A Lightweight Keypoint Matching Framework for Morphometric Landmark Detection

Hoang Ha Nguyen
University of Science and Technology of Hanoi
Bich Hai Ho
Vietnam Academy of Science and Technology, Institute of Information Technology
Hien Phuong Lai
FPT University, Department of Computing Fundamentals.
Hoang Tung Tran
University of Science and Technology of Hanoi
Huu Ton Le (le-huu.ton@usth.edu.vn)
University of Science and Technology of Hanoi
Anne - Laure Banuls
MIVEGEC, Univ of Montpellier, IRD, CNRS, LMI DRISA, IRD Montpelllier
Jorian Prudhomme
MIVEGEC, Univ of Montpellier, IRD, CNRS, LMI DRISA, IRD Montpelllier

Research Article

Keywords: lightweight, framework, morphometric landmark, Geometric morphometrics

Posted Date: December 2nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1109399/v1

A lightweight keypoint matching framework for morphometric landmark detection

Hoang Ha Nguyen[1,*], Bich Hai Ho[2], Hien Phuong Lai[3], Hoang Tung Tran[1], Anne Laure Bañuls[4], Jorian Prudhomme[4], and Huu Ton Le[1,*]

[1]University of Science and Technology of Hanoi, ICT Lab, Hanoi, 10000, Vietnam
[2]Vietnam Accademy of Science and Technology, Institute of Information Technology, Hanoi, 10000, Vietnam
[3]FPT University, Department of Computing Fundamentals, Hanoi, 10000, Vietnam
[4]MIVEGEC, Univ of Montpellier, IRD, CNRS, LMI DRISA, IRD Montpelllier, 34000, France
[*]nguyen-hoang.ha@usth.edu.vn,le-huu.ton@usth.edu.vn

ABSTRACT

Geometric morphometrics has become an important approach in insect morphology studies because it capitalizes on advanced quantitative methods to analyze shape. Shape could be digitized as a set of landmarks from specimen images. However, the existing tools mostly require manual landmark digitization, and previous works on automatic landmark detection methods do not focus on implementation for end-users. Motivated by that, we propose a novel approach for automatic landmark detection, based on visual features of landmarks and keypoint matching techniques. While still archiving comparable accuracy to that of the state-of-the-art method, our framework requires less initial annotated data to build prediction model and runs faster. It is lightweight also in terms of implementation, in which a four-step workflow is provided with user-friendly graphical interfaces to produce correct landmark coordinates both by model prediction and manual correction. The utility iMorph is freely available at https://github.com/ha-usth/WingLanmarkPredictor, currently supporting Windows, MacOS, and Linux.

Introduction

The study of biological forms increasingly utilizes the geometric morphometrics approach as it allows quantitative evaluation of morphology. The form of an object could be characterized by shape and size. They could be considered together, for example, in allometry or solely shape to differentiate forms of interest. Being invariant to position, orientation, and scale of object, shape is both indicative of organism evolution and suitable for comparative study. Shape variation among individuals and populations is due to biological processes such as environmental adaptation, disease, evolutionary diversification, etc. Defined as a collection of landmarks, shape could be analyzed using various advanced mathematical methods. Landmark itself is a two- or three-dimensional point represented by locus coordinates. Landmarks that constitute a shape are numbered, as shown in Fig. 3. The annotation of landmarks, the so-called landmark digitizing, in specimen images is normally done by a domain expert; however, it is time-consuming and error-prone. Therefore, in this work, we propose a framework to automatically identify landmarks using computer vision techniques.

We used several insect wing datasets in experiments as described in Table 1. These wings were color-stained and captured at a similar scale. All consists of two-dimensional landmarks located at the intersections of wing veins with the wing margin and at the intersections of cross veins with major veins. The landmarks of interest in each dataset are predefined by domain experts. In our framework, each landmark is considered independently, and we use the number assigned to each as a landmark class. The input is, hence, a set of landmark-annotated images, so-called sample set and a set of non-annotated images, so-called prediction set. The output contains estimated coordinates of landmarks for images in the prediction set.

Although there exist a number of previous works on automatic landmark detection as shown in the next section, they were not implemented as software for end-users. Tools for landmark manipulation and analysis such as tpsDig[1], XYOM[2], StereoMorph R package[3] provides manual landmark digitization only. To our understanding, there does not yet exist a utility specifically for landmark detection from images. Therefore, we implemented a utility called iMorph for that purpose. The utility features four main steps: (1) Data preparation, (2) Learning, (3) Prediction, and (4) Landmark correction. Step (2) and (3) illustrates our proposed method. We employ various visual feature extraction and the keypoint matching approach to learn prediction models. Once applied to testing data, the model performances are evaluated by a distance-to-ground-truth measure. Steps (1) and (4) allow end-users to preprocess images for improved performance and manually edit incorrect output landmarks. Our contribution is two-fold, regarding both methodology and implementation. Firstly, we propose a lightweight approach, requiring a small initial annotated dataset in a quick annotation process and producing a comparable performance

to the state-of-the-art methods. Secondly, the open-source iMorph utility comes with only a few graphical user interfaces, designed to be user-friendly and straightforward. We also provide recommended parameter choices in online tutorials.

Related works

Landmarks store important information about the shape of the object (insect wing in our case). Landmark detection is based on the visual information of the local region around different keypoints detected in the image. Landmarks of the same class usually have similar visual features while landmarks of different classes have distinguished features. Landmark detection methods are divided into two main approaches: (1) detection using handcrafted features and (2) detection using a deep learning model in which features are automatically extracted during the learning process (non-handcrafted features).

For methods using handcrafted features, candidate landmarks are first identified by using keypoint detection methods such as Harris detector, Laplacian of Gaussian detector, Difference of Gaussians detector, *etc*. Each candidate landmark is then described by a set of local visual features (*e.g.* Haarlike[4], SURF[5], HOG[6], etc.). Finally, the estimated landmark is detected, based on the extraction features, by using different classification models such as Random Forest, SVMs, and Logistic Regression. Several works for landmark detection in biological images have been proposed. The system presented by Loh *et al.*[7] for Drosophila wing landmarks still requires the user to annotate three specific keypoints in each image before calculating the landmarks automatically by template matching. The system of Palaniswamy *et al.*[8] for grayscale Drosophila wings automatically identifies landmarks by using the Probabilistic Hough Transform coupled to a template matching algorithm; however, the time for estimating the coordinates of 15 landmarks on an image is about 3 minutes. Vandaele *et al.*[9] describe each keypoint by a vector of visual features (RAW, SUB, GAUSSIAN SUB, SURF, or Haar-like) at different resolutions and use the Extremely Randomized Trees algorithm for training the landmark classifier. The results of their work show better performance compared to the algorithm of Lindner *et al.*[10] which is originally developed to detect cephalometric landmarks using Random Forest regression-voting and to the algorithm of Donner *et al.* (DMBL)[11] which trains a Random Forest classifier for predicting landmark locations and a regression using Hough Forests for refining the estimated positions.

