Visibility Maps for Any-Shape Robots*

Tiago Pereira[1], Manuela Veloso[2] and António Moreira[3]

Abstract— **We introduce in this paper visibility maps for robots of any shape, representing the reachability limit of the robot's motion and sensing in a 2D gridmap with obstacles. The brute-force approach to determine the optimal visibility map is computationally expensive, and prohibitive with dynamic obstacles. We contribute the Robot-Dependent Visibility Map (RDVM) as a close approximation to the optimal, and an effective algorithm to compute it. The RDVM is a function of the robot's shape, initial position, and sensor model. We first overview the computation of RDVM for the circular robot case in terms of the *partial morphological closing* operation and the optimal choice for the critical points position. We then present how the RDVM for any-shape robots is computed. In order to handle any robot shape, we introduce in the first step multiple layers that discretize the robot orientation. In the second step, our algorithm determines the frontiers of actuation, similarly to the case of the the circular robot case. We then derive the concept of *critical points* to the any-shape robot, as the points that maximize expected visibility inside unreachable regions. We compare our method with the ground-truth in a simulated map compiled to capture a variety of challenges of obstacle distribution and type, and discuss the accuracy of our approximation to the optimal visibility map.**

I. INTRODUCTION

Scheduling of tasks and motion planning are common problems in robotics, usually associated with goal positions in a map. Planning for mobile robots has been widely studied, and many efficient algorithms are known for a wide variety of problems. For a given environment, the ability for a robot to execute a task depends on its own physical characteristics. Hence, planning needs to determine if a task is feasible for a specific robot. In our work, we further consider reconnaissance tasks, where robots do not need to necessarily reach the goal position, as long as they can sense it, i.e., as long as the goal is within the robot's "visibility".

The ground-truth visibility map can be obtained using a brute force approach through ray casting, determining for each point in the map if there is a reachable robot position from where it is visible. However, the brute force approach is inefficient as it is computationally expensive, making it difficult to use in dynamic environments or when determining visibility for a wide range of robot characteristics.

In our previous work, we presented an alternative that gives a very close approximation to the ground-truth, significantly outperforming brute force in terms of computation time [1]. We used morphological operations to represent the actuation space of the robot, and then determined the visibility inside unreachable regions, introducing the concept of critical points. Considering only visibility from these points allowed to have a much faster and scalable computation of the visibility map, although it resulted in an approximation of the true visibility map. Nevertheless, with a wise choice of critical points, we could obtain a very close approximation to the ground-truth. Throughout that work we considered the assumption that robots have a circular shape.

In this work we extend that technique to determine effectively the visibility map to any-shape robots. So, instead of the visibility map depending on the robot size and maximum sensing range, it is now a function of the robot shape, initial position, and sensor model, with a parametrization that allows to represent many different robots. We determine the overall sensing capability of a robot in a 2D gridmap (a discretized representation of the environment), as a function of the robot. Our algorithm determines the Robot-Dependent Visibility Map (RDVM), i.e., what can be sensed from some point that is reachable from the robot initial position.

The first step of our algorithm uses the partial morphological closing operation in order to obtain the actuation space. However, because the robot model is not rotation invariant, we discretize orientation. We contribute a multi-layer representation of the environment, and apply the partial morphological operation to each layer, so the actuation space is determined for each possible orientation. After projecting all the layers together, we determine the unreachable regions and frontiers in a similar fashion to the circular robot case.

In the second step of our algorithm, we change the definition of critical points to the more general any-shape robot case. While before the critical point was chosen as the closest point to the frontier, here that solution does not work because the visibility inside unreachable regions depends not only on the 2D position of the critical point, but also on its orientation and sensor model. Therefore, we extend our previous idea that the critical point should maximize the expected visibility inside unreachable regions, in order to improve the accuracy of our approximation of the visibility map using only critical points. We contribute an algorithm to search for points that maximize expected utility based on the critical point position and orientation, the frontier extremes,

*This work is supported in part by INESC Porto-Institute for Systems and Computer Engineering of Porto, and by Portuguese Foundation for Science and Technology (FCT), within project POCI-01-0145-FEDER-006961, and under Grant SFRH/BD/52158/2013, through the Carnegie Mellon—Portugal Program managed by ICTI, by ONR grant number N00014-09-1-1031, by AFRL grant number FA87501220291, and by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme. The views and conclusions contained in this document are those solely of the authors.

