Robust Semi-Automatic Annotation of object Data Sets with Bounding Rectangles.

Abdelhamid ZAIDI (✉ A.ZAIDI@qu.edu.sa)

Qassim University

Research Article

Keywords: Rotated object detection, Rotated object detection datasets, Annotation with bounding rectangles, Minimum bounding rectangles, Estimation of the range.

Posted Date: September 7th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-860574/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Robust Semi-Automatic Annotation of object Data Sets with Bounding Rectangles

Abdelhamid Zaïdi*, *Department of mathematics, College of Science, Qassim University, P.O.Box 6644, Buraydah 51452, Saudi Arabia

Abstract—Object datasets used in the construction of object detectors are typically manually annotated with horizontal or rotated bounding rectangles. The optimality of an annotation is obtained by fulfilling two conditions (i) the rectangle covers the whole object (ii) the area of the rectangle is minimal. Building a large-scale object dataset requires annotators with equal manual dexterity to carry out this tedious work. When an object is horizontal, it is easy for the annotator to reach the optimal bounding box within a reasonable time. However, if the object is rotated, the annotator needs additional time to decide whether the object will be annotated with a horizontal rectangle or a rotated rectangle. Moreover, in both cases, the final decision is not based on any objective argument, and the annotation is generally not optimal. In this study, we propose a new method of annotation by rectangles, called Robust Semi-Automatic Annotation, which combines speed and robustness. Our method has two phases. The first phase consists in inviting the annotator to click on the most relevant points located on the contour of the object. The outputs of the first phase are used by an algorithm to determine a rectangle enclosing these points. To carry out the second phase, we develop an algorithm called RANGE-MBR, which determines, from the selected points on the contour of the object, a rectangle enclosing these points in a linear time. The rectangle returned by RANGE-MBR always satisfies optimality condition (i). We prove that the optimality condition (ii) is always satisfied for objects with isotropic shapes. For objects with anisotropic shapes, we study the optimality condition (ii) by simulations. We show that the rectangle returned by RANGE-MBR is quasi-optimal for the condition (ii), and that its performance increases with dilated objects, which is the case for most of the objects appearing on images collected by aerial photography.

Index Terms—Rotated object detection, Rotated object detection datasets, Annotation with bounding rectangles, Minimum bounding rectangles, Estimation of the range.

I. INTRODUCTION

The problem of detecting objects in an image consists in finding the position of all existing objects in that image and assigning a class to each object found. The first task is called object localization, while the second is called object classification. In computer vision, there are several ways to define the shape and location of the object, such as bounding rectangles, bounding polygons, segmentation mask, key-Point and landmark, and so on ([1],[2]). However, the use of bounding rectangles is the simplest and most commonly used type for locating objects in computer vision applications. We say that an object is well detected if the bounding rectangle has a minimum area and covers the maximum of visible parts of the object.

The development of a robust object detector generally goes through two or even three phases which are learning, testing,

and tuning phases. Each phase requires an independent set of meticulously prepared images by a team of human experts called annotators. The main function of the annotator is to draw, with the highest possible accuracy, a rectangle around the object of interest, and to choose from a list of classes the one that describes the object. These rectangles are commonly known as the ground truth bounding boxes. The combination of the raw image, the ground truth bounding box, and the object label constitutes the elements of an annotated image. The annotation is done using commercial or free software. Some software are optimized for a given type of annotation, and others offer many tools that allow the user to process various types of annotations.

To create a dataset for object detection, we need to collect a large number of images, annotated with the same method, and distributed according to a predefined list of object classes. For each object class, the dataset must contain several images corresponding to different instances of this class. We can divide the existing annotation methods into two main classes. The first class includes all manual annotation methods, while the second class groups all semi-automatic annotation methods.

Advanced object detectors are based on convolutional neural network approach. These detectors can be divided into two main classes [3]. The first class involves two-stage detectors (one for localization and the other for classification) such as Faster Region Based Convolutional Neural Networks (R-CNN) [4] and cascade R-CNN [5], while the second class includes one-stage detectors (for which localization and classification are carried out in a single-stage) such as You Only Look Once (YOLOv3) [6] and Single Shot Detector (SSD) [7]. The first class of detectors is distinguished by its accuracy rate, while the second class of detectors is distinguished by its inference speed [3]. A convolutional neural network-based detector uses the first subset of the annotated dataset in order to conduct the training stage.

When an object appearing on an image is aligned with the horizontal axis, it is natural to represent it by a horizontal rectangle. In contrast, if the object is aligned with an oblique axis, then the representation of the object by a horizontal rectangle certainly leads to a bad localization and identification of the object, since this rectangle includes a more or less important part of the image background. The worst situation is when the object of interest is aligned with an oblique axis making an angle of 45 degrees with the horizontal axis. Based on the orientation of the object of interest, the object detection problem can be divided into two sub-problems, which are Horizontal Object Detection (HOD) and Rotated Object Detection

(ROD) [8]. The first paragraph was an introduction to HOD. In order to complete the description of the object detection problem, we give here a brief overview of the ROD and we specify the inherent difficulties with this topic.

Advanced rotated object detectors are based on convolutional neural networks approach. The learning and testing sets used in the different phases of these detectors are annotated with rotated rectangles. Most of these datasets contain images collected from aerial photographs where objects are rarely aligned horizontally. As examples of common use rotated datasets, we cite UCAS-AOD [9] whose objects are restricted to vehicles and planes, HRSC2016 [10] which contains only ship instances, and DOTA [11] which is considered to be the richest rotated dataset, with fifteen object classes and 2806 aerial images. All the above-mentioned object datasets are hand-annotated by experts with rotated rectangles. Like HOD methods, ROD methods are also divided into two main classes. The first class involves two-stage detectors such as CIN [12], Region of Interests (RoI) Transformer [13], and Small Cluttered and Rotated objects Detector (SCRDet) [14], while the second class includes one-stage detectors such as Efficient and Accurate Scene Text detector (EAST) [15] and Refined Rotation RetinaNet Detector (R³Det) [16]. In addition, the prediction bounding rectangles generated by rotated object detectors are also assumed to be rotated.

II. STATE OF THE ART

A. Manual annotation

There are two main methods of manually annotating objects with bounding boxes which are used in the construction of most large scale object datasets. The first one is called the consensus method and the second one is called the sequential tasks method. For each instance of an object in an image, the first method asks several annotators to draw a rectangle around the object, then defines the position of the object by the rectangle elected by the majority of annotators. To annotate an object by the second method, we need at least three annotators. The first one is asked to draw a rectangle around a single instance of the object. The task of the second annotator is to validate the drawn rectangle. The third person checks if there are other instances of the same object class that need to be annotated. Determining an accurate bounding rectangle requires much more human and material resources than verifying the annotation. Determining an accurate bounding rectangle requires much more human and material resources than checking the annotation, that is why the consensus method is more efficient than the sequential tasks method. The quality of the annotation has a direct impact on the development of precise object detectors. The determination of a large-scale object dataset requires expert annotators, significant working time, and remuneration in line with the desired accuracy. As examples of common use datasets, annotated with the manual methods, we cite PASCAL VOC [17], MS COCO [18], and IMAGENET [19].

B. Semi-automatic annotation

The traditional semi-automatic approach of annotating objects with rectangles consists of four phases.

First phase: in this phase, the images of the object dataset are divided into two subsets of unequal sizes. The smallest subset is annotated with rectangles by the manual method.

Second phase: the second phase consists of choosing an object detector and training this detector with the images from the dataset already annotated.

Third phase: once the detector is ready, it will be used to annotate the images of the second object dataset with the prediction rectangles.

Fourth phase: during this phase, the annotator validates the correctly annotated objects during the third phase, and manually draws the bounding rectangles of the poorly annotated objects.

As this study is limited to locating objects by bounding boxes, we cite the Faster Bounding Box (FBB) as an example of semi-automatic annotation method [20]. This method was used to generate the Tampere University of Technology (TUT) indoor dataset. The largest subset is annotated with prediction rectangles, generated by the Faster R-CNN object detector [4], trained on the smallest annotated subset.

C. Performance measure of detectors using rectangles

The evaluation of the effectiveness of the object detector is then performed on the second subset of the annotated dataset. Since we are only dealing here with annotations using bounding boxes, the most used criterion to compare two rectangles is the Jaccard's similarity index, also known as the Intersection over Union (IoU). For each image of the testing dataset, the IoU measures the percentage of overlap between the prediction bounding rectangle $\hat{\mathbb{A}}$ generated by the detector, and the ground truth bounding box $\mathbb{A}$, as follows:

$$\text{IoU}(\hat{\mathbb{A}}, \mathbb{A}) = \frac{\text{Area}(\hat{\mathbb{A}} \cap \mathbb{A})}{\text{Area}(\hat{\mathbb{A}} \cup \mathbb{A})} \quad (1)$$

Note that the IoU score lies in the interval $[0, 1]$. The closer this index gets to 1, the better the detection of the object. We say that an object is well detected (or true positive) if the IoU score is greater than or equal to a threshold discussed by experts. In most studies, the threshold value is set at 0.5 [21]. The IoU score causes a problem when its value is zero. This value is embarrassing since it does not explain how far is the prediction bounding rectangle from the ground truth bounding box. To work around this problem, Rezatofighi *et al* [21] suggest replacing the IoU with the Generalized Intersection over Union (GIoU) index. The GIoU between two rectangles $\hat{\mathbb{A}}$ and $\mathbb{A}$ is defined as follows:

$$\text{GIoU}(\hat{\mathbb{A}}, \mathbb{A}) = \text{IoU}(\hat{\mathbb{A}}, \mathbb{A}) - \frac{\text{Area}(\mathbb{C} \setminus (\hat{\mathbb{A}} \cup \mathbb{A}))}{\text{Area}(\mathbb{C})} \quad (2)$$

where $\mathbb{C}$ is the smallest convex set enclosing both $\hat{\mathbb{A}}$ and $\mathbb{A}$.

