V. PIERLOT, M. VAN DROOGENBROECK, and *M. Urbin-Choffray*. **A new three object triangulation algorithm based on the power center of three circles**. In *Research and Education in Robotics (EUROBOT)*, volume 161 of Communications in Computer and Information Science, pages 248-262, 2011. Springer.

A New Three Object Triangulation Algorithm Based on the Power Center of Three Circles

V. Pierlot, M. Urbin-Choffray, and M. Van Droogenbroeck

INTELSIG Laboratory, Montefiore Institute, University of Liège, Belgium

{vpierlot,M.VanDroogenbroeck}@ulg.ac.be

Abstract Positioning is a fundamental issue in mobile robot applications that can be achieved in multiple ways. Among these methods, triangulation is a proven technique. As it exists for a long time, many variants of triangulation have been proposed. Which variant is most appropriate depends on the application because some methods ignore the beacon ordering while other have blind spots. Some methods are reliable but at a price of increasing complexity or special cases study. In this paper, we present a simple and new three object triangulation algorithm. Our algorithm works in the whole plane (except when the beacons and the robot are concyclic or colinear), and for any beacon ordering. Moreover, it does not need special cases study and has a strong geometrical meaning. Benchmarks show that our algorithm is faster than existing and comparable algorithms. Finally, a quality measure is intrinsically derived for the triangulation result in the whole plane, which can be used to identify the pathological cases, or as a validation gate in Kalman filters.

Keywords: mobile robots, positioning, triangulation

1 Introduction

Positioning is a fundamental issue in mobile robot applications. Indeed, in most cases, a mobile robot that moves in its environment has to position itself before it can execute its actions correctly. Therefore the robot has to be equipped with some hardware and software capable to provide a sensory feedback related to its environment [1]. Positioning methods can be classified into two main groups [3]: (1) *relative* positioning, and (2) *global* or *absolute* positioning. The first group (also called *dead-reckoning*) achieves positioning by odometry which consists to count the number of wheel revolutions to compute the offset relative to a known position. Odometry is very accurate for small offsets but is not sufficient because of the unbounded accumulation of errors over time (due to wheel slippage, imprecision in the wheel circumference, or wheel inter axis) [3]. Furthermore odometry needs an initial position and fails when the robot is "waken-up" (after a forced reset for example) or is raised and dropped somewhere, since the reference position is unknown or modified. A global positioning system is thus required to recalibrate the robot position periodically.

V. PIERLOT, M. VAN DROOGENBROECK, and *M. Urbin-Choffray*. **A new three object triangulation algorithm based on the power center of three circles**. In *Research and Education in Robotics (EUROBOT)*, volume 161 of Communications in Computer and Information Science, pages 248-262, 2011. Springer.

Relative and global positioning are complementary to each other [1,4] and are typically merged together by using a Kalman filter [12,14]. In many cases, global positioning is ensured by beacon-based triangulation or trilateration. Triangulation is the process of determining the location of a point by measuring angles to it from known points, while trilateration methods involve the determination of absolute or relative locations of points by measurement of distances. Because of the large variety of angle measurement systems, triangulation has emerged as a widely used, robust, accurate, and flexible technique [10]. Another advantage of triangulation versus trilateration is that the robot can compute its orientation (or heading) in addition to its position, so that the complete *pose* of the robot can be found. The process of determining the robot *pose* from three beacon angle measurements is termed *Three Object Triangulation* [7]. Fig. 1 illustrates the process of triangulation. In the remainder of this paper, we concentrate on three object triangulation methods.

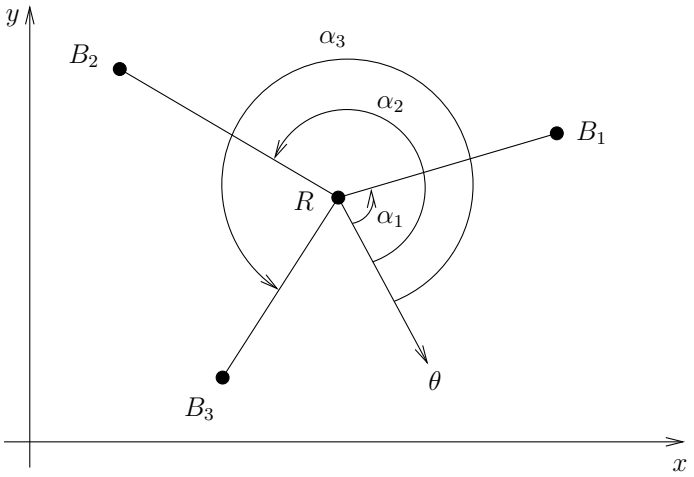

Figure 1. Triangulation setup in the 2D plane. R denotes the robot. B_1, B_2, and B_3 are the beacons. α_1, α_2, and α_3 are the angle measurements respectively for B_1, B_2, and B_3, relatively to the robot reference orientation θ.

