



HAL
open science

Considering adjacent sets for computing the visibility region

Quentin Brateau, Fabrice Le Bars, Luc Jaulin

► **To cite this version:**

Quentin Brateau, Fabrice Le Bars, Luc Jaulin. Considering adjacent sets for computing the visibility region. 2024. hal-04847934

HAL Id: hal-04847934

<https://hal.science/hal-04847934v1>

Preprint submitted on 19 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Considering adjacent sets for computing the visibility region*

Quentin Brateau, quentin.brateau@ensta-bretagne.org,
Fabrice Le Bars, fabrice.le_bars@ensta-bretagne.fr,
Luc Jaulin, lucjaulin@gmail.com

December 19, 2024

Abstract

This paper explores the problem of the paving of the union of adjacent contractors. The focus is first put on the analysis of the topology of a set operator, which can be stable or not stable. Then, depending on the stability of the union operator, solutions are proposed to avoid fake boundaries in stable and non-stable union of sets. For stable union of sets, a boundary preserving form will be developed to add a set overlapping the fake boundary in the expression of the union, whereas for non-stable union of sets, a boundary approach will be developed to avoid fake boundaries. Some problem specific solutions are also developed to avoid fake boundaries. As an example, an enhancement of the separator on the visibility constraint is proposed. This avoids fake boundaries while characterizing the set of non-visible points from an observation point relative to a polygon.

Keywords— Set methods, Interval analysis, Contractors, Set Inversion, Topology

1 Introduction

Interval Arithmetic and contractor programming have emerged as powerful tools in the field of robotics [1, 2, 3, 4], offering robust methods for handling uncertainty and performing set-based computations. These techniques have been widely applied in robotics area such as in localization [2, 5], in path planning [6, 7], and in control of systems [8, 9].

Interval analysis is a subset of set methods where sets are represented by intervals. Some operators are defined in classical set theory, such as union, intersection, complementary of sets and so on [10]. As intervals are representing sets, these operators are also defined for intervals by interval arithmetic [11].

*This work has been supported by the French Government Defense procurement and technology agency (AID)

Contractors are a mathematical function acting on intervals. They are used to contract a domain of feasible values relative to a constraint. Denoting by \mathbb{IR}^n the set of axis-aligned boxes of \mathbb{R}^n , the operator $\mathcal{C} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a contractor for $\mathbb{X} \subseteq \mathbb{R}^n$ if it meets the condition Equation (1).

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathcal{C}([\mathbf{x}]) \subseteq [\mathbf{x}] & (\text{Contractance}) \\ \mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & (\text{Completeness}) \end{cases} \quad (1)$$

A contractor is representing a set as defined in [12]. Set operators are then defined for contractors as it is for sets. Then contractors can be combined by computing their union, their intersection, their cartesian product, and so on.

However, while most of these operations are well-defined and straightforward to implement, the union operation presents unique challenges, particularly when dealing with non-overlapping sets. The union of adjacent or non-overlapping sets using contractors can sometimes result in the appearance of fake boundaries at the interface of these sets. This phenomenon, was first highlighted in [13], and a solution was proposed using appropriate Disjunctive Normal Forms (DNF) and Conjunctive Normal Forms (CNF) [14] to avoid these fake boundaries. However, this solution is not always applicable as sets are not always defined as unions and intersections of sets.

For instance, set of visible points from an observation point relative to an obstacle is defined and implemented in [15]. By paving this set, fake boundaries may appear in the non-visible area. Figure 1 shows an example of the set of visible points from an observation point relative to a polygon obstacle. The observation point is shown in red. The set of visible points from this observation point is shown in blue, the set of non-visible points is shown in pink, and the set of uncertain points is shown in yellow. The obstacle is shown in black. There is fake boundaries appearing in the pink area, which are lines of yellow boxes in the non-visible area.

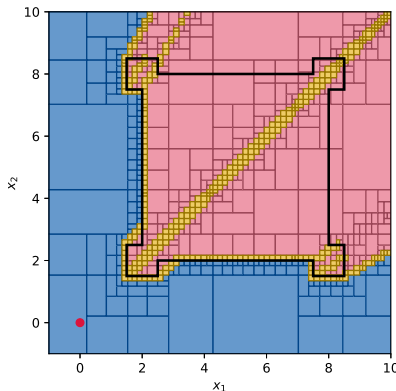


Figure 1: Separator on the visibility constraint

This article aims to address the problem of union operations on adjacent

contractors, with a focus on eliminating fake boundaries. By developing new techniques for performing union operations on adjacent sets, we seek to minimize the added pessimism to the results, to improve the efficiency of the paving algorithm, and to enhance the accuracy and reliability of set-based computations in robotic applications.

This paper is organized as follows. Section 2 present the problem of the union of adjacent contractors by an introducing example. Then, Section 3 analyze the problem from a topological point of view, and distinguish stable and non-stable set operators. Section 4 and Section 5 present the solutions to avoid fake boundaries in both cases. Section 6 presents an application of the boundary approach on the separator on the visibility constraint. Finally, Section 7 concludes the paper.

2 Problem statement

2.1 Illustrative example

Consider three sets \mathbb{A} , \mathbb{B} and \mathbb{C} defined by Equation (2).

$$\begin{aligned} \mathbb{A} &: \{x_1 + 3 \cdot x_2 \in [-\infty, 0]\} \\ \mathbb{B} &: \{(x_1 + 0.5)^2 + x_2^2 \in [-\infty, 4]\} \\ \mathbb{C} &: \{(x_1 - 0.5)^2 + x_2 \in [-\infty, 4]\} \end{aligned} \quad (2)$$

These sets are shown in Figure 2. The interior of the set is shown in pink, and the exterior is shown in blue.

