Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 3001939, 16 pages
https://doi.org/10.1155/2022/3001939

Research Article

Mathematical Methods for IoT-Based Annotating Object Datasets with Bounding Boxes

Abdelhamid Zaidi

Department of Mathematics, College of Science, Qassim University, P. O. Box 6644, Buraydah 51452, Saudi Arabia

Correspondence should be addressed to Abdelhamid Zaidi; a.zaidi@qu.edu.sa

Received 21 May 2022; Revised 29 June 2022; Accepted 11 July 2022; Published 23 August 2022

Academic Editor: Savita Gupta

Object datasets used in the construction of object detectors are typically annotated with horizontal or oriented bounding rectangles for IoT-based. The optimality of an annotation is obtained by fulfilling two conditions: (i) the rectangle covers the whole object and (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 for IoT-based, it is easy for the annotator to reach the optimal bounding box within a reasonable time. However, if the object is oriented, the annotator needs additional time to decide whether the object will be annotated with a horizontal rectangle or an oriented rectangle for IoT-based. 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 for IoT-based, 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.

1. Introduction

The construction of an object detector generally goes through a learning phase, followed by a testing phase, and ends with a tuning phase. Each phase requires an independent annotated object dataset. Annotating an image IoT-based signal and image processing applications consists of locating all the objects present in this image and determining their categories. The way to locate an object varies from detector to detector. For example, mask R-CNN detector uses segmentation mask to locate objects [1], CoKe detector uses key points and landmarks to locate objects [2], and poly-YOLO detector represents objects using polygons [3]. However, the rectangle is considered to be the simplest and most used polygonal shape for locating or representing an object in many computer vision applications for IoT-based [4]. Locating an object using a rectangle consists of drawing a rectangle surrounding this object. The annotation of an object dataset is performed using free or commercial software designed for a particular annotation choice.

There are two types of annotations with rectangles depending on the orientations of the objects in the images for IoT-based. To locate objects, the first type uses horizontal bounding rectangles (HBR), while the second type uses oriented bounding rectangles (OBR). Annotation with horizontal rectangles is suitable for natural scenes, where the photographer is usually in front of the object and adjusts their camera so that the objects appear aligned with the horizontal edges of the image. By contrast, for aerial photography, where images are captured by Earth observation satellites or other flying devices, objects often appear in the image with arbitrary orientations.

The optimality of an annotation results in the satisfaction of two conditions: (i) the bounding rectangle must cover the whole of the object and (ii) the area of the bounding rectangle must be minimal. For a horizontal object, it is easy to reach the optimal HBR that satisfies conditions (i) and (ii) mentioned above. On the other hand, for an oriented object, it is difficult to reach the optimal OBR if the annotation method does not render account of optimality conditions (i) and (ii).

In the literature, we distinguish two types of object detectors. The first type includes horizontal object detectors, which detect objects using HBR. The second type includes oriented object detectors, which detect objects using OBR. Any detector of the first class is trained, tested, and tuned on horizontal object datasets (HOD). On the other hand, some oriented object detectors are trained and tuned on HOD, such as OAOD [5] and BBAVectors [6], but others are trained tuned on to oriented object datasets (OOD), such as RoI transformer [7], R-RoI [8], RRPN [9], R2CNN++ [10], DMPNet [11], FOTS [12], RPN [13], and DDR [14]. However, all of these oriented object detectors must be tested on OOD.

A large-scale HOD (OOD, respectively) is a set made up of a large number of images annotated with HBR (OBR, respectively). Each image contains objects of a wide variety of scales, orientations, and shapes. These objects are divided into classes (or categories) that vary from one dataset to another.

Table 1 (Table 2, respectively) presents the number of classes, instances, images, and year of creation of the most cited HOD (OOD, respectively) in the literature. To our knowledge, DOTA is the largest public Earth vision object detection dataset [27]. It contains objects exhibiting a wide variety of scales, orientations, and shapes. Moreover, images of DOTA are manually annotated by experts in aerial image interpretation.

Table 1: Examples of large-scale horizontal object datasets.

Name	Classes	Instances	Images	Year
DIOR [15]	20	190288	23463	2019
TUT indoor [16]	7	4595	22137	2018
MS COCO [17]	80	886266	123287	2014
PASCAL VOC [18]	17	1793658	11268	2012
ImageNet [19]	200	478807	456567	2009

Table 2: Examples of large-scale oriented object datasets.

Name	Classes	Instances	Images	Year
DOTA-v2.0 [20]	17	1793658	11268	2021
FGSD [21]	1	5634	2612	2020
HRSC2016 [22]	1	2976	1061	2016
UCAS-AOD [23]	2	14596	1510	2015
DLR 3k [24]	2	14235	20	2015
VEDAI [25]	3	2950	1268	2015
SZTAKI-INRIA [26]	1	665	9	2012

2. State of the Art

In order to generate a dataset for object identification, we need to gather a large number of photographs, all of which must be annotated using the same technique and organized in accordance with a set of categories that have been predetermined. The dataset has to include many photographs that match to various instances of each object class. These images must be included for each object class. The currently available annotation techniques are amenable to being split up into two primary categories. In the first class, all methods of manual annotation are grouped together, and in the second class, all techniques of semi-automatic annotation are categorized together [28].

2.1. 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 and 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 investigates whether additional object class instances require annotation. Determining a precise bounding rectangle takes more time and resources than validating the annotation, and thus, the consensus technique is more efficient. Annotation quality affects the development of accurate 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 [18], MS COCO [17], ImageNet [19], DOTA [20], etc.

Although studies on the description of OOD are abundant, very few of them explain how to draw a rectangle enclosing an oriented object. The only annotation method explicitly described in the literature appears in the article [27] by Ding et al. This method consists in drawing a HBR around the object of interest and then adjusting the angle manually by rotating the rectangle with the mouse.

2.2. 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 [16]. 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 [29], trained on the smallest annotated subset.

2.3. 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 $\widehat{\mathbb{A}}$ generated by the detector, and the ground truth bounding box $\mathbb{A}$, as follows:

$$IoU\left(\widehat{\mathbb{A}}, \mathbb{A}\right) = \frac{\text{Area}\left(\widehat{\mathbb{A}} \cap \mathbb{A}\right)}{\text{Area}\left(\widehat{\mathbb{A}} \cup \mathbb{A}\right)}. \tag{1}$$

We 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 [30]. 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. [30] suggest replacing the IoU with the Generalized Intersection over Union (GIoU) index. The GIoU between two rectangles $\widehat{\mathbb{A}}$ and $\mathbb{A}$ is defined as follows:

$$\text{GIoU}\left(\widehat{\mathbb{A}}, \mathbb{A}\right) = IoU\left(\widehat{\mathbb{A}}, \mathbb{A}\right) - \frac{\text{Area}\left(\mathbb{C} \backslash \left(\widehat{\mathbb{A}} \cup \mathbb{A}\right)\right)}{\text{Area}(\mathbb{C})}, \tag{2}$$

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

The exact IoU computation between two HBR is simple. Much software integrates functions to calculate this index. On the other hand, the exact IoU computation between two OBR is not as simple as that between two HBR. Liu et al. (Yao et al., respectively) proposed in Ref. [31] (in [32]) heuristic to estimate the IoU between two OBR. Recently, Zaidi has shown in Ref. [28] that these heuristics give reliable results only when the centers of the rectangles are very close and the angle between the rectangles is small. Moreover, he has developed in Ref. [28] an algorithm that calculates the exact IoU value between two OBR, as well as an (ε, α) estimator of the IoU.

The annotation method has a direct bearing on the performance measurement of an object detector. The OOD reserved for the test phase must be carefully annotated so that the IoU computation between the ground truth and prediction rectangles is not biased.

3. Motivation for the Study

The manual or semi-automatic annotation methods described in Section 2 have made it possible to build many large-scale object datasets. These datasets have given rise to very powerful object detectors. However, we cannot ignore the following concerns:

(1) According to paragraph 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 1. In this case, he will take a considerable time to decide whether the object will be annotated with a horizontal rectangle or an oriented rectangle. Moreover, in both cases, the final decision is not based on any objective argument and depends solely on the dexterity of the annotator.