To the best of our knowledge, in the literature of ROD, the IoU between the ground-truth box $\mathbb{A}$ and the predicted bounding rectangles $\mathbb{B}$, has never been calculated explicitly as in the case of HOD. Liu *et al* proposed in [22] to approximate $\text{IoU}(\mathbb{A}, \mathbb{B})$ by the angle-related IoU defined by the relation (3):

$$\text{ArIoU}(\mathbb{A}, \mathbb{B}) = \frac{\text{Area}(\mathbb{A}' \cap \mathbb{B})}{\text{Area}(\mathbb{A}' \cup \mathbb{B})} \cos(\alpha - \beta) \quad (3)$$

Fig. 1. Liu's rule approximates $\text{IoU}(\mathbb{A}, \mathbb{B})$ by $\text{IoU}(\mathbb{A}', \mathbb{B}')$, where $\mathbb{A} = [A_1, A_2, A_3, A_4]$ and $\mathbb{A}' = [A'_1, A'_2, A'_3, A'_4]$.

Fig. 2. Yao's rule approximates $\text{IoU}(\mathbb{A}, \mathbb{B})$ by $\text{IoU}(\mathbb{A}', \mathbb{B}')$, where $\mathbb{A} = [A_1, A_2, A_3, A_4]$ and $\mathbb{A}' = [A'_1, A'_2, A'_3, A'_4]$.

where α and β are the angles of $\mathbb{A}$ and $\mathbb{B}$ respectively, and $\mathbb{A}'$ coincides with the rotation of $\mathbb{A}$, round the center A_0 , by the angle $\theta = (\beta - \alpha)$ measured counterclockwise from the positive x -axis (as shown in Figure 1). If $\mathbb{A}$ and $\mathbb{B}$ are parametrized as described in Section IV, then α (resp. β) is the angle between the positive x -axis and vector $\overrightarrow{A_1A_2}$ (resp. $\overrightarrow{B_1B_2}$). Yao *et al* [23] approximate $\text{IoU}(\mathbb{A}, \mathbb{B})$ by $\text{IoU}(\mathbb{A}', \mathbb{B}')$, where $\mathbb{A}' = [A'_1, A'_2, A'_3, A'_4]$ (resp. $\mathbb{B}' = [B'_1, B'_2, B'_3, B'_4]$) is the horizontal rectangle obtained by rotating $\mathbb{A}$ (resp. $\mathbb{B}$) round the center A_0 (resp. B_0) as shown in Figure 2.

D. Real-life use of annotation with bounding rectangles

We present in this paragraph a non-exhaustive list of applications using annotation with bounding rectangles.

- 1) **Perform missions with drones:** object detectors, used by drones, are also trained with rotated object datasets containing images collected from aerial photographs. So, these object detectors help drones to locate and identify their target during a future mission.
- 2) **Autonomous vehicles:** the development of drivers for autonomous vehicles uses images of road traffic annotated with bounding rectangles to train detectors of objects commonly encountered on the roads, such as pedestrians, cars, motorcycles, bicycles, traffic signs and so on. Thus, these object detectors can help autonomous vehicles to move safely while avoiding all types of obstacles.
- 3) **Online retail:** web search engines used by online sales sites are trained with images of clothing and accessories, annotated with bounding rectangles. Thus, customers can

easily find the items they want to purchase using images containing those items.

For more details on annotation methods, performance metrics, object datasets, and real-life applications, we refer the reader to [24] and [25].

III. MOTIVATION FOR THE STUDY

The manual or semi-automatic annotation methods described in Section II, have made it possible to build many large-scale object datasets. These datasets have given rise to very powerful object detectors. However, we can not ignore the following concerns:

- 1) According to paragraph II-A, accurate manual annotation is expensive, time consuming, and requires annotation experts.
- 2) In many situations, the annotator can be in front of an object with an ambiguous orientation as shown in Figure 3. In this case, he will take a considerable time to decide whether the object will be annotated with a horizontal rectangle or a rotated rectangle. Moreover, in both cases, the final decision is not based on any objective argument and depends solely on the dexterity of the annotator.
- 3) The semi-automatic annotation defined in paragraph II-B, does not guarantee the optimality of the annotation rectangle, in the sense that it must have a minimum area and cover the whole of the object.
- 4) According to paragraph II-C, the use of any approximation methods of $\text{IoU}(\mathbb{A}, \mathbb{B})$, whether to measure the accuracy of a rotated object detector or to compare the performance of two rotated object detectors, could lead to biased results. Indeed, the function $g(x) = \frac{x}{a+b-x}$ is strictly increasing over the interval $[0, a+b]$ over the interval, where a and b denotes the areas of $\mathbb{A}$ and $\mathbb{B}$ respectively. Consequently, an underestimation (resp. overestimation) of the overlapping area between $\mathbb{A}$ and $\mathbb{B}$ directly induces an underestimation (resp. overestimation) of $\text{IoU}(\mathbb{A}, \mathbb{B})$.

Contribution of the this study

In this paper,

- we develop an algorithm called RANGE-MBR, which determines from a set $\mathcal{M}_n$ of the n most relevant (in the sense given by Definition 3.1 below) points picked on the object outline, a rectangle enclosing $\mathcal{M}_n$ and having a quasi-minimal area, in $\mathcal{O}(n)$ time,
- we propose a new approach to simultaneously build HOD and ROD datasets from a large-scale image bank, based on both the RANGE-MBR algorithm and threshold angles,
- we conduct a large experimental study to quantify the performance of the RANGE-MBR algorithm.
- we compare the performance of RANGE-MBR to that of the benchmark algorithm, RC-MBR which determines the minimum rectangle enclosing $\mathcal{M}_n$ in $\mathcal{O}(n \ln(n))$ time.

Definition 3.1 (relevant point of an object): Any point located on the contour of the object is said to be relevant

Fig. 3. Figure (b): Intuitively, it seems that the horizontal rectangle $\mathbb{B} = [B_1, \dots, B_4]$ is best suited to annotate the colored object in blue (Figure (a)). The method we propose takes as input the set $\mathcal{M}_6 = \{M_1, \dots, M_6\}$ and returns the rotated rectangle $\mathbb{A} = [A_1, \dots, A_4]$. Since $\mathbb{A}$ covers the whole object and $\text{Area}(\mathbb{A}) < \text{Area}(\mathbb{B})$, then it is natural to annotate the object by the rectangle $\mathbb{A}$.

if it is a local maximum, a local minimum, the most right point, the most left point, a cusp of the first type, a cusp of the second type, an inflection point, and so on.

The remainder of this manuscript is structured as follows. Section IV deals with the parametrization of rectangles. Section V is devoted to the development of the RANGE-MBR algorithm. We explain in section VI how to use the RANGE-MBR (or RC-MBR) algorithm to simultaneously generate horizontal and rotated datasets from a large-scale image bank. In section VII, we perform various numerical experiments to evaluate the performance of the RANGE-MBR algorithm.

IV. PARAMETRIZATION OF BOUNDING BOXES

A. Sorting the vertices of a rectangle

Definition 4.1 (Polar angle): Let Δ be a horizontal line and G a point of Δ . For any point $M = (x, y)^t$ of the Euclidean affine plane, we define the polar angle of M with respect to $G = (\bar{x}, \bar{y})^t$ as the angle between the positive x -axis with origin G and the vector $\overrightarrow{GM}$ as shown in Figure 4. This angle can be determined using the operation $\text{mod}(\text{atan}_2(y - \bar{y}, x - \bar{x}), 2\pi)$, where mod designates the modulo symbol and atan_2 is a mathematical function available in most programming languages.

We use the notation $\mathbb{R} = [R_1, R_2, R_3, R_4]$ to denote a rotated rectangle whose vertices are R_1, R_2, R_3 , and R_4 . We assume that these vertices are sorted in counterclockwise order, such that R_1 is the left vertex with the smallest vertical coordinate.

To sort the vertices set $\{A_1, A_2, A_3, A_4\}$ of the rectangle $\mathbb{A}$ as described above, we first consider the barycenter G of

Fig. 4. θ (resp. α) is the polar angle of M (resp. N) relative to G .

the rectangle $\mathbb{A}$. Then, we calculate the polar angle of each vertex relative to the barycenter G . Denote by $A_{(1)}, \dots, A_{(4)}$ the sorted vertices in ascending order of their polar angle. If $A_{(3)}(2) \leq A_{(4)}(2)$, then $\mathbb{A} = [A_{(3)}, A_{(4)}, A_{(1)}, A_{(2)}]$, else $\mathbb{A} = [A_{(4)}, A_{(1)}, A_{(2)}, A_{(3)}]$. Algorithm 1 summarizes the steps for sorting the vertices of $\mathbb{A}$.

