Image Shape Extraction using Interval Methods

L. Jaulin * and S. Bazeille **

* *ENSIETA, 2 rue François Verny, 29806, Brest. email: jaulinlu@ensieta.fr, web: http://www.ensieta.fr/jaulin/*
** *ENSTA, UEI Lab, 32, Boulevard Victor, 75015 Paris, email: stephane.bazeille@ensta.fr*

Abstract: This paper proposes a new method for recognition of geometrical shapes (such as lines, circles or ellipsoids) in an image. The main idea is to transform the problem into a bounded error estimation problem and then to use an interval-based method which is robust with respect to outliers. The approach is illustrated on an image taken by an underwater robot where a spheric buoy has to be detected. The results will then be compared to those obtained by the more classical generalized Hough transform.

Keywords: constraint propagation, Hough transform, image, interval analysis, outliers.

1. INTRODUCTION

The problem to be considered in this paper is to extract known shapes such as ellipses, circles or lines from an image Russ [2002] Bovik [2000]. Fig. 1 represents a photo taken by the underwater robot Sauc'isse (Fig. 2) that participated in the SAUC'E (Student Autonomous Underwater Competition, European) competition that took place in July 2008 in Brest, France. For more information, see

`www.dstl.gov.uk/news_events/competitions/sauce/`

On Fig. 1 we have a spheric buoy that had to be detected by the robot in order to touch it. In this paper, we search for an automatic method to detect some given geometrical shapes in the image and also return the parameters of the shape. For instance, in Fig. 1, the shape to be extracted is a circle and has three parameters: the two coordinates of the center and the radius. The first steps to be performed before the shape extraction are a preprocessing of the image Bazeille [2008] followed by an edge detection Canny [1986]. The resulting black-and-white image is represented on Fig. 3. The principle of the edge detection is to compute the modulus of the gradient of the image and then to threshold the resulting gradient image. All edge points (in white) of the edge image could potentially belong to the edge of a shape. The approach to be considered here is to find the parameters of a chosen shape that are consistent with a given percentage of the edge points. For our buoy example, the shape will be a circle.

Section 2 shows that our shape extraction problem translates into an implicit parameter estimation problem which requires a resolution robust with respect to outliers. Some notions on interval propagation are presented on Section 3. These notions are then used on Section 4 to build a robust

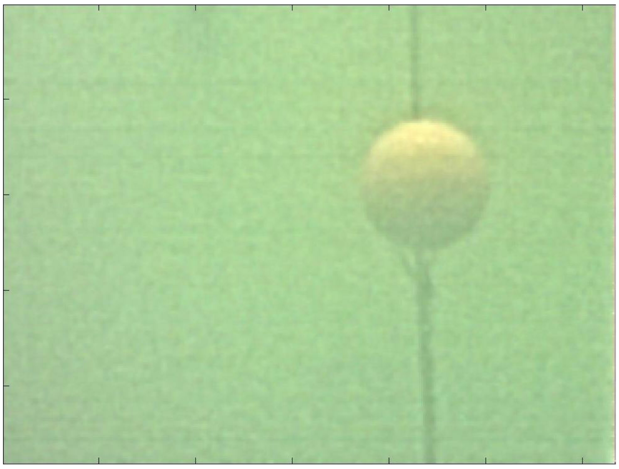

Fig. 1. A spheric buoy taken by the underwater robot Sauc'isse

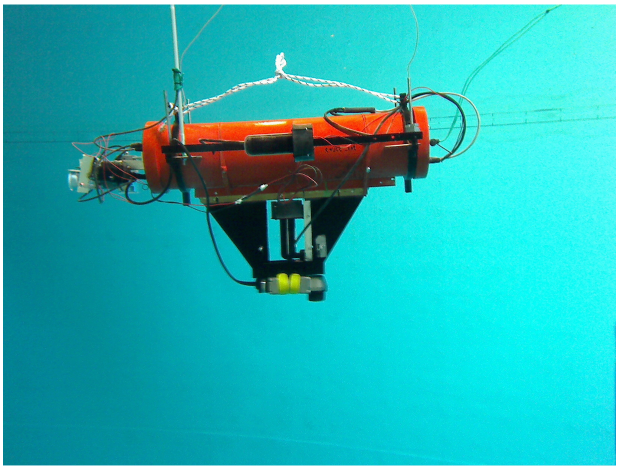

Fig. 2. Sauc'isse robot inside a swimming pool

⋆ This work was supported in part by the GESMA (Groupe Etude Sous Marine Atlantique).