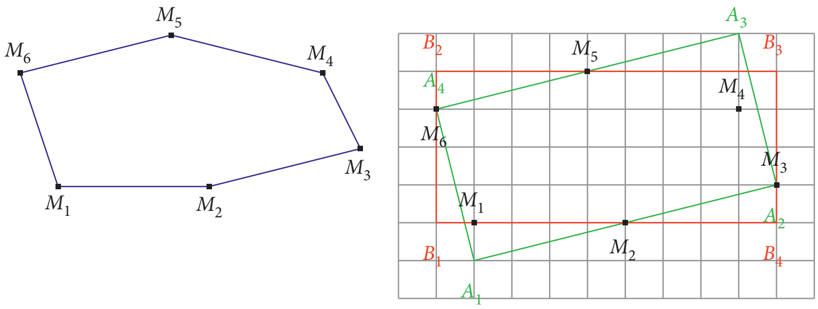
(3) The semi-automatic annotation defined in paragraph B does not guarantee the optimality of the annotation rectangle, in the sense that it must have a minimum area and cover the whole of the object.

(4) According to paragraph C, the use of any approximation methods of $\text{IoU}(\mathbb{A}, \mathbb{B})$, whether to measure the accuracy of an oriented object detector or to compare the performance of two oriented object detectors, could lead to biased results. Indeed, the function $g(x) = x/a + b - x$ is strictly increasing over the interval $[0, a + b[$, where a and b denote the areas of $\mathbb{A}$ and $\mathbb{B}$, respectively. Therefore, a large or a small ground truth bounding rectangle directly induces a bias in the computation of the IoU.

The contribution of this study is the scarcity of annotation methods motivated us to develop a robust semi-automatic annotation method of OOD. Our 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. More precisely, the steps are as follows:

(i) We develop an algorithm called RANGE-MBR, which determines from a set $\mathscr{M}_n$ of the n most relevant (in the sense given by Definition 1) points picked on the object outline, a rectangle enclosing $\mathscr{M}_n$ and having a quasi-minimal area, in $\mathcal{O}(n)$ time.

(ii) We propose a new approach to simultaneously build HOD and OOD from a large-scale image



(a)

(b)

Figure 1: (b) Intuitively, it seems that the horizontal rectangle $\mathbb{B} = [B_1, \ldots, B_4]$ is best suited to annotate the colored object in blue (Figure (a)). The method we propose takes as input the set $\mathscr{M}_6 = \{M_1, \ldots, M_6\}$ and returns the oriented rectangle $\mathbb{A} = [A_1, \ldots, A_4]$. 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 $\mathbb{A}$.

bank, based on both the RANGE-MBR algorithm and threshold angles.

(iii) We conduct a large experimental study to quantify the performance of the RANGE-MBR algorithm.

(iv) We compare the performance of RANGE-MBR to that of the benchmark algorithm, RC-MBR, which determines the minimum rectangle enclosing $\mathscr{M}_n$ in $\mathcal{O}(n \ln(n))$ time.

This study seems essential to us because it responds to a real need in the development of oriented object detectors. In addition, it allows researchers to build their own oriented object datasets in a rigorous manner.

Definition 1. (relevant point of an object) Any point located on the contour of the object is said to be relevant 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 4 deals with the parametrization of rectangles. Section 5 is devoted to the development of the RANGE-MBR algorithm. We explain in Section 6 how to use the RANGE-MBR (or RC-MBR) algorithm to simultaneously generate horizontal and oriented datasets from a large-scale image bank. In Section 7, we perform various numerical experiments to evaluate the performance of the RANGE-MBR algorithm.

4. Parametrization of Bounding Rectangles

First, we give an overview of the different types of parameterization encountered in the literature. Then, we justify the choice of the parameterization associated with the annotation method that we propose in this study. Our choice of parameterization was discussed in a previous study on the accurate computation of the IoU [28]. However, it seems useful to us to recall this choice to facilitate the reading of the manuscript.

4.1. 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 [18] 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, $i = 1, \ldots, 4$ (cf. Figure 2). The object dataset Ms COCO [17] 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 2). 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 2).

4.2. Parameterization of Oriented Rectangles. Five parameters—b_1, b_2, w, h, and θ—are necessary for the minimal parameterization of an orientated rectangle denoted by the formula $\mathbb{B} = \{B_1, B_2, B_3, B_4\}$; these are the only parameters that are required. The coordinates for the center B_0 of $\mathbb{B}$ are (b_1, b_2), which correspond to the coupling b_1 and b_2. The lengths of the line segments $[B_1, B_2]$ and $[B_1, B_4]$ are indicated by the parameters w and h, respectively (cf. Figure 2). The parameter $\theta \in [0, \pi/2]$ is used to determine the acute angle that exists between the lines Δ and (B_1, B_2), where Δ denotes the horizontal line that goes through B_1. These two object datasets, HRSC2016 [22] and UCAS-AOD [23], are annotated with the help of this parametrization.

The above θ-based parametrization is not suitable for annotating satellite image datasets. These OOD have the particularity of containing a large number of instances per image, at different scales, and whose parts overlap [20]. A simple solution that overcomes the drawbacks of the θ-based annotation consists in defining a rectangle $\mathbb{B} = \{B_1, B_2, B_3, B_4\}$ using the coordinates $(x_i, y_i)_{1 \leq 4}$ of the vertices $B_1, \ldots, B_4$, respectively. To solve the indeterminacy problem linked to the permutations of the vertices, we can sort the vertices clockwise and fix the first point according to the following rules [27]:

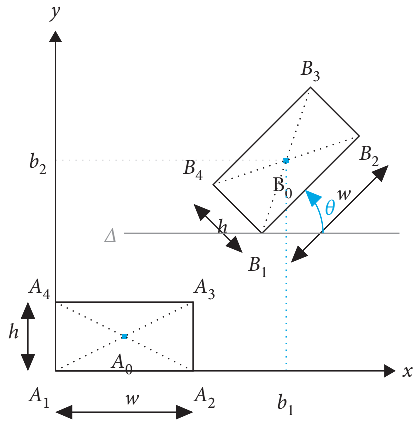


Figure 2: Parameterization of horizontal and oriented rectangles.

(i) For objects with a distinguished head and a tail (e.g., vehicles and helicopters), the annotator carefully selects B_1 to indicate the left corner of the instance head.

(ii) For other objects (e.g., tennis courts and bridges), B_1 is the point at the top left of the instance.

This type of parametrization was used for the annotation of the DOTA object dataset [20].

4.3. Our Choice of Parametrization. We use the notation $\mathbb{R} = [R_1, R_2, R_3, R_4]$ to denote an oriented 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 $\{B_1, B_2, B_3, B_4\}$ of the rectangle $\mathbb{B}$ as described above, we first consider the barycenter G of the rectangle $\mathbb{B}$. Then, we calculate the polar angle of each vertex relative to the barycenter G [28]. We denote by $B_{(1)}, \ldots, B_{(4)}$, the sorted vertices in ascending order of their polar angle. If $B_{(3)}(2) \leq B_{(4)}(2)$, then $\mathbb{B} = [B_{(3)}, B_{(4)}, B_{(1)}, B_{(2)}]$, else $\mathbb{B} = [B_{(4)}, B_{(1)}, B_{(2)}, B_{(3)}]$ (cf. Figure 2). For more details on the implementation of this sorting method, we refer the reader to our article [28].

In this investigation, we describe a bounding rectangle $\mathbb{B}$ by using the eight-tuple $(x_1, y_1, \ldots, x_4, y_4)$, where $(x_i, y_i)^t$ are the coordinates of the i-th vertex of the sorted rectangle $[B_1, \ldots, B_4]$.

This choice of parameterization is ideally suited to the annotation approach that we suggest since it eliminates the need for extra computations to draw rectangles or to locate the images of rectangles by applying affine transformations. This is one of the reasons why the method is so efficient. In addition, we demonstrate in Section B how to format the outputs of the annotation technique that we propose in accordance with the annotation rule of the DOTA dataset. This is done by referring to the annotation rule.