1.1 Related Works

Various triangulation algorithms may be found in [1,5,6,7,8,10,11,12,13,14,15,16]. These algorithms can be classified into four groups: (1) *Geometric Triangulation*, (2) *Geometric Circle Intersection*, (3) *Iterative methods* (Iterative Search, Newton-Raphson, etc), and (4) *Multiple Beacons Triangulation*. The first group could be named *Trigonometric Triangulation* because it makes an intensive use of trigonometric computations. The second group computes the parameters (radius and center) of two (of the three) circles passing through the beacons and the

V. PIERLOT, M. VAN DROOGENBROECK, and *M. Urbin-Choffray*. **A new three object triangulation algorithm based on the power center of three circles**. In *Research and Education in Robotics (EUROBOT)*, volume 161 of Communications in Computer and Information Science, pages 248-262, 2011. Springer.

robot, then it computes the intersection between these two circles. The first and second groups are typically used as a solution of the three object triangulation problem. The third group linearizes the trigonometric relations to converge to the robot position after some iterations, from a starting point (usually the last known robot position). The fourth group addresses the more general problem of finding the robot pose from multiple angle measurements (usually corrupted by errors). It appears that the second group (*Geometric Circle Intersection*) is the most popular for solving the three object triangulation problem [13,16].

These algorithms all have advantages and drawbacks, and the method has to take the requirements of a particular application into account, which leads to make some compromises. For example, if the setup contains three beacons only or if the robot platform has a low computing power, methods of the first and second groups are the best candidates. Methods of the third and fourth groups are appropriate if the application must handle multiple beacons and if it can accommodate to a higher computational cost. The main drawback of the third group is the convergence issue (existence or uniqueness of the solution) [7]. The main drawback of the fourth group is the computational cost [1,5].

The drawbacks of the first and second group are usually a lack of precision in the following points: (1) the beacon ordering needed to get the correct solution, (2) the consistency of the methods when the robot is located outside the triangle defined by the three beacons, (3) the strategy to follow when falling into some particular geometrical cases (typically mathematical undeterminations when solving trigonometric equations with an argument equal to 0 or π, division by 0, etc), and (4) the quality measure of the computed position. Simple methods of the first and second groups usually fail to propose an answer to all these raised issues. To work in the whole plane and for any beacon ordering (for instance [11]), they have to consider a set of special geometrical cases, resulting in a lack of clarity in the method. Finally, none of these algorithms gives a realistic quality measure of the computed position.

1.2 Overview

Our paper presents a new three object triangulation algorithm that works in the whole plane (except when the beacons and the robot are concyclic or colinear), and for any beacon ordering. Moreover it uses a minimal number of trigonometric computations and, finally, it leads to a natural and quantitative quality measure of the computed position.

The paper is organized as follows. Our triangulation algorithm is described in Section 2. Section 3 presents simulation results. Then, we conclude the paper in Section 4.

2 Description of a New Three Object Triangulation Algorithm

Our algorithm belongs to the second group, that is: *Geometric Circle Intersection*. It first computes the parameters of the three circles passing through the

robot and the three pairs of beacons. Then it computes the intersection of these three circles, by using all the three circles, not only two of them.

Our algorithm relies on two assumptions: (1) the beacons are distinguishable (a measured angle can be associated to a given beacon), and (2) the angle measurements from the beacons are taken separately, and relatively to some reference angle θ, usually the robot heading (see Fig. 1). Note that the second hypothesis simply states that angles are given by a rotating angular sensor (goniometer). Such sensors are common in mobile robot positioning using triangulation [2,3,6,15,16,17]. By convention, in the following, we consider that angles are measured counterclockwise (CCW), like angles on the trigonometric circle. Changing the rotating direction to clockwise (CW) requires a minimal changes of our algorithm.

2.1 First Part of the Algorithm: the Circle Parameters

In a first step, we have to find the locus of points R that *see* two fixed points B_1 and B_2 with a constant angle α_{12}, in the 2D plane. It is a well-known result that this locus is an arc of the circle passing through B_1 and B_2, whose radius depends on the distance between B_1 and B_2, and α_{12} (Proposition 21 of Book III of EUCLID's Elements). More precisely, this locus is composed of two arcs of circle, which are the reflection of each other through the line joining B_1 and B_2 (see the left drawing of Fig. 2).

A robot that measures an angle α_{12} between two beacons without any caution can stand on either of these two arcs. It would be the case if the beacons were not distinguishable or if the angular sensor was not capable to measure angles larger than π. To avoid this ambiguity, we impose that, as shown in the right-hand drawing of Fig. 2, the measured angle between two beacons B_1 and B_2, denoted α_{12}, is always computed as $\alpha_{12} = \alpha_2 - \alpha_1$ (this choice is natural for a CCW rotating sensor). This is consistent with our measurement considerations and it removes the ambiguity about the locus. For now the locus is a single circle passing through R, B_1, and B_2. But it requires that beacons are indexed and that the robot is capable to guess the index of any beacon.

The circle equation may be derived by using the complex representation of 2D points (ARGAND diagram). The idea consists to express that the complex argument of $(B_2 - R)$ is equal to the complex argument of $(B_1 - R)$, plus α, or:

