Orbit fitting of satellites crossing the entire field of view

Adrien Cadet
Laboratory of Astrophysics (LASTRO)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
adrien.cadet@epfl.ch

***Abstract*—Knowing the orbit of space objects is crucial for ensuring safety and security of space operations. However, orbit represented by two-line element set (TLE) at a specific epoch can lose accuracy in a few days due to their simplified propagation models. Moreover, existing techniques of orbit determination with telescope images require to know precisely the time of the observation, meaning that the start or the end of the streak should be visible, which is often not the case. This paper proposes an approach to determine a new TLE fitting better the observations of objects crossing the entire field of view directly from a single telescope image. Using an initial TLE, as well as right ascension (RA) and declination (DEC) measurements of the observed streak, we investigate the possibility to estimate the time at which the object was crossing the detector using the Mahalanobis distance as a correction metric for this time uncertainty. Furthermore, we will address some issues encountered during this project and its past iteration from Sarah Marciniak [1]. Nevertheless, the results were not convincing as the metrics used do not follow the expected behavior because of their sensitivity to several factors.**

I. Introduction

As the space near Earth is becoming increasingly crowded with small satellites or debris, being able to predict accurately the position of an object at a given date is critical for ensuring the safety of space operations. When telescopes observe the sky, they capture light for a certain time, called exposure time, that usually spans from a few seconds to several minutes. For this reason, moving space objects appear as streaks on the image, as you can see in the observation below:

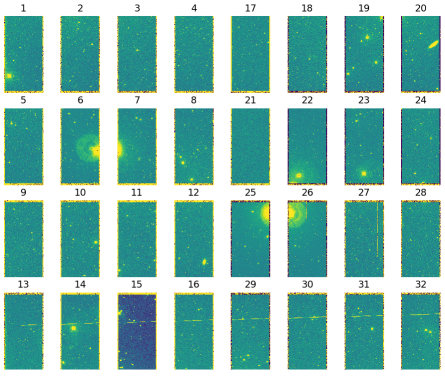


Fig. 1: Streak crossing a telescope image.

This image, taken on `2022-05-27T01:35:45.988`, is coming from the VLT Survey Telescope (VST) and its 256-megapixel camera called OmegaCAM, located at ESO's Paranal Observatory in Chile. Moreover, because the OmegaCAM is made up of 32 individual charge-coupled devices (CCDs), that capture and store the light, arranged in a mosaic, each observation is composed of 32 "smaller images" called tiles. However, unlike the above example, streaks often cross the entire image, as the exposure usually do not start or end when the moving object was in the field of view.

Furthermore, orbits of moving objects can be described by two-line element sets (TLE), a data format encoding the orbital elements of the object at a specific point in time, called the epoch. From the above image informations, we can know which object has caused this streak and retrieve its TLE with the epoch closest to the observation time, for example by using the SatChecker service available online [2]. Thanks to this TLE, we can predict the positions in the sky where the object was supposed to be during the observation. However, TLEs have inherent uncertainties and can lose accuracy in a few days due to their simplified propagation models. That is why the predicted sky positions by the TLEs and the observation do not always match perfectly. Indeed, you can see below the same previous example from the VST, where the predicted positions of the TLEs at different epochs are superimposed with the tile 13 of Fig. 1:

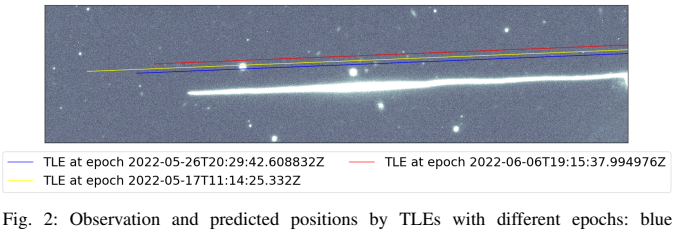


Fig. 2: Observation and predicted positions by TLEs with different epochs: blue corresponds to a TLE with epoch closest to the observation, yellow corresponds a TLE with epoch around 10 days before the observation, and red corresponds to a TLE with epoch around 10 days after the observation. The Z in the times indicates that the time is expressed with respect to the time zone of Greenwich, in UTC time scale.

In order to estimate a new TLE fitting better the observation, we need first to have some measured positions. However, we also need to know precisely the time when the moving object was at these positions. This time is available for images where the streaks starts (or ends) in the image, as we know that

at the time of the start (or end) of the exposure, the object was at the measured sky position of the streak start (or end). However, for objects crossing the entire field of view, this time is not available as the moving object could have crossed the detector at any time between the start and end of the exposure. If we could know precisely at which time the object was at the observed positions, we would be able to find a new TLE fitting accurately the observation, even for an object crossing the entire field of view. This new TLE could be useful for instance to combine multiple measurements of the same object observed from different locations. Therefore, by first finding an initial estimation of the times when the object was at the observed positions and by then using a correction metric to refine them, while making use of the streak's start information to compare our results, the goal of this project is to find a precise estimation of the time at which an object crossed the entire field of view.

II. Method

A. General approach

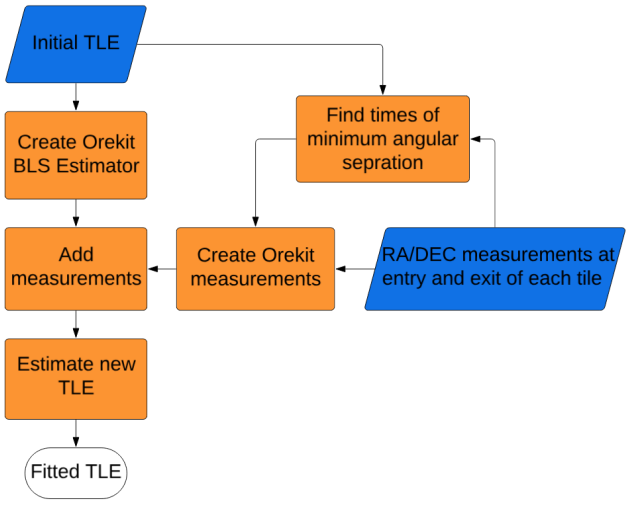


Fig. 3: General appoach flowchart.

