

MAELab: a framework to automatize landmark estimation

LE Van Linh	BEURTON-AIMAR Marie	KRAHENBUHL Adrien	PARISEY Nicolas
LaBRI-CNRS 5800 ITDLU, Dalat Univ-V linh@dlu.edu.vn/ van-linh.le@labri.fr	LaBRI-CNRS 5800 Bordeaux University 33400 Talence-F beurton@labri.fr	LaBRI-CNRS 5800 Bordeaux University 33400 Talence-F adrien.krahenbuhl@labri.fr	IGEPP INRA 1349 35653 Le Rheu-F nparisey@rennes.inra.fr

Abstract

Phenotype of beetle species are characterized by informations like age, sex, morphological criteria or environmental parameters. Biologists are used to proceed to manual measurements in case of analysis at macro level as for example tissues or animal members. They can directly measure the geometrical characteristic of elements on the body of the animal: length, width, diameter, angles, ... Another way to obtain morphological measures is to take pictures of the members and to apply image processing algorithms. In order to evaluate a population of beetles from Brittany lands, a collection of 293 beetles has been built. For each beetle, biologists took images of the left and right mandibles. A set of landmarks has been manually located: 16 for the left mandibles and 18 for the right ones. They constitute the ground truth to evaluate the automatically estimated landmarks. In a previous work [4], we have shown that the Probabilistic Hough Transform (PHT) is efficient to compute the centroid position of the mandible. If the goal is to consider more precisely the position or the geometry of the landmark areas, the results are not accurate enough to consider the replacement of manual landmarks by the automatically estimated landmarks. In this paper, we propose a method to improve the previous segmentation algorithm using the Canny algorithm. Moreover, we propose a registration procedure based on an Iterative Principal Component Analysis. Finally, the landmarks are located using a SIFT descriptor computed in the landmarks area of the model image. The complete workflow is implemented in MAELab, freely available as library on a GitHub website.

Keywords

Morphology, image registration, detector/descriptor, SIFT, beetle, mandible

1 INTRODUCTION

In biology, the morphological analysis is widely used to detect the inter-organisms variations. It allows to determine the evolution of an organism family and to classify these organisms. Especially in agriculture, morphology is one of the best ways to learn about the evolution of the beetle on crops (see Fig. 1).

The morphology methods may be divided into groups considering the features used such as shape, structure, color, pattern or size of the object. In order to study the potential links between these characteristics and agricultural ecosystems, a set of beetles has been analyzed to collect a lot of informations. Their sex, place where they were found and agricultural practices in each field were recorded. For each beetle, the morphometric landmarks has been manually identified on the mandible im-

ages. The morphometric landmarks are precise points defined by biologists. Landmarks are used in many biological studies and are currently included into the classification procedures.



Figure 1: Sample of pictures of beetle mandibles took by biologists.

In this paper, we focus on a chain of algorithms to automatically identify landmarks in 2D images. The method mainly includes three steps: firstly, a segmentation of the mandible shape, then an iterative principal component analysis to register a query image on a model image, and finally a landmark estimation using the SIFT descriptor.

The section 2 presents the complete workflow then the section 3 details the experiments and expose the results.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2 METHOD

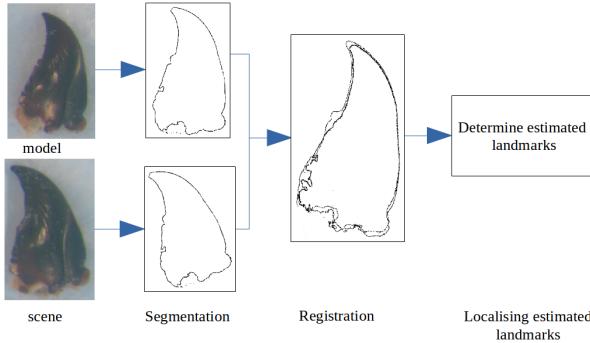


Figure 2: Overview of the proposed method

The addressed problem is the automatic detection of landmarks on mandible photographies to replace the manual operation by an operator. We propose hereafter a workflow including (1) the segmentation of each image, (2) a registration step on a model image then (3) the detection of landmark positions. The Fig. 2 shows the summarize the workflow. It is worth to note that the model image is chosen randomly from the set of images without broken mandibles. Moreover, all pictures of mandibles have been taken in the same conditions with the same camera and at the same resolution.

