500 ms are the most suitable option.

Regarding the length of the observation window, we can consider options between 501 ms and 510 ms. In the case of 501 ms, the time between maxima or minima would be of the order of 8 minutes (1000/2 = 500 s) with up to 125 (500/4) points to define the regression line before and after a maximum and a minimum. In the case of 510 ms the time between minima or maxima would be 50 s and the number of points for each regression line would be only 12. For 501 ms, the time of 8 minutes is considered excessive to establish the total time of the video inorder to have enough maxima and minima to allow averaging. On the other hand, at 510 ms the number of points for the regressions is too low. The 505 ms option is a suitable compromise, as it provides sufficient and well-sampled maxima and minima to obtain the regression lines in the 5-minute measurement.

Effect of the Rolling Shutter on the Time Stamp of the Measurements

Most CMOS cameras use a rolling shutter, which is an advantage as it increases the frame rate by not having to stop between exposures. But this way of reading implies that the different parts of the captured image correspond to different events in the time domain [18].

Suppose an image where the object to be measured is in the first row, the associated timestamp would be t1 (Figure 14), but if the centroid was in row 1000, the associated timestamp would be t1000. The difference between the two times is the readout time

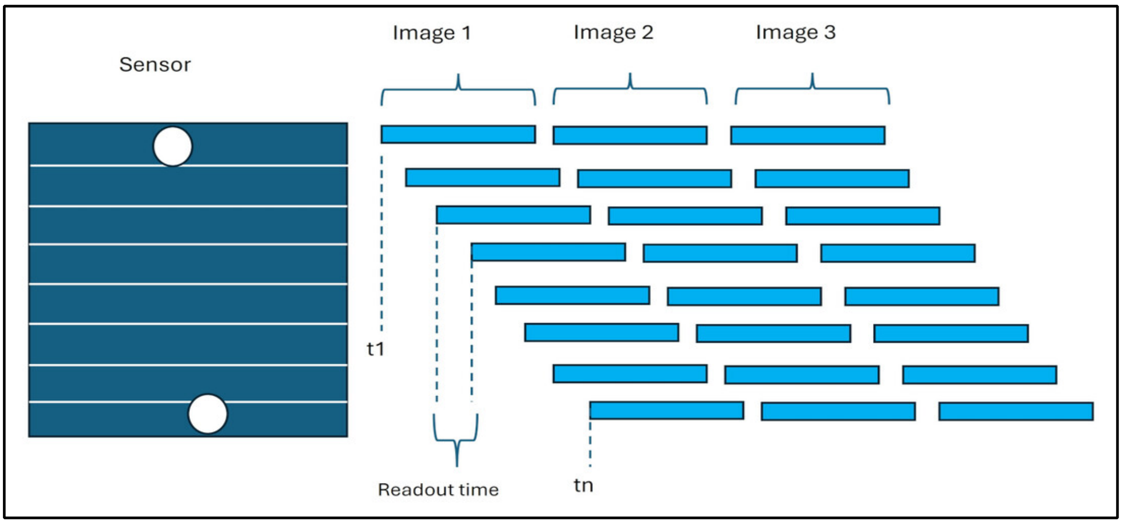

Figure 14. Rolling shutter. t1, tn refer to the text.

per row multiplied by the number of rows between the two measurements. Using the ASI120MM mini camera as an example, a readout time per row of approximately 66 µs has been calculated. The error introduced, if the occultation is measured on row 1000 and the readout time per line is not taken into account, is 66 ms, which in some applications may be unacceptable.

If the camera used for the occultation is a rolling shutter camera, which is in fact the most common, knowing the impact of the rolling shutter described above, it is possible to improve the timestamp of our measurement by adding to the timestamp of the image the result of multiplying the readout time per row by the row number of the centroid where the occultation occurs.

But Table 3 shows that the correction due to the readout time is not the only one to take into account, except if small ROI's are used. The recommendation is to characterize our system in order to be able to apply the correction properly. To do this, we recommend a scenario similar to the one used in this study, in which we have verified that during the tests the PC was very well synchronised (average offset less than one millisecond) and with the stroboscopic method we can measure the offset of our configuration: camera, ROI, approximate line on which the concealment is going to be measured, binning, etc. This measurement, repeated to check that it is stable, reveals the offset to be applied to the timestamp we measure in the occultation. In addition, if possible, this measurement can be made before and after the occultation to include an eventual offset of the PC. It only takes 5 minutes to make a measurement with this method.