As you can see in the above flowchart (Fig. 3), the method starts with an initial TLE: the TLE belonging to the object that caused the observed streak, with the epoch closest to the observation date. Indeed, as stated in the introduction, we can retrieve a list of TLEs that crossed the field of view of the telescope around the observation date, for example by using the SatChecker service available online [2]. From that list, we can identify the moving object that caused the observed streak in the image and retrieve its TLE closest to the observation date. In this project we used the specific observation from the VST that you can see on Fig. 1 presented in the introduction. An object caused the streak that you can observe in Fig. 1 on the 8 following tiles: 13, 14, 15, 16, 29, 30, 31 and 32. For this specific example, the object has a NORAD Catalog ID of 38094 and its TLE with epoch `2022-05-26T20:29:42.608832`, closest to the observation time of `2022-05-27T01:35:45.988`, is:

```
1 38094U 12009B   22146.85396538 -.00000219  00000-0  00000-0 0  9991
2 38094  18.2529 291.2232 6217657 112.2434 299.5415  2.11807545 79313
```

This TLE is then used to create a Batch Least Square estimator (BLS) in the Orekit python wrapper library. Moreover, thanks to another project, we have the right ascension (RA) and declination (DEC) of the entry and exit points of each tile. However, as opposed to other measurements techniques, we do not have the precise times at which the object was at these positions in the sky. In order to add measurements to the estimator, we therefore need to first estimate the time at which the object was at these RAs and DECs. This is done by propagating the initial TLE to times in the interval $[t_{\text{obs, start}} - 10\text{ s}\ ;\ t_{\text{obs, start}} + \Delta t_{\text{exptime}} + 10\text{ s}]$ where $t_{\text{obs, start}}$ denotes the exposure start and $\Delta t_{\text{exptime}} = 320$ s the exposure time. We add ten seconds of margin on each side of the interval to handle the cases where the object was just at the border of the field of view when the exposure started or ended. In our case, propagating a TLE at a given time means to compute the RA and DEC of the object predicted by the TLE at that time. This propagation is done thanks to simplified analytical perturbation models like SGP4 and SDP4 that take into account the main perturbations acting on the object (earth's gravitational field, atmospheric drag, solar radiation pressure, etc.). We then used an algorithm similar to dichotomy to search for the time among the interval at which the predicted position is the closest in angular separation with the RA and DEC measurements. Once we found these times, we can then use them to create RA and DEC measurements, add them to the BLS estimator and estimate a new TLE.

For example, if we want to estimate a new TLE using the RA and DEC measurements made on both the entry and exit of tile 30, we can compute, thanks to the initial TLE above, the times at which we assume the object was at these positions:

TABLE I: Entry and exit times of the object in tile 30.

	RA [rad]	DEC [rad]	Estimated time
Entry	3.4517744	-0.2929217	2022-05-27T01:38:35.499963Z
Exit	3.4539852	-0.2929990	2022-05-27T01:39:10.725139Z

These two measurements at entry and exit of tile 30 are then provided to the estimator and a new fitted TLE can be obtained:

```
1 38094U 12009B   22147.06846644 -.00000219  00000-0  00000-0 0  9993
2 38094  18.2487 291.0965 6215912 112.3592 103.2309  2.11959050 79317
```

We can now propagate this new fitted TLE to times in the interval $[t_{\text{obs, start}}\ ;\ t_{\text{obs, start}} + \Delta t_{\text{exptime}}]$ and display the tile of the streak start, *i.e.* tile 13, as you can see on Fig. 4.

In this Fig. 4, you can see that already with only two measurements, the fitted TLE is much more aligned to the streak than the initial TLE. Indeed, we can compute the angular separation, between the streak's start (green cross) and the fitted TLE's start and compare it with the angular separation between the streak's start and the initial TLE's start. This was computed in arc-seconds, because as a rule of thumb, we know that a difference of 1 pixel horizontally

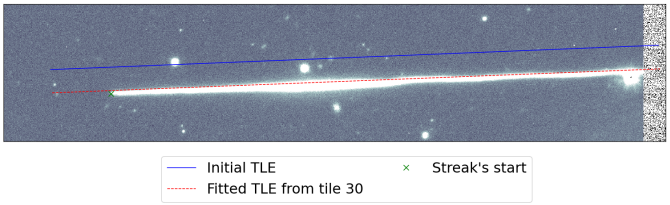


Fig. 4: Comparison between the initial TLE (in blue) and the fitted TLE (in red) using measurements from tile 30.

or vertically corresponds to around 0.2 arc second. We obtain an angular separation between the streak's start and the initial TLE of 0.0001459 radians or 30.1 arc seconds corresponding to a distance of 141 pixels. The angular separation between the streak's start and the fitted TLE's start with the measurements of tile 30 improved to 0.000132 radians or 27.3 arc seconds corresponding to a distance of 128 pixels. As you can see in the picture, the fitted TLE is much better aligned with the streak. However, the start of the fitted TLE is still offset compared to the real streak's start, as the angular separation improved (decreased) by only 9.3 %. Adding more measurements to that fitted TLE would only improve the position fitting (meaning that the TLE would pass closer to the measurements), but not the fit between the streak's start and the fitted TLE's start. It would improve the fitting mostly across the streak, but not along it. Therefore, in order to find a fitted orbit that is closer to the streak's start, we need to find a better estimation of the measurement times.

B. Correction metric approach

Indeed, those times were only estimated from the inital TLE and the goal is now to find a time shift Δt, that makes the fitted TLE better align with the streak's start without using the streak's start information, as it will not be available in cases where the object crosses the entire field of view. This time shift should be applied on all the previously computed times at once in order to not change the angular velocity of the object. However, doing the first orbit determination with the wrong measurement times is in some sense "breaking" the initial orbit. The idea is therefore that if we estimate a new TLE using the true times where the objects was at the measured positions, the fitted orbit would be closer, in a orbital distance sense, to the inital TLE. That is why we assume that their exist a time shift that would minimize the "orbital distance" between the fitted TLE and the initial TLE, as this time shift would lead to the fitted TLE to be closest, in the orbital space, to the initial orbit.

The term distance here can take several forms, as different metrics in the orbits space have been developed for identifying orbits of common origin. We investigated three different metrics: the metric ϱ_2, with and without the mean anomaly, detailed in [3] and [4] respectively, as well as the Mahalanobis distance following this metric ϱ_2, also described in [3]. All these metrics depends on the six Keplerian orbital elements: semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node (RAAN) Ω, argument of perigee ω and mean anomaly M. We will now describe the different metrics used.

