A Scan Matching Method based on the Area Overlap of Star-Shaped Polygons

Dario Lodi Rizzini, Stefano Caselli
RIMLab - Robotics and Intelligent Machines Laboratory
Dipartimento di Ingegneria dell'Informazione
University of Parma, Italy
E-mail {dlr,caselli}@ce.unipr.it

***Abstract*— We illustrate a method that performs scan matching by maximizing the intersection area of the scans. The intersection area is a robust parameter that is less prone to measurement errors with respect to alternative techniques. Furthermore, such technique does not require to associate each point of one scan to a point of the other one like in some popular algorithms. The relative pose that maximizes the overlap is estimated iteratively. Since the scans are represented by star-shaped polygons due to visibility properties, their intersection can be computed using an efficient linear-time traversal of the vertices. Then, the relative pose is updated under the hypothesis that the combinatorics of intersection is left unchanged and the procedure is repeated until the scans are aligned with sufficient precision.**

I. INTRODUCTION

Scan registration is a common approach to improve the estimation of robot motion over a noisy odometry. The popularity of registration techniques is mainly due to the proliferation of laser range finders that have become part of standard sensor equipment of mobile robots in the last two decades. Furthermore, scan matching is a standard primitive for laser-based simultaneous localization and mapping (SLAM) applications [1], [2]. The underlying assumption of scan matching techniques is the partial overlapping of the environment region observed by the robot in two different locations. The pose displacement between two robot locations is evaluated by finding the rigid motion that overlaps the two laser scans.

Scan matching methods perform the estimation by iterating the association and the optimization steps. During the association, some geometric elements identifying places in each scan are detected and paired with similar elements in the other scan. The elements can be points, geometric features like lines, or the cells of a histogram or of a grid map, as discussed in section II. The similarity between elements is measured by a specific metric. Point-wise association is the most popular one, since points immediately correspond to the range measurements and are pretty generic elements.

Since the detection of correspondences between scans is a difficult and critical task, it would be convenient to avoid such operation and to find a different metric to estimate the quality of the placement. Since each scan represents the free space enclosed by obstacles that can be observed from a location, an area-based criterion can be adopted to match the shapes. The area of the overlap of two scans is a global parameter that does not depend on data association. Furthermore, the area of overlap is a continuous function with respect to the relative placement of the scans [3], [4] and, thus, it is less prone to measurement errors and approximations. A scan can be effectively represented by a polygonal shape. Scan matching therefore can be formulated as the problem of finding the rigid motion that maximizes the overlap of two polygons. Unfortunately, no exact solution is known for the general case [5], but the specific conditions of scan matching problem may be exploited.

In this paper, we present a method to estimate the relative pose that maximizes the area of overlap of two *star-shaped polygons*. The scan data acquired by a range finder can be represented by star-shaped polygons due to operational and visibility condition of the observation. Thus, the proposed algorithm provides a solution of the scan matching problem. The estimation is performed iteratively. First, the boundary of the intersection is detected using an efficient linear-time traversal of the two circular lists of the polygon vertices. The feasibility of such exploration is proved by exploiting the star-shapeness of the two data. Then, the relative pose of the two polygons is updated under the hypothesis that the combinatorial type of the placement, i.e. the set of pairs of vertices or edges that intersect one another, remains unchanged. The expression of the intersection area is derived and used to estimate the rigid motion that maximizes the overlap. These operations are repeated until the alignment is achieved with sufficient precision.

The paper is organized as follows. Section III illustrates the algorithm to detect the intersection of two star-shaped polygons. Section IV provides the formulation of the area-of-overlap problem and its approximate solution. Section V reports experiments to assess the proposed scan matching algorithm. Finally, section VI gives conclusion remarks.

II. RELATED WORK

Several scan registration techniques have been proposed in the last two decades. Approaches can be roughly classified according to the elements that are associated in the two scans. Thus, there are *point-to-point*, *point-to-feature*, *feature-to-feature* and *correlation* based scan matching methods.

The point-wise approach is the most popular one since the proposal of *iterative closest point* (ICP) algorithm [6] and *iterative dual correspondence* (IDC) [7]. ICP associates

each point of the current scan with the closest point on the polygonal line enclosing the reference scan, computes the pose displacement and iterates the two operations. The IDC improves ICP by proposing a different association criterion based on adjacent range points for the estimation of rotation. Several variants of ICP have been proposed to improve specific parts of the algorithm like point weighting [8], metric [9] and polar-oriented association [10].

Few methods perform scan matching by pairing features extracted from laser scans. Segment lines are the most common patterns that can be detected from planar points. Cox's algorithm [11] can be considered the first example of feature-based scan matching algorithm, since it matches the points of a scan with the lines extracted from the previous scan. HAYAI (high-speed and yet accurate indoor-outdoor tracking) [12] provides a generic closed-form and efficient computation of rigid motion based on features. The method in [13] detects segments and matches their endpoints, after a careful evaluation of visibility constraints.