[1] Tiago Pereira is with the Electrical & Computer Engineering Department, Carnegie Mellon University, and with Faculty of Engineering, University of Porto, Portugal tpereira@andrew.cmu.edu

[2]Manuela Veloso is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA mmv@cs.cmu.edu

[3]A. Moreira is with the ECE Department, Faculty of Engineering, University of Porto, Portugal amoreira@fe.up.pt

and the sensor model. After determining the optimal position for the *critical points*, it is possible to determine the visibility inside non-traversable regions using ray casting, similar to the case of circular robots. This method allows to get a good estimate of the true visibility with less effort than the brute-force approach.

In the next section we briefly discuss related work. We then present in depth our work, explaining how it effectively solves the observability problem with the new representation for any-shape robots. We present illustrative experimental results of our algorithm in a simulated (challenging, multi-featured) environment. Finally, we present our conclusions and the directions for future work.

II. Related Work

Visibility maps can be used as a pre-processing step for planning, improving efficiency in motion planning for perception tasks [1]. We had a circular robot assumption in that work, which we drop here by considering any-shape robots with any sensor model.

Techniques borrowed from image processing have already been used for map transformation, e.g. automatically extracting topology from an occupancy grid [2]. They robustly find the big spaces in the environment like humans would, separating it into regions. Morphological operations have also been used in robotics to determine the actuation space of a robot, used to coordinate multi-robot teams [3].

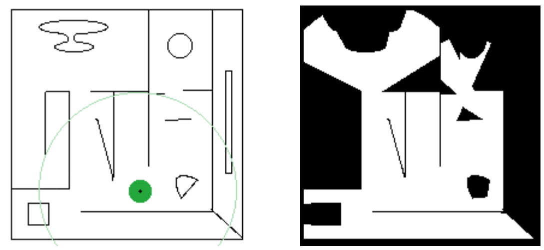
In other work it was proposed that robots maintain reachability and visibility information, both of a robot and a human partner in a shared workspace. However, it uses non-mobile robotic platforms [4]. Visibility graphs are considered in [5], but the focus is on generating points for a motion plan with other goals in mind, such as patrolling. Moreover, it assumes vectorial obstacles, so visibility can easily be calculated using ray casting at the extremes of lines.

Another class of problems for visibility is the inspection problem. In order to determine a path that can sense multiple targets, a neural network approach was used to solve the NP-hard Watchman Routing Problem. In order to do so efficiently, a fast method was proposed to answer visibility queries [6], an approach that has been extended also to 3D [7]. However, queries ask for visibility from one specific point, while in our work we aim at finding the overall visibility from any reachable position to the robot. Therefore, we can have an overall overview of the capacities of robots in terms of perception.

III. Background

In this section we will summarize our previous technique for effective determination of observability in 2D gridmaps, considering circular robots and omnidirectional sensing. Morphological operations are used to obtain the actuation space, and then critical points were introduced to reduce the computational effort at the expense of the output being an Approximate Visibility Map.

In our previous work robots were assumed to have circular shape and a maximum sensing range. The goal was to efficiently determine the observability of a robot in a certain environment, i.e., determine what regions can be sensed from a position that is reachable from the initial robot position [1]. The algorithm is a function of robot size and sensing range. We show in Figure 1 a simulated environment with obstacles, and the Approximate Visibility Map (A-VM).



(a) Map (b) A-VM

Fig. 1. Given a black and white gridmap, an omnidirectional circular robot (green), and a sensing range (green circumference), the A-VM determines what can be sensed from reachable positions.