The second approach for landmark detection uses deep learning models, specifically the convolutional neural networks (CNNs) to automatically extract features in the landmark recognition process. Each hidden layer in the CNN is built to extract some specific features. The adjustment of the weights of the convolutional matrices to extract different features is done automatically during the learning process of the CNN network. The advantage of this approach is that the optimal feature is automatically extracted by the model. However, this kind of method requires a large training dataset and the network architecture needs to be selected for each specific problem. Convolutional neural networks have been used and given good results for facial landmark recognition problem. Sun *et al.*[12] proposed a three-level convolutional network for estimate the position of facial keypoints. Zhang *et al.*[13] proposed a task-constrained deep model to optimize facial landmark detection together with correlated tasks. Few models are proposed for morphometric landmark detection in insect wing images because of the limited size of the dataset. Le *et al.*[14] used three convolutional models to predict the morphometric landmarks on 260 beetle images; some augmentation techniques are used to increase the size of the dataset but overfitting still appears in two models.

The existing methods are proposed for specific datasets. They usually require a large enough training dataset (especially for deep learning models) which is not always available for insect wing images. The training time is thus also long. Moreover, the existing methods are not feasible in practice for end-user because they require the user to deeply understand the training and learning processes and the user has to prepare a good training dataset to provide as input for these systems. The framework for landmark detection in biological images proposed in this paper requires few sample data, less training time, and the processes for all necessary steps are integrated in one open source tool.

Method

In this paper, we propose a lightweight keypoint matching framework for morphometric landmark detection. Without the need of training any machine learning model, the framework aims to predict each landmark class in the input images by matching based upon the visual features the candidate keypoints with the ground-truth landmark of the same class in the annotated sample images.

Firstly, we adopt metrics that our framework aims to optimize. The prediction *accuracy* is evaluated by the *Mean Radial Square Error* (*MRSE*). The smaller the *MRSE* value gets, the higher the *accuracy* of the system is. Consider a test (or validation) set with T input images; each has M landmark classes to be predicted, then:

- The error metric for a landmark class $k \in [1,M]$ is defined as: $MRSE\_LM_k = \frac{\sum_{i=1}^{T} d_{ki}}{T}$. The d_{ki} is the *Radial Square Error* for the prediction of landmark class k in the tested image $i \in [1,T]$, i.e. $d_{ik} = \sqrt{(x'-x)^2 + (y'-y)^2}$ where x' and y' are the coordinates of the predicted landmark, x and y are the coordinates of the ground truth landmark.
- The metric for all landmark classes is: $MRSE = \frac{\sum_{k=1}^{M} MRSE\_LM_k}{M} = \frac{\sum_{k=1}^{M}\sum_{i=1}^{T} d_{ki}}{M*T}$

Regarding the *speed*, we consider how long in average our tool predicts a landmark class in an image: $speed = \frac{\sum_{k=1}^{M}\sum_{i=1}^{T} t_{ik}}{M*T}$ where t_{ik} is the duration in second the framework takes to predict the landmark class k in the input image i.

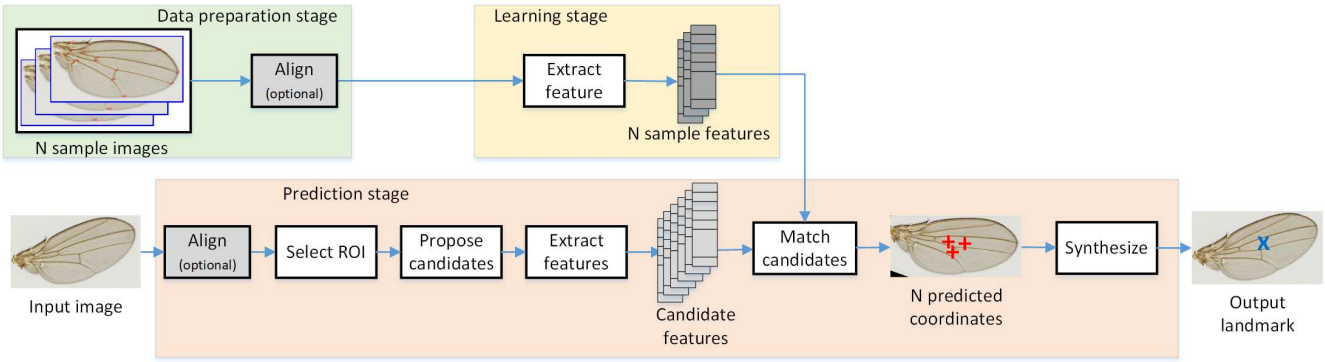

Figure 1. Workflow of our lightweight keypoint matching framework for detection of a landmark class.

Our framework consists of two main stages, namely learning and prediction as depicted in Fig.1. The goal of the learning stage is to extract some sample feature vectors for each landmark class. In the prediction stage, we pick up some candidate keypoints and choose the one having the most similar feature vector to the sample feature vectors extracted in the learning stage. Besides, data preparation stage should be used to enhance the robustness of our framework. The following subsections cover the details of these stages.

Data preparation stage

This stage consists of three procedures below:

Estimate dataset compatibility: We present a preliminary check to predict whether the framework is able to process a dataset well and which feature is the most compatible for each landmark class. In logic, our keypoint matching framework can make good predictions for landmark class k if its visual features in the input image are similar to the ones of the corresponding annotated landmarks in the sample images. We consider the similarity of a landmark class k between two images i and j as $cos(v_i, v_j) = \frac{v_i.v_j}{|v_i||v_j|}$ where v_i, v_j are the corresponding feature vectors using the chosen feature descriptor of the landmark class k in the image i and j; $v_i.v_j$ is the dot product and $|v_i|$, $|v_j|$ are the magnitudes of the two feature vectors. Having S annotated images as a subset of the working dataset, the similarity of S images on the landmark class k is $sim_k = \frac{2!(S-2)!}{S!}\sum_{i=1}^{S}\sum_{j=i+1}^{S}\frac{v_i.v_j}{|v_i||v_j|}$ Among feature descriptors, one having the biggest sim_k should be appointed for the landmark class. The overall similarity of the set S is the mean of M landmark classes $SIM = \frac{\sum_{k=1}^{M} sim_k}{M}$. In order to increase SIM, the outlier images which consist of too much noise, stains *etc*. should be excluded from the dataset. Although we cannot compare the SIM of different datasets directly, the ones with low $SIMs$ on all feature descriptors often cause difficulties for our framework.

