bio-protocol

Open Access

Supplementary information for Low-Latency Multicamera 3D Tracking of Insects with Braid

Michael J.M. Harrap[1] and Andrew D. Straw[1, 2, *]

[1]Institute of Biology I, Faculty of Biology, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany
[2]Bernstein Center Freiburg, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany
*For correspondence: straw@bio.uni-freiburg.de

Supplementary materials 1: PC specifications

Below are the components of the Linux PC used in the protocol that are critical to Braid performance. Other hardware (tower case, USB keyboard, mouse, HDMI screen) is also required, but their specifications will not impact Braid functionality.
Motherboard: ASRock, Taipei, Taiwan **model:** X570S PG Riptide
CPU: AMD, Santa Clara CA, USA **model:** Ryzen 7 5800X 8-core
Graphics card: NVIDIA, Santa Clara CA, USA **model:** GP106GL [Quadro P2000], with Graphics driver: nvidia v: 550.163.01
RAM/Memory: Kingston Technology, Fountain Valley CA, USA **model:** 9905734-415.A00G type: DDR4 size: 16 GiB speed: 3,200 MT/s
Note: Two of the listed RAM modules were used in the PC.
Appropriate CPU and PC tower fans and heat sinks are also required as per your lab conditions. Additionally, sufficient hard drive data storage is required if video is also to be saved; the PC used in the protocol had a 2 TB hard drive. This is far more than that required for the data collected in the protocol (see below).

Approximate minimum specifications

Braid has no fixed minimum hardware requirements: the disk space, memory, processor, and network capacity needed scale with the number of cameras, their resolution, their frame rate, the number of objects tracked, and the number of cameras recording video. They can be estimated with the equations below and Tables S1 and S2. To ensure the desired performance, users are advised to allow up to 50% headroom on these specifications. Table S3 summarizes the estimated minimum specifications for this protocol, including this headroom.

Table S1. Setup parameters and where they are set.

Symbol	Description	This protocol	Set in
N_{cam}	Number of cameras	5	Braid configuration file
f	Camera frame rate	40 fps	Braid configuration file (trigger period)
F	Size of one raw frame: width × height × bytes per pixel	≈ 2.3 MB (1920 × 1200 × 1)	Camera settings file (.pfs)
d	2D detections per camera per frame	1	Experiment (≤ max_num_points)
N_{obj}	Objects tracked simultaneously	1	Experiment
k	Mean number of cameras contributing to each 3D estimate	≈ 3.8 (measured)	Experiment (camera placement)

T	Recording duration (s)	–	Experiment
N_{files}	Number of .braidz recordings	–	Experiment
N_{rec}	Number of cameras recording MP4 video	5	Braid GUI
b_i	MP4 bitrate of recording camera i (bit/s)	≤ 1 Mbit/s (0.25–1.0 measured)	Strand Camera GUI (MP4 Bitrate; default 1 Mbit/s for the NVENC only)
B	Post-trigger buffer size (frames)	0	Strand Camera GUI (default 0)
N_{live}	Number of live camera feeds open	0	Braid or Strand Camera GUI

Table S2. Values measured on the PC used in this protocol (AMD Ryzen 7 5800X, 16 threads; one thread = one fully used logical core).

Symbol	Description	Measured value
s_{2D}	2D detection row in a .braidz file (≈ 16B without and ≈ 54B with a detection)	≈42 B
s_{3D}	3D estimate row in a .braidz file	≈166 B
s_{DA}	Data association row in a .braidz file	≈2.2 B
s_{img}	Fixed content of each .braidz file, per camera (one image plus settings)	≈3.4 MB
M_{OS}	Memory: operating system, desktop and web browser	≈2.9 GB
M_{Braid}	Memory: Braid	≈0.2 GB
M_{cam}	Memory: each Strand Camera instance (not recording)	≈0.45 GB
M_{enc}	Memory: additional, per camera recording MP4	≈0.6–0.85 GB
c_{cam}	CPU: each Strand Camera instance (acquisition and 2D detection)	≈1.1 threads
c_{Braid}	CPU: Braid 3D tracking	≈0.3 threads
$c_{enc,i}$	CPU: MP4 encoding with NVENC, per recording camera	≈0.15 (1 fps) to 0.9 (40 fps) threads
c_{live}	CPU: each open live feed (Strand Camera and web browser)	≈0.7 threads

Disk space (bytes):

$$D_{track} = N_{files} \cdot N_{cam} \cdot s_{img} + T \cdot f \cdot (N_{cam} \cdot \max(1, d) \cdot s_{2D} + N_{obj} \cdot s_{3D} + N_{obj} \cdot k \cdot s_{DA})$$
$$D_{total} = 2 \cdot D_{track} + T \cdot \Sigma_i \, b_i / 8$$

At least one 2D row is saved per camera per frame, and tracking data are briefly stored twice while each .braidz file is created. The MP4 bitrate b_i is a Strand Camera setting, not a camera property; each 1 Mbit/s fills ≈0.45 GB per hour per camera. Without a supported NVIDIA GPU, video is encoded with libx264, and the bitrate setting does not apply (1.1–2.3 Mbit/s measured). Strand Camera can also record uncompressed FMF video, which is not recorded in the current protocol; such files are very large ≈330 GB per hour per camera.

Memory (bytes):

$$M_{max} = M_{OS} + M_{Braid} + N_{cam} \cdot M_{cam} + N_{rec} \cdot M_{enc} + (N_{cam} + N_{rec}) \cdot B \cdot F$$

The last term applies only when post-triggering is used ($B > 0$); it is calculated from the software design and was not measured.

Processor (CPU threads):

$$C = N_{cam} \cdot c_{cam} + c_{Braid} + \Sigma_i \, c_{enc,i} + N_{live} \cdot c_{live}$$

plus ≈0.5 threads for the operating system. Video recording and open live feeds also increase the mean latency of 3D reconstruction (from ≈49 to ≈57–63 ms). Having too many live feeds open can lead to the Braid GUI becoming too slow to use, so closing unneeded live feeds while recording is advised. Software encoding (libx264) needs considerably more CPU than NVENC.