Correlation-based methods perform scan matching by comparing occupancy grids or other discretized representations of scans. These approaches have been originally developed for range sensors with low angular resolution like sonars. Duckett *et al.* [14] present a similar grid-based alignment technique that independently compares the occupancy histograms for each position coordinate. Olson [15] improves the correlative techinque using multi-resolution cell refinement. *Hough spectrum scan matching* (HSM) [16] exploits translation invariance of the Hough spectrum to compute first orientation shift and then position displacement of two scans. Hough spectrum invariance is due to the invariance in the collinearity of points: if some points lie on a line, after a translation they will lie on another line with the same slope.

III. INTERSECTION OF STAR-SHAPED POLYGONS

This section provides a formulation of the problem of estimating the rigid motion corresponding to the best overlap of a pair of consecutive range scans. The proposed method measures the quality of the alignment as the intersection area of the two shapes. Each scan represents an information on the free space visible from the sensor field of view and it is represented by a *star-shaped polygon*. The pairwise alignment of scans is estimated through geometric operations of intersection and computation of the intersection area.

A. Polygonal Representation of Laser Scans

The measurements acquired by a range finder provide a representation of the free space of the environment that can be observed from the current robot location. The free space can be described by a simple polygon, whose vertices are the points corresponding to the sensor measurements. If a sequence of consecutive points are collinear, it is convenient to keep only the endpoints of such interval to avoid degenerate polygons. This can be easily performed using the Ramer-Douglas-Peuker algorithm [17].

Each polygon P is represented by a sequence of n vertices $p_0, p_1, \ldots, p_{n-1}$ in which each vertex has real valued x and y coordinates $(p_{i,x}, p_{i,y})$. The sequence of vertices is stored in a circular list of vertices sorted in *counterclockwise* order, and the value of the index i is taken modulo n so that $p_n \equiv p_0$. Since the vertices are sorted counterclockwise, the direction of each edge $p_{i-1}p_i$ is such that the interior of P lies on the left of the edge. Further properties, which require the following definition, hold for all the polygons built using range measurements.

Definition 1: Given a simple polygon P with vertices $p_0, \ldots, p_{n-1}$ (counter-clockwise order), the *kernel* of P, $\ker P$, is defined as the set of the points p_{org} of P such that for each point p of P the segment $p_{org}p$ lies entirely within P or, formally,

$$\ker P \triangleq \{p_{org} \in P | p_{org}p \in P,\ \forall p \in P\} \tag{1}$$

A polygon P is *star-shaped* if and only if $\ker P$ is not empty.

Hence, the kernel is the set of the points p_{org} such that the segment connecting p_{org} with any other point $p \in P$ belongs to P. Since each point of a scan is obtained by measuring the time-of-flight of the path from the center of the sensor to an obstacle (and the inverse path), every point of the polygon representing the scan can be connected to the center of the sensor. Thus, the kernel of such polygon contains the center of the sensor and the polygon is star-shaped. This property will be exploited to compute efficiently the intersection of two polygons.

The area of a polygon and several other geometric operations can be easily computed using the signed area of a triangle. Given three points p, q and r, the signed area of the triangle with the three points as vertices is given by

$$\begin{aligned} \mathcal{A}(p,q,r) &\triangleq \frac{1}{2}\ \det\left(\begin{bmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{bmatrix}\right) \\ &= \frac{(q_x - p_x)\ (r_y - p_y) - (q_y - p_y)\ (r_x - p_x)}{2} \end{aligned} \tag{2}$$

The sign of $\mathcal{A}(p,q,r)$ is positive if p, q and r are counterclockwise, null if p, q and r lie on the same line, and negative otherwise. A well known elementary proposition states that, given an arbitrary point q, the area of a polygon P is equivalent to the sum of the signed areas of the triangles from each edge $p_{i-1}p_i$ to q (usually $q = 0$):

$$\begin{aligned} \mathcal{A}(P) &= \sum_{i=0}^{n-1} \mathcal{A}(q, p_i, p_{i+1}) = \frac{1}{2} \sum_{i=0}^{n-1} (p_{i,x}\ p_{i+1,y} - p_{i,y}\ p_{i+1,x}) \\ &= \frac{1}{2} \sum_{i=0}^{n-1} \begin{bmatrix} p_{i,x} & p_{i,y} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{J} \begin{bmatrix} p_{i+1,x} \\ p_{i+1,y} \end{bmatrix} \end{aligned} \tag{3}$$

If the polygon P is a star-shaped polygon and $\mathcal{A}(P)$ is computed as the sum of the triangles built on the edges and on a point p_{org}, the signed area of all such triangles is positive.

