Automated parallax software for meteor analysis

Patrick Benito Eberhard

September 21, 2019

Abstract

Analyzing meteor trajectories using the parallax method and calculating parameters such as the altitude, speed, angle of incidence and position in space of the object can be very exhausting. Furthermore, meteors can reach speeds up to 60 kilometers per second and are only observable for a short fraction of time. In this article, we work on two video recordings from the Perseid meteor shower, taken at different places, and perform the parallax method for each frame. In addition, we estimate that a shifting of a tenth of a frame can modify our result by 10 kilometres, which means that it is necessary to improve accuracy.

For this reason, this article will focus on the design of a method to improve frame synchronisation and the creation of a python script capable of obtaining three dimension meteor trajectories from two video sources. Thanks to the method, we are able to reduce our error to ±5.25km and obtain useful data from the meteors, such as the trajectory, velocity, GPS coordinates, altitude, angle of incidence and brightness per unit of time.

1 Introduction

On August 13th of 2019, the Asociación Shelios performed an observation of the Perseid meteor shower using two Sony Alpha cameras, one located at the Oukaïmeden Observatory (Morocco) and the second one placed at a distance of 800 meters from the Observatory. Even though the clocks of the cameras were synchronised, there was a difference in time of several seconds, which is unacceptable in our case, as the meteors from the Perseid meteor shower have an average speed of 59 km/s (Serra-Ricart et al., 2014).

In consequence, we are using a plot of the brightness from the meteor, being a reference to find synchronized frames or use interpolations to improve the accuracy. The most commonly used method for this is to consider the brightest frame from both samples as if they were taken at the same instant. It is a valid method, yet it can often be difficult to find a specific absolute maximum value.

Figure 2 shows the brightness of the meteor measured by our two observers. As the meteor is being observed at the same time, it should show a similar graph. We can clearly see that there is an absolute maximum in the first sample. Nevertheless, sample 2 has its absolute maximum shifted to the left side.

The first part of the article is based on finding a way to resolve this problem mathematically, by fitting both shapes together. The second part of the article focuses on astrometrizing our frames automatically, obtaining the equatorial coordinates of the meteor, converting them to horizontal coordinates (altitude and azimuth) and triangulating the meteors position per unit of time using three dimensional vectors. Finally we estimate the error committed in these calculations.

Figure 1: Two frames captured by different observers.

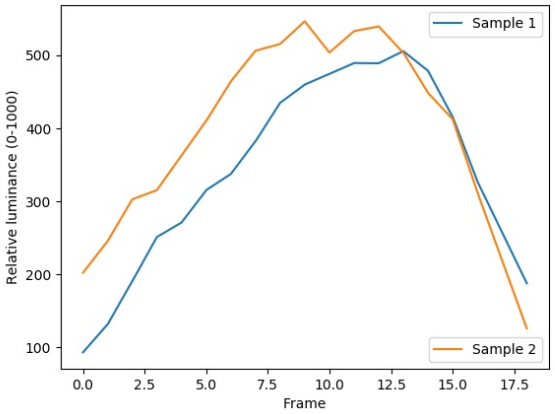

Figure 2: This is our sample. The blue line represents the brightness of the meteor taken by the first camera. The orange line corresponds to the second video source. The data has been filled with a linear interpolation.

2 Synchronizing frames

Both cameras have a different calibration and sensitivity. This means that we would not necessary measure the same relative luminance in a certain instant. Nevertheless, both cameras should measure a similar increase or decrease of brightness when time passes by. As a result, both samples should share similar shapes, but can be displaced vertically (due to a different sensitivity) and horizontally (caused by the lack of synchronisation in time).

Instead of taking just the maximum from the brightness data, we want our graphs to merge as perfect as possible. This way, we can ensure that we are taking into account the hole function and not only a specific range. After calculating the brightness of the meteor using MaximDL, we obtain similar but not identical brightness plots. These variations could be explained as we use an area plot, which is selected manually, to determine the objects brightness in each frame. On the other hand, we have had the case that there was a star behind or near the meteor, which disturbs our curve. For this reason it is better to take all the data into account, so we can reduce the impact of these unwanted modifications.

In order to fit both graphs, we want the area difference between them to be as small as possible. We first need to divide the x-axis into tiny fragments whose base will tend to zero. Then we multiply each base by its corresponding y-value and calculate the difference between both samples. The difference must remain positive. After adding up all the small differences, we obtain the surface area that remains between the plots. The area is equal to the following function:

$$\lim_{h\to\infty}\left[\sum_{i=max(1,a)}^{min(len(x1),len(x2)+a)}\left[\frac{|F2(i-a)+b-F1(i)|}{h}\right]\right]$$

The limit indicates that our base tends to zero. As our functions do not have an infinite domain (only a specific range), we use the values above and bellow the summation to correct this. The function inside the summation calculates the area difference depending on the values a and b (horizontal and vertical shifting) for a value i that increases each time.