$$\arg\left\{\frac{B_2 - R}{B_1 - R}\right\} = \alpha$$

$$\Rightarrow \arg\left\{(B_2 - R)\,\overline{(B_1 - R)}\right\} = \alpha$$

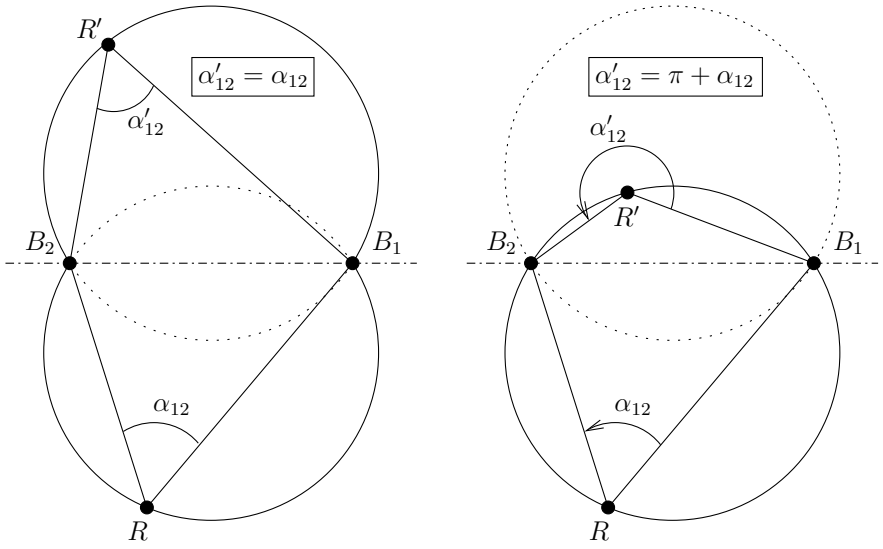

Figure 2. Left-hand side drawing: the locus of points R that *see* two fixed points B_1 and B_2 with a constant angle α_{12}, in the 2D plane, is formed by two arcs of circle. Right-hand side drawing: the ambiguity about the locus is removed by taking the following convention: $\alpha_{12} = \alpha_2 - \alpha_1$.

Then replacing R by $(x + iy)$, B_1 by $(x_1 + iy_1)$, and B_2 by $(x_2 + iy_2)$, we have that

$$\begin{aligned} &\arg\left\{(x_2 + iy_2 - x - iy)\,(x_1 - iy_1 - x + iy)\,e^{-i\alpha}\right\} = 0 \\ \Rightarrow\; &Im\left\{[(x_2 - x) + i\,(y_2 - y)]\,[(x_1 - x) + i\,(y - y_1)]\,[\cos\alpha - i\sin\alpha]\right\} = 0 \\ \Rightarrow\; &-\sin\alpha\,(x_2 - x)\,(x_1 - x) + \sin\alpha\,(y_2 - y)\,(y - y_1) \\ &+\cos\alpha\,(x_2 - x)\,(y - y_1) + \cos\alpha\,(y_2 - y)\,(x_1 - x) = 0 \end{aligned}$$

where $i = \sqrt{-1}$. After many simplifications, we find the locus

$$(x - x_{12})^2 + (y - y_{12})^2 = R_{12}^2$$

which is a circle whose center $\{x_{12},\, y_{12}\}$ is located at

$$x_{12} = \frac{(x_1 + x_2) + \cot\alpha\,(y_1 - y_2)}{2}, \qquad y_{12} = \frac{(y_1 + y_2) - \cot\alpha\,(x_1 - x_2)}{2}$$

and whose squared radius equals

$$R_{12}^2 = \frac{(x_1 - x_2)^2 + (y_1 - y_2)^2}{4\,\sin^2\alpha}$$

These equations may also be found in [13]. The replacement of α by $\pi + \alpha$ in the above equations yields the same circle parameters (Fig. 2, right), which is

consistent with our measurement considerations. For an angular sensor turning in CW direction, one have to change the sign of $\cot\alpha$ in the center coordinates equations. In the following, we use these notations:

- B_i is the beacon $\#i$, with coordinates $\{x_i, y_i\}$,
- R is the robot position, with coordinates $\{x_R, y_R\}$,
- α_i is the measured angle for beacon B_i with respect to the robot orientation,
- $\alpha_{ij} = \alpha_j - \alpha_i$ is the bearing angle between beacons B_i and B_j,
- $T_{ij} = \cot(\alpha_{ij})$,
- $\mathcal{C}_{ij}$ is the circle passing through B_i, B_j, and R,
- c_{ij} is the center of $\mathcal{C}_{ij}$, with coordinates $\{x_{ij}, y_{ij}\}$:

$$x_{ij} = \frac{(x_i + x_j) + T_{ij}\,(y_i - y_j)}{2}, \qquad y_{ij} = \frac{(y_i + y_j) - T_{ij}\,(x_i - x_j)}{2} \tag{1}$$

- R_{ij} is the radius of $\mathcal{C}_{ij}$, derived from:

$$R_{ij}^2 = \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{4\,\sin^2\alpha_{ij}} \tag{2}$$

All the previous quantities are valid for $i = 1, 2, 3$ and $j = (i)\,mod\,3 + 1$. The special cases ($\alpha_{ij} = 0$ or $\alpha_{ij} = \pi$) are discussed later.

2.2 Second Part of the Algorithm: the Circles Intersection

From the previous section, each bearing angle α_{ij} between beacons B_i and B_j constraints the robot to be on a circle $\mathcal{C}_{ij}$, passing through B_i, B_j, and R (Fig. 3). The parameters of the circles are given by Equ. 1 and 2. Note that we are in the case of a trilateration problem with virtual beacons (the circle centers) and virtual range measurements (the circle radii). Common methods use two of the three circles to compute the intersections (when they exist), one of which is the robot position, the second being the common beacon of the two circles. This requires to solve a quadratic system and to choose the correct solution for the robot position [13]. Moreover the choice of the two circles is arbitrary and usually static, whereas this choice should depend on the measured angles and beacons configuration.