1) Metric ϱ_2: This orbital distance between two orbits does not take into account the mean anomaly. It is defined in [3]:

$$\varrho_2 = \sqrt{\frac{(\mathbf{u_1} - \mathbf{u_2})^2 + (\mathbf{v_1} - \mathbf{v_2})^2}{L}}$$

Where we took $L = 1$, and the required vectors $\mathbf{u}$ and $\mathbf{v}$ are defined in [3] as:

$$\mathbf{u} = \begin{pmatrix} s\sqrt{p}\sin\Omega \\ -s\sqrt{p}\cos\Omega \\ c\sqrt{p} \end{pmatrix}, \mathbf{v} = \begin{pmatrix} e\sqrt{p}(\cos\omega\cos\Omega - c\sin\omega\sin\Omega) \\ e\sqrt{p}(\cos\omega\sin\Omega + c\sin\omega\cos\Omega) \\ es\sqrt{p}\sin\omega \end{pmatrix}$$

with $c = \cos i$, $s = \sin i$ and p is the *semilactus rectum*, defined for elliptical orbits as $p = a(1-e^2)$. ϱ_2 has a S.I. unit of $\sqrt{\text{m}}$.

2) Metric ϱ_2 extended with the mean anomaly: If we incorporate the mean anomaly inside metric ϱ_2, we can use the derived metric by A. Vananti et al. in [4]:

$$\varrho_{2M} = \sqrt{(\mathbf{u_1} - \mathbf{u_2})^2 + (\mathbf{v_1} - \mathbf{v_2})^2 + (\mathbf{w_1} - \mathbf{w_2})^2}$$

Where $\mathbf{u}$, $\mathbf{v}$ are defined in [3] as above and $\mathbf{w}$ is defined in [4] as:

$$\mathbf{w} = \sqrt{p}\begin{pmatrix} \cos M \\ \sin M \end{pmatrix}$$

Note that we call this metric ϱ_{2M} for clarity here, to indicate that this is the ϱ_2 metric extended with the mean anomaly, but this metric was simply called d in [4].

3) Mahalanobis distance following the ϱ_2 metric: A last metric that we tried is the Mahalanobis distance following Kholshevnikov's metric as defined in [4]:

$$d_M = \sqrt{(\mathbf{z}_1 - \mathbf{z}_2)^\top \mathbf{C}^{-1}_{\mathbf{z}_1 - \mathbf{z}_2}(\mathbf{z}_1 - \mathbf{z}_2)},$$

Where:

$$\mathbf{z} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{pmatrix},$$

$\mathbf{C}_{\mathbf{z}_1 - \mathbf{z}_2} = \mathbf{C}_{\mathbf{z}_1} + \mathbf{C}_{\mathbf{z}_2}$ and $\mathbf{C}_\mathbf{z} = \mathbf{T}\mathbf{C}_\mathbf{p}\mathbf{T}^\top$.

The $\mathbf{T}$ matrix is defined in the article [4], but this also requires the covariance matrix of the orbital elements $\mathbf{C_p}$ for both orbits. This metric has no unit. The covariance matrix of the orbital elements of the fitted orbit is obtained directly from the orbit determination process in Orekit. However, the covariance matrix of the initial orbit can be obtained by gathering a sufficient number of TLE at an epoch ranging from 10 days before until 10 days after the observation. We can then propagate them to the comparison time (the exposure start), extract the orbital elements and compute the 6 x 6 covariance of the orbital elements. Thanks to those metrics, we can now try to refine the previously computed times.

However, due to a software issue we encountered, we could not add more than two measurements to the estimator.

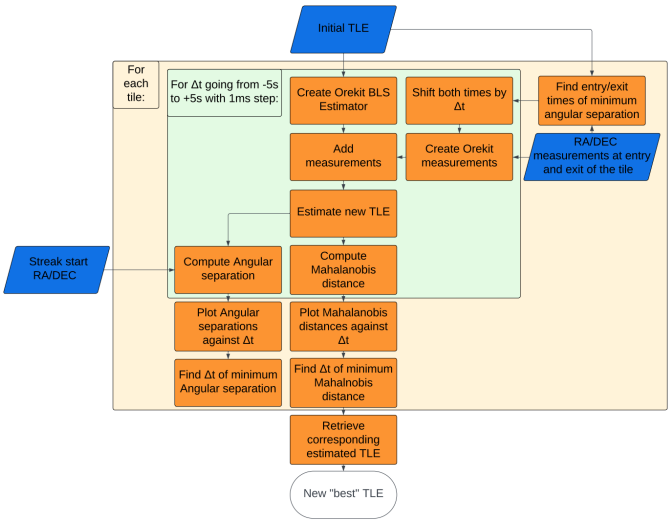


Fig. 5: Correction metric approach flowchart when adding only two measurements at a time to the estimator the Mahalanobis distance metric.

Therefore, we added the two measurements for each individual tile. Indeed, as you can see in Fig. 5, we have two loops: one for the tiles and one for the time shift. For each tile, we start by computing the entry and exit times by finding the time of the minimum angular separation between the initial TLE and the measurements of entry and exit of this tile. Then for each time shift Δt going from -5 s to +5 s with 1 ms steps, new Orekit measurements are created using the same RA/DEC at entry and exit, but shifting both times by the same amount Δt. We then follow the same process as before: we recreate a new BLS Estimator thanks to a Levenberg-Marquardt optimizer and the initial TLE. The measurements with the shifted times are then provided to the BLS estimator and a new TLE is estimated. This orbit is then propagated at the exposure start in order to compute the angular separation with the real streak start. In parallel, the Mahalanobis distance between this new orbit and the initial TLE is computed. After, a new TLE was estimated and the angular separation and the Mahalanobis distance was computed for every time shift, we can plot the two metrics against the time shift Δt for every tile.