It is convenient to define a function to check whether a point p strictly lies inside triangle $\widehat{p_0p_1p_2}$ where p_0, p_1 and p_2 are its vertices.

$$\mathcal{T}(q, p_0p_1p_2) \triangleq \begin{cases} 1 & \text{if } \mathcal{A}(p_i, p_{i+1}, q) > 0 \text{ with } i = 0, 1, 2 \\ -1 & \text{if } \mathcal{A}(p_i, p_{i+1}, q) < 0 \text{ with } i = 0, 1, 2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Hence, point q lies inside the triangle *iff* function $\mathcal{T}(q, p_0p_1p_2) \neq 0$.

Algorithm 1: Star-shaped polygon intersection

Data: $p_0, \ldots, p_{n-1}$: vertices of polygon P;
$q_0, \ldots, q_{m-1}$: vertices of polygon Q;
p_{org}: point of $\ker P \cap Q$; q_{org}: point of $\ker Q$;
Result: $\mathcal{L}$: list of vertices on the boundary of intersection;

arbitrary initialization of i and j;
$i_c \leftarrow 0$, $j_c \leftarrow 0$;
inside $\leftarrow$ *unknown*;
while *($i_c < n$ or $j_c < m$) and $i_c < 2n$ and $j_c < 2m$* **do**
  `// check intersection of` $p_{i-1}p_i$ `and` $q_{j-1}q_j$
  if $p_{i-1}p_i$ *and* $q_{j-1}q_j$ *intersects in* c_{ij} **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{c_{ij}\}$;
    if inside $=$ *unknown* **then** `// first cross`
      $i_c \leftarrow 0$, $j_c \leftarrow 0$;
    end
    if $\mathcal{A}(q_{j-1}, q_j, p_i) \geq 0$ **then** `//` p_i `left` $q_{j-1}q_j$
      inside $\leftarrow$ *inP*;
    else `//` q_j `left` $p_{i-1}p_i$
      inside $\leftarrow$ *inQ*;
    end
  end
  `// Advance rules for indices` i `and` j
  turnP $\leftarrow \mathcal{A}(p_{org}, p_i, q_j)$;
  turnQ $\leftarrow \mathcal{A}(q_{org}, p_i, q_j)$;
  inTriaP $\leftarrow \mathcal{T}(q_j, p_{org}p_iq_{org})$;
  inTriaQ $\leftarrow \mathcal{T}(p_i, q_{org}q_jp_{org})$;
  if *(*turnP > 0 *and* turnQ > 0*) or* inTriaQ $\neq 0$ **then**
    $i \leftarrow i + 1$;
  end
  else if *(*turnP < 0 *and* turnQ < 0*) or* inTriaP $\neq 0$ **then**
    $j \leftarrow j + 1$;
  end
  else $i \leftarrow i + 1$;
  $i_c \leftarrow i_c + 1$, $j_c \leftarrow j_c + 1$;
end

B. Intersection Area of Polygons

Now we will consider the problem of finding the intersection area of two star-shaped polygons P and Q. In general, the intersection of two simple polygons that are not convex is not a polygon, but a set of simple polygons or *intersection polygons*. Each intersection polygon corresponds to a connected region of the intersection set. Furthermore, in order to compute the intersection area we need to find the edges of P that lie inside Q and vice-versa, and the crossing edges. The intersection points of the edges of P and Q can be found in $O((n+m)\log(n+m)+k)$ using a modified version of the classic Bentley-Ottman algorithm [18].

However, a specific intersection polygon defined by the intersection of two star-shaped polygons P and Q can be computed in linear time under the restrictive hypothesis that at least one point $p_{org} \in \ker P$ lies in the interior of polygon Q (swap P with Q if it exists $q_{org} \in \ker Q \cap P \neq \emptyset$). If p_{org} lies inside polygon Q, the connected intersection region $I \subset P \cap Q$ s.t. $p_{org} \in I$ is a star-shaped polygon since p_{org} lies on the left of all the edges $q_{j-1}q_j$ of Q that intersect the boundary of I. Hence, all the intersections between an edge $p_{i-j}p_i$ of P and an edge $q_{i-j}q_j$ of Q that occurs on the boundary of I can be sorted around p_{org}. The consecutive edge intersection is then found advancing edge $p_{i-j}p_i$, edge $q_{i-j}q_j$ or both. Such condition is consistent with the scan matching problem: to estimate the relative pose of a laser scan P with respect to another scan Q, the observation point of P must lie in the free space represented by Q.