Hereafter, we propose a novel method to compute this intersection, by using all the three circles, and reducing the problem to a linear problem. To understand this simple method, we first have to remind the notion of the *power center* (or *radical center*) of three circles. The *power center* of three circles is the unique point of equal *power* with respect to these circles [9]. The *power* of a point p relative to a circle $\mathcal{C}$ is defined as:

$$\mathcal{P}_{\mathcal{C},p} = (x - x_c)^2 + (y - y_c)^2 - R^2 \tag{3}$$

where $\{x, y\}$ are the coordinates of point p, $\{x_c, y_c\}$ are the circle center coordinates and R is the circle radius. The power of a point is null onto the circle,

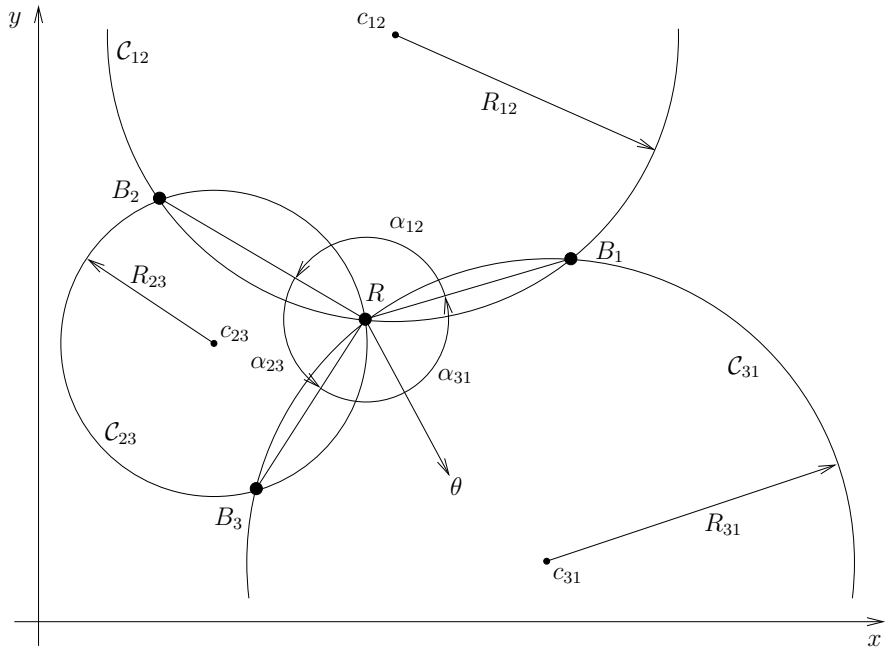

Figure 3. Triangulation setup in the 2D plane, using the geometric circle intersection. R is the robot. B_1, B_2, and B_3 are the beacons. α_{ij} are the angles between B_i, R, and B_j. $\mathcal{C}_{ij}$ are the circles passing through B_i, R, and B_j. R_{ij} and c_{ij} are respectively the radii and center coordinates of $\mathcal{C}_{ij}$. θ is the robot orientation in the world coordinate.

negative inside the circle and positive outside the circle. It defines a sort of distance of a point relative to a circle. The *power line* (or *radical axis*) of two circles is the locus of points having the same power with respect to each circle [9]. It is perpendicular to the line joining the circle centers and passes through the circle intersections, when they exist. When considering three circles, the three power lines, defined by the three pairs of circles are concurring in the power center [9]. Fig. 4 shows the power center of three circles for various configurations. The power center is always defined, except when at least two of the three circle centers are equal, or when the circle centers are colinear (parallel power lines).

The third case of Fig. 4 (right-hand drawing) is remarkable as it perfectly matches to our triangulation problem (Fig. 3). Indeed, the power center of three concurring circles corresponds to their unique intersection. In our case, we are sure that the circles are concurring since $\alpha_{31} = -(\alpha_{12} + \alpha_{23})$ by construction (only two of the three bearing angles are independent). It has the advantage that this intersection may be computed by intersecting the power lines, which is a linear problem. The power line of two circles is obtained by equating the power of the points relatively to each circle (Equ. 3). In our problem, the power

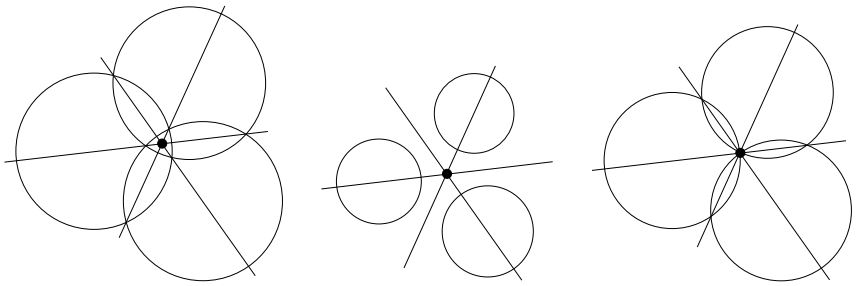

Figure 4. The black point is the power center of three circles for various configurations. It is the unique point having the same power with respect to the three circles. It is the intersection of the three power lines.

line of $\mathcal{C}_{12}$ and $\mathcal{C}_{23}$ is given by:

$$\begin{aligned}
&(x-x_{12})^2+(y-y_{12})^2-R_{12}^2=(x-x_{23})^2+(y-y_{23})^2-R_{23}^2\\
\Rightarrow\ &x\,(x_{12}-x_{23})+y\,(y_{12}-y_{23})=\frac{x_{12}^2+y_{12}^2-R_{12}^2}{2}-\frac{x_{23}^2+y_{23}^2-R_{23}^2}{2}\\
\Rightarrow\ &x\,(x_{12}-x_{23})+y\,(y_{12}-y_{23})=k_{12}-k_{23}
\end{aligned}$$

where we introduce a new quantity k_{ij} which only depends on $\mathcal{C}_{ij}$ parameters (k_{ij} is the power of the origin relatively to $\mathcal{C}_{ij}$, divided by two):

$$k_{ij}=\frac{x_{ij}^2+y_{ij}^2-R_{ij}^2}{2} \tag{4}$$

In our triangulation problem, we have to intersect the three power lines, that is to solve this linear system:

$$\begin{aligned}
x\,(x_{12}-x_{23})+y\,(y_{12}-y_{23})&=k_{12}-k_{23}\\
x\,(x_{23}-x_{31})+y\,(y_{23}-y_{31})&=k_{23}-k_{31}\\
x\,(x_{31}-x_{12})+y\,(y_{31}-y_{12})&=k_{31}-k_{12}
\end{aligned}$$

