Supplementary Materials for *The Growing Strawberries* Dataset: Tracking Multiple Objects with Biological Development over an Extended Period

Junhan Wen[1] Camiel R. Verschoor[2] Chengming Feng[1] Irina-Mona Epure[2]
Thomas Abeel[1] Mathijs de Weerdt[1]

[1]Delft University of Technology, [2]Birds.ai B.V.
{junhan.wen,c.feng-1,t.abeel,m.m.deweerdt}@tudelft.nl, {camiel,irina}@birds.ai

In this appendix, we present more examples of the data. We further specify the information and metadata about the *Growing Strawberries* Dataset (*GSD*). We provide details about the parameter tuning and camera-wise performance of the algorithms. We also present more details of the dataset collection setup.

A. Hosting, licensing, and organization info.

A.1. Data Structure

```
The Growing Strawberries
├── GSD-Images
│   └── <year> (2021/2022)
│       └── <camera> (RGB/OCN-1/2/3)
│           └── img
│               └── *.jpg
├── GSD-Annotations
│   ├── <cam.>(RGB/OCN-1/2/3-'21/22).json
│   └── <cam.>(RGB/OCN-1/2/3-'21/22).txt
└── GSD-Sample
    ├── img
    │   └── *.JPG(with original filenames)
    ├── gt.json
    ├── gt.txt
    └── GSD-sample_video.mp4
```

A.2. License

GSD is released under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.

A.3. Terms of Use

By accessing and using *GSD*, users agree to comply with the terms and conditions outlined in the CC BY-NC-ND 4.0 license. Users are responsible for ensuring the appropriate use of the dataset in accordance with the license and any applicable laws or regulations.

A.4. Author statement

The corresponding authors state that they collected the data as described in this document and in the main paper. The authors have the right to publish this dataset. *GSD* is licensed under the CC BY-NC-ND 4.0 license. Users of this dataset are required to comply with the license terms, including providing proper attribution when using the dataset. We provide the dataset "as is", without any warranty or guarantee of its accuracy or reliability. We disclaim any liability for errors, damages, or consequences arising from the use of the dataset.

A.5. Hosting and Maintenance Plan

GSD is hosted and maintained on *4TU.ResearchData* Platform. It is published with a DOI: doi.org/10.4121/e3b31ece-cc88-4638-be10-8ccdd4c5f2f7.v1 for long-term accessibility and versioning under the CC BY-NC-ND 4.0 license.

B. Examples of data and annotation

B.1. Data Sample

We provide a small sample of *GSD* along with the original dataset, under the file folder *GSD-Sample*. The sample includes images collected from 2021-09-01 to 2021-09-02 by *RGB-3*, and the corresponding annotations in a coco-format JSON file and a TXT file compatible with the MOT evaluation tools. The images are with the original filenames assigned by the cameras when the photos were taken.

Along with the data sample, we also provide a short video to illustrate a subsequence of *GSD*. The video presents the growth monitoring of strawberries from 2021-09-01 to 2021-09-07 in *RGB-3*. In this video, two drastic location changes happened between 12-1 pm, 01.09, and between 5-6 pm, 06.09 because of the harvests. It could be noticed that strawberries 355 and 354 switched positions suddenly between frame 1-2 pm, 07.09, because of the harvest of 391.

This exemplifies the irregular movements of the *GSD* objects. The video is accessible in the *GSD-Sample* file folder.

B.2. Examples camera views and bbox annotations

Fig. 4 gives a sample view of each camera, collected at the same moment. The view of RGB and OCN cameras in each pair has a horizontal shift due to the parallel setup, as can be seen in Fig. 10. Some small dislocations among the images resulted from the camera shaking from practice. Many static reference objects can be found in the images for re-alignment of the two views.

B.3. Selection of image data

We have divided the set of images into a daytime subset, which has a brightness (Luma) of at least 50, and a dark image set which is not annotated. We calculated the brightness (Luma) of images according to [1], and here exemplify the brightness levels in Fig. 1. We illustrate the example images and the corresponding Luma to show that 50 is a rational threshold to select the “day-image" and “dark-image" subsets. An overview of the proportions of images with different levels of brightness is shown in Fig. 2.

B.4. Examples of trajectories

Since the cameras are static and strawberries do not travel long distances in their life cycle, many strawberries in *GSD* have a complete trajectory of their life cycle. *Figure 1* in the main paper depicts a strawberry located at the outer layer and was mostly observable during its growth. However, not all of the strawberries were completely monitored. According to Fig. 3, there are still relatively short tracks.

