TITLE:

Long-term Video Tracking of Cohoused Aquatic Animals: A Case Study of the Daily Locomotor Activity of the Norway Lobster (*Nephrops norvegicus*)

AUTHORS AND AFFILIATIONS:

Jose A. Garcia[1,2] *, Valerio Sbragaglia[3,4] *, David Masip[1], Jacopo Aguzzi[2]

[1]Universitat Oberta de Catalunya, Barcelona, Spain
[2]Institute of Marine Sciences, Spanish National Research Council (CSIC), Barcelona, Spain
[3]Institute for Environmental Protection and Research (ISPRA), Livorno, Italy
[4]Department of Biology and Ecology of Fishes, Leibniz-Institute of Freshwater Ecology and Inland Fisheries, Berlin, Germany

*These authors contributed equally.

Corresponding Author:
Jose A. Garcia (jagarco@uoc.edu; jagarcia@icm.csic.es)

Email Addresses of Co-authors:
Valerio Sbragaglia (valeriosbra@gmail.com)
David Masip (dmasipr@uoc.edu)
Jacopo Aguzzi (jaguzzi@icm.csic.es)

KEYWORDS:
OpenCV, Python, video analysis, space occupancy, daily activity rhythms, tracking

SUMMARY:
Here we present a protocol to individually track animals over a long period of time. It uses computer vision methods to identify a set of manually constructed tags by using a group of lobsters as case study, simultaneously providing information on how to house, manipulate, and mark the lobsters.

ABSTRACT:
We present a protocol related to a video-tracking technique based on the background subtraction and image thresholding that makes it possible to individually track cohoused animals. We tested the tracking routine with four cohoused Norway lobsters (*Nephrops norvegicus*) under light-darkness conditions for 5 days. The lobsters had been individually tagged. The experimental setup and the tracking techniques used are entirely based on the open source software. The comparison of the tracking output with a manual detection indicates that the lobsters were correctly detected 69% of the times. Among the correctly detected lobsters, their individual tags were correctly identified 89.5% of the times. Considering the frame rate used in the protocol and the movement rate of lobsters, the performance of the video tracking has a good quality, and the representative results support the validity of the protocol in producing valuable data for research needs (individual space occupancy or locomotor activity patterns). The protocol

presented here can be easily customized and is, hence, transferable to other species where the individual tracking of specimens in a group can be valuable for answering research questions.

INTRODUCTION:

In the last few years, automated image-based tracking has provided highly accurate datasets which can be used to explore basic questions in ecology and behavior disciplines[1]. These datasets can be used for the quantitative analysis of animal behavior[2,3]. However, each image methodology used for tracking animals and behavior evaluation has its strengths and limitations. In image-based tracking protocols that use spatial information from previous frames in a movie to track animals[4-6], errors can be introduced when the paths of two animals cross. These errors are generally irreversible and propagate through time. Despite computational advances that reduce or almost eliminate this problem[5,7], these techniques still need homogeneous experimental environments for accurate animal identification and tracking.

The employment of marks that can be uniquely identified in animals avoids these errors and allows the long-term tracking of identified individuals. Widely used markers (*e.g.*, barcodes and QR codes) exist in industry and commerce and can be identified using well-known computer vision techniques, such as augmented reality (*e.g.*, ARTag[8]) and camera calibration (*e.g.*, CALTag[9]). Tagged animals have previously been used for high-throughput behavioral studies in different animal species, for example, ants[3] or bees[10], but some of these previous systems are not optimized for recognizing isolated tags[3].

The tracking protocol presented in this paper is especially suitable for tracking animals in one-channel imagery, such as infrared (IR) light or monochromatic light (particularly, we use blue light). Therefore, the method developed does not use color cues, being also applicable to other settings where there are constraints in the illumination. In addition, we use customized tags designed so as not to disturb the lobsters and, at the same time, allow recording with low-cost cameras. Moreover, the method used here is based on frame-independent tag detection (*i.e.*, the algorithm recognizes the presence of each tag in the image regardless of the previous trajectories). This feature is relevant in applications where animals can be temporarily occluded, or animals' trajectories may intersect.

The tag design allows its use in different groups of animals. Once the parameters of the method are set, it could be transferred to tackle other animal-tracking problems without the need for training a specific classifier (other crustaceans or gastropods). The main limitations of exporting the protocol are the size of the tag and the need for attachment to the animal (which makes it not suitable for small insects, such as flies, bees, *etc.*) and the 2D assumption for the animal movement. This constraint is significant, given that the proposed method assumes the tag size remains constant. An animal moving freely in a 3D environment (*e.g.*, fish) would show different tag sizes depending on its distance to the camera.

The purpose of this protocol is to provide a user-friendly methodology for tracking multiple tagged animals over a long period of time (*i.e.*, days or weeks) in a 2D context. The methodological approach is based on the use of open source software and hardware. Free and open source software permits adaptations, modifications, and free redistribution; therefore, the generated software improves at each step[11,12].

The protocol presented here focuses on a laboratory set up to track and evaluate the locomotor activity of four aquatic animals in a tank for 5 days. The video files are recorded from a 1 s time-lapse image and compiled in a video at 20 frames per second (1 recorded day occupies approximately 1 h of video). All video recordings are automatically postprocessed to obtain animal positions, applying computer vision methods and algorithms. The protocol allows obtaining large amounts of tracking data, avoiding their manual annotation, which has been shown to be time-intensive and laborious in previous experimental papers[13].

We use the Norway lobster (*Nephrops norvegicus*) for the case study; thus, we provide species-specific laboratory conditions to maintain them. Lobsters perform well-studied burrow emergence rhythms that are under the control of the circadian clock[14,15], and when cohoused, they form dominance hierarchy[16,17]. Hence, the model presented here is a good example for researchers interested in the social modulation of behavior with a specific focus on circadian rhythms.

The methodology presented here is easily reproduced and can be applied to other species if there is a possibility to distinguish between animals with individual tags. The minimum requirements for reproducing such an approach in the laboratory are (i) isothermal rooms for the experimental setup; (ii) a continuous water supply; (iii) water temperature control mechanisms; (iv) a light control system; (v) a USB camera and a standard computer.

In this protocol, we use Python[18] and OpenCV[19] (Open Source Computer Vision Library). We rely on fast and commonly applied operations (both in terms of implementation and execution), such as background subtraction[20] and image thresholding[21,22].

PROTOCOL:

The species used in this study is not an endangered or protected species. Sampling and laboratory experiments followed the Spanish legislation and internal institutional (ICM-CSIC) regulations regarding animal welfare. Animal sampling was conducted with the permission of the local authority (Regional Government of Catalonia).

1. Animal Maintenance and Sampling

NOTE: The following protocol is based on the assumption that researchers can sample *N. norvegicus* in the field during the night to avoid damage to the photoreceptors[23]. Exposure of *N. norvegicus* to sunlight must be avoided. After sampling, the lobsters are supposed to be housed in an acclimation facility similar to the one reported on previously[17,24], with a continuous flow of refrigerated seawater (13 °C). The animals used in this study are male at the intermoult state with a cephalothorax length (CL; mean ± SD) of 43.92 ± 2.08 mm (N = 4).

1.1. Keep the individuals in isolated compartments to avoid any damages due to individual fights (see **Figure 1a-d**).

1.2. Feed them about 3x a week at random times to not interfere with the circadian rhythms.

NOTE: In this experiment, mussels (approximately 4 g per lobster) were used as food. Mussels were bought from frozen food suppliers and were suitable for human consumption.