Algorithm 1 Normalize_Rectangle($\mathbb{A}$)

Input: A list of vertices $\{A_1, A_2, A_3, A_4\}$ of the rectangle $\mathbb{A}$

Output: Vertices of $\mathbb{A}$ sorted as described in Section IV

```

1: Set  $\mathbf{x}$  = the  $x$ -coordinates of  $\mathbb{A}$ 
2: Set  $\mathbf{y}$  = the  $y$ -coordinates of  $\mathbb{A}$ 
3: for ( $i = 1 : 4$ ) do
4:    $t = \text{mod}(\text{atan}_2(y - \bar{y}, x - \bar{x}), 2\pi)$ 
5:   Append  $t$  to Angles
6: end for
7: [Sorted_Angle, idx] = sort(Angles, 'ascending')
8: if ( $\mathbf{y}(3) \leq \mathbf{y}(4)$ ) then
9:    $\mathbb{A} = [A_{\text{idx}(3)}, A_{\text{idx}(4)}, A_{\text{idx}(1)}, A_{\text{idx}(2)}]$ 
10: else
11:    $\mathbb{A} = [A_{\text{idx}(4)}, A_{\text{idx}(1)}, A_{\text{idx}(2)}, A_{\text{idx}(3)}]$ 
12: end if
13: return  $\mathbb{A}$ 

```

B. Parameterization of horizontal rectangles

The minimum parameterization of a horizontal rectangle $\mathbb{A} = [A_1, A_2, A_3, A_4]$ requires four parameters. The object dataset PASCAL VOC [17] annotates the rectangle $\mathbb{A}$ with (x_4, y_4, x_2, y_2) , where (x_i, y_i) are the coordinates of the vertex A_i , $i = 1, \dots, 4$ (cf. Figure 5). The object dataset MS COCO [18] annotates the rectangle $\mathbb{A}$ with (x_4, y_4, w, h) , where w and h denote the lengths of the line segments $[A_1, A_2]$ and $[A_1, A_4]$ respectively, and (x_4, y_4) are the coordinates of the vertex A_4 (cf. Figure 5). The object dataset IMAGENET [19] annotates the rectangle $\mathbb{A}$ with (x_0, y_0, w, h) , where w and h denote the lengths of the line segments $[A_1, A_2]$ and $[A_1, A_4]$ respectively, and (x_0, y_0) are the coordinates of the center A_0 of $\mathbb{A}$ (cf. Figure 5).

C. Parameterization of rotated rectangles

The minimum parameterization of a rotated rectangle $\mathbb{B} = [B_1, B_2, B_3, B_4]$ requires five parameters (b_1, b_2, w, h, θ) . The

Fig. 5. Parameterization of horizontal and rotated rectangles.

Fig. 6. $\mathcal{M}_5 = \{M_1, \dots, M_5\}$. The green and red rectangles are two distinct solutions of Problem 5.1.

couple (b_1, b_2) corresponds to the coordinates of the center B_0 of $\mathbb{B}$. The parameters w and h denote the lengths of the line segments $[B_1, B_2]$ and $[B_1, B_4]$ respectively (cf. Figure 5). The parameter $\theta \in [0, \frac{\pi}{2}]$ define the acute angle between the lines Δ and (B_1, B_2) , where Δ is the horizontal line passing through B_1 . This parametrization is used to annotate the object datasets HRSC2016 [10] and UCAS-AOD [9]. For the sake of simplicity, other object datasets, such as DOTA [11], parameterize a rectangle using the eight coordinates of its vertices. This over-parameterization avoids additional calculations to plot the rectangle, or to find the image of this rectangle by an affine transformation.

In this study, we opt for the parametrization of a rectangle $\mathbb{B}$ with the eight-tuple $(x_1, y_1, \dots, x_4, y_4)$, where $(x_i, y_i)^t$ are the coordinates of the vertex B_i , $i = 1, \dots, 4$, knowing that one can easily switch from one parametrization to another.

V. DETERMINATION OF THE BOUNDING RECTANGLE

It is natural to annotate the object with the minimum rectangle enclosing the set of relevant points. This problem is formulated as follows:

Problem 5.1: Given a set of points $\mathcal{M}_n = \{M_i = (x_i, y_i)^t \in \mathbb{R}^2; i = 1, \dots, n\}$, find a rectangle with the smallest area, enclosing $\mathcal{M}_n$. Such a rectangle is called a Minimum Bounding Rectangle of $\mathcal{M}_n$ and denoted by $\text{MBR}(\mathcal{M}_n)$.

The solution of Problem 5.1 is not unique as shown in Figure 6. So, $\text{MBR}(\mathcal{M}_n)$ denotes any solution of Problem 5.1.

Freeman and Shapira proved in 1975 that one edge of $\text{MBR}(\mathcal{M}_n)$ must be collinear with an edge of the convex hull ($\text{CH}(\mathcal{M}_n)$) of $\mathcal{M}_n$. They proposed in [27] a natural algorithm to find $\text{MBR}(\mathcal{M}_n)$ in $\mathcal{O}(n^2)$ time, based on sweeping all minimum rectangles, enclosing $\mathcal{M}_n$ and having an edge collinear with an edge of $\text{CH}(\mathcal{M}_n)$. In 1978 Shamos proposed in [28] the famous rotating calipers algorithm, which return all pairs of antipodal vertices of a n -sided convex polygon in $\mathcal{O}(n)$ time. In 1983 Toussaint used the rotating calipers technique, and developed the algorithm RC-MBR to find $\text{MBR}(\mathcal{M}_n)$ in $\mathcal{O}(n \ln(n))$ time [29]. In 2006, Dimitrov *et al* proposed in ([30],[31]) the algorithm PCA-MBR, to approximate $\text{MBR}(\mathcal{M}_n)$, in $\mathcal{O}(n \ln(n))$ time, by the minimum bounding rectangle which is aligned with the eigenvectors of the covariance matrix of $\text{CH}(\mathcal{M}_n)$. They also proved that the relative error between $\text{PCA-MBR}(\mathcal{M}_n)$ and $\text{MBR}(\mathcal{M}_n)$ is bounded from above by $\sqrt{8\sqrt{2}} - 1 \approx 2.36$. The main drawback of the algorithm PCA-MBR is that it admits an infinite number of solutions if the covariance matrix of $\text{CH}(\mathcal{M}_n)$ has a double eigenvalue. To overcome the problem of non-uniqueness inherent in algorithms RC-MBR and PCA-MBR, we propose the method RANGE-MBR to approximate $\text{MBR}(\mathcal{M}_n)$ in $\mathcal{O}(n)$ time.

A. The RANGE-MBR algorithm

Let $\mathcal{B} = (O, \mathbf{i}, \mathbf{j})$ be a Cartesian frame of the two dimensional affine plane. All angles we refer to here are measured counterclockwise from the positive x -axis. Let Δ be the line passing through O and making an angle θ with the axis $(O, \mathbf{i})$. Let $\Delta^\perp$ be perpendicular to Δ and passing through O . Let P_1 and Q_1 (resp. P_2 and Q_2) be the extreme points of the orthogonal projection of $\mathcal{M}_n$ on Δ (resp. $\Delta^\perp$). Let $\mathbb{A} = [A_1, A_2, A_3, A_4]$ be the minimum bounding rectangle having an edge collinear with Δ . Then $A_1, \dots, A_4$ are the intersection points of the two parallel lines to $\Delta^\perp$ and passing through P_1 and Q_1 respectively, with the two parallel lines to Δ and passing through P_2 and Q_2 respectively. The area of $\mathbb{A}$ is given by the product of lengths of the line segments $[P_1, Q_1]$ and $[P_2, Q_2]$. From now on, the MBR $\mathbb{A}$ enclosing $\mathcal{M}_n$ and having an edge collinear with the direction making an angle θ from the positive x -axis, will be denoted by $\text{MBR}(\mathcal{M}_n, \theta)$. Let $\mathbf{u} = (\cos(\theta), \sin(\theta))^t$ (resp. $\mathbf{v} = (-\sin(\theta), \cos(\theta))^t$) be a unit direction vector of Δ (resp. $\Delta^\perp$). Then, $\mathcal{B} = (O, \mathbf{i}, \mathbf{j})$ and $\mathcal{B}' = (O, \mathbf{u}, \mathbf{v})$ are two Cartesian frames of the two dimensional affine plane. For all $i = 1, \dots, n$, denote by $(x_i, y_i)^t$ (resp. $(s_i, t_i)^t$) the coordinates of M_i with respect to the frame $\mathcal{B}$ (resp. $\mathcal{B}'$). Let R_θ be the rotation matrix of angle θ , then $(x_i, y_i)^t = R_\theta(s_i, t_i)^t$, where $\mathbf{u}^t$ is the transpose of $\mathbf{u}$. Let X be a random variable with mean μ and standard deviation σ . Let $\mathbf{x}_{1:n} = (x_1, \dots, x_n)$ be n observations on the variable X . Let $x_{(1)} \leq \dots \leq x_{(n)}$ be the order statistics of $\mathbf{x}_{1:n}$. We define the range of $\mathbf{x}_{1:n}$ by $\text{Range}(\mathbf{x}_{1:n}) = x_{(n)} - x_{(1)}$. Therefore, the lengths of $[P_1, Q_1]$ and $[P_2, Q_2]$ are given by:

$$\begin{aligned} P_1Q_1 &= \text{Range}(\{s_1, \dots, s_n\}) = s_{(n)} - s_{(1)} \\ P_2Q_2 &= \text{Range}(\{t_1, \dots, t_n\}) = t_{(n)} - t_{(1)} \end{aligned} \quad (4)$$