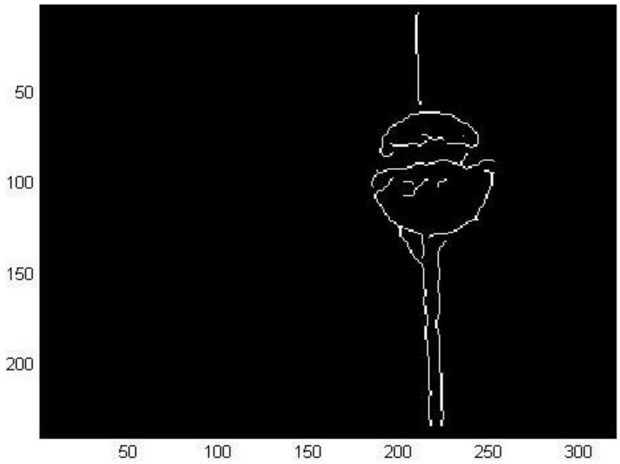

Fig. 3. Image after an edge detection

interval based set estimation algorithm. Some results are given on Section 5 and Section 6 concludes the paper.

2. APPROACH

2.1 Set estimation

In a bounded-error estimation context Walter and Pronzato [1997], a parameter estimation problem can generally be cast into the problem of characterizing a set defined by

$$\mathbb{P} = \bigcap_{i\in\{1,\ldots,m\}} \underbrace{\{\mathbf{p}\in\mathbb{R}^{n_p}, \exists\mathbf{y}\in[\mathbf{y}](i), \mathbf{f}(\mathbf{p},\mathbf{y}) = \mathbf{0}\}}_{\mathbb{P}_i} \qquad (1)$$

where $\mathbf{p}$ is the parameter vector, $[\mathbf{y}](i) \subset \mathbb{R}^{n_y}$ is the ith measurement box and $\mathbf{f} : \mathbb{R}^{n_p} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_f}$ is the model function (latter we will see that n_f can be greater than 1). The set $\mathbb{P}_i$ is the set of all parameter vectors that are consistent with the ith measurement vector. This estimation problem can be qualified as *implicit* because of it involves an implicit equation $\mathbf{f}(\mathbf{p},\mathbf{y}) = \mathbf{0}$. In the particular situation where $\mathbf{y}$ can be isolated, *i.e.*, $\mathbf{f}(\mathbf{p},\mathbf{y}) = \mathbf{0} \Leftrightarrow \mathbf{g}(\mathbf{p}) = \mathbf{y}$, we get

$$\begin{aligned}\mathbb{P} &= \bigcap_{i\in\{1,\ldots,m\}} \underbrace{\{\mathbf{p}\in\mathbb{R}^{n_p}, \mathbf{g}(\mathbf{p})\in[\mathbf{y}](i)\}}_{\mathbb{P}_i} \\ &= \bigcap_{i\in\{1,\ldots,m\}} \mathbf{g}^{-1}([\mathbf{y}](i)),\end{aligned}$$

and the problem becomes a set inversion problem Jaulin and Walter [1993].

Example: Consider the example taken from Jaulin and Walter [1999] where one wants to find the set of parameter vectors $\mathbf{p} = (p_1, p_2)^{\mathrm{T}}$ such that the graph of the function

$$20\exp(-p_1 t) - 8\exp(-p_2 t),$$

goes through all ten boxes of Fig. 4. For this problem, the model function is

$$f(\mathbf{p},\mathbf{y}) = 20\exp(-p_1 y_1) - 8\exp(-p_2 y_1) - y_2,$$

and the boxes $[\mathbf{y}](1), \ldots, [\mathbf{y}](10)$ are those represented on Fig. 4.

2.2 Shape extraction as a set estimation problem

Consider the *shape function* (which is analog to the model function presented above)

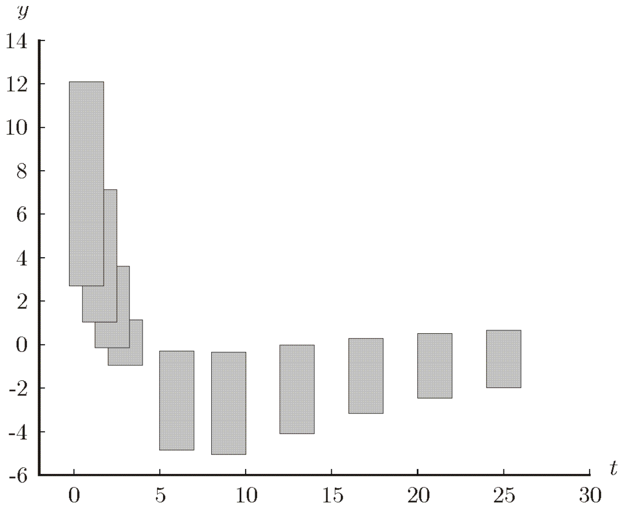

Fig. 4. Parameter estimation problem

$$\mathbf{f} : \begin{cases} \mathbb{R}^{n_p} \times \mathbb{R}^2 & \to \mathbb{R}^{n_f} \\ (\mathbf{p},\mathbf{y}) & \to \mathbf{f}(\mathbf{p},\mathbf{y}). \end{cases}$$

The $\mathbb{R}^2$ space corresponds to the image, the vector $\mathbf{y}$ to a pixel of this image and $\mathbf{p}$ is the parameter vector of the chosen shape. We define the *shape* associated with $\mathbf{p}$ as follows

$$\mathcal{S}(\mathbf{p}) \overset{\text{def}}{=} \{\mathbf{y}\in\mathbb{R}^2, \mathbf{f}(\mathbf{p},\mathbf{y}) = \mathbf{0}\}.$$

Consider a set of (small) boxes in the image

$$\mathcal{Y} = \{[\mathbf{y}](1), \ldots, [\mathbf{y}](m)\}.$$

Each of this box is assumed to intersect the edge of the shape we want to extract.