For that purpose, we used morphological operations, which can be applied on images using a structuring element with a given shape. Here the structuring element is a circle representing a circular robot. The domain is a grid of positions G. The input is a black and white binary image representing the map, with M being the set containing the positions that correspond to obstacles. The structuring element, R, represents the robot. The morphological operation dilation on the obstacle set M by R is

$$M \oplus R = \bigcup_{z \in R} M_z \quad (1)$$

where $M_z = \{p \in G \mid p = m + z, m \in M\}$, i.e., the translation of M by z over the grid G.

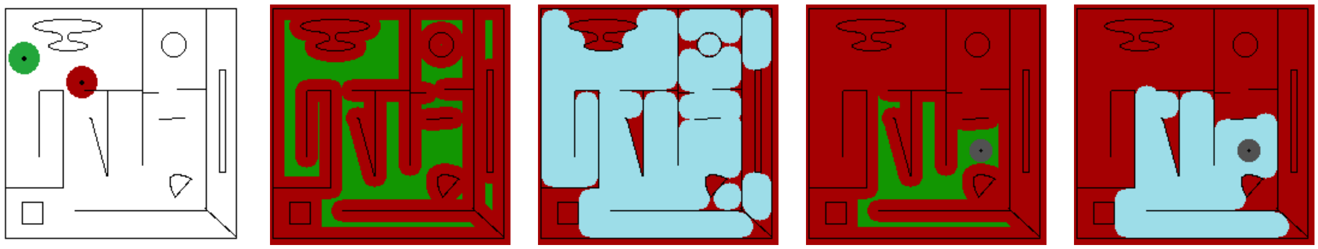
When applying the dilation operation to black points in the image, the algorithm inflates the obstacles of the map by the robot size, achieving the configuration space:

$$C^{free} = \{p \in G \mid p \notin M \oplus R\} \quad (2)$$

Given the free configuration space C^{free}, it is possible to find the points that are reachable from the initial robot position S.

$$Reach(S) = \{p \in C^{free} \mid p \text{ connected to } S\} \quad (3)$$

Morphological closing is the combination of a dilation operation followed by an erosion. Dilation and erosion are dual operations. In order to apply the closing operation to the obstacles, we apply an erosion to the dilated obstacles. But being dual operations, the morphological closing of obstacles is equivalent to the dilation of the free configuration space. In order to consider only positions reachable from the starting position, we use the partial morphological closing, applying the second morphological operation only to the reachable set, $Reach(S)$, instead of C^{free}.



(a) Original Map (b) Dilated Map (c) Closed Map (d) Reachable Space (e) Actuation Space

Fig. 2. In (a) two possible positions for the robot are shown, the green one being feasible, while in the red the robot overlaps with obstacles. This positions correspond to green and red points in the configuration space in (b), obtained by application of the morphological operation dilation to the map (C^{free} is set of green regions). The morphological closing is shown in (c). From the configuration space, the connected parts to the initial position (grey robot) are determined, which results in the reachable space presented in (d). If the second dilation operation is only applied to the reachable space instead of all C^{free}, applying the partial morphological operation to the reachable space, it is possible to determine the *actuation space*, in (e). [1]

$$A(S) = Reach(S) \oplus R \tag{4}$$

The actuation space, $A(S)$, can be seen as a first approximation of the visibility map, if the maximum sensing range considered is less than the robot size (Figure 2).

From $A(S)$, it is possible to define the unreachable regions, i.e., regions that are not reachable to the robot body, and thus cannot be actuated.

$$U(S) = \{p \in G \mid p \notin A(S) \land p \notin M\} \tag{5}$$

$U(S)$ is then divided in a set of different disconnected components $U^l(S)$. The separation of the unreachable regions of the actuation space in disconnected parts is useful, allowing to determine visibility independently (Figure 3). Each region $U^l(S)$ has its unique openings to the actuation space, from where visibility inside $U^l(S)$ is possible. These openings are the frontiers, defined as the points of the unreachable space that connect with $A(S)$:

$$F^l(S) = \{p \in U^l(S) \mid \exists p' : p' \text{ is adjacent to } p \land p' \in A(S)\} \tag{6}$$

The frontier set can be composed of multiple disjoin segments $F^{li}(S)$, and visibility inside the unreachable region should be determined for each segment independently.

