Dodging Satellites with the Rubin Scheduler

Jinghan Alina Hu

Abstract

To address the problem of bright commercial satellite streaks in images from the Vera C. Rubin Observatory, we create heuristics for satellite dodging strategies using the survey's scheduling algorithm. We computationally forecast satellite trajectories to account for the growing satellite population during Rubin operations. Our results help maximize efficiency and quality of this flagship observatory.

Introduction

Vera C. Rubin Observatory's Legacy Survey of Space and Time (LSST) is a ten-year astronomical imaging survey that will begin in 2024 from a new telescope in Chile [2]. Instead of soliciting individual requests for what the telescope should point at, the LSST will uniformly survey the sky every three nights in six color filters to essentially create a decade-long high-resolution survey of the entire southern sky, and share massive quantities of data products with the astronomy community [2].

To accomplish this, there is a complex scheduler algorithm that takes into account various science priorities, including slew time, survey uniformity, and image depth [4]. The Rubin Scheduler algorithm creates weightings for various science priorities and uses the weighting to create a path for pointing observation.

However, one challenge for the LSST is that increasing numbers of bright Low-Earth-Orbit (LEO) satellites (e.g., Starlink) are being launched, which could leave streaks in astronomical pointings. Over the last three years, many astronomers have raised concerns about the impact of this on the LEO ecosystem and astronomical surveys [3][6]. LEO satellites are visible from Earth because they reflect sunlight, especially during twilight. As the illuminated LEO satellites move across the field of view of an astronomical pointing, they leave a streak in the image, which negatively impacts the scientific value of the pointing. Astronomers predict there will be at least one satellite streaks in the majority of exposures of the Rubin Observatory [3]. An example of satellite streaks in an astronomical image is shown in Figure 1.

Figure 1. Two Starlink satellites (Jeremy Tregloan-Reed, Calar Alto Observatory, September 2020)

There have been efforts to reduce the impact of satellite streaks in astronomical pointings. Satellite companies like SpaceX have worked on darkening the exterior of satellites so they will be less illuminated [5]. However, even with extra darkening, the Starlink satellites are still one magnitude too bright for the observatory cameras, meaning these darkened satellite movements will still be captured by the camera. Astronomers have worked on algorithms for masking satellite trails in images, but covering the outer rim of the trails without losing extra pixels remains a challenge [6].

The rapid increase in population of LEO satellites threatens to compromise the quality and scientific value of the LSST images and also necessitates expensive computational power to mask the trails. Thus, we explore another option: incorporating known commercial satellites into the Rubin Scheduler so the worst of them may be avoided. In this project, we create realistic simulated forecasts of satellite trajectories and brightness, build a tool that uses that data to create new scheduler constraints, and test the impact of the new scheduler on various Rubin LSST observing programs.

Methodology

We begin by creating realistic forecasts of three different commercial satellite constellations: StarlinkV1 (4408 satellites, altitude 550 km), OneWeb (6372 satellites, altitude 550km), and StarlinkV2 (29988 satellites, altitude 550km). All three of these constellations are highly likely to launch as designed, or close to it, as both operators presently have functional satellites in orbit and are working to build and launch more. These simulated satellite constellations are illustrated in Figure 2.

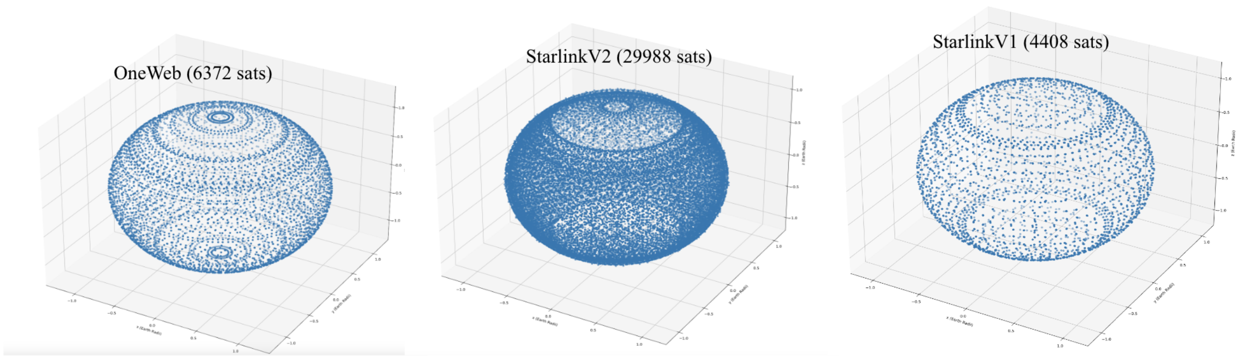

Figure 2. Three Simulated Satellite Constellations based on FCC filings for operators with existing satellites in orbit