Network (bits per second):

$R_{cam} = f \cdot F \cdot 8, \quad R_{total} = N_{cam} \cdot R_{cam}$

R_{cam} must stay below each camera's gigabit link (≈0.9 Gbit/s in practice) and R_{total} below the link between the switch and the PC. Here, R_{cam} ≈0.74 Gbit/s and R_{total} ≈3.7 Gbit/s (measured, with no packet loss), hence the 10 Gbit/s PC connection; at this resolution, gigabit cameras are limited to ≈48 fps. Network use does not change when recording video.

Table S3. Approximate minimum specifications for this protocol (using five cameras at 1920 × 1200 pixels and 40 fps, one object), including 50% headroom; measured values in brackets. Network values have no headroom added, as link capacities are fixed.

Requirement	Tracking only	Tracking with MP4 video as in protocol
Processor (CPU)	≈9 threads (measured ≈6)	≈13 threads (measured ≈9)
Memory (RAM)	≈8 GB (measured ≈5.3 GB)	≈16 GB (measured ≈8.4 GB, at most ≈10 GB)
Disk space	≈0.1GB per hour (measured ≈0.06 GB)	≈2 GB per hour (measured ≈1.4 GB)
Network (no headroom)	1 Gbit/s link per camera; ≥ 4 Gbit/s (e.g., 10 Gbit/s) into the PC	Same as tracking only
Graphics card (GPU)	Not required	NVIDIA GPU with NVENC recommended (≈19% of a Quadro P2000 used)

Note: AprilTag detection and checkerboard calibration are processor-intensive and may need slower camera frame rates (see main text, Troubleshooting, Problem 2) but add little to disk or memory requirements.

Supplementary materials 2: Additional equipment diagrams

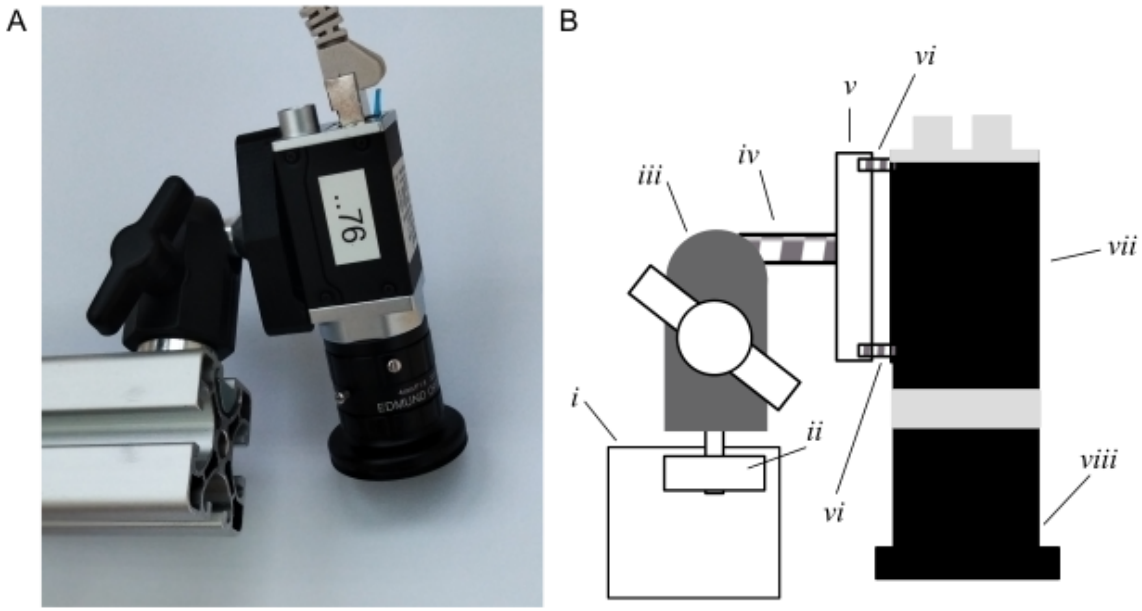


Figure S1. Camera and mount assembly. (A) Photographic image of the assembled components mounted on the support and camera mounting frame. (B) Diagrammatic representation of image A; note that this diagram is not to scale. Italic numerals indicate components and features. i) The aluminum profile of the support and camera mounting frame. ii) The aluminum T-nut fitting, which fixes the ball mount to the support and camera mounting frame. iii) The ball mount. iv) The screw thread of the ball mount, which screws into the 1/4" nut in the socket of the mounting plate. v) The mounting plate. vi) The 3 mm screw fittings of the Basler camera that are threaded through the holes in the mounting plate (two are shown; in reality, there are three). vii) The Basler camera. viii) The camera lens.