Select N sample images: Choosing images for the sample set used in the learning stage is crucial to the performance since sample images provide our framework with patterns and demonstrations for landmark prediction stage. The sample images are chosen from the dataset such that they are good representatives of the whole dataset. Abnormal images with stains, tattered wings, abnormal noise should be excluded. Different groups (of genus, micro-population, sex, genotype, *etc*.) in the dataset should have delegates in the sample set with appropriate proportion according to their number of individuals in the whole dataset.

Inspired by the way machine learning models are trained and evaluated, we suggest a procedure to determine the size N of the sample set as follows. To asset the quality of a sample set, we employ an optimization image set then compute the $MRSE$ of the prediction on this set. The sample set initially consists of few images, often one representative for each group found in the dataset. The size N of the sample set is increased at each round with respect to the proportion of each group until the $MRSE$ on the optimization set is stable. To reduce the effects of biased selection, different image selections of a sample size are tested and the $MRSE$ is computed on all the groups.

Align: This step is optionally applied for both sample and input images. It is useful for datasets with objects (wings) located in different images at different positions, scales, and directions. The target image chosen for the alignment should have its object and landmarks being at common positions, scales, and directions with respect to the whole dataset.

In order to align an input image with a target image, we extract a set of keypoints in the input image and another set in the target image, then match pairs between keypoints of those two sets as demonstrated in Fig. 2 (a). We employ the A-KAZE method[15] for searching keypoints and for calculating the feature descriptors of keypoints since it outperforms BRISK, ORB,

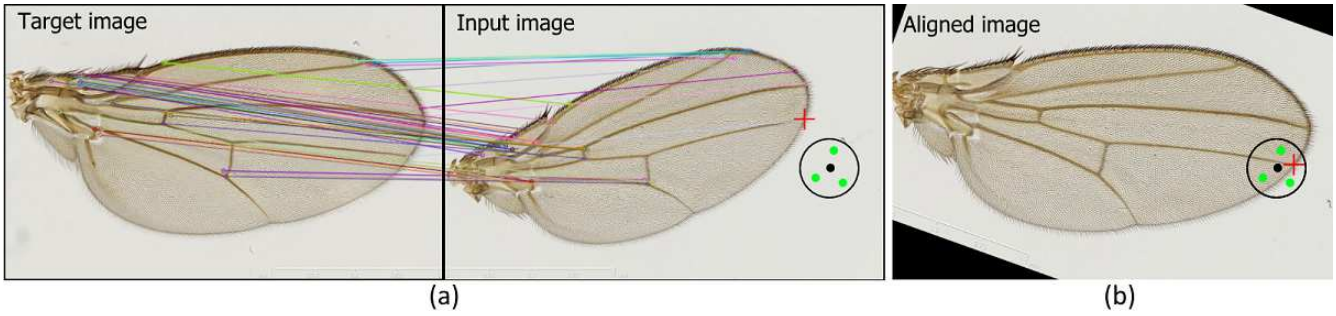

Figure 2. Aligning the input image to force the real landmark appears in the ROI. Figure (a) illustrates the pairs of matched keypoints between the target image on the left and the input image on the right in which the ground truth landmark (red cross) falls outside the ROI (black circle). Figure (b) shows the aligned image in which the ROI covers the ground truth landmark. In both (a) and (b), the black dot represents the ROI center calculated as the centroid of N annotated landmarks (green dots) of the same class with the ground truth landmark in N sample images.

SURF, and SIFT according to[15]. As A-KAZE returns binary descriptor, the Hamming distance is used to compare and match the feature vectors. If at least three matched pairs are found, we use RANSAC[16] algorithm to optimize the affine transform matrix with 4 degrees of freedom (rotation angle, scaling factor, and translation in 2 directions) between two 2D point sets. Applying the optimal affine matrix to the input image results in the aligned image as demonstrated in Fig. 2 (b).

Learning stage

The input of the learning stage is N sample images (already aligned if needed) with annotated positions of ground truth landmark classes. Then, for each sample image, the local features of the image region around each ground truth landmark are extracted by using different feature extraction methods. At this step, each annotated landmark in a sample image is represented by a feature vector. Provided with N sample images, we obtain N feature vectors for each landmark class. In our experiments, five feature extraction methods are used: SURF[5], Haar-like[4], LBP[17], and two schemes of HOG[6]. We solely process the native resolution for all of these extraction methods except Haar-like due to its essence that requires multi-scale options.

Prediction stage

The objective of landmark prediction stage is to estimate the position of different landmark classes for a new input image. We may need to align the input image so that the object in the input image has the same direction, scale, and position with the objects in the sample images used in the learning stage. The prediction for an input image is divided into five steps:

- *Select Region of Interest (ROI)*: to estimate the position of each landmark class, we define in the input image a Region of Interest (ROI) to which the landmark red should belong. The ROI is defined as a circle whose center is the arithmetic mean of N annotated landmarks of the same class in N sample images, and a radius defined by the user. In Fig. 2 (b), the green dots indicate the annotated positions of a landmark class in 3 sample images; the black dot represents the centroid calculated from the position of all the green ones, and the circle presents the ROI of the corresponding landmark class in the input image.

- *Propose candidate points*: for each landmark class, we pick up some points in the ROI as the candidates for the landmark of this class. Three options are provided to select the candidate points: (1) *Random sampling* in which C points are randomly selected; (2) *Gaussian sampling* picks C points according to a 2D-Gaussian distribution; and (3) *Keypoint-detection* employs the A-KAZE key-point extraction method[15] to choose candidate points. For *Random sampling* and *Gaussian sampling*, the number of selected points C is defined by the user, whereas the A-KAZE mehod defines by itself the number of keypoints depending on the image.

- *Extract features*: the local features representing the visual information of the image region around each candidate point in the ROI region are extracted by using the same feature extraction methods used in the learning stage. Each candidate point is then described by a feature vector.