There are several reasons for the incomplete observations:

1. The strawberries were growing in dense gathers, as can be observed in the sample views and the video (particularly, the branch in the camera *RGB-2* and *OCN-2* in Fig. 4). Inner-layer strawberries were occluded, but they might also switch positions with others due to different speeds of weight growing.
2. Strawberries from the inner layer started to have more complete observations when the outer-layer strawberries were harvested. For example, radical position changes can be observed in the demo videos.
3. The increases in size and weight might squeeze some strawberries out of frame. For example, the strawberry #397 in the view from *RGB-2* in Fig. 4; the strawberry #596 in the video (RGB part) also moves back and forth at the edge of the frame.
4. Some strawberries grew above the cameras, which were not intended to be monitored. For example, the strawberry # 451 in the view from *RGB-2* in Fig. 4.
5. In cooler weather such as in May or September, the strawberries grow slower and less dense. Both factors make the trajectories longer than in warmer times.

C. Statistics about OCN images

In the main paper, we provide only the statistics of the RGB images of *GSD* because our benchmarking experiments worked on merely the daytime subset of the RGB images. Below, Tab. 1 lists the statistics of the OCN images of *GSD*.

Table 1. Statistical overview of the OCN images of *GSD*. The 2nd column lists the duration of data collection. The 3rd and 4th columns note the amounts of all images and the images used in the benchmarking studies respectively. The last two columns present the total number of bboxes and trajectories.

Camera	Period	Total img	Anno. img	Total bbox	Total track
OCN-1	Apr 23 - Nov 9, 2021	4786	2648	73729	560
OCN-2	Apr 23 - Nov 9, 2021	4785	2688	72158	455
OCN-3	Jun 29 - Nov 9, 2021	3182	1735	68642	488
OCN-1	Feb 22 - Oct 3, 2022	5159	3273	70706	618
OCN-2	Feb 22 - Oct 3, 2022	5162	3337	89201	705
OCN-3	Feb 22 - Oct 3, 2022	5158	3346	108100	742

D. Implementation of algorithms

We introduce further details about the implementations of the algorithms and explain the reason of our parameter settings in this section.

All the object detection and MOT algorithms that we used for the benchmarking experiments are open-source: YOLOX-x and Faster R-CNN (model established with the Detectron2 framework [9]) come with Apache License 2.0. ByteTrack and OC-SORT use the MIT License. DeepSORT and StrongSORT use GNU General Public License v3.0.

D.1. Detection models

We trained two object detection models, YOLOX-x [3] and Faster R-CNN [8], to perform the object detection stage of the MOT algorithms. Since they had similar performances, as presented in Appendix E.1, we selected the MOT results produced only from the YOLOX-x model’s predictions in the main paper.

This section presents the hyper-parameters for building up and training the models. Both models are trained with the complete RGB image dataset, with a “leave-one-camera-out" train-test-split strategy since the RGB cameras monitored different plants. The two models shared the same scales of

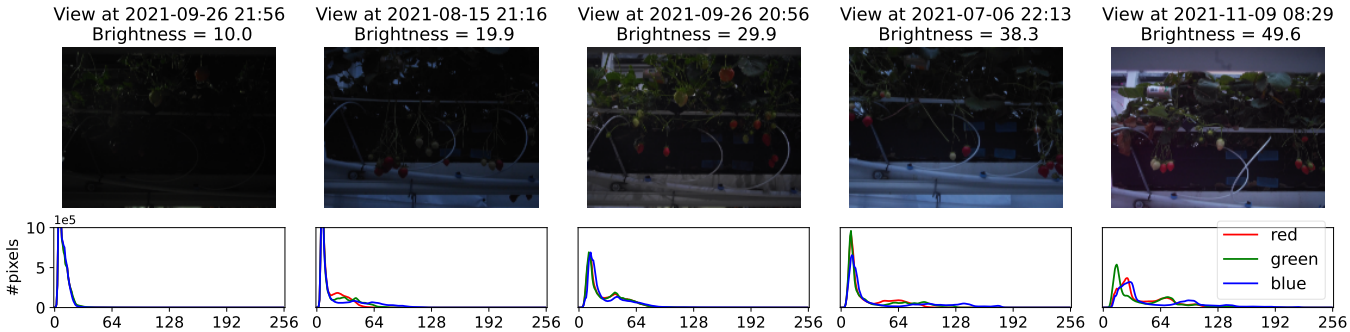

Figure 1. Example dark images at different brightness levels. The collection time and the average Luma are written on top of each image. The RGB spectrum is drawn beneath.