5. 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 1. Given a set of points $\mathcal{M}_n = \{M_i = (x_i, y_i)^t \in \mathbb{R}^2; i = 1, \ldots, n\}$, we 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 MBR $(\mathcal{M}_n)$.

The solution of problem 5.1 is not unique as shown in Figure 3. So, MBR $(\mathcal{M}_n)$ denotes any solution of Problem 1.

Freeman and Shapira proved in 1975 that one edge of MBR $(\mathcal{M}_n)$ must be collinear with an edge of the convex hull (CH $(\mathcal{M}_n)$) of $\mathcal{M}_n$. They proposed in Ref. [33] a natural algorithm to find 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 CH $(\mathcal{M}_n)$. In 1978, Shamos proposed in Ref. [34] 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 MBR $(\mathcal{M}_n)$ in $\mathcal{O}(n \ln(n))$ time [35]. In 2006, Dimitrov et al. proposed in Ref. [36, 37] the algorithm PCA-MBR, to approximate 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 CH $(\mathcal{M}_n)$. They also proved that the relative error between PCA-MBR $(\mathcal{M}_n)$ and 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 CH $(\mathcal{M}_n)$ has a double eigenvalue. To overcome the problem of nonuniqueness inherent in algorithms RC-MBR and PCA-MBR, we propose the method RANGE-MBR to approximate MBR $(\mathcal{M}_n)$ in $\mathcal{O}(n)$ time.

5.1. 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 (P_2 and Q_2, respectively) be the extreme points of the orthogonal projection of $\mathcal{M}_n$ on Δ ($\Delta^\perp$, respectively). Let $\mathbb{A} = [A_1, A_2, A_3, A_4]$ be the minimum bounding rectangle having an edge collinear with Δ. Then, $A_1, \ldots, A_4$ 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 $MBR(\mathcal{M}_n, \theta)$.

Let $\mathbf{u} = (\cos(\theta), \sin(\theta))^t$, and $\mathbf{v} = (-\sin(\theta), \cos(\theta))^t$) be two unit direction vector of Δ and $\Delta^\perp$, respectively. 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 $i = 1, \ldots, n$, we denote by $(x_i, y_i)^t$ ($(s_i, t_i)^t$, respectively) the coordinates of M_i with respect to the frame $\mathcal{B}$ ($\mathcal{B}'$, respectively). Let R_θ be the rotation matrix of angle θ, and 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 $\mathbf{x}_{1:n} = (x_1, \ldots, x_n)$ be n observations on the

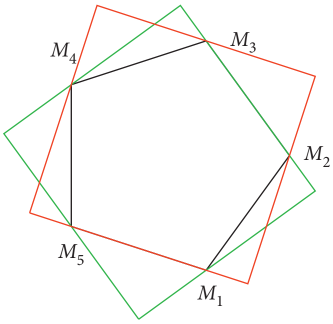


Figure 3: $\mathscr{M}_5 = \{M_1, \dots, M_5\}$. The green and red rectangles are two distinct solutions of Problem 1.

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:

$$\begin{aligned} P_1Q_1 &= \text{Range}(\{s_1, \dots, s_n\}) = s_{(n)} - s_{(1)}, \\ P_2Q_2 &= \text{Range}(\{t_1, \dots, t_n\}) = t_{(n)} - t_{(1)}. \end{aligned} \tag{3}$$

We denote by $\widehat{\sigma}_n$ 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 ([38–43]), we can assume that there is a real constant $\alpha(X, n)$ such that

$$\widehat{\sigma}_n = \frac{\text{Range}(\mathbf{x}_{1:n})}{\alpha(X, n)}. \tag{4}$$

Besides, it is well known that

$$\text{Var}(\mathbf{x}_{1:n}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x}), \tag{5}$$

which is an estimate of σ^2, where $\overline{x} = 1/n \sum_{i=1}^{n} x_i$. It follows that there is a real constant $\alpha(X, n)$ such that:

$$\text{Range}(\mathbf{x}_{1:n})^2 \approx \alpha(X, n)^2 \text{Var}(\mathbf{x}_{1:n}). \tag{6}$$

We assume that the (x, y) coordinates of the set of points $\mathscr{M}_n$ are n observations of two random variables X and Y. Since the $\text{Area}(\mathbb{A}) = P_1Q_1 \cdot P_2Q_2$, then combining equations (6) and (3) gives an approximation $\widehat{A}_n^2(\theta)$ of $\text{Area}(\mathbb{A})$, satisfying the following relation:

$$\widehat{A}_n^2(\theta) = \alpha(S, n)^2 \alpha(T, n)^2 \text{Var}(\mathbf{s}_{1:n}) \text{Var}(\mathbf{t}_{1:n}), \tag{7}$$

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

$$\begin{aligned} \mathbb{V}_s &= \cos^2(\theta)\mathbb{V}_x + \sin^2(\theta)\mathbb{V}_y + \sin(2\theta)\mathbb{C}_{x,y}, \\ \mathbb{V}_t &= \sin^2(\theta)\mathbb{V}_x + \cos^2(\theta)\mathbb{V}_y - \sin(2\theta)\mathbb{C}_{x,y}. \end{aligned} \tag{8}$$

Using (8) and some trigonometric identities, we prove that

$$8\mathbb{V}_{\mathbf{s}}\mathbb{V}_{\mathbf{t}} = K - f(\theta), \tag{9}$$

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

$$\begin{aligned} f(\theta) = &\left(\mathbb{V}_{\mathbf{x}}^2 + \mathbb{V}_{\mathbf{y}}^2 - 2\mathbb{V}_{\mathbf{x}}\mathbb{V}_{\mathbf{y}} - 4\mathbb{C}_{\mathbf{x,y}}^2\right)\cos(4\theta) \\ &+ 4\mathbb{C}_{\mathbf{x,y}}\left(\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}}\right)\sin(4\theta). \end{aligned} \tag{10}$$

Combining equations (7), (9), and (10) gives

$$\widehat{A}_n^2(\theta) = \frac{\alpha(S, n)^2 \alpha(T, n)^2}{8} (K - f(\theta)). \tag{11}$$

Thus, the area of MBR $(\mathscr{M}_n, \theta)$ is approximately equal to

$$\frac{\alpha(S, n)^2 \alpha(T, n)^2}{8} (K - f(\theta)). \tag{12}$$

Therefore, we propose to approximate MBR $(\mathscr{M}_n)$ by MBR $(\mathscr{M}_n, \theta^*)$ such that:

$$\theta^* = \underset{\theta \in [-(\pi/2), \pi/2]}{\arg\min} \widehat{A}_n^2(\theta) = \underset{\theta \in [-(\pi/2), \pi/2]}{\arg\max} f(\theta). \tag{13}$$

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

$$\dot{f}(4\theta) = -4\lambda \sin(4\theta) + 4\delta \cos(4\theta). \tag{14}$$

where

$$\begin{aligned} \lambda &= \left(\mathbb{V}_{\mathbf{x}}^2 - \mathbb{V}_{\mathbf{y}}\right)^2 - 4\mathbb{C}_{\mathbf{x,y}}^2, \\ \delta &= 4\mathbb{C}_{\mathbf{x,y}}\left(\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}}\right). \end{aligned} \tag{15}$$

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

$$\tan(4\theta^*) = \frac{\delta}{\lambda} = \frac{4\mathbb{C}_{\mathbf{x,y}}\left(\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}}\right)}{\left(\mathbb{V}_{\mathbf{x}}^2 - \mathbb{V}_{\mathbf{y}}\right)^2 - 4\mathbb{C}_{\mathbf{x,y}}^2}. \tag{16}$$

Lemma 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:

$$\ddot{f}(\theta) = -16(\lambda \cos(4\theta) + \delta \sin(4\theta)). \tag{17}$$