It should be noted that these rolling shutter considerations do not apply to cameras with global shutter as is the case for cameras with CCD sensors and CMOS with global shutter.

Summary

The described method, based on a stroboscopic effect, allows us to measure the deviation of the timestamp of our system with sub-millisecond precision respect to a PPS signal coming from a GPS, in a simple (by just placing a LED in front of the telescope) and cheap (the cost of the flash generator can be around 60 €) way. Any available device that generates a 500 ms pulse synchronised with PPS can be used.

The image processing procedure for the calculation of the deviations is also very simple as it only requires the photometry (for instance using *Tangra* or *PyMovie/PyOTE*) of the area of the image where we are going to do the occultation to generate a light curve. With a *Python* script it is easy to calculate the deviation.

This method allows us to analyse the offset time generated by our setup, i.e. the computer with its synchronisation mechanism, the software and the camera. On the other hand, it allows us to know if any corrections need to be made to our observations, by measuring the delay/ahead minutes before and after the event.

Finally, the importance of considering the effect of the rolling shutter in most CMOS cameras has been highlighted, especially in captures with ROIs of many rows. It has been explained how to calculate the readout time per row (two possible methods, both stroboscopic and with a lamp connected to AC power such as a Neon) and how to use this data to report a more accurate timestamp.

It should be noted that this method is only useful for measuring small deviations (less than 500 ms) and therefore it is necessary that the PC used for the measurements has an adequate synchronisation system.

References

[1] Rommel, F. L. et al. Stellar occultations enable milliarcsecond astrometry for trans-neptunian objects and centaurs. A&A, 644 A40 (2020) https://doi.org/10.1051/0004-6361/202039054
[2] https://www.qhyccd.com/qhy174gps-imx174-scientific-cooledcamera/
[3] Niu, X. et al., Quality evaluation of the pulse per second (pps) signals from commercial gnss receivers. GPS Solutions, 19:141–150, 01 (2014), https://doi.org/10.1007/s10291-014-0375-7
[4] Barry, M. A. et al., Verifying timestamps of occultation observation systems (2015), https://www.cambridge.org/core/journals/publications-of-the-astronomical-society-of-australia/article/verifying-timestamps-ofoccultation-observation-systems/F71907D2D928CE75150FA8E629D8A126
[5] Kamiński, K. et al., Reaching submillisecond accuracy in stellar occultations and artificial satellite tracking. Publications of the Astronomical Society of the Pacific, 135(1044):025001 (February 2023) https://iopscience.iop.org/article/10.1088/1538-3873/acacc8
[6] https://www.zwoastro.com/product/asi183/
[7] https://www.zwoastro.com/product/mini-cameras/
[8] https://learn.adafruit.com/adafruit-ultimate-gps-logger-shield/overview
[9] https://store.arduino.cc/collections/boards-modules/products/arduinouno-rev3-smd?pos =7f id = e49402708ss = c
[10] https://static.garmin.com/pumac/GPS_18x_Tech_Specs.pdf
[11] https://www.thorlabs.com/thorProduct.cfm?partNumber=PDA10A
[12] https://www.keysight.com/us/en/product/DSOX3024T/oscilloscope-200-mhz-4-analog-channels.html
[13] Meinberg NTP software, https://www.meinbergglobal.com/english/sw/
[14] Stirk, A., 5 minute guide to making a GPS Locked Stratum 1 NTP Server with a Raspberry Pi, https://ava.upuaut.net/
[15] Glover, R. SharpCap software, https://www.sharpcap.co.uk/
[16] Pavlov, H. Tangra software, http://www.hristopavlov.net/Tangra3/
[17] Anderson, B., PyMovie software https://occultations.org/observing/software/pymovie/
[18] Gurgul, A. et al., LightStream – enhancements to the Astrometry24.NET software for processing SST and asteroid images from CCD and CMOS. In 2nd NEO and Debris Detection Conference, page 76 (January 2023)