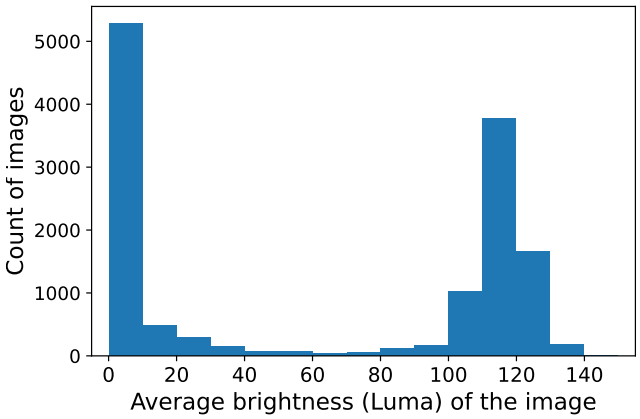

Figure 2. Histogram of the number of *GSD* images under different ranges of brightness. The brightness value is calculated in Luma and averaged over all pixels.

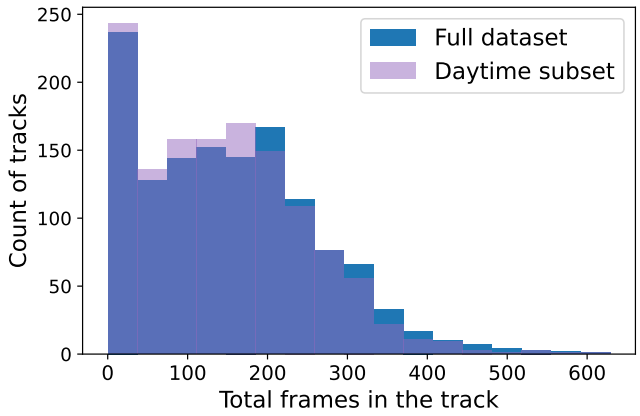

Figure 3. Histogram of lengths of trajectories in *GSD*. The track length is defined as the number of bboxes that are assigned to the track under the ground-truth annotation.

data augmentation, listed in Tab. 2. The detection results are filtered with a confidence threshold of 0.1 before going to the association stages of the MOT algorithms.

Table 2. Data augmentation for training the object detectors. The first column lists how we augment the data, and the second column indicates the value ranges.

Data Augmentation Method	Scale
Random flip horizontal	(probability =) 50%
Random flip vertical	(probability =) 50%
Random rotation	0-90 degrees
Random brightness	$\times$ 0.92-1.12
Random contrast	$\times$ 0.92-1.12

The YOLOX-x model was initialized with a COCO [7] pre-trained model. The images are scaled to 2133$\times$1600 and then padded to 2174$\times$1600 to fit the input aspect ratio. The batch size for training is 4. The model is trained with a cosine annealing learning rate $3.125e^{-6}$ with a warm-up, and a weight decay of $5e^{-4}$. The model is trained by 100 epochs on an Nvidia Tesla V100 GPU. We select the checkpoint with the optimal parameters on the validation set to predict the object detection results for further steps of the experiments.

We trained another Faster R-CNN model using the original ResNet-50 from MSRA [5] and Feature Pyramid Network (FPN) [6] as the model backbone. The model is pre-trained with ImageNet [2].

D.2. MOT algorithms

Before starting the evaluations of the MOT algorithms, we first conducted grid searches to figure out the optimal parameters for the strawberry growth-tracking scenario. The grid search was conducted on the YOLOX-x detections of the *RGB-1* set.

Tab. 3 and 4 present part of our grid-search results. As is shown, the Intersection over Union (IoU) threshold ("iou-thre") was the dominant variable of performance of OC-SORT, both in terms of MOTA and IDF1. One reason could be the irregular movements of objects, illustrated by the Figure 4 in the main text. The confidence threshold had limited effects when using low IoU Threshold.