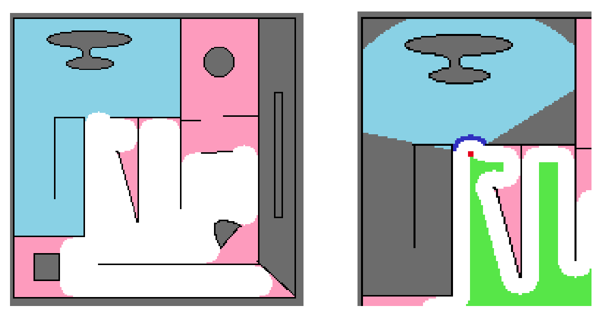
So, when determining visibility for sensing range greater than robot size, it is necessary to find points that have line of sight inside of $U^l(S)$ through $F^{li}(S)$. There are multiple candidate points, and all of them have to be in $Reach(S)$, the feasible positions for the robot center. In order to have the true visibility map, all those points should be considered.

The brute-force solution is computational expensive, so we proposed an alternative, where the visibility inside unreachable regions through each frontier segment is considered only from one point of the reachable space (Figure 3).

As only one point is being used, the final visibility map is an approximation of the ground-truth. In order to obtain a better approximation, the point chosen has to maximize the expected visibility inside the unreachable region. That is accomplished by choosing a point close to the frontier, as being closer is equivalent to having a deeper and wider expected visibility inside $U^l(S)$, maximizing the expected visibility area. In order to find this ideal point, $c^*_{li}(S)$, the sum of the distance to all frontier points is minimized:

$$c^*_{li}(S) = \underset{p \in Reach(S)}{\operatorname{argmin}} \sum_{f \in F^{li}(S)} \|p - f\|^2 \tag{7}$$

Ray casting is then used to test visibility inside unreachable regions from the critical point. As a result, the Approximate Visibility Map is obtained.



(a) Unreachable Regions (b) Critical Point

Fig. 3. In (a) we show $A(S)$ in white, in pink the unreachable regions that connect with $A(S)$, and in blue an example of a disconnected unreachable region $U^l(S)$. In (b) we highlight that disconnected region, showing in dark blue the frontier segment points $F^{li}(S)$, and in red the critical point $c^*_{li}(S)$. The points from the Reachable Space are shown in green. Finally, light blue represents the expected visibility from the critical point, through the frontier, into the unreachable region.

IV. Any-Shape Robot Visibility Map

In order to extend the approach described in the previous section to any-shape robots with any sensor model, we used a multi-layered representation to discretize orientation and a new method to choose the critical points.

A. Actuation Space with Discretized Orientation

First, we parametrized both the robot shape and sensor model. Both are given by images to represent them, that can be rotated and scaled to represent any robot. It is also possible to define the sensor and robot centers, and their relative position.

Here we assumed the discretization of the angle is given by n_θ levels. After the initial parametrization and definition

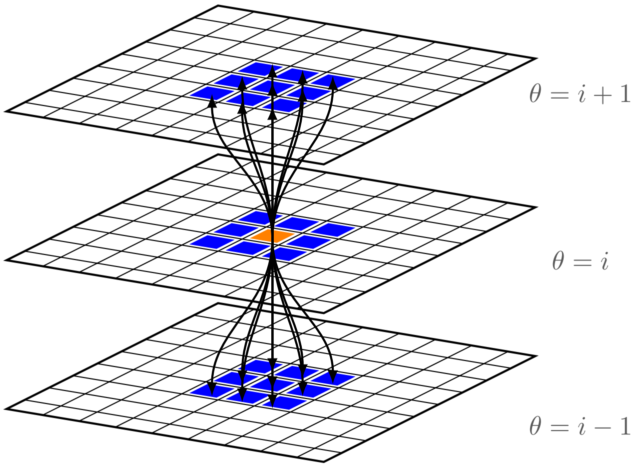


Fig. 4. Layered representation of discretized orientation θ. We consider the neighborhood of the central orange point all the 8 closest locations in the same layer (translation without rotation), and the 9 corresponding points in the previous and next layers (smallest rotation in either direction, plus translation). If no additional motion constraints are considered, this represents an omnidirectional motion model.

