

## Comment on egusphere-2022-1136

Anonymous Referee #2

---

Referee comment on "PyFLEXTRKR: a Flexible Feature Tracking Python Software for Convective Cloud Analysis" by Zhe Feng et al., EGU sphere, <https://doi.org/10.5194/egusphere-2022-1136-RC2>, 2022

---

The paper presents an open-source framework (pyFLEXTRKR), describing an atmospheric feature tracking software package in python language, which is able to track meteorological objects at different level of processes from deep convective cells to mesoscale convective systems. The authors show that the tracking algorithm can be apply on various observation systems (3D radar reflectivity, geostationary infrared observation) and models (Permitting Models, Large-Eddy Simulations), at different spatial resolution (from large-eddy resolving (~100m) to mesoscale (~10km)).

The tracking software package is based on two steps, an identification step on a single image and a tracking step based on an area-overlap technique. Different identification methods and tracking techniques can be selected according to the type of meteorological feature studied and the type of observation/simulations used.

Critical questions/issues must be addressed before being accepted. Here are major questions:

We thank the reviewer for the constructive comments. Below please find our response to each comment in blue, and we have revised the manuscript accordingly.

### Major comments:

**1) Line 50** The authors assert: “The Tracking Of Organized Convection Algorithm through a 3-D segmentation (TOOCAN, Fiolleau and Roca, 2013) tracks convective systems using satellite IR brightness temperature (T<sub>b</sub>) data. Their underlying technique uses area overlap between successive T<sub>b</sub> images for tracking and is similar to those first developed nearly three decades ago (Williams and Houze, 1987; Velasco and Fritsch, 1987; Laing and Fritsch, 1997; Machado et al., 1998).” and line 115: “This simple overlap tracking technique has been used in previous studies (e.g., Williams and Houze, 1987) and other tracking software (e.g., TOOCAN, MTD, TAMS).

This is a misleading statement. I urge the authors to read once again the Fiolleau and Roca publication describing the TOOCAN algorithm published in 2013, to present their works at fair values and to correct the manuscript. Indeed, page 3 of Fiolleau and Roca, It is written: “The new tracking algorithm then works in a time sequence of infrared images to identify and track MCSs not anymore with the traditional detection and tracking steps but in a single 3-D (spatial+time) segmentation step. For this purpose, a spatiotemporal image, whose spatial axes are longitude and latitude, is generated by the time series of infrared images derived from geostationary satellites.”

Thus, TOOCAN is not a tracking method based on a area-overlap technique as described in Williams and Houze, 1987; Velasco and Fritsch, 1987; Laing and Fritsch, 1997; Machado et al., 1998.

The error has to be corrected in the manuscript.

We thank the reviewer for pointing out the details of the TOOCAN methodology. We have revised the statements on line 50-55 as follows:

“The method for object-based diagnostic evaluation time-domain (MODE time-domain, or MTD, Clark et al., 2014) (<https://met.readthedocs.io/>) uses area overlap techniques for tracking convective objects, similar to those first developed nearly three decades ago (Williams and Houze, 1987; Velasco and Fritsch, 1987; Laing and Fritsch, 1997; Machado et al., 1998). The Tracking Of Organized Convection Algorithm through a 3-D segmentation (TOOCAN, Fiolleau and Roca, 2013) tracks convective systems using satellite IR brightness temperature ( $T_b$ ) data. **Their technique uses 3-D segmentation on a sequence of  $T_b$  images (spatial and temporal) and improves upon traditional area overlap techniques.**”

**2) Concerning the splitting and merging issues, It has been shown for years that the tracking algorithms based on an identification step and an area-overlapping tracking technique lead to splits and merges artefacts. As discussed in Machado et al (1998), the frequency of the unphysical splits and merges is dependent of the time–space resolution, the area-overlap criteria used and a minimum detected cloud area...**

I invite the authors to discuss on these criteria used in their algorithm and their limitations to track meteorological objects according to the type of features, to the observation systems, to the models and the time-space resolution. For instance, are the area-overlap criteria similar for a tracking of convective cells from radar observation and for a tracking of convective systems from geostationary satellites?

We thank the reviewer for pointing out the importance of temporal and spatial resolution of the dataset and the area-overlap criteria to tracking. In PyFLEXTRKR, minimum detected cloud area and area-overlap fraction are both user-defined thresholds. Area-overlap fraction is calculated in both temporal directions (forward and backward) to account for objects that are either growing or shrinking in size.

On lines 122-126, we have stated the following:

“If their overlap area fraction exceeds *the user defined threshold*, their label numbers are recorded in a correspondence pair. ... The underlying *assumption with the overlap technique* is that *the temporal resolution of the dataset is sufficient to resolve the spatial movements of the features*, and that *objects with sufficient overlap between two timesteps belong to the same feature*. In PyFLEXTRKR, the overlap area fraction is calculated *in both temporal directions* (i.e., from time 1 to time 2 and from time 2 to time 1), such that objects that are either growing or shrinking in size are considered.”