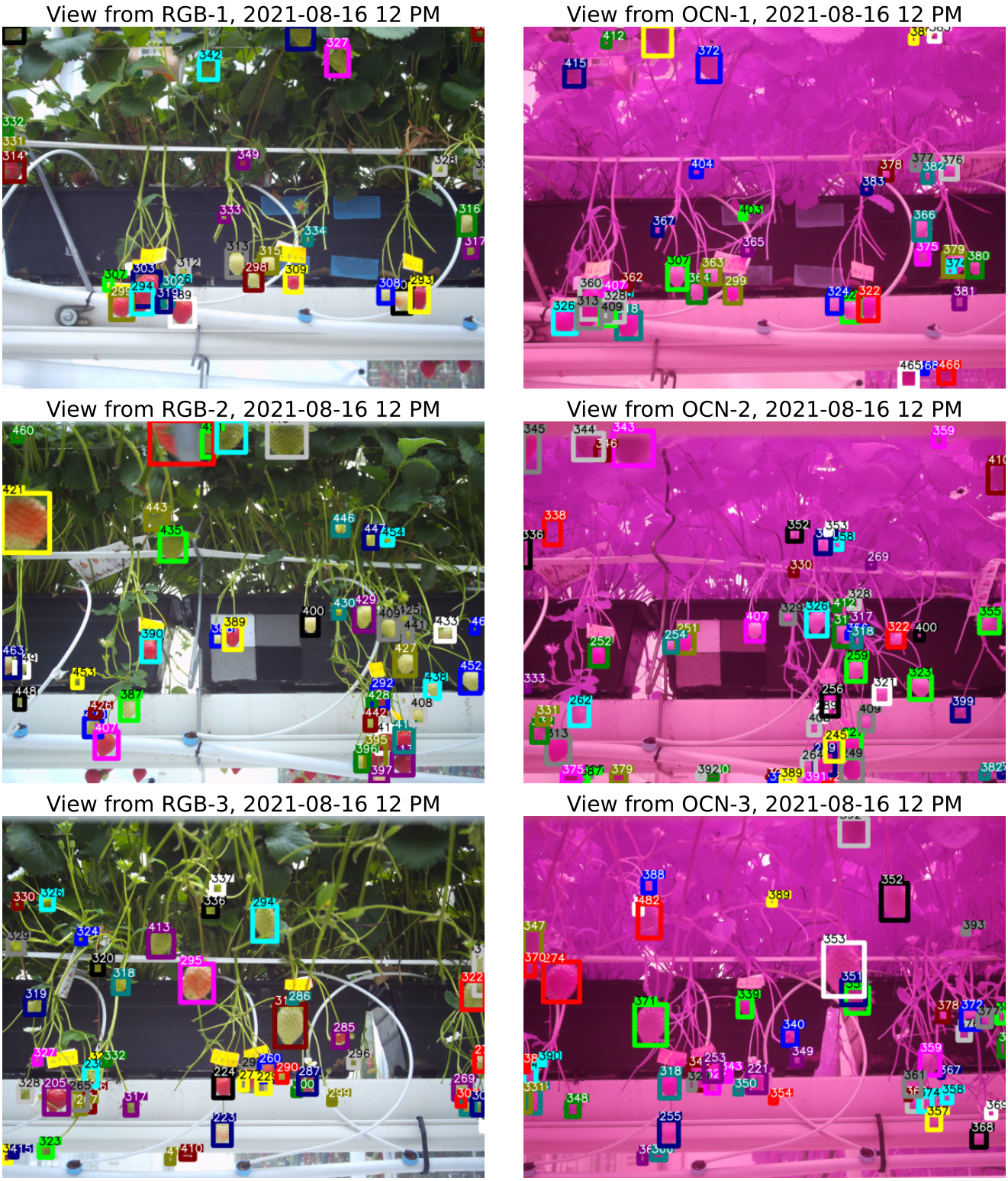

Figure 4. Example views of the three pairs of cameras. The numbers in the camera name indicate how the cameras are paired. The colored rectangles are the ground-truth bbox annotations. The trajectory IDs are noted at the top of the bboxes and color-coded. The trajectory IDs from the OCN and RGB cameras are not consistent, but the mapping of the IDs is manually noted in a separate data sheet.

Table 3. Grid search of iou-thre and conf-thre in OC-SORT. Performances are indicated by MOTA. All experiments have a default setting of min-hits=3 and max-age=30. The selections of conf-thre are the indices of rows, and the selections of iou-thre are indicated by the columns.