Multiplying both sides of the relation (17) by $\cos(4\theta)$ yields

$$\cos(4\theta)\ddot{f}(\theta) = -16\left(\lambda \cos^2(4\theta) + \delta \sin(4\theta)\cos(4\theta)\right), \tag{18}$$

Multiplying both sides of the relation (14) by $\sin(4\theta)$ yields

$$\sin(4\theta)\dot{f}(\theta) = -4\lambda \sin^2(4\theta) + 4\delta \sin(4\theta)\cos(4\theta). \tag{19}$$

Using relations (19) and (18) with $\theta = \theta^*$ (i.e., $\dot{f}(\theta^*) = 0$), we obtain:

$$\begin{aligned} 4\delta \cos(4\theta^*)\sin(4\theta^*) &= 4\lambda \sin^2(4\theta^*), \\ \cos(4\theta^*)\ddot{f}(\theta^*) &= -16\lambda. \end{aligned} \tag{20}$$

Therefore, $\ddot{f}(\theta^*)$ has the same sign as $-\lambda \cos(4\theta^*)$. Since $4\theta^* = \arctan(\delta/\lambda) \in \,]-(\pi/2), \pi/2[$, then $\cos(4\theta^*) > 0$ what ever the set $\mathscr{M}_n$ may be. Thus, $\ddot{f}(\theta^*)$ has the same sign as $-\lambda$.

Therefore, the determination of θ^* goes through six cases:

Case 1: if $\lambda > 0$, then $\theta^* = 1/4\arctan(\delta/\lambda)$.

Case 2: if $\lambda < 0$ and $\mathbb{V}_{\mathbf{x}} \neq \mathbb{V}_{\mathbf{y}}$, then θ^* is determined through the eigen decomposition 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 $\theta^* = \pi/4$.

Case 4: if $\lambda = 0$ and $\mathbb{C}_{\mathbf{x}}(\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}}) > 0$, then $\theta^* = \pi/8$.

Case 5: if $\lambda = 0$ and $\mathbb{C}_{\mathbf{x}}(\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}}) > 0$, then $\theta^* = \pi/8$.

Case 6: if $\lambda = 0$ and $\mathbb{C}_{\mathbf{x}}(\mathbb{V}_{\mathbf{x}} - \mathbb{V}_{\mathbf{y}}) > 0$, then it is equivalent to $\mathbb{V}_{\mathbf{x}} = \mathbb{V}_{\mathbf{y}}$ and $\mathbb{C}_{\mathbf{x}} = 0$. This case is discussed below. □

5.1.1. 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, and human faces. Moreover, we have pointed out this case to warn the user to pay attention to regularly shaped objects when using our annotation method.

Proposition 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

$$M_k = \left(\cos\left(\frac{2\pi(k-1)}{n}\right),\ \sin\left(\frac{2\pi(k-1)}{n}\right)\right)^t. \tag{21}$$

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

$$\sum_{k=0}^{n-1} z^k = n(\overline{x} + \overline{y}i). \tag{22}$$

Since $\sum_{k=0}^{n-1} z^k = (1 - z^n/1 - z) = 0$, then $\overline{x} = \overline{y} = 0$, and

$$\begin{aligned}\mathbb{C}_{x,y} &= \frac{1}{n}\sum_{k=1}^{n} x_k y_k,\\ \mathbb{V}_x &= \frac{1}{n}\sum_{k=1}^{n} x_k^2,\\ \mathbb{V}_y &= \frac{1}{n}\sum_{k=1}^{n} y_k^2.\end{aligned} \tag{23}$$

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

$$\begin{aligned}\mathbb{V}_x &= \frac{1}{2} + \frac{1}{2n}\sum_{k=0}^{n-1}\cos\left(\frac{4\pi k}{n}\right),\\ \mathbb{V}_y &= \frac{1}{2} - \frac{1}{2n}\sum_{k=0}^{n-1}\cos\left(\frac{4\pi k}{n}\right).\end{aligned} \tag{24}$$

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

$$\sum_{k=0}^{n-1}\left(z^2\right)^k = \frac{1 - z^{2n}}{1 - z^2} = 0. \tag{25}$$

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

$$\mathbb{C}_{\mathbf{x},\mathbf{y}} = \frac{1}{2n}\sum_{k=0}^{n-1}\sin\left(\frac{4\pi k}{n}\right). \tag{26}$$

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

$$\sum_{k=0}^{n-1}\left(z^2\right)^k = \frac{1 - z^{2n}}{1 - z^2} = 0, \tag{27}$$

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 $\mathrm{MBR}_i(\mathcal{M}_n)$, $i = 1,\ldots,n$. For all $i = 1,\ldots,n-1$, $\mathrm{MBR}_i(\mathcal{M}_n)$ is the image of $\mathrm{MBR}_0(\mathcal{M}_n)$ by the rotation of angle $\theta_i = (2i/n)\pi$. □

5.1.2. 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.

(i) 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.

(ii) 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

$$\mathcal{M}_n' = \left\{M_i' = (x_i', y_i')^t \in \mathbb{R}^2; i = 1,\ldots,n\right\}, \tag{28}$$

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

$$\Gamma = \frac{1}{n}\sum_{i=1}^{n}\left(M_i - \overline{M}\right)\left(M_i - \overline{M}\right)^t, \tag{29}$$

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

$$\Gamma' = \frac{1}{n}\sum_{i=1}^{n}\left(M_i' - \overline{M'}\right)\left(M_i' - \overline{M'}\right)^t = \Delta, \tag{30}$$

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

Proposition 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 $\widehat{A}_n^2(\theta)$ 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 (16), the function f has a unique critical point $\theta^* = 0$. Using Lemma 1, $f(\theta)$ is the maximum of $f(\theta)$.

Case 6. To get out of the indeterminacy of Case 6, we propose to add to the set $\mathscr{M}_n$ an artificial point M_{n+1} located in the convex hull of $\mathscr{M}_n$ so that the empirical covariance matrix of the new set $\mathscr{M}_{n+1} = \mathscr{M}_n \cup M_{n+1}$ is different from a scalar matrix. We assume that $(x_{n+1}, y_{n+1})^t$ are the Cartesian coordinates of M_{n+1} with respect to $\mathscr{B}_G = (G, \mathbf{i}, \mathbf{j})$, where $G = (\overline{x}, \overline{y})^t$ is the barycenter of $\mathscr{M}_n$. Without any loss of generality, we assume that $\mathbf{x}_{1:\,n+1}$ ($\mathbf{y}_{1:\,n+1}$, respectively) is the vector of the x-coordinates (y-coordinates, respectively) of the points of $\mathscr{M}_{n+1}$ with respect to $\mathscr{B}_G = (G, \mathbf{i}, \mathbf{j})$. We denote by $\widetilde{\mathbb{V}}_{\mathbf{x}} = \mathrm{Var}(\mathbf{x}_{1:\,n+1})$ and $\widetilde{\mathbb{V}}_{\mathbf{y}} = \mathrm{Var}(\mathbf{y}_{1:\,n+1})$. Lemma 2 gives the relation between $\mathbb{V}_{\mathbf{x}}, \mathbb{V}_{\mathbf{y}}, \mathbb{C}_{\mathbf{x},\mathbf{y}}$ and $\widetilde{\mathbb{V}}_{\mathbf{x}}, \widetilde{\mathbb{V}}_{\mathbf{y}}, \widetilde{\mathbb{C}}_{\mathbf{x},\mathbf{y}}$, respectively. □

Lemma 2. *If $\mathbb{V}_{\mathbf{x}} = \mathbb{V}_{\mathbf{y}}$ and $\overline{x} = \overline{y} = \mathbb{C}_{\mathbf{x},\mathbf{y}} = 0$, then*

$$\begin{aligned}\widetilde{\mathbb{V}}_{\mathbf{x}} &= \frac{n}{n+1}\mathbb{V}_{\mathbf{x}} + \frac{n}{(n+1)^2}x_{n+1}^2,\\ \widetilde{\mathbb{V}}_{\mathbf{y}} &= \frac{n}{n+1}\mathbb{V}_{\mathbf{y}} + \frac{n}{(n+1)^2}y_{n+1}^2,\\ \widetilde{\mathbb{C}}_{\mathbf{x},\mathbf{y}} &= \frac{n}{(n+1)^2}x_{n+1}y_{n+1}.\end{aligned} \tag{31}$$

