Monitoring the phenology of individual flowers using automatic tracking and filtering

Hjalte M. R. Mann

Alexandros Iosifidis

Toke T. Høye

January 13, 2022

ABSTRACT | Often simple variables will be used to describe the flowering phenology of a population of plants, e.g. onset or peak of flowering. Here we show that image-based monitoring of field plots at very high temporal resolution can return information on flowering phenology at the level of individuals. Further, we present a framework for automatically tracking, filtering, and evaluating flowers in time-lapse image series. We compare the results of automatic tracking with manual tracking.

# Introduction

For the individual plant, timing of flowering is of utmost importance. Precocious flowering means that the plant has failed to exploit the whole temporal window for accumulating resources before allocating energy to flowering. Oppositely, flowering too late limits the time for reproduction before the end of the growing season (Elzinga et al., 2007). Further, flowering may need to be synchronous with pollinator activity for successful reproduction. Flowering phenology may plastically change as a response to abiotic cues in the environment, such as timing of spring, temperature, and photoperiod, but variation in flowering phenology is partly heritable and shaped by selective forces from the abiotic and biotic environment and thus sensitive to climate change.

Monitoring of flower phenology at high temporal resolution is laboursome and time-consuming, particularly in logistically challenging environments such as the Arctic. Consequently, simple variables are often used as proxies for the flowering phenology of a population, such as the date for onset of flowering, often derived from weekly observations of sample plots. Such proxies may fail to reveal dynamics in flowering phenology for example caused by changes in climate.

Automatic image-based monitoring of flowering phenology can return phenology data for specific species at very high temporal resolution (Mann et al., in prep), but phenological responses at the individual level may be indiscernible regardless of the temporal resolution of the data at population level. For example, a shortening of individual flower longevity may not be directly obvious at the population level. In fact, many research questions can only be explored on the basis of individual phenology data. For example, investigating the association between reproductive success and timing of flowering or flower longevity requires phenology data at the level of individuals. Similarly so, for investigating whether flower visitation rates and/or reproductive success depends on the timing of flowering for the individual flower.

Here, we show that information on phenology at the level of individuals can be derived from image-based monitoring of flower phenology.

We present and evaluate an automatic flower tracking and filtering algorithm. Many tracking algorithms, such as the Simple online and real-time tracking (SORT) (Bewley et al., 2016), use Kalman filtering to predict the future location of an object by estimating its velocity (Kalman, 1960). However, Kalman filtering assumes linear motion of the object and does not handle abrupt motion well. As the flowers are constrained in movement by their stalk, their movement does not follow the assumption of linear motion. We therefore base the track assignment on the minimum distance between centroids of bounding boxes. However, we introduce three user-set parameters that can help optimise tracking performance.

For complex scenes with many flowers in close vicinity to each other, we suggest a conservative filtering approach. The approach may remove correct tracks, but the tracks that remain will have a low risk of tracking errors. The approach allows for confident upscaling of the method. I.e., when running the method on a large number of image series, it is preferable to extract individual high-confidence tracks from each series and ignore the remaining tracks.

Tracking individual flowers enables the possibility of assigning reproductive success to the individual, for example by observation of seed set. Thereby, it could be explored whether reproductive success is affected by timing and length of flowering. By simultaneously tracking flower visits, these could be assigned to the individual flower and visitation rates per flower could be calculated per flower and related to reproductive success. Further, any information of taxonomic grouping could refine these analyses. We argue that the possibility of tracking phenology at the level of individuals will facilitate investigation of a novel set of research questions and is an important contribution to the shift towards efficient and automatic monitoring of ecological processes.

# Material and methods

# 

## Study site

Toke, perhaps you can fill in some stuff here. NARS 2018, NARS 2019, NYAA 2019, THUL 2018.

# 

## The image series

Original time-lapse intervals, explanation and result of sampling scheme and subsequent temporal resolution.

# 

## Flower annotations

We manually annotated all flowers in the sampled image series using the rectangular bounding box tool in the VIA VGG annotation software [?] and assigned each individual flower a unique ID. These annotations constitute our ground truth tracks.

# 

## Automatic tracking

We built a framework for tracking, filtering, and evaluating tracking of objects in time-lapse image series.