- *Match candidates*: among all the candidate points for each landmark class, we match the one with the most similar features to the features extracted in the learning stage of the ground truth landmarks of the corresponding class. For histogram-based feature descriptors (LBP and HOG), the matches are performed by finding the maximal intersection between feature vectors. For SURF and Haar-like feature descriptors, the matches are found based on the minimal

Euclidean distance between feature vectors. For each landmark class, N sample feature vectors will result in N matched candidates, some of which may be coincident.

- *Synthesize*: This step aims to decide the final position of each landmark class from N matched candidates found in the previous step. First, the outliers among matched candidates are eliminated using Isolation Forest algorithm[18], then the predicted position of a landmark class is chosen as the arithmetic mean of the positions of all remaining matched candidates of this class.

Experiments

The objectives of our experiments are: (1) to compare the accuracy and the speed of our framework with the state of the art method on the *Droso-small* dataset, (2) to study the influence of the sample size and different landmark classes on the accuracy, and (3) to test the compatibility of our framework with various datasets.

Datasets

Short name	Species	Number of landmark classes	Dataset size	Resolution
Droso-small[9]	Drosophila	15	138	1400x900
Droso-big[19]	Drosophila	15	1134	1360x1024
Fly[20]	Tsetse fly	10	15	2031x1180
Bactro[21]	Bactrocera tau	12	53	2048x1563
Diacha[22]	Diachasmimorpha longicaudata	10	92	1000x750

Table 1. Summary of the experimental datasets.

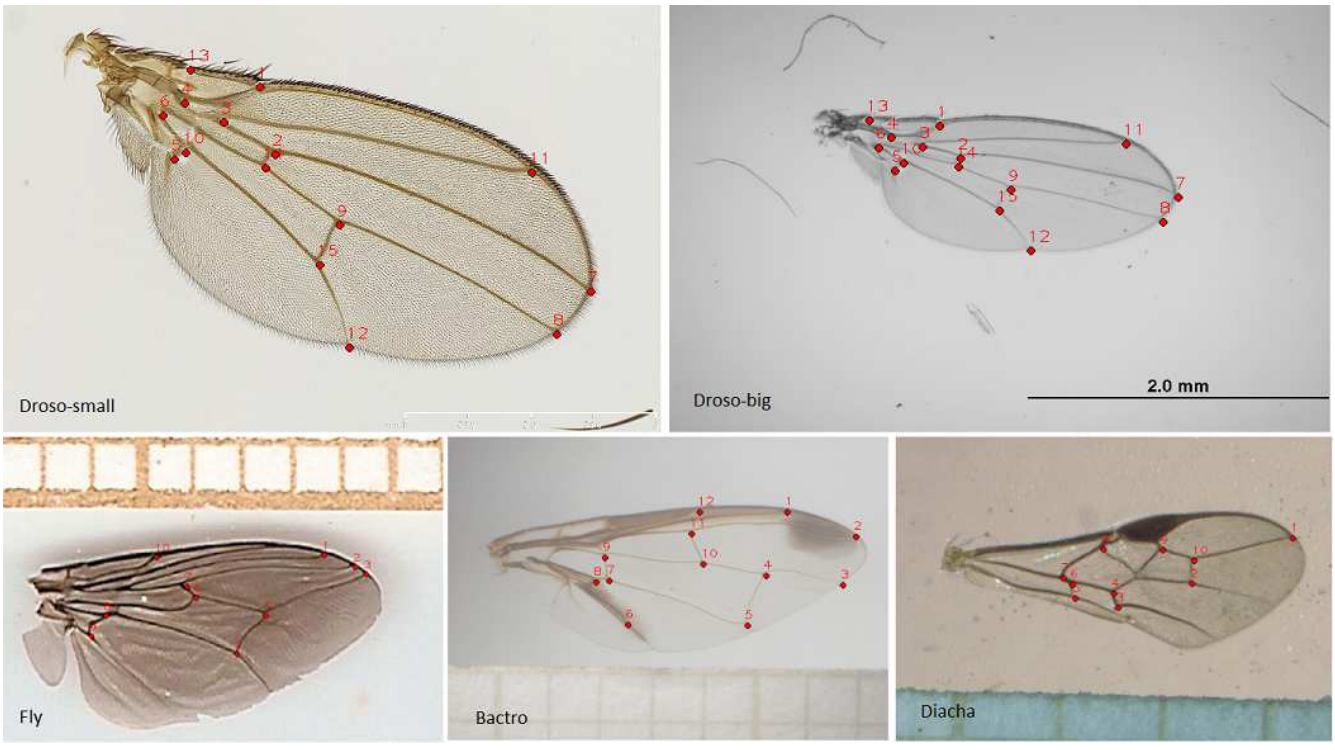

Figure 3. Sample images with the corresponding landmark classes of 5 tested datasets. From left to right side, upper to lower: Droso-small, Droso-big, Fly, Bactro, Diacha.

We evaluated the prediction accuracy of our framework with five public insect-wing datasets summarized in Table 1 in which the "Short name" is entitled for the sake of mentioning afterward. Fig. 3 provides some examples with annotated landmark classes of these datasets. Their details are discussed as follows:

- *Droso-small* is a Drosophila wings dataset that was employed to test the method proposed by Valdaele *et al.*[9]. For comparison with this state of the art method, we conducted thorough benchmarks on both accuracy and speed.
- *Droso-big* is another dataset of Drosophila melanogaster wings introduced in[19]. Among insect species, drosophila can be encountered in numerous public datasets as it has drawn significant attention in the studies of automatic wing shape analysis[19,23]. This dataset was published with the target to be the general resource for the community. Accordingly, it contains a more considerable amount of wing images (1134 images for each side) than the other four datasets. Its genotype diversity leads to the variation of landmark shape and location that may challenge our framework. To prepare for future comparison tasks between the two datasets on drosophila, we annotated the landmark classes of *Droso-big* as in *Droso-small*. Since the left and right wings of organisms are almost symmetrical, we made annotation for the right side solely to save our effort.
- *Fly* is a small dataset[20] of Glossina palpalis in the southern Ivory Coast. The small distance between some landmark classes pose a risk of confusion in landmark identity.
- *Bactro* dataset[21] comprises 53 images of both male and female bactrocera tau from Kanchanaburi and Nan region in Thailand. This diversity of geographic origins and sexes may result in variations in morphology and locations of landmarks. Moreover, some of landmark classes fall into the opaque areas of wing leading to low contrast levels which is probably a difficulty for the detection task.
- *Diacha* is a dataset of Diachasmimorpha longicaudata[22], a genus of endoparasitoid. The obstacles for landmark detection methods are the existences of stains, water drops, and wet areas on wings.