Denote by $\hat{\sigma}_n$ any estimation of σ obtained from $\mathbf{x}_{1:n}$. On the basis of numerous works carried out on the estimation of the standard deviation from the sample range ([32], [33], [34]), we can assume that there is a real constant $\alpha(X, n)$ such that $\hat{\sigma}_n = \frac{\text{Range}(\mathbf{x}_{1:n})}{\alpha(X, n)}$. Besides, it is well known that $\text{Var}(\mathbf{x}_{1:n}) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$ is an estimate of σ^2 , where $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$. It follows that there is a real constant $\alpha(X, n)$ such that:

$$\text{Range}(\mathbf{x}_{1:n})^2 \approx \alpha(X, n)^2 \text{Var}(\mathbf{x}_{1:n}) \quad (5)$$

Assume that the x -coordinates (resp. the y -coordinates) of the set $\mathcal{M}_n$ are n observations of a random variable X (resp. Y). Since the $\text{Area}(\mathbb{A}) = P_1 Q_1 \cdot P_2 Q_2$, then combining equations (5) and (4) gives an approximation $\hat{A}_n^2(\theta)$ of $\text{Area}(\mathbb{A})$, satisfying the following relation:

$$\hat{A}_n^2(\theta) = \alpha(S, n)^2 \alpha(T, n)^2 \text{Var}(\mathbf{s}_{1:n}) \text{Var}(\mathbf{t}_{1:n}) \quad (6)$$

where $(S, T)^t = R_\theta^t(X, Y)^t$. To alleviate notions, we will also use $\mathbb{V}_{\mathbf{X}}$ and $\mathbb{C}_{\mathbf{X}, \mathbf{Y}}$ to designate $\text{Var}(\mathbf{x}_{1:n})$ and $\text{Cov}(\mathbf{x}_{1:n}, \mathbf{y}_{1:n})$. We can easily Since $(x_i, y_i)^t = R_\theta(s_i, t_i)^t$, we obtain:

$$\begin{aligned} \mathbb{V}_{\mathbf{S}} &= \cos^2(\theta) \mathbb{V}_{\mathbf{X}} + \sin^2(\theta) \mathbb{V}_{\mathbf{Y}} + \sin(2\theta) \mathbb{C}_{\mathbf{X}, \mathbf{Y}} \\ \mathbb{V}_{\mathbf{T}} &= \sin^2(\theta) \mathbb{V}_{\mathbf{X}} + \cos^2(\theta) \mathbb{V}_{\mathbf{Y}} - \sin(2\theta) \mathbb{C}_{\mathbf{X}, \mathbf{Y}} \end{aligned} \quad (7)$$

Using (7) and some trigonometric identities, we prove that:

$$8\mathbb{V}_{\mathbf{S}} \mathbb{V}_{\mathbf{T}} = K - f(\theta) \quad (8)$$

where, $K = \mathbb{V}_{\mathbf{X}}^2 + \mathbb{V}_{\mathbf{Y}}^2 + 6\mathbb{V}_{\mathbf{X}} \mathbb{V}_{\mathbf{Y}} - 4\mathbb{C}_{\mathbf{X}, \mathbf{Y}}^2$ and

$$\begin{aligned} f(\theta) &= (\mathbb{V}_{\mathbf{X}}^2 + \mathbb{V}_{\mathbf{Y}}^2 - 2\mathbb{V}_{\mathbf{X}} \mathbb{V}_{\mathbf{Y}} - 4\mathbb{C}_{\mathbf{X}, \mathbf{Y}}^2) \cos(4\theta) \\ &+ 4\mathbb{C}_{\mathbf{X}, \mathbf{Y}} (\mathbb{V}_{\mathbf{X}} - \mathbb{V}_{\mathbf{Y}}) \sin(4\theta) \end{aligned} \quad (9)$$

Combining equations (6), (8) and (9) gives:

$$\hat{A}_n^2(\theta) = \frac{\alpha(S, n)^2 \alpha(T, n)^2}{8} (K - f(\theta)), \quad (10)$$

Thus, the area of $\text{MBR}(\mathcal{M}_n, \theta)$ is approximately equal to $\frac{\alpha(S, n)^2 \alpha(T, n)^2}{8} (K - f(\theta))$. Therefore, we propose to approximate $\text{MBR}(\mathcal{M}_n)$ by $\text{MBR}(\mathcal{M}_n, \theta^*)$ such that:

$$\theta^* = \underset{\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]}{\text{argmin}} \hat{A}_n^2(\theta) = \underset{\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]}{\text{argmax}} f(\theta) \quad (11)$$

The first derivative of f with respect to θ is given by:

$$\dot{f}(\theta) = -4\lambda \sin(4\theta) + 4\delta \cos(4\theta) \quad (12)$$

where,

$$\begin{aligned} \lambda &= (\mathbb{V}_{\mathbf{X}}^2 - \mathbb{V}_{\mathbf{Y}}^2) - 4\mathbb{C}_{\mathbf{X}, \mathbf{Y}}^2 \\ \delta &= 4\mathbb{C}_{\mathbf{X}, \mathbf{Y}} (\mathbb{V}_{\mathbf{X}} - \mathbb{V}_{\mathbf{Y}}) \end{aligned} \quad (13)$$

The function f has a unique critical point θ^* such that:

$$\tan(4\theta^*) = \frac{\delta}{\lambda} = \frac{4\mathbb{C}_{\mathbf{X}, \mathbf{Y}} (\mathbb{V}_{\mathbf{X}} - \mathbb{V}_{\mathbf{Y}})}{(\mathbb{V}_{\mathbf{X}}^2 - \mathbb{V}_{\mathbf{Y}}^2) - 4\mathbb{C}_{\mathbf{X}, \mathbf{Y}}^2} \quad (14)$$

Lemma 5.1: The second derivative of f at the critical point θ^* has the same sign as $-\lambda$.

Proof : The second derivative of f with respect to θ is given by:

$$\ddot{f}(\theta) = -16(\lambda \cos(4\theta) + 4\delta \sin(4\theta)) \quad (15)$$

Fig. 7. When θ is fixed, the minimum bounding rectangle enclosing the blue polygon (object) is the rectangle $\mathbb{A}$ whose vertices $A_1, \dots, A_4$ are the intersection points of the two parallel lines to $\Delta^\perp$ and passing through P_1 and Q_1 respectively, with the two parallel lines to Δ and passing through P_2 and Q_2 respectively. $\mathbb{A}$ is the rectangle whose edges pass through the red dotted segments.

Multiplying both side of the relation (15) by $\cos(4\theta)$ yields:

$$\cos(4\theta) \ddot{f}(\theta) = -16(\lambda \cos^2(4\theta) + 4\delta \sin(4\theta) \cos(4\theta)) \quad (16)$$

Multiplying both side of the relation (12) by $\sin(4\theta)$ yields:

$$\sin(4\theta) \dot{f}(\theta) = -4\lambda \sin^2(4\theta) + 16\delta \sin(4\theta) \cos(4\theta) \quad (17)$$

Using relations (17) and (16) with $\theta = \theta^*$, we obtain:

$$\cos(4\theta^*) \ddot{f}(\theta^*) = -16\lambda \quad (18)$$

Therefore, $\ddot{f}(\theta^*)$ has the same sign as $-\lambda \cos(4\theta^*)$. Since $4\theta^* = \arctan(\frac{\delta}{\lambda}) \in]-\frac{\pi}{2}, \frac{\pi}{2}[$, then $\cos(4\theta^*) > 0$ what ever the set $\mathcal{M}_n$ may be. Thus, $\ddot{f}(\theta^*)$ has the same sign as $-\lambda$. $\blacksquare$ Therefore, the determination of θ^* goes through six cases:

case 1: if $\lambda > 0$, then $\theta^* = \frac{1}{4} \arctan(\frac{\delta}{\lambda})$

case 2: if $\lambda < 0$ and $\mathbb{V}_{\mathbf{X}} \neq \mathbb{V}_{\mathbf{Y}}$, then θ^* is determined through the eigendecomposition of the empirical covariance matrix of $\mathcal{M}_n$. This case is discussed below.

case 3: if $\lambda < 0$ and $\mathbb{V}_{\mathbf{X}} = \mathbb{V}_{\mathbf{Y}}$, then $\theta^* = \frac{\pi}{4}$

case 4: if $\lambda = 0$ and $\mathbb{C}_{\mathbf{X}} (\mathbb{V}_{\mathbf{X}} - \mathbb{V}_{\mathbf{Y}}) > 0$, then $\theta^* = \frac{\pi}{8}$

case 5: if $\lambda = 0$ and $\mathbb{C}_{\mathbf{X}} (\mathbb{V}_{\mathbf{X}} - \mathbb{V}_{\mathbf{Y}}) < 0$, then $\theta^* = -\frac{\pi}{8}$

case 6: if $\lambda = 0$ and $\mathbb{C}_{\mathbf{X}} (\mathbb{V}_{\mathbf{X}} - \mathbb{V}_{\mathbf{Y}}) = 0$, which is equivalent to $\mathbb{V}_{\mathbf{X}} = \mathbb{V}_{\mathbf{Y}}$ and $\mathbb{C}_{\mathbf{X}} = 0$. This case is discussed below.

1) *Illustrative example of case 6:* It is difficult to illustrate all the scenarios which fall into case 6. The special case of regular polygons is the one that the annotator may encounter when annotating regular objects such as road signs, buildings, human faces, etc. Moreover, we have pointed out this case to warn the user to pay attention to regularly shaped objects when using our annotation method.