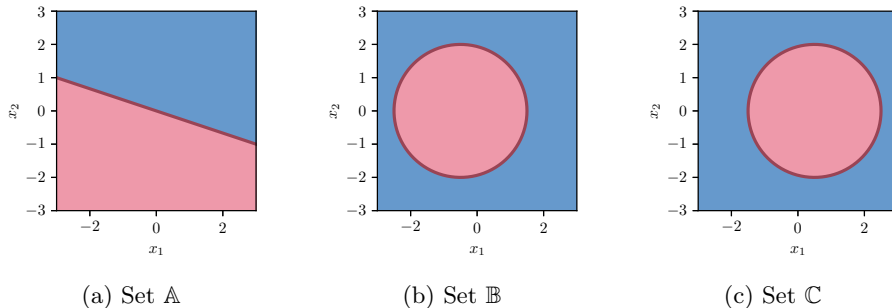


Figure 2: Sets \mathbb{A} , \mathbb{B} and \mathbb{C}

Define a set \mathbb{Z} computed using \mathbb{A} , \mathbb{B} and \mathbb{C} by Equation (3).

$$\mathbb{Z} = (\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C}) \quad (3)$$

This set \mathbb{Z} shown in Figure 10c is built on the union of sets $\mathbb{A} \cap \mathbb{B}$ and $\overline{\mathbb{A}} \cap \mathbb{C}$ represented in Figure 3a, and Figure 3b.

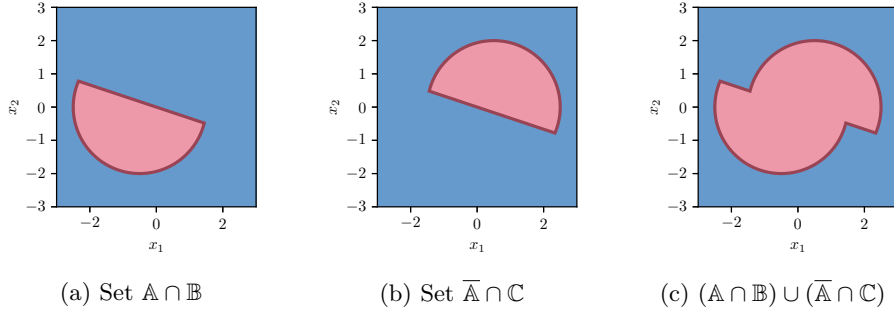


Figure 3: Construction of set \mathbb{Z} from A , B and C

Note that these two sets $A \cap B$ and $\bar{A} \cap C$ share a common and non-overlapping boundary. While paving this set \mathbb{Z} using the SIVIA algorithm [11], this common boundary is appearing as shown in Figure 4. This boundary is called a fake boundary [13] as it is not supposed to belong to \mathbb{Z} .

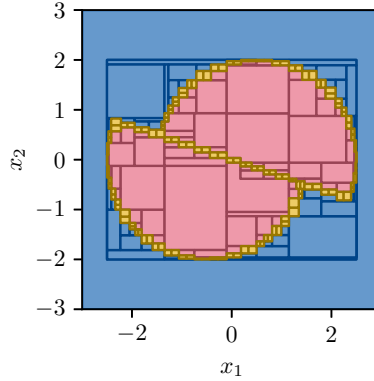


Figure 4: Paving of set \mathbb{Z}

2.2 Paving point of view

As shown in Figure 5, the paving algorithm is unable to classify an inner box $[b]$ overlapping the fake boundary as fully inside \mathbb{Z} . Using contractors defined for \mathbb{Z} , inner parts $[b_1] = \mathcal{C}_{A \cap B}([b])$, and $[b_2] = \mathcal{C}_{\bar{A} \cap C}([b])$ are well classified. The remaining part $[b_3] = [b] \setminus [b_1] \setminus [b_2]$ is classified as unknown and is bisected until the paving algorithm reaches the desired precision.

To avoid this issue, the paving algorithm has to take into account the fact that $A \cup \bar{A} = \mathbb{R}^n$. With this piece of information, the box $[b]$ can be classified as fully inside \mathbb{Z} in one step.

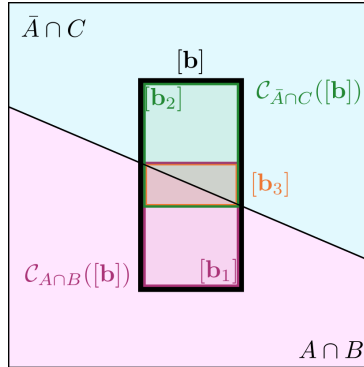


Figure 5: Paving of the fake boundary

2.3 Karnaugh map point of view

Karnaugh maps for $(A \cap B) \cup (\bar{A} \cap C)$ and Z are respectively shown in Figure 6a and Figure 6b. The interior is shown in pink, the exterior is shown in blue, and the boundary is shown in yellow. Although the interior and the exterior of these two sets are equal, the boundaries differ. The fake boundary appearing on the paving in Figure 4 is $\partial A \cap B \cap C$ and is exactly the difference between the boundaries of $(A \cap B) \cup (\bar{A} \cap C)$ and Z .