Parameter choice

This section provides a guideline for users to choose appropriate parameters for our framework.

Candidate point option: With images having high contrast level around the vein region, the keypoint-based candidate mode is able to propose few but quality candidates. According to our experiments, it works well with all of our five experiment datasets. However, with images in which keypoints locate in a gradient or low contrast regions, the *Random* or *Gaussian sampling* is suggested although they increase the number of candidates, hence reducing the processing speed.

Radius of ROI: As this parameter determines the size of the ROI for candidate searching, it should be carefully set such that the ROI is compact enough to limit many too far and redundant candidates while it is large enough to cover the position of real landmarks. We suggest users making a coarse inspection on positions of different landmark classes on sample images, then set the *Radius of ROI:* to be the furthest distance from the ROI center.

Feature descriptor configurations: Each feature descriptor has its own parameters that need to be configured properly. The meaning and suggestion for settings of Haar-like and SURF are discussed thoroughly in[9] that ones can follow. We use these optimized values as reported in[9] for the dataset *Droso-small*, and adjust the *Window size* and the number of resolutions of Haar-like feature for other datasets according to image resolution and landmark distribution. For LBP descriptor, our tests show that 8 neighbors around the extracted position and 3-pixel radius are good settings for most of the cases. The patch to compute the histogram from the LBP matrix should be selected so that it encloses the texture pattern of landmarks. Similarly, the *Window size* of R-HOG or *Radius size* of C-HOG is set to represent a plausible region around the interest position. The *Number of bins* for gradient direction is often set to 9 or 18 depending on the angle quantization level of the gradient vector we expect. With *Droso-small*, we learn from out tests that the following values are good for our method: *Patch size* of LBP = 60, *Windows size* of R-HOG = 60, *Radius size* of C-HOG = 30, *Number of bins* for both HOG schemes = 18.

Results and discussion

Our framework was implemented in Python with two libraries: OpenCV[24], a well-known open-source library for image processing tasks, and PyQt5 for cross-platform GUI. All the experiments are performed on a laptop equipped with an Intel Core i5-1035G7 processor and 8GB of RAM.

Comparison with existing method

After tuning parameters and choosing optimized sample size ($N = 15$) for *Droso-small*, we report in Table 2 the accuracy and the speed of our framework along with those of the method by Vandaele *et al.*[9]. The *MRSEs* of Vandaele method are roughly extracted from the chart in the original paper[9]. As we have not implemented the multi-scale option for feature descriptors except Haar-like, the comparisons are made for the native resolution. Since we had difficulty accessing the source code of Vandaele *et al.*, as well as our framework is independent of and can not run on the Cytomine platform as the method of Vandaele *et al.*[9], we have re-implemented the method of Vandaele *et al.* to homogenize the programing language, libraries, and the bench-marking environment for the speed comparison.

		Vandaele et al[9].					Our framework				
Feature extractor		*SURF*	*Haar-like*	*RAW*	*SUB*	*GSUB*	*SURF*	*Haar-like*	*LBP*	*R-HOG*	*C-HOG*
MRSE (pixels)		10.5	5	13	10.2	10.3	14	14.8	11.4	10.6	9.4
Speed (seconds)	*Learning*	66235	12984	318	1348	1523	0.06	0.06	0.14	0.33	0.69
	Prediction	97.35	21.23	0.76	3.19	3.52	0.16	0.14	0.19	0.62	1.41

Table 2. Comparison of accuracy and speed between our framework and the state of the art method[9].

Table 2 reveals that overall, the *MRSEs* made by our framework are comparable with those of Vandaele *et al.* method[9]. Our method works well with the three histogram-based feature descriptors (LBP and two HOG schemes), and the best *MRSE* which is only 9.4 pixel with C-HOG is good in practice with reference to human work. The method of Vandaele *et al.* merely shows the advancement in Haar-like feature which is in fact a multi-scale descriptor. However, this can be considered as a trade-off between *accuracy* and *speed* when our framework runs much faster, especially in the learning stage. This difference in speed comes from several reasons. First, our method requires fewer sample data in the learning stage than the method in[9]. The learning stage of our framework simply extract features of landmarks on sample images, whereas Vandaele *et al.*[9] perform feature extraction for a large number of points (e.g. 5000) around each landmark then train an Extremely Randomized Tree classifier to decide if a candidate point is a landmark or not. Second, the proposed method needs only a few sample images (3-15) to learn while[9] and other methods require hundreds of images. Third, the proposed method requires fewer candidate points in the prediction stage. Finally, in the prediction stage, the proposed method only compares the similarity of feature vectors while other method uses a classifier to classify if a candidate point is positive or not.

Influence of sample size and landmark classes

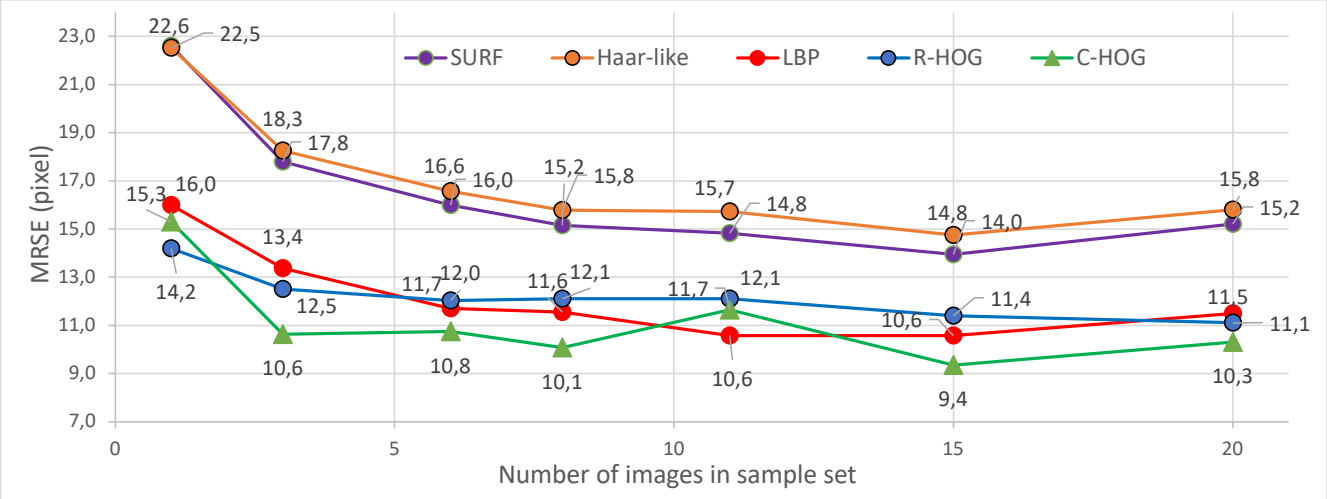

Figure 4. Influences of sample sizes and feature descriptors in *Droso-small* dataset.