Proposition 5.1: If the elements of $\mathcal{M}_n$ are the vertices of a regular n -sided polygon, then $\mathbb{V}_{\mathbf{X}} = \mathbb{V}_{\mathbf{Y}}$ and $\mathbb{C}_{\mathbf{X}, \mathbf{Y}} = 0$.

Proof : In the general case, the vertices of a regular polygon are uniformly distributed over a circle with radius r and center G equal to the barycenter of $\mathcal{M}_n$. Thus, without loss of generality, we can assume that $r = 1$, $G = (0, 0)^t$, and the k^{th} vertex of $\mathcal{M}_n$ is

$$M_k = \left(\cos\left(\frac{2\pi(k-1)}{n}\right), \sin\left(\frac{2\pi(k-1)}{n}\right) \right)^t.$$

Consider the complex sequence $(z^k)_{0 \leq k \leq n-1}$, where $z = \exp(\frac{2\pi}{n}i)$ and i is the imaginary unit. Then

$$\sum_{k=0}^{n-1} z^k = n(\bar{x} + \bar{y}i)$$

Since $\sum_{k=0}^{n-1} z^k = \frac{1-z^n}{1-z} = 0$, then $\bar{x} = \bar{y} = 0$, and

$$\mathbb{C}_{\mathbf{x}, \mathbf{y}} = \frac{1}{n} \sum_{k=1}^n x_k y_k; \mathbb{V}_{\mathbf{x}} = \frac{1}{n} \sum_{k=1}^n x_k^2; \mathbb{V}_{\mathbf{y}} = \frac{1}{n} \sum_{k=1}^n y_k^2.$$

Since $\cos(2x) = 2\cos^2(x) - 1 = 1 - 2\sin^2(x)$, then

$$\mathbb{V}_{\mathbf{x}} = \frac{1}{2} + \frac{1}{2n} \sum_{k=0}^{n-1} \cos(\frac{4\pi k}{n}); \mathbb{V}_{\mathbf{y}} = \frac{1}{2} - \frac{1}{2n} \sum_{k=0}^{n-1} \cos(\frac{4\pi k}{n}).$$

Besides, $\sum_{k=0}^{n-1} \cos(\frac{4\pi k}{n})$ is the real part of

$$\sum_{k=0}^{n-1} (z^2)^k = \frac{1-z^{2n}}{1-z^2} = 0.$$

Thus $\mathbb{V}_{\mathbf{x}} = \mathbb{V}_{\mathbf{y}} = \frac{1}{2}$. Moreover, using $2\cos(x)\sin(x) = \sin(2x)$ we deduce that

$$\mathbb{C}_{\mathbf{x}, \mathbf{y}} = \frac{1}{2n} \sum_{k=0}^{n-1} \sin(\frac{4\pi k}{n}).$$

Since $\sum_{k=0}^{n-1} \sin(\frac{4\pi k}{n})$ is the imaginary part of

$$\sum_{k=0}^{n-1} (z^2)^k = \frac{1-z^{2n}}{1-z^2} = 0,$$

then $\mathbb{C}_{\mathbf{x}, \mathbf{y}} = 0$. ■

Note that in the case where the elements of $\mathcal{M}_n$ are the vertices of a regular n -sided polygon, Problem 5.1 has exactly n different solutions $\text{MBR}_i(\mathcal{M}_n)$, $i = 1, \dots, n$. For all $i = 1, \dots, n-1$, $\text{MBR}_i(\mathcal{M}_n)$ is the image of $\text{MBR}_0(\mathcal{M}_n)$ by the rotation of angle $\theta_i = \frac{i2\pi}{n}$.

2) *Handling of the exceptional cases 2 and 6:* Cases 2 and 6 are indeterminate cases, for which the minimum of the objective function $f(\theta)$ does not exist. This is why we have thought of a trick to get out of the indeterminacy of each case, by returning to the initial definition of the solution.

- For case 2, the application of an affine isometry to the set of points $\mathcal{M}_n$ makes it possible to migrate to another case for which the angle θ is well determined. We then obtain the solution of the initial problem by applying the inverse isometry to the solution of the transformed problem, since isometry preserves the areas.
- For case 6, the use of an extra point M_{n+1} belonging to the convex hull of the set $\mathcal{M}_n$, allows to migrate to another case for which the angle θ is well determined. Another alternative consists in asking the user to click on a point of the object's outline until we get out this state.

Case 2: $\lambda = (\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}})^2 - 4\mathbb{C}_{\mathbf{x}, \mathbf{y}}^2$ is negative because of $\mathbb{C}_{\mathbf{x}, \mathbf{y}}^2$. To get rid of $\mathbb{C}_{\mathbf{x}, \mathbf{y}}^2$, we have just to follow the same reasoning on a set of points

$$\mathcal{M}'_n = \{M'_i = (x'_i, y'_i)^t \in \mathbb{R}^2; i = 1, \dots, n\}$$

whose coordinates $\mathbf{x}'_{1:n} = (x_1, \dots, x_n)$ and $\mathbf{y}'_{1:n} = (y_1, \dots, y_n)$ are uncorrelated. Let

$$\Gamma = \frac{1}{n} \sum_{i=1}^n (M_i - \bar{M})(M_i - \bar{M})^t$$

be the covariance matrix of the elements of $\mathcal{M}_n$, where $\bar{M} = (\bar{x}, \bar{y})^t$. Let $\Gamma = U\Delta U^t$ be an eigendecomposition of Γ , where $U \in \mathcal{O}_2(\mathbb{R})$ (the set of 2-orthogonal matrices), and $\Delta = \text{diag}(\delta_1, \delta_2)$. For all $i = 1, \dots, n$, set $M'_i = U^t M_i$ and $\mathcal{M}'_n = (M'_i)_{1 \leq i \leq n}$. Then

$$\Gamma' = \frac{1}{n} \sum_{i=1}^n (M'_i - \bar{M}')(M'_i - \bar{M}')^t = \Delta$$

where $\bar{M}' = U^t \bar{M}$. Since $U \in \mathcal{O}_2(\mathbb{R})$, then the linear map defined by $L(M) = U^t M$ preserves the norm and the dot product. Thus, $\text{Area}(\text{MBR}(\mathcal{M}_n)) = \text{Area}(\text{MBR}(\mathcal{M}'_n))$ and $\text{MBR}(\mathcal{M}_n)$ is the image of $\text{MBR}(\mathcal{M}'_n)$ under the map L .

Proposition 5.2: If the vectors $\mathbf{x}_{1:n}$ and $\mathbf{y}_{1:n}$ are uncorrelated ($\mathbb{C}_{\mathbf{x}, \mathbf{y}} = 0$) and have different variances ($\mathbb{V}_{\mathbf{x}} \neq \mathbb{V}_{\mathbf{y}}$), then $\hat{A}_n^2(\theta)$ has a unique minimum at the angle $\theta = 0$.

Proof : Since $\mathbf{x}_{1:n}$ and $\mathbf{y}_{1:n}$ are uncorrelated then $\mathbb{C}_{\mathbf{x}, \mathbf{y}} = 0$, and $\lambda = (\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}})^2$ and $\delta = 0$. According to equation (14), the function f has a unique critical point $\theta^* = 0$. Using the Lemma 5.1, $f(0)$ is the maximum of $f(\theta)$. ■

Case 6: to get out of the indeterminacy of case 6, we propose to add to the set $\mathcal{M}_n$ an artificial point M_{n+1} located in the convex hull of $\mathcal{M}_n$, so that the empirical covariance matrix of the new set $\mathcal{M}_{n+1} = \mathcal{M}_n \cup M_{n+1}$ is different from a scalar matrix. Assume that $(x_{n+1}, y_{n+1})^t$ are the Cartesian coordinates of M_{n+1} with respect to $\mathcal{B}_G = (G, \mathbf{i}, \mathbf{j})$, where $G = (\bar{x}, \bar{y})^t$ is the barycenter of $\mathcal{M}_n$. Without any loss of generality, we assume that $\mathbf{x}_{1:n+1}$ (resp. $\mathbf{y}_{1:n+1}$) is the vector of the x -coordinates (resp. y -coordinates) of the points of $\mathcal{M}_{n+1}$ with respect to $\mathcal{B}_G = (G, \mathbf{i}, \mathbf{j})$. Denote by $\tilde{\mathbb{V}}_{\mathbf{x}} = \text{Var}(\mathbf{x}_{1:n+1})$ and $\tilde{\mathbb{V}}_{\mathbf{y}} = \text{Var}(\mathbf{y}_{1:n+1})$. Lemma 5.2 gives the relation between $\mathbb{V}_{\mathbf{x}}, \mathbb{V}_{\mathbf{y}}, \mathbb{C}_{\mathbf{x}, \mathbf{y}}$ and $\tilde{\mathbb{V}}_{\mathbf{x}}, \tilde{\mathbb{V}}_{\mathbf{y}}, \tilde{\mathbb{C}}_{\mathbf{x}, \mathbf{y}}$ respectively.