1.3. Use blue light (425 - 515 nm) to simulate light hours according to the spectral sensitivity of the species[25] and the environmental conditions at 400 m deep[26] (see **Figure 1c,d**).

NOTE: The facility used here has a vertical ceiling of two blue (478 nm) fluorescent lamps that produced a light intensity of 12 lx at 1 m of distance from the lamps. See **Figure 1a** for the ceiling lamps' position and see the **Table of Materials** for the manufacturer's and technical lamps' characteristics.

1.4. Adjust the photoperiod of the acclimation facility to 12/12 light/darkness hours or simulate the natural photoperiod of the local latitude.

1.5. Regulate the facility temperature to 13 °C and monitor 2x daily to check the temperature of the inflowing seawater is around 13 °C (see **Figure 1e**).

1.6. Regulate the inflow of seawater at a rate of about 4 L/min to maintain good oxygenation.

NOTE: The seawater circulates in an open circuit (no filters and additional pumps are used). The water supply depends on the main aquarium plant services.

[Place **Figure 1** here]

2. Tag's Construction

NOTE: The tag used here can be changed according to the characteristics of the target animal or other specific considerations.

2.1. Cut four circles of 40 mm in diameter from a black plastic sheet.

2.2. Cut from a white PVC plastic sheet two equilateral triangles with 26 mm sides.

2.3. Cut from a white PVC plastic sheet two circles of 26 mm in diameter.

2.4. Mark the center of the white triangles and circles and make a 10 mm hole in it.

2.5. Glue the four white shapes to the center of the four black circles.

[Place **Figure 2** here]

3. Experimental Setup

NOTE: The experimental arena is supposed to be in an experimental chamber independent from but in close proximity to the acclimation facility.

3.1. Set up an experimental chamber where the air temperature can be controlled and maintained at the same temperature as the seawater in the experimental arena.

3.2. Modify a fiberglass tank (1,500 x 700 x 300 mm) to be used as an experimental arena. Add four burrows using PVC flexible pipes at the bottom of the tank and stick sand on the surface where the lobsters are supposed to move (**Figure 3b-e**). For more details, see Sbragaglia *et al.*[17] and Aguzzi *et al.*[27].

3.2.1. Provide the experimental arena with submergible blue LEDs (472 nm, simulating light hours) and IR LEDs (850 nm, dark conditions) (see also **Figure 3a**)[17,24].

NOTE: LED light is used due to its low heat impact and the availability of usable electronic control and free hardware. An isolated facility with an environmental and seawater temperature of 13 ± 0.5 °C was used.

3.2.2. Always keep the IR LEDs switched on.

NOTE: The IR is needed to video record in dark conditions and in light conditions. It is not necessary to switch it off.

3.2.3. Connect the blue LEDs with an apparatus to manage the photoperiod. See the suggestions in the **Table of Materials**, and for more details, consult Sbragaglia *et al.*[17] (also shown in **Figure 3a**).

NOTE: Illumination in video- or image-automated analyses is a critical factor. Regular illumination without shadows all over the arena avoiding water surface reflections makes the posterior video or image analysis easier. In the context of this protocol, only 12/12 light/darkness conditions were used. Light and darkness were gradually achieved within 30 min, and a light-controller script is added as **Supplementary File**.

3.2.4. Place the chilled seawater inlet at one corner of the tank and the corresponding outlet at the opposite corner.

3.2.5. Regulate the seawater input at a flow rate of about 4 L/min.

3.2.6. Surround the tank with a black curtain in order to provide a full isolation from other light (**Figure 3a**).

3.3. Place the tripod to which the web camera is fixed to the side of the experimental arena and position the video camera above (130 cm) and at the center of the experimental arena(75 cm x 32.5 cm (see **Figure 3a**).

3.4. Check whether the video camera is in the centered position (see step 3.3) to make sure it has not been moved involuntarily.

3.5. Connect the web camera to a computer that is placed outside the curtain (**Figure 3a**).

3.5.1. Install the software to manage the time-lapse recording with the video camera.

NOTE: Time-lapse recordings depend on the movement's speed of the species. Also, see the **Table of Materials** for the camera, fisheye lens, PC, and software characteristics and manufacturers used here.

3.5.2. Adjust the parameters of the video recording according to the characteristics of the species.

NOTE: Considering the mobility rate of *N. norvegicus*, a 1 s time-lapse recording was used here, and the video was saved every 24 h.

3.5.3. Make sure to create a timestamp (including the date) in the time-lapse video (as this can help for the future manual scoring of the behavior).

[Place **Figure 3** here]

4. Experimental Trial and Animal Preparation

NOTE: All steps with animals must be done in the acclimation facility and under red light conditions according to the spectral sensitivity of the Norway lobster[25]. When moving the animals between the acclimation and the experimental facility, avoid any exposure of the lobsters to light, using an opaque black bag to cover the icebox.

4.1. Prepare an icebox previously separated into four submerged compartments with water at about 7 °C.

4.2. Prepare the four tags previously constructed and a fast glue, like cyanoacrylate.

4.3. Prepare a tray with crushed ice.

4.4. Select the four lobsters to be tagged in the acclimation facility and put each of them in a compartment of the icebox.

4.5. Wait for 30 min and, then, start the tagging procedure.

4.5.1. Take a lobster and put it on the crushed ice for 5 min to immobilize it and facilitate the tagging operation.

4.5.2. Dry the upper part of the lobster's cephalothorax with adsorptive paper and put a drop of fast glue on it.

4.5.3. Place the tag horizontally on top of the animal's cephalothorax, in contact with the glue, and wait enough time for it to harden (for about 20 s).

4.5.4. Return the lobster to its compartment in the icebox and proceed with the other three animals in the same way.

4.5.5. Put the lobsters back in the cell where they were previously and wait for 24 h to be sure that the tag is properly glued on.

4.5.6. Transfer the lobsters from the acclimation facility to the experimental chamber using the same icebox that was used for the tagging procedure.

4.6. Launch the video recording and wait for 5 min before introducing the tagged lobsters. Obtain an averaged background image from the initial 100 frames.

NOTE: Waiting a minimum of 1 min is mandatory to obtain background frames without tagged lobsters; they are needed for video processing.

4.7. Introduce the animals one by one in the experimentation tank inside their respective compartment, keeping the water in it (**Figure 4**).

4.8. Wait for them to get out; if they do not come out, help them gently by tilting the compartment.

[Place **Figure 4** here]

5. Video Analysis Script

5.1. Perform the analysis after completion of the experiment.

5.1.1. Launch the computer vision script for video analysis.

5.1.2. Launch Java program to calculate the positions and distance covered by the lobsters and insert the data in the database.

NOTE: This program is a Euclidean distance-based algorithm[28].

5.1.3. Launch SQL script to binning data as desired time interval (ex. 10 min).

6. Computer Vision Script for Video Analysis

NOTE: The script avoids fisheye image correction because it does not introduce a relevant error in the experimental setup. Nonetheless, it is possible to correct this with OpenCV[29] camera calibration functions based on vector and matrix rotation methods[30,31].

6.1. Select the Python[18] program language.

6.2. Select the OpenCV[19] image and video processing library.

6.3. Load a video.

NOTE: Video formats .avi or .mp4 were used in this experiment, but this is not mandatory. It depends on the FourCC[32] codecs installed in the operating system.

6.4. Perform the following steps for each frame *Fi* in the video.

6.4.1. Subtract the background[20] ***B*** (average of the last 100 frames, obtained from step 4.6) from the current frame ***Fi***, and update the background image ***B*** as ***Fi***. Use the function *BackgroundSubtractorMOG2* from the OpenCV[19] library (see the scripts in the **Supplementary File**).