Our python script loops through all different possibilities by changing a and b values and selects the best candidate. After running the software we obtain Figure 2. We can see that the two peaks have been aligned almost perfectly, but there is still a slight displacement. We can also apply this method to more situations where there is no peak and automate the process easily, which is a great benefit.

3 Astrometry

Once we merge the brightness charts, we can proceed to obtain the celestial coordinates from our image. This means that we want to know the Right Ascension and Declination of our meteor in each frame. In order to achieve this, the python script loads one image from each camera to an ftp server and then sends the link to an API called http://nova.astrometry.net. This service calibrates our image and returns the following parameters:

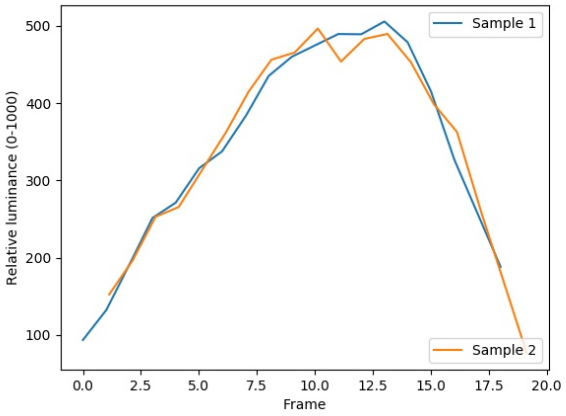

Figure 3: The second sample (orange) has been shifted 1.13 frames to the right and we obtained an average error of $0.1278u^2$ (relative luminance).

```
ra: 84.52932092993754,
pixscale: 36.85724854374876,
dec: 1.061210980135474,
orientation: 55.1221077687581
```

The software now has to obtain the celestial coordinates starting from the position of the meteor in pixels. Thanks to the API, we know the position of the center of our image in the sky. Using the *pixscale* we know how many seconds of arc each pixel equals. Finally, the orientation has to be considered. There are several ways to convert the position in pixels into celestial coordinates. Our program calculates the angle between the orientation line and our meteor and uses simple trigonometry to obtain the desired values.

Let's consider $\vec{p} = (x - C_x, C_x - y)$ as the vector that indicates the meteors position from the image center.

$$\theta(\vec{p}) = \arctan(\frac{p_x}{p_y}),\ \forall\ 0° < \theta$$
$$\alpha(\vec{p}) = \sin(\theta(\vec{p}) - orientation) \cdot p \cdot pixscale$$
$$\delta(\vec{p}) = \cos(\theta(\vec{p}) - orientation) \cdot p \cdot pixscale$$

The right ascension and declination will be given by $\alpha(\vec{p})$ and $\delta(\vec{p})$. It is important that our angle θ corresponds to its quadrant. For example, if both components of $\vec{p}$ are negative, we obtain an angle between 0° and 90°, but need to add up 180° in this case.

4 Finding the meteors position

After obtaining the position of the meteor in each frame, we want to convert the equatorial coordinates into horizontal coordinates. Knowing the two GPS positions of the observers and the time when the images were taken, we can calculate the altitude and azimuth mathematically using the parameters mentioned above (Burnett, 1998). There are several libraries which simplify this calculation process. Our python script uses a function called *eq2hor*, provided by the Hamburger Sternwarte.

Once this process is finished, we can start creating a three dimensional space with two vectors, which correspond to the observers locations. We could consider the earth as a plane, but as we want to work with spherical coordinates, we want our earth to have a spherical shape and take its center as the origin. We then obtain the observers vectors using latitude (lat), longitude (lng) and altitude (alt) the following way:

$$obs_x = (R_{earth} + alt) \cdot \cos(lat) \cdot \cos(lng)$$
$$obs_y = (R_{earth} + alt) \cdot \cos(lat) \cdot \sin(lng)$$
$$obs_z = (R_{earth} + alt) \cdot \sin(lat)$$

The vectors length will be the sum of its altitude and the earths radius (R_{earth}).