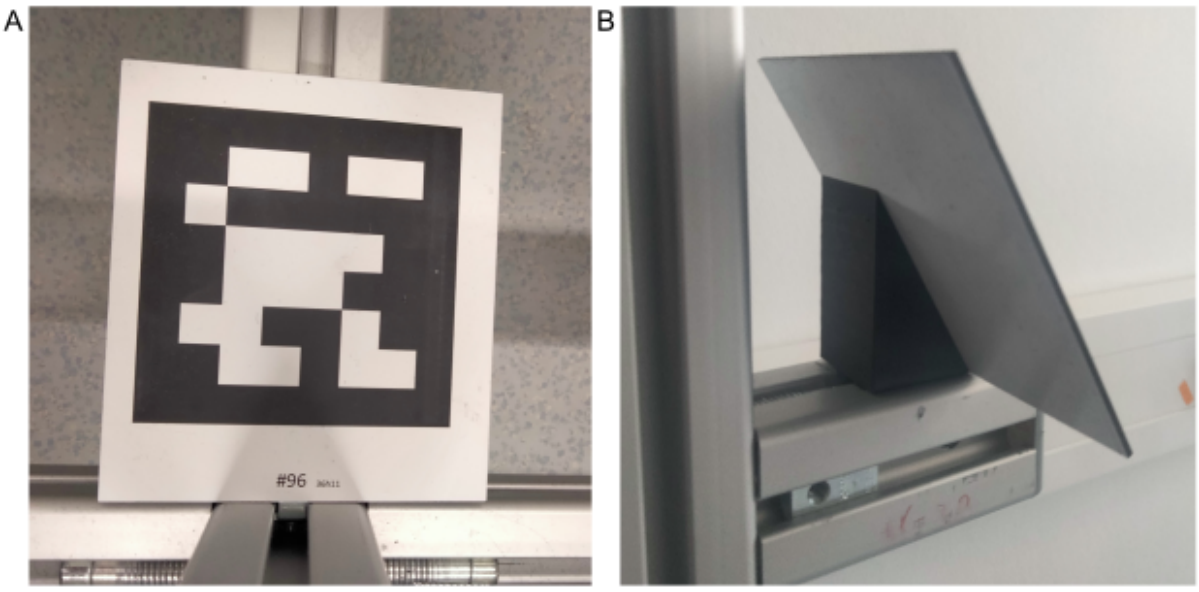


Figure S2. Examples of the AprilTag plates, prepared as described in the protocol using the CAD files provided in [63] (*protocol_downloads/custom_equipment_cad_files/AprilTag_plates/*). The plates used are either attached flat to the frame directly using the TroLase plate's adhesive backing (A), or at a sloped angle with the sloped mounting blocks (B).

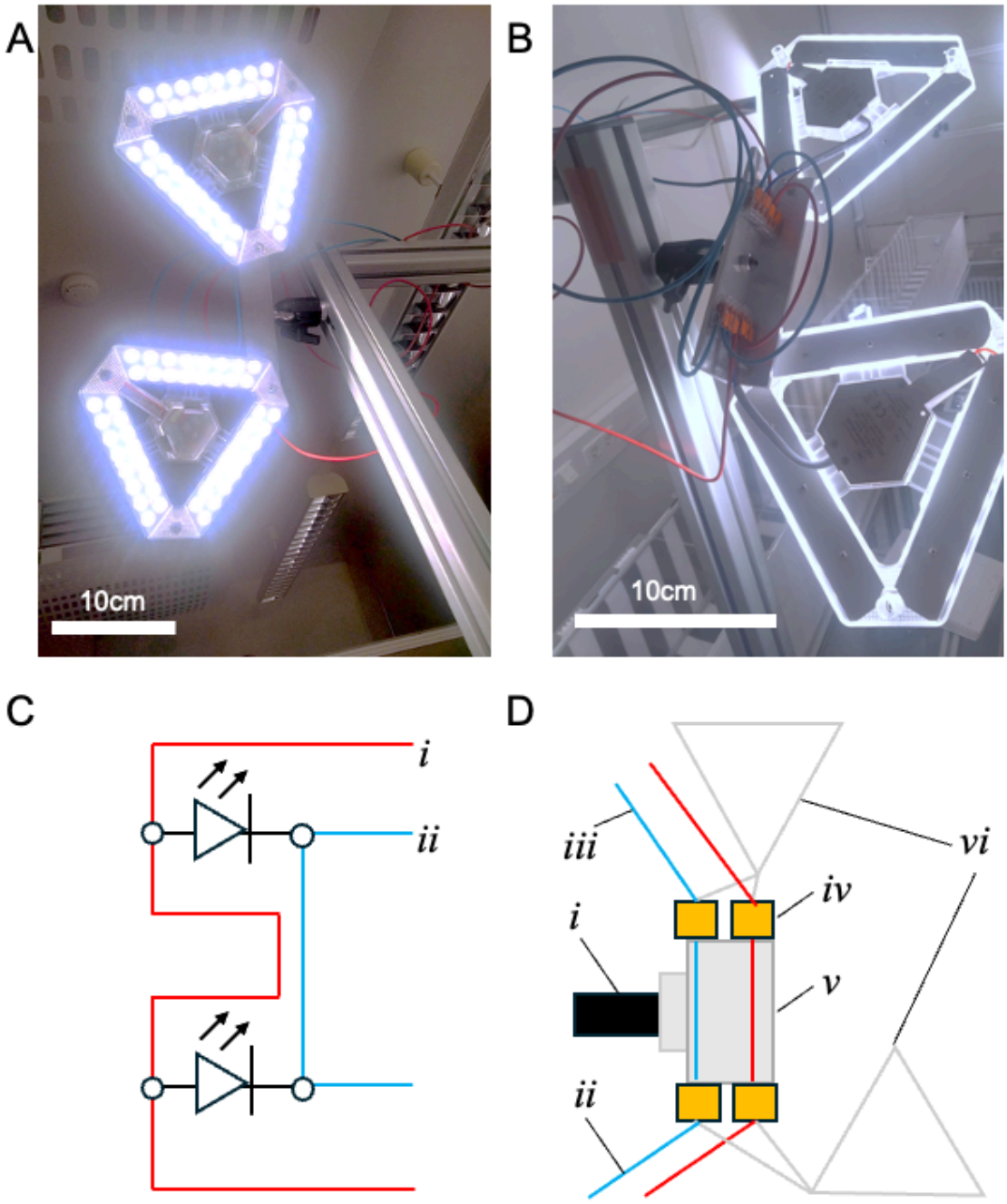


Figure S3. Light arrays used in the protocol. (A) The array from the front. (B) The array from behind. (C) A circuit diagram of each array. In C, each LED unit is represented by a single LED. Red wire in circuit diagrams indicate outgoing positive wire, blue wire indicates incoming negative wire. Circles in circuit diagrams indicate the "Wago" Connectors. Four such units are used. See Figure S4 for details on wiring of the units together and arrangement. Note that wires terminating in C and wires indicated by i and ii are not required for the last unit; see Figure S4. (D) Diagrammatic representation of B. Italic numerals indicate components: i) the ball mount. ii) incoming positive wire, red, outgoing negative wire, blue. iii) Outgoing positive wire, red, incoming negative wire, blue. iv) "Wago" connectors. v) L-shaped metal plate that screws into the ball mount and to which connectors are attached. Note that positive and negative wires span the plate between connectors. vi) Two LED light units. Note that photographic images have been sharpened and color-adjusted to clarify the view of the key features.