6.4.2. Determine the set of regions of interest (ROIs) ***R*** from the pixels with relevant motion indicated by the background subtractor. Use the method *apply* from *BackgroundSubtractorMOG2* in the OpenCV[19] library (see the scripts in the **Supplementary File**). In the set, include the animal detections from the previous frame, to take into account nonmoving animals.

6.4.3. Perform the following steps for each ROI ***Ri***:

6.4.3.1. Apply the dilate function and compute the contours[33] of ROI ***Ri***. Use the functions *dilate* and *findContours* from the OpenCV[19] library (see the scripts in the **Supplementary File**).

6.4.3.2. Compute the hull area[34] ***hi*** in the number of pixels. Use the function *convexHull* from the OpenCV[19] library (see the scripts in the **Supplementary File**).

6.4.3.3. Compute the radius[35] ***ri*** of the ROI ***Ri***. Use the function *minEnclosingCircle* from the OpenCV[19] library (see the scripts in the **Supplementary File**).

6.4.3.4. Compute the solidity ***si*** of the ROI ***Ri***. Solidity is the ratio of the contour area (obtained in step 6.4.3.1) to its convex hull area (obtained in step 6.4.3.2) of the *Ri*.

6.4.3.5. Compute the aspect ratio ***ai*** of the ROI ***Ri***. Aspect ratio is the ratio between the width and the height of the *Ri*-bounding rectangle. The bounding rectangle is computed using the function *boundingRect* from the OpenCV[19] library.

6.4.4. Select a reduced set of ROIs as a candidate to contain the animals, by adjusting the properties for hull area, radius, solidity, and aspect ratio.

6.4.4.1. Check if ***hi*** is less than *500.0* or greater than *100000.0*. If so, discard the ROI ***Ri***. Otherwise, keep the ***Ri*** as a candidate ROI for the animal location.

6.4.4.2. Check if the ***ri*** is less than *40.0*. If so, discard the ROI ***Ri***. Otherwise, keep the ***Ri*** as a candidate ROI for the animal location.

6.4.4.3. Check if the ***si*** is less than *-4.0* discard the ROI ***Ri***. Otherwise, keep the ***Ri*** as a candidate ROI for the animal location.

6.4.4.4. Check if the ***ai*** is less than *0.15* or greater than *4.0*. Is so, discard the ROI ***Ri***. Otherwise, keep the ***Ri*** as a candidate ROI for the animal location.

NOTE: The use of ROIs reduces the computational cost, focusing the tag search on the animal's body region. Animal detections from previous frames are included to avoid wrong detections when the animals are not moving.

6.4.5. Analyze the animal ROIs to determine the tag identities. Execute de following steps for each ROI ***Ri*** and for each internal ROI ***Pi***, and extract the internal ROIs ***P***.

6.4.5.1. Binarize the grayscale image ***Pi*** using the *Otsu*[36] *thresholding* algorithm.

6.4.5.2. Compute the contours[33] of ***Pi***, as in step 6.4.3.1.

6.4.5.3. Compute the hull area[34] ***hi*** and the aspect ratio ***ai***, as in steps 6.4.3.2 and 6.4.3.5.

6.4.5.4. Compute the shape *moments*[37,38] ***mi*** of ***Pi***. Use the function *moments* from the OpenCV[19] library (see the scripts in the **Supplementary File**).

6.4.5.5. Select a reduced set of ROIs as a candidate to contain the tags, using the following criteria.

6.4.5.5.1. Check if ***hi*** is less than *150.0* or greater than *500.0*. If so, discard the ROI ***Pi***. Otherwise, keep the ***Pi*** as a candidate ROI for the tag location.

6.4.5.5.2. Check if the ***ai*** is less than *0.5* or greater than *1.5*. If so, discard the ROI ***Pi***. Otherwise, keep the ***Pi*** as a candidate ROI for the animal location.

6.4.5.5.3. Check if the ***mi*** is greater than *0.3*. If so, discard the ROI ***Pi***. Otherwise, keep the ***Pi*** as a candidate ROI for the animal location.

6.4.6. Classify the tag ROIs. Approximate a polygon[39] using the OpenCV[8] library for each selected ROI ***Pi***[19].

6.4.6.1. Check if there are exactly three vertices in the approximated polygon; assign the tag to the **triangle** class. Otherwise, assign the **circle** class to the tag region.

NOTE: Approximated polygon is stored using a matrix with the vertices.

6.4.6.2. Check the central pixel of the ROI **Pi**. If it is a **black** pixel, assign the ***Pi*** to the **holed** class. Otherwise, assign the **Pi** to the **white** class.

NOTE: The shape center is deduced from the moments calculated in step 6.4.5.4. Search the black pixels in an area of a 4-pixel radius around the center.

6.5. Save the frame data: frame date, frame time, shape class, x center shape coordinate, and y center shape coordinate.

6.6. Continue with the next frame or end the process (see **Figure 4** as a visual example of the script execution). See **Figure 5** below as a visual example of the working script steps and watch **Video 1** as an example of script functioning.

[Place **Figure 5** here]

REPRESENTATIVE RESULTS:

We manually constructed a subset of the experimental data to validate the automated video analysis. A sample size of 1,308 frames with a confidence level of 99% (which is a measure of security that shows whether the sample accurately reflects the population, within its margin of error) and a margin of error of 4% (which is a percentage that describes how close the response the sample gave is to the real value in the population) was randomly selected, and a manual annotation of the correct identification of ROIs and the correct identification of the tag within each ROI was performed. Note that a single frame may contain a variable number of ROIs within an undetermined range because some lobsters may be concealed inside the burrows or one ROI contains two or more animals or false detections.

The total number of animals in the 1,308 frames was 3,852 (manually annotated ROIs). The method revealed 3,354 animal detections. A total of 701 (21%) of these detections were false positives (*i.e.*, the number of ROIs where the lobster was confused with the background). Of the total number of animals counted, 2,653 detections (79%) were correctly matching (*i.e.*, the number of times the classifier correctly recognized the presence of a lobster in the detected regions; see also **Figure 6a, b**). With respect to the total 3,852 ROIs present in the 1,308 frames, the script detects 69% of the individuals.

Regarding the tag detection, the script identified 2,353 ROI candidates as tags (89% of the 2,653 detected regions with animals). The classifier successfully identified as class tag 1,808 of these tags (in which the candidate is classified as a circle, triangle, holed circle, or holed triangle) and missed 545 cases (23% of the 2,353 ROI candidates for tag). Related to the tag classification, 1,619 are correctly identified (89.5%, **Figure 6f**). Only 70 tags where wrongly classified (3.8% error, **Figure 6e**), and the remaining 119 (6.6%) corresponded to false positives (internal ROIs identified as tag that corresponded to animal parts, such as claws; **Figure 6d**).

[Place **Figure 6** here]

After the video analysis was completed, the obtained positions (X, Y) data can be used to evaluate different behavioral patterns of the lobsters. For example, we plotted a space occupancy map using two-dimensional kernel density estimation with an axis-aligned bivariate normal kernel, evaluated on a square grid[41,42] with the best performance are automated estimated by the statistical algorithm. A higher color intensity represents the areas were the lobsters spent a higher percentage of their time (**Figure 7**). **Video 2** gives a visual example of animal tracking.

Another example is represented by the daily activity rhythms of the lobsters, plotted as millimeters and covered at 10 min binned time intervals (**Figure 8**). We removed the data corresponding to the first 24 h of the experiment, which corresponded to the animals' environmental adaptation process.