Remark: In our buoy example, $\mathcal{Y}$ corresponds to small boxes with center the edge points in Fig. 3 and with a width of 2 pixels. Since we want to extract a circle, the shape function is

$$f(\mathbf{p},\mathbf{y}) = (y_1 - p_1)^2 + (y_2 - p_2)^2 - p_3^2.$$

The parameter vector is $\mathbf{p} = (p_1, p_2, p_3)^{\mathrm{T}}$ where p_1, p_2 are the coordinates of the center of the circle and p_3 is its radius.

The feasible set $\mathbb{P}$ is the set of all $\mathbf{p}$ such that the corresponding shape crosses all boxes $[\mathbf{y}](i)$. Now, in our shape extraction problem, a lot of boxes $[\mathbf{y}](i)$ do not correspond to the shape and should then be considered as outliers. As a consequence, except in atypical situations, the resulting feasible set $\mathbb{P}$ is empty. The following section explains how one can robustify the estimation process with respect to some outliers.

2.3 Robust set estimation

Consider m sets $\mathbb{X}_1, \ldots, \mathbb{X}_m$ of $\mathbb{R}^n$. The q-relaxed intersection denoted by $\overset{\{q\}}{\bigcap} \mathbb{X}_i$ is the set of all $\mathbf{x}\in\mathbb{R}^n$ which belong to all $\mathbb{X}_i$'s, except q at most. Fig. 5 illustrates this notion for $m = 6$ and $q = 2, 3, 4$. For this example, we have

$$\overset{\{0\}}{\bigcap} \mathbb{X}_i = \overset{\{1\}}{\bigcap} \mathbb{X}_i = \emptyset, \ \overset{\{5\}}{\bigcap} \mathbb{X}_i = \bigcup \mathbb{X}_i \text{ and } \overset{\{6\}}{\bigcap} \mathbb{X}_i = \mathbb{R}^2.$$

In the same way, we define the q relaxed feasible set as

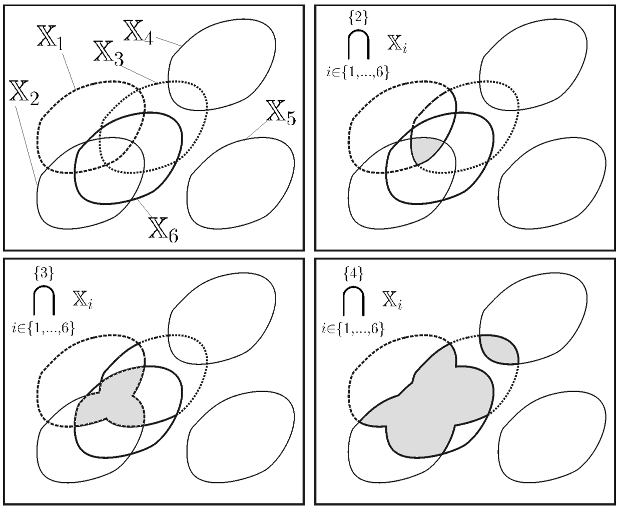
$\mathbb{X}_1$
$\mathbb{X}_4$
$\mathbb{X}_3$
$\mathbb{X}_2$
$\mathbb{X}_5$
$\mathbb{X}_6$
$\bigcap_{i\in\{1,\dots,6\}}^{\{2\}} \mathbb{X}_i$
$\bigcap_{i\in\{1,\dots,6\}}^{\{3\}} \mathbb{X}_i$
$\bigcap_{i\in\{1,\dots,6\}}^{\{4\}} \mathbb{X}_i$

Fig. 5. Illustration (in gray) of the q-relaxed intersection the 6 sets $\mathbb{X}_1, \dots, \mathbb{X}_6$ where $q \in \{2, 3, 4\}$

$$\mathbb{P}^{\{q\}} \overset{\text{def}}{=} \bigcap_{i\in\{1,\dots,m\}}^{\{q\}} \left\{\mathbf{p} \in \mathbb{R}^{n_p}, \exists \mathbf{y} \in [\mathbf{y}](i), \mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}\right\}. \tag{2}$$

The characterization of $\mathbb{P}^{\{q\}}$ can be done efficiently using interval techniques.