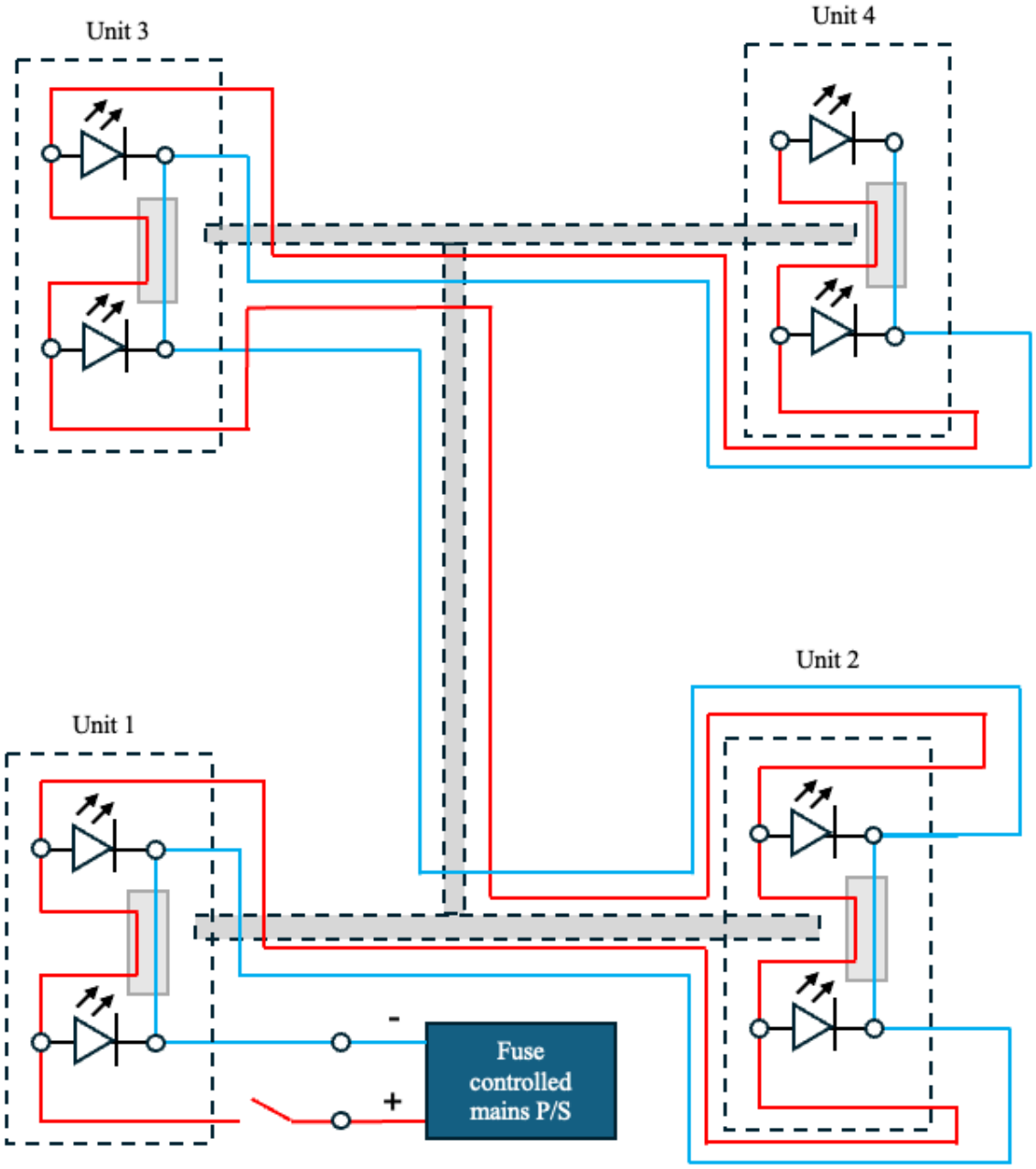


Figure S4. Simple circuit diagram and sketch of light array positions across the support and camera mounting frame. Each LED unit is represented by a single LED. Red wire in circuit diagrams indicate outgoing positive wire, blue wire indicates incoming negative wire. Circles in circuit diagrams indicate the "Wago" Connectors. Note that there are four light array units (shown in Figure S3), each with two LED units, and that Unit 4, the unit at the end of the circuit, lacks incoming negative and outgoing positive wires. Solid gray rectangles represent the mounting plate of each unit. Dashed rectangles represent the support and camera mounting frame viewed from above. Note that the diagram shows the wiring of components and is not to scale.