The variation of *MRSE* on *Droso-small* when the number of sample images increases is exhibited in Fig. 4. It can be seen from the chart that the initial attempts with a single sample image do not give good MRSEs. When we add more sample images, the *MRSEs* reduce quickly then become stable because the deviations of predictions determined by different samples tend to neutralize them-self. The *MRSEs* converge quickly, meaning our framework can reach robustness with only a few sample images. In terms of visual features, C-HOG still often gives the lowest *MRSE*, then LBP and the two HOG variations show better performance than SURF and Haar-like in all sample set sizes.

We analyze further the prediction accuracy for each specific landmark class by plotting the *MRSE_LM* in Fig.5 with a sample size big enough for the stabilization of MRSEs, herein $N = 15$ for *Droso-small* dataset. C-HOG still provides the best accuracy for landmark classes that have simple texture patterns or clear vein texture. This suggests that C-HOG is the most appropriate descriptor for seeking landmarks in regions where veins radiate from the landmark positions. The reason might come from the fact that when we compute C-HOG descriptor for the landmark, the vein line mainly appears in only one or two blocks; thus the vein is locally characterized by the histograms corresponding to these blocks. The minor change in the angle of the vein does not affect other blocks. On the other hand, the vein line often leaves its "footprint" in some blocks of C-HOG feature vector of a non-landmark point. Therefore, C-HOG features might distinguish well the landmark and non-landmark

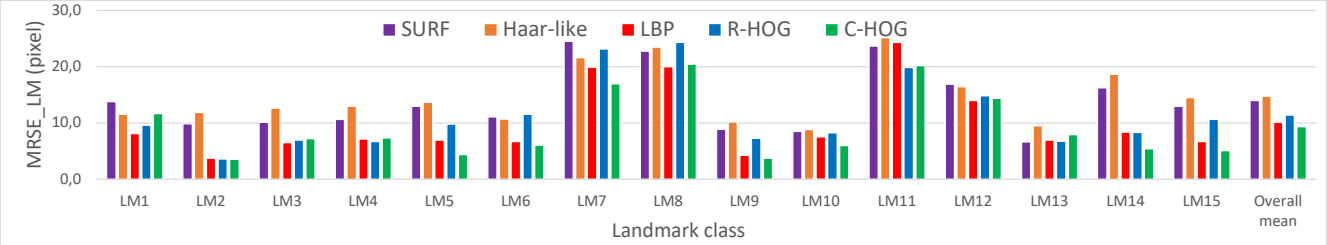

Figure 5. Comparison of accuracy of different feature descriptors on different landmark classes in *Droso-small* dataset with sample size $N = 15$.

points. In contrast, with R-HOG and LBP, a vein line passes through many regular-grid blocks, making them oversensitive for minor changes in the shape and angle of vein lines. Nevertheless, at hairy and complex texture regions (landmark classes 1 and 13), LBP outperforms the others. To minimize the overall *MRSE*, each landmark class should be processed by its most appropriate feature descriptor.

According to our inspection from the *Droso-small* dataset, the positions of landmark classes at the end of wings (7, 8, and 11) are distributed arbitrarily in wide zones. Consequently, the real landmark positions fall outside the ROI sometimes despite the effort of the Align step, leading to high *MRSEs* for corresonding landmark classes. Increasing the size of ROIs for only these landmark classes may reduce this risk; however, we have to take the precaution to avoid the confusion between landmark classes when the ROIs may overlap, and these landmark classes look similar in terms of vein-line patterns (as the landmarks 7, 8, 11 of Drosophilas in Fig. 3). Moreover, a bigger size of ROI will reduce the speed since a larger number of candidate landmarks are needed to extract features.

Compatibility with datasets

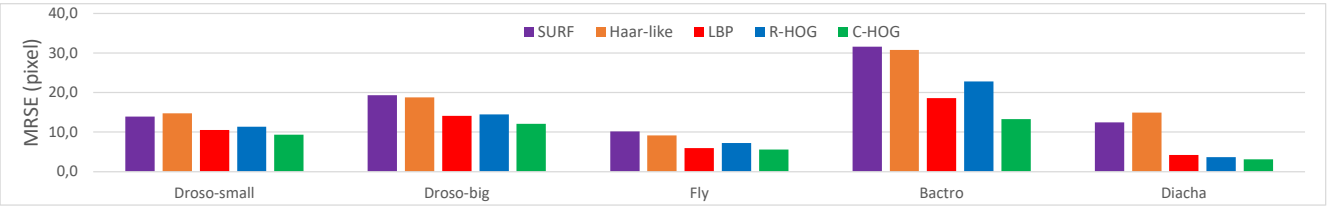

Figure 6. Effects of different feature descriptors in 5 datasets.

After tuning parameters and choosing an optimized sample set for each dataset, the *MRSEs* of our framework on five datasets using different feature descriptors are reported in Fig. 6. The reported *MRSEs* confirm that our framework can be applied to these 5 datasets. With datasets in which all landmark classes positioned at the junction points of clear veins such as in *Fly* and *Diacha* (see Fig. 3), our framework using LBP and two HOG schemes often gives nearly perfect landmark predictions as proved by small values of MRSEs (3 to 7 pixels). Nevertheless, all descriptors encounter difficulties with *Bactro* dataset as some of its landmark classes (1, 2, 6, 8, 9, 12) fall into gradient zones of opaque wings. In terms of feature descriptors, C-HOG always gives the best *MRSE*, followed closely by R-HOG and LBP; meanwhile SURF and Haar-like often lead to significantly higher deviations.