3. INTERVAL PROPAGATION

With an interval approach, a random variable x of $\mathbb{R}$ is often represented by an interval $[x]$ which encloses the support of its probability function. This representation is of course poorer than that provided by its probability density distribution, but it presents several advantages. *(i)* Since an interval with non zero length is consistent with an infinite number of probability distribution functions, an interval representation is well adapted to represent random variables with imprecise probability density functions. *(ii)* An arithmetic can be developed for intervals, which makes it possible to deal with uncertainties in a reliable and easy way, even when strong nonlinearities occur. *(iii)* When the random variables are related by constraints (*i.e.*, equations or inequalities) a propagation process (which will be explained later) makes it possible to get efficient polynomial algorithms to compute intervals that are guaranteed to contain all feasible values for the random variables.

3.1 Interval arithmetic

An interval is a closed and connected subset of $\mathbb{R}$. Consider two intervals $[x]$ and $[y]$ and an operator $\diamond \in \{+, -, ., /\}$, we define $[x] \diamond [y]$ as the smallest interval which contains all feasible values for $x \diamond y$, if $x \in [x]$ and $y \in [y]$ (see Moore [1979]). For instance

$$\begin{aligned} [-1,3] + [2,5] &= [1,8], \\ [-1,3].[2,5] &= [-5,15], \\ [-1,3]/[2,5] &= [-\frac{1}{2}, \frac{3}{2}]. \end{aligned}$$

If f is an elementary function such as $\sin, \cos, \dots$ we define $f([x])$ as the smallest interval which contains all feasible values for $f(x)$, if $x \in [x]$.

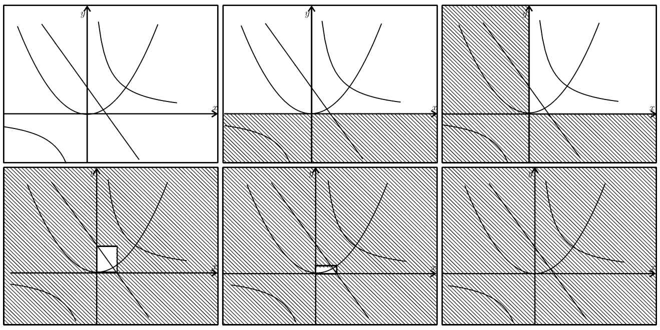

Fig. 6. Illustration of the propagation procedure

3.2 Contractors

Consider a constraint $\mathcal{C}$ (*i.e.*, an equation or an inequality), some variables $x_1, x_2, \dots$ involved in $\mathcal{C}$ and prior interval domains $[x_i]$ for the x_i's. Interval arithmetic makes it possible to contract the domains $[x_i]$ without removing any feasible values for the x_i's. For instance, consider the equation $x_3 = x_1 + x_2$ where the domains for x_1, x_2, x_3 are given by $[x_1] = [-\infty, 5], [x_2] = [-\infty, 4]$ and $[x_3] = [6, \infty]$. These domains can be contracted to $[\bar{x}_1] = [2, 5], [\bar{x}_2] = [1, 4]$ and $[\bar{x}_3] = [6, 9]$. The resulting interval calculation is as follows:

$$\begin{aligned} x_3 = x_1 + x_2 \Rightarrow z \in\ & [6,\infty] \cap ([-\infty,5] + [-\infty,4]) \\ &= [6,\infty] \cap [-\infty,9] = [6,9]. \\ x_1 = x_3 - x_2 \Rightarrow x \in\ & [-\infty,5] \cap ([6,\infty] - [-\infty,4]) \\ &= [-\infty,5] \cap [2,\infty] = [2,5]. \\ x_2 = x_3 - x_1 \Rightarrow y \in\ & [-\infty,4] \cap ([6,\infty] - [-\infty,5]) \\ &= [-\infty,4] \cap [1,\infty] = [1,4]. \end{aligned}$$

This contraction procedure can be performed with much more complex constraints. A contraction operator is called a *contractor.*

3.3 Propagation

When several constraints are involved, the contractions are performed sequentially, until no more significant contraction can be observed (see Jaulin et al. [2001], for more details). To illustrate the propagation process, consider the three following equations

$$\begin{cases} (C_1): & y = x^2 \\ (C_2): & xy = 1 \\ (C_3): & y = -2x + 1. \end{cases}$$

Using interval propagation, we want to prove that this system has no solution. To each of the variables, we assign the domain $[-\infty, \infty]$. Then, we contract the domains with respect to the constraints in the following order: C_1, C_2, C_3, C_1, C_2 and we get empty intervals for x and y. A geometric interpretation of the propagation is given on Fig. 6. The resulting interval computation is as follows.