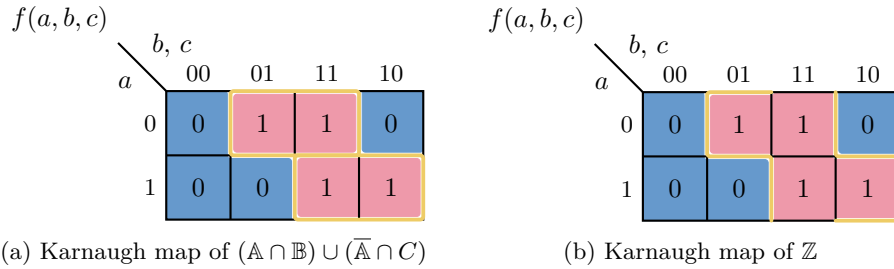


Figure 6: Comparing Karnaugh maps of $(A \cap B) \cup (\bar{A} \cap C)$ and Z

2.4 Raised issues

This fake boundary raises two issues. First, it adds pessimism to the results by classifying boxes around the common boundary as uncertain, whereas they clearly belong to the union of the two sets. Secondly, this fake boundary slows down the paving algorithm by unnecessarily bisecting boxes around the fake boundary.

3 Stability of set operators

3.1 Topological analysis of set operators

To better understand the issue around the union of adjacent sets, we need to define some tools to analyze the origin of these fake boundaries. Actually, this fake boundary may occur when a set operator is Hausdorff-stable, and when it is not Hausdorff-stable, but solutions to avoid these fake boundaries are not the same in the two cases.

3.2 Hausdorff distance

Let (\mathbb{S}, d) be a metric space. Define the ϵ -fattening [16] of a set \mathbb{X} of \mathbb{S} by Equation (4).

$$\mathbb{X}_\epsilon = \bigcup_{x \in \mathbb{X}} \{z \in \mathbb{S} \mid d(z, x) \leq \epsilon\} \quad (4)$$

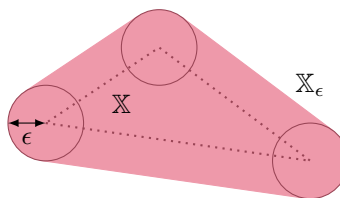


Figure 7: ϵ -fattening of a set

The Hausdorff distance [16] between two sets \mathbb{X} and \mathbb{Y} of \mathbb{S} is defined by Equation (5).

$$d_H(\mathbb{X}, \mathbb{Y}) = \inf\{\epsilon \in \mathbb{R}^+ \mid \mathbb{X} \subseteq \mathbb{Y}_\epsilon \text{ and } \mathbb{Y} \subseteq \mathbb{X}_\epsilon\} \quad (5)$$

Introducing the complementary Hausdorff distance defined in Equation (6).

$$\overline{d}_H(\mathbb{X}, \mathbb{Y}) = d_H(\overline{\mathbb{X}}, \overline{\mathbb{Y}}) \quad (6)$$

Example 1. Figure 8 illustrate cases where Hausdorff distance and complementary Hausdorff distance are significant. Figure 8a shows an example of two sets \mathbb{A} and \mathbb{B} with $d_H(\mathbb{A}, \mathbb{B})$ large because of the small part of \mathbb{A} far from the main part, but $\overline{d}_H(\mathbb{A}, \mathbb{B})$ is tiny, whereas Figure 8b shows an example where $d_H(\mathbb{A}, \mathbb{B})$ is tiny and $\overline{d}_H(\mathbb{A}, \mathbb{B})$ is large because of the hole in \mathbb{A} .

To take into account the general topology of sets, and to be able to compare it, the generalized Hausdorff distance is introduced and defined in Equation (7). It is the maximum between the Hausdorff distance and the complementary Hausdorff distance.

$$H_d(\mathbb{X}, \mathbb{Y}) = \max\{d_H(\mathbb{X}, \mathbb{Y}), \overline{d}_H(\mathbb{X}, \mathbb{Y})\} \quad (7)$$

3.3 Hausdorff stability

Consider two sets \mathbb{X} and \mathbb{Y} of \mathbb{S} . Define two sets $\tilde{\mathbb{X}}$ and $\tilde{\mathbb{Y}}$ of \mathbb{S} close to \mathbb{X} and \mathbb{Y} in the generalized Hausdorff definition. Then a binary operator \diamond acting on set \mathbb{X} and \mathbb{Y} is stable if it meets condition of Equation (8).

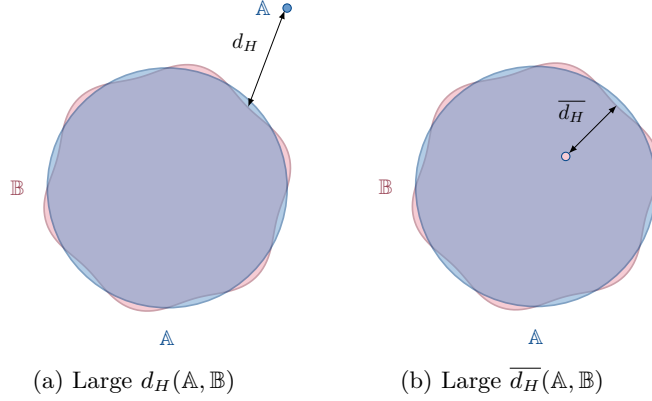


Figure 8: Illustration of large Hausdorff and complementary Hausdorff distances

$$\forall \eta \in \mathbb{R}, \quad \exists \epsilon \in \mathbb{R}, \quad \begin{cases} H_d(\mathbb{X}, \tilde{\mathbb{X}}) \leq \epsilon \\ H_d(\mathbb{Y}, \tilde{\mathbb{Y}}) \leq \epsilon \end{cases} \implies H_d(\mathbb{X} \diamond \mathbb{Y}, \tilde{\mathbb{X}} \diamond \tilde{\mathbb{Y}}) \leq \eta \quad (8)$$

Example 2. Consider two sets \mathbb{A} and \mathbb{B} shown in Figure 9a and their respective Hausdorff-close sets $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{B}}$ shown in Figure 9b.