[Place **Figure 7** here]

[Place **Figure 8** here]

FIGURE AND TABLE LEGENDS:

Figure 1: Facility acclimation views. (**a**) Tank shelves. (**a1**) Seawater input. (**a2**) Fluorescent ceiling lights. (**b**) Detail of blue light illumination. (**c**) Animal cell detail. (**d**) Detail of an isolated facility control panel. (e) Temperature setting for one of the entrances.

Figure 2: The four tags used for the individual tagging of the lobsters. Circle, circle-hole, triangle, triangle-hole.

Figure 3: Experimental setup. (**a**) Diagram of the assembly of the experimental tank and video acquisition. (**b**) General view of the experimental tank. (**c**) Bottom view of the experimental tank, indicating the artificial burrows. (**d**) Top view, showing the bottom of the experimental tank. (**e**) Detail of one of the burrow entrances.

Figure 4: Raw video frame. An example of a representative frame from one of the time-lapse videos collected during the experiments. At the upper right corner, we show the time stamp with the date, time, and frame. Notice the differences in the tank illumination in the image's lower

corner.

Figure 5: Relevant steps of the video-processing script. (**1**) Evaluate the background subtraction motion over the mean of the last 100 frames. (**2**) Result of the background subtraction algorithm. (**3**) Apply a dilate morphological operation to the white-detected areas. (**4**) Apply a fix, static, main ROI; the yellow polygon corresponds to the bottom tank area. (**5**) Calculate contours for each white-detected region in the main ROI and perform a structural analysis for each detected contour. (**6**) Check structural property values and, then, select second-level ROI candidates. (**7**) Binarize the frame using an Otsu thresholding algorithm; the script works only with second-level ROIs. (**8**) For each binarized second-level ROI, calculate the contours of the white regions and perform a structural analysis for each detected contour. (**9**) Check the structural property values and, then, select internal ROI candidates. (**10**) For each contour in the internal ROI candidate, calculate the descriptors/moments. (**11**) Check if the detected shape matches with the model shape and approximate a polygon to the best match candidates. (**12**) Check the number of vertices of the approximate polygon and determine the geometric figure: circle or triangle. (**13**) Calculate the figure center and check if black pixels occur; if yes, it is a holed figure. (**14**) Visual result after frame analysis.

Figure 6: Representative views from frames showing the most common experimental situations during video analysis. (**a**) Wrong animal detection, a background area is detected. (**b**) Animal misdetection. Two animals are close together and only one is detected. (**c**) Shape misdetection. The animal is detected (blue rectangle) but the tag is not detected. (**d**) Fake shape detection. Two shapes are detected, one is a claw. (**e**) Incorrect classification of a shape. A triangle is classified as triangle-hole. (**f**) Ideal situation. All animals are detected, and the tags are correctly identified.

Figure 7: Space occupancy map. The chart only shows the bottom tank area that is the animal displacement area (see the yellow polygon in **Figure 5**). The areas were the different tagged lobsters spent more time appear colored; a higher color intensity means more occupancy time.

Figure 8: Daily activity rhythms of the lobsters plotted as millimeters and covered at 10 min binned time intervals. Grey bands indicate the hours of darkness at 12/12 light/darkness, with the sunset time starting at 7.00 a.m. and the sunrise time starting at 7.00 p.m.

Figure 9: Detail of frame binarization errors. A red circle shows how lobsters and tags are detected as a unique object.

Video 1: Desktop record of an example of a running video analysis script. The video shows in 2 min and 27 s 1 h of real-time footage (3,625 frames). Notice that there is no error accumulation for the animal and tag misdetections and unidentified events while the recording is being made.

Video 2: Video of the animal tracking after the locomotor analysis. We used X, Y image pixel coordinates obtained from the video analysis and stored them into the database, to draw the animal track in the recorded videos as an example of the video analysis script. The longer the track, the faster the animal moves and the more distance traveled. In this case, 30 s of video corresponds to 12 min of real-time.

DISCUSSION:

The performance and representative results obtained with the video-tracking protocol confirmed its validity for applied research in the field of animal behavior, with a specific focus on social modulation and circadian rhythms of cohoused animals. The efficiency of animal detection (69%) and the accuracy of tag discrimination (89.5%) coupled with the behavioral characteristics (*i.e.*, movement rate) of the target species used here suggest that this protocol is a perfect solution for long-term experimental trials (*e.g.*, days and weeks). Moreover, the protocol offers the basic advantage of being easy to use and faster in its development and customization with respect to other techniques, such as automatic learning algorithms and neural networks[43]. The tracking techniques used here represent the final refinement of an experimental activity started with a slightly different approach[44].

A critical step in the protocol is the tag design; it should be considered that the implementation of other tag designs could improve the performance of the Otsu binarization. For example, one of the sources of error reported here was the misdetection between the black outside circle in the tag and the white internal geometric form (see **Figure 9** with a binarized frame with a detail of this error). It is possible to improve the binarization process, increasing the diameter (2 - 3 mm) of the black circle outside the white internal geometric form, or checking the colors (white/black). We do not consider the use of image morphological functions like *erode* or *dilate* when trying to correct this error, given that these operations modify the structural properties of the tag imaged, being, therefore, not possible to maintain the threshold values of the script. In conclusion, it is advisable to adapt the tag design to the target animal species anatomy. That would involve the adjustment of the script threshold values and the structural properties according to the new design.

[Place **Figure 9** here]

The most relevant source of errors was the missed detection of the ROIs (both external and internal). The video analysis script presented here is only able to keep track of individuals that are not moving or are hidden for a period of fewer than 100 frames; to avoid problems with this, we stored the last position of an individual until it is detected again. This parameter might influence the results of missing immobile or hidden animals. This fact must be taken into account when using this protocol with species showing different mobility rates than the ones presented here for the lobsters. The video frame rate and analysis script should be modified and adjusted to the species used according to its specific behavior.

One major challenge was to obtain a monochromatic blue (472 nm) and IR (850 nm) illumination, to avoid the possibility of retinal damage and adjust the light environmental conditions to the animal's habitat[23,45]. Therefore, the color component in video recording is not relevant and video recordings were performed in grayscale. The system helps to program different light time periods and modifies the illumination system according to the target species' characteristics changing the

LED's light spectrum. Another customization to consider in the protocol presented here is the movement rate of the target animal. For this specific case, the frame rate used was 1 frame per second, producing video files of about 1 h length corresponding to 24 h of experimental recording. These two customizations (grayscale and frame rate) allowed us to obtain video files with a reduced size that were easy to work with and reduced the storage capacity and machine time for tracking.

A limitation of the described method is that it has only been tested with the species presented here; however, there are no specific reasons concerning the application of this protocol to other species that allow the carrying of identification tags. Another significant limitation is that the protocol is not suited to track the movements of animal appendices (*e.g.*, chelae). For example, decapod crustaceans use chelae movements to display dominance among conspecifics. Future implementations are aimed at improving this aspect.

The proposed protocol resembles previous existing commercial software[46] and published methods SwisTrack and idTracker[7,47]. The commercial software[46] uses background subtraction to detect animals, similar to the scripts presented here. Although it covers a wider spectrum of applications, it is programmed using a commercially interpreted program language[48], which is not an open source solution and is economically costly. The SwisTrack[47] method uses the OpenCV library, similarly to the approach presented here. Nevertheless, it is coded in C++. We used Python code, which is usually easier to adapt to the particular needs of each environment. IdTracker[7] is a strong proposal coded in a commercially interpreted program language[48] but targets nonmarked animal applications. The correct outcomes of the tracking can be compromised when animals are occluded for a long period of time as occurs in the experimental conditions presented here. The method presented here processes each frame independently and is not influenced by the previous trajectory of the animal. Therefore, an error in a specific frame does not propagate to future frames. This fact is relevant in this application but also constraints the method presented here to a specific set of animals (those that allow manual tagging).