Lemma 5.2: If $\mathbb{V}_{\mathbf{x}} = \mathbb{V}_{\mathbf{y}}$ and $\bar{x} = \bar{y} = \mathbb{C}_{\mathbf{x}, \mathbf{y}} = 0$, then

$$\begin{aligned} \tilde{\mathbb{V}}_{\mathbf{x}} &= \frac{n}{n+1} \mathbb{V}_{\mathbf{x}} + \frac{n}{(n+1)^2} x_{n+1}^2 \\ \tilde{\mathbb{V}}_{\mathbf{y}} &= \frac{n}{n+1} \mathbb{V}_{\mathbf{y}} + \frac{n}{(n+1)^2} y_{n+1}^2 \\ \tilde{\mathbb{C}}_{\mathbf{x}, \mathbf{y}} &= \frac{n}{(n+1)^2} x_{n+1} y_{n+1} \end{aligned} \quad (19)$$

Proof : Set $\tilde{x} = \frac{1}{n+1} \sum_{i=1}^{n+1} x_i$ and $\tilde{y} = \frac{1}{n+1} \sum_{i=1}^{n+1} y_i$. Since $\bar{x} = \bar{y} = 0$, then $\tilde{x} = \frac{x_{n+1}}{n+1}$ and $\tilde{y} = \frac{y_{n+1}}{n+1}$. On the one hand, since $\bar{x} = 0$, we have:

$$\begin{aligned} \sum_{i=1}^{n+1} (x_i - \tilde{x})^2 &= \sum_{i=1}^n x_i^2 + x_{n+1}^2 - (n+1)\tilde{x}^2 \\ &= n\mathbb{V}_{\mathbf{x}} + \frac{n}{n+1} x_{n+1}^2 \end{aligned}$$

On the other hand, since $\sum_{i=1}^{n+1} x_i y_i = 0$, we have:

$$\begin{aligned} \sum_{i=1}^{n+1} (x_i - \tilde{x})(y_i - \tilde{y}) &= \sum_{i=1}^n x_i y_i - (n+1)\tilde{x}\tilde{y} \\ &= \left(\frac{n}{n+1}\right) x_{n+1} y_{n+1} \end{aligned}$$

Let $\tilde{f}(\theta)$ be the function defined from $\mathcal{M}_{n+1}$ in the same way that $f(\theta)$ was defined from $\mathcal{M}_n$ in (9). Then,

$$\tilde{f}(\theta) = \tilde{\lambda} \cos(\theta) + \tilde{\delta} \sin(\theta) \quad (20)$$

where,

$$\begin{aligned}\tilde{\lambda} &= (\tilde{V}_x^2 + \tilde{V}_y^2 - 2\tilde{V}_x\tilde{V}_y - 4\tilde{C}_{x,y}^2) \\ \tilde{\delta} &= 4\tilde{C}_{x,y}(\tilde{V}_x - \tilde{V}_y)\sin(4\theta)\end{aligned}\quad (21)$$

Using the identities (19), we have:

$$\tilde{\lambda} = \frac{n^2}{(n+1)^4}(x_{n+1}^4 + x_{n+1}^4 - 6x_{n+1}^2y_{n+1}^2) \quad (22)$$

It follows that:

- 1) $\tilde{\lambda} \neq 0$ i.f.f $M_{n+1} \notin D_1 \cup D_2 \cup D_3 \cup D_4$, where $D_1, \dots, D_4$ are the lines whose Cartesian equations are given by:

$$\begin{aligned}D_1 : x &= (1 + \sqrt{2})y & D_2 : x &= (1 - \sqrt{2})y \\ D_3 : x &= (\sqrt{2} - 1)y & D_4 : x &= (-1 - \sqrt{2})y\end{aligned}\quad (23)$$

- 2) $\tilde{\delta} \neq 0$ i.f.f $M_{n+1} \notin D_5 \cup D_6$, where $D_5 : x = y$ and $D_6 : x = -y$.

In summary, any point $M_{n+1} = (\bar{x} + x_{n+1}, \bar{y} + y_{n+1})^t$ defined as a convex combination of $M_1, \dots, M_n$, such that $x_{n+1}y_{n+1} \neq 0$ overcomes the indeterminacy problem posed by case 6. However, the resulting RANGE-MBR($\mathcal{M}_n$) depends on the values of x_{n+1} and y_{n+1} . We tested this technique on a set $\mathcal{M}_n$ composed of vertices of a regular n -sided polygon, and we observed that the choice

$$M_{n+1} = \frac{1}{2}(M_j + M_{\text{nod}(j+1,4)+1}) \quad (24)$$

allows our algorithm to reach MBR($\mathcal{M}_n$) whatever the value of n and $j = 1, \dots, n$. The generalization of the technique applied to the vertices of a regular polygon requires the calculation of CH($\mathcal{M}_n$), and the determination of the mid point, M_{n+1} , of any two consecutive vertices of CH($\mathcal{M}_n$), such that the product of the coordinates of M_{n+1} with respect to $B_G = (G, \mathbf{i}, \mathbf{j})$, is different from 0.

VI. ROBUST SEMI-AUTOMATIC ANNOTATION

In this section, we provide a new approach for building object detection datasets based on both MBR algorithm and threshold angles. This method is semi-automatic because it consists of a manual step followed by a computer-assisted step. In addition, this method is robust because the bounding rectangle generated by our algorithm is insensitive to the dexterity of the annotator.

- The problem of MBR was dealt with in Section V. If the user needs an optimal annotation in the sense given by Definition 7.1, then he calls the RC-MBR algorithm. Otherwise, he calls the RANGE-MBR algorithm for quasi-optimal annotation. However, optimality and complexity are inversely proportional.
- By default the threshold angle is equal to zero. It can also be adjusted by experts in object detection, or it can be defined experimentally as the largest angle between the ground truth bounding box and the positive x -axis that gives no significant difference between the performance of horizontal and rotated detectors when tested on rotated objects. The experimental determination of the threshold angle, require HOD and ROD object detectors, as well as the already existing ROD datasets.

A. Properties of the robust semi-automatic annotation method

By construction, the robust semi-automatic annotation method ensures the following properties:

- 1) The bounding rectangle generated by our approach is quasi-optimal in the sense that it covers the whole object, and its area is close to the area of the MBR enclosing the object.
- 2) The angle of the rectangle is determined by an algorithm based on some relevant points collected on the contour of the object.
- 3) The bounding rectangle is insensitive to annotators provided all relevant points on the object have been selected.
- 4) The determination of the bounding rectangle requires $\mathcal{O}(n)$ elementary operations, where n is the number of relevant points.
- 5) Allows the user to build simultaneously, from an image bank, two databases: one for horizontal objects and another for rotated objects.

In summary, the robust semi-automatic annotation provides a simple solution to all the drawbacks mentioned in paragraph III, which are inherent in the old annotation methods.

Let:

- θ_0 be a threshold angle fixed by the user (by default, $\theta_0 = 0$)
- $\mathcal{M}_n$ be the set of relevant points selected on the contour of the object,
- H-MBR($\mathcal{M}_n$) be the horizontal minimum bounding rectangle enclosing $\mathcal{M}_n$,
- R-MBR($\mathcal{M}_n$) be the minimum bounding rectangle enclosing $\mathcal{M}_n$,
- θ be the angle of R-MBR($\mathcal{M}_n$),

then, the Algorithm 2, and the flowchart shown in Figure 8 summarize the stages of construction of horizontal and rotated datasets, from a large-scale image bank.

Algorithm 2 Robust Semi-Automatic Annotation

Input: An image ; threshold angle ; HOD and ROD datasets

Output: Annotated image assigned to HOD or ROD

- 1: **for** (each object in the image) **do**
 - 2: Determine the class of the object
 - 3: Ask the annotator to click on the most relevant points ($\mathcal{M}_n$) of the object as defined in Definition 3.1
 - 4: Determine the RANGE-MBR $\mathbb{A} = [A_1, A_2, A_3, A_4]$ enclosing $\mathcal{M}_n$ as described in Section V-A
 - 5: Set α the angle between the positive x -axis and $\overrightarrow{A_1A_2}$
 - 6: **if** ($\alpha > \text{threshold angle}$) **then**
 - 7: Assign (image, object, $\mathbb{A}$, class) to the ROD dataset
 - 8: **else**
 - 9: Determine the horizontal MBR $\mathbb{A} = [A_1, A_2, A_3, A_4]$ enclosing the relevant points, as shown in Figure 7, by setting $\theta = 0$
 - 10: Assign (image, object, $\mathbb{A}$, class) to the HOD dataset
 - 11: **end if**
 - 12: **end for**
-

Fig. 8. Our approach consists in choosing between R-MBR($\mathcal{M}_n$) and H-MBR($\mathcal{M}_n$). If the angle of R-MBR($\mathcal{M}_n$) is less than or equal to a threshold angle, then we annotate the object using H-MBR($\mathcal{M}_n$) instead of R-MBR($\mathcal{M}_n$).

VII. EXPERIMENTAL STUDY

This experimental study was carried out exclusively with the MATLAB R2007b software. We implemented the RC-MBR, PCA-MBR, RANGE-MBR algorithms in Matlab language, and we wrote a script (see the Appendix) which:

- 1) reads the image then displays it
- 2) ask the annotator to click on the most relevant points of the object ($\mathcal{M}_n$ is a $2 \times n$ matrix)
- 3) determine the rectangle $\mathbb{A}$ corresponding to RC-MBR($\mathcal{M}_n$)
- 4) determine the rectangle $\mathbb{B}$ corresponding to RANGE-MBR($\mathcal{M}_n$)
- 5) draw the two rectangles with the colors red and green, respectively

The images used in experiments VII-B,...,VII-D are free of rights, and collected on the net. Moreover, the optimality criterion of an annotation is given in Definition 7.1.