Using these simulated satellite constellations, we add an additional weighting for satellite dodging to the Rubin Scheduler. Given an input satellite distribution, the scheduler predicts satellite maps across a determined time frame. Based on the predicted location of satellites, the scheduler creates a reward weighting: areas with more illuminated satellites will receive a more negative reward weighting, or in other words, the scheduler is incentivized to avoid them. The scheduler then combines the satellite reward weights with other weights to determine where to point the telescope to take the next set of simulated observations. Figure 3 shows an example of scheduler weighting for the three simulated satellite constellations at two different sun altitudes.

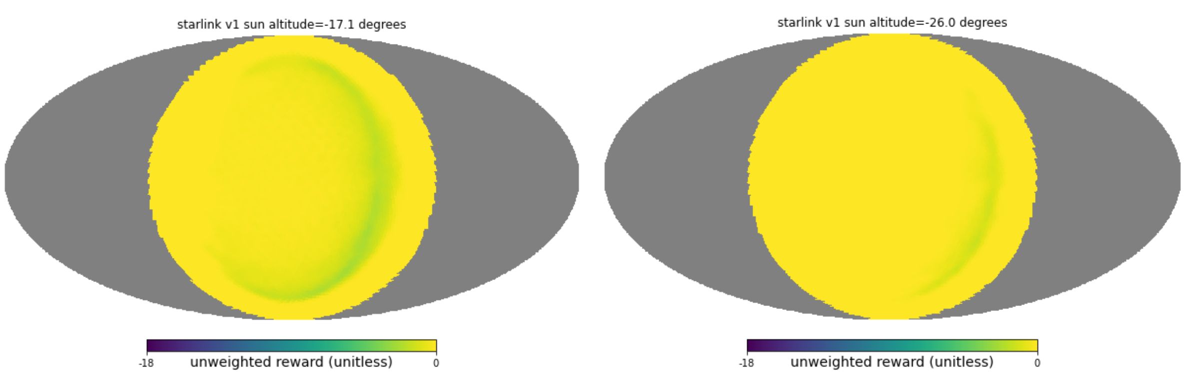

Figure 3(a). Simulated scheduler reward weighting for StarlinkV1 Constellation at sun altitude = -17.1 and sun altitude = -26 degrees.

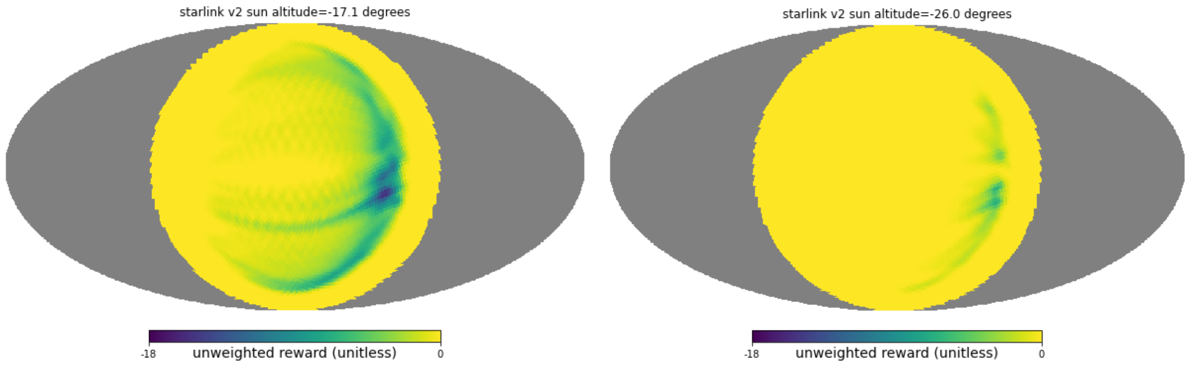

Figure 3(b). Simulated scheduler reward weighting for StarlinkV2 Constellation at sun altitude = -17.1 and sun altitude = -26 degrees.