To compare our results, we use the angular separation in tile 13 as a "validation" metric. Indeed, to see if the "optimal" time shift that we find using this correction metric in orbit space matches the one of the minimum angular separation between the streak's start and the fitted TLE's start, that is the "target" Δt that we would like to find without using the information of the Streak's start. This approach could be then used in cases where the object crosses the entire field of view to find a precise estimation of the time when the object crossed the detector. Towards the end of the project, we fortunately managed to solve the software issue by propagating the initial TLE to the measurement times before creating the estimator. I will explain how we solved this issue in more detail in the Results section. Nevertheless, in the case were we could add all the measurements, the flowchart of the methods changes a little, as you can see in Fig. 6. Indeed, in this case we add all the measurement with the shifted times at once. There is thus no loop on all the tiles anymore.

Fig. 6: Correction metric approach flowchart when adding all the measurements at once to the estimator for the Mahalanobis distance metric.

The goal of using the correction metric approach is thus to find the time shift minimizing the angular separation, by finding the time shift minimizing the orbital distance between the fitted TLE and the initial TLE.

III. Results

A. Results when adding only two measurements

We will now describe the results when following the approach as described in the flowchart in Fig. 5.

Example 1

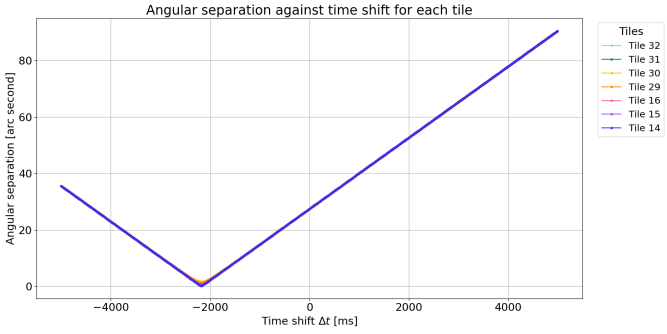


Fig. 7: Angular separation against time shift for a time shift going from -5 s to 5 s with a step of 10 ms.

You can see in Fig. 7 the angular separation plot for every tile. This plots exhibits a clear minimum in angular separation

at a time shift of around -2180 ms. Furthermore, this minimum occurs for the same time shift for all the tiles, indicating that the measurements on the different tiles are consistent. Nevertheless, we can notice very small differences between the tiles in the value of this minimum angular separation. Indeed, some tiles end up having a slightly higher angular separation than others even after the time shifting: for example, tile 14 has the lowest angular separation of around 0.194 arc second at a time shift of -2180 ms, while tile 30 has a minimum angular separation of around 1.68 arc seconds also at a time shift of -2180 ms. For computational time reasons, we plotted this graph for a time step of 10 ms only, but going to lower time steps would not change the overall shape of the graph. We will now describe the results we obtained for the different metrics.

1) Metric ϱ_2: In order to retrieve this time shift without using the streak's start information, we computed in parallel the metric ϱ_2 described in the Method section between the fitted orbit and the initial TLE. However, when computing this metric between the initial TLE and the fitted TLE at each time shift we obtain the results in Fig. 8.

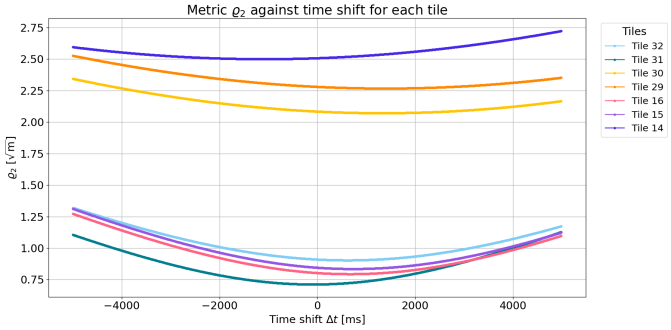


Fig. 8: Metric ϱ_2 against time shift for a time shift going from -5 s to 5 s with a step of 100 ms with a Levenberg-Marquardt optimizer.

We can see that we do not obtain similar results for every tile, which was not expected and is not what we were searching for, as we would like to see a clear minimum around -2.2 seconds as in the angular separation plot in Fig. 7. Furthermore, we noticed that changing the optimizer used for the estimation from a Levenberg-Marquardt optimizer to a Gauss-Newton optimizer was changing the shape of the metric plot as you can see in Fig. 9, but not the shape of the angular separation.

Moreover, by looking closer to the measurements on the images, as for this example the streak is relatively thick, we noticed that the RA/DEC measurements are not always located at the center of the streak thickness. Indeed, by projecting all the measurements on the line that was going through the measurements on tile 14, we obtain again different results in Fig. 10.

By slightly changing the positions of the measurements on the streak, we obtain the same angular separation graph as before, but as different correction metric graph. The thickness

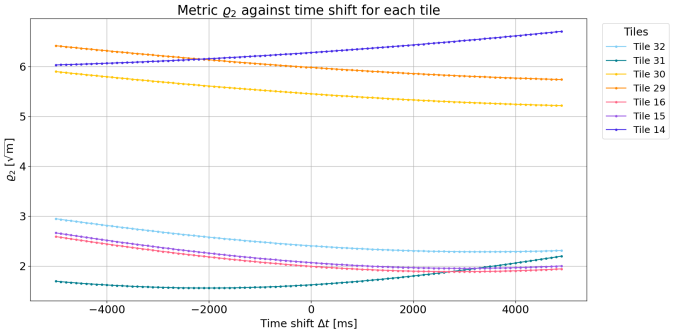


Fig. 9: Metric ϱ_2 against time shift for a time shift going from -5 s to 5 s with a step of 100 ms with a Gauss-Newton optimizer.

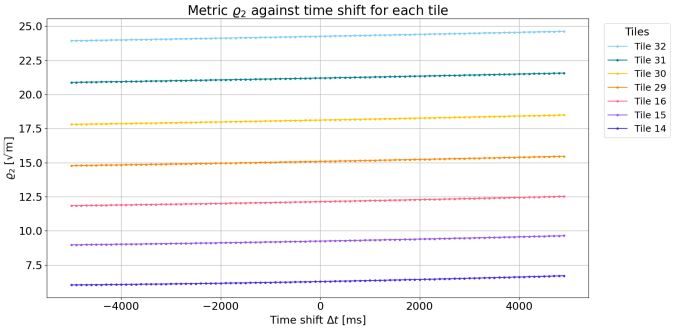


Fig. 10: Metric ϱ_2 against time shift for a time shift going from -5 s to 5 s with a step of 100 ms for measurements projected onto a line fitted on tile 14.

of the streak, causing this uncertainty on where along this thickness we should place the measurement, may be causing changes in orbital elements across the different tiles and is thus producing different graphs of the metric for the different tiles. We can see here the trend that the further away we are from the tile where we fitted the line on which we projected the measurements, the bigger the distance because the fit to our measurements is poorer. However, all the tiles seems to now behave in the same way. Therefore, the only way to get similar Mahalanobis distances between the tiles seems to be tweaking the positions of the measurements. However, we would like to have a more "robust" behavior as in the angular separation graph were the results do not change significantly when applying minor changes to the measurements. Note that unless specified, the optimizer we used is a Levenberg-Marquardt optimizer.