Algorithm 1 Clustering: Reachable set from initial position

Require: Free configuration space C^{free}, initial position S

$Reach(S, i) \leftarrow \emptyset \quad \forall i < n_\theta$
$open \leftarrow S$ ▷ Open list of positions to expand
$closed \leftarrow \emptyset$ ▷ list of expanded positions
while $open$ not empty **do**
  $p \leftarrow open$.pop()
  for $p' \in$ Neighbor(p) **do**
    if $p' \in C^{free}$ **then**
      if $p' \notin closed \wedge p' \notin open$ **then**
        if $p \rightarrow p'$ satisfies motion constraints **then**
          $open \leftarrow open \cup p'$
        end if
      end if
    end if
  end for
  $closed \leftarrow closed \cup p$
  $Reach(S, p'.\theta) \leftarrow Reach(S, p'.\theta) \cup \{(p'.x, p'.y)\}$
end while
Return $Reach$

of the robot models, we rotate them by $2k\pi/n_\theta$, where $0 \le k < n_\theta$, in order to obtain the representation of the robot for each possible orientation.

Therefore, we have two structuring elements, R and $Sens$, that represent the robot and sensor models respectively. After the rotation, we get $R(i)$ and $Sens(i)$, with $0 \le i < n_\theta$.

After we create the structuring elements, we can use the morphological operations to determine the free configuration space for each layer, by dilating the map using the robot shape for each orientation. We use a circular representation, where the next layer after $i = n_\theta - 1$ is layer $i = 0$.

$$C^{free}(i) = \{p \in G \mid p \notin M \oplus R(i)\} \quad \forall 0 \le i < n_\theta \tag{8}$$

Using the connectivity graph from Figure 4, it is possible to find all points in each layer of the free configuration space that connect with the initial position S, obtaining $Reach(S, i)$, as shown in Algorithm 1. From there, if we apply the second dilation operation to the reachable space in each layer, we obtain the actuation space for each orientation. However, instead of using the structuring element R again, we have to dilate the space with information related with the sensor $Sens$ in order to find the visibility bounded by the robot shape. For that purpose, the second operation has to still be bounded by the robot shape, so we use the intersection of both structuring elements, after aligning them with their relative displacement.

$$Sens_B(i) = R(i) \cap Sens(i) \tag{9}$$

Then, the actuation space is given by

$$A(S, i) = Reach(S, i) \oplus Sens_B(i) \tag{10}$$

The actuation space gives the visibility bounded by the robot shape, for each orientation. So, if a point belongs to $A(S, i)$, then it can be visible by the robot, while being bounded by the robot shape. After determining the actuation space for each layer, we can obtain the overall actuation space for any orientation just by projecting the multiple layers into one single 2D image.

$$PA(S) = \bigcup_i A(S, i) \tag{11}$$

$PA(S)$ has the same kind of representation we had with the circular robot, where the actuation space is a single 2D image not depending on the orientation. Therefore, the unreachable regions $U^l(S)$ and respective frontiers $F^l(S)$ can be determined in the same fashion as we did with the circular robot. $PA(S)$ is not used to obtain the critical points though, as orientation must be taken into account for that.

If the intersection between R and $Sens$ is empty, the actuation space is also going to be empty. As a result, all points are going to belong to the unreachable set, and there is going to be no frontiers.