Definition 7.1 (optimal annotation): A rectangle $\mathbb{A} = [A_1, \dots, A_4]$ enclosing an object is said to be optimal, if it fulfills the conditions (i) and (ii) below:

- (i) the rectangle $\mathbb{A}$ covers the whole object,
- (ii) the area of $\mathbb{A}$ is minimal.

A. Experiment 1

This experiment consists of studying the optimality condition (ii) of the RANGE-MBR algorithm. Equation (25) describes how to generate the coordinates of the vertices $\{M_n = (x_k, y_k); k = 1 \dots, n\}$ of a random n -sided polygon

Fig. 9. P_1 and P_2 are two random heptagons, generated according to Equation 25 with $v_y = 4$. P_1 is concave while P_2 is convex.

Fig. 10. The original object is the regular blue colored heptagon. The black, red, and green polygons correspond to the dilation of the blue polygon by a factor $v_y = 2, 3, 4$ respectively, in the vertical direction.

where each rand is call of a generator of uniform random numbers on the interval $[0, 1]$. The parameter v_y , called the factor of dilation, controls the aspect ratio of the polygon as shown in Figure 10. The more v_y is greater than 1, the more the polygon is dilated in the direction of the y -axis. Note that such a polygon is not necessarily convex as shown in Figure 9. This is also the case for any polygon whose vertices are defined by the relevant points of an object.

Since,

- 1) the optimal annotation criterion that we have chosen results in a rectangle which has a minimum area and which covers the maximum of visible parts of the object,
- 2) RC-MBR is the fastest algorithm that determines the smallest rectangle enclosing a set of points,

it seems natural to consider this algorithm as a reference in the comparative study that we carried out.

$$\begin{cases} x_k &= (1 + \text{rand}) \cos \left(\frac{2\pi k}{n} + \frac{2\pi}{n} \text{rand} \right) \\ y_k &= (1 + v_y \text{rand}) \sin \left(\frac{2\pi k}{n} + \frac{2\pi}{n} \text{rand} \right) \end{cases} \quad (25)$$

For each value of the pair (n, v_y) , we generate $r = 10000$ random n -sided polygons. For each polygon $\mathcal{M}_n^{(i)}$, $i = 1, \dots, r$, we determine the relative error e_i , between the areas of RANGE-MBR($\mathcal{M}_n^{(i)}$) and RC-MBR($\mathcal{M}_n^{(i)}$), as well as the CPU times $\hat{t}_i$ and t_i , used by the algorithms RANGE-MBR and RC-MBR to compute RANGE-MBR($\mathcal{M}_n^{(i)}$) and RC-MBR($\mathcal{M}_n^{(i)}$) respectively. Finally, we denote by $\bar{e}$, $\bar{t}$, $\bar{t}$, and

Fig. 11. Mean of $(e_i)_{1 \leq i \leq r}$ versus n and v_y .

Fig. 12. Standard deviation of $(e_i)_{1 \leq i \leq r}$ versus n and v_y .

$\text{Std}(e)$, the means and the standard deviation of the sequences $(e_i)_{1 \leq i \leq r}$, $(\hat{t}_i)_{1 \leq i \leq r}$, $(t_i)_{1 \leq i \leq r}$, and $(e_i)_{1 \leq i \leq r}$ respectively.

Figures 11 and 12 represent $\bar{e}$ and $\text{Std}(e)$ versus n and v_y for $n \in \{4, \dots, 20\}$ and $v_y \in \{1, \dots, 4\}$. We have retained the ranges of values $[4, 20]$ for n and $[1, 4]$ for v_y , because in our estimation, they are those which correspond most to reality. In practice, rare are objects that have more than 20 relevant points or less than 4 relevant points.

We deduce from Figures 11 and 12 that:

- 1) the more the polygon is dilated in one direction, the more the algorithm RANGE-MBR is accurate and precise
- 2) the more vertices the polygon has, the more the algorithm RANGE-MBR is accurate and precise

We ran other simulations with larger values of n and v_y , and we got the same previous conclusion. The parameter v_y controls the dilation of the polygon in the vertical direction. In fact, we could choose any other direction, since the RANGE-MBR algorithm is not sensitive to the direction of expansion. On the other hand, it is sensitive to the number of vertices of the polygon and to its dilation.

Out of the 680000 generated polygons, we do not encounter any case for which $\mathbb{V}_{\mathbf{x}} = \mathbb{V}_{\mathbf{y}}$ and $\mathbb{C}_{\mathbf{x}, \mathbf{y}} = 0$. In addition, The algorithm RANGE-MBR is 9 times faster than algorithm RC-MBR. The mean CPU times of RANGE-MBR and CR-MBR are both independent of n and v_y . It is about 1.38×10^{-4} seconds for the algorithm RANGE-MBR, and 1.20×10^{-3} seconds for the algorithm RC-MBR. Since the complexity of the algorithm RANGE-MBR is $\mathcal{O}(n)$, and that of the algorithm RC-MBR is $\mathcal{O}(n \ln(n))$, then the difference in CPU time, for each algorithm, is only observed if n is large enough. Based on the response time per click given in [35] and considering the mean CPU time of the RANGE-MBR algorithm, we can state that determining a bounding rectangle requires $2.5 + 1.5(n-1)$ seconds, where n is the number of relevant points collected

TABLE I
RELATIVE ERROR e OF PCA-MBR ON REGULAR n -SIDED POLYGON.

n	4	6	8	9	12	14	15	16	20
$100e$	100	6.6	17.2	0.8	0.7	1.3	0.3	4.0	2.5

Fig. 13. Illustration of the semi-automatic annotation using RC-MBR and RANGE-MBR algorithms.

on the contour of the image.

B. Experiment 2

We made a comparison between RANGE-MBR and PCA-MBR based on sets of vertices of regular n -sided polygons, with $n = 4, \dots, 20$. We observe that the relative error of the algorithm RANGE-MBR is always equal to 0, while that of the algorithm PCA-MBR is different from 0 for the values of n reported in Table I. Although case 6 does not contain only regular polygons, this experiment shows that the RANGE-MBR method achieves optimality on regular polygons, but for the other shapes (which go in case 6) it offers a better solution than that obtained by PCA-MBR.

C. Experiment 3

This experiment consists in studying the effect of the dilation factor on the performance of the RANGE-MBR algorithm. Figure 13 represents a basic experiment. The relevant points $\mathcal{M}_n$ are colored in yellow. We observe that $\text{RANGE-MBR}(\mathcal{M}_n)$ is close to $\text{RC-MBR}(\mathcal{M}_n)$. Although the bird on the left is smaller than the bird on the right, the relative error e equals 0.65% for the bird on the right, and 3.12% for the bird on the left. These scores are expected, since the bird on the right has a more elongated shape than the bird on the left, and RANGE-MBR is more effective on dilated objects.

D. Experiment 4

This experiment consists in confirming the conclusion obtained in the experiment VII-D. Figure 14 contains two mangoes. The one on the left is almost circular, while the one on the right is clearly elongated. The relevant points $\mathcal{M}_n$ are colored in yellow. The red rectangle corresponds to $\text{RANGE-MBR}(\mathcal{M}_n)$, while the green rectangle corresponds to $\text{RC-MBR}(\mathcal{M}_n)$. The relative error e equals 6.93% for the left mango, and 1.86% for the right mango. This real example agrees with the simulation results shown in Figure 11. The

Fig. 14. The relevant points $\mathcal{M}_n$ are colored in yellow. The red rectangle corresponds to RANGE-MBR($\mathcal{M}_n$), while the green rectangle corresponds to RC-MBR($\mathcal{M}_n$).

Fig. 15. The relevant points $\mathcal{M}_n$ are colored in yellow. The red rectangle corresponds to RANGE-MBR($\mathcal{M}_n$), while the green rectangle corresponds to RC-MBR($\mathcal{M}_n$).

mango on the right is more dilated than the one on the left, that is why the relative error for the circular mango is greater than relative error for the oval mango.

E. Experiment 5

In order to verify the robustness of our annotation method with respect to the annotator, we asked a colleague to click on the relevant points of the two mangoes. Figure 15 illustrates the result of this experiment. The relative error e equals 8.92% for the left mango, and 2.32% for the right mango. The relative errors reported in Figure 15 are larger than those reported in Figure 14. We explain this difference by the number of points used in each experiment: the more points we use, the more the RANGE-MBR algorithm reduces the relative error. This real example agrees with the simulation results shown in Figure 11.

F. Experiment 6

This experiment consists in comparing our annotation method to the FBB method introduced in Section II-B. For this, we have chosen a random image from the TUT indoor dataset, and we have annotated the objects it contains with FBB, RC-MBR, and RANGE-MBR. Note that the FBB annotation uses blue rectangles, RC-MBR annotation uses green rectangles, RANGE-MBR annotation uses red rectangles, and the relevant points are marked in yellow.

Based on Figure 16 and Table II, it can be seen that the annotation by the FBB method is not optimal. Indeed,

TABLE II
AREAS OF BOUNDING RECTANGLES IN FIGURE 16

Method	FBB	RANGE-MBR	RC-MBR
Object			
Upper fire extinguisher	7227	8295	8046
Lower extinguisher	21726	19917	19844
Exit sign	4902	6927	6146

Fig. 16. Annotation of an image with FBB, RANGE-MBR, and RC-MBR.