2.1 Image segmentation

The segmentation step is the first major task of a large number of image processing chain. We have chosen a contour-based algorithm, the Canny algorithm [2], allowing to determine the contour belonging to the shape of the mandible. To use this method, two threshold values have to be set. As it is mentioned in [3], determine the right thresholds could be difficult. The mandatory *threshold value* used by Canny algorithm is determined by analyzing the image histogram (see [4] for detail). Most often authors define these thresholds as a lower and an upper one with a usual ratio of $T_{lower} = (1/2) \times T_{upper}$. In order to consider a larger range of values, we prefer set $T_{lower} = 1/3 \times T_{upper}$. We can note that to optimize the computing time, the direction of the gradient of each pixel belonging to the mandible contour is computed during the Canny algorithm and will be used later. To obtain the segmentation of the mandible, the contours obtained with Canny are discriminated to only keep the mandible contours. As shown in Fig. 3, the Canny algorithm generates some contours which do not belong to the mandible shape. With a simple algorithm, the contour image is parsed to suppress the edges inside the bigger contour.

2.2 Image registration

As noted in introduction, all images have been captured with the same scale. But the size of mandible can vary from a beetle to another one and/or the orientation and

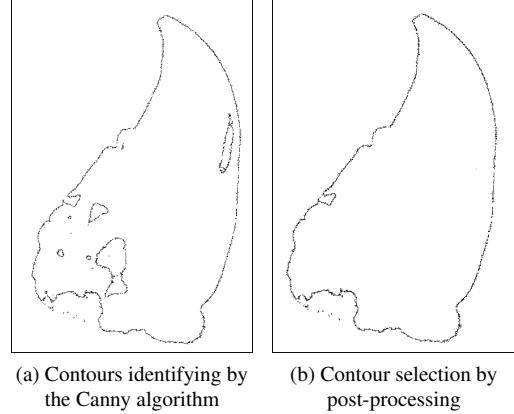


Figure 3: Step of the mandible contour detection.

position can be different from a photography to another one. The next step concerns the registration of an image on a reference image.

We have chosen to use a Principal Component Analysis (PCA) [6], [7] to determine the rotation and translation parameter values between the two images. As input values, we use the lists of contour points defined at the segmentation step. Firstly, the centroid point and principal axis of each image are defined. The centroid point coordinates are computed like the mean coordinate of all contour points. The principal axis is a line connecting the centroid point to a contour point, determined as detailed in algorithm 1.

Algorithm 1: The algorithm for finding the principal axis of a list of contour points

Input : Centroid point c , list of contour points l

Output: The principal axis a

```

1 for all points  $p_i$  in  $l$  do
2   for all points  $p_j$  in  $l$  do
3     if  $p_i \neq p_j$  then
4       Compute the perpendicular distance  $d_{ij}$ 
        between line  $(c, p_i)$  and  $p_j$ .
5     end
6   end
7   Compute the average distance ( $d_{mean}$ ) of all
      distances  $d_{ij}$ ;
8   if  $d_{mean}$  is minimal then
9     |  $p_{min} = p_i$ ;
10  end
11 end
12 The principal axis is:  $a = (c, p_{min})$ .
  
```

The translation is calculated like the distance between the centroid points of the scene and the model. The rotation angle is the angle between the principal axes of these two images. Translation then rotation are applied to register the scene one the model. However, in some cases, the translation and rotation between two

images are not enough precise because the result of the segmentation could contain some noise. To improve registration, we have enhanced the PCA by iterating until stabilization (PCAI). We have considered some specificity of our images and observed that the tip of mandible is less noisy than its base on images. So, we sort the contour points according to their y-value to build the subset containing the half upper part of contour points. PCA is again completed for this subset to refine the rotation and translation values. This operation is iterated until the new computing angle is lower than 1.5 degrees (see Fig. 4).

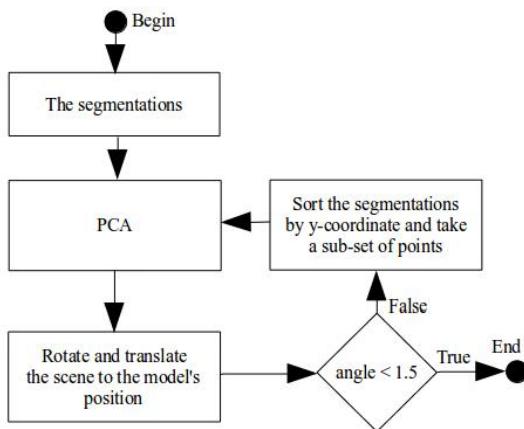


Figure 4: The workflow of the PCAI, allowing to refine the translation and rotation values.