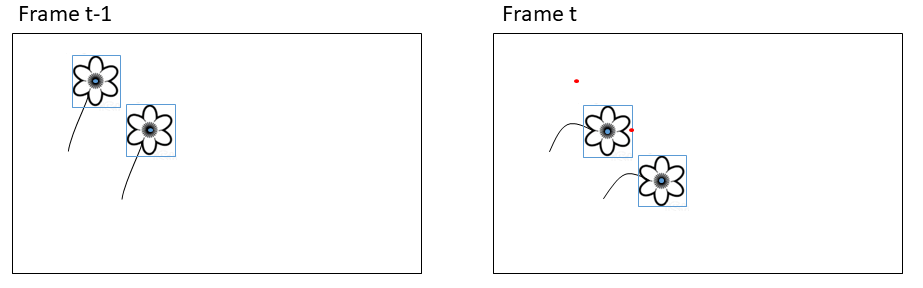
The tracking algorithm tracks objects based on distances between centroids of bounding boxes. The algorithm has a set of user-adjusted parameters that can optimize tracking accuracy. The parameters are particularly relevant for optimal tracking of objects that are constrained to a specific area such as flowers. It is important to note, however, that the tracking algorithm could be used to track any objects. The algorithm can be applied both offline (on a set of detections/annotations that have already been produced) or online (real-time tracking frame per frame). The speed of the tracking algorithm depends on the computational power available as well as the number of objects that are being tracked. The method is fast, however. Tracking of a series containing 85 objects ran at ~0.02 seconds per frame on a standard laptop.

Several things complicate the task of tracking individual flowers through the time-lapse series. First, as the wind shifts, the flower heads changes direction. This can happen instantaneously (i.e., between two consecutive frames). However, as they are constrained by their stalk, there is a limit to the distance they can move. Establishing associations between points based on just the distance between points in the current and the previous frame can cause errors when flowers are in close vicinity of each other. With the tracking parameter **running mean**, we base the tracking on the distance between a point in the current frame and the running mean of the positions of the previous X points in a track.

Second, when the location of the flowers changes, they may temporarily occlude the view of other flowers and flowers close to the edge of the frame may move in and out of view. This can cause a track to be lost and a new track erroneously being initiated when the flower reappears. However, if a flower reappears in the same area as a flower is already being tracked after disappearing in a few frames, it is a reasonable assumption that it is the same individual and not that the old flower wilted/disappeared and a new one developed. The parameter **max disappeared** sets the number of frames a track can be lost before a new track is initiated for points appearing in the same area. Concurrently, this deals with potential false negatives. If a given flower has not been annotated in a few frames, the track will not be lost. The counter for number of disappeared frames for a track is reset when a new point is associated with the track within the threshold.

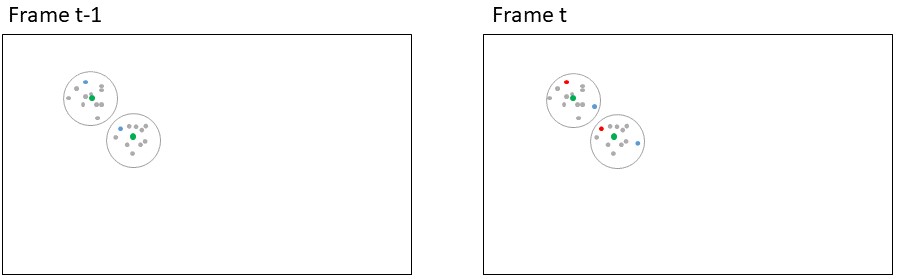
Third, a centroid tracking algorithm will associate a point with a tracking based on minimum distance only, disregarding the absolute distance. As flowers are constrained in the movement, we can assume a maximum distance that will occur between points and force the initiation of tracks for points that exceed this threshold. We do that with the parameter **max distance**.

# 



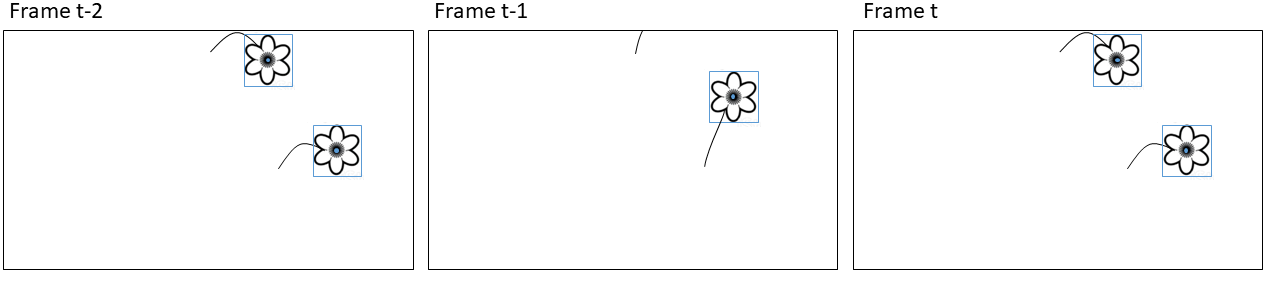
**Figure 1:** Simple centroid tracking may produce erroneous associations when objects move between frames. Blue shows detections in the current frame (bounding box and centroid point). Red shows centroid points for the detections in the previous frame.