As can be seen, any of these equations may be obtained by adding the two others, which is way to prove that the three power lines coincide in a unique point: the power center. The coordinates of the power center, that is the robot position is given by:

$$x_R=\frac{\begin{vmatrix}k_{12}-k_{23} & y_{12}-y_{23}\\ k_{23}-k_{31} & y_{23}-y_{31}\end{vmatrix}}{\begin{vmatrix}x_{12}-x_{23} & y_{12}-y_{23}\\ x_{23}-x_{31} & y_{23}-y_{31}\end{vmatrix}},\qquad y_R=\frac{\begin{vmatrix}x_{12}-x_{23} & k_{12}-k_{23}\\ x_{23}-x_{31} & k_{23}-k_{31}\end{vmatrix}}{\begin{vmatrix}x_{12}-x_{23} & y_{12}-y_{23}\\ x_{23}-x_{31} & y_{23}-y_{31}\end{vmatrix}} \tag{5}$$

The denominator D, common to x_R and y_R is equal to:

$$D=\begin{vmatrix}x_{12}-x_{23} & y_{12}-y_{23}\\ x_{23}-x_{31} & y_{23}-y_{31}\end{vmatrix}=\begin{vmatrix}x_{12} & y_{12} & 1\\ x_{23} & y_{23} & 1\\ x_{31} & y_{31} & 1\end{vmatrix} \tag{6}$$

V. PIERLOT, M. VAN DROOGENBROECK, and *M. Urbin-Choffray*. **A new three object triangulation algorithm based on the power center of three circles**. In *Research and Education in Robotics (EUROBOT)*, volume 161 of Communications in Computer and Information Science, pages 248-262, 2011. Springer.

which is the signed area between the circle centers, multiplied by 2. This result confirms that the power center exists, that is $D \neq 0$, if the circle centers are not colinear. The special case ($D = 0$) is discussed later.

2.3 First (Naive) Version of the Algorithm

A first, but naive version of our algorithm is described by Algorithm 1.

Algorithm 1 First version of the algorithm.

1. compute the three cot(.): $T_{ij} = \cot(\alpha_{ij})$,
2. compute the six circle centers coordinates $\{x_{ij}, y_{ij}\}$ by Equ. 1,
3. compute the three squared radii R_{ij}^2 by Equ. 2,
4. compute the three parameters k_{ij} by Equ. 4,
5. compute the denominator D by Equ. 6. Return with an error if $D = 0$.
6. compute the robot position $\{x_R, y_R\}$ by Equ. 5 and return.

This method is correct but it is possible to simplify it with some considerations about the involved equations. First, note that the squared radii R_{ij}^2 only appear in the parameters k_{ij}. If we replace the expression of R_{ij}^2 (Equ. 2) in the expression of k_{ij} (Equ. 4), we find, after many simplifications that:

$$k_{ij} = \frac{x_i x_j + y_i y_j + T_{ij}(x_j y_i - x_i y_j)}{2} \tag{7}$$

which is much simpler than Equ. 2 and 4 (no squared terms anymore). In addition, the $1/2$ factor involved in the circle centers coordinates (Equ. 1) as well as in the parameters k_{ij} (Equ. 4) disappears in the robot position coordinates (Equ. 5). This factor can thus be omitted. For now, we use these modified circle center coordinates $\{x'_{ij}, y'_{ij}\}$:

$$x'_{ij} = (x_i + x_j) + T_{ij}\,(y_i - y_j)\,, \qquad y'_{ij} = (y_i + y_j) - T_{ij}\,(x_i - x_j) \tag{8}$$

and parameters k'_{ij}:

$$k'_{ij} = x_i x_j + y_i y_j + T_{ij}(x_j y_i - x_i y_j) \tag{9}$$

The second version of our algorithm is given in Algorithm 2.

2.4 Final Version of the Algorithm

The most important simplification consists in translating the world coordinate frame into one of the beacons, that is solving the problem relatively to one beacon and then add the beacon coordinates to the computed robot position. In the following, we arbitrarily choose B_2 as the origin ($B_2 = \{0, 0\}$). The

Algorithm 2 Second version of the algorithm.

1. compute the three cot(.): $T_{ij} = \cot(\alpha_{ij})$,
2. compute the modified circle centers coordinates $\{x'_{ij}, y'_{ij}\}$ by Equ. 8,
3. compute the modified parameters k'_{ij} by Equ. 9,
4. compute the denominator D by Equ. 6. Return with an error if $D = 0$.
5. compute the robot position $\{x_R, y_R\}$ by Equ. 5 and return.

other beacon coordinates become: $B_1 = \{x_1 - x_2, y_1 - y_2\} = \{x'_1, y'_1\}$ and $B_3 = \{x_3 - x_2, y_3 - y_2\} = \{x'_3, y'_3\}$. Since $x'_2 = 0$ and $y'_2 = 0$, we have $k'_{12} = 0$, $k'_{23} = 0$. Also, we can compute the value of one cot(.) by referring to the two other cot(.) because the three angles are not independent ($\alpha_{31} = -(\alpha_{12} + \alpha_{23})$):

$$T_{31} = \frac{1 - T_{12}T_{23}}{T_{12} + T_{23}} \tag{10}$$

The final algorithm is given in Algorithm 3.