The Fig. 5 shows an example of the successive results obtained with PCAI. The red contours belong to the model, the black ones are the scene contours after one iteration, and the blue ones are the resulting contours.

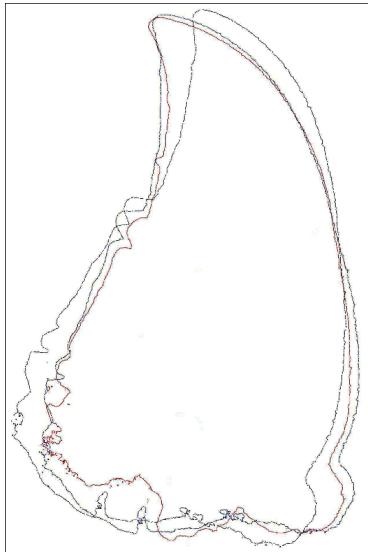


Figure 5: Iterations of the registration step between the model contour (in red) and the contours of the scene image.

2.3 Landmark detection using SIFT

The last task of the workflow consists in estimating the landmarks of the scene from the manual ones of the model. We use the SIFT method [5] but we do not consider all points of the image as usually, only the area around each landmark of the model. Firstly, the region (called patch) around each landmark of the model is computed then extracted in the scene image at the same position. Then, the SIFT descriptor is computed: the orientation and gradient magnitude are calculated for each pixel by using the gradient values computed during the Canny step by applying the following equations (1):

$$m(x,y) = \sqrt{v_x^2 + v_y^2} \quad (1)$$

$$\theta(x,y) = \tan^{-1}(v_y/v_x)$$

Where:

- $v_x = I(x+1,y) - I(x-1,y)$
- $v_y = I(x,y+1) - I(x,y-1)$
- $I(x,y)$ is the gray value at position (x,y) in the image,
- $m(x,y)$ is the gradient magnitude of the pixel at position (x,y) ,
- $\theta(x,y)$ is the orientation of the pixel at position (x,y) .

The SIFT descriptor for each patch is an histogram containing the sum of pixel gradients for each considered direction. As usually, eight directions are taking into account ($0^\circ - 45^\circ, 46^\circ - 90^\circ, 91^\circ - 135^\circ, 136^\circ - 180^\circ, 181^\circ - 225^\circ, 226^\circ - 270^\circ, 271^\circ - 315^\circ, 316^\circ - 360^\circ$). Finally, the feature vector is normalized to reduce the effects of illumination changes.

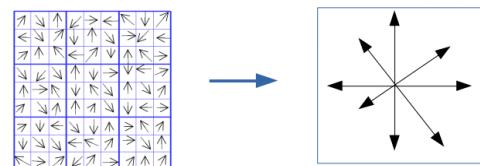


Figure 6: Calculus of the SIFT descriptor for a patch. The arrow length corresponds to the gradient value in the right figure.

The Fig. 6 shows a patch sample of 9×9 pixels centered in each landmark on the model. The size of 9×9 has been retained after several tests. Patch sizes 18×18 , 36×36 and 54×54 have been also computed but gave unsatisfactory results. From the patch histogram, we obtain the global gradient value for each direction.

The comparison between two SIFT descriptors is done by using the L_2 -distance with the following equation (2):

$$L(D1, D2) = \sum_{i=0}^n \sqrt{(D1_i - D2_i)^2} \quad (2)$$

Where:

- n is the number of directions
- D_1 and D_2 are two descriptors of size n ,
- D_{1i} and D_{2i} are the i^{th} descriptor values.

The Fig. 7 illustrates how we apply SIFT into our workflow. To detect the scene landmarks, the patches P_m of the model and P_s of the scene are created with the size of P_m smaller than the size of P_s . After experiments, we have kept 36 pixels as the size of P_s . For each pixel in the patch P_s , a sub-patch P'_s is extracted with the same size than P_m . When the P'_s have a part outside P_s , the outside pixels are also considered. Then, the distance $L(P_m, P'_s)$ is computed using equation (2). The position of the estimated landmark corresponds to the position of the sub-patch P'_s with the smallest distance L to P_m . Finally, the position of the estimated landmarks are set in the original location of the original scene image by applying the reverse operation of rotation and translation.