Furthermore, we want to draw a director vector for each observer that represents the direction the camera is observing the meteor. The idea is to find the intersection between these two lines and consider it as the real position of the observed celestial body at a certain time. We build the director vector using the altitude and azimuth values obtained from the *eq2hor* function and the observers position vector. The line will have a similar format to this: $\vec{r} = \vec{obs} + \lambda \cdot \vec{direction}$

As the horizontal coordinates cannot be transformed into a vector like in the previous step, we create a north ($\vec{n}$), east ($\vec{e}$) and up ($\vec{up}$) vector for each observer. By projecting the azimuth and altitude over these three vectors, we can get the direction of the lines that we want to intersect.

The up vector is going to be parallel to the position vector and will have a length of one unit. We only need to divide the position vector by its length to achieve this. Then we calculate the north vector, which we obtain by considering that $\vec{n}$, $\vec{obs}$ and the z-axis vector (0,0,1) are co-planar and the north vector is perpendicular to the observers vector. This is the simplification of the process:

$$n_z = 1 \qquad n_y = -up_z/(\frac{up_x^2}{up_y} + up_y)$$
$$n_x = \frac{up_x}{up_y} \cdot n_y$$

Finally, the east vector will be equal to the cross product of the north and the up vectors. Using trigonometry we can obtain the direction of the lines like in the previous step.

Now we can create two lines and find the nearest point between them, as they may not share one specific point due to previous errors. There are many ways to calculate this. An example can be found using the following link: http://morroworks.palitri.com/Content/Docs/Rays%20closest%20point.pdf

5 Results

By calculating the meteors position in each frame and interpolating these results, we obtain the following data:

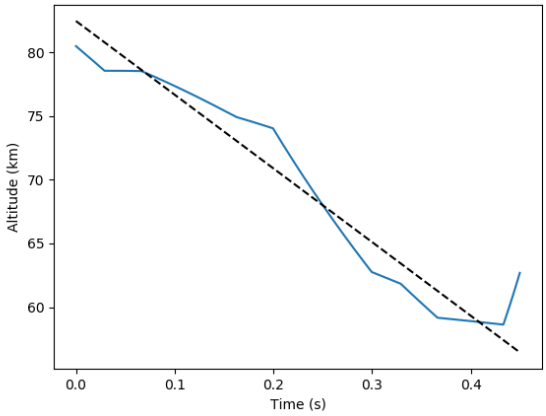

Figure 4: The figure shows the altitude of the meteor above sea level as a function of time.

In order to calculate its vertical speed, we can use the slope of the grey line, which is a regression line, to obtain an approximate value. In this case, the vertical speed would be $v_{vertical} = -57.8km/s$, with an r value of 0.97.

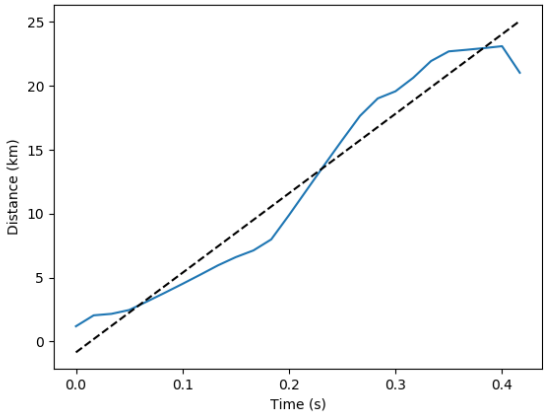

Figure 5: Here we can see the distance travelled by the meteor as time passes by.

Using the same process as above, the travelling speed of the meteor is $v = 62.2km/s$, which is very close to the average speed of the Perseid meteor shower (59km/s) (Serra-Ricart et al., 2014).

The angle of incidence can be calculated using the vertical and travelling speed, by calculating the angle between them. This can be achieved this way:

$$\alpha_{incidence} = \arcsin \left| \frac{v_{vertical}}{v} \right|$$

We obtained that $\alpha_{incidence} = 68.3°$ from the horizon (90° would be a vertical trajectory).

Last but not least, as we created the observers position vector using GPS coordinates, we can use the position vectors of the meteor and convert them back to geographical data. By doing this, we can estimate where the object would approximately land (if it reached the surface), its origin and the trajectory in Google Earth for example. This is how it would look like (Figure 6 and 7).

Figure 6: The meteor observed from a random location.