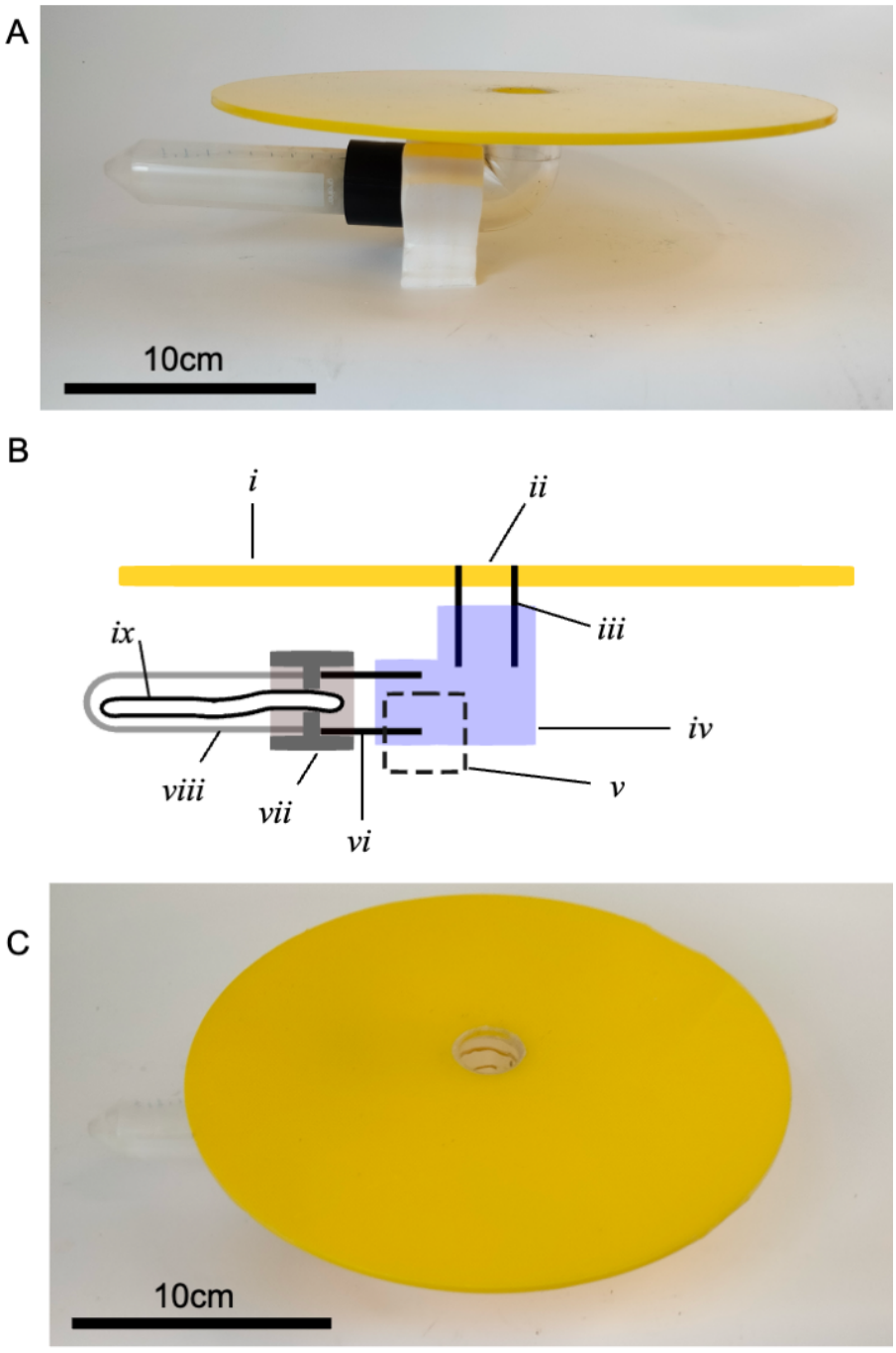


Figure S5. Type A artificial flower. (A) Photographic image of the artificial flower in profile. (B) Diagrammatic representation of image A; note that this diagram is not to scale. Italic numerals indicate components and features. i) The yellow acrylic disc. ii) The entrance hole to the flower. iii) A 5-cm section of a 32-mm diameter pipe. iv) A 90° pipe fitting for a 32-mm pipe. v) Polystyrene stand. vi–ix) The Falcon tube collar and wick design feeding reservoir, consisting of a 5-cm section of 32-mm diameter pipe (vi) that is inserted into a 3D-printed flower collar (vii) that links the pipe to a Falcon tube filled with sucrose solution (viii), through which a wick (ix) is inserted. (C) Artificial flower as it appears from above. Note that scale bars in photographs are approximate due to the image's perspective view.

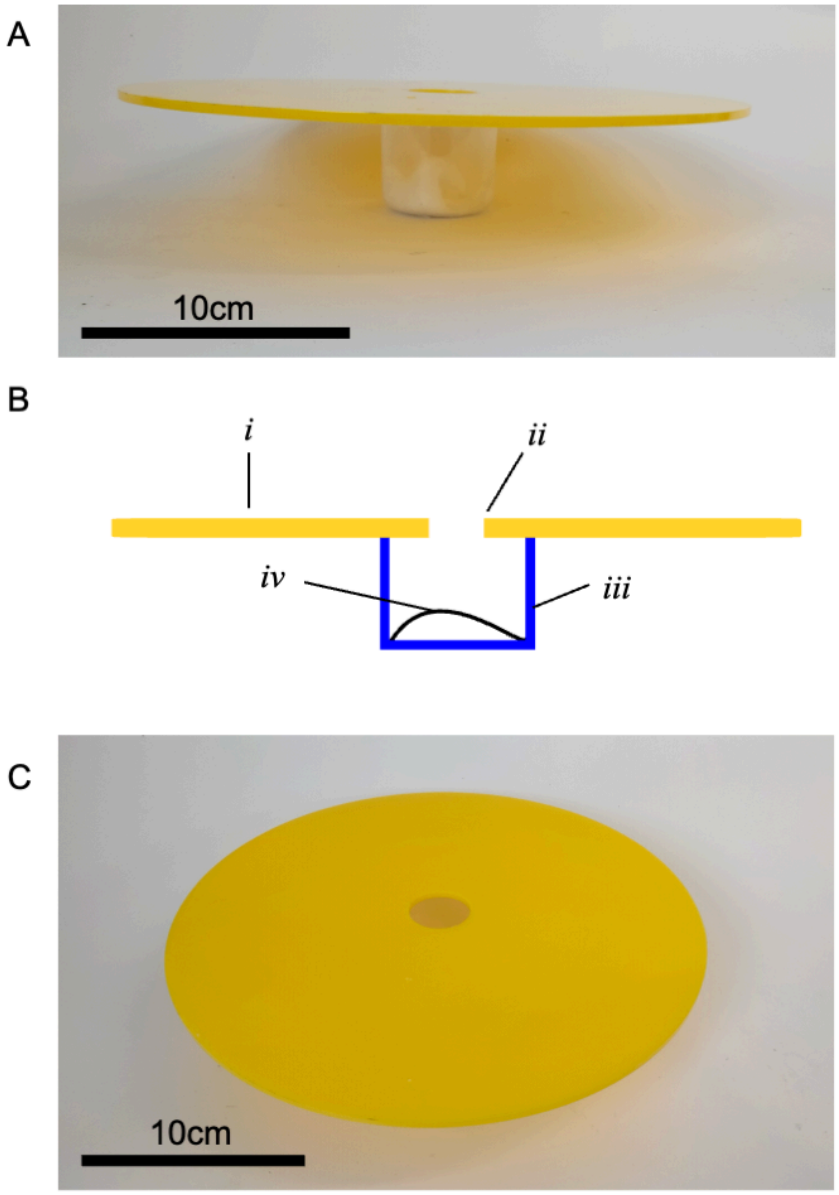


Figure S6. Type B artificial flower. (A) Photographic image of the artificial flower in profile. (B) Diagrammatic representation of image A; note that this diagram is not to scale. Italic numerals indicate components and features: i) the yellow acrylic disc. ii) The entrance hole to the flower. iii–iv) Plastic jar design feeding reservoir, consisting of a plastic jar (iii) and a cotton wool wick soaked in sucrose (iv). (C) Artificial flower as it appears from above. Note that scale bars in photographs are approximate due to the image's perspective view.