For the union operator, $d_H(\mathbb{A} \cup \mathbb{B}, \tilde{\mathbb{A}} \cup \tilde{\mathbb{B}})$ is small, but $\overline{d_H}(\mathbb{A} \cup \mathbb{B}, \tilde{\mathbb{A}} \cup \tilde{\mathbb{B}})$ is large as the union of $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{B}}$ generates holes at the common boundary of \mathbb{A} and \mathbb{B} . Then $H_d(\mathbb{A} \cup \mathbb{B}, \tilde{\mathbb{A}} \cup \tilde{\mathbb{B}})$ is large, and the union operator is not Hausdorff-stable for these sets, as it does not meet the condition of Equation (8).

Example 3. Consider two sets \mathbb{A} and \mathbb{B} shown in Figure 9a and their respective Hausdorff-close sets $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{B}}$ shown in Figure 9b.

For the intersection operator, $\overline{d_H}(\mathbb{A} \cap \mathbb{B}, \tilde{\mathbb{A}} \cap \tilde{\mathbb{B}})$ is small, but $d_H(\mathbb{A} \cap \mathbb{B}, \tilde{\mathbb{A}} \cap \tilde{\mathbb{B}})$ is large as the intersection of $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{B}}$ generates residual sets at the common boundary of \mathbb{A} and \mathbb{B} . Then $H_d(\mathbb{A} \cap \mathbb{B}, \tilde{\mathbb{A}} \cap \tilde{\mathbb{B}})$ is large, and the intersection operator is not Hausdorff-stable for these sets, as it does not meet the condition of Equation (8).

Example 4. Consider the illustrative example presented in section 2. The union operator between $\mathbb{A} \cap \mathbb{B}$ and $\overline{\mathbb{A}} \cap \mathbb{C}$ is Hausdorff stable as the generalized Hausdorff distance is small. This comes from the fact that the same sets \mathbb{A} and its associated Hausdorff-close set $\tilde{\mathbb{A}}$, are involved in the computation of $H_d(\mathbb{A} \cap \mathbb{B}, \tilde{\mathbb{A}} \cap \tilde{\mathbb{B}})$ and $H_d(\overline{\mathbb{A}} \cap \mathbb{C}, \tilde{\mathbb{A}} \cap \tilde{\mathbb{C}})$.

This Hausdorff stability condition characterizes the fact that a small perturbation on sets will change the topology of the result by opening boundaries or creating additional ones. It allows identifying topologically different problems. Adapted solutions for Hausdorff-stable and non Hausdorff-stable problems will be proposed in the following sections.

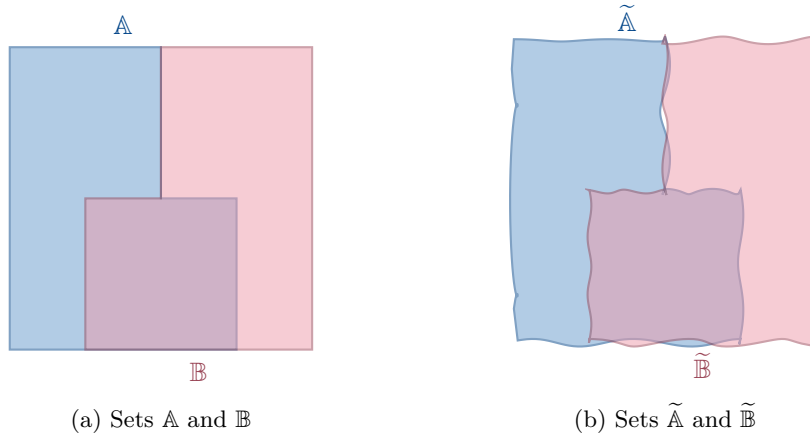


Figure 9: A and B are not Hausdorff-stable for union and intersection operators

4 Stable case solution: Boundary preserving form

In the Hausdorff-stable case, it is possible to change the expression of the computed set Z by adding a set overlapping the fake boundary. This set helps in the classification of boxes around the fake boundary in the paving algorithm. It must be chosen such that the interior and the exterior of Z are preserved, but also its boundary. In this example, the set $D = B \cap C$ is added to the expression of Z which becomes Z' Equation (9).

$$Z' = (A \cap B) \cup (\bar{A} \cap C) \cup (B \cap C) \quad (9)$$

The Karnaugh map of the set D is shown in Figure 10a, and the paving of D is shown in Figure 10b. This set ensures that the Karnaugh map of Z' is the same as the Karnaugh map of Z shown in Figure 6b. The resulting paving of Z' is shown in Figure 10c. There is no more fake boundaries.

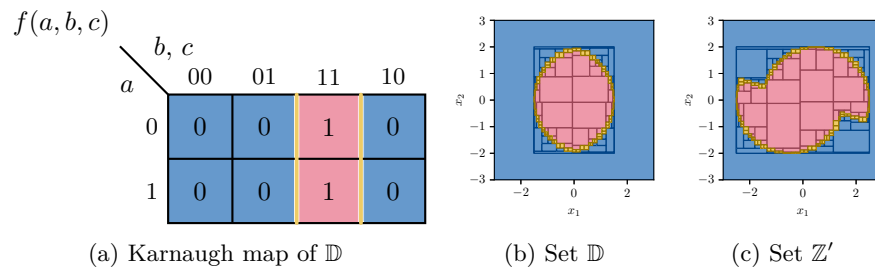


Figure 10: Boundary preserving form

Using the boundary preserving form leads to a correct paving without any fake boundaries. Therefore, to use this solution, the set boundaries have to

be analyzed to find the fake boundaries and add a set overlapping these fake boundaries in the expression of the set to pave. This approach is working but is problem-specific and needs to be adapted on a case-by-case basis. This method is well working for Hausdorff-stable operations on sets, as there is the possibility to add a boundary overlapping set to the computed sets. For non Hausdorff-stable operators, the boundary preserving form is not possible, and another approach is needed.