Confidence Threshold	IoU Threshold **0.1**	0.3	0.5	0.7	0.9
0.1	64.5	61.4	56.1	45.6	15.7
0.3	64.6	61.4	56.2	45.6	15.7
0.5	64.5	61.4	56.1	45.6	15.7

Table 4. Grid search of iou-thre and conf-thre in OC-SORT with performances indicated by the IDF1 score under different settings. All the experiment settings are the same as in Table 2. The selections of conf-thre and iou-thre are the indices of rows and columns respectively.

Confidence Threshold	IoU Threshold **0.1**	0.3	0.5	0.7	0.9
0.1	67.3	64.9	60.6	51.2	20.2
0.3	67.4	64.9	60.6	51.2	20.2
0.5	67.3	64.8	60.5	51.2	20.2

DeepSORT uses the maximum cosine distance of features ("max-cos-dist") as a gating threshold. Considering the changing appearance of the *GSD* objects, we checked the cosine distance of the features of the same object over the frame. As depicted by Fig. 5, the features of adjacent observations of the object have an average cosine distance of 0.44. Hence, we regard distances larger than the value are large enough for distinctive objects. Therefore, we select 0.45 as the max-cosine-distance value when implementing DeepSORT and StrongSORT.

The final decision on the parameters is made by referring to the grid search results and the default settings of the MOT algorithms. Details are listed in Tab. 5.

Table 5. Detailed parameter settings of the benchmark experiments of the MOT algorithms.

	conf-thre	iou-thre	min-hit	max-age	max-cos-cost
OC-SORT	0.1	0.1	3	30	-
ByteTrack	0.1	-	3	30	-
Deep-SORT	0.1	0.1	1	30	0.45
Strong-SORT	0.1	0.1	1	30	0.45

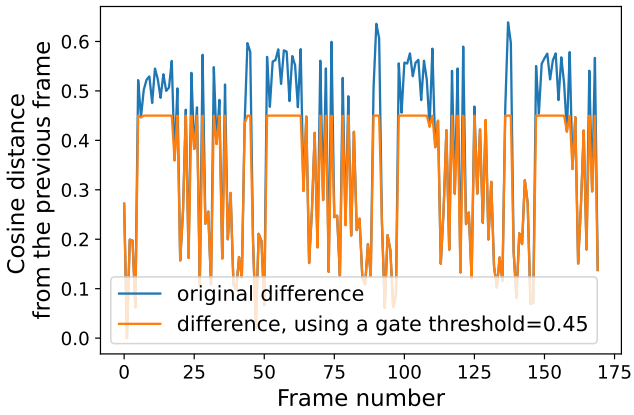

Figure 5. The cosine distances of the normalized features of adjacent observations during the complete track of an example strawberry. The x-axis shows the temporal sequence of the observations, and the y-axis indicates the cosine distance from the previous observation. The features are encoded by the same extractor as we used for DeepSORT and StrongSORT. The blue line presents the original cosines distances, and the orange line depicts the effect of add a gate threshold = 0.45.

E. Detailed model performance

With the "leave-one-camera-out" policy, we validate the two object detection models in three datasets, i.e. *RGB-1/2/3*. As such, the same policy is also applied to the implementation of the MOT algorithms. The model performances in the main paper are presented by grouping all the validation results together. This section provides the specific performances on each validation set.

E.1. Performance of object detection

In the experiment of object detection, we averaged the performances of implementing each object detector over *RGB-1/2/3*. Tab. 6 lists the specific model performance on each validation set. The YOLOX-x model performs similarly, with AP in the range 51 to 54 and AP_{50} in 86 to 89. The Faster R-CNN model has an average precision between 53 to 59 (AP_{50} between 85 to 92). The performances on *RGB-2* are relatively worse than the others, but the differences are limited. We also list the benchmark Average Precision (AP) of the model on the COCO dataset, as claimed by the model developers. The results validate that the models perform similarly on the three validation sets, so it is reasonable to use the averaged AP for discussion in the main paper. The comparison in the main paper demonstrates that both models perform at a comparable level with the corresponding benchmarks that are stated by the model developers [3, 9]. Hence, in the main paper, we argue that the difficulty level of object detection on *GSD* is not significantly higher than other MOT datasets.