# 



**Figure 2:** Simple centroid tracking may produce erroneous associations when objects move between frames. Basing the association on the running mean of the positions of the previous n number of tracks may alleviate this issue. In this case basing the association on only the previous point would produce a wrong result while basing it on the running mean would produce a correct result. Red points: Centroids for bounding boxes in current frame, blue points: Centroids for bounding boxes in previous frame; grey points: Centroids for bounding boxes in a number of frames before t-1; green points: Running mean of the previous n points. Circles delimit the two individuals.

# 



**Figure 3:** Simple centroid tracking may produce erroneous associations when objects disappear periodically from the frame. Here the top flower moves out of frame and the bottom flower would be assigned to the track of the top flower in frame t-1.

# 

### Identifying optimal user parameters

To explore the effect of the user parameters and to identify the optimal combination of parameters for our case of tracking flowers, we ran the tracking algorithm on each of the five image-series with every combination of a range of values for each parameter (3.179 combinations): Values for max disappeared and running mean were 0-160 with a step size of 10 and for max distance we used values 0-1000 with a step size of 100. Note that set to zero ignores the parameter altogether. We identified the setting(s) that returned the lowest number of track mismatches and compare the tracking results between optimal settings and all parameters set to zero.

Additionally, we performed a set of analyses on the five ground tracks. First, we calculated the largest distance between any two points within any track for any flower in each of the five series. Further, we calculated the maximum number of frames a flower track was lost and subsequently reappeared for each series.

# 

## Evaluating tracking performance

The optimal way of quantifying tracking performance depends on the goal of the tracking. To associate other information obtained in the images to the individual flower, for example flower visits, we want as much as possible of the track to be correct. To derive flowering length, in theory we just need to track the most extreme points correctly and can ignore the intermediate points. Lastly, we may be interested in the number of flowers that existed in a plot, in which case we want the number of tracks obtained by automatic tracking to be as close as possible to the actual number of individuals in the series.

The multiple object tracking accuracy (MOTA) score quantifies tracking performance based on counts of tracking mismatches (Bernardin & Stiefelhagen, 2008). Mismatches occur when objects swap track identity because they are in close vicinity to each other or when an object periodically disappears and is assigned a new track identity when it reappears. Only the shifts in tracking identity are counted as mismatches while the number of points assigned to each track is not considered. We calculate the number of mismatches and the MOTA score to evaluate performance of our tracking and filtering algorithms. Further, we compare the number of tracks identified by the automatic tracking with the true number of flowers in a series. These should ideally be equal.

# 

## Filtering tracks

When deploying the automatic tracking algorithm on naive data without ground truth tracks, it is not possible to manually filter for correct tracks. Therefore, we present a conservative filtering method that extracts the most trustworthy tracks from a scene.

We base our filtering on a density-based clustering of the centroid of a geometry of each track. The geometries are derived in the following way: For tracks consisting of only a single point, the coordinates of the point are used as the centroid. For tracks consisting of two points, we establish the straight line between the points and calculate the centroid of the line. For tracks consisting of three points, we establish the triangle from the points and calculate the centroid of the triangle. For tracks consisting of more than three points, we calculate the convex hull of all the points included in the track and derive the polygon from the vertices of the convex hull and calculate the centroid of this polygon. For tracks that contain more than two points but where the points a colinear, we establish the line through the points and calculate the centroid of the line.

We then apply the filtering based on the geometry centroids to remove tracks in areas with a high density of tracks as these have a high risk of tracking mismatches. The filtering is done in two steps. First, the DBSCAN clustering algorithm is run on the centroids with a conservatively high value for the eps parameter (350), meaning that tracks in close vicinity to each other will be clustered together. Second, all tracks that were not assigned to a unique cluster are removed. Tracks that are spatially isolated remains. We evaluate the tracking accuracy of the remaining tracks. We demonstrate our filtering approach for the image series for which our tracking algorithm did not produce perfect results.