5 Non-stable case: Boundary approach

5.1 Topology of the boundary

Let $T = (\mathbb{S}, \tau)$ be a topological space. $\forall \mathbb{X} \in \mathbb{S}$, denote by $\overline{\mathbb{X}}$ the complementary of \mathbb{X} in \mathbb{S} , by $cl_{\mathbb{S}}(\mathbb{X})$ the closure of \mathbb{X} in \mathbb{S} , by $int_{\mathbb{S}}(\mathbb{X})$ the interior of \mathbb{X} in \mathbb{S} , and by $\partial\mathbb{X}$ the boundary of \mathbb{X} in \mathbb{S} .

Theorem 1. *Let $T = (\mathbb{S}, \tau)$ be a topological space. Then*

$$\forall (\mathbb{A}, \mathbb{B}) \in \mathbb{S}^2 \quad \partial(\mathbb{A} \cup \mathbb{B}) \subseteq \partial\mathbb{A} \cup \partial\mathbb{B}$$

Proof. By definition of the boundary

$$\forall \mathbb{A} \in \mathbb{S}, \quad \partial\mathbb{A} = cl_{\mathbb{S}}(\overline{\mathbb{A}}) \cap cl_{\mathbb{S}}(\mathbb{A})$$

By property, intersection is a subset of each set

$$\forall (\mathbb{A}, \mathbb{B}) \in \mathbb{S}^2, \quad \begin{cases} \mathbb{A} \cap \mathbb{B} \subseteq \mathbb{A} \\ \mathbb{A} \cap \mathbb{B} \subseteq \mathbb{B} \end{cases}$$

Then

$$\begin{aligned} \partial(\mathbb{A} \cup \mathbb{B}) &= cl_{\mathbb{S}}(\overline{\mathbb{A} \cup \mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{A} \cup \mathbb{B}) \\ &= cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{A} \cup \mathbb{B}) \\ &= cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap (cl_{\mathbb{S}}(\mathbb{A}) \cup cl_{\mathbb{S}}(\mathbb{B})) \\ &= (cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{A})) \cup cl_{\mathbb{S}}(\overline{\mathbb{A}} \cap \overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{B}) \\ &\subseteq (cl_{\mathbb{S}}(\overline{\mathbb{A}}) \cap cl_{\mathbb{S}}(\mathbb{A})) \cup (cl_{\mathbb{S}}(\overline{\mathbb{B}}) \cap cl_{\mathbb{S}}(\mathbb{B})) \\ &= \partial\mathbb{A} \cup \partial\mathbb{B} \end{aligned}$$

□

Theorem 1 demonstrates that the boundary is not preserved over union of sets as $\partial(\mathbb{A} \cup \mathbb{B}) \subseteq \partial\mathbb{A} \cup \partial\mathbb{B}$. This is why the paving of the union of contractors leads to fake boundaries.

Theorem 2 present the general formula for the union of the boundary of two sets.

Theorem 2. *Let (S, τ) be a topological space. Then*

$$\forall (\mathbb{A}, \mathbb{B}) \in \mathcal{S}^2 \quad \partial \mathbb{A} \cup \partial \mathbb{B} = \partial(\mathbb{A} \cup \mathbb{B}) \cup \partial(\mathbb{A} \cap \mathbb{B}) \cup (\partial \mathbb{A} \cap \partial \mathbb{B})$$

Proof. Theorem 2 is proven in [17]. □

From Theorem 2, it is clear that union of boundaries is not the boundary of union. This is the reason why \mathbb{Z} and $(\mathbb{A} \cap \mathbb{B}) \cup (\overline{\mathbb{A}} \cap \mathbb{C})$ do not have the same boundaries when paving these sets. An illustration of Theorem 2 is shown in Figure 11.

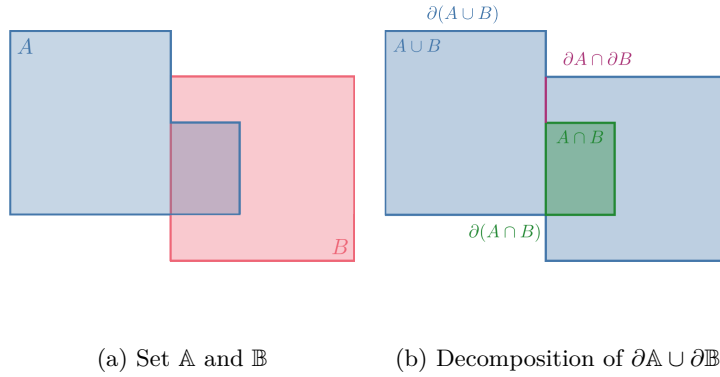


Figure 11: Illustration of Theorem 2

Remark 1. *When $\mathbb{A} \cap \mathbb{B} = \emptyset$ and $\partial \mathbb{A} \cap \partial \mathbb{B} = \emptyset$ in Theorem 2, the union of boundaries is the boundary of union. This is the case where the sets are non-overlapping with no common boundary.*

5.2 Boundary approach

To get rid of this fake boundary, a boundary approach can be used. This approach consists in computing the boundary of the set \mathbb{Z} . This boundary will separate an inner and an outer subpaving. The classification of the resulting subpavings as inside or outside is done using a predicate. Boundary approach method was first introduced in [18] to speed up the solving of set inversion problems.