Table 6. Performances of object detectors on *RGB-1/2/3* of *GSD* respectively. We use AP, AP_{50} and AP_{75} as the metrics. The first column gives the model of the object detector, and the second column indicates the validation set. The last row of each detector section, written as "average", is the averaged performance of the models on the three validation sets, calculated without a weight. Performance of object detectors on *GSD*, evaluated by AP, AP_{50} and AP_{75}. All the values are averaged over the metrics of the three models that tested on the camera *RGB-1/2/3* respectively.

Detector	Validation Set	AP	AP_{50}	AP_{75}
YOLOX-x				
	RGB-1	53.7	88.3	57.7
	RGB-2	51.0	86.0	54.7
	RGB-3	62.4	87.5	71.7
	Average	55.7	87.3	61.4
Faster R-CNN				
	RGB-1	58.2	91.5	65.9
	RGB-2	53.3	87.9	58.0
	RGB-3	56.0	85.8	65.7
	Average	55.8	88.4	63.2
YOLOX-x	COCO [3]	59.2	86.3	61.9
Faster R-CNN	COCO [9]	40.2	60.9	43.8

E.2. Performance of online MOT algorithms

In the main paper, we present the benchmark of the four MOT algorithms on *GSD* by the overall metrics from the daytime subset. This section shows metrics in specific. Below, Tab. 7 and Tab. 8 present the detailed performance of ByteTrack and StrongSORT respectively. The performance is assessed on each camera subset on the full annotated dataset and on the daytime subset. In general, there are no significant performance gaps in the algorithms when using the data from different cameras. In terms of *RGB-1*, which we used as a test set to decide whether to use a daytime subset or not, we could notice a slight improvement in the MOT metrics, yet the performance on the entire dataset shows limited differences when using different subsets for evaluation.

E.3. Performance of end-to-end MOT algorithms

Since GSD consists of long series of high-resolution images, we excluded offline MOT solvers in the scope of benchmarking experiments. However, we argue that end-to-end is feasible for the task, but it does not give a better performance than the other real-time MOT algorithms that we've applied in the paper. We implemented a demo of *GMTracker* [4] on the YOLO-X detection of the first few frames of *GSD-2021-CAM-1*. However, the performance metrics are not very positive: without using the quadratic matching function to represent the 2nd order relationship, the HOTA score is 30 (on the first 1000 frames); and when the quadratic matching function is activated and its GNN uses the parameters trained on MOT17, the HOTA score improved to 38 (on the first 750 frames, as shown in *Table 2* of the main paper).

We noticed that with the involvement of more frames, the performance slightly raised, nevertheless, the processing time becomes exponentially longer with the greater amount of detections-to-be-matched between frames, as shown in Fig. 6 – hence we only applied the demo in the first 750 frames of GMT, which has already taken nearly a week to run with a trained object-matching model.

Noted that our implementation used a model with pre-trained parameters, due to the fact that the training process requires tremendous computation effort. Particularly, a quadratic affinity matrix as described in [4] requires much larger memories (e.g. matching 50 objects with another 50 requires 40+GB of memory when training on the *MOT17* dataset) than the Hungarian-algorithm-based methods, yet could not result in a significant performance increase in our demonstration.

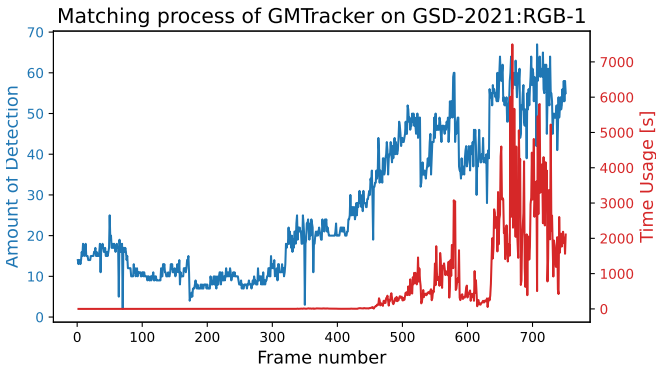

Figure 6. Reaction time of GMTracker when processing each frame of the daytime subset of *GSD-2021-RGB-1*. The x-axis indicates the frame number, which was re-indexed according to the daytime subset. The blue line with the y-axis on the left indicates the amount of detected bbox in each frame. The red line with the y-axis on the right illustrate the time that the model needed to match the detected bbox within that frame with those in the previous frame.

