Review of “TrackMatcher” GMD manuscript by Bräuer & Tesche

Summary:

The manuscript presents a tool to find intersects of data points in two types of data sets (as exemplified by aircraft or cloud tracks and satellite data with narrow swath).

The manuscript is well-written and concise. The problem that is solved and the algorithm itself are not overly complex, but I believe still worth to be published in such a separate publication as this give room for sensitivity studies that are do usually not appear in other journals.
I have only some minor concerns and recommend accepting the manuscript once those are resolved.

Minor issues:

1. Some sections I regard as too technical, even for GMD standards. The dependencies in section 2.2 could be removed from the paper and only listed on the GitHub/Zenodo README file. Furthermore, section 2.3. is very technical, at least for someone who is not familiar with Julia. It is good to define what is meant with all those technical terms that are written in italics. But I question a bit the necessity to talk about constructors at that stage. If a reader finds this information helpful clearly depends on whether he/she wants to understand how your algorithm works or intends to use or extend the existing code.
2. How/where is the term „continuous“ used in line 42 defined?
3. Around line 200: It is not clear to me, how the width and height of the bounding boxes are computed.
4. Fig. 2: In your example, 4 dots are yellow (acceptable time diff) and one is red (delay to large). Is this typical that spatial intercepts have an acceptable delay? I would imagine, that in most cases, the time difference of the collocated observations is too large. Or this example a special case where aircraft needs about the same time as the satellite to jump from one satellite track to the next?
5. line 350: To me it does not sound reasonable that using a different interpolation scheme increases the number of intercepts. I understand that the coordinates of interceptions may change. Moreover, it is conceivable that intercepts are only identified for one of the two interpolation schemes. But I do not see why Hermitian interpolation give such a strong bias towards more intercepts. Could it be that there was bug in the previous algorithm or some other parameters may have changed between the two version you compare? What happens, when TrackMatcher alternatively uses a linear interpolation scheme?
   With the new intercept data, do the findings of Tesche et al, 2016 still hold? Can you give at least a short summary on this in this manuscript?
6. line 451-452: ambiguity of the term performance: do you mean run time or the algorithm success? Apart from this wording issue, I do not understand, why your algorithm does not work well for a finer resolution.
7. comment to line 460: One could also do the computations in double precision and the output in single precision.

Technical issues:

1. The abbreviation ESM is nowhere defined.
2. line 146: Eq 1??
3. line 169: “source code” or “source code execution”?
4. last line of page 7: acceptable

5. line 216: replace “set” by “specified”
6. line 385: it would fair to write “2.5 min”
7. Caption of Fig.4: Gridded map of THE NUMBER of intercept points?
8. line 479: pointS
9. line 483: spational -> spatial?