First, $\partial \mathbb{Z}$ has to be expressed from set \mathbb{A} , \mathbb{B} , and \mathbb{C} without the fake boundary. Figure 12 and Figure 13 respectively show Karnaugh maps and paving of intermediate sets involved in the building of $\partial \mathbb{Z}$. Then, $\partial \mathbb{Z}$ is computed as the union of these boundaries, and it matches the Karnaugh map of \mathbb{Z} shown in Figure 6b.

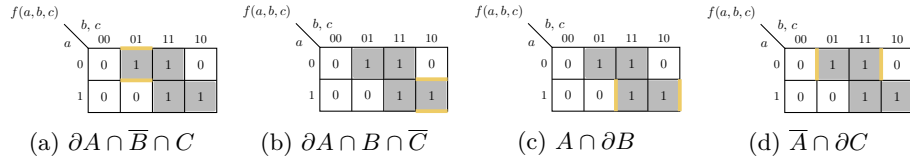


Figure 12: Karnaugh map of the boundaries

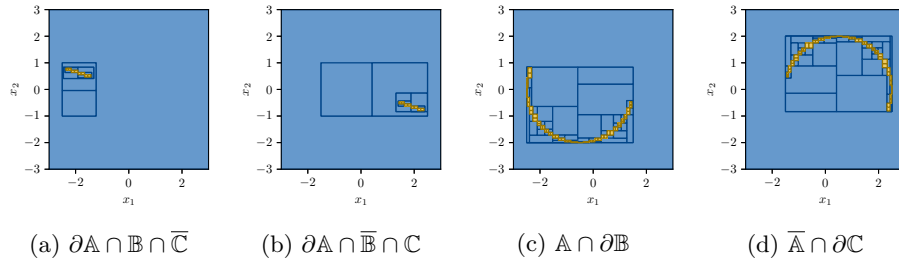


Figure 13: Building the boundary of Z

Then using a predicate, the connected subsets separated by ∂Z are classified as inside or outside. This predicate is based on the expression of Z of Equation (3), and is tested on box corners until an in and an out points are found. Then, boxes containing each point are classified as in and out boxes, and the information is propagated near to near without crossing the boundary. Finally, each box is classified as in, out or uncertain.

Figure 14a shows ∂Z built from boundaries shown in Figure 14b, and the resulting paving of Z , which is classified using the subpaving coloration method.

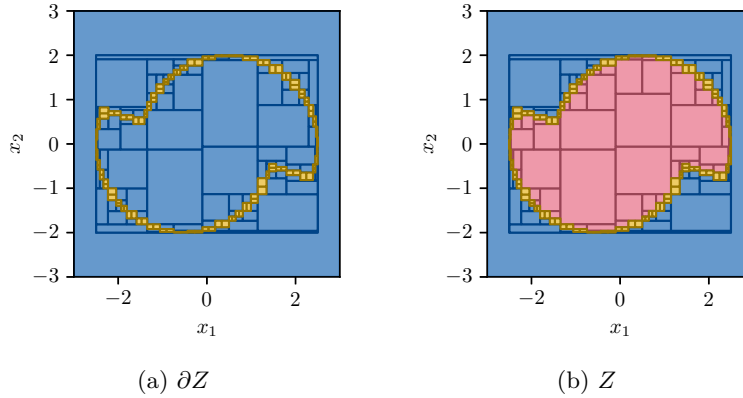


Figure 14: Boundary approach

This boundary approach is efficient to get rid of fake boundaries. Set Z is computed from the union of two separators, $\mathcal{S}_{A \cap B}$, and $\mathcal{S}_{\bar{A} \cap C}$, and this union is

reinforced by a contractor on the boundary $\mathcal{C}_{\partial Z}$.

Remark 2. *This method is also working for the Hausdorff-stable case, but it is more efficient to use the boundary preserving form presented in Section 4, as the contractor on the boundary is not easy to define, and the subpaving coloration method is not needed.*

6 Application

6.1 Boundary approach application to the separator on the visibility constraint

Separator over the visibility constraint, as implemented in [15], suffer from this fake boundaries when it deals with polygon obstacles. In fact, the contractor on the visibility constraint is defined for an obstacle segment. The extension to polygons involves the union of non-visible areas relative to each segment, and this union leads to fake boundaries.

Figure 15a shows an illustration of the separator on the visibility constraint as implemented in [15]. For each obstacle segment is defined three segments defining visible and non-visible parts of the space. For segment e_1 is defined relative to the observation point p , the oriented half space on the left of segment a , the one on the left of segment b , and the same for segment c . It is the same for the set of visible points for segment e_2 defined by half planes on the left of segments d , e , and f . The set of masked points from p by e_1 and e_2 is then the union of these two sets A_1 and A_2 . Paving this separator shows a fake boundary as shown in Figure 15b.