In parallel, this did not affect the shape of the angular separation graph: these changes are affecting mostly the Mahalanobis distance graph compared to the angular separation graph, indicating that the Mahalanobis distance calculation, based on the orbital elements of the fitted TLE is much more sensitive in the change of a measurement location by a few pixels than the angular separation between the streak's start and the fitted TLE's predicted streak start.

2) Metric ϱ_{2M}: Finally, we also tested to change the metric used. If we now compute the metric ϱ_{2M} incorporating the mean anomaly, for the original measurements, we obtain again different results for each tile, as you can see in Fig. 11. Switching metric does not change the angular separation graph as you can see in the flowchart, those are two parallel computations.

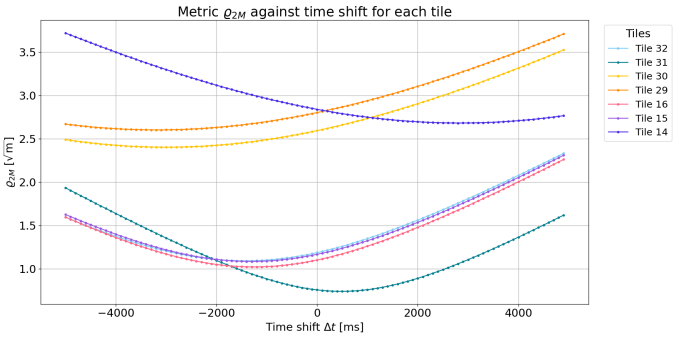


Fig. 11: Metric ϱ_{2M} against time shift for a time shift going from -5 s to 5 s with a step of 100 ms.

3) Metric d_M: Unfortunately, we were not able to compute this metric because when adding only two measurements, we obtained an singular matrix error when computing the `getPhysicalCovariances` method in Orekit.

Example 2

By looking at a second example from the VST telecope at a different date, we can follow the same procedure, but this times there are only two tiles when we can see the streak crossing the entire tile. Indeed, you can see in Fig.12 the angular separation graph for this second example. We are therefore searching to see an "optimal" time shift of around 130 ms. This time could be more precise, but again for computational time reasons, we did not put here the results for a smaller time step.

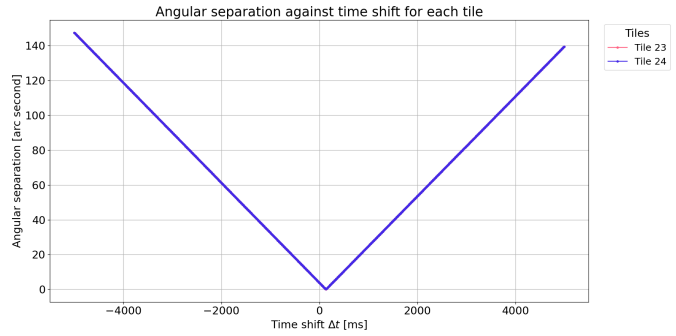


Fig. 12: Angular separation against time shift for a time shift going from -5 s to 5 s with a step of 10 ms.

For this example, we can see a clear minimum between for each tiles around 0 ms in Fig. 13. The fact that the tiles seems to match between each other might be due to the line that is much more thin that in the previous example, thus reducing the uncertainty in the measurements over the line thickness. However, this minimum is not coinciding exactly with the angular separation one. By changing the optimizer, we again obtain very different results as you can see in Fig. 14. When we project the measurements of tile 24 on a line fitted by the measurements on tile 23, you can see the results in Fig.15. If we change the metric used to ϱ_{2M} with we get the results in Fig. 16.

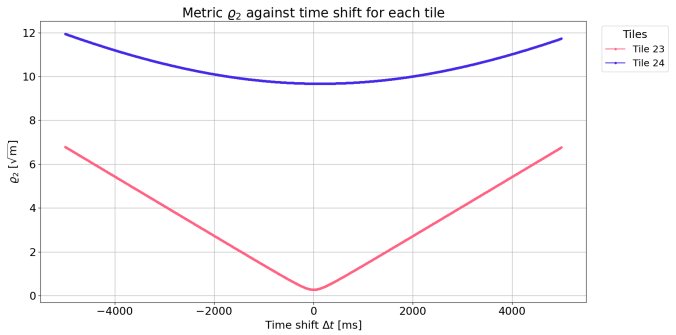


Fig. 13: Metric ϱ_2 against time shift for a time shift going from -5 s to 5 s with a step of 10 ms with a Levenberg-Marquardt optimizer.

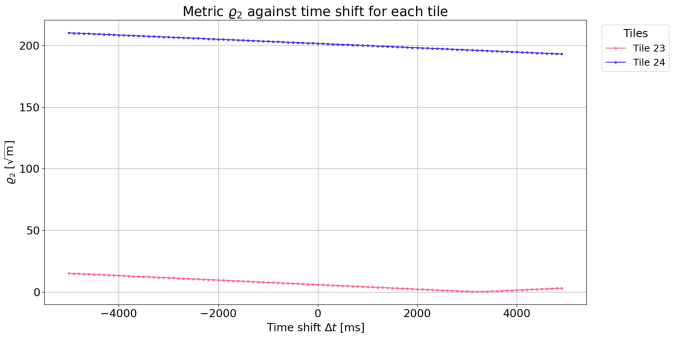


Fig. 14: Metric ϱ_2 against time shift for a time shift going from -5 s to 5 s with a step of 100 ms with a Gauss-Newton optimizer.