B. Critical Points for Any Sensor Model

After determining the frontiers of actuation $F^l(S)$ we still have the same problem of determining visibility inside unreachable regions $U^l(S)$. The true visibility can be obtained by determining the visibility from all points in the reachable space, but for efficiency purposes, we only use one. However, in the any-shape robot case with general sensor models, we do not use the center of the robot as a critical point, but a viable position for the sensor center instead. Because these two positions might not overlap, we enforce a more accurate estimate for visibility when considering the sensor position as critical point.

If again the critical point was chosen as the closest position to the frontier points, the result could be very poor, because a position with a non-optimal orientation towards the frontier might be chosen. The orientation problem happens because the sensor model is not guaranteed to be omnidirectional. In the circular case, searching for the closest point to the frontier was a dual problem of finding the point with greater expected visibility through the frontier. However, that assumption does not hold in the general case.

Therefore, in order to choose a point that improves accuracy of our estimation of visibility, we will still use the criteria that maximizes expected visibility. For that purpose, we build an histogram with the percentage of points in the sensor model in different directions, using the angle to the sensor center. The histogram represents the visibility of the sensor in each direction, as a function of robot orientation. So, $b_s(\phi)$ is the bin that corresponds to angle ϕ, where ϕ is the angle of the vector from the sensor center to the point in question. b_s returns values between 0 and $n_s - 1$, with n_s being the number of bins in the histogram.

$$b_s(\phi) = \underset{0 \leq n < n_s}{\operatorname{argmin}} |\text{angleNorm}(\phi - 2n\pi/n_s)| \tag{12}$$

with *angleNorm* normalizing the angle between π and $-\pi$.

The histogram is built by iterating over all points in the sensor model $Sens$, determining the angle ϕ for each point, and the respective histogram bin $b_s(\phi)$.

With the sensor histogram, it is possible to estimate the expected visibility inside the unreachable regions using only the frontier points, namely the frontier extremes. $F^{li}(S)$ is a chain of connected points, all connected to two or more other points of the frontier, except the extremes, which are only connected to one other. Thus it is possible to find the frontier extremes $F_1^{li}(S)$ and $F_2^{li}(S)$ by searching over all points in the frontier for the ones with only one connection. This assumption holds because we use 4-neighborhood when extracting the frontier points.

With the frontier extremes, it is possible to determine the direction of the frontier with

$$\vec{F^{li}} = \frac{F_1^{li}(S) - F_2^{li}(S)}{||F_1^{li}(S) - F_2^{li}(S)||} = (F_x^{li}, F_y^{li}) \tag{13}$$

The frontier normal can be any vector such as $\vec{N^{li}} \cdot \vec{F^{li}} = 0$. Assuming unit vectors, there are two possible directions, $(F_y^{li}, -F_x^{li})$ or $(-F_y^{li}, F_x^{li})$. Moreover, we can also define the frontier center as

$$C_f^{li} = \frac{1}{\#F^{li}(S)} \sum_{f \in F^{li}(S)} f \tag{14}$$

where $\#$ is the set cardinality. It is possible to find the perpendicular vector as the one in the direction of the frontier center by using the following test:

$$\vec{N_c^{li}} = \underset{|\vec{N^{li}}|=1}{\operatorname{argmax}} \vec{N^{li}} \cdot \Big((C_f^{li} - F_1^{li}) + (C_f^{li} - F_2^{li}) \Big) \tag{15}$$

Similarly, if we find the neighbors of $F^{li}(S)$ in $A(S)$, we will obtain the frontier of actuation, i.e. $F_a^{li}(S)$ and we can similarly find its center:

$$C_a^{li} = \frac{1}{\#F_a^{li}(S)} \sum_{f \in F_a^{li}(S)} f \tag{16}$$

The perpendicular direction to the frontier that points to the unreachable region is

$$\vec{N_u^{li}} = \underset{|\vec{N^{li}}|=1}{\operatorname{argmax}} \vec{N^{li}} \cdot (C_f^{li} - C_a^{li}) \tag{17}$$

Finally, we use the two normal directions to estimate the concavity of the frontier, which is used later to estimate the visibility inside unreachable regions from the critical points. If $\vec{N_c^{li}} \cdot \vec{N_u^{li}} > 0$, the frontier is concave, otherwise it is convex. This is important to consider when searching for a critical point in the any-shape scenario, while in the circular robot case the frontiers were always concave.