Our framework certainly cannot attain the "one size fits all" ideal dream. It cannot work well with datasets having low *similarity* in terms of visual features at specific landmark classes in different images due to the existence of unexpected objects such as noises, stains as well the inconsistencies of sharpness and contrast. These factors drastically ruin the pattern of local visual features of landmark classes. Fig. 7 shows one of our in-house datasets that our framework fails at both image alignment and landmark prediction.

Software

Our algorithm was implemented and integrated into a utility named "iMorph" whose GUI is shown in Fig. 8. The source code and runnable files of iMorph are published to GitHub at https://github.com/ha-usth/WingLanmarkPredictor. The landmark positions of each image (both sample and prediction data) are stored in a simple text file having the same name as the image file, yet with ".txt" extension. Each row of coordinate text files holds the 2D coordinate of a landmark. Users can follow the below steps to process a dataset.

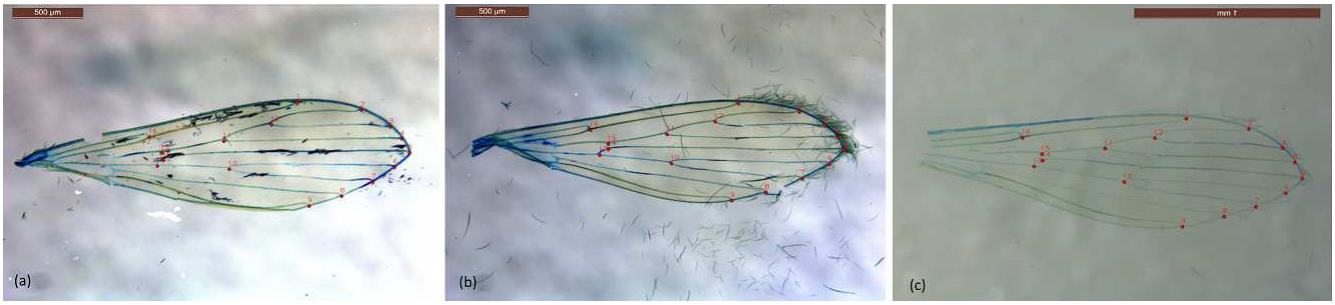

Figure 7. Our framework cannot work with this sand fly dataset due to the heterogeneity: stains appear arbitrarily in (a), the end of wing is too hairy and there are cracks in (b), the image is too blur in (c).

Data preparation stage:

1. Load the sample data: locate the folder including sample images, each should go along with the corresponding text file containing the positions of annotated landmark classes.
   (a) Set the border size of key-point matching region used for image alignment
   (b) Choose an image in the sample set as the alignment target image and select pre-process button to align all other images in the sample set with the target image.

Learning stage:

2. Select feature type: the utility supports 5 feature descriptors: SURF, Haar-like, LBP, R-HOG, and C-HOG
3. Select the candidate landmarks proposition method from 3 options: Key-point extraction, Random sampling, or Gaussian sampling.
4. Select the sizes of the ROI.
5. Select the number of random points if the “Random” or "Gaussian" method is used in step 3.
6. Train to extract sample feature vectors from sample images

Prediction stage:

7. Load the predict data: locate the folder including images to predict landmarks.
8. Predict: the tool predicts the location of landmark classes for each image in the prediction folder and saves the predicted landmark positions into the corresponding coordinate text files.

Once an image name in Sample data or Prediction data panels is selected, the panel on the right will show the corresponding image. If there exists the corresponding coordinate text file, the landmarks are plotted on the image to let the user check the positions of landmark classes. In case the user does not satisfy, they can tune landmark classes by dragging and dropping them to the desired positions, then click the "Save" button to overwrite the coordinate text file.

Conclusion

In conclusion, the proposed approach has the advantage of being lightweight, requiring a small annotated dataset to build a prediction model. It is faster and accurate compared to previous methods. We also provide user-friendly interfaces to easily setup learning, prediction, and error correction stages. The framework is provided as an open-source utility.

Our approach to landmark detection is based on keypoint matching, therefore largely depending on the similarity of the visual features extracted from the sample set and those from the prediction set. Although it is possible to align images in the data preparation step, the consistency regarding how images are produced still determines the overall performance. The A-KAZE method for alignment fails to find the ROI once the images are considerably different. It is, thus, recommended that the users investigate the consistency of their datasets by the proposed similarity index and visually to put aside possibly problematic ones. Furthermore, typical noises on insect wing images such as stain, hair, non-intact, or blur regions are not

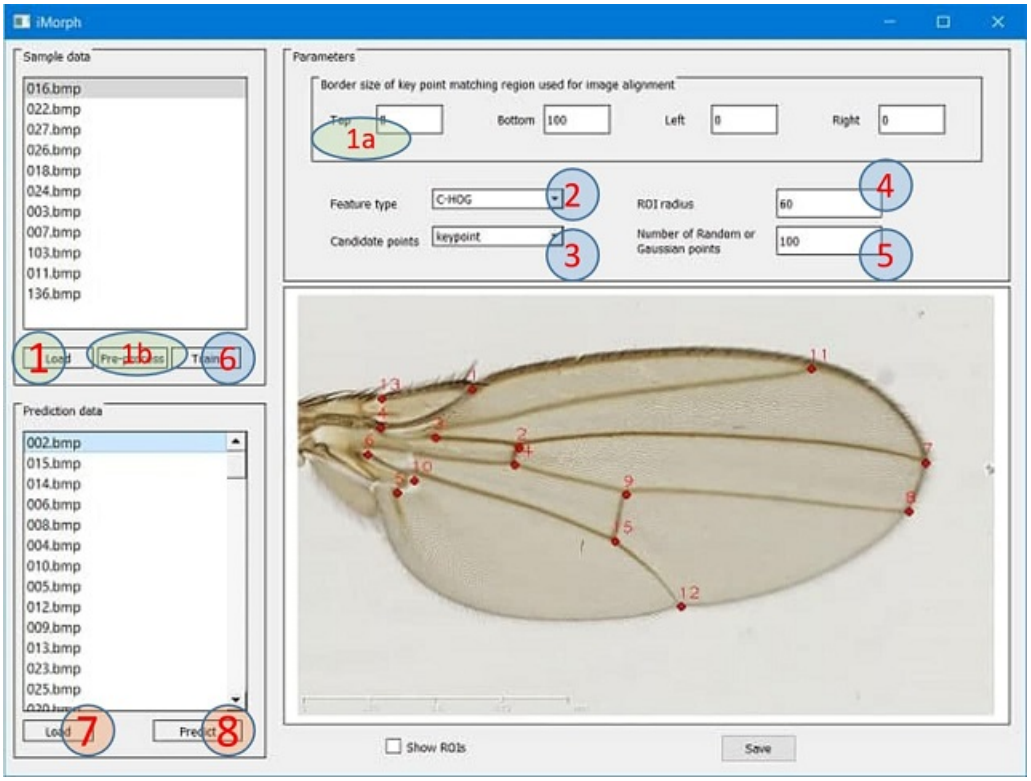

Figure 8. Screenshot of iMorph and guidance steps to use.

treated in this version of iMorph. Fig. 7 shows some images with noise for which our current framework fails. We anticipate developing an image preprocessing module in the data preparation stage to reduce those noise. As shown in our experiments with different visual features for landmark classes, the prediction of each class could be improved if its appropriate visual feature is chosen, guided by the local texture pattern or a held-out optimization dataset. We plan to provide the feature options for landmark class in the next version too.