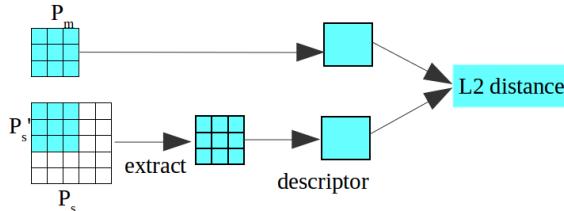


Figure 7: Steps of descriptors comparisons between the patch P_m of the model image and the patches P'_s of the scene image.

3 EXPERIMENTS AND RESULT

The complete method is implemented in MAELab¹. The beetles has been analyzed after a split in right and left mandible sets. After verifying the quality of the image, it remains 290 usable images of right mandibles and 286 images of left mandibles. The removed images include the images without mandible or with broken mandibles. In all valid images, a set of manual landmarks is indicated by biologists: 18 for right mandibles, 16 for left mandibles.

We have run the full workflow on all the usable images. The results have shown differences in algorithm accuracy: estimated landmarks are well positioned on some scene images but not satisfying on others. As we mentioned before, mandibles images can exhibit different sizes because beetles have also different sizes of mandible. We detected that our method is sensible to this parameter. To improve the results, we have inserted a pre-processing step before the computing of the SIFT descriptors to estimate the scale between a scene image

and the model. The bounding boxes of the mandible of the model image and the scene image are computed and the scales in the x- and y-directions are determined by the ratio between the corresponding sides of the bounding boxes. Then, the scene contours are rescaled to fit the model contours.

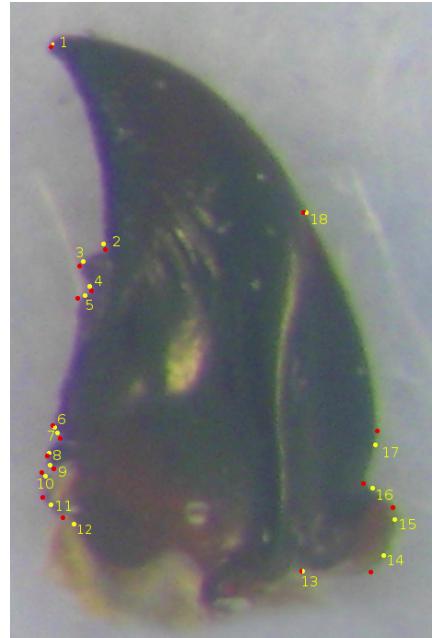


Figure 8: The manual (in red) and estimated (in yellow) landmarks on a right mandible.

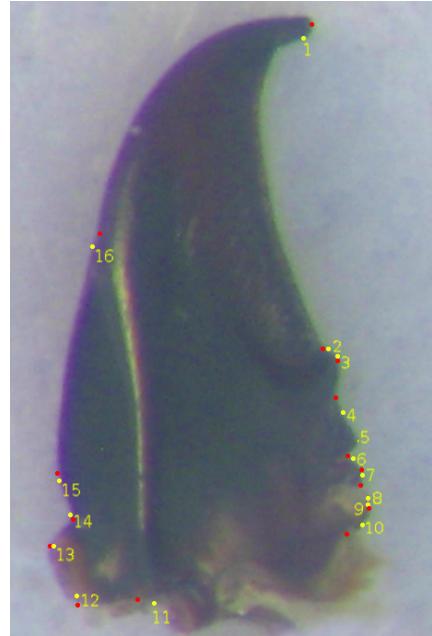


Figure 9: The manual (in red) and estimated (in yellow) landmarks of a left mandible.

¹ MAELab is a free software written in C++. It can be directly and freely obtained by request at the authors.

The Figs. 8 and 9 show the final results for a right and a left mandible with the manual and estimated landmarks. The estimated landmarks are quite near with the manual landmarks, as you can see with the following statistical evaluation.

The statistic are obtained for all landmarks of the scene images. We have compared the positions between the manual and estimated landmarks by accepting an error from 1% to 2% of the bounding box's size. According to this way, a global statistic compares all pairs of corresponding landmarks on all images as presented in Fig. 10. It shows the global results with a score of well positioned landmarks equal to **87.03%** for right mandibles and **78.82%** for left mandibles.

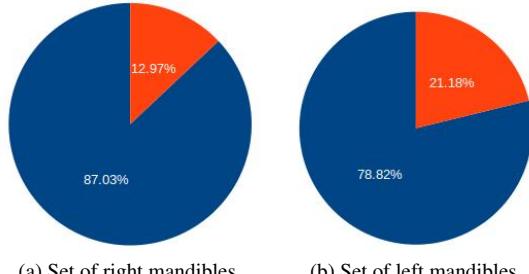


Figure 10: The mean proportion of well and bad landmark locations of the two sets of left and right mandibles.