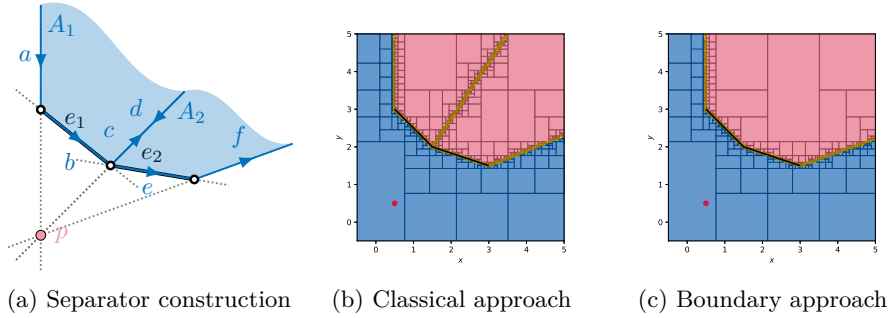


Figure 15: Separator on the visibility constraint using the boundary approach

To avoid this problem, the boundary approach can be applied. The set of masked points from observation point p relative to segments e_1 and e_2 should be defined by half planes on the left of segments a , b , e , and f . The simplification of $c = -d$ has to be taken into account while contracting to avoid this fake boundary. This simplification is based on algebraic topology [19] in which boundary simplifications are defined and used. Figure 15c shows the paving of

this separator using the boundary approach. There is no longer fake boundaries appearing.

Remark 3. *This could be done by hand, but neither [15] nor this work propose an automatic boundary simplification to avoid fake boundaries in union of adjacent sets. Therefore, it is necessary to find solutions that are problem-specific in order to avoid fake boundaries.*

6.2 Toward a generic implementation of the separator on the visibility constraint

In the case of the visibility constraint another approach to solve this problem can be proposed. The set of visible points from an observation point relative to a shape \mathbb{Y} can be defined by :

$$\mathbb{S} = \{x \in \mathbb{R}^2, \exists \alpha \in [\alpha] \mid \alpha \cdot \mathbf{x} \in \mathbb{Y}\} \quad (10)$$

Denoting by f the homothety of expression Equation (11).

$$\begin{aligned} f : \mathbb{R}^3 &\mapsto \mathbb{R}^2 \\ (\mathbf{x}, \alpha) &\rightarrow \alpha \cdot \mathbf{x} \end{aligned} \quad (11)$$

The set \mathbb{S} can be then defined as the projection of $f(\mathbb{Y})$ for $\alpha \in [0, 1]$. Listing 1 show the implementation of this separator using the Codac Library [20]. Figure 16a shows the paving of this implementation of the visibility constraint. The comparison with Figure 16b, where the classical implementation of this constraint from [15] on the same obstacle polygon is shown, validates that the problem of fake boundaries is avoided with this method.

```

1 import codac as cd
2
3 # Set Y definition
4 polygon = [[1, 2.5], ... ]
5 Sy = cd.SepPolygon(polygon)
6
7 # Set Z definition
8 f = cd.Function("x", "y", "a", "(a*x,a*y)")
9 Sz = cd.SepInverse(Sy, f)
10
11 # Projection of for a in [0, 1]
12 epsilon = 0.1
13 Sx = cd.SepProj(Sz, cd.Interval(0, 1), epsilon)

```

Listing 1: Separator on the visibility constraint using Codac Library

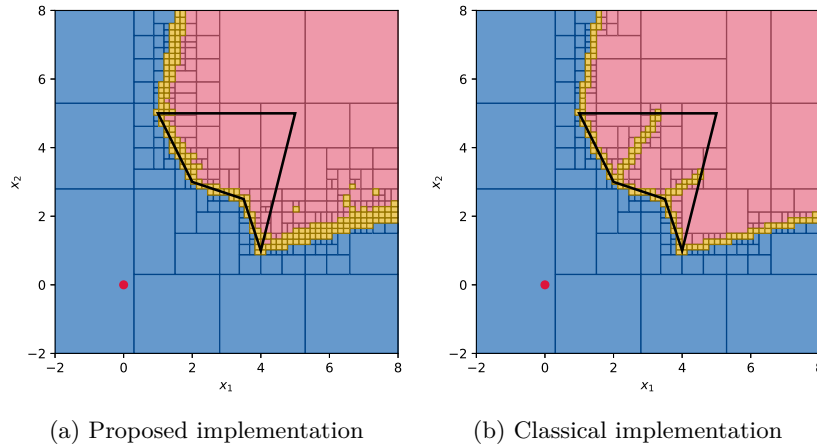


Figure 16: Generic SepVisible implementation

Figure 17 shows the paving of the visibility separator on the same obstacle presented in Figure 1, but there is no fake boundary appearing. This method is then more efficient than the classical implementation of the visibility constraint.

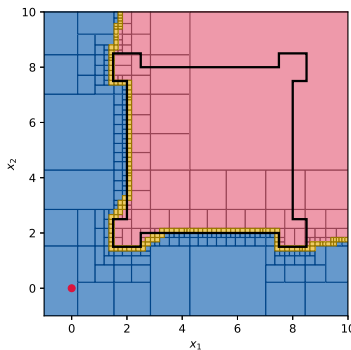


Figure 17: Separator on the visibility constraint on a room

Remark 4. *The separator representing the obstacle could be any separator. However, the separator must be on a closed form with an interior. This method is not applicable to segments or open polygons for instance. But the advantage of this approach is that it can be applied to an ellipsis obstacle.*

Remark 5. *It exists polygons such that the separator on the visibility constraint is not generating fake boundaries. In these cases the classical implementation proposed in [15] is more efficient than the proposed approach as the algorithm used to project a separator is based on contractors over quantifiers which requires bisections [12, 21]. Figure 18 shows the comparison between the classical and*

the projection approaches on the paving of a visibility separator without fake boundaries.