Algorithm 3 Final version of the algorithm.

Given the three beacon coordinates $\{x_i, y_i\}$ and the three angle measurements α_i:

1. compute the modified beacon coordinates:

$$x'_1 = x_1 - x_2, \qquad y'_1 = y_1 - y_2, \qquad x'_3 = x_3 - x_2, \qquad y'_3 = y_3 - y_2$$

2. compute the three cot(.):

$$T_{12} = \cot(\alpha_2 - \alpha_1), \qquad T_{23} = \cot(\alpha_3 - \alpha_2), \qquad T_{31} = \frac{1 - T_{12}T_{23}}{T_{12} + T_{23}}$$

3. compute the modified circle center coordinates $\{x'_{ij}, y'_{ij}\}$:

$$x'_{12} = x'_1 + T_{12}\, y'_1, \qquad y'_{12} = y'_1 - T_{12}\, x'_1$$

$$x'_{23} = x'_3 - T_{23}\, y'_3, \qquad y'_{23} = y'_3 + T_{23}\, x'_3$$

$$x'_{31} = (x'_3 + x'_1) + T_{31}\,(y'_3 - y'_1), \qquad y'_{31} = (y'_3 + y'_1) - T_{31}\,(x'_3 - x'_1)$$

4. compute k'_{31}:

$$k'_{31} = x'_1x'_3 + y'_1y'_3 + T_{31}(x'_1y'_3 - x'_3y'_1)$$

5. compute the denominator D (if $D = 0$, return with an error):

$$D = (x'_{12} - x'_{23})(y'_{23} - y'_{31}) - (y'_{12} - y'_{23})(x'_{23} - x'_{31})$$

6. compute the robot position $\{x_R, y_R\}$ and return:

$$x_R = x_2 + \frac{k'_{31}(y'_{12} - y'_{23})}{D}, \qquad y_R = y_2 + \frac{k'_{31}(x'_{23} - x'_{12})}{D}$$

V. PIERLOT, M. VAN DROOGENBROECK, and *M. Urbin-Choffray*. **A new three object triangulation algorithm based on the power center of three circles**. In *Research and Education in Robotics (EUROBOT)*, volume 161 of Communications in Computer and Information Science, pages 248-262, 2011. Springer.

2.5 Discussion

This algorithm is very simple, while keeping a strong geometrical meaning (each involved quantity has a physical meaning). Moreover, the number of conditional statements is reduced (there are two tests related to the argument of cot(.) and one conditional test to check if $D = 0$), which increases its readability and eases its implementation. Furthermore, computations are limited to basic arithmetic computations and two cot(.) computations. Among these computations, we have to take care of the cot(.) and the division by D. If a bearing angle α_{ij} between two beacons is equal to 0 or π, that is the robot stands on the line B_iB_j, the $\cot(\alpha_{ij})$ is infinite. The corresponding circle degenerates as the line B_iB_j (infinite radius and center coordinates). The robot position is the intersection of the remaining power line and the line B_iB_j. It can be shown that the mathematical limit $\lim_{T_{ij}\to\pm\infty}\{x_R, y_R\}$ exists and corresponds to this situation. The algorithm could deal with these special cases but it is not necessary. In practice, we have to avoid *Inf* or *NaN* values in floating point computations. This can be done by limiting the cot(.) value to a minimum or maximum value, corresponding to a small angle that is far below the measurement precision. In practice, we limit the value of the cot(.) to $\pm 10^8$, which corresponds to an angle of about $\pm 10^{-8}\,[rad]$; this is indeed far below the existing angular sensor precisions.

The denominator D is equal to 0 when the circle centers are colinear. For non colinear beacons, this situation occurs when the beacons and the robot are concyclic (they all stand on the same circumference, termed the *critic circumference*). In that case, the three circles are equal as well as their centers, which causes $D = 0$. For colinear beacons, this situation is encountered when the beacons and the robot all stand on this line. For these cases, it is impossible to compute the robot position. This is referred as *the restrictions common to all three object triangulation*, whatever the algorithm used [10,13,16]. The value of D, computed in the final algorithm, is the signed area delimited by the circle centers, multiplied by 8. It is quiet natural to use the absolute value of D as a quality measure of the computed position. Indeed $|D|$ decreases to 0 when approaching the critic circumference (almost colinear circle center, almost parallel power lines). In the next section, we show that $1/|D|$ is a good approximation of the position error. In practice, $|D|$ can be used as a validation gate after the triangulation algorithm or when using a Kalman filter with triangulation. Finally, it should be noted that the robot orientation may be determined by using any beacon B_i and its corresponding angle measurement α_i, once the robot position is known.