Another aspect to consider is that we have used free software during the development of the protocol, including the postprocessing and storage of the data generated by the video analysis script, as well as the code used to control the lighting system. The processed data are stored in a free relational database system (MySQL). These processed data can be obtained through queries in Standard Query Language (SQL) according to the desired format. The reader can modify and adapt the proposed open code and freely adapt it to particular needs.

With regard to the method toxicity, the only delicate step is the gluing of the tag to the animal. We used cyanoacrylate glue due to its low toxicity, its wide medical use[28], and its wide use in aquaria for fragging corals and fixing the fragments with glue[29]. The major concern about its use is the vapor toxicity for humans. We reduced the exposition to the minimum. The Health and Safety Executive and the United States National Toxicology Program have concluded that the use of ethyl cyanoacrylate is safe[49].

Future applications of this protocol are the automation of the detection of other behaviors of burrowing crustaceans (*e.g.*, fights, burrow dominance). We also plan to improve the algorithm to obtain real-time video analysis and to use Convolutional Neural Networks[50] for improved animal detection.

ACKNOWLEDGMENTS:
The authors are grateful to the Dr. Joan B. Company that funded the publication of this work. Also, the authors are grateful to the technicians of the experimental aquarium zone at the Institute of Marine Sciences in Barcelona (ICM-CSIC) for their help during the experimental work.

This work was supported by the RITFIM project (CTM2010-16274; principal investigator: J. Aguzzi) founded by the Spanish Ministry of Science and Innovation (MICINN), and the TIN2015-66951-C2-2-R grant from the Spanish Ministry of Economy and Competitiveness.

DISCLOSURES:
The authors have nothing to disclose.

REFERENCES:
1. Dell, A.I. *et al.* Automated image-based tracking and its application in ecology. *Trends in Ecology & Evolution*. **29** (7), 417-428, doi: 10.1016/j.tree.2014.05.004 (2014).

2. Berman, G.J., Choi, D.M., Bialek, W., Shaevitz, J.W. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*. **11** (99), doi: 10.1098/rsif.2014.0672 (2014).

3. Mersch, D.P., Crespi, A., Keller, L. Tracking Individuals Shows Spatial Fidelity Is a Key Regulator of Ant Social Organization. *Science*. **340** (6136), 1090, doi: 10.1126/science.1234316 (2013).

4. Hedrick, T.L. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspiration & Biomimetics*. **3** (3), 034001 (2008).

5. Branson, K., Robie, A.A., Bender, J., Perona, P., Dickinson, M.H. High-throughput ethomics in large groups of *Drosophila*. *Nature Methods*. **6** (6), 451-457, doi: 10.1038/nmeth.1328 (2009).

6. de Chaumont, F. *et al.* Computerized video analysis of social interactions in mice. *Nature Methods*. **9**, 410 (2012).

7. Pérez-Escudero, A., Vicente-Page, J., Hinz, R.C., Arganda, S., de Polavieja, G.G. idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature Methods*. **11** (7), 743-748, doi: 10.1038/nmeth.2994 (2014).

8. Fiala, M. ARTag, a fiducial marker system using digital techniques. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego, CA (June 20-25, 2005).

9. Atcheson, B., Heide, F., Heidrich, W. CALTag: High Precision Fiducial Markers for Camera Calibration. In *Proceedings of the Vision, Modeling, and Visualization Workshop 2010, Siegen, Germany*. Edited by Koch, R., Kolb, A., Rezk-Salama, C., 41-48. Siegen, Germany (2010).

10. Crall, J.D., Gravish, N., Mountcastle, A.M., Combes, S.A. BEEtag: A Low-Cost, Image-Based Tracking System for the Study of Animal Behavior and Locomotion. *PLoS ONE*. **10** (9), e0136487, doi: 10.1371/journal.pone.0136487 (2015).

11. Charpentier, R. Free and Open Source Software: Overview and Preliminary Guidelines for the Government of Canada | TIM Review. http://timreview.ca/article/127 (2008).

12. Crowston, K., Wei, K., Howison, J. Free/Libre Open Source Software Development: What We Know and What We Do Not Know. *ACM Computing Surveys*. **44** (2), 1-35 (2012).

13. Edmonds, N.J., Riley, W.D., Maxwell, D.L. Predation by *Pacifastacus leniusculus* on the intra-gravel embryos and emerging fry of *Salmo salar*. *Fisheries Management and Ecology*. **18** (6), 521-524, doi: 10.1111/j.1365-2400.2011.00797.x (2011).

14. Sbragaglia, V. *et al.* Identification, Characterization, and Diel Pattern of Expression of Canonical Clock Genes in *Nephrops norvegicus* (Crustacea: Decapoda) Eyestalk. *PLoS ONE*. **10** (11), e0141893, doi: 10.1371/journal.pone.0141893 (2015).

15. Sbragaglia, V. *et al.* Dusk but not dawn burrow emergence rhythms of *Nephrops norvegicus* (Crustacea: Decapoda). *Scientia Marina*. **77** (4), 641-647 (2013).

16. Katoh, E., Sbragaglia, V., Aguzzi, J., Breithaupt, T. Sensory Biology and Behaviour of *Nephrops norvegicus*. *Advances in Marine Biology*. **64**, 65-106, doi: 10.1016/B978-0-12-410466-2.00003-0 (2013).

17. Sbragaglia, V. *et al.* Fighting over burrows: the emergence of dominance hierarchies in the Norway lobster (*Nephrops norvegicus*). *The Journal of Experimental Biology*. **220** (24), 4624-4633, doi: 10.1242/jeb.165969 (2017).

18. Welcome to Python.org. https://www.python.org/.

19. Bradski, G. The OpenCV Library | Dr Dobb's. http://www.drdobbs.com/open-source/the-opencv-library/184404319 (2000).

20. Piccardi, M. Background subtraction techniques: a review. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*. The Hague, Netherlands (October 10-13, 2004).

21. Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*. **13** (1), 146, doi: 10.1117/1.1631315 (2004).

22. Lai, Y.-K., Rosin, P.L. Efficient Circular Thresholding. *IEEE Transactions on Image Processing*.

23 (3), 992-1001, doi: 10.1109/TIP.2013.2297014 (2014).

23. Gaten, E. Light-induced damage to the dioptric apparatus of *Nephrops norvegicus* (L.) and the quantitative assessment of the damage. *Marine Behaviour and Physiology*. **13** (2), 169-183, doi: 10.1080/10236248809378671 (1988).

24. Sbragaglia, V. *et al.* An automated multi-flume actograph for the study of behavioral rhythms of burrowing organisms. *Journal of Experimental Marine Biology and Ecology*. **446**, 177-185, doi: 10.1016/j.jembe.2013.05.018 (2013).

25. Johnson, M.L., Gaten, E., Shelton, P.M.J. Spectral sensitivities of five marine decapod crustaceans and a review of spectral sensitivity variation in relation to habitat. *Journal of the Marine Biological Association of the United Kingdom*. **82** (5), 835-842, doi: 10.1017/S0025315402006203 (2002).

26. Markager, S., Vincent, W.F. Spectral light attenuation and the absorption of UV and blue light in natural waters. *Limnology and Oceanography*. **45** (3), 642-650, doi: 10.4319/Lo.2000.45.3.0642 (2000).