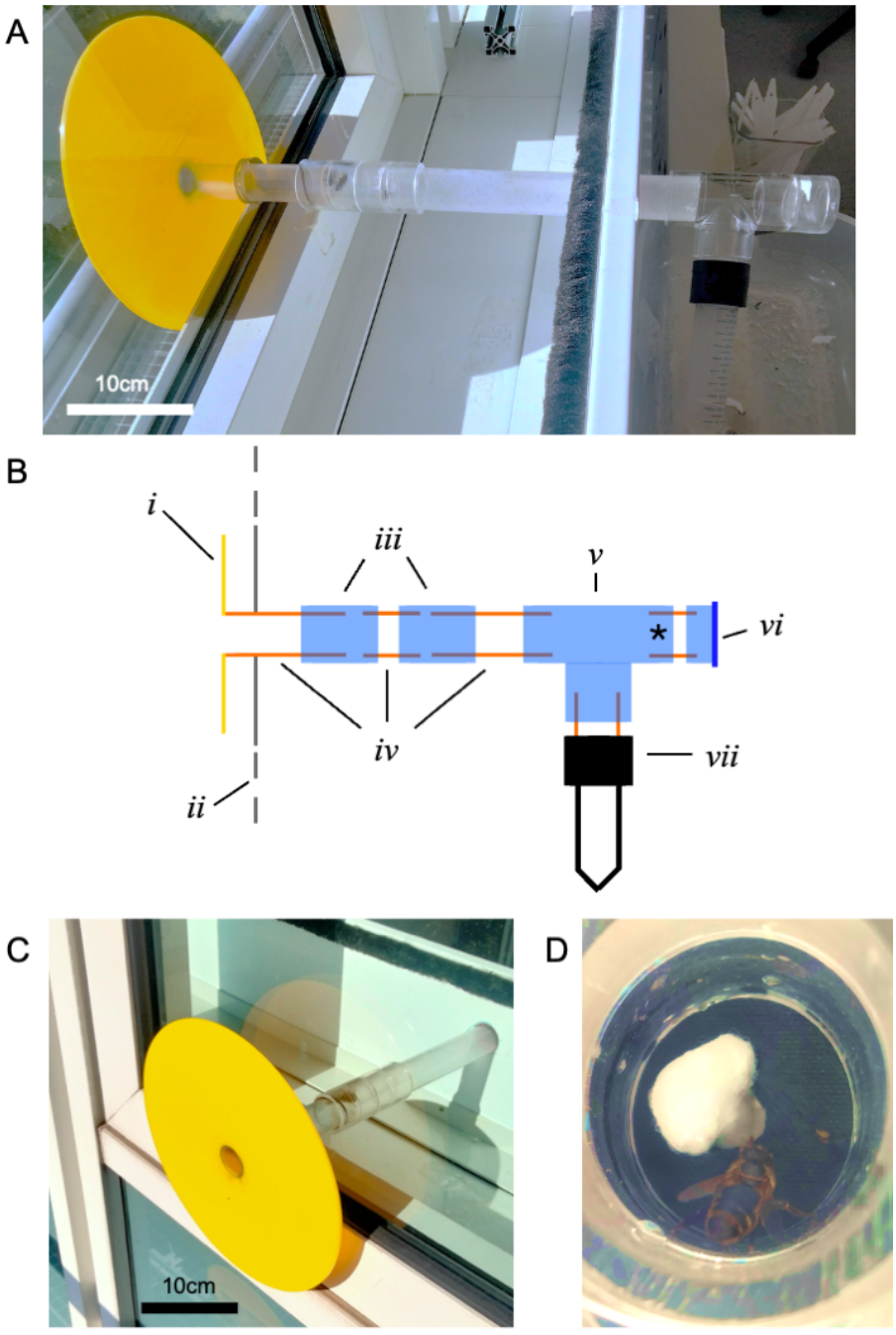


Figure S7. Window access when access to the flight arena is blocked. (A) Photographic image. (B) Diagrammatic representation of image A; note that this diagram is not to scale. Italic numerals indicate components: i) an artificial flower disc. ii) The window pane. iii) Horizontal pipe connector fitting for the 32-mm tubing. iv) 32-mm diameter piping (lengths left to right: 10 cm, 5 cm, 30 cm). v) T-junction pipe connector for 32-mm tube. vi) 5-cm length of 32-mm pipe that leads to a closed cap. vii) Falcon tube feeding reservoir. Note that between items iii and v, there is a hole drilled into the windowsill through which piping passes for support. This is not shown in the diagram. The point indicated by * indicates where the arena access tube connects to the window access. In A and B, this connection is blocked by a cap (vi), as at the start of the protocol. This point is also indicated in Figure S8B by the same symbol. (C) Exterior view of the lab-window access. (D) A honeybee feeding from the wick in the Falcon tube collar of the window access. The complete width of D is approximately 3 cm. Note that photographic images have been sharpened and color-adjusted to clarify the view of the key features. Note that scale bars in photographs are approximate due to the image's perspective view.