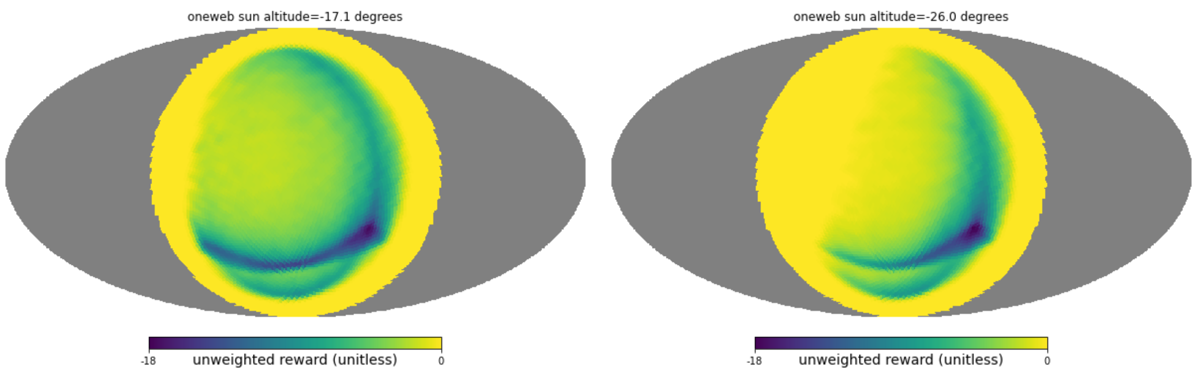

Figure 3(c). Simulated scheduler reward weighting for OneWeb Constellation at sun altitude = -17.1 and sun altitude = -26 degrees.

As shown in Figure 3, the predicted OneWeb constellation map has more negative reward weighting than other constellations. Although the OneWeb constellation has fewer satellites than the StarlinkV2 constellation, the OneWeb constellation is at a higher altitude, meaning that its satellites will be illuminated for a longer portion of the night, and not just close to twilight. Therefore, OneWeb shows more negative weighting than StarlinkV2 though it has a smaller satellite population.

To investigate whether the scheduler behaves how we expect with the new satellite weighting added we create a testing function that measures the length of satellite streaks in the simulated field of views. To ensure efficiency, only satellites that are above the altitude limit and illuminated by the sun will be considered. Satellites below the altitude limit (indicated in the gray region in figure 3) can't be observed in pointings and are therefore not included for efficiency.

For each satellite, we first determine whether they are in the field of view for any given telescope pointing by calculating their distance from the center of the field of view. If their distance from the field of view is less than the radius of the field of view, the satellite has crossed through the pointing. To quantify the impact of the satellite on the pointing, we then project the satellite location as well as the pointing to a 2D x,y plane. In the 2D plane, the field of pointing becomes roughly a circle and the start and end locations of the satellite become two points on the plane. Using the shapely library, we calculate the intersection length of the satellite location points and the circle [1].

Therefore, with a given simulated satellite constellation and a schedule of observations, we record the number of satellites that occur in each field of view and measure how long the satellite streaks are. Using a realistic estimated satellite streak width of 50 pixels, we can convert the streak length results into the number of affected image pixels, which allows us to quantitatively measure the efficiency of the dodging algorithm with pixel loss. The code is available at https://github.com/dirac-institute/satellite-dodging.

Results

We begin by investigating the efficiency of the added weighting algorithm on dodging satellites in observations. As shown in Figure r, using a conservative estimate of 50 pixels as streak width, we found that higher dodging weights reduces pixels lost to satellite streaks. We also found that smaller constellations at lower orbital altitudes (StarlinkV1 for example) inherently results in less pixel loss per pointing, nearly independent of the dodging weight. Figure 4 shows that the added weighted dodging algorithm was able to effectively avoid satellite streaks in pointings.

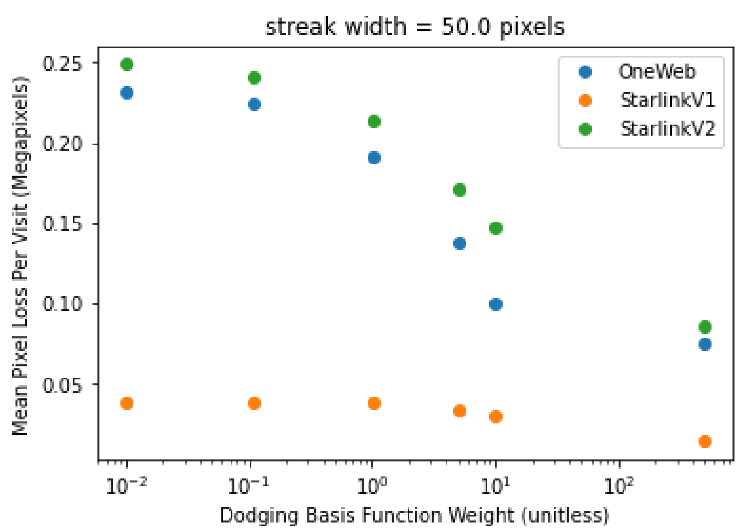

Figure 4. Mean pixel loss per visit decreases with higher dodging weight.