Proof. We set $\widetilde{x} = (1/n+1)\sum_{i=1}^{n+1} x_i$ and $\widetilde{x} = (1/n+1)\sum_{i=1}^{n+1} y_i$. Since $\overline{x} = \overline{y} = 0$, then $\widetilde{x} = x_{n+1}/n+1$ and $\widetilde{y} = y_{n+1}/n+1$. On the one hand, since $\overline{x} = 0$, we have

$$\begin{aligned}\sum_{i=1}^{n+1}\left(x_i - \widetilde{x}\right)^2 &= \sum_{i=1}^{n} x_i^2 + x_{n+1}^2 - (n+1)\widetilde{x}^2\\ &= n\mathbb{V}_{\mathbf{x}} + \frac{n}{n+1}x_{n+1}^2.\end{aligned} \tag{32}$$

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

$$\begin{aligned}\sum_{i=1}^{n+1}\left(x_i - \widetilde{x}\right)\left(y_i - \widetilde{y}\right) &= \sum_{i=1}^{n+1} x_i y_i - (n+1)\widetilde{x}\widetilde{y}\\ &= \left(\frac{n}{n+1}\right)x_{n+1}y_{n+1}.\end{aligned} \tag{33}$$

Let $\widetilde{f}(\theta)$ be the function defined from $\mathscr{M}_{n+1}$ in the same way that $f(\theta)$ was defined from $\mathscr{M}_n$ in (10). Then,

$$\widetilde{f}(\theta) = \widetilde{\lambda}\cos(\theta) + \widetilde{\delta}\sin(\theta), \tag{34}$$

where

$$\begin{aligned}\widetilde{\lambda} &= \left(\widetilde{\mathbb{V}}_{\mathbf{x}}^2 + \widetilde{\mathbb{V}}_{\mathbf{y}}^2 - 2\widetilde{\mathbb{V}}_{\mathbf{x}}\widetilde{\mathbb{V}}_{\mathbf{y}} - 4\widetilde{\mathbb{C}}_{\mathbf{x},\mathbf{y}}^2\right),\\ \widetilde{\delta} &= 4\widetilde{\mathbb{C}}_{\mathbf{x},\mathbf{y}}\left(\widetilde{\mathbb{V}}_{\mathbf{x}} - \widetilde{\mathbb{V}}_{\mathbf{y}}\right)\sin(4\theta).\end{aligned} \tag{35}$$

Using the identities (31), we have

$$\widetilde{\lambda} = \frac{n^2}{(n+1)^4}\left(x_{n+1}^4 + x_{n+1}^4 - 6x_{n+1}^2 y_{n+1}^2\right), \tag{36}$$

It follows that

(1) $\widetilde{\lambda} \neq 0$ 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

$$\begin{aligned}&D_1\colon\ x = (1 + \sqrt{2})yD_2\colon\ x = (1 - \sqrt{2})y,\\ &D_3\colon\ x = (\sqrt{2} - 1)yD_4\colon\ x = (-1 - \sqrt{2})y.\end{aligned} \tag{37}$$

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

In summary, any point $M_{n+1} = (\overline{x} + x_{n+1}, \overline{y} + y_{n+1})^t$ 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 $(\mathscr{M}_n)$ depends on the values of x_{n+1} and y_{n+1}. We tested this technique on a set $\mathscr{M}_n$ composed of vertices of a regular n-sided polygon, and we observed that the choice

$$M_{n+1} = \frac{1}{2}\left(M_j + M_{\mathrm{nod}(j+1,4)+1}\right). \tag{38}$$

We allow our algorithm to reach MBR $(\mathscr{M}_n)$ 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 $(\mathscr{M}_n)$, and the determination of the midpoint, M_{n+1}, of any two consecutive vertices of CH $(\mathscr{M}_n)$, such that the product of the coordinates of M_{n+1} with respect to $\mathscr{B}_G = (G, \mathbf{i}, \mathbf{j})$, is different from 0. □

5.2. Annotate the DOTA Dataset Using Range-MBR. To adapt our annotation method to the output format required by the DOTA dataset, it suffices to ask the annotator that the first click points to the object's head, and to add a binary variable T equals to 1 if we can differentiate the head of the object from its tail, and zero otherwise. When we pass the set $\mathscr{M}_n$ to the RANGE-MBR algorithm, we save the coordinates of the first point M_1. Next, we determine the rectangle $\mathbb{R} = [R_1, R_2, R_3, R_4] = \text{RANGE} - MBR(\mathscr{M}_n)$.

(1) If the binary variable T equals one, then we determine the vertex R_j of $\mathbb{R}$ that is closest to M_1, and we permute the vertices of $\mathbb{R}$ clockwise so that R_j occupies the first position in this permutation.

(2) If the binary variable T equals zero, then we determine the top left vertex R_j of $\mathbb{R}$, and we permute the vertices of $\mathbb{R}$ clockwise so that R_j occupies the first position in this permutation.

6. 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.

(i) The problem of MBR was dealt with in Section 5. If the user needs an optimal annotation in the sense given by Definition 2, 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.

(ii) 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 oriented detectors when tested on oriented objects. The experimental determination of the threshold angle requires horizontal and oriented object detectors, as well as the already existing oriented object datasets.

6.1. 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) It allows the user to build simultaneously, from an image bank, two databases: one for horizontal objects and another for oriented 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:

(i) θ_0 be a threshold angle fixed by the user (by default, $\theta_0 = 0$),

(ii) $\mathscr{M}_n$ be the set of relevant points selected on the contour of the object,

(iii) H-MBR ($\mathscr{M}_n$) be the horizontal minimum bounding rectangle enclosing $\mathscr{M}_n$,

(iv) O-MBR ($\mathscr{M}_n$) be the minimum bounding rectangle enclosing $\mathscr{M}_n$,

(v) θ be the angle of O-MBR ($\mathscr{M}_n$),

then, the Algorithm 1, and the flowchart shown in Figure4 summarize the stages of construction of horizontal and oriented datasets, from a large-scale image bank.

7. Experimental Study

This experimental study was carried out exclusively with the Matlab R2007b software. We implemented the RC-MBR, PCA-MBR, and 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 ($\mathscr{M}_n$ is a $2 \times n$ matrix)

(3) determine the rectangle $\mathbb{A}$ corresponding to RC-MBR ($\mathscr{M}_n$)

(4) determine the rectangle $\mathbb{B}$ corresponding to RANGE-MBR ($\mathscr{M}_n$)

(5) draw the two rectangles with the colors red and green, respectively

The images used in experiments B,. . .,D are free of rights and collected on the net. Moreover, the optimality criterion of an annotation is given in Definition 2.

Definition 2. (optimal annotation) A rectangle $\mathbb{A} = [A_1, \ldots, A_4]$ enclosing an object is said to be optimal, if it fulfills the conditions (i) and (ii):

(i) the rectangle $\mathbb{A}$ covers the whole object,

(ii) the area of $\mathbb{A}$ is minimal.