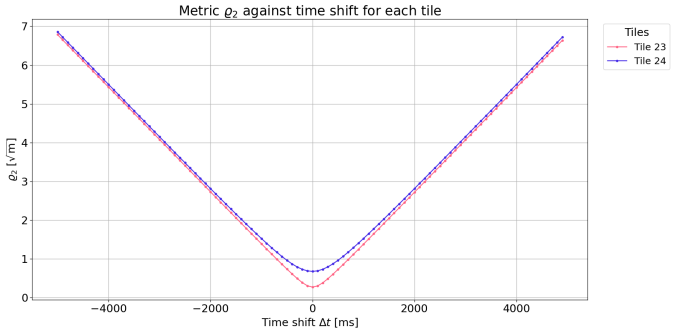


Fig. 15: Metric ϱ_2 against time shift for a time shift going from -5 s to 5 s with a step of 100 ms for measurements projected onto a line fitted on tile 23.

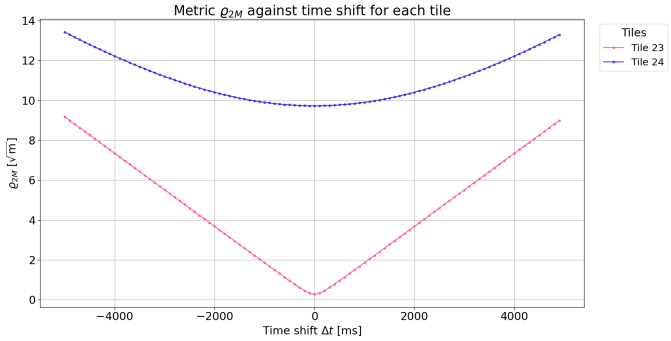


Fig. 16: Metric ϱ_{2M} against time shift for a time shift going from -5 s to 5 s with a step of 100 ms with a Gauss-Newton optimizer.

B. Results when adding all the measurements at once

After fixing the issue of adding all the measurements, we could compute again the results for the different metrics described. We will now describe the results when following the approach as described in the flowchart in Fig. 6.

Example 1

For all the different, metrics, we obtain the angular separation graph in Fig. 17 when adding all the measurement at once. This result in consistent with the previous case.

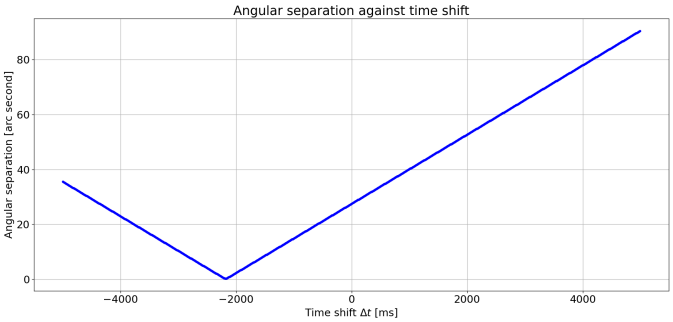


Fig. 17: Angular separation against time shift for a time shift going from -5 s to 5 s with a step of 10 ms.

We can observe that the minimum in angular separation is consistent with the one computed in Fig. 7 for the different tiles, occurring around a time shift of -2180 ms.

1) Metric ϱ_2: For example, if we compute the ϱ_2 metric for adding all the measurements, we get the results in Fig. 18.

2) Metric ϱ_{2M}: If we now incorporate the mean anomaly and use metric ϱ_{2M}, we obtain the same angular separation plot and the results in Fig. 19.

3) Mahalanobis distance d_M: Finally, by using the Mahalanobis distance following Kholshevnikov's metric, d_{M}, we obtain the results in Fig. 20.

Example 2

For all the different metrics, we of course obtain the same angular separation graph when adding all the measurement at

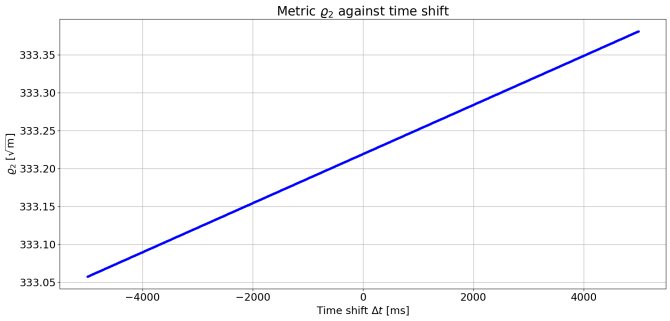


Fig. 18: Metric ϱ_2 against time shift with all the measurements for a time shift going from -5 s to 5 s with a step of 10 ms.

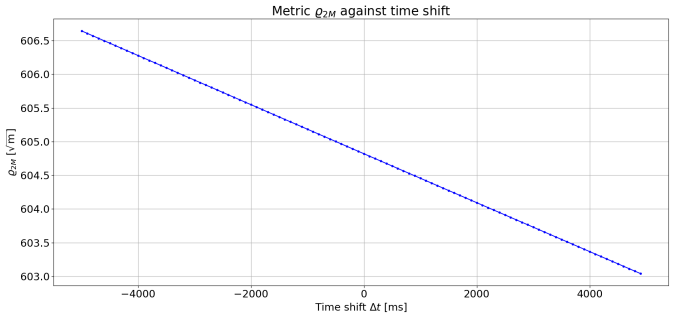


Fig. 19: Metric ϱ_{2M} against time shift with all the measurements for a time shift going from -5 s to 5 s with a step of 100 ms.

once. You can find the angular separation graph, consistent with the one when we were adding only two measurements in Fig. 21. The other metrics are displayed in Fig. 22, 23 and 24.

When adding more than two measurements, we can see that different metrics lead to different results, but each time the result is a line. For this case of adding more than two measurements to the estimator, we tried also changing parameters to see the effect on the metric. The Gauss-Newton optimizer was giving similar results as the Levenberg one, but was sometimes trapped in a local minimum. Moreover, trying to use the measurements projected onto a line for the d_M metric was not possible as, as the orbit determination process was returning an estimation error.

C. Improvements from the previous iteration of the project

We are now going to introduce the improvements made with respect to the past iteration of the project. Indeed, this project was based on the code and results from Sarah Marciniak, that you can find in [1]. We made mainly five major improvements of the previous implementation.