$$\begin{aligned} (C_1) &\Rightarrow y \in [-\infty,\infty]^2 \ = \ [0,\infty] \\ (C_2) &\Rightarrow x \in 1/[0,\infty] \ = [0,\infty] \\ (C_3) &\Rightarrow y \in [0,\infty] \ \cap \ ((-2).[0,\infty] + 1) \\ &\qquad = [0,\infty] \ \cap \ ([-\infty,1]) = [0,1] \\ &\qquad x \in [0,\infty] \ \cap \ (-[0,1]/2 + 1/2) = [0, \frac{1}{2}] \\ (C_1) &\Rightarrow y \in [0,1] \cap [0,1/2]^2 = [0,1/4] \\ (C_2) &\Rightarrow x \in [0,1/2] \ \cap \ 1/[0,1/4] = \emptyset \\ &\qquad y \in [0,1/4] \cap 1/\emptyset = \emptyset. \end{aligned}$$

3.4 Forward-backward propagation

The interval propagation method converges to a box which contains all solution vectors of our set of constraints. If this box is empty, it means that there is no solution. It can be shown that the box to which the method converges does not depend on the order to which the contractors are applied Jaulin et al. [2001], but the computing time is highly sensitive to this order. There is no optimal order in general, but in practice, one of the most efficient is called *forward-backward propagation.* It consists in writing the equation under the form $\mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}$. Then, using interval arithmetic, the intervals are propagated from $\mathbf{p}, \mathbf{y}$ to $\mathbf{0}$ in a first step (*forward propagation*) and, in a second step, the intervals are propagated from $\mathbf{0}$ to $\mathbf{p}, \mathbf{y}$ (*backward propagation*). As an illustration, consider again our shape extraction problem on the buoy image. The following forward-backward contraction algorithm returns the smallest box $[\bar{\mathbf{p}}]$ which encloses the set

$$\left\{\mathbf{p} \in [\mathbf{p}], \exists \mathbf{y} \in [\mathbf{y}], (y_1 - p_1)^2 + (y_2 - p_2)^2 - p_3^2 = 0\right\}.$$

```
FB(in: $[\mathbf{y}], [\mathbf{p}]$, out: $[\bar{\mathbf{p}}]$)
1    $[d_1] := [y_1] - [p_1]$ ;
2    $[d_2] := [y_2] - [p_2]$ ;
3    $[c_1] := [d_1]^2$ ;
4    $[c_2] := [d_2]^2$ ;
5    $[c_3] := [p_3]^2$ ;
6    $[e] := [0, 0] \cap ([c_1] + [c_2] - [c_3])$ ;
7    $[c_1] := [c_1] \cap ([e] - [c_2] + [c_3])$ ;
8    $[c_2] := [c_2] \cap ([e] - [c_1] + [c_3])$ ;
9    $[c_3] := [c_3] \cap ([c_1] + [c_2] - [e])$ ;
10   $[\bar{p}_3] := [p_3] \cap \sqrt{[c_3]}$;
11   $[d_2] := [d_2] \cap \sqrt{[c_2]}$;
12   $[d_1] := [d_1] \cap \sqrt{[c_1]}$;
13   $[\bar{p}_2] := [p_2] \cap ([y_2] - [d_2])$ ;
14   $[\bar{p}_1] := [p_1] \cap ([y_1] - [d_1])$ ;
```

where $[d_1], [d_2], [c_1], [c_2], [c_3], [e]$ are intermediate interval variables of the algorithm. Steps 1 to 6 form the forward propagation and Steps 7 to 14 form the backward step.

4. ROBUST SET ESTIMATION ALGORITHM

4.1 Relaxed intersection

Since the q-relaxed intersection can be written as a combination of unions and intersections, it is inclusion monotonic, *i.e.*,

$$(\mathbb{X}_1 \subset \mathbb{Y}_1, \ldots, \mathbb{X}_m \subset \mathbb{Y}_m) \Rightarrow \bigcap_{i \in \{1,\ldots,m\}}^{\{q\}} \mathbb{X}_i \subset \bigcap_{i \in \{1,\ldots,m\}}^{\{q\}} \mathbb{Y}_i.$$

This inclusion monotonicity is illustrated by Fig. 7 in the case where the $\mathbb{Y}_i$'s are boxes. Note that the q-relaxed intersection of m boxes is not necessarily a box, but we can easily compute the smallest box which contains the relaxed intersection.

Computing the q relaxed intersection of m boxes has a polynomial complexity, if the dimension n of the boxes is fixed (see, *e.g.*, B. Rosgen [2007]), but the complexity of this problem is exponential with respect to n. Fig. 8 illustrates the principle of such an algorithm. First, generate $(2m - 1)^n$ boxes as on the subfigure (b). Select all boxes whose centers belong to at least $m - q$ of the m initial boxes Finally, take the enveloping box of all selected boxes.

