Draft version January 27, 2022
Typeset using LaTeX **twocolumn** style in AASTeX631

Investigating Timescales in Pipeline

Noah Swimmer

ABSTRACT

Compare t_{WCS}, t_{PSF}, and t_{bin}.

1. INTRODUCTION

We have concerns and questions regarding the timescales in the pipeline and how they affect the outputs that we see. These are:

- t_{WCS}: The time between WCS solutions being calculated
- t_{PSF}: The time it takes for the PSF to blur/smudge
- t_{bin}: The time between frames (e.g. the duration of a frame in a movie/gif)

and we need to consider how they interact when they are different relative to one another. There are 6 possibilities this can take:

1. $t_{bin} < t_{WCS} < t_{PSF}$
2. $t_{bin} < t_{PSF} < t_{WCS}$
3. $t_{WCS} < t_{bin} < t_{PSF}$
4. $t_{WCS} < t_{PSF} < t_{bin}$
5. $t_{PSF} < t_{bin} < t_{WCS}$
6. $t_{PSF} < t_{WCS} < t_{bin}$

We will discuss each in its own section.

1.1. $t_{bin} < t_{WCS} < t_{PSF}$

This configuration is when the frame duration in your output is less than both the WCS calculation time and the PSF blurring time. This should always be allowed (although the frame-to-frame difference may not show much, since the sky isn't moving enough to blur).

It should be said that although this is a *legal* configuration, having $t_{WCS} < t_{PSF}$ does not buy you anything, and you may be best served setting the two equal to one another.

In essence, having t_{WCS} less than t_{PSF} allows you to 'subsample' in time, but your data shouldn't change because of it.

VERDICT: Go for it. But it won't do much extra good.

IN CODE: nothing

1.2. $t_{bin} < t_{PSF} < t_{WCS}$

This configuration will also produce frames that are shorter than the time for 'sky' features (i.e. real, not from the telescope) to move appreciably, but now the user will run into an issue where the WCS is being recalculated less frequently than how long it takes the PSF to move on the sky. This will lead to blurring or smearing of the PSF.

Because t_{bin} is the smallest value here, most of the frames should not show PSF motion. The PSF motion would show up in the frames where time elapsed is greater than t_{PSF} but less than t_{WCS}.

To prevent this blurring (which does not have any obvious scientific merit), if this case is encountered, t_{WCS} should be set to t_{PSF} so that blurring or smearing is prevented, even though it will only show up in a fraction of the frames.

In short, this case seems like it *should* be modified when attempted to $t_{bin} < t_{PSF}$, $t_{PSF} = t_{bin}$.

VERDICT: You run the risk of getting weird behavior at the edges here. Most of the output should be unaffected, leaving you in a weird state with some affected, weird frames that blur. This may hurt your data, or you may not care, but common sense seems to say that this should be avoided/headed off by not allowing the second inequality to go up.

IN CODE: Throw warning that could lead to blurring, don't alter values

1.3. $t_{WCS} < t_{bin} < t_{PSF}$

This is another case that should not give many issues (or any) that I can think of. Although you may lose time/resources in computing the WCS more often than is necessary, you will not degrade/improve the output by doing it.

In the same vein, if you desire the frames to subsample the PSF motion (i.e. multiple frames before the PSF moves appreciably), then this is alright.

VERDICT: Again, this seems like it may be computationally wasteful since recalculating the WCS over and over within each frame when the PSF isn't moving doesn't buy you much in the way of data improvement, but there also shouldn't be much weirdness in here. A potential 'remedy' to this could be turning this into $t_{bin} < t_{PSF}$, $t_{WCS} = t_{bin}$. This is not required, but would result in fewer unnecessary recalculations of the WCS solution.

IN CODE: Set $t_{WCS} = t_{bin}$, add logging

1.4. $t_{WCS} < t_{PSF} < t_{bin}$

This is a funky case and I think that the behavior from it depends on the mode that the user is running (derotation or ADI).

To start: in both cases - as discussed previously - having $t_{WCS} < t_{PSF}$ is not significantly improving for your output. You could take the previously suggested route of setting t_{WCS} equal to t_{PSF} and not lose much.

Now we deal with what happens when derotating or not. This is a weird case because you have multiple WCS solutions in each frame, or you can also say that there will be multiple realizations of the PSF within the frame. They may not be discrete (separated by λ/D or some resolvable distance), but they will be enough to blur/smear the PSF, by definition.

In the case of derotating, this *shouldn't* create an issue since - as per my understanding as of 15 December 2021 - the derotation machinery operates on each WCS solution.

In the case of the ADI mode, this will likely run into an issue because it doesn't do any derotation. This will result in the blurring of the PSF within your image and degrading the image quality. This can be insignificant (if you're only looking for a companion's presense) to very significant (you want to measure photometry, astrometry, and SNR). A *potential* solution here is to allow the behavior to go on - it shouldn't break anything in spite of making the data poor quality - and send the user a warning that they likely want to decrease the size of t_{bin}.