1) Time truncation: Indeed, the first problem we encountered was that after the times were computed by finding the ones minimizing the angular separation between the measurements and the initial TLE, they were rounded down to the nearest integer second. This was causing some undesired shifts for some tiles of the Mahalanobis distance graph. Indeed, the

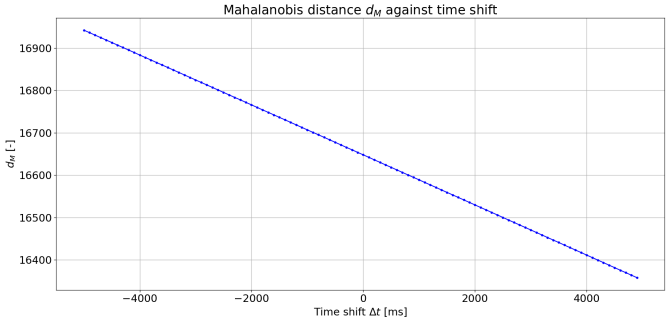


Fig. 20: Metric d_{M} against time shift with all the measurements for a time shift going from -5 s to 5 s with a step of 100 ms.

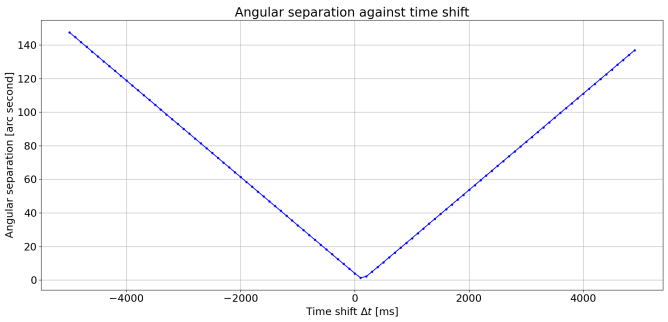


Fig. 21: Angular separation against time shift for a time shift going from -5 s to 5 s with a step of 10 ms.

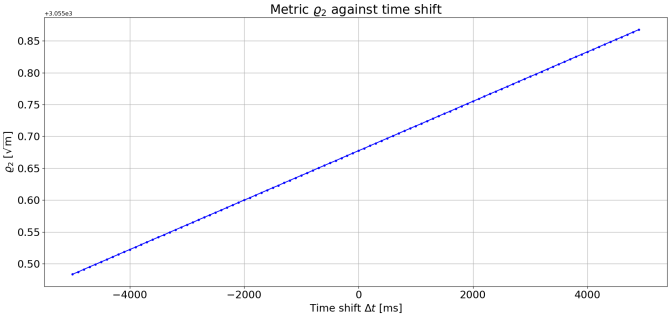


Fig. 22: Metric ϱ_2 against time shift with all the measurements for a time shift going from -5 s to 5 s with a step of 10 ms.

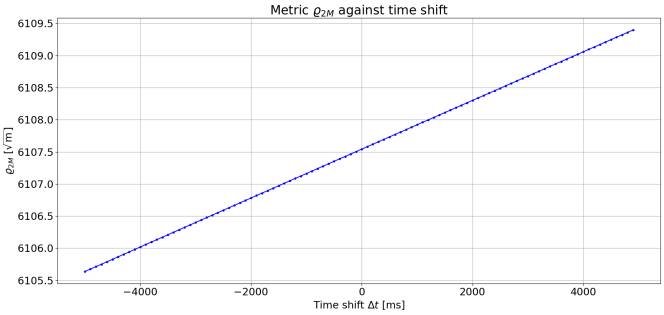


Fig. 23: Metric ϱ_{2M} against time shift with all the measurements for a time shift going from -5 s to 5 s with a step of 100 ms.

times must be very precise (at least at the millisecond-level precision) as it is crucial to preserve precision and to not ”break” the orbit in orbit determination by changing the some physical parameters like the angular velocity. A change of 1 second in the time can be considered as be big difference. After estimating the times of the measurements, we therefore now create the Orekit RA/DEC measurements with the computed times in the full precision.

2) Time shifting: Furthermore, when shifting the times of each tile to minimize the correction metric, we discovered that only the entry times were shifted and not both the entry and exit. This was changing the angular velocity of the object and ”breaking” the initial orbit in an undesired way. Moreover, we we were able increase the previous resolution of the time shift from 0.5 s to 1 ms. For computational time purposes, we only displayed here plots with 10 ms or 100 ms precision.

3) Time computation: Initialy, the times were computed using the Skyfield python library. The propagation was done using a method giving positions relatively far way from the Orekit propagation results. Therefore, we changed the propagation method done in Skyfield to a more precise one by loading Earth parameters and taking into account the light time delay, deflection and aberration by using the Skyfield `apparent` method. This change was now giving results very similar to Orekit. However, in a purpose of simplification of the code and of consistency of the library used, as we needed Orekit for the orbit determination, we choose to compute the times only using the Orekit library.

4) Mahalanobis distance calculation: Moreover, apart from cleaning the code, another improvement was made by identifying that the way the Mahalanobis distance was calculated was not correct. Indeed, the previous implementation was computing the covariance matrix of two vectors containing the 6 Keplerian orbital elements of the initial and fitted orbits. The computation should not be done in this way, as in the orbital elements space, the elements have a very different scale and some work has already been done on the development of metrics for identifying orbits of common origin, as in the already cited articles [3] and [4]. Therefore, we changed the metric computation to the already introduced metrics ϱ_2, ϱ_{2M} and d_M.