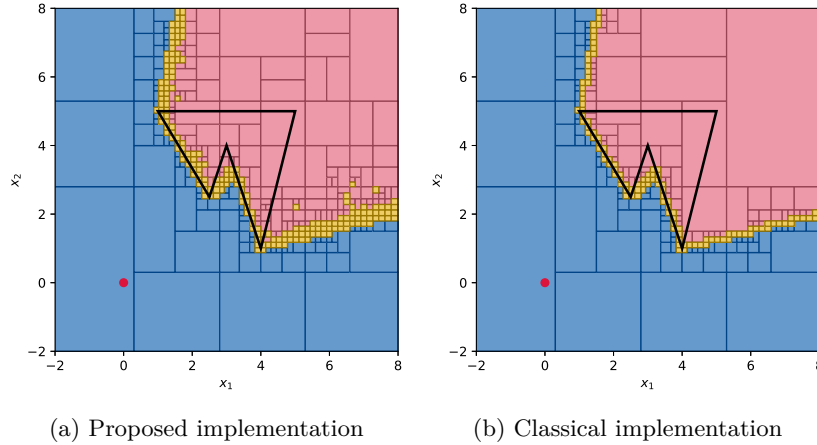


Figure 18: Generic implementation of the separator on the visibility constraint

7 Conclusion

In conclusion, this work has highlighted the problem associated with the union of adjacent contractors. Paving the union of these contractors creates fake boundaries that add pessimism to the results and increase the computation time.

This problem occurs in two cases, when an operator applied to sets is Hausdorff-stable, and when it is non-Hausdorff-stable. An approach for Hausdorff-stable is to use a boundary preserving form by adding sets overlapping the fake boundary in the expression of the paved set. For non-Hausdorff-stable operators, a boundary approach is proposed to get rid of this fake boundary.

The result shows that both these approaches are efficient in fake boundary avoidance. The drawback of these methods is that they are problem-specific, they need to be tuned for each problem, and this paper does not provide an automatic way to remove fake boundaries.

Finally, a generic implementation of the separator on the visibility constraint has been proposed. This approach shows that sometimes the fake boundary problem can also be avoided by expressing the problem differently. There are, nevertheless, cases in which no false boundary appears in the classical implementation of the separator on the visibility constraint. In these cases, the classical implementation is more efficient than the proposed method, since the latter involves the projection of a separator, which is computationally time-consuming.

References

- [1] E. Seignez, M. Kieffer, A. Lambert, E. Walter, and T. Maurin, “Experimental vehicle localization by bounded-error state estimation using interval analysis,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1084–1089, 2005.
- [2] L. H. Rémy Guyonneau, Sébastien Lagrange and P. Lucidarme, “Guaranteed interval analysis localization for mobile robots,” *Advanced Robotics*, vol. 28, no. 16, pp. 1067–1077, 2014.
- [3] J. P. Merlet, “Interval analysis and robotics,” in *Robotics Research* (M. Kaneko and Y. Nakamura, eds.), (Berlin, Heidelberg), pp. 147–156, Springer Berlin Heidelberg, 2011.
- [4] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer London Ltd, Aug. 2001. <http://www.springer.com/engineering/computational+intelligence+and+complexity/book/978-1-4471-1067-5>.
- [5] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, “Anchor-based localization via interval analysis for mobile ad-hoc sensor networks,” *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3226–3239, 2009.
- [6] A. Chauhan, P. Vyas, L. Vachhani, and A. Maity, “Optimal path planning for a non-holonomic robot using interval analysis,” in *2018 Indian Control Conference (ICC)*, pp. 184–189, 2018.
- [7] L. Jaulin, “Path Planning Using Intervals and Graphs,” *Reliable Computing*, vol. 7, no. 1, pp. 1–15, 2001.
- [8] J. Vehí, I. Ferrer, and M. Ángel Sainz, “A survey of applications of interval analysis to robust control,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 389–400, 2002. 15th IFAC World Congress.
- [9] A. Rauh and E. P. Hofer, “Interval methods for optimal control,” in *Variational Analysis and Aerospace Engineering*, (New York, NY), pp. 397–418, Springer New York, 2009.
- [10] R. Stoll, *Set Theory and Logic*. Dover books on advanced mathematics, Dover Publications, 1979.
- [11] L. Jaulin, M. Kieffer, O. Didrit, E. Walter, L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Interval analysis*. Springer, 2001.
- [12] G. Chabert and L. Jaulin, “Contractor programming,” *Artificial Intelligence*, vol. 173, pp. 1079–1100, 07 2009.

- [13] A. Welte, L. Jaulin, M. Ceberio, and V. Kreinovich, “Avoiding Fake Boundaries in Set Interval Computing,” *Journal of Uncertain Systems*, vol. 11, no. 2, pp. 137 – 148, 2017.
- [14] P. Andrews, *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Applied Logic Series, Springer Netherlands, 2013.
- [15] R. Guyonneau, *Méthodes Ensemblistes Pour La Localisation En Robotique Mobile*. PhD Thesis, Angers, Jan. 2013.
- [16] J. Munkres, *Topology*. Featured Titles for Topology, Prentice Hall, Incorporated, 2000.
- [17] M. law Wysocki and A. Darmochwa, “l. subsets of topological spaces,” *Journal of Formalized Mathematics*, vol. 1, 1989.
- [18] L. Jaulin, “A boundary approach for set inversion,” *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104184, 2021.
- [19] T. tom Dieck, *Algebraic topology*, vol. 8. European Mathematical Society, 2008.
- [20] S. Rohou, B. Desrochers, and F. Le Bars, “The codac library,” *Acta Cybernetica*, Mar. 2024.
- [21] J. Ninin, “Global optimization based on contractor programming: An overview of the ibex library,” in *Mathematical Aspects of Computer and Information Sciences* (I. S. Kotsireas, S. M. Rump, and C. K. Yap, eds.), (Cham), pp. 555–559, Springer International Publishing, 2016.