# 

# Results

The results of the tracking parameter test are given in table 1. We present the tracking performance without the use of the three tracking parameters along with the best performance (lowest number of mismatches/highest MOTA), and the best performance where the number of tracks correspond to the number of individual flowers in the series.

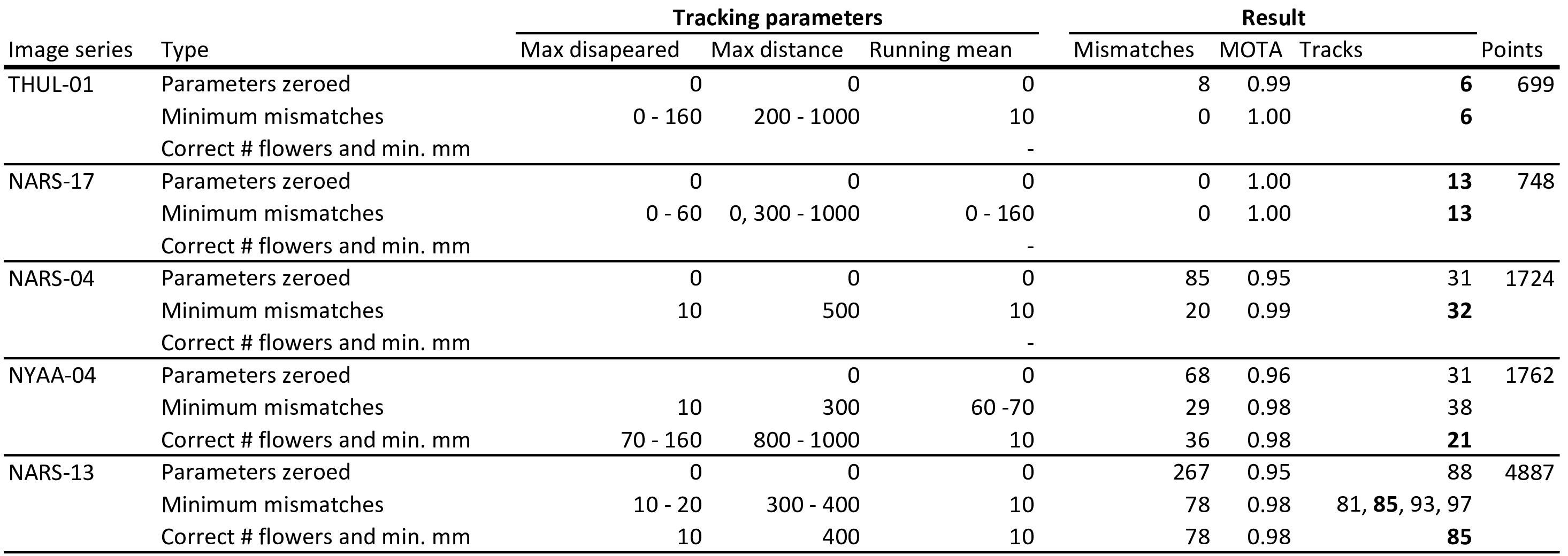
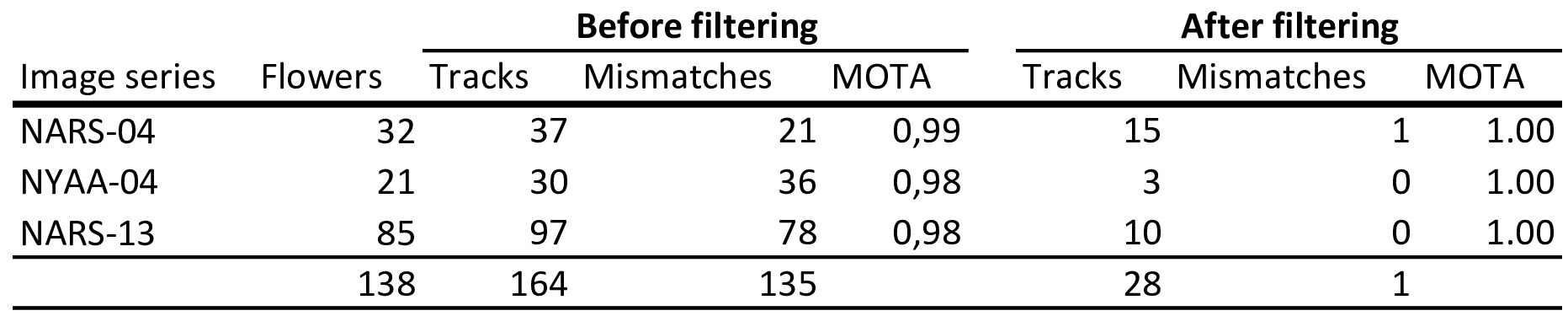
**Table 1:** Table description… # 