Hence, we did not apply the end-to-end MOT on the entire daytime subset as the other four two-stage MOT algorithms in the main paper. Nevertheless, we still noted the metrics down in *Table 2* of the main paper, so as to compare the performance with other algorithms and with the metrics that *GMTracker* achieved on other popular MOT datasets.

F. Metrics correlation with data characteristics

In the main paper, we highlight two primary challenges presented by this dataset: irregular movements and significant appearance changes exhibited by a majority of the objects. To further investigate the impact of dataset characteristics, we conducted MOT performance evaluation over

Table 7. Detailed performance of ByteTrack

Camera	HOTA	MOTA	IDF1	AssA	AssRe	AssPr	IDS/Tr	FM/Tr
PERFORMANCE ON THE FULL DATASET:								
RGB-1	39.77	64.68	40.58	27.30	32.44	66.95	4.4	7.1
RGB-2	39.25	64.65	40.07	27.22	30.82	70.12	5.0	6.8
RGB-3	40.17	80.78	38.09	23.01	25.72	72.64	6.3	7.7
All	39.74	70.29	39.59	25.72	29.49	70.06	5.2	7.2
PERFORMANCE ON THE DAYTIME SUBSET:								
RGB-1	40.03	65.53	40.74	27.38	32.43	66.79	4.4	4.6
RGB-2	39.25	64.67	39.95	27.15	30.64	70.37	5.1	5.8
RGB-3	39.93	81.17	37.40	22.63	25.27	72.49	6.3	5.8
All	39.75	70.73	39.38	25.58	29.26	70.01	5.2	5.4

Table 8. Detailed performance of StrongSORT

Camera	HOTA	MOTA	IDF1	AssA	AssRe	AssPr	IDS/Tr	FM/Tr
PERFORMANCE ON THE FULL DATASET:								
RGB-1	34.11	33.96	32.93	23.98	28.76	60.36	8.2	6.5
RGB-2	35.31	40.84	33.95	25.34	28.61	65.50	9.1	7.0
RGB-3	37.29	66.36	33.41	21.14	24.01	66.73	11.2	7.3
All	35.51	47.45	33.41	23.38	26.98	64.39	9.5	6.9
PERFORMANCE ON THE DAYTIME SUBSET:								
RGB-1	35.17	36.86	33.83	24.93	30.16	60.94	7.3	4.3
RGB-2	35.76	41.96	34.64	25.89	29.46	65.01	8.5	5.8
RGB-3	37.58	67.74	33.48	21.18	23.95	67.53	10.7	5.4
All	36.14	49.32	33.98	23.87	27.66	64.74	8.8	5.1

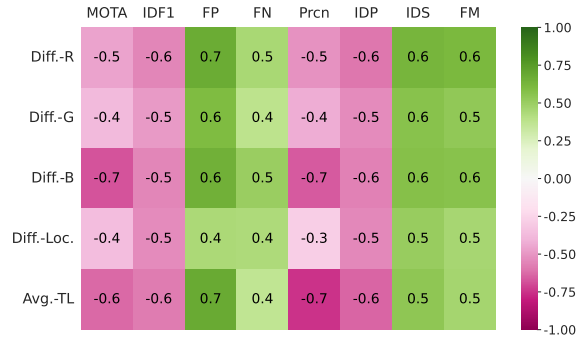

Figure 7. Correlation analysis among the performance metrics of DeepSORT and the characteristics of the trajectories. The values in the grid indicate the exact value of correlation between the MOT metric from the horizontal axis, which is mentioned on the top of the figure, and the characteristics indicator from the vertical axis and listed on the left. The color of each grid is defined by the correlations, according to the scale shown on the right.

fixed-duration periods. These periods were determined using a rolling window of 14 days, with a stride of 7 days. We measured changes in object appearances by calculating the χ^2 distance of each color spectrum at 2 p.m. daily, so as to limit the daily illumination variance. Differences in object locations were quantified by averaging the location changes of the same objects, based on bounding box (bbox) coordinates. Additionally, we computed the average lengths of trajectories (TL) within each period, which is defined as the number of bbox annotations. To assess the correlation between these indicators and the MOT metrics derived from DeepSORT, which incorporates both location and appearance features during data association, we analyzed the results.