27. Aguzzi, J. *et al.* A New Laboratory Radio Frequency Identification (RFID) System for Behavioural Tracking of Marine Organisms. *Sensors*. **11** (10), 9532-9548, doi: 10.3390/s111009532 (2011).

28. Audin, M. *Geometry [Recurso electrónico]*. Springer. Berlin and Heidelberg, Germany (2003).

29. Structural Analysis and Shape Descriptors — OpenCV 2.4.13.7 documentation. https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#void%20HuMoments(const%20Moments&%20m,%20OutputArray%20hu) (2018).

30. Slabaugh, G.G. CiteSeerX — Computing Euler angles from a rotation matrix. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.371.6578.

31. Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **22** (11), 1330-1334, doi: 10.1109/34.888718 (2000).

32. www.FOURCC.org - Video Codecs and Pixel Formats. https://www.fourcc.org/ (2011).

33. Suzuki, S., be, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*. **30** (1), 32-46, doi: 10.1016/0734-189X(85)90016-7 (1985).

34. Sklansky, J. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*. **1** (2), 79-83, doi: 10.1016/0167-8655(82)90016-2 (1982).

35. Fitzgibbon, A.W., Fisher, R.B. A Buyer's Guide to Conic Fitting. In *Proceedings of the British*

Machine Conference. Edited by Pycock, D., 51.1-51.10, BMVA Press. doi: 10.5244/C.9.51 (1995).

36. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. **9** (1), 62-66, doi: 10.1109/TSMC.1979.4310076 (1979).

37. Hu, M.-K. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*. **8** (2), 179-187, doi: 10.1109/TIT.1962.1057692 (1962).

38. Structural Analysis and Shape Descriptors — OpenCV 2.4.13.6 documentation. https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=cvmatchshapes#humoments (2018).

39. Douglas, D.H., Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*. **10** (2), 112-122, doi: 10.3138/FM57-6770-U75U-7727 (1973).

41. Kahle, D., Wickham, H. ggmap: Spatial Visualization with ggplot2. *The R Journal.* **5/1**, 144-161 (2013).

42. Venables, W.N., Ripley, B.D. *Modern Applied Statistics with S*. Springer. New York, NY (2010).

43. Abbas, Q., Ibrahim, M.E.A., Jaffar, M.A. A comprehensive review of recent advances on deep vision systems. *Artificial Intelligence Review*. doi: 10.1007/s10462-018-9633-3 (2018).

44. Menesatti, P., Aguzzi, J., Costa, C., García, J.A., Sardà, F. A new morphometric implemented video-image analysis protocol for the study of social modulation in activity rhythms of marine organisms. *Journal of Neuroscience Methods*. **184** (1), 161-168, doi: 10.1016/j.jneumeth.2009.07.013 (2009).

46. Noldus. Video tracking software | EthoVision XT. https://www.noldus.com/animal-behavior-research/products/ethovision-xt.

47. Correll, N. *et al.* SwisTrack: A Tracking Tool for Multi-Unit Robotic and Biological Systems. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China (October 9-15, 2006).

48. MATLAB - MathWorks - MATLAB & Simulink. https://www.mathworks.com/products/matlab.html.

49. Cary, R. *Methyl Cyanoacrylate and Ethyl Cyanoacrylate*. World Health Organization. Geneva, Switzerland (2001).

50. Krizhevsky, A., Sutskever, I., Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*. Edited

by Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (2012).