VERDICT: This configuration can cause weird behavior that will cause data degradation. In the derotation mode, I need to explore this further, in the ADI mode this will *certainly* cause problems if done unintentionally.

IN CODE: Throw warning leave values

1.5. $t_{PSF} < t_{bin} < t_{WCS}$

This is a nightmare right off the bat. Here you're basically saying that the PSF is the fastest-changing "thing" in the equation, meaning that it's going to be moving throughout everything. Not only that, but in contrast to previous configurations where the PSF motion would still be seen in spite of that motion being relegated to a subset of the frames in the output, there will be PSF motion through ALL frames in this configuration.

VERDICT: This configuration is BAD. The only solution I see to it is to (1) warn the user immediately that they've made a bad choice and (2) set the t_{WCS} down to t_{PSF}. Option 2 can be optional, but then the user just needs to be notified of their misdeeds and have to either use the bad output or fix it. Pay for your sins and may god have mercy on your soul! and your data

IN CODE: Throw warning

1.6. $t_{PSF} < t_{WCS} < t_{bin}$

This configuration appears similar to the other option where t_{bin} is the longest timescale in the system. In contrast to that mode (where a derotated option should perform alright but ADI mode will be degraded), we would expect both derotation and ADI modes to be problematic here.

The ADI mode would still exhibit the same trouble as in 1.4: you'd see blurring or smearing of the PSF within the frame since there is more than 1 t_{PSF} elapsed within the frame.

However another issue will show up that will make the derotation data worse and add an additional complication to the ADI data. Namely, this is that the PSF will be moving before the WCS is recalculated, meaning that each WCS realization will have some smudginess to it. Whether this is apparent by eye or not, it is still a degradation of the data and needs a fix. The potential fix that I see here is to set t_{WCS} equal to t_{PSF}. This will functionally take you back to where you were with case 1.4, removing the issues seen in the derotation mode but still leaving a problem for ADI where you have inter-frame smearing. Similarly to 1.5, this could be mitigated by not allowing the user to do it at all, or by sending the user a warning that this configuration stands a good chance at mucking up their data.

VERDICT: Not great. Seems to present problems for any mode the user may want.

IN CODE: Throw warning

2. DISCUSSION

The different timescale configurations seem to fall into the categories of 'Not an Issue', 'Potential Issues Here', and 'Always an Issue'.

The configurations that appear to always be okay are

- $t_{bin} < t_{WCS} < t_{PSF}$
- $t_{WCS} < t_{bin} < t_{PSF}$

While they do present opportunities to be somewhat computationally wasteful, they should not need any finagling from the code to make the data okay.

The configurations that present potential issues are

- $t_{bin} < t_{PSF} < t_{WCS}$
- $t_{WCS} < t_{PSF} < t_{bin}$

primarily I see these issues occuring in the first case in a subset of the output frames where you see some smudginess when t_{PSF} has elapsed but t_{WCS} has not. In the second case, I think there is the potential issue in ADI mode where the user will see in-frame smudging (to different degrees depending on the discrepancy between the two shorter timescales).

The configurations that I foresee ALWAYS causing an issue are

- $t_{PSF} < t_{bin} < t_{WCS}$
- $t_{PSF} < t_{WCS} < t_{bin}$

It's fairly clear to see why. If the PSF is zipping around the star and we don't do anything to remove that motion, it's going to cause us fits when we reduce the data. I think the course of action here is either to (1) allow it but yell at the user and output crappy outputs, (2) disallow these entirely and not output anything (still telling the user they're in a wonky configuration), or (3) quietly fix the mistake by reducing t_{WCS} appropriately and reducing them both to $t_{PSF} < t_{bin}$, which in turn gives the same potential issues as condition ($t_{WCS} < t_{PSF} < t_{bin}$).

3. TESTING

Here I'll just put the timescales we have available to us and how I'll test. First and foremost, I'm using an on-sky dataset from Hip 99770 data from October 2021. The one timescale we have a maximum on here is that $t_{bin,max}$ ~15s. So, that is what we'll roughly anchor everything else to.

To generate the data with the proper t_{PSF}, we first choose a separation for our object to look at. 0.″3 separation seems as good a choice as any. It's not too close to be swamped by the speckle halo and not so far that it'll leave the array for long times. At this separation, the object needs to rotate ~2.0° to 'smear' by 1 pixel. This doesn't seem enough for us to visually tell (that well) that smearing is occurring. For our purpose, we'll call the smearing distance 3.5 pixels, which is a rotation of 6.9°[1]. Using the relation that

$$t_{PSF} = \frac{Allowable\ Pixel\ Smear}{Rotation\ Rate} = \frac{\theta_{smear}}{R_{rot}} \qquad (1)$$

we invert that to find our rotation rate should be $R_{rot} = \theta_{smear}/t_{PSF}$. This equality gives us the rotation rate for a desired t_{PSF}, and we use this to generate a faux PA list to inject our fake planet into the real data with.

3.1. $t_{bin} < t_{PSF} < t_{WCS}$

- $t_{bin} = 1$ s
- $t_{PSF} = 8$ s
- $t_{WCS} = 20$ s

For this, r_{rot}=0.8625°/s.