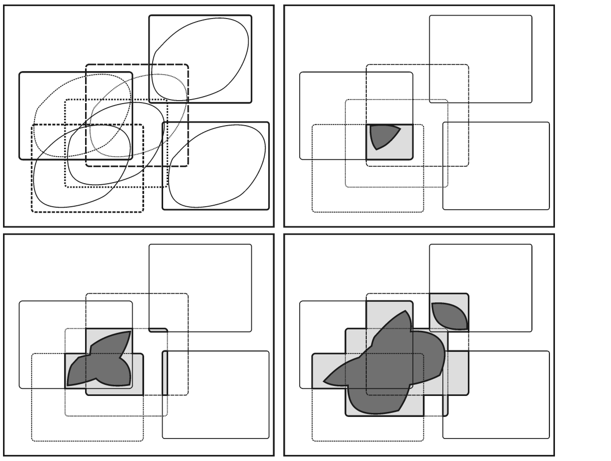

Fig. 7. Illustration of the inclusion monotonicity of the q-relaxed intersection (in dark grey for the $\mathbb{X}_i$'s and in light grey for the $\mathbb{Y}_i$'s)

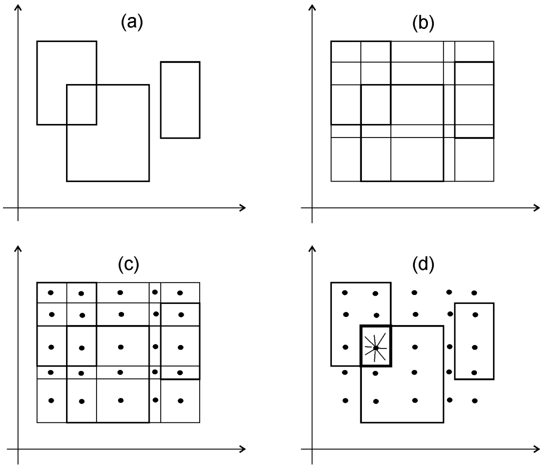

Fig. 8. Illustration of the principle of polynomial method which computes the smallest box which contains the q-relaxed intersection of m boxes. Here, $q = 1, m = 3$.

4.2 Algorithm

We shall now present an algorithm to characterize the q-relaxed feasible set $\mathbb{P}^{\{q\}}$ Jaulin and Walter [2002], Jaulin [2008]. The principle of the method is illustrated by Fig. 9. Fig 9,(a) represents the sets $\mathbb{P}_i$ with the solution set $\mathbb{P}^{\{q\}}$ (hatched), representing the q-relaxed intersection we would like to enclose (here, $q = 1$). For each i, we first enclose the sets $[\mathbf{p}] \cap \mathbb{P}_i$ by boxes $[\mathbf{p}](i)$ as represented with dash line boxes on Fig 9,(b). On Fig 9,(c), the two grey boxes represents the q-relaxed intersection of the boxes $[\mathbf{p}](i)$. We compute a box enclosure (hatched box) of this q-relaxed intersection. On Fig 9,(d), we are in the same situation as we were on Fig 9,(a). The current box $[\mathbf{p}]$ still encloses $\mathbb{P}^{\{q\}}$ but is now smaller. The process can be iterated once more as illustrated by Fig 9,(e) and Fig 9,(f). We will then converge to a steady box. The accuracy of the enclosure can be controlled by allowing several bisections of the current box $[\mathbf{p}]$ into subboxes and by iterating the

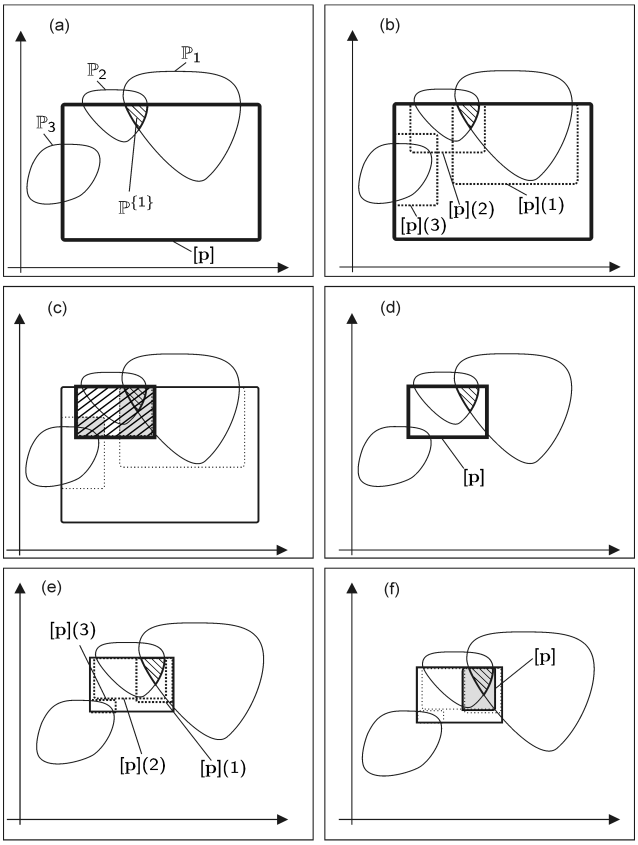

Fig. 9. Principle of the contraction procedure for the relaxed set inversion problem

contraction procedure on each subbox. The corresponding algorithm is given by Table 1.