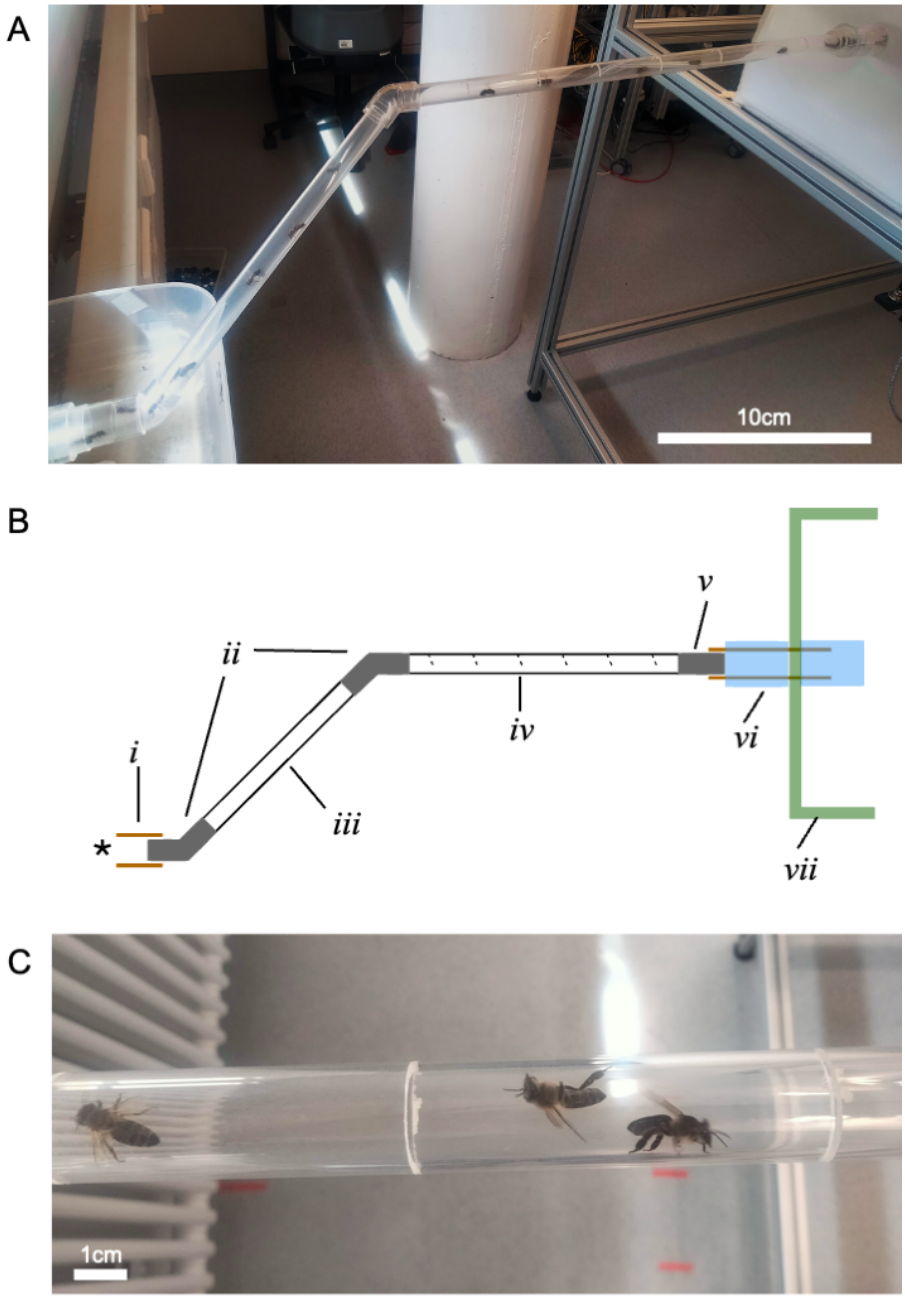


Figure S8. Arena access tube. (A) Photographic image (image sharpened and contrast increased for clarity). (B) Diagram of the access tube. Italic numerals indicate components: i) a 5-cm length section of 32-mm diameter tube, inserted into the T-junction of window access (vi in Figure S7). * indicates where the arena access tube connects to the window access, at the point also indicated in Figure S7B by the same symbol. ii) 45° pipe fittings for 20-mm diameter tube. iii) A 30-cm length of 20-mm diameter tubing. iv) 50-cm length of 20-mm diameter tubing with slits cut along its width. v) Horizontal pipe connector fitting for 20-mm tube. vi) A 10-mm section of 32-mm diameter pipe inserted through the flight arena hole (see main text) with two 32-mm horizontal pipe fittings fitted over either side to lock the tube in place. vii) Flight arena. (C) Honeybees within the arena access tube. Note that photographic images have been sharpened and color-adjusted to clarify the view of the key features.