3.2. $t_{WCS} < t_{PSF} < t_{bin}$

- $t_{WCS} = 1$ s
- $t_{PSF} = 8$ s
- $t_{bin} = 15$ s

For this, r_{rot}=0.8625°/s.

3.3. $t_{PSF} < t_{bin} < t_{WCS}$

- $t_{PSF} = 1$ s
- $t_{bin} = 15$ s
- $t_{WCS} = 50$ s

For this, r_{rot}=6.9°/s.

3.4. $t_{PSF} < t_{WCS} < t_{bin}$

- $t_{PSF} = 1$ s
- $t_{WCS} = 5$ s
- $t_{bin} = 15$ s

For this, r_{rot}=6.9°/s.

To note, the injected companion for 3.1 and 3.2 (r_{rot}=0.8625°/s) is stored on `glados` at `/work/nswimmer/20211016/injectp/rrate0/` and for 3.3 and 3.4 (r_{rot}=6.9°/s) is stored at `/work/nswimmer/20211016/injectp/rrate1/`. Due to length of time it takes to inject fake planets, we're only using the first 10 dither steps. We will make ADI and derotated temporal drizzles for each of the 4 timescales of interest.

[1] This was done with simple trig arguments, not the 'non-blurring max' that drizzler uses. We just noted the distance we'd call the smearing distance (3.5 pixels), the separation (0.″3), and found the angle the sky would need to rotate for an object to travel that much. Strictly speaking, this is a linear estimate using right triangle math (θ =arctan(3.5 pix / (0.″3 / (10.4 mas/pix)))), but we just care that the smearing is visible to the users eye while allowing us fine enough control over t_{PSF} so as to be able to test these different regimes.

4. NOTES

The outputs in both ADI and derotated mode temporal drizzle cubes have been created for 3.2, 3.3, and 3.4. The pipeline is running into errors in creating the temporal drizzle for 3.1.

5. RESULTS

In this section we'll talk about / show results from actually putting observations with these timescales through the pipeline.

The intermediate products live in a few different places due to the need to trick the pipeline into creating outputs from h5s with the faster-moving planet, slower-moving planet (both injected), and no planet (for reference) as well as injected fake companions in the first place. The 'final' output products live on `glados` in `/work/nswimmer/20211016/injectp/hip99770_dither_0/`. Each is specified with the 'case' number corresponding to where it is denoted in section 3 (i.e. Subsection 3.1 is called 'case3-1'), has each timescale specified in the file name, and then specifies whether it was reduced in ADI mode or not. For output files with *no* planet injected, 'tpsf' is replaced with 'noplanetinjected'. For example, one file is titled "`hip99770_dither_0_case3-2_tpsf8_twcs1_tbin15_adi_drizzle.fits`", which refers to the case from section 3.2, where $t_{WCS} < t_{PSF} < t_{bin}$, $t_{WCS} = 1$ s, $t_{PSF} = 8$ s, $t_{bin} = 15$ s, and ADI mode was set to True. The 'reference' output with no planet injected will have the same title, but with 'tpsf8' replaced.

For ease, 'custom' fits with the format `c3-N_derotatemode.fits` and `c3-N_adimode.fits` have also been made in the same directory where the fit['CPS'].data attribute is the difference between the injected planet output and the identically created output (same timescales) without any planet injected.

5.1. $t_{bin} < t_{PSF} < t_{WCS}$

As of the testing configuration that has been made, there is not any clear and obvious smudging at any point with either the ADI or Derotated modes. The time slices do show discrete motion of the PSF (it 'jumps' between frames because we're causing it to really zip around at slightly less than 1°/s), but none of the frames show smudging when the WCS is recalculated.

Where might this *still* be an issue?

- Drizzling a single image (non-temporal) the user may see smudging (or stamping, depending on rotation speed) because a single WCS solution has a lot of PSF movement.

5.2. $t_{WCS} < t_{PSF} < t_{bin}$

As expected, the Derotate mode reproduces the PSF without significant (or visible at all) distortion, while the ADI mode shows significant smearing (as expected).

Using this configuration in which you have a timestep that encompasses multiple PSF blurring timescales will causing blurring in ADI mode even if the WCS time is shorter than the PSF blurring timescale (which is as designed). Therefore, things work as expected, and this is a use case that should be avoided, especially because for ADI you want to capture rotation *between* frames rather than *within* a frame.

5.3. $t_{PSF} < t_{bin} < t_{WCS}$

Both cases as expected result in smudging and smearing of the PSF. Regardless of performing ADI or Derotation, this mode will be smeary/jumpy and the PSF is going to move around a whole bunch when it really shouldn't.

5.4. $t_{PSF} < t_{WCS} < t_{bin}$

Again - as expected - both cases cause significant smearing/jumping within frames/ In contrast to the previous case, the WCS solution being recalculated within the t_{bin} duration and so the single frames do not show *as much* smearing, but they do still smear and distort the PSF significantly.