3 Simulations

In order to validate our algorithm, we have performed some simulations in a square shaped area ($4\times4\,[m^2]$), with three non colinear beacons. For each point in this area, we compute the exact angles α_i seen by the robot (the robot orientation is arbitrary set to 0 degree). Then we add Gaussian noise to these angles, with zero mean, and with two different standard deviations ($\sigma = 0.1°$ and $\sigma = 1°$). The

V. PIERLOT, M. VAN DROOGENBROECK, and *M. Urbin-Choffray*. **A new three object triangulation algorithm based on the power center of three circles**. In *Research and Education in Robotics (EUROBOT)*, volume 161 of Communications in Computer and Information Science, pages 248-262, 2011. Springer.

noisy angles are then used as inputs of our algorithm to compute the estimated position. The position error Δd_R is the Euclidean distance between the exact and estimated position. The orientation error $\Delta\theta_R$ is the difference between the exact and estimated orientation. The experiment is repeated several times to compute the mean of the position error $E\{\Delta d_R\}$ and the mean of the orientation error $E\{\Delta\theta_R\}$. The means of the position and orientation error are drawn in Fig. 5. The beacon positions are represented by black and white circles. The left column is the result for $\sigma = 0.1°$, the right column is the result for $\sigma = 1°$. The first, second and third rows show the position error, the orientation error, and the quality measure $1/|D|$ respectively.

Our simulation results are consistent with common three object triangulation algorithms [11,13]. In particular, we can easily spot the critic circumference where errors are large. From these graphics, one can see that $1/|D|$ has a similar shape than the position or orientation error, up to a constant multiplicative factor. It can be proven (but this beyond the scope of this paper), by a detailed sensitivity analysis of the robot position error with respect to angles, that

$$\Delta d_R \simeq \frac{1}{|D|}\,\Delta\alpha\, f(.)$$

where Δd_R is the position error, $\Delta\alpha$ is the angle error, and $f(.)$ is some function of all the other parameters. This explain why $1/|D|$ can be used as an approximation of the position error, up to a constant multiplicative factor.

In this paper, we do not provide results obtained with a real sensor. An absolute comparison between simulation results and real data is at least difficult if not impossible to produce because many unknown or uncontrolled parameters impact on results. The precision of the sensor is one of these critical parameters but the absolute position of beacons and the sensor position onto the robot are very difficult to measure with a high accuracy. These difficulties did not prevented us to implement the algorithm and run it on our platform with success during the qualification of the Eurobot competition.

We have also compared the computational time of our algorithm against two other algorithms. The first algorithm is the *Generalized Geometric Triangulation* from Esteves *et al.* [11]. The second is the *Geometric Circle Intersection* from Font-Llagunes *et al.* [13]. We chose these algorithms because they work in the whole plane and for any beacon ordering, like ours. The simulations were performed in the same square shaped area, with a resolution of $0.5\,[mm]$. It appears that our algorithm is about 40 % faster than Esteves *et al.* [11], and 20 % faster than Font-Llagunes *et al.* [13], for exactly the same precision.

4 Conclusions

This paper presents a new and simple three object triangulation algorithm based on the power center of three circles. As it exists for a long time, many variants of triangulation have been proposed. Which variant is most appropriate depends on the application because some methods ignore the beacon ordering while other