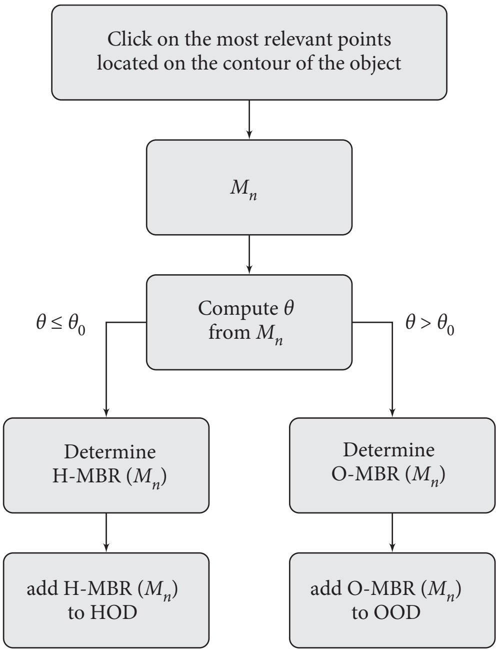


Figure 4: Our approach consists in choosing between O-MBR ($\mathscr{M}_n$) and H-MBR ($\mathscr{M}_n$)). If the angle of O-MBR ($\mathscr{M}_n$) is less than or equal to a threshold angle, then we annotate the object using H-MBR ($\mathscr{M}_n$) instead of O-MBR ($\mathscr{M}_n$).

```
Input: An image; threshold angle; HOD and OOD datasets
Output: Annotated image assigned to HOD or OOD
for (each object in the image) do
    Determine the class of the object
    Ask the annotator to click on the most relevant points ($\mathscr{M}_n$) of the object as defined in Definition 1
    Determine the RANGE-MBR $\mathbb{A} = [A_1, A_2, A_3, A_4]$ enclosing $\mathscr{M}_n$ as described in Section A
    Set $\alpha$ the angle between the positive $x$-axis and $\overrightarrow{A_1A_2}$
    if ($\alpha >$ threshold angle) then
        Assign (image, object, $\mathbb{A}$, class) to the OOD dataset
    else
        Determine the horizontal MBR $\mathbb{A} = [A_1, A_2, A_3, A_4]$ enclosing the relevant points, as shown in Figure 5, by setting $\theta = 0$
        Assign (image, object, $\mathbb{A}$, class) to the HOD dataset
    end if
end for
```

Algorithm 1: Robust semi-automatic annotation

7.1. Experiment 1. This experiment consists of studying the optimality condition (ii) of the RANGE-MBR algorithm. Equation (28) describes how to generate the coordinates of the vertices $\{M_n = (x_k, y_k); k = 1, \ldots, n\}$ of a random n-sided polygon 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 6. 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 7. 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.

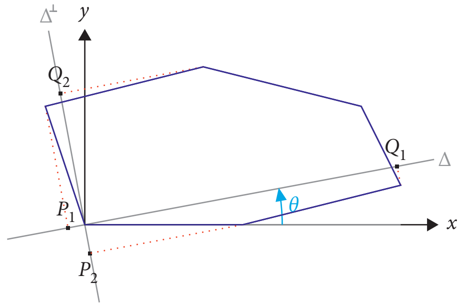


Figure 5: When θ is fixed, the minimum bounding rectangle enclosing the blue polygon (object) is the rectangle $\mathbb{A}$ whose vertices $A_1, \ldots, A_4$ are the intersection points of the two parallel lines to $\Delta^\perp$ and passing through P_1 and Q_1, respectively, with the two parallel lines to Δ and passing through P_2 and Q_2, respectively. $\mathbb{A}$ is the rectangle whose edges pass through the red dotted segments.

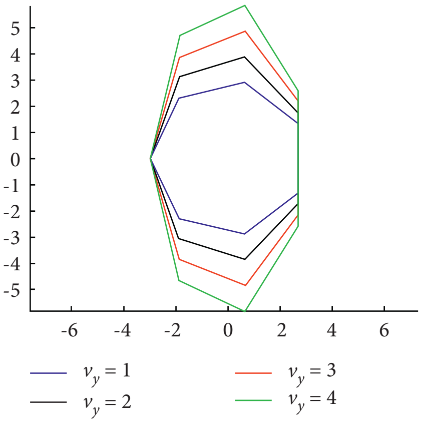


Figure 6: The original object is the regular blue colored heptagon. The black, red, and green polygons correspond to the dilation of the blue polygon by a factor $v_y = 2, 3, 4$, respectively, in the vertical direction.

$$\begin{cases} x_k = (1 + \text{rand})\cos\left(\dfrac{2\pi k}{n} + \dfrac{2\pi}{n}\text{rand}\right), \\ \\ y_k = \left(1 + v_y \text{rand}\right)\sin\left(\dfrac{2\pi k}{n} + \dfrac{2\pi}{n}\text{rand}\right). \end{cases} \tag{39}$$

For each value of the pair (n, v_y), we generate $r = 10000$ random n-sided polygons. For each polygon $\mathcal{M}_n^{(i)}$, $i = 1, \ldots, r$, 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 $\bar{e}$, $\bar{\hat{t}}$, $\bar{t}$, and Std(e), the means and the standard deviation of the sequences $(e_i)_{1\le i\le r}$, $(\hat{t}_i)_{1\le i\le r}$, $(t_i)_{1\le i\le r}$, and $(e_i)_{1\le i\le r}$ respectively.

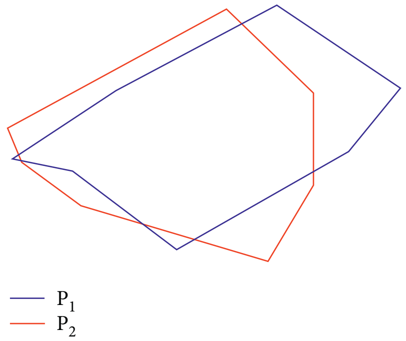


Figure 7: P_1 and P_2 are two random heptagons, generated according to Equation (28) with $v_y = 4$. P_1 is concave, while P_2 is convex.

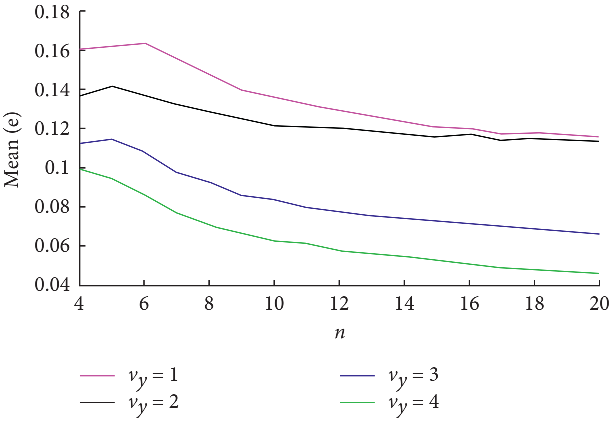


Figure 8: Mean of $(e_i)_{1\le i\le r}$ versus n and v_y.

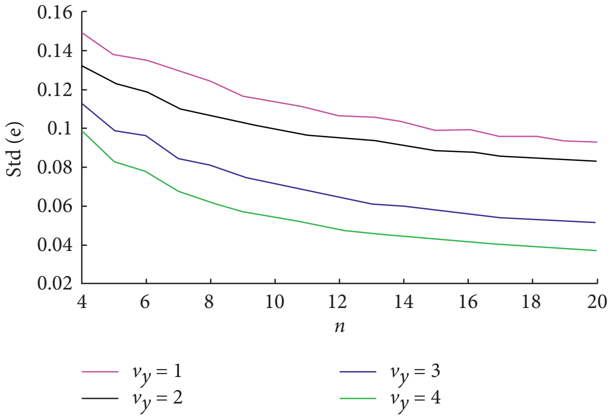


Figure 9: Standard deviation of $(e_i)_{1\le i\le r}$ versus n and v_y.

Figure 8 and 9 represent $\bar{e}$ and Std(e) versus n and v_y for $n \in \{4, \ldots, 20\}$ and $v_y \in \{1, \ldots, 4\}$. 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 8 and 9 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 [44] 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 on the contour of the image.

7.2. Experiment 2. We made a comparison between RANGE-MBR and PCA-MBR based on sets of vertices of regular n-sided polygons, with $n = 4, \ldots, 20$. 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 3. 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.

Table 3: 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

7.3. Experiment 3. This experiment consists in studying the effect of the dilation factor on the performance of the RANGE-MBR algorithm. Figure 10 represents a basic experiment. The relevant points $\mathscr{M}_n$ are colored in yellow. We observe that RANG-MBR $(\mathscr{M}_n)$ is close to RC-MBR $(\mathscr{M}_n)$. 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.