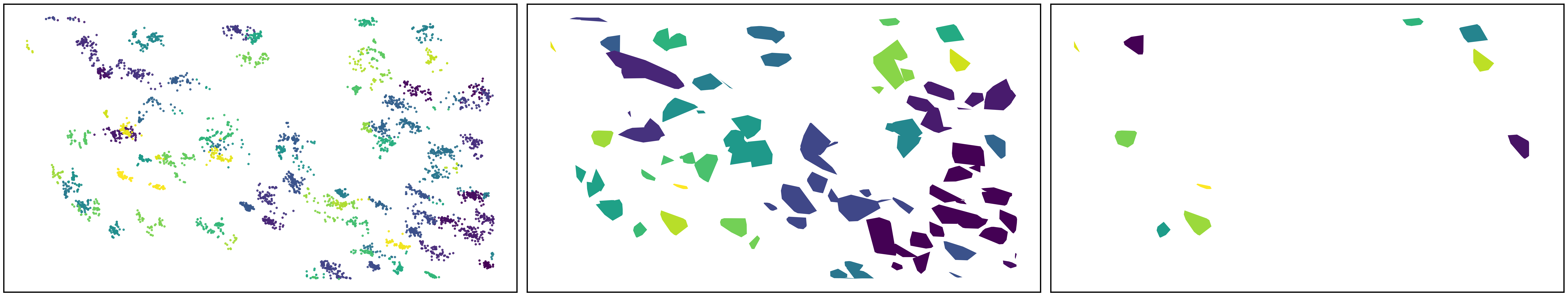
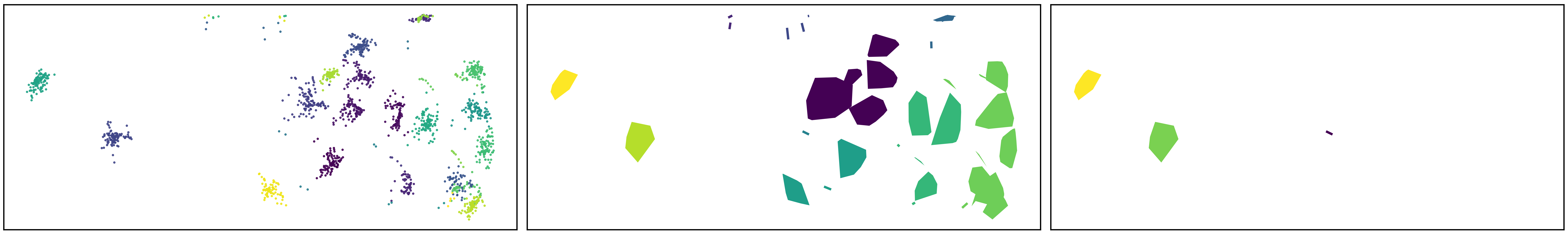
Table 1 shows the performance of the tracking algorithm.

The results of the filtering algorithm on the three series in which our tracking algorithm did not return perfect results are given in table 2 and the filtering process is visualised in fig. 4. Our filtering method successfully extracted 28 tracked flowers with only a single mismatch from the three series with complex scenes using a fixed value of 350 for eps and fixed values for the tracking parameters (10, 10, 300, for running mean, max disappeared and max distance, respectively).

**Table 2:** Results of the filtering algorithm with an eps value of 350 applied across all three . Table 1 series.



# 

**Figure 4:** Track filtering process. Row 1, 2, and 3 is NARS-13, NARS-04, and NYAA-04, respectively. First column shows the centroid points in the given series, coloured by the track id from the centroid tracking algorithm. Second column shows the polygons calculated from the tracks. The DBSCAN clustering algorithm with eps = 350 was applied to the centroids of these polygons and the polygons are coloured by cluster id. Third column shows the results of the filtering where all tracks that were not assigned a unique cluster has been removed.

# Discussion

## Tracking

We ran our tracking algorithm on 3xxx combinations of the three parameters. Using the parameters increased the performance of the tracking substantially. Our steps in parameter values were crude, however, and it is very likely that finer steps in these values would identify combinations that produce even better results. Here we do not perform this analysis, however, as our goal is to show that the parameters can be used for optimizing tracking performance in general.