Table 1. Algorithm for solving the relaxed set inversion problem

```
Algorithm ENCLOSE(in: $[\mathbf{p}], [\mathbf{y}](1), \dots, [\mathbf{y}](m), q$, out: $[\bar{\mathbf{p}}]$)
1     $\mathcal{L} := \{[\mathbf{p}]\}$;
2     repeat
3       pull ($[\mathbf{p}]$,$\mathcal{L}$);
4       while the contraction are significant
5         for $i = 1$ to $m$, compute $[\mathbf{p}](i)$ enclosing $[\mathbf{p}] \cap \mathbb{P}_i$
6         $[\mathbf{p}] := \left[ \bigcap_{i \in \{1,\dots,m\}}^{\{q\}} [\mathbf{p}](i) \right]$
7       end repeat
8       bisect $[\mathbf{p}]$ and push the resulting boxes into $\mathcal{L}$
9     until all boxes of $\mathcal{L}$ have a width smaller than $\varepsilon$.
10    $[\bar{\mathbf{p}}] = \sqcup(\mathcal{L})$
```

Step 1: The list $\mathcal{L}$ contains boxes, the union of which encloses $\mathbb{P}^{\{q\}}$. It is initialized with the single box $[\mathbf{p}]$.

Step 2: A *repeat-until* loop is run until all boxes of $\mathcal{L}$ have a width smaller that a given accuracy ε.

Step 3: The largest box is pulled out from the list.

Step 4: The contraction procedure is iterated until no more significant contraction can be produced.

Step 5: For all i, a box $[\mathbf{p}](i)$ enclosing $[\mathbf{p}] \cap \mathbb{P}_i$ is computed. For the application presented in this paper, a single forward-backward contraction procedure (see the example of Section 3.4) is implemented.

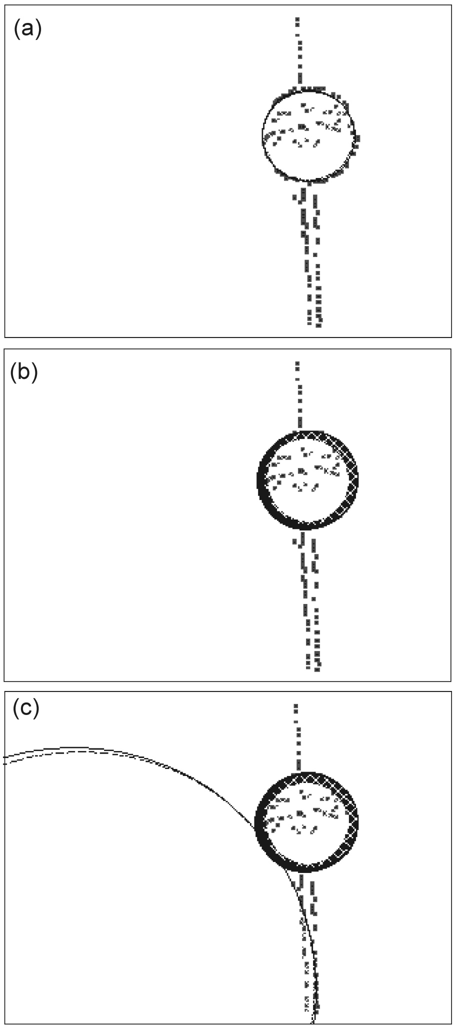

Fig. 10. Extraction of the circle for different q

Step 6: A box enclosing the q-relaxed intersection of the $[\mathbf{p}](i)$'s is computed. Here, $[\mathbb{A}]$ represents a the smallest box enclosing the set $\mathbb{A}$.

Step 8: The current box is bisected into two smaller boxes. These two boxes are pushed at the end of the queue $\mathcal{L}$.

Step 10: The algorithm returns the smallest box $[\bar{\mathbf{p}}]$ enclosing all boxes stored in $\mathcal{L}$ (represented here by the box union operator $\sqcup$).

5. RESULTS

Let us apply the interval method to our shape extraction problem for different values of q. We get Fig. 10. Subfigures (a),(b),(c) represent circles consistent with 30%, 20%, 19% of the data, respectively. The computing time is less than 15 seconds. These results are comparable to those obtain by a classical generalized Hough transform approach devoted to circle extraction Kimme et al. [1975].

An improvement suggested by O'Gorman and Clowes O'Gorman and Clowes [1976], in the context of the Hough transform Duda and Hart [1972], is to take into account the fact that local gradient of the image intensity is orthogonal to the edge (See Fig. 11).