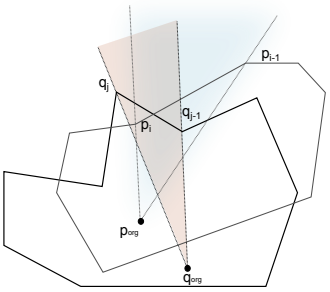

Fig. 1. Edges $p_{i-1}p_i$ and $q_{j-1}q_j$ intersect each other only if $p_{i-1}p_i$ lies in wedge $q_{j-1}\widehat{q_{org}}q_j$ and $q_{j-1}q_j$ lies in wedge $p_{i-1}\widehat{p_{org}}p_i$.

The proposed approach is inspired by the algorithm for finding the intersection of two convex polygons illustrated in [19]. Two indices i and j that point respectively to edges $p_{i-1}p_i$ of P and $q_{j-1}q_j$ of Q, are defined and at each iteration the intersection of $p_{i-1}p_i$ and $q_{j-1}q_j$ is tested. In order to avoid an exhaustive search of intersection (complexity $O(nm)$), the indices i and j must be updated carefully.

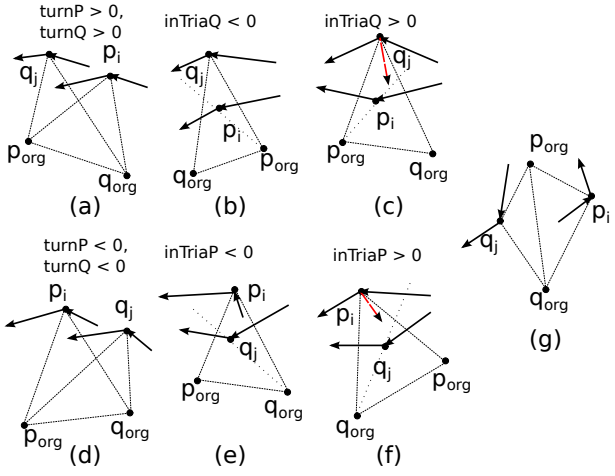

Fig. 2. Relative placements of vertices p_i and q_j according to the values of variables turnP, turnQ, inTriaP and inTriaQ. When cases (a), (b) and (c) occur, index i of vertex p_i is incremented to check the intersections between the edges. Cases (d), (e) and (f) are specular for vertex q_j. In case (g), the choice of the index to increment, either i or j, is arbitrary.

Given a point p_{org} and two rays from the origin p_{org}

to respectively p_{i-1} and p_i, the *wedge* $p_{i-1}\widehat{p_{org}}p_i$ is the region delimited by the two rays. The intersection of the edge $q_{j-1}q_j$ of Q and the edge $p_{i-1}p_j$ of P occurs only if $q_{j-1}q_j$ partially lies in the wedge $p_{i-1}\widehat{p_{org}}p_i$ or $p_{i-1}p_i$ partially lies in the wedge $q_{j-1}\widehat{q_{org}}q_j$ (see Figure 1). The vertices of P and Q can be cyclically sorted according to the turn around respectively $p_{org} \in \ker P$ or $q_{org} \in \ker Q$. Since the intersecting edge pairs can be sorted as assumed before, the occurrence of next intersection is found by incrementing the index of the current point (either p_i or q_j) that lies inside or on the right of the other wedge. The intuitive idea of our intersection algorithm is to update the indices i and j so that the corresponding wedges have an intersection. The following variables support the decisions of the algorithm