Figure 4 Click here to access/download;Figure;figure4.eps

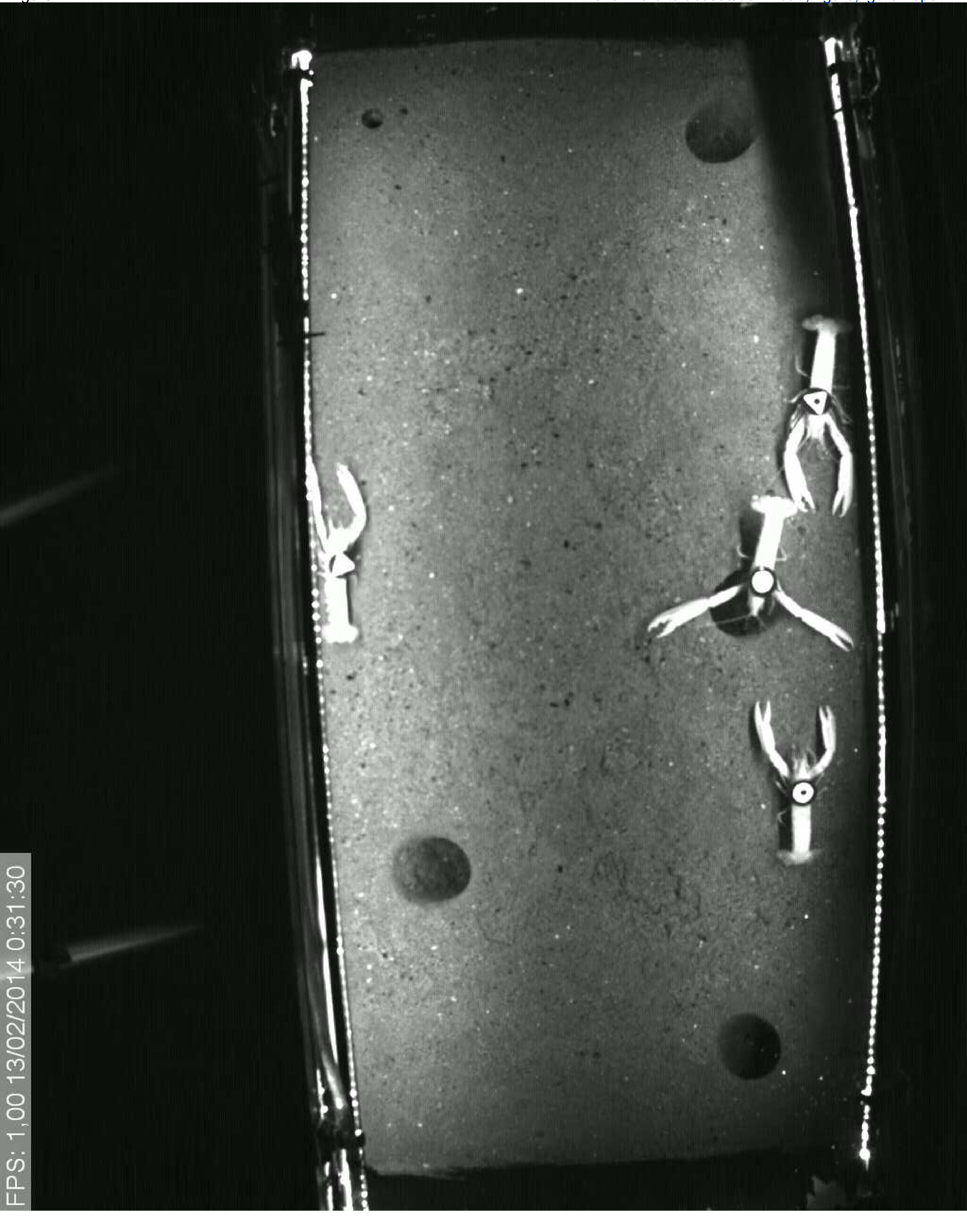


Figure 5

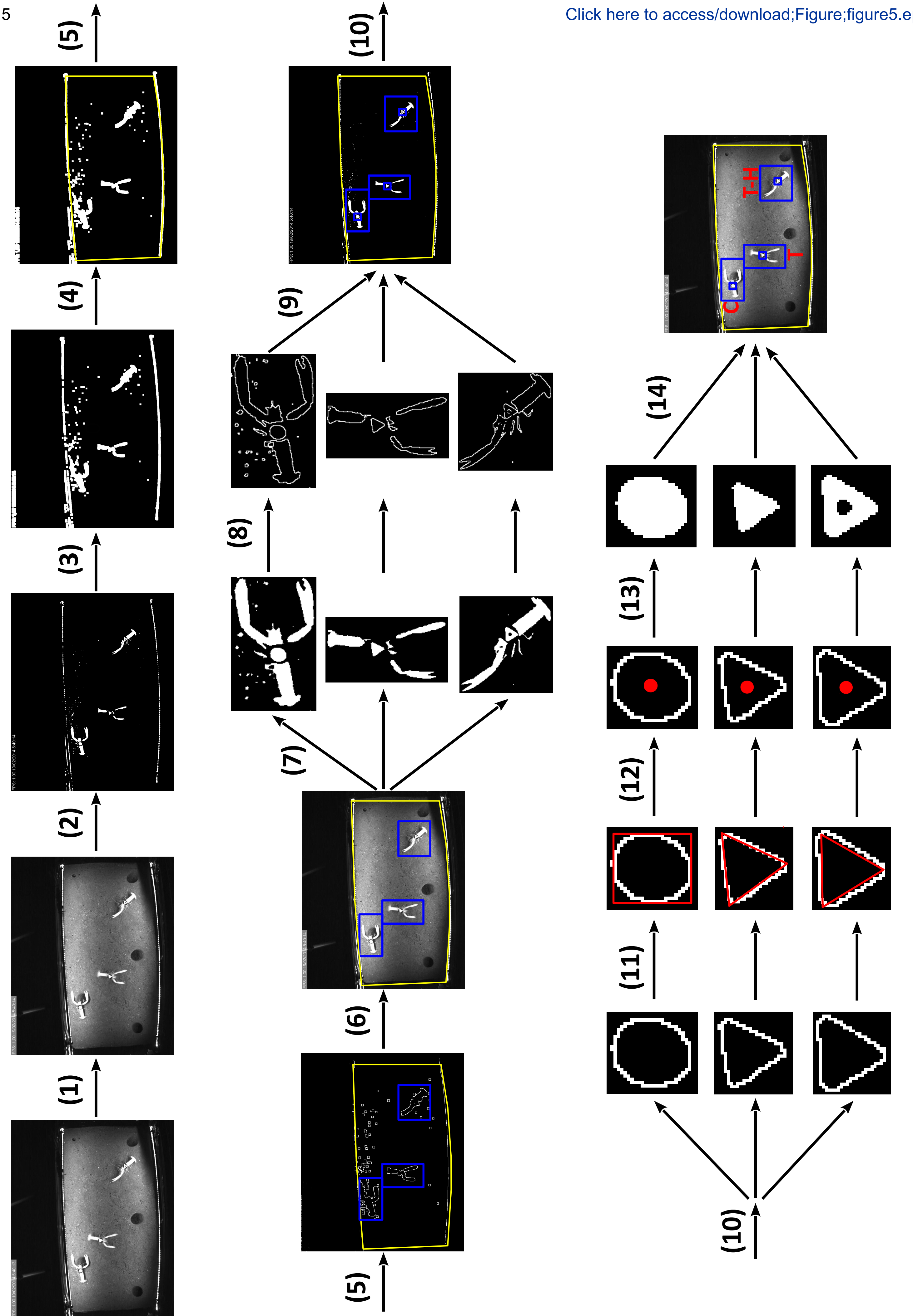
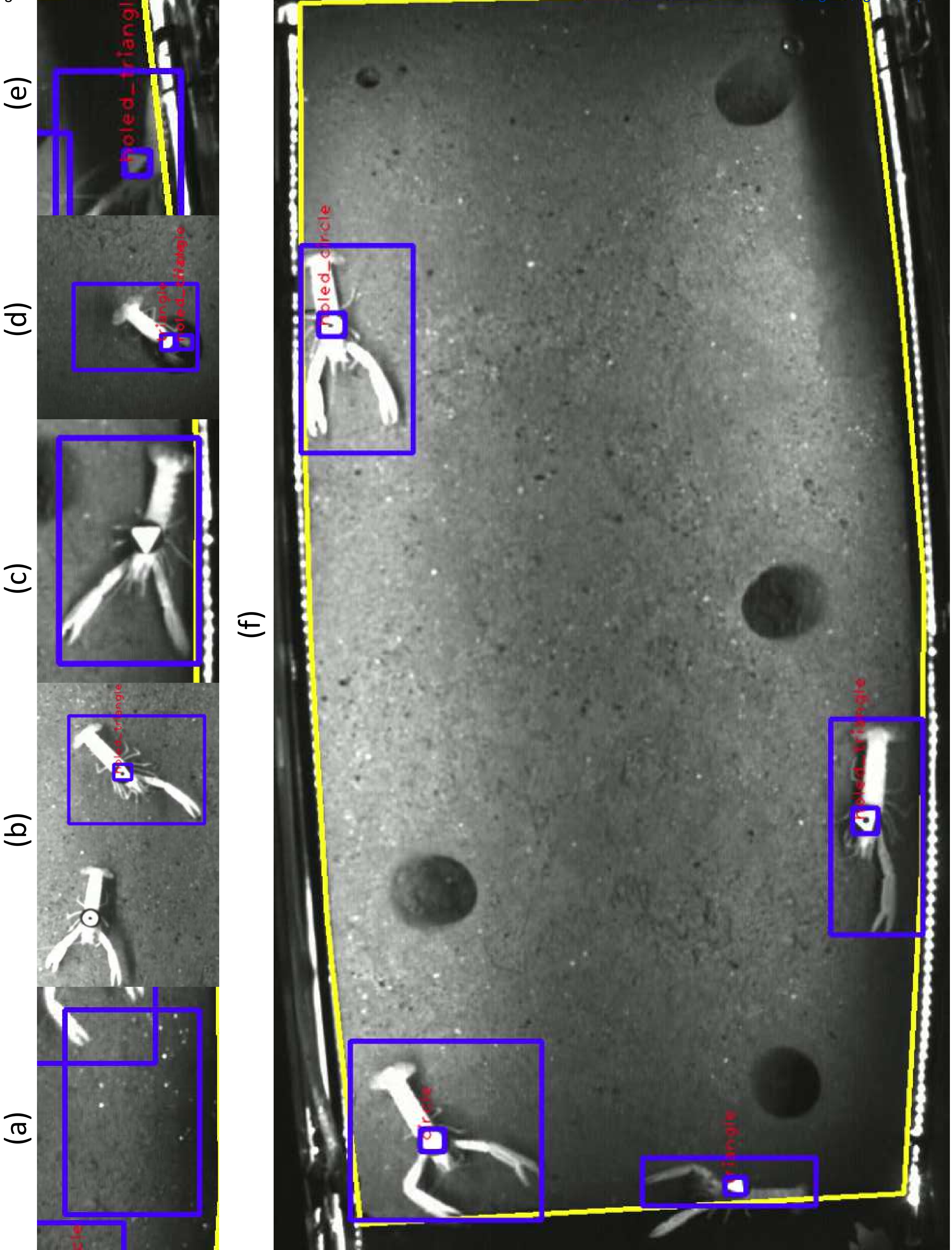