- for the upper extinguisher and the exit sign, the condition (i) is violated.
- for the lower extinguisher, the condition (ii) is violated.

In addition, the annotation by the RANGE-MBR method satisfies condition (i) and the relative error between the area of RANGE-MBR and RC-MBR is 12% for the exit sign, 3% for the upper extinguisher, and 0.3% for the lower extinguisher. We can conclude that condition (ii) is almost satisfied by the RANGE-MBR method.

- The values of the relative error are in agreement with the simulation results presented in Figure 11. Indeed, in terms of dilation, the lower extinguisher is the most dilated, followed by the upper extinguisher, then the exit sign.
- The exit sign and the upper extinguisher have regular shapes, while the lower extinguisher has an irregular shape. We have already underlined in Section V-A1, that the RANGE-MBR method is sensitive to regular forms. This example then illustrates the situation of case 6.

For all these reasons, the annotation of the lower extinguisher by the RANGE-MBR method is the best.

G. Experiment 7

This experiment highlights the importance of the annotation method on the calculation of the IoU. Figure 17 corresponds to image P_{1128} from the DOTA dataset. The objects of interest in these images are the airplanes. We use the BBAVectors detector [25] to generate the red prediction bounding boxes. Ground truth bounding boxes used in [25] are colored green. The blue rectangles correspond to the annotations of the airplanes by the RANGE-MBR method. For all $i = 1, 2, 3$, we denote by IoU_i^g (resp. IoU_i^b) the Intersection over Union between the green (resp. blue) rectangle enclosing the airplane A_i , and the corresponding red rectangle. The results of this experiment are reported in Table III. The use of minimum ground truth rectangles best reflects the performance of an object detector.

In view of the results of Table III, it is reasonable to rely on the results of the second row rather than those of the first row.

TABLE III
IOU COMPUTATION FOR THE OBJECTS IN FIGURE 17.

Airplane	1	2	3
IoU_i^g	0.5230	0.5114	0.7252
IoU_i^b	0.6222	0.6186	0.7087

Fig. 17. RANGE-MBR allows to build quasi-minimum bounding rectangles.

VIII. CONCLUSION

In this article, we have developed a new robust and semi-automatic object annotation method. Based on the experimental study, we can state that Robust Semi-Automatic Annotation,

- 1) is quasi-optimal in the sense given by Definition 7.1, and that its optimality increases with dilated objects, which is the case for most objects appearing on images collected by aerial photography,
- 2) is fast and robust in the sense that the bounding rectangle is insensitive to the dexterity of the annotator,
- 3) is easy to implement and could be easily integrated into annotation platforms,
- 4) is sensitive to the annotation of objects exhibiting symmetry, with respect to one or more directions. In this case, the relevant points should not follow the same symmetry so that the generated rectangle is not far from the optimal rectangle,

The threshold angle as defined in Section VI, could be the subject of an experimental study based on a large-scale rotated object dataset and advanced horizontal and rotated object detectors. Once the threshold angle is well estimated, we can use Robust Semi-Automatic Annotation algorithm to simultaneously build, from an image bank, two datasets: one for horizontal objects and another for rotated objects.

IX. DECLARATIONS

- 1) The author declares that they have no known competing financial interests or personal relationships that could

have appeared to influence the work reported in this paper.

- 2) The author confirms having obtained the agreement of the scientific committee of Qassim University, to carry out all the experiments of the experimental study.
- 3) The Author confirms that all the experiments were carried out with relevant guidelines and regulations.
- 4) The author states that the scientific committee of Qassim University approves all the experiments in this section.
- 5) The author confirms that informed consent was obtained from all subjects.
- 6) The author assumes his responsibility in all his statements.

APPENDIX

SCRIPT. 1: COLLECT $\mathcal{M}_n$, DRAW RC-MBR($\mathcal{M}_n$) AND RANGE-MBR($\mathcal{M}_n$)

Here, we assume that we have already implemented the RANGE-MBR and RC-MBR functions, in the MATLAB language.

```

1 % read the image
2 photo=imread('mango.jpg');
3 % display the image
4 image(photo)
5 button=1;
6 X=[];
7 Y=[];
8 % read ginputs until a mouse right-button occurs
9 while(sum(button)<=1)
10     [x,y,button]=ginput(1);
11     X=[X x];
12     Y=[Y y];
13 end
14 n=length(X);
15 Mn=[X;Y];
16 % A is a (2,4)-matrix containing the coordinates of
17   RC-MBR(Mn)
18 A=RC_MBR(Mn);
19 A=[B B(:,1)];
20 % B is a (2,4)-matrix containing the coordinates of
21   RANGE-MBR(Mn)
22 B=RANGE_MBR(Mn);
23 B=[B B(:,1)];
24 hold on
25 % draw the points of Mn in yellow
26 plot(M(1,:), M(2,:), '*y')
27 % draw rectangle A in green
28 plot(A(1,:), A(2,:), '-g')
29 % draw rectangle B in red
30 plot(B(1,:), B(2,:), '-r')
31 axis equal

```

REFERENCES

- [1] A. Endres *et al*, "The benefits and challenges of collecting richer object annotations," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2010.
- [2] K. Su and X. Geng, "Soft facial landmark detection by label distribution learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5008–5015, 2019.
- [3] Z. Zou *et al*, "Object detection in 20 years: A survey," *ArXiv*, vol. abs/1905.05055, 2019.
- [4] S. Ren *et al*, "Faster r-cnn: Towards realtime object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 11371149, 2017.
- [5] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2018.

- [6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.
- [7] W. Liu *et al*, "Ssd: Single shot multibox detector," *European conference on computer vision*, pp. 21–37, 2016.
- [8] W. Qian *et al*, "Learning modulated loss for rotated object detection," *ArXiv*, vol. 1911.08299, 2019.
- [9] H. Zhu *et al*, "Orientation robust object detection in aerial images using deep convolutional neural network," *IEEE International Conference on Image Processing*, vol. 13, pp. 3735–3739, 2015.
- [10] Z. Liu *et al*, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, pp. 1074–1078, 2016.
- [11] G. Xia *et al*, "Dota: A large-scale dataset for object detection in aerial images," *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [12] S. Azimi *et al*, "Towards multi-class object detection in unconstrained remote sensing imagery," *Asian Conference on Computer Vision*, pp. 150–165, 2018.
- [13] J. Ding *et al*, "Learning roi transformer for oriented object detection in aerial images," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2844–2853, 2019.
- [14] X. Yang *et al*, "Towards more robust detection for small, cluttered and rotated objects," *IEEE International Conference on Computer Vision*, pp. 8232–8241, 2019.
- [15] X. Zahou *et al*, "East: an efficient and accurate scene text detector," *IEEE conference on computer vision and pattern recognition*, pp. 5551–5560, 2017.
- [16] X. Yang *et al*, "R3det: Refined single-stage detector with feature refinement for rotating object," *ArXiv*, vol. abs/1908.05612, 2019.
- [17] M. Everingham *et al*, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, p. 98136, 2015.
- [18] T. Lin *et al*, "Microsoft coco: common objects in context," *European Conference on Computer Vision*, 2014.
- [19] O. Russakovsky *et al*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.
- [20] B. Adhikari *et al*, "Faster bounding box annotation for object detection in indoor scenes," *European Workshop on Visual Information Processing*, pp. 1–6, 2018.
- [21] H. Rezatofighi *et al*, "Generalized intersection over union: A metric and a loss for bounding box regression," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.
- [22] L. Liu, Z. Pan, and B. Lei, "Learning a rotation invariant detector with rotatable bounding box," *ArXiv*, vol. abs/1711.09405, 2017.
- [23] C. Yao *et al*, "Detecting texts of arbitrary orientations in natural images," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, 2012.
- [24] A. Vahab *et al*, "Applications of object detection system," *International Research Journal of Engineering and Technology*, vol. 06, no. 04, pp. 4186–4192, 2019.
- [25] R. Padilla *et al*, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 03, 2021.
- [26] D. Lo, *Finite Element Mesh Generation*, 1st ed. CRC Press, 2014.
- [27] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," *Communication of the ACM*, vol. 18, no. 7, pp. 409–413, 1975.
- [28] M. Shamos, "Computational geometry," *Ph.D. thesis, Yale University*, 1978.
- [29] H. Toussaint, "Solving geometric problems with the rotating calipers," *Proceedings of IEEE MELECON*, pp. A10.02/1–4, 1983.
- [30] D. Dimitrov, C. Knauer, K. Kriegel, and G. Rote, "On the bounding boxes obtained by principal component analysis," *European Workshop on Computational Geometry*, pp. 193–1196, 2006.
- [31] D. Dimitrov *et al*, "Bounds on the quality of the pca bounding boxes," *Computational Geometry*, vol. 42, no. 8, pp. 772–789, 2009.
- [32] A. Ramirez and C. Cox, "Improving on the range rule of thumb," *Rose Hulman Undergraduate Mathematics Journal*, vol. 13, no. 2, 2012.
- [33] S. Hozo *et al*, "Estimating the mean and variance from the median, range, and the size of a sample," *BMC Medical Research Methodology*, vol. 5, no. 13, 2005.
- [34] X. Wan *et al*, "Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range," *BMC Medical Research Methodology*, vol. 135, no. 14, 2014.
- [35] D. Papadopoulos *et al*, "Extreme clicking for efficient object annotation," *IEEE International Conference on Computer Vision*, pp. 4940–4949, 2017.