5) Fixing the error of adding more than two measurements: One diffcult point of this project was to figure out why adding more than two measurements to the BLS Estimator was causing this ”hyperbolic error”. That is why we were initially only adding two measurements at a time to the estimator because we could not add more without getting an error. We approached the problem through many different angles, trying to tweak the convergence threshold, the scale, the times, the positions, etc. Only a few approach successfully removed this error. The first approach that worked was to do a ”first round” by estimating a new TLE with two measurements on one tile and to then use this newly fitted TLE to reset a completely new estimator. Then, in a ”second round”, we can add the other

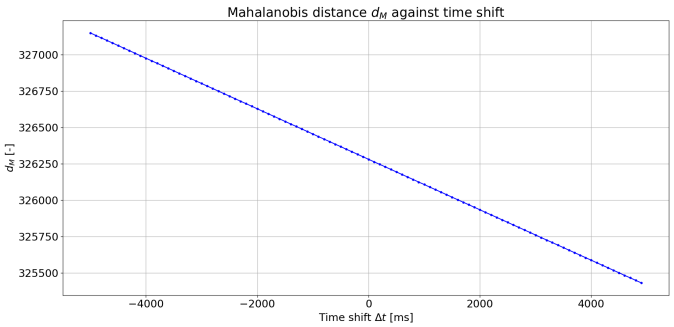


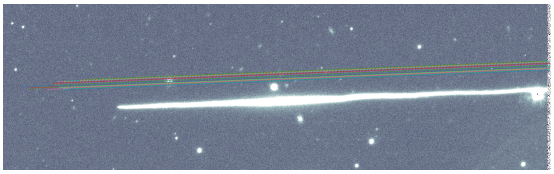
Fig. 24: Metric d_{M} against time shift with all the measurements for a time shift going from -5 s to 5 s with a step of 100 ms.

measurements to this estimator without any error. It served at first as a temporary solution, before finding a "cleaner" and proper way of removing this error. Much later in the project, we tried to see what was causing this error, by displaying the RMS and the cost of the objective function over the different iterations. We could see two types of problem: local minima and divergence. Indeed, for some cases, we could observe that some time shift in the measurements were causing an override of the maximum iterations allowed, even after increasing it. By looking at the value of the Root Mean Square (RMS) of the residuals, we could observe that the value was almost not changing over the iterations, indicating that the cost function was getting to a minimum, but was not passing below the threshold. This indicates that the optimizer was trapped in a local minima, probably because of a poor initial guess at that time shift. In some other cases, the cost would also diverge after a few iterations, also indicating a poor initial guess. It is after reading the non-linear least squares section of this book [5], that we understood the importance of the initial guess when doing a non-linear least square estimation. Indeed, this initial guess is used to do a Taylor expansion, and must therefore relatively close to the measurements. The estimation is computed in a loop until the cost becomes lower than a certain threshold, as this article [6] described the working principle of a BLS estimator. That is why, because the TLE is far away from the observations, we propagated it to the average time between the first and last measurements and were able to get rid of this error. However, as a long part of the project was done without the error being solved, we will first present the results we got with the correction metric computed for all the tiles separately, and then with all the measurement added at once.

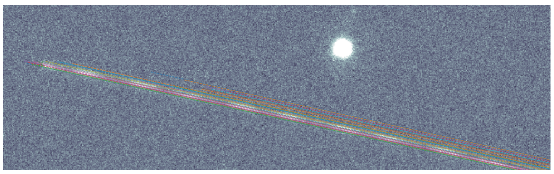
D. TLEs offset

We conducted the experiment on three different examples from the VST, you can see the three differnt streak and the predicted posistions by the TLEs in Fig 25.

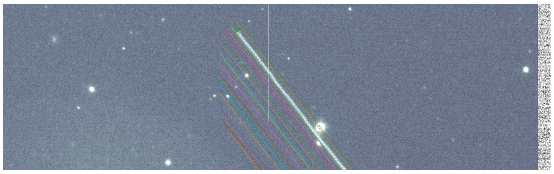
On the contrary of the two other examples, the TLEs are offset compared to the observation, but not because of its inherent uncertainties as in the two other cases. Indeed, in



(a) Start of observation: `2022-05-27T01:35:45.988Z`.



(b) Start of observation: `2022-06-09T08_43_38.103Z`.



(c) Start of observation: `2022-06-09T08_43_38.103Z`.

Fig. 25: Observations and predicted sky positions on several examples by TLEs with different epochs. Figure 25a corresponds to tile 13 of Figure 1.

the first picture, we can see that the TLEs are always offset compared to the observation, and because we are sure that the coordinates of the observatory and the pixel to RA / DEC conversion is correct, we can assume that either there is a bias in the way those TLEs are computed, or there is biais is the way we propagate those TLEs.

IV. Conclusion

Knowing the orbits of moving objects is crucial for the safety of space operations, especially with the increasing number of small satellites or debris. Due to inherent uncertainties, or biases in their computation, the TLEs can sometimes not match the observation. However, for objects crossing the entire field of view, orbit determination is not trivial as it is not possible to pinpoint the exact time where the object crossed the detector. This project was therefore investigating the possibility of using different metrics, in the orbital elements space, in order to precisely estimate that time. Nevertheless, the results were not convincing as the metrics used seem to be very sensitive to the measurements, to the optimizer used, and to the metric itself. We therefore obtained inconsistent results between the different tiles and examples on the contrary to the angular separation, that showed clear minima for all tiles and examples. Furthermore, the metrics do not behave as expected, for a reason that, we have to admit, is not very clear to us yet. It might come from a mistake in the code we used, despite scanning it in details several times, but from the results of the project nothing indicates that the metric used are able to retrieve the desired time without using the

streak's start or end information. Future work could therefore focus on understanding the reasons of the observed behaviors scanning the code in search for incoherencies or mistakes. Finally, validating the results obtained by using another library doing orbit determination could also enhance our confidence of the method.

REFERENCES

[1] S. Marciniak, "Orbit fitting of satellites that cross the entire field of view," unpublished.

[2] SatChecker, Satellite Passes Through FOV — SatChecker documentation, [Online]. Available: https://satchecker.readthedocs.io/en/latest/fov.html#satellite-passes-through-fov. [Accessed: Jun. 5, 2025].

[3] K. V. Kholshevnikov, G. I. Kokhirova, P. B. Babadzhanov, and U. H. Khamroev, "Metrics in the space of orbits and their application to searching for celestial objects of common origin," Monthly Notices of the Royal Astronomical Society, vol. 462, no. 2, pp. 2275–2283, Oct. 2016.

[4] A. Vananti, M. Meyer zu Westram, and T. Schildknecht, "Metrics on space of closed orbits for near-Earth objects identification," Celestial Mechanics and Dynamical Astronomy, vol. 135, no. 5, Oct. 2023.

[5] D. A. Vallado, Fundamentals of Astrodynamics and Applications, 4th ed. Hawthorne, CA, USA: Microcosm Press, 2013.

[6] B. Cazabonne and P. J. Cefola, "Towards accurate orbit determination using semi-analytical satellite theory," AAS Paper, no. 21-309, Feb. 1, 2021.