$$\mathrm{turnP} \triangleq \mathcal{A}(p_{org}, p_i, q_j) \quad (5)$$

$$\mathrm{turnQ} \triangleq \mathcal{A}(q_{org}, p_i, q_j) \quad (6)$$

$$\mathrm{inTriaP} \triangleq \mathcal{T}(q_j, p_{org}p_iq_{org}) \quad (7)$$

$$\mathrm{inTriaQ} \triangleq \mathcal{T}(p_i, q_{org}q_jp_{org}) \quad (8)$$

Figure 2 illustrates the configurations corresponding to the values of such variables. If both the signs of turnP and turnQ are positive, then p_i is on the right of both $p_{org}q_j$ and $q_{org}q_j$ (Figure 2(a)). Thus, p_i is moved forward to make the wedges $p_{i-1}\widehat{p_{org}}p_i$ and $q_{j-1}\widehat{q_{org}}q_j$ intersect. A similar reasoning is applied when turnP and turnQ are both negative (Figure 2(d)). In the case where turnP and turnQ have opposite signs, the relative position of p_i, q_j and p_{org} and q_{org} is considered. In particular, a relevant configuration occurs when current vertex p_i lies inside the triangle $q_{org}q_jp_{org}$ or, similarly, q_j lies inside the triangle $p_{org}p_iq_{org}$. These cases occurs when respectively $\mathrm{inTriaQ} \neq 0$ or $\mathrm{inTriaP} \neq 0$. Although p_i lies inside the triangle $q_{org}q_jp_{org}$ independently from the sign of variable $\mathrm{inTriaQ}$, the sign discriminates between two different sub-cases. If $\mathrm{inTriaQ} < 0$ (Figure 2(b)), p_i lies on the right of $q_{org}q_j$ and the potential next intersection is detected by checking edge p_ip_{i+1} and its successors. When $\mathrm{inTriaQ} > 0$ and without any other specific assumption, a general decision rule could not be stated since vertex q_j may turn positive or negatively around p_{org} (see the black or red edges from q_j in Figure 2(c)). However, since we assume that all the edges that intersect $P \cap Q$ turn left around p_{org} and the intersection polygon, the latter case cannot occur. The discussion of the sign of $\mathrm{inTriaP}$ is similar. The remaining case shown in Figure 2(g) requires only the alignment of the two vertices p_i and q_j. Hence, we can summarize the above discussion with the following proposition.

Proposition 1: Let P and Q be star-shaped polygons respectively with vertices $p_0, \ldots, p_{n-1}$ and $q_0, \ldots, q_{m-1}$, $p_{org} \in \ker P \cap Q$ and $q_{org} \in \ker Q$. If the boundaries of P and Q intersect non-degenerately, Algorithm 1 will find the boundaries of the intersection of P and Q with the vertices.

Algorithm 1 visits the vertices of the polygon exploiting the criterion suggested above. Indices i and j are initialized with arbitrary values and i_c and j_c are the counters of the number of explored edges used for the stop condition. At each iteration the algorithm increments i or j (lines 16-26) according to the illustrated advance rules. The intersection of edges $p_{i-1}p_i$ and $q_{j-1}q_j$ is checked and every intersection point is added to the vertices list $\mathcal{L}$ of the boundary of the intersection area. The flag inside is equal to inP or inQ if the latest vertex of $\mathcal{L}$ belongs respectively to P or Q (lines 5-15). The algorithm requires a segment intersection test that handles also the degenerate intersections and in particular when two intersecting edges are collinear.

IV. ESTIMATION OF RELATIVE POSE

The aim of scan matching is to compute the rigid motion that achieves the best overlap of two laser scans. Using the polygonal representation of scans illustrated in the previous section the quality of the overlap can be measured as the intersection area of two star-shaped polygons. Let P and Q be two polygons represented by a list of vertices, respectively $p_0, \ldots, p_{n-1}$ and $q_0, \ldots, q_{m-1}$, $t = [x, y]^T \in \mathbb{R}^2$ a translation vector, $\theta \in S^1$ a rotation angle and φ be an operator that applies a planar rigid motion $\varphi = [t^T\ \theta]^T$ to a single point or a set of points. The coordinates of the two polygons are usually given w.r.t. a reference frame fixed on Q. Symbol φ refers to both the operator and the rigid motion vector with an abuse of notation and will be also called *placement* in the following. Observe that $\varphi(\partial P) = \partial(\varphi P)$ (∂ is the border) and then one can simply reason on the boundary of the set consisting of the vertices $\bar{p}_0, \ldots, \bar{p}_{n-1}$ defined as

$$\bar{p}_i = \begin{bmatrix} \bar{p}_{i,x} \\ \bar{p}_{i,y} \end{bmatrix} = \begin{bmatrix} x + p_{i,x}\ \cos\theta - p_{i,y}\ \sin\theta \\ y + p_{i,x}\ \sin\theta + p_{i,y}\ \cos\theta \end{bmatrix} = t + R_\theta\ p_i \quad (9)$$

where R_θ is the rotation matrix of an angle θ. Thus, the best overlap is the problem of finding φ^{opt}

$$\varphi^{opt} = \underset{\varphi}{\operatorname{argmax}}\, \mathcal{A}(\varphi P \cap Q) \quad (10)$$

The estimation of φ^{opt} is difficult due to the high number of combinatorially distinct placements of polygon P with respect to Q. Two placements φ_0 and φ_1 are combinatorially equivalent when the corresponding edges of $\varphi_0 P$ and $\varphi_1 P$ intersect the same edge of Q. Following and extending the definitions given in [3], the combinatorial equivalence is formalized using the notion of critical placement.

Definition 2: A placement φ is said to be *critical* if either a vertex p of φP lies on an edge $q_{j-1}q_j$ of Q or a vertex q of Q lies on an edge $\bar{p}_{i-1}\bar{p}_i$ of φP, i.e. $\mathcal{A}(q_{j-1}, q_j, p) = 0$ or $\mathcal{A}(\bar{p}_{i-1}, \bar{p}_i, q) = 0$. Two placements φ_0 and φ_1 are *equivalent* if exists a smooth function $\varphi : [0, 1] \to \mathbb{R}^2 \times S^1$ such that $\varphi(0) = \varphi_0$ and $\varphi(1) = \varphi_1$ and $\varphi(s)$ is not a critical placement $\forall s \in [0, 1]$.