We then investigate the relationship between number of exposures taken and the dodging weight. As shown in Figure 5, higher dodging weight results in fewer exposures, most likely due to longer slew times. With the additional avoidance weighting, the telescope may be prompted to slew to a location other than the desired pointing location and then slew back in order to avoid some satellites, resulting in fewer overall exposures. We also found that larger constellations like

StarlinkV2 tend to decrease the number of exposures slightly more than smaller constellations (StarlinkV1), which is expected. More satellites or satellites at higher orbital altitudes result in bigger areas of avoidance on the sky, which results in more slewing to avoid the affected areas, which subsequently reduces exposures.

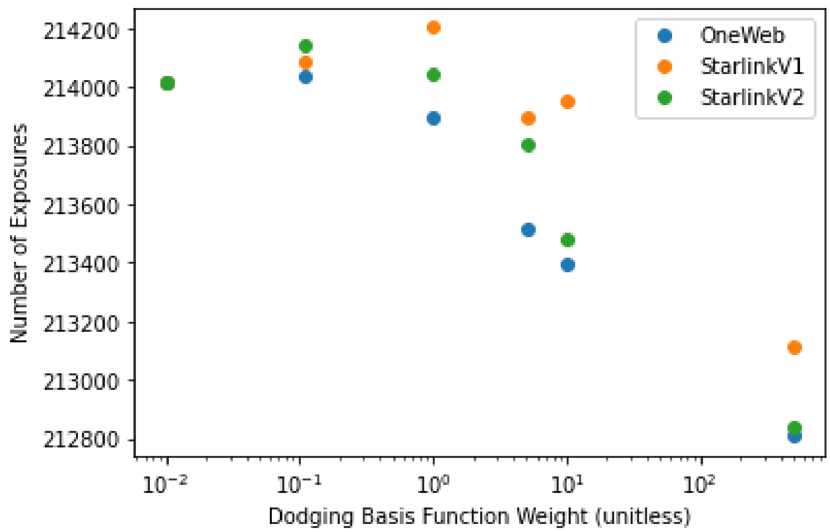

Figure 5. higher dodging weight and larger illuminated satellite population results in fewer exposures.

We now investigate the tradeoff between survey depth and satellite avoidance. One goal of LSST is to collect a large number of exposures of all parts of the southern sky so the exposures can be coadded to reveal faint structures that aren't visible in individual images. Therefore, survey depth is crucial to LSST science goals and the trade off must be evaluated. With the satellite avoidance algorithm, the scheduler is prompted to avoid regions with illuminated satellites, which could result in longer slew time or less desirable pointing conditions, which contribute to a loss in survey depth. Therefore, we evaluate the tradeoff between survey depth loss and satellite avoidance using the dodging algorithm.