Besides the global results, we are also interested by the accuracy of the individual positions of the estimated landmarks. We have computed the distance between the manual landmarks and their corresponding estimated landmarks in order to examine the proportion of well positioned landmarks. The Fig. 11 and 12 show the proportion of well estimated landmarks for each landmark of the model. With 18 landmarks of right mandible, the position of the 1st estimated landmarks is very accurate with **98.62%**. The lowest proportion is **74.48%** for the 14th landmark. The remaining landmarks are also estimated with a accuracy greater than 75%. For left mandibles, the highest and lowest success rates are **93.01%** for the 1st landmark and **60.14%** for the 16th landmark. The statistic is done on each estimated landmark of all the images with a standard deviation error [1]. As we can see in Fig. 3, the noise of the contour part located at the base of a mandible is higher than the noise located at the tip of the mandible. This explains why the correct proportion on 11th and 12th landmark of the left mandible and 13th and 14th landmark of the right mandible are less accurate than other landmarks. Moreover, when we reconsider the datasets, the left mandible images have bigger scale values than the right mandible images. This explains than the success rate on the right mandibles is always greater than on the left mandibles in all experiments.

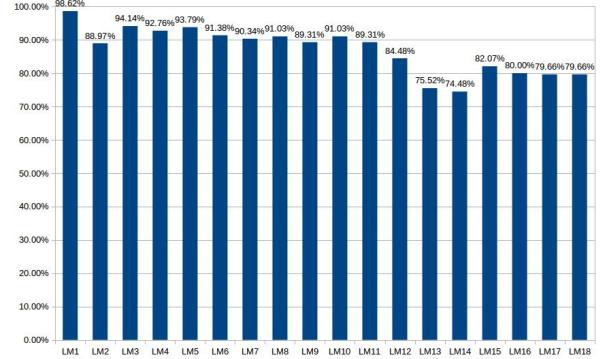


Figure 11: The proportions of well estimated landmark for each model landmark of right mandibles.

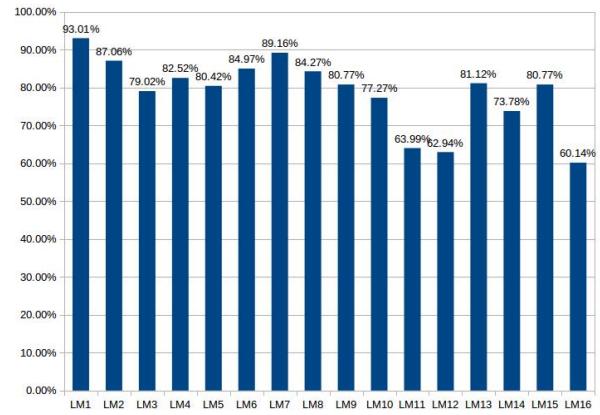


Figure 12: The proportions of well estimated landmark for each model landmark of left mandibles.

4 CONCLUSION

The morphometric analysis is a powerful tool in biology for the beetle species classification. In this paper, we have design a method to segment their mandibles and automatically locate landmarks which have been determined on a model by biologists. Each mandible has been segmented by using the Canny algorithm before to be registered using PCAI to align the images. The estimation step of the landmark position use the SIFT descriptor to find the best matching position. The results show that the method succeed in locating the landmarks for all images. The accuracy of the method is sufficient to be proposed to biologists as a replacement of the manual measures, but with a manual check for the bottom landmark. Moreover, considering the previous work in [4], this method reduces the drastically the number of outlier landmarks and the MAELab implementation also reduce the global computing times and memory cost. From now, the next stage consists in improving the registration step in order to increase the matching step accuracy and completely remove manual interventions. By example, we could investigate the isomorphic registration methods to be more robust to the inter-species variations of mandible shapes.

5 REFERENCES

- [1] J Martin Bland and Douglas G Altman. Statistics notes: measurement error. *Bmj*, 313(7059):744, 1996.
- [2] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.
- [3] Dehua Li Jun Zeng. An adaptive canny edge detector using histogram concavity analysis. *International Journal of Digital Content Technology and its Applications*, 5, 2011.
- [4] L Lê Vănh, M Beurton-Aimar, JP Salmon, A Marie, and N Parisey. Estimating landmarks on 2d images of beetle mandibles. *WSCG*, 2016.
- [5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [6] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.
- [7] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.