Figure 6



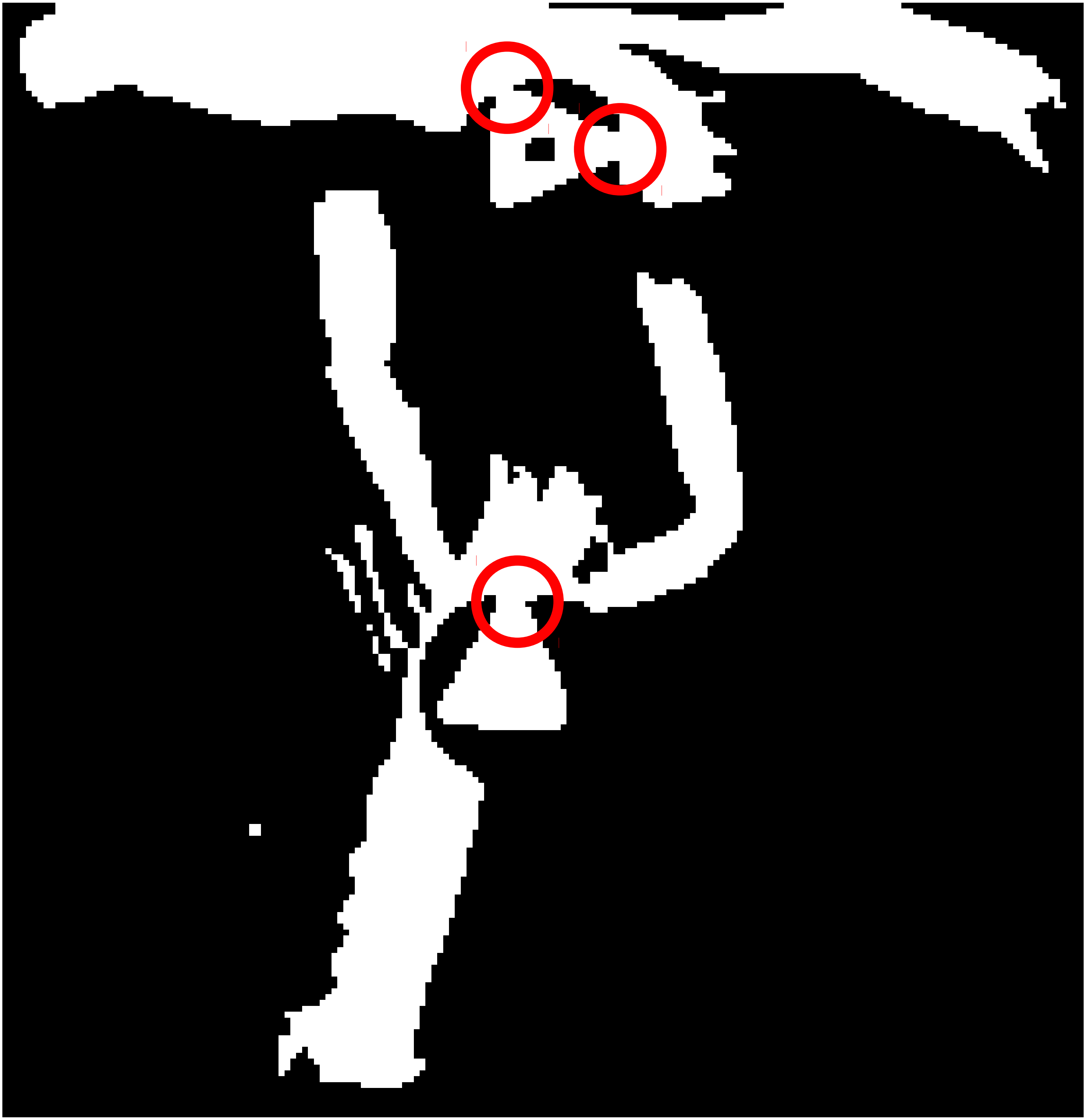


Figure 1

Click here to access/download
Video or Animated Figure
figure1.svg

Figure 2

Click here to access/download
Video or Animated Figure
figure2.svg

Figure 3

Click here to access/download

Video or Animated Figure

figure3.svg

Figure 7

Click here to access/download

Video or Animated Figure

figure7.svg

Figure 8

Click here to access/download
Video or Animated Figure
figure8.svg

Click here to access/download
Video or Animated Figure
video1.mov

Click here to access/download

Video or Animated Figure

video2.mov

Name of Material/ Equipment	Company	Catalog Number	Comments/Description
Tripod 475	Manfrotto	A0673528	Discontinued
Articulated Arm 143	Manfrotto	D0057824	Discontinued
Camera USB 2.0 uEye LE	iDS	UI-1545LE-M	https://en.ids-imaging.com/store/products/came
Fish Eye Len C-mount f=6mm/F1.4	Infaimon	Standard Optical	https://www.infaimon.com/es/estandar-6mm
Glass Fiber Tank 1500x700x300 mm			
Black Felt Fabric			
Wood Structure Tank			5 Wood Strips 50x50x250 mm
Wood Structure Felt Fabric			10 Wood Strips 25x25x250 mm
Stainless Steel Screws			As many as necessary for fix wood strips structur
PC			2-cores CPU, 4GB RAM, 1 GB Graphics, 500 GB HI
External Storage HDD			2 TB capacity desirable
iSPY Sotfware for Windows PC	iSPY		https://www.ispyconnect.com/download.aspx
Zoneminder Software Linux PC	Zoneminder		https://zoneminder.com/
OpenCV 2.4.13.6 Library	OpenCV		https://opencv.org/
Python 2.4	Python		https://www.python.org/
Camping Icebox			
Plastic Tray			
Cyanocrylate Gel			To glue tag’s
1 black PVC plastic sheet (1 mm thickness)			Tag's construction
1 white PVC plastic sheet (1 mm thickness)			Tag's construction
4 Tag’s Ø 40 mm			Maked with black & white PVC plastic sheet
3 m Blue Strid Led Ligts (480 nm)			Waterproof as desirable
3 m IR Strid Led Ligts (850 nm)			Waterproof as desirable
6m Methacrylate Pipes Ø 15 mm			Enclosed Strid Led
4 PVC Elbow 45^{o} Ø 63 mm			Burrow construction
3 m Flexible PVC Pipe Ø 63 mm			Burrow construction
4 PVC Screwcap Ø 63 mm			Burrow construction
4 O-ring Ø 63 mm			Burrow construction
4 Female PVC socket glue / thread Ø 63 mm			Burrow construction
10 m DC 12V Electric Cable			Light Control Mechanism
Ligt Power Supply DC 12V 300 w			Light Control Mechanism

MOSFET, RFD14N05L, N-Canal, 14 A, 50 V, 3-Pin, IPAK (TO-251)	RS Components	325-7580	Light Control Mechanism
Diode, 1N4004-E3/54, 1A, 400V, DO-204AL, 2-Pines	RS Components	628-9029	Light Control Mechanism
Fuse Holder	RS Components	336-7851	Light Control Mechanism
2 Way Power Terminal 3.81mm	RS Components	220-4658	Light Control Mechanism
Capacitor 220 µF 200 V	RS Components	440-6761	Light Control Mechanism
Resistance 2K2 7W	RS Components	485-3038	Light Control Mechanism
Fuse 6.3x32mm 3A	RS Components	413-210	Light Control Mechanism
Arduino Uno Atmel Atmega 328 MCU board	RS Components	715-4081	Light Control Mechanism
Prototipe Board CEM3,3 orific.,RE310S2	RS Components	728-8737	Light Control Mechanism
DC/DC converter,12Vin,+/-5Vout 100mA 1W	RS Components	689-5179	Light Control Mechanism
2 SERA T8 blue moonlight fluorescent bulb 36 watts	SERA		Discontinued / Light isolated facility