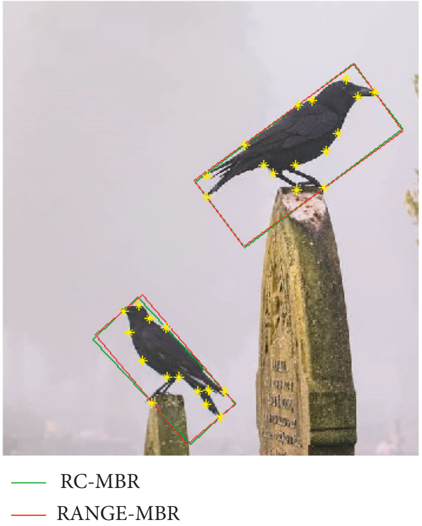


Figure 10: Illustration of the semi-automatic annotation using RC-MBR and RANGE-MBR algorithms.

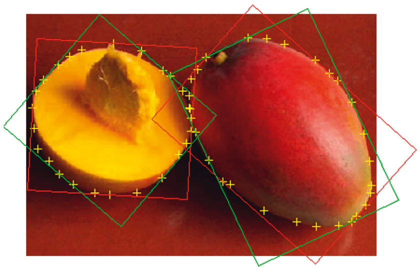


Figure 11: The relevant points $\mathscr{M}_n$ are colored in yellow. The red rectangle corresponds to RANG-MBR $(\mathscr{M}_n)$, while the green rectangle corresponds to RC-MBR $(\mathscr{M}_n)$.

7.4. Experiment 4. This experiment consists in confirming the conclusion obtained in the experiment D. Figure 11 contains two mangoes. The one on the left is almost circular, while the one on the right is clearly elongated. The relevant points $\mathscr{M}_n$ are colored in yellow. The red rectangle corresponds to RANG-MBR $(\mathscr{M}_n)$, while the green rectangle corresponds to RC-MBR $(\mathscr{M}_n)$. 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 8. The 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.

7.5. 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 12 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 12 are larger than those reported in Figure 11. 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 8.

7.6. Experiment 6. This experiment consists in comparing our annotation method to the FBB method introduced in Section B. For this, we have chosen a random image from the

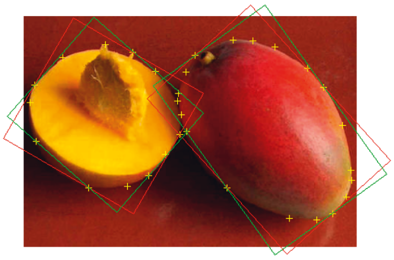


Figure 12: The relevant points $\mathcal{M}_n$ are colored in yellow. The red rectangle corresponds to RANG-MBR ($\mathcal{M}_n$), while the green rectangle corresponds to RC-MBR ($\mathcal{M}_n$).

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 13 and Table 4, it can be seen that the annotation by the FBB method is not optimal. Indeed,

(i) For the upper extinguisher and the exit sign, condition (i) is violated

(ii) For the lower extinguisher, 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.

(i) The values of the relative error are in agreement with the simulation results presented in Figure 8. Indeed, in terms of dilation, the lower extinguisher is the most dilated, followed by the upper extinguisher, then the exit sign.

(ii) The exit sign and the upper extinguisher have regular shapes, while the lower extinguisher has an irregular shape. We have already underlined in Section 1 and 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.

7.7. Experiment 7. This experiment highlights the importance of the annotation method on the calculation of the IoU. Figure 14 corresponds to image P_{1128} from the DOTA dataset. The objects of interest in these images are the airplanes. We use the BBAVectors detector [6] to generate the red prediction bounding boxes. Ground truth bounding boxes used in [6] 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 (IoU_i^b, respectively) the Intersection over Union between the green (blue, respectively) rectangle enclosing the airplane A_i and the corresponding red rectangle. The results of this experiment are reported in Table 5. The use of minimum ground truth rectangles best reflects the performance of an object detector. In view of the results of Table V, it is reasonable to rely on the results of the second row rather than those of the first row.

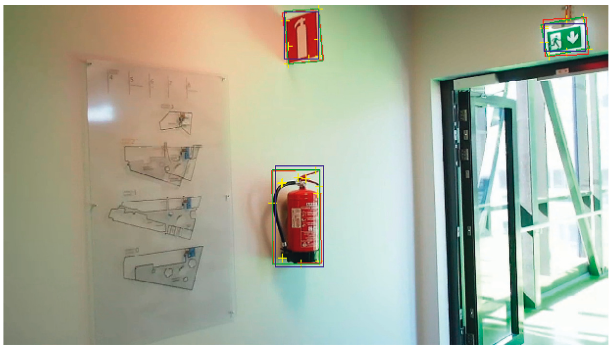


Figure 13: Annotation of an image with FBB, RANGE-MBR, and RC-MBR.

Table 4: Areas of bounding rectangles in Figure 13.

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

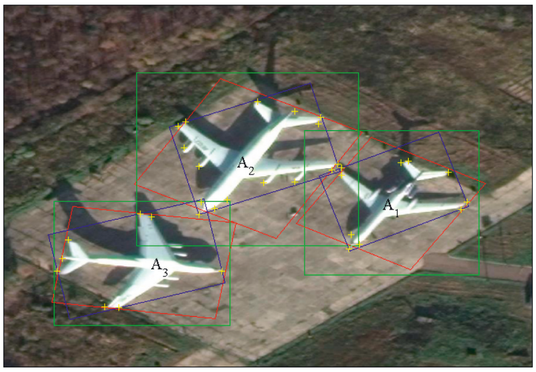


Figure 14: RANGE-MBR allows to build quasi-minimum bounding rectangles.

Table 5: IoU computation for the objects in Figure 14.

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

8. Conclusion

In this piece, we provide a novel approach that is both resilient and semi-automatic in nature for the annotation of objects. According to the results of the experimental research, we are able to assert that

(1) Robust semi-automatic annotation is quasi-optimal in the sense that Definition 2 describes, and that its optimality increases with dilated objects, which is the case for the majority of the objects that appear on the images that were collected by the use of aerial photography.

(2) Robust semi-automatic annotation is quick and reliable in the sense that the bounding rectangle does not have any gaps, and is oblivious to the skill of the person doing the annotation.

(3) Robust semi-automatic annotation is simple to put into action and might be easily incorporated into platforms for annotating text.

(4) Robust semi-automatic annotation is sensitive to the annotation of items displaying symmetry, with respect to one or more directions. Given this scenario, it is not appropriate for the relevant points to follow the same symmetry in order for the created rectangle to be somewhat close to the ideal rectangle.

Thqe critical angle, as described in Section 6, would look somewhat like this: the emphasis on experimental research using large-scale collected data on horizontal and oriented object datasets, as well as detectors of state-of-the-art oriented objects. Once a good estimate of the threshold angle has been made, the robust semi-automatic annotation technique is the one that allows the simultaneous construction, from the same image library, of two different data sets: one for the elements horizontal and another for oriented objects.

Appendix

A. [Script. 1: Collect $\mathscr{M}_n$, draw RC-MBR ($\mathscr{M}_n$) and RANGE-MBR ($\mathscr{M}_n$)] Here, we assume that we have already implemented the RANGE-MBR and RC-MBR functions, in the MATLAB language. The following scripts

(1) Read the image and then display it

(2) Ask the annotator to click on the most relevant points of the object ($\mathscr{M}_n$ is a $2 \times n$ matrix)

(3) Determine the rectangle $\mathbb{A}$ corresponding to RC-MBR ($\mathscr{M}_n$)

(4) Determine the rectangle $\mathbb{B}$ corresponding to RANGE-MBR ($\mathscr{M}_n$)

(5) Draw the two rectangles with the colors red and green, respectively

```
% read the image
photo=imread(’mango.jpg’);
% display the image
image(photo)
button=1;
X=[];
Y=[];
% read ginputs until a mouse right-button occurs
while(sum(button)<=1)
    [x,y,button]=ginput(1);
    X=[X x];
    Y=[Y y];
end
n=length(X);
Mn=[X;Y];
% A is a (2,4)-matrix containing the coordinates of RC-MBR(Mn)
A=RC_MBR(Mn);
A=[B B(:,1)];
% B is a (2,4)-matrix containing the coordinates of RANGE-MBR(Mn)
B=RANGE_MBR(Mn);
B=[B B(:,1)];
hold on
% draw the points of Mn in yellow
plot(M(1,:), M(2,:), ’*y’)
% draw rectangle A in green
plot(A(1,:), A(2,:), ’-g’)
% draw rectangle B in red
plot(B(1,:), B(2,:), ’-r’)
axis equal
```

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

There is a basic version [45] of this study accessible at https://www.researchsquare.com/article/rs-860574/v1. The authors confirm that this study has not been published elsewhere.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] K. He, “Mask r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, October 2017.