We added several sentences to emphasize the importance of temporal resolution of the datasets on lines 130-135:

**“It is important to note that the accuracy of feature tracking for any automated technique hinges upon sufficient spatial and temporal resolution of the dataset to resolve the feature of interest. While higher resolution dataset (particularly in the temporal dimension) is typically desired for more accurate tracking, lower resolution datasets (e.g., earlier generations of geostationary satellite  $T_b$  images, large-scale model simulation outputs) are more widely available. Caution is needed when applying PyFLEXTRKR to lower temporal resolution datasets and specific examples for tracking convective cells and MCSs are further discussed in Sections 3 and 4.”**

For specific usage examples, we added the following texts (in bold):

Lines 264-269:

**“Our previous work developing the C-band radar convective cell tracking database during the CACTI field campaign used an area overlap fraction of 0.3, which seems to work well with the advection technique and the 15-min volume scan update (Feng et al., 2022).** PyFLEXTRKR thus enables individual convective cell tracking on observations and model simulations with radar reflectivity outputs at 15 min or shorter for at least most deep convective situations. This capability reduces the need for model simulations to output very frequent 3D radar reflectivity data (e.g., 10 min or less) but still allows tracking of individual convective cells.”

Lines 345-347:

**“After CCSs are identified, they are tracked by running the first four steps in Fig. 1. For tracking DCSs using satellite-observed  $T_b$  or model-simulated OLR, an area overlap fraction of 0.5 typically works well for datasets with hourly temporal resolution and  $O(10\text{ km})$  spatial resolution, although finer resolution data is preferred for more accurate tracking.”**

**3) The authors say that the tracking algorithm handles merging and splitting explicitly. Splits and Merges are then flagged so that the users are able to reassemble the whole lifetime behaviors.**

The authors have to discuss on what basis the users can reassemble a complete life cycle of a meteorological feature. What does a split and merge event mean for convective cells? what is the frequency of such events? Are splits and merges physically possible at the cell level or the result of an algorithmic problem? Similarly, at the convective system level, what does a split and merge event mean? Is it the result of a split and merge of their cirriform clouds, of their convective core parts? At the convective system level, on what basis, the users can certify that splits and merges are due to an algorithm issue or not?

We have added more details regarding what information related to merging/splitting is saved on lines 155-159:

**“In addition, the start and end status of each track including whether a track starts as a split or ends as a merge, the track number it splits from or merges with, and the time and object number associated with the merger or split are all recorded as 1D arrays for each track. These variables enable users to make decisions on how to treat these merge/split tracks based on specific scientific needs. An example of including merger and splits as parts of the identified MCS cloud shields is provided in Section 4.3.”**

Merger and splits occur physically in deep convective scenes. High resolution radar observations (e.g., U.S. NEXRAD radar observations with 2-5 min volume scan updates) show individual convective cells nearby can merge and grow upscale into larger convective clusters; similarly, large convective cells can split into multiple smaller cells. On the other hand, merger and splits can also be a result of insufficient temporal resolution of the dataset and/or algorithmic issues.

For example, the “region growing” technique (used in PyFLEXTRKR and TOOCAN, although differing in 2D vs. 3D implementation) applied to satellite  $T_b$  images typically identifies a contiguous area with low  $T_b$  as a separate “cold core” and subsequently grows outward to include surrounding regions with higher  $T_b$ . This approach could artificially segment a convective system that has multiple “cold cores”, which could result in merging/splitting in overlap-based tracking techniques.

In PyFLEXTRKR MCS tracking, short-lived (user-defined duration threshold) non-MCS tracks that merge with or split from an identified MCS track are included as part of the MCS. This treatment helps reduce large fluctuations of MCS cloud shield area due to merging/splitting or cloud segmentation artifacts. We have added a new schematic Figure 8 to illustrate this process along with the revised texts on lines 349-355 (bold text are revised):

**“Short-lived (a user-defined duration threshold) non-MCS tracks that merge with or split from the identified MCSs are also retained. The labeled cloud numbers and sizes from these merge/split tracks at the same corresponding times with MCSs are stored in the MCS track statistics file, which allow their areas to be included in calculating the total MCS cloud shield area by users. This treatment helps reduce large fluctuations of MCS cloud shield area due to merging/splitting or cloud segmentation artifacts in Step 1. The cloud mask associated with these merge/split tracks are included as part of the MCS cloud shield in subsequent steps (see illustration in Fig. 8).”**

**4) Finally, I think It would be relevant to warn the future users of the importance to perform some quality control on the data, some inter-calibration procedures, and data harmonization... before applying a tracking software.**

Thank you for the suggestion. We added a new Section 4.5 on page 19 to point out the importance of data quality control and briefly describe a few simple procedures included in PyFLEXTRKR for this purpose:

#### **“4.5 Quality Control on Input Dataset**

**The accuracy of convective cloud tracking could be affected by various issues in the input dataset, such as missing data and various data artifacts (e.g., bad data, calibration error, etc.) often found in observations. Thus, data quality control is an important factor to consider before applying feature tracking to observations. While advanced data quality control is beyond the scope of PyFLEXTRKR, a few procedures have been developed for tracking DCSs using satellite  $T_b$  data. For example, a median filter (using the `scipy.signal.medfilt2d` function) with a user-defined window size is applied to the  $T_b$  images to fill in pixels with missing  $T_b$  data that can occur with bad scan lines from satellite imagers. An acceptable range of  $T_b$  data can be defined by the user to filter out unphysical data (e.g.,  $T_b$  must be between 160 K and 330 K). These simple procedures help reduce cloud object identification errors due to satellite data quality issues. In addition, satellite images with missing data larger than a fraction of the domain (user-defined) are skipped in PyFLEXTRKR, and if the time gap of missing satellite images is larger than a user-defined threshold (e.g., 3 hours), tracking is stopped and restarted from the next available  $T_b$  image to avoid erroneous tracking. For convective cell tracking using radar reflectivity data, no quality control procedure is currently implemented in PyFLEXTRKR. Users should perform quality control of radar observations prior to applying PyFLEXTRKR to track convective cells.”**