Supplementary materials 3: Additional bee tracking figures

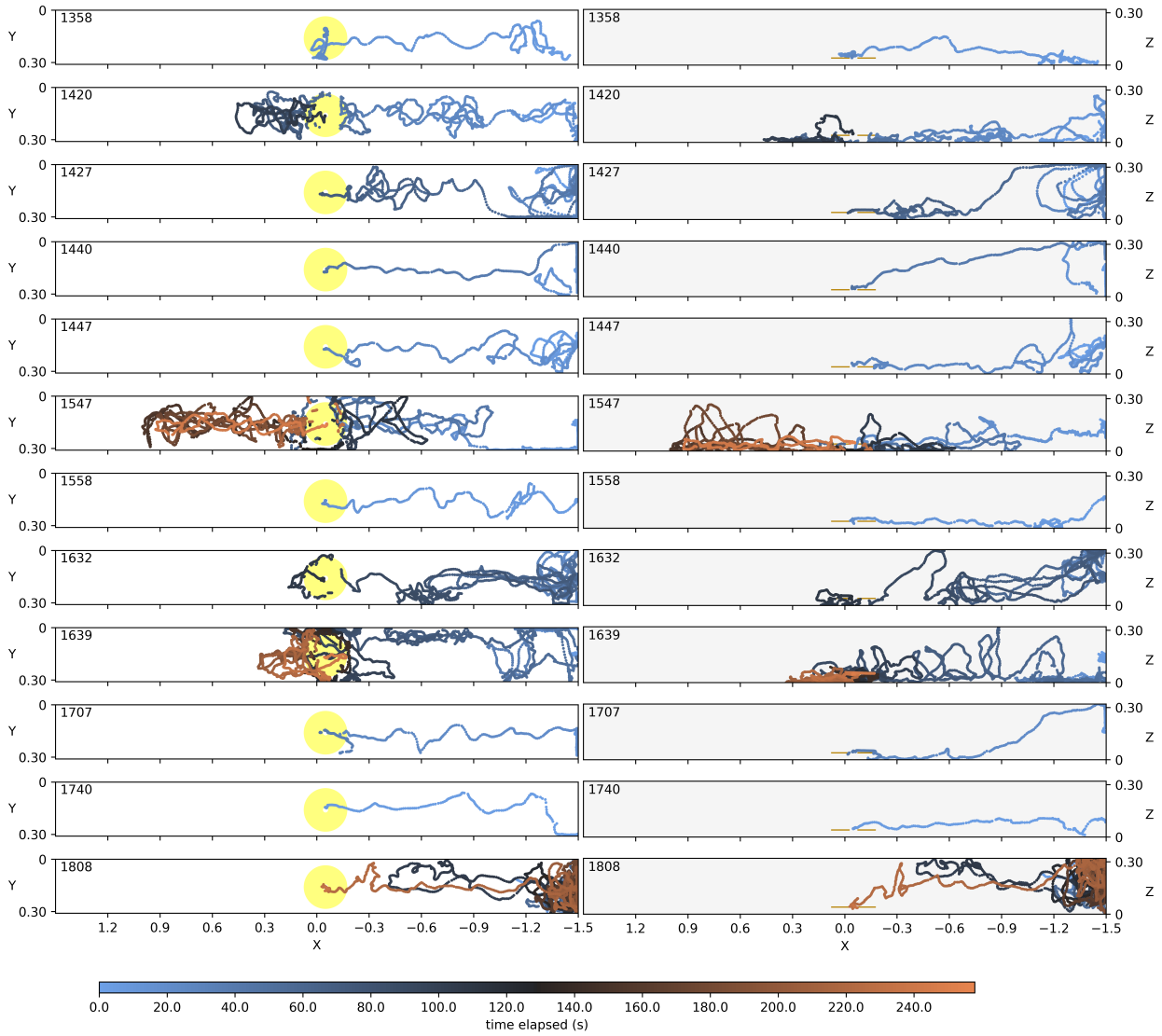


Figure S9. 3D tracking results of the 12 bees that successfully located the flower, re-tracked using the original tracking parameters given in config.toml. Each row of graphs shows the tracking for an individual bee. Individual bees are identified by a 4-digit code corresponding to the 24-h time when tracking began. This ID is shown in the top left of each panel (this time can be used to identify the BRAIDZ file, as all data in this step was collected on the same day). For each bee, graphs show the Braid position estimates throughout tracking in the x-y (left-hand side, white background) and x-z (right-hand side, gray background) axes. Axes are in meters and are as described for the tracking arena. Color of points indicates time elapsed (in seconds) since the bee entered the arena. This color scale is shared across all graphs. The artificial flower position during tracking is indicated by the yellow disc (xy graphs) or lines (xz graphs). As discussed in the protocol, Braid cannot track the bee once it is sufficiently inside the artificial flower or under its disc, so tracking ends when the bee enters the flower.

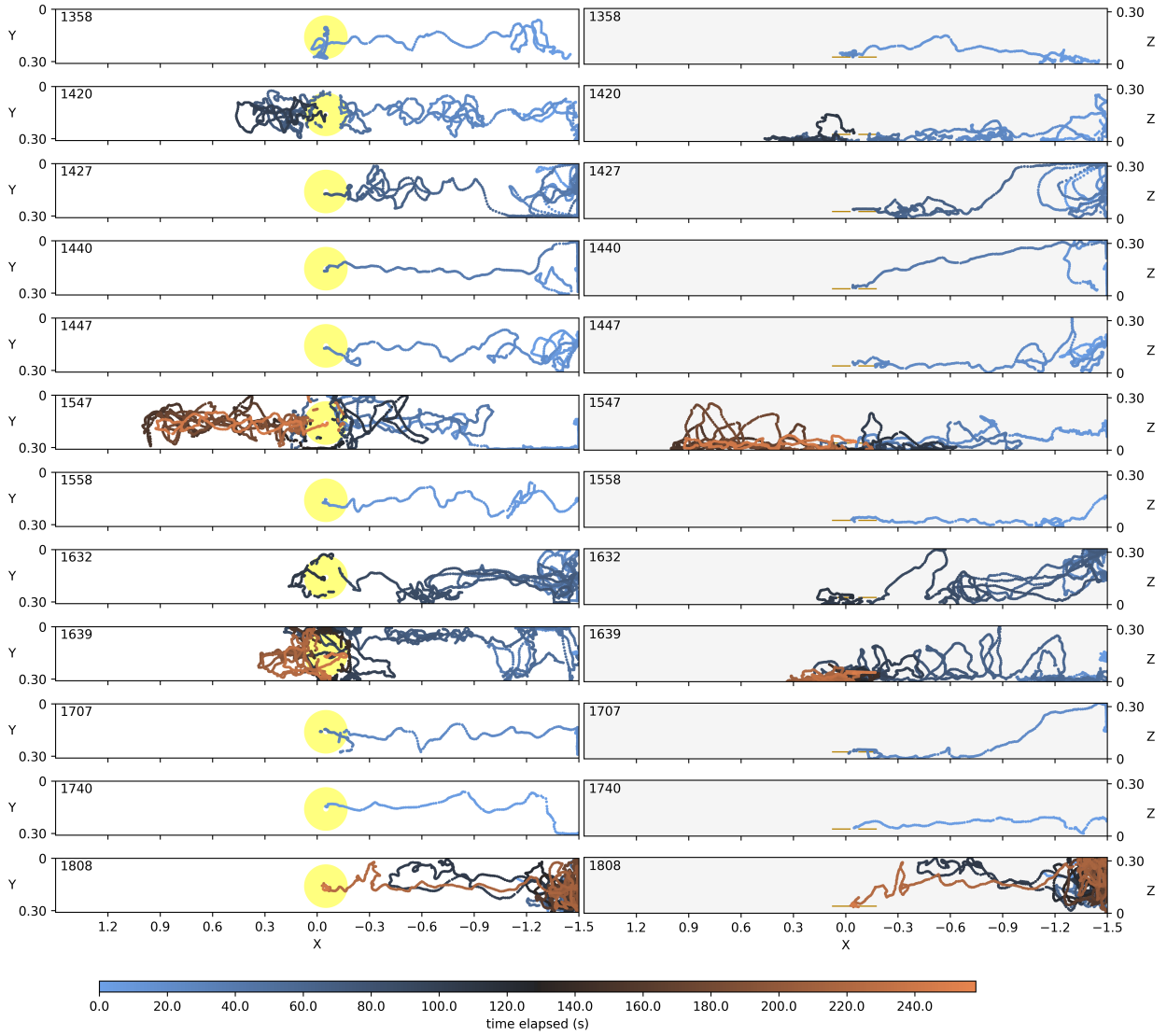


Figure S10. 3D tracking results of the 12 successful bees after being re-tracked by Braid using the alternative tracking parameters provided in retrack_paras.toml. For all other aspects of the graph, see the legend of Figure S9.