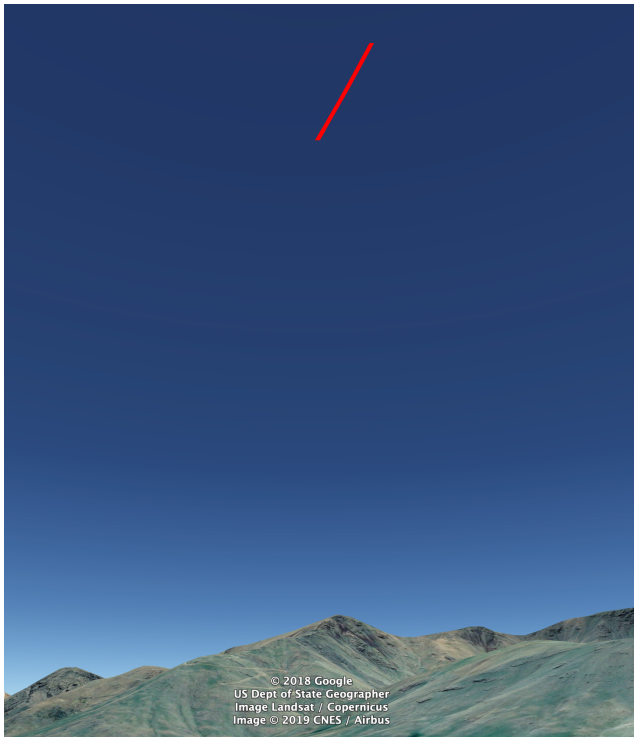

Figure 7: This is the view from Oukaimeden, where the videos were taken. The trace should be almost identical to the samples that we took.

6 Error estimation

Estimating the error of our calculations is not very simple. First of all, we need to consider that our cameras are not perfect and can show some slight variations. For example, we detected that the camera had missed some frames and duplicated others in several ocations. The frame rate is very accurate, but can sometimes be altered too. These errors need to be considered, but are different for each device.

On the other hand, when we select our meteor in a frame and calculate its brightness using an area plot, the measured luminance can also be different depending on the size of the area plot. Even though our script removes the noise from the background, we still obtained different values for the same frame by only varying the size of the area plot. This can cause a false displacement of 0,15 frames (equivalent to ±14km error), and needs to be resolved in the future for more accuracy. We made fifteen different setups and calculated the average values for the brightness curves for our example.

Another important factor is the position of the meteor that we estimate in each frame. It can be located in the center of the line bar of the meteor or at the front tip for example. We observed that selecting the front part results in a straighter trajectory, whereas the center of the bright area was giving us very deformed trajectories. This section also needs to be improved, by finding the best location and minimizing errors.

Now we want to analyze the errors that our software takes into account and is responsible for. The function that shifts our frames and reduces the area difference between both brightness curves has an error of ± 0,03 frames. We used our method using linear, quadratic and cubic interpolations and obtained this result by comparing them.

Another possible source of errors would be the Nova Astrometry API. We calculated the equatorial coordinates of different stars and compared it to databases. There was a difference of 1.3 arc min, which seems very high, but does not affect the trajectory prediction, as it applies equally to all frames. However, it can vary the range of speeds and altitude estimations by ± 3km in our example.

Finally, there is also an error of approximately ±250 meters while triangulating the meteors position. This

is the minimum distance between our two position lines that originate from each observer. As we are dragging errors from previous steps, there is a high chance that they will not intersect. Nevertheless, this error is very low compared to other decisive factors.

By introducing this possible mistakes made during the calculations into our program, we have measured a total error of ± 5.25 kilometers for our results (position, altitude, speed and angle of incidence), without taking into consideration the ± 0.15 frame shifting due to the area selection method and selecting the front tip of the meteor.

7 Conclusion

The automated parallax software allows us to calculate several parameters of the trajectory of a meteor using two video sources. Even though the results can vary remarkably depending on the selection method of the brightness area plot and the selection of the position of the meteor, we can ensure that our software commits an error of ± 5.25km in our example starting off the inputs that we, as users, provide to it.

It is easy to manage and allows us to calculate trajectories very fast and completely autonomously, given a certain input of positions, images and brightness area plots.

This software can also be used for bolid or plane tracking, as well as other visible objects that travel through our atmosphere.

This project can be downloaded and used (MIT License) via this url: https://github.com/PatrissTV/PyParalax.

8 Acknowledgments

This work has been possible to carry out thanks to Fundació Catalunya-La Pedrera and Asociación Shelios, for allowing me to participate in an unforgettable expedition to observe the Perseid Meteor Shower in Morocco.

I would also like to express my deep gratitude to Miquel Serra-Ricart, Miguel Rodriguez Alarcón and Manuel Mallorquín Díaz for their support in this project.

References

Burnett K., 1998, Converting RA and DEC to ALT and AZ

Serra-Ricart M., Pio Jimenez M. A., Casado J. C., 2014, Cálculo altura de formación de los meteoros