In order to search for the critical point, we use the angle of an annulus sector defined by the critical point and the frontier extremes. If the frontier is convex, the critical point has to stay behind it, so $(c_{li}^*(S) - F_1^{li}) \cdot \vec{N_u^{li}} < 0$, and the angle of the annulus sector is less than π. In the case of a concave frontier, there is two possibilities: the critical point is still behind the frontier, with $(c_{li}^*(S) - F_1^{li}) \cdot \vec{N_u^{li}} < 0$, and the angle between the two extremes is less than π; or the critical point is in the middle of the concavity, with $(c_{li}^*(S) - F_1^{li}) \cdot \vec{N_u^{li}} > 0$ and the angle between the extremes being greater than π.

From now on, we assume we ordered the extremes appropriately, so the frontier goes from angle $\phi_1^{cli} = \arctan 2(F_{1y}^{li} - c_{liy}, F_{1x}^{li} - c_{lix})$ to $\phi_2^{cl} = \arctan 2(F_{2y}^{li} - c_{liy}, F_{2x}^{li} - c_{lix})$. Then, the expected visible area inside the unreachable region is given by the sensor histogram:

$$V_e^{cli} = \Big(\sum_{\phi=\phi_1^{cli}}^{\phi_2^{cli}} \text{hist}(b_s(\phi)) \Big) - ||c_{li} - C_f^{li}||^2 (\phi_2^{cli} - \phi_1^{cli})/2 \tag{18}$$

where $||c_{li} - C_f^{li}||^2 (\phi_2^{cli} - \phi_1^{cli})/2$ accounts for the area from the critical point to the frontier already visible and counted in the actuation space $PA(S)$.

Finally, the critical point is given by search in the layered image representation

$$c_{li}^*(S) = \underset{c_{li} \in Reach(S)}{\operatorname{argmax}} V_e^{cli} \tag{19}$$

After finding the critical point, the visibility inside unreachable regions is determined using ray casting, as we did in the work with the circular shape assumption.

The overall algorithm to create the Robot-Dependent Visibility Map for any-shape robots with general sensor models, in 2D gridmaps, is shown in Algorithm 2.

V. Results

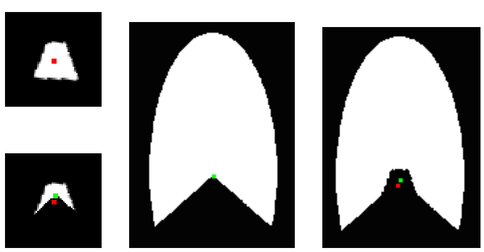
To test the extension of Visibility Maps to any shape robots with any sensor model, we created test examples for both models, as shown in Figure 5. The robot has a trapezoidal shape, and the sensor model is ellipsoid, representing better sensing in the forward direction. Moreover, the sensor field of

Algorithm 2 RDVM: Creating robot-dependent visibility maps from grid maps

Require: Gridmap G, start S, robot R and sensing $Sens$

$M \leftarrow$ im2bw(G) ▷ b&w image from gridmap
$\{C^{free}(i)\} \leftarrow$ dilation(M,$\{R(i)\}$) ▷ inflation obstacles
$\{Reach(S,i)\} \leftarrow$ labeling($\{C^{free}(i)\}$, S)
$\{A(S,i)\} \leftarrow$ dilation($\{Reach(S,i)\}$, $\{R(i) \cap Sens(i)\}$) ▷ partial morph. closing
$PA(S) \leftarrow \bigcup_i A(S,i)$
$V(S) \leftarrow PA(S)$ ▷ visibility initialized with PA
$U(S \leftarrow$ unreachable($PA(S)$,M) ▷ unreachable regions
$\{U^l(S)\} \leftarrow$ labeling($U(S)$) ▷ find disconnected regions
for each $U^l(S)$ **do**
    $F^{li}(S) \leftarrow$ frontier($U^{li}(S)$,M) ▷ find frontiers
    $V_e^{cli}(x) \leftarrow$ expected_visibility(U^{li},F^{li},$Sens$)
    $c_{li}^*(S) = \underset{c_{li} \in Reach(S)}{\text{argmax}} \; V_e^{cli}$ ▷ critical point
    $V_t^{cli}(S) \leftarrow$ brute-force($V_e^{cli}(S), c_{li}^*(S)$)
    $V(S) \leftarrow V(S) \bigcup V_t^{cli}(S)$
end for
Return V