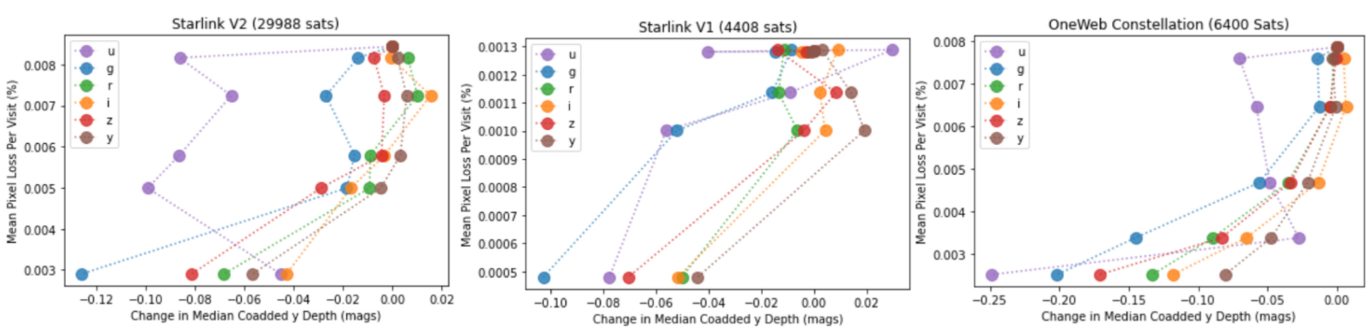

Figure 6. Trade-off between pixel loss and coadded depth

As shown in Figure 6, for all three constellations, minor reduction in mean pixel loss per visit (0.001% decrease) results in a significant decrease in median coadded depth. Currently, pixel loss due to satellite streaks only constitutes a very low percentage of total pixels in a pointing (about 0.008% without any dodging). Given that the percentage is currently so small, a very minor reduction in mean pixel loss per visit results in significant reduction in image depth.

Discussion and Conclusion

We have demonstrated that adding a weighted term in the scheduler algorithm for illuminated satellites can effectively reduce the amount of satellite streaks in observations, and subsequently reduce mean pixel loss per visit. This is shown in Figure 4 where all three constellations caused less pixel loss with increasing dodging weights. However, with the new added priority on dodging, the telescope can be pushed to take an observation path that does not optimize slew time, and subsequently reduces the number of exposures and overall survey depth.

The tradeoff essentially comes down to the relationship between mean pixel loss reduction and survey depth reduction. If the amount of pixels saved outweighs the amount of survey depth lost, then the algorithm is valuable to be added to the scheduler. The tradeoff relationship is evaluated in Figure 6. We found that, for all three constellations, the tradeoff between reduction of pixel loss and loss in coadded image depth is so significant that satellite dodging is overwhelmingly not worth it, provided the simulated constellations we selected are representative of reality. Since each pointing contains such a large number of pixels, the number of pixels destroyed by satellite streaks constitutes a very small percentage of the overall image. Therefore, while the weighted dodging algorithm is effective at reducing satellite streaks in images, at the current stage, satellite avoidance is mostly unnecessary considering the trade-off between pixel loss and loss in overall survey depth.

Future Work

While the satellite dodging algorithm is mostly unnecessary for the three satellite constellations we considered, many astronomers estimate that there will be sharp increases in satellite population in the next 5–10 years, which notably overlaps the LSST operations period (2024–2035). With the dramatic increase in satellite population, the dodging algorithm might become more relevant. In addition, satellites from other operators may be significantly brighter than present-day Starlink and OneWeb satellites, and may saturate the LSST Camera's detectors. In this scenario, dodging may be required to salvage some science, even if it means losing out on survey depth.

Related to this, one future work direction involves adding in brightness weighting to the dodging algorithm. The idea is to only avoid satellites brighter than a certain brightness threshold. This could potentially reduce the region of avoidance, therefore reducing the loss in coadded depth — it could reduce the tradeoff between pixel loss and loss in survey depth, making this preliminary work directly relevant for a successful LSST. Finally, it may be possible to compute optimal starting locations for observation schedules based on predicted satellite maps in order to optimize satellite avoidance.

Bibliography

[1] Gillies, S., et all. “Shapely: manipulation and analysis of geometric objects”. 2007. https://github.com/Toblerity/Shapely.

[2] Ivezić, Ž., et all. “LSST: From Science Drivers to Reference Design and Anticipated Data Products”, *The Astrophysical Journal*, vol. 873, no. 2, 2019. doi:10.3847/1538-4357/ab042c.

[3] Lawrence, A., Rawls, M.L., Jah, M., Boley, A., et all. “The case for space environmentalism”, *Nature Astronomy*, vol. 6, pp. 428–435, 2022. doi:10.1038/s41550-022-01655-6.

[4] Naghib, E., Yoachim, P., Vanderbei, R. J., Connolly, A. J., and Jones, R. L., “A Framework for Telescope Schedulers: With Applications to the Large Synoptic Survey Telescope”, *The Astronomical Journal*, vol. 157, no. 4, 2019. doi:10.3847/1538-3881/aafece.

[5] SpaceX. “Brightness Mitigation Best Practices for Satellite Operators”.

[6] Tyson, J.A., Ivezić, Z., BradShaw, A., Rawls, M.L., et all. “Mitigation of LEO Satellite Brightness and Trail Effects on the Rubin Observatory LSST”, *The Astronomical Journal*, vol. 160, no. 5, 2020. doi:10.3847/1538-3881/abba3e.