The correlation values, depicted in Fig. 7, demonstrate the extent of influence on the metrics. It is evident that color changes exert a significant impact on all performance metrics. Notably, the False Positive (FP) rate of detected-and-associated bounding boxes is particularly affected, indicating an increased number of missed object matches when relying on appearance-based association across frames. The influence of object movement variations, while comparatively less pronounced than color changes, is more strongly correlated with tracklet identification performance. This observation aligns with the challenges posed by sudden position changes resulting from horticulture activity interruptions and the sparsity of data collection. The correlations between FP rate and redness, as well as between FP rate and tra-

jectory length (TL), reach values of 0.7, underscoring their significant contributions to overall performance. Moreover, longer trajectories exhibit a strong correlation with reduced Multiple Object Tracking Accuracy (MOTA) and precision, indicating a substantial negative impact resulting from the extended duration of the tracking task. Taken together, these correlations provide evidence of the challenges introduced by the *GSD*: i) the appearance change of objects over the long period and ii) the irregular movements recorded in the sparse frames.

G. Comparisons of trajectories

Fig. 8 gives a more abstracted comparison of trajectories in *GSD-2021-RGB-1* and *MOT20-01*, using the ground-truth annotations of the first and last observation of each track. The visualization demonstrates that a majority of objects in both sequences experienced changes in their locations, with object movements in *MOT20-01* generally being more substantial. This observation, when compared with Figure 4 in the main text, further supports the distinctive pattern of movement exhibited by objects in the *GSD*: predominantly static yet with sudden and irregular changes.

The Euclidean distances and their ratios to the area of the initial observation (1st bbox) in both sequences exhibit similar distributions, indicating that the scales of movements in *GSD-2021-RGB-1* and *MOT20-01* are comparable.

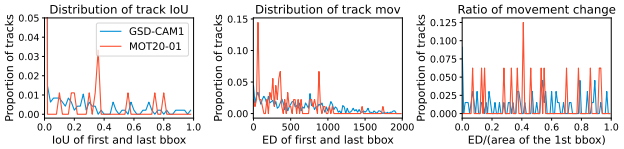

Figure 8. Quantitative comparisons of tracks in *GSD-2021-RGB-1* and *MOT20-01*. The 1st plot shows the IoU distribution of the first and last bbox of each track. The 2nd plot illustrates the IoU Euclidean distance (ED) of the first and last bbox of each track. The 3rd plot presents the proportion of the Euclidean distance (ED) of the first and last bbox of each track to the size of the first bbox.

H. Dataset collection

This section introduces the detailed data collection hardware setup of *GSD*.

H.1. Data collection setup

The strawberries in the greenhouse were cultivated in planting baskets, which were hung in parallel lines. Figure 9 gives a side view of the rows. Strawberries grew out from both sides of the baskets.

The cameras were grouped as three pairs of RGB and OCN cameras. They were installed on the opposite row from where the strawberries were growing, as shown in Figure 2 in the main text. As Figure 10 shows, they were fixed on the heating pipe with camera clamps. They were connected to the local electrical grid with a powered USB hub, so they could stay awake all the time.

Figure 9. A side view of the planting baskets. The cameras were attached to the heating pipe at the neighbor's row. For example, if the strawberries grew in the left row in the image, cameras would be installed at the highlighted heating pipe. This particular image is not taken by the data collection devices, so the distortion in the image is not related to the strawberry observations.

Figure 10. Camera installation for data collection. The cameras were fixed to the heating pipe and connected to the electrical grid with the yellow USB hub.

Acknowledgement

We would like to acknowledge the generous support provided by Topsector Tuinbouw & Uitgangsmaterialen (The Netherlands), who fully funded this project. We would also like to acknowledge the Computer Vision Lab of TU Delft for the funding support towards one of the co-authors. We express our gratitude to Delphy Improvement Centre B.V. (The Netherlands) for their assistance in providing the necessary locations and strawberry plants for monitoring, as well as their valuable advice on data collection. We are also very thankful to Xucong Zhang, Yuki Watabe, and Ilja Tigges for their contributions and insightful suggestions, greatly elevating the quality of this paper.

References

[1] Sergey Bezryadin, Pavel Bourov, and Dmitry Ilinih. Brightness Calculation in Digital Image Processing. *International Symposium on Technologies for Digital Photo Fulfillment*, 1(1):10–15, 1 2007. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009. 3

[3] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 2, 5, 6

[4] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5295–5305, 2021. 6

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[7] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. 5 2014. 3

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 6 2017. 2

[9] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. 2, 5, 6