view is limited to approximately 270 degrees. We used scaled and rotated versions of these models to obtain visibility in the simulated map presented before.



(a) Robot Model R (b) Sensor Model $Sens$ (c) Relative Position ($Sens_B$)

Fig. 5. We show in (a) a non-circular robot and its center marked with a red dot. In the middle, the sensor model is shown, with its center marked in green. In (c) we superpose both models to show the relative position parametrization. In the left bottom image, the intersection of robot and sensor models, used as structuring element in the morphological closing.

We show in Figure 6 the comparison between the RDVM and the ground-truth, showing a great similarity in the overall visibility. For the example, the precision and recall were 99% and 96% respectively, which is very close to the results previously obtained for the circular shape. Moreover, considering the instance shown in the figure, the computation time for the ground truth was approximately 200 times more than the time used for the RDVM (590 milliseconds). This demonstrates our proposed method scales well from the circular shape to the any-shape scenario. We run tests with other shapes, sizes, and environments, noticing that for convex shapes with sensing models bigger than the robot (bigger intersection between robot and sensor model), our algorithm has very good results, close to the circular case, with similar accuracy and time results. For the other cases, the frontiers became more complex, and the choice of only one critical point per frontier was not sufficient for a good approximation. A possible solution would be to break the frontier chains in smaller concave frontiers.

(a) Projected Actuation Space (b) Comparison of Visibility

Fig. 6. In (a) we show the projected actuation space, as the union of the partial morphological closing for all orientation layers. The models presented in Figure 5 were applied to the map in Figure 2. In the right we compare the Robot-Dependent Approximate Visibility Map with the Ground-Truth. White represents true positives (visible in both), black are true negatives, red represents false positives, and blue represents false negatives (visibility missed with the approximate solution).

VI. Conclusion

Our algorithm is able to create maps that separate the visible regions using image processing techniques. We use the concept of partial morphological closing, and introduce a multi-layer representation to discretize the orientation and be able to deal with any-shape robots and general sensor models. We also proposed a new optimization function to choose the critical points, allowing an efficient solution for the visibility problem with high accuracy and less computation time. We tested our algorithm in a simulated test scenario, built to capture the different difficulties of getting a visibility map. The results show that we are able to efficiently obtain the visibility maps, with high precision and recall. As future work, we want to study the use of non-critical points in RDVM in order to improve our approximation without losing its time efficiency advantage.

References

[1] T. Pereira, A. P. Moreira, and M. Veloso, "Improving Heuristics of Optimal Perception Planning using Visibility Maps," *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2016.

[2] E. Fabrizi and A. Saffiotti, "Extracting topology-based maps from gridmaps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[3] T. Pereira, M. Veloso, and A. Moreira, "Multi-robot planning using robot-dependent reachability maps from morphological operations," ROBOT'2015 - Second Iberian Robotics Conference, 2015.

[4] A. K. Pandey and R. Alami, "Mightability maps: A perceptual level decisional framework for co-operative and competitive human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[5] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[6] J. Faigl, "Approximate solution of the multiple watchman routes problem with restricted visibility range." *IEEE transactions on neural networks*, vol. 21, no. 10, 2010.

[7] P. Janousek and J. Faigl, "Speeding up coverage queries in 3D multi-goal path planning," *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.