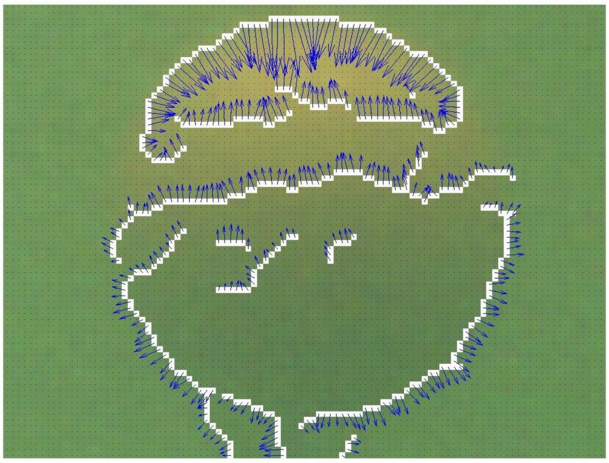

Fig. 11. Gradient of the buoy image (zoom)

Let us append the direction y_3 of the gradient (in radian) to the measurement vector $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$. We thus have a three dimensional measurement vector $\mathbf{y} = (y_1, y_2, y_3)^{\mathrm{T}}$. The gradient condition translates into

$$\det \begin{pmatrix} \dfrac{\partial f(\mathbf{p}, \mathbf{y})}{\partial y_1} & \cos(y_3) \\ \dfrac{\partial f(\mathbf{p}, \mathbf{y})}{\partial y_2} & \sin(y_3) \end{pmatrix} = 0.$$

For the circle detection, since $f(\mathbf{p}, \mathbf{y}) = (y_1 - p_1)^2 + (y_2 - p_2)^2 - p_3^2$, this condition becomes

$$(y_1 - p_1)\sin(y_3) - (y_2 - p_2)\cos(y_3) = 0.$$

The model function to be considered for circle detection becomes

$$\mathbf{f}(\mathbf{p}, \mathbf{y}) = \begin{pmatrix} (y_1 - p_1)^2 + (y_2 - p_2)^2 - p_3^2 \\ (y_1 - p_1)\sin(y_3) - (y_2 - p_2)\cos(y_3) \end{pmatrix}.$$

This new condition introduces new outliers: the edge points that are on the actual shape, but that do not satisfy the gradient condition. For our problem, we are able to be consistent with 20% of the edge points. But the computing time is now significantly reduced since it is now less that 2 seconds instead of 15 seconds when the gradient condition was not used.

6. CONCLUSION

In this paper, a new method for image shape extraction has been presented. The problem has been transformed into a parameter estimation problem. Since the corresponding estimation problem is nonlinear and is corrupted by many outliers, a classical robust interval method has been used. The Builder 5 C++ code is available at

`www.ensieta.fr/jaulin/cpp_hough.zip`

A classical approach for extracting geometrical shapes is based on the generalized Hough transform. The method is very similar to that presented in this paper except that it uses point methods instead of intervals. In short, the principle is to compute the Hough transform defined by

$$\eta(\mathbf{p}) = \mathrm{card}\left\{i \in \{1, \ldots, m\}, \exists \mathbf{y} \in [\mathbf{y}](i), \mathbf{f}(\mathbf{p}, \mathbf{y}) = \mathbf{0}\right\},$$

and then to keep all $\mathbf{p}$ such that $\eta(\mathbf{p})$ is higher than a given threshold $m - q$. Instead, our approach directly solves the inequality $\eta(\mathbf{p}) \geq m - q$.

Acknowledgement: The authors want to thank Alexandre Goldsztejn, Gilles Chabert and Vladik Kreinovich for helpful comments and for giving the idea, the proof and the reference related to the fact that computing the q-relaxed intersection of m boxes was an NP hard problem that could be solved in a polynomial time when n was fixed.

REFERENCES

L. Stewart B. Rosgen. Complexity results on graphs with few cliques. *Discrete Mathematics and Theoretical Computer Science*, 2007.

S. Bazeille. *Vision sous-marine monoculaire pour la reconnaissance d'objets*. PhD thesis, Université de Bretagne Occidentale, 2008.

A. Bovik. *Handbook of image and video processing*. Academic Press, 2000.

J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6*, 1986.

R. O. Duda and P. E. Hart. Use of the hough transform to detect lines and curves in pictures. *Comm. ACM Vol 15.*, 1972.

L. Jaulin. Robust set membership state estimation. *Automatica, accepted*, 2008.

L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29 (4):1053–1064, 1993.

L. Jaulin and E. Walter. Guaranteed bounded-error parameter estimation for nonlinear models with uncertain experimental factors. *Automatica*, 35(5):849–856, 1999.

L. Jaulin and E. Walter. Guaranteed robust nonlinear minimax estimation. *IEEE Transaction on Automatic Control*, 47(11):1857–1864, 2002.

C. Kimme, D. H Ballard, and J. Sklansky. Finding circles by an array of accumulators. *CACM*, 1975.

R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

F. O'Gorman and M.B. Clowes. Finding Picture Edges Through Collinearity of Feature Points. *IEEE Transactions on Computers*, 25(4):449–456, 1976.

J. C. Russ. *Image processing handbook (The)*. CRC Press, 2002.

E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, London, UK, 1997.