Thus, the result of Algorithm 1 applied to Q and φP changes only at a critical placement. An approximate estimation can be computed iteratively by assuming that no critical placement is found and evaluating only the area of the connected component containing p_{org}. Under such

hypothesis and given the Algorithm 1, the mathematical expression of the area can be computed.

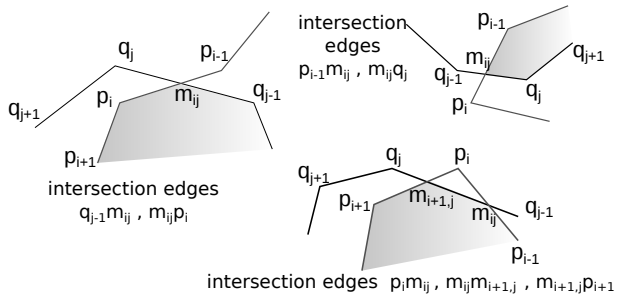

Fig. 3. Examples of intersection edges defined by the intersection of the boundaries of P and Q.

The area of a polygon can be decomposed into the area of the triangles built connecting an external point and each edge of the polygon as reported in equation (3). The connected intersection polygon $V^\varphi \subset \{\varphi P \cap Q \text{ s.t. } p_{org} \in V^\varphi\}$ is defined by the list of vertices $\mathcal{L}_\varphi = \{v_0^\varphi, \ldots, v_{n_v-1}^\varphi\}$ returned by Algorithm 1. (Apex φ is often omitted when the placement is clear from the context). Each vertex v_k is either a vertex of $\bar{p}_i$ of φP lying inside of Q, a vertex q_j of Q lying inside of φP, or the intersection point m_{ij} of two edges $q_{j-1}q_j$ and $\bar{p}_{i-1}\bar{p}_i$. Figure 3 illustrates different conditions of intersection. The cartesian coordinates of point m_{ij} are given by the following formula

$$m_{ij} = \underbrace{\frac{\mathcal{A}(0, \Delta q_j, \bar{p}_i - q_{j-1})}{\mathcal{A}(0, \Delta q_j, \Delta \bar{p}_i)}}_{\alpha_{ij}^i} \bar{p}_{i-1} - \underbrace{\frac{\mathcal{A}(0, \Delta q_j, \bar{p}_{i-1} - q_{j-1})}{\mathcal{A}(0, \Delta q_j, \Delta \bar{p}_i)}}_{\alpha_{ij}^{i-1}} \bar{p}_i \tag{11}$$

where the short notation $\Delta q_j = q_j - q_{j-1}$, $\Delta p_i = p_i - p_{i-1}$ and $\Delta \bar{p}_i = \bar{p}_i - \bar{p}_{i-1}$ has been adopted. If polygon P is continuously moved from a placement φ_0 to φ_1 without meeting a critical placement, then the equation of $\mathcal{A}(V^\varphi)$ does not change. Our aim is to estimate the rigid motion that maximizes the area under such constraint. Since the closed-form expression is difficult to estimate, an approximate gradient descent can be used instead. The gradient of the intersection area $\mathcal{A}(V)$ w.r.t. $\varphi = [t, \theta]^T$ is equal to

$$\frac{\partial \mathcal{A}(V)}{\partial(t,\theta)} = \frac{1}{2} \sum_{k=1}^{n_v-1} \frac{\partial}{\partial(t,\theta)} \left(v_k^T \; J \; v_{k+1} \right) \tag{12}$$

$$= \frac{1}{2} \sum_{k=1}^{n_v-1} \left(v_{k+1}^T \; J^T \; \frac{\partial v_k}{\partial(t,\theta)} + v_k^T \; J \; \frac{\partial v_{k+1}}{\partial(t,\theta)} \right) \tag{13}$$

The Jacobians $\partial v_k / \partial(t, \theta)$ depend on the type of vertex v_k. If vertex $v_k \equiv q_j$ where q_j a vertex of Q, then the corresponding Jacobian is null since the position of q_j does not depend on the placement. If $v_k \equiv \bar{p}_i = R_\theta p_i + t$ where $\bar{p}_i$ a vertex of P, then the Jacobian is equal to $[I_2 \quad \dot{R}_\theta p_i]$ where $\dot{R}_\theta$ is the derivate of rotation matrix R_θ. If v_k is an intersection point m_{ij}, then the Jacobian of $v_k = m_{ij}$ is obtained by differentiating the equation (11) as