Depending on the nature of the objects being tracked and the complexity of the scene, the user parameters can be estimated from visual examination of the tracking results. Often it may be preferable to manually annotate a subset of the objects in the image series and derive a set of user parameters from these results.

Number of tracks equal to number of flowers does not necessarily mean that the tracking is correct. I.e., the tracking algorithm could split one ground truth track in two, but combine two other ground truth tracks into one, which would even each other out.

Our tracking algorithm consistently returns high MOTA scores.

For complex scenes, all three parameters make a difference.

The parameters are interdependent.

In cases where we are tracking perfectly with maxDisap = 0, setting it any value will not make a difference. Not quite right. Explore more…

Paragraph about applying the technique on detection instead of manual annotations. Detections introduce false positives. Either manual or automatic quality control to remove these before tracking or after.

Our manually tracked data is ground truth, but for example when flowers periodically move out of the frame, this mimics false negatives. Similarly, when one flower occludes another.

Automatically detecting flowers would likely introduce a degree of false negatives which would decrease the MOTA score if max disappeared is set to zero. Introducing a value for this parameter can deal with the problem of false negatives.

Our tracking algorithm returned reasonably high MOTA scores even with all three parameters set to zero. The degree to which mismatches are accepted may depend on the application?

# 

## Filtering

Extracting tracks that are spatially isolated does not guarantee that the tracks are correct/without errors. However, as spatially isolated objects are easier to track, it increases confidence in the remaining tracks.

We applied a single value for the DBSCAN eps parameter in our tracking algorithm. We note that this value could be fine-tuned for improved results for the individual series (i.e. more flowers extracted without increased number of mismatches). However, as our goal here was to show that a single conservative value can be applied across series, we do not show those results here. In a naive setting, a general value could be chosen or the value could be adjusted for each series based on visual examination of performance or testing on a subset of data.

Our method for filtering tracks using DBSCAN on track centroids ensures that all tracks are given the same weighting in the filtering since each track is represented by a single point. In some cases e.g. if it is given a priori that an object will always appear in a minimum of two frames, then single point tracks can be filtered out. However, when such a priori knowledge is not accessible, a conservative approach as the one we present is preferable.

# 

## The image series

A point on the fact the these series are complex.

# 

## Conclusion

A paragraph about the ecological perspectives of being able to track the individual flowers (at scale).

# Acknowledgements

# Data availability

The code that supports the results in this paper will be made openly available at <https://github.com/TECOLOGYxyz/FlowerTracking>. A publicly available web application can be accessed from the Github repository through which users can run the tracking algorithm on their own data. Raw data as well as the trained flower detection model will be archived on <https://zenodo.org/>.

**NOTES**

* **Flowering phenology**
  + Importance of studying flowering phenology
  + Responses to climate change
  + Phenology of communities, populations, individuals
  + Traditional methods for studying
  + Onset of flowering says little about true distribution
    - Even true distribition of community says little about flowering lengths of individuals and for example how it varies accross the season.
  + Difficult to study at the individual level - requires high temporal resolution and keeping track of individuals
* **Image based monitoring**
  + Automatic, high temporal resolution, remote sites
  + High temporal resolution means that we can annotate individuals and get phenology of the individual
* **Tracking**
  + Offline and online
  + Online often coupled with CNNs that attempt to distinguish individuals from each other and recognize them through frames
  + Flowers appear very similar and
  + Many methods for offline tracking
  + Hungarian/Kahlmann filter ++ May not be applicable for objects that move weirdly, e.g. change directions between frames.
  + Tracking based on distance.
    - Good but has some problems
    - Two points always associated disregarding absolute distance
    - Tracks lost when objects disappear
    - Objects close to each other may swap tracks
* **Our solution**
  + Here we demonstrate a framework for automatic flower tracking and evaluation of tracking performance
  + Ground truth tracks

For each series, we calculate the ratio of flowers for which the automatic tracking algorithm returns the correct flowering length compared to the ground truth tracks.

# References

Bernardin, K., & Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, *2008*, 1–10.

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*. <https://doi.org/10.1109/icip.2016.7533003>

Elzinga, J. A., Atlan, A., Biere, A., Gigord, L., Weis, A. E., & Bernasconi, G. (2007). Time after time: Flowering phenology and biotic interactions. *Trends in Ecology & Evolution*, *22*(8), 432–439. <https://doi.org/10.1016/j.tree.2007.05.006>

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, *82*(1), 35–45. <https://doi.org/10.1115/1.3662552>