[2] Y. Bai, “Coke: localized contrastive learning for robust keypoint detection,” 2020, https://arxiv.org/abs/2009.14115.

[3] P. Hurtik, “Poly-yolo: higher speed, more precise detection and instance segmentation for yolov3,” 2020, https://arxiv.org/abs/2005.13243.

[4] A. Ullah, “Mathematical problems in engineering. experimental and numerical research of paved microcrack using histogram equalization for detection and segmentation,” *Mathematical Problems in Engineering*, vol. 2022, Article ID 2684983, 13 pages, 2022.

[5] J. Iqbal, M. A. Munir, A. Mahmood, A. R. Ali, and M. Ali, “Leveraging orientation for weakly supervised object detection with application to firearm localization,” *Neurocomputing*, vol. 440, pp. 310–320, 2021.

[6] J. Yi, “Oriented object detection in aerial images with box boundary-aware vectors,” 2020, https://arxiv.org/abs/2008.07043.

[7] J. Ding, “Learning roi transformer for detecting oriented objects in aerial images,” in *Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, CA, USA, June 2019.

[8] S. Azimi, “Towards multi-class object detection in unconstrained remote sensing imagery,” in *Proceedings of the Asian Conference on Computer Vision*, pp. 150–165, Perth, Australia, 2018.

[9] J. Ma, W. Shao, H. Ye et al., “Arbitrary-oriented scene text detection via rotation proposals,” *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[10] X. Yang, “R2cnn++: multi-dimensional attention based rotation invariant detector with robust anchor strategy,” 2018, https://arxiv.org/pdf/1811.07126.

[11] Y. Liu, “Deep matching prior network: toward tighter multi-oriented text detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1962–1969, IEEE, Honolulu, HI, USA, July 2017.

[12] X. Liu, Q. Yu, and J. Yan, “Fots: fast oriented text spotting with a unified network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5676–5685, IEEE, Salt Lake City, UT, USA, June 2018.

[13] L. Li, Z. Zhou, B. Wang, L. Miao, and H. Zong, “A novel cnn-based method for accurate ship detection in hr optical remote sensing images via rotated bounding box,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 686–699, 2021.

[14] W. He, Y. Fei, and L. Cheng-Lin, “Deep direct regression for multi-oriented scene text detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 745–753, IEEE, Venice, Italy, October 2017.

[15] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, “Object detection in optical remote sensing images: a survey and a new benchmark,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.

[16] B. Adhikari, P. Jussi, and H. Heikki, “Faster bounding box annotation for object detection in indoor scenes,” in *Proceedings of the European Workshop on Visual Information Processing*, pp. 1–6, London, UK, 2018.

[17] T. Lin, M. Michael, B. Serge et al., “Microsoft coco: common objects in context,” *European Conference on Computer Vision*, Springer, Cham, New York, NY, USA, 2014.

[18] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: a retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[19] O. Russakovsky, J. Deng, H. Su et al., “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] G. Xia, D. Xiang, Z. Jian, and B. Zhen, “Dota: a large-scale dataset for object detection in aerial images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, June 2018.

[21] K. Chen and M. Wu, “Fgsd: a dataset for fine-grained ship detection in high resolution satellite images,” 2020, https://arxiv.org/abs/2003.06832.

[22] Z. Liu, H. Wang, L. Weng, and Y. Yang, “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, no. 8, pp. 1074–1078, 2016.

[23] H. Zhu, X. Chen, W. Dai, K. Fu, and Q. Ye, “Orientation robust object detection in aerial images using deep convolutional neural network,” *IEEE International Conference on Image Processing*, vol. 13, pp. 3735–3739, 2015.

[24] K. Liu and G. Mattyus, “Fast multiclass vehicle detection on aerial images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1938–1942, 2015.

[25] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery: a small target detection benchmark,” *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

[26] C. Benedek, X. Descombes, and J. Zerubia, “Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 33–50, 2012.

[27] J. Ding, B. Xiang, Y. Wen et al., “Object detection in aerial images: a large-scale benchmark and challenges,” 2021, https://arxiv.org/abs/2102.12219.

[28] A. Zaïdi, “Accurate IoU computation for rotated bounding boxes in $${\mathbb {R}}2$$ and $${\mathbb {R}}3$$,” *Machine Vision and Applications*, vol. 32, no. 6, 2021.

[29] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: towards realtime object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, and I. Reid, “Generalized intersection over union: a metric and a loss for bounding box regression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 658–666, IEEE, Long Beach, CA, USA, June 2019.

[31] L. Liu, Z. Pan, and B. Lei, “Learning a rotation invariant detector with rotatable bounding box,” 2017, https://arxiv.org/abs/1711.09405.

[32] C. Yao, B. Xiang, and L. Wenyu, “Detecting texts of arbitrary orientations in natural images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, IEEE, Providence, RI, USA, June 2012.

[33] H. Freeman and R. Shapira, “Determining the minimum-area encasing rectangle for an arbitrary closed curve,” *Communications of the ACM*, vol. 18, no. 7, pp. 409–413, 1975.

[34] M. Shamos, *Computational geometry*, Ph.D. thesis, Yale University, New Haven, CT, USA, 1978.

[35] H. Toussaint, “Solving geometric problems with the rotating calipers,” in *Proceedings of the IEEE MELECON*, Canada, CA, USA, May 1983.

[36] D. Dimitrov, C. Knauer, K. Kriegel, and G. Rote, “On the bounding boxes obtained by principal component analysis,” in *Proceedings of the European Workshop on Computational Geometry*, pp. 193–1196, March 2006.

[37] D. Dimitrov, C. Knauer, K. Kriegel, and G. Rote, “Bounds on the quality of the pca bounding boxes,” *Computational Geometry*, vol. 42, no. 8, pp. 772–789, 2009.

[38] A. Ramirez and C. Cox, “Improving on the range rule of thumb,” *Rose Hulman Undergraduate Mathematics Journal*, vol. 13, no. 2, 2012.

[39] S. P. Hozo, B. Djulbegovic, and I. Hozo, “Estimating the mean and variance from the median, range, and the size of a sample,” *BMC Medical Research Methodology*, vol. 5, no. 1, 2005.

[40] Z. Yi, K. Zhao, J. Sun, L. Wang, K. Wang, and Y. Ma, “Prediction of the remaining useful life of supercapacitors,” *Mathematical Problems in Engineering*, vol. 2022, Article ID 7620382, 8 pages, 2022.

[41] M. Elloumi, H. Gassara, and O. Naifar, “An overview on modelling of complex interconnected nonlinear systems,”

Mathematical Problems in Engineering, vol. 2022, Article ID 4789405, 18 pages, 2022.

[42] H. Sun, J. Sun, K. Zhao, L. Wang, and K. Wang, "Data-driven ica-bi-lstm-combined lithium battery soh estimation," *Mathematical Problems in Engineering*, vol. 2022, Article ID 9645892, 8 pages, 2022.

[43] X. Wan, W. Wang, J. Liu, and T. Tong, "Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range," *BMC Medical Research Methodology*, vol. 14, no. 1, 2014.

[44] D. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4940–4949, IEEE, Venice, Italy, 2017.

[45] A. Zaïdi, "Robust semi-automatic annotation of object data sets with bounding rectangles," 2021.