Dataset	Translational error [m]		Rotational error [rad]	
	ICP SM	Area SM	ICP SM	Area SM
Aces	0.41 ± 1.80	0.46 ± 1.93	0.02 ± 0.07	0.01 ± 0.04
Intel	1.00 ± 1.35	1.67 ± 2.50	0.06 ± 0.09	0.06 ± 0.10
Mit K.	7.13 ± 18.23	8.54 ± 24.81	0.09 ± 0.21	0.12 ± 0.29
Mit C.	1.13 ± 3.68	1.62 ± 5.40	0.05 ± 0.14	0.07 ± 0.19
Freib. 79	0.49 ± 1.03	0.46 ± 0.69	0.07 ± 0.13	0.06 ± 0.11

TABLE I

ERRORS ON TRANSLATION AND ROTATION OF ICP AND AREA OVERLAP SCAN MATCHING.

$$M_{ij} = \frac{\partial m_{ij}}{\partial(t,\theta)} = A_{ij}^i \; \bar{p}_{i-1} + \alpha_{ij}^i \; \begin{bmatrix} I_2 & \dot{R}_\theta p_{i-1} \end{bmatrix} - A_{ij}^{i-1} \; \bar{p}_i - \alpha_{ij}^{i-1} \; \begin{bmatrix} I_2 & \dot{R}_\theta p_i \end{bmatrix} \tag{14}$$

$$A_{ij}^* = \frac{\partial \alpha_{ij}^*}{\partial(t,\theta)} = \frac{\Delta q_j^T J}{\Delta q_j^T J R_\theta \Delta p_i} \begin{bmatrix} I_2 & \dot{R}_\theta p_* \end{bmatrix} - \frac{\Delta q_j^T J (\bar{p}_* - q_{j-1})}{(\Delta q^T J R_\theta \Delta p_i)^2} \Delta q_j^T J \begin{bmatrix} 0_{2\times 2} & \dot{R}_\theta \Delta p_i \end{bmatrix} \tag{15}$$

The symbol $*$ in the above formula is referred to index $i-1$ or i depending on the context.

Thus, the φ^{opt} is estimated by performing iteratively the following procedure. Given an approximate estimate of the placement $\hat{\varphi} = [\hat{t}^T, \hat{\theta}]^T$, the vertices list $\mathcal{L}^{\hat{\varphi}}$ is estimated using Algorithm 1 and the Jacobian $\partial v_k / \partial(t, \theta)$ is associated to each vertex. The gradient is then computed according to equation (13) and used to compute the new value of $\hat{\varphi}$. The step size of the update is decreased with the numer of iterations.

V. RESULTS

In this section, the correctness of the proposed scan matching algorithm is assessed using the SLAM benchmarking criterion illustrated in [20] and the related datasets. In particular, we used datasets acquired in the ACES building at the University of Texas (Aces), in Intel Research Lab (Intel), in MIT Killian Court (Mit K.), in MIT CSAIL (Mit C.)and in building 079 at the University of Freiburg (Freib. 79) [1]. Each dataset consists of the sensor data log file and a relation file that contains information on the groundtruth. In particular, the relation file contains an accurate estimation of the pose displacement between reference poses that allow the estimation of the position and orientation errors in the map construction. Such collection of datasets covers several typologies of indoor environments from office-like cluttered rooms to long corridors. Figure 4 shows an example of the correction performed on a subset of the scans of dataset Intel.

The proposed algorithm has been compared with the ICP algorithm implemented in the *Canonical Scan Matcher* [21]. The translational and rotational errors as defined in [20] and the corresponding standard deviation values are illustrated in Table I. The ICP scan matcher seems to better estimate the position of the robot than the proposed method, but even

[1] See http://ais.informatik.uni-freiburg.de/slamevaluation/.

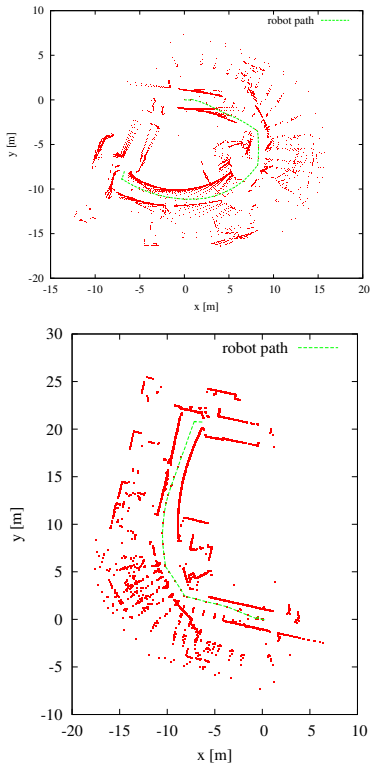

Fig. 4. Map built with the first 1000 scans of Intel dataset using odometry (top) or the area overlap scan matching (bottom).

with the current version of the area overlap-based approach results are quite comparable with [20]. The execution of the proposed scan matching algorithm requires on average $38\ ms$ to perform the alignment of two scans on Intel Core Duo processor T2300 with 1 GB memory, while the ICP is requires about $46\ ms$. Thus, the proposed approach is slightly faster than ICP.