References

1. Rohlf, F. J. tpsdig, digitize landmarks and outlines, version 2.05. *Dep. Ecol. Evol. State Univ. New York at Stony Brook* (2005).
2. Dujardin, S. & Dujardin, J.-P. Geometric morphometrics in the cloud. *Infect. Genet. Evol.* **70**, 189–196, DOI: https://doi.org/10.1016/j.meegid.2019.02.018 (2019).
3. Olsen, A. & Haber, A. Stereomorph: Stereo camera calibration and reconstruction (2021). R package version 1.6.4.
4. Viola, P. & Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, I–I, DOI: 10.1109/CVPR.2001.990517 (2001).
5. Bay, H., Tuytelaars, T. & Van Gool, L. Surf: Speeded up robust features. In Leonardis, A., Bischof, H. & Pinz, A. (eds.) *Computer Vision – ECCV 2006*, 404–417 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006).
6. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 886–893 vol. 1, DOI: 10.1109/CVPR.2005.177 (2005).
7. Loh, S., Ogawa, Y., Kawana, S., Tamura, K. & Hwee-Kuan, L. Semi-automated quantitative drosophila wings measurements. *BMC Bioinforma.* **18**, DOI: 10.1186/s12859-017-1720-y (2017).
8. Palaniswamy, S., Thacker, N. & Klingenberg, C. Automatic identification of landmarks in digital images. *IET Comput. Vis.* **4**, 247–260, DOI: 10.1049/iet-cvi.2009.0014 (2010).
9. Vandaele, R. *et al.* Landmark detection in 2D bioimages for geometric morphometrics: a multi-resolution tree-based approach. *Sci. Reports* **8**, 538, DOI: 10.1038/s41598-017-18993-5 (2018).

10. Lindner, C. & Cootes, T. Fully automatic cephalometric evaluation using random forest regression-voting. In *host publication* (2015). IEEE International Symposium on Biomedical Imaging (ISBI) 2015 – Grand Challenges in Dental X-ray Image Analysis – Automated Detection and Analysis for Diagnosis in Cephalometric X-ray Image ; Conference date: 01-01-1824.

11. Donner, R., Menze, B. H., Bischof, H. & Langs, G. Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization. *Med. Image Analysis* **17**, 1304–1314, DOI: https://doi.org/10.1016/j.media.2013.02.004 (2013).

12. Sun, Y., Wang, X. & Tang, X. Deep convolutional network cascade for facial point detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3476–3483, DOI: 10.1109/CVPR.2013.446 (2013).

13. Zhang, Z., Luo, P., Loy, C. C. & Tang, X. Facial landmark detection by deep multi-task learning. In Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T. (eds.) *Computer Vision – ECCV 2014*, 94–108 (Springer International Publishing, Cham, 2014).

14. Le, V.-L., Beurton-Aimar, M., Zemmari, A., Marie, A. & Parisey, N. Automated landmarking for insects morphometric analysis using deep neural networks. *Ecol. Informatics* **60**, 101175, DOI: https://doi.org/10.1016/j.ecoinf.2020.101175 (2020).

15. Alcantarilla, P. F., Nuevo, J. & Bartoli, A. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In Burghardt, T., Damen, D., Mayol-Cuevas, W. W. & Mirmehdi, M. (eds.) *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, DOI: 10.5244/C.27.13 (BMVA Press, 2013).

16. Fischler, M. A. & Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381–395, DOI: 10.1145/358669.358692 (1981).

17. Ojala, T., Pietikainen, M. & Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis Mach. Intell.* **24**, 971–987, DOI: 10.1109/TPAMI.2002.1017623 (2002).

18. Liu, F. T., Ting, K. M. & Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, 413–422, DOI: 10.1109/ICDM.2008.17 (2008).

19. Sonnenschein, A., VanderZee, D., Pitchers, W., Chari, S. & Dworkin, I. An image database of drosophila melanogaster wings for phenomic and biometric analysis. *GigaScience* **4** (2015).

20. Kaba, D. *et al.* Phenetic and genetic structure of tsetse fly populations (glossina palpalis palpalis) in southern ivory coast. *Parasites & Vectors* **153**, DOI: https://doi.org/10.1186/1756-3305-5-153 (2012).

21. Kitthawee, S. & Dujardin, J.-P. The geometric approach to explore the bactrocera tau complex (diptera: Tephritidae) in thailand. *Zoology* **113 4**, 243–9 (2010).

22. Kitthawee, S. & Dujardin, J.-P. The diachasmimorpha longicaudata complex: Reproductive isolation and geometric patterns of the wing. *Biol. Control.* **51**, 191–197, DOI: https://doi.org/10.1016/j.biocontrol.2009.06.011 (2009).

23. Houle, D., Mezey, J., Galpern, P. & Carter, A. Automated measurement of drosophila wings. *BMC evolutionary biology* **3**, 25, DOI: 10.1186/1471-2148-3-25 (2004).

24. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* (2000).

Acknowledgements

This work has been done with the support from the project coded "VAST01.01/19-20", titled "Research and development of methods for automatic morphometric landmark detection on insect wing images".

Author contributions statement

Prudhomme and Bañuls suggested the study, identified the requirements and provided the sand fly dataset. Ho formulated the problem. Lai surveyed the related works. Nguyen raised the main idea while Le, Lai, and Tran gave discussions to detail the framework. Le implemented the framework and iMorph utility. Nguyen completed the framework and the utility, collected datasets, and conducted the experiments. All authors wrote and reviewed the manuscript.