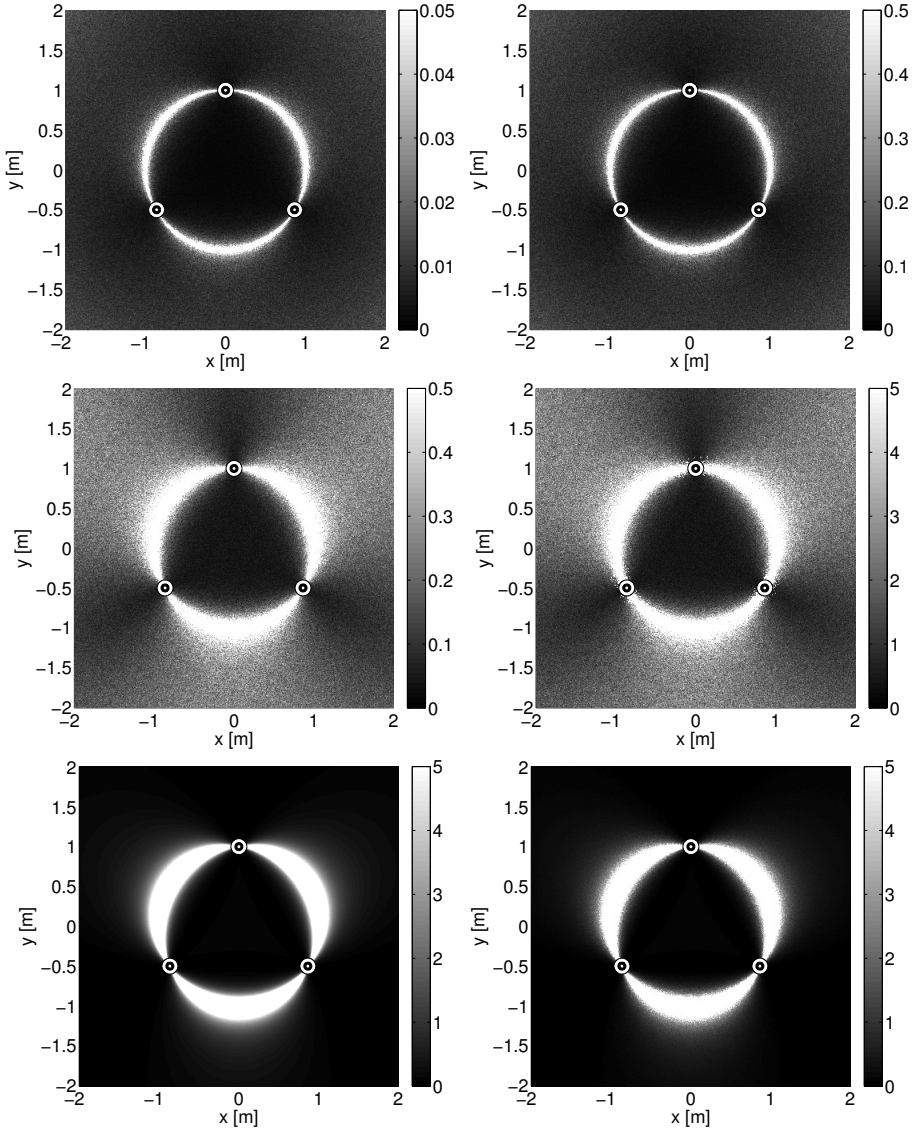

Figure 5. Simulation results giving position and orientation error with noisy angle measurements. The beacon positions are represented by black and white circles. The left column is the result for $\sigma = 0.1°$, the right column is the result for $\sigma = 1°$. The first, second and third rows show the position error (expressed in $[m]$), the orientation error (expressed in $[degree]$), and the quality measure $1/|D|$ (D being expressed in $[m^2]$) respectively.

have blind spots. Some methods are reliable but at a price of increasing complexity or special cases study. Our algorithm works in the whole plane (except when the beacons and the robot are concyclic or colinear), and for any beacon ordering. It does not need special cases study, which makes it clear. Moreover it has a strong geometrical meaning (each involved quantity has a physical meaning), while keeping simple. Furthermore it uses only basic arithmetic computations, and two cot(.) computations. Benchmarks show that our algorithm is faster than existing and comparable algorithms. Finally it naturally gives a quality measure of the triangulation result in the whole plane. Simulation intuitively show that $\frac{1}{|D|}$ is a natural and efficient criterion to estimate the precision of the positioning. To our knowledge, algorithms of the same family do not provide such a criterion. This quality measure can be used to identify the pathological cases (critic circumference), or as a validation gate in Kalman filters based on triangulation.

References

1. Betke, M., Gurvits, L.: Mobile robot localization using landmarks. IEEE Transactions on Robotics and Automation 13(2), 251–263 (April 1997)
2. Borenstein, J., Everett, H., Feng, L.: Where am I? Systems and methods for mobile robot positioning. Tech. rep., University of Michigan (March 1996)
3. Borenstein, J., Everett, H., Feng, L., Wehe, D.: Mobile robot positioning - sensors and techniques. Journal of Robotic Systems 14(4), 231–249 (April 1997)
4. Borenstein, J., Feng, L.: Umbmark: A benchmark test for measuring odometry errors in mobile robots. In: SPIE. Philadelphia (October 1995)
5. Briechle, K., Hanebeck, U.: Localization of a mobile robot using relative bearing measurements. IEEE Transactions on Robotics and Automation 20(1), 36–44 (February 2004)
6. Casanova, E., Quijada, S., Garcia-Bermejo, J., González, J.: A new beacon-based system for the localization of moving objects. In: IEEE International Conference on Mechatronics and Machine Vision in Practice. Chiang Mai, Tailand (September 2002)
7. Cohen, C., Koss, F.: A comprehensive study of three object triangulation. In: Mobile Robots VII. vol. 1831, pp. 95–106. SPIE (1993)
8. Demaine, E., López-Ortiz, A., Munro, J.: Robot localization without depth perception. In: Proceedings of Scandinavian Workshop on Algorithm Theory (SWAT). Lecture Notes in Computer Science 2368, Springer. pp. 177–194 (July 2002)
9. Eiden, J.D.: Géometrie analytique classique. Calvage & Mounet, Paris (2009)
10. Esteves, J., Carvalho, A., Couto, C.: Generalized geometric triangulation algorithm for mobile robot absolute self-localization. In: International Symposium on Industrial Electronics (ISIE). vol. 1, pp. 346–351. Rio de Janeiro, Brazil (June 2003)
11. Esteves, J., Carvalho, A., Couto, C.: Position and orientation errors in mobile robot absolute self-localization using an improved version of the generalized geometric triangulation algorithm. In: IEEE International Conference on Industrial Technology (ICIT). pp. 830–835. Mumbai, India (December 2006)
12. Font-Llagunes, J., Batlle, J.: Mobile robot localization. Revisiting the triangulation methods. In: International Federation of Automatic Control (IFAC). Symposium on Robot Control. vol. 8. Bologna, Italy (September 2006)

13. Font-Llagunes, J., Batlle, J.: Consistent triangulation for mobile robot localization using discontinuous angular measurements. Robotics and Autonomous Systems 57(9), 931–942 (September 2009)
14. Hu, H., Gu, D.: Landmark-based navigation of industrial mobile robots. International Journal of Industry Robot 27(6), 458–467 (2000)
15. Lee, C., Chang, Y., Park, G., Ryu, J., Jeong, S.G., Park, S., Park, J., Lee, H., Hong, K.S., Lee, M.: Indoor positioning system based on incident angles of infrared emitters. Conference of the IEEE Industrial Electronics Society (IECON) 3, 2218–2222 (November 2004)
16. McGillem, C., Rappaport, T.: A beacon navigation method for autonomous vehicles. IEEE Transactions on Vehicular Technology 38(3), 132–139 (August 1989)
17. Pierlot, V., Van Droogenbroeck, M.: A simple and low cost angle measurement system for mobile robot positioning. In: Workshop on Circuits, Systems and Signal Processing (ProRISC). pp. 251–254. Veldhoven, The Netherlands (November 2009)