VI. CONCLUSION

In this paper, we have presented an algorithm that performs scan matching by maximizing the intersection area of the range data. The range data are represented by polygons that are star-shaped due to the visibility condition of the observation. The intersection area measures the quality of the overlap. The estimation of the best relative pose is performed iteratively. First, the boundary of the intersection is found using an efficient linear-time traversal of the two circular lists of the polygon vertices. The correctness of the proposed exploration algorithm is proved by exploiting the fact that the polygons are star-shape. Then, the relative pose of the two polygons is updated under the hypothesis that the configuration of the intersection, i.e. the set of pairs of vertices or edges that intersect one another, remains unchanged. Under such hypothesis the equation of the intersection area is provided and used to find the rigid motion that maximizes the overlap. These operations are repeated until the scan alignment is performed with sufficient precision.

The proposed approach has the advantage of avoiding the need to detect correspondences between scans. However, the estimation of the best overlap is difficult due to combinatorial complexity of the polygon intersection. In future works, we will investigate the combinatorial structure of the intersection area in order to improve the optimization.

REFERENCES

[1] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999, pp. 318–325.

[2] G. Grisetti, D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate SLAM with Rao-Blackwellized Particle Filters," *Journal of Robotics & Autonomous Systems*, vol. 55, pp. 30–38, 2007.

[3] D. Mount, R. Silverman, and A. Wu, "On the area of overlap of translated polygons," *Computer Vision and Image Understanding*, vol. 64, no. 1, pp. 53–61, 1996.

[4] A. Vigneron, "Geometric optimization and sums of algebraic functions," in *Proc. ACM–SIAM Sympos. Alg.*, 2010, pp. 906–917.

[5] H. Ahn, O. Cheong, C.-D. Park, C. Shin, and A. Vigneron, "Maximizing the overlap of two planar convex sets under rigid motions," *Comp. Geom. Theory Appl.*, vol. 37, pp. 3–15, 2007.

[6] P. Besl and H. Mckay, "A method for registration of 3-d shapes," *IEEE Trans. Pat. Anal. Mach. Intel*, vol. 14, no. 2, pp. 239–256, 1992.

[7] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1994, pp. 935–938.

[8] S. Pfister, K. Kriechbaum, S. Roumeliotis, and J. Burdick, "Weighted range sensor matching algorithms for mobile robot displacement estimation," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002, pp. 1667–1674.

[9] J. Minguez, F. Lamiraux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005, pp. 3557–3563.

[10] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *Int. Journal of Robotics Research*, vol. 26, no. 10, pp. 1125–1153, 2007.

[11] I. Cox, "Blanche–an experiment in guidance and navigation of autonomous robot vehicle," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 193–204, 1991.

[12] K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg, "High-speed laser localization for mobile robots," *Journal of Robotics & Autonomous Systems*, vol. 51, no. 4, pp. 275–296, 2005.

[13] A. A. Aghamohammadi, H. D. Taghirad, A. H. Tamjidi, and E. Mihankhah, "Feature-based laser scan matching for accurate and high speed mobile robot localization," in *Proc. of the European Conference on Mobile Robots (ECMR)*, 2007, pp. 253–258.

[14] T. Duckett and U. Nehmzow, "Mobile robot self-localisation using occupancy histograms and a mixture of gaussian location hypotheses," *Journal of Robotics & Autonomous Systems*, vol. 34, no. 3, pp. 119–130, March 2001.

[15] E. Olson, "Real-time correlative scan matching," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009, pp. 4387–4393.

[16] A. Censi, L. Iocchi, and G. Grisetti, "Scan matching in the Hough domain," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[17] D. Douglas and T. Peuker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartogr. Int. J. Geogr. Inf. Geovis.*, vol. 10, no. 2, pp. 112–122, Dec 1973.

[18] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry – Algorithms and Applications, 2nd ed.* Springer-Verlag, 2000.

[19] J. O'Rourke, C. Chien, T. Olson, and D. Naddor, "A new linear algorithm for intersecting convex polygons," *Comput. Graph. Image Process.*, vol. 19, no. 4, pp. 384–391, Aug 1982.

[20] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Journal of Autonomous Robots*, vol. 27, no. 4, pp. 387–407, 2009.

[21] A. Censi, "An ICP variant using a point-to-line metric," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008.