A Contractor for the Non-Overlapping of Objects Described by Non-Linear Inequalities

Ignacio SALAS & Gilles CHABERT

MINES Nantes

Content

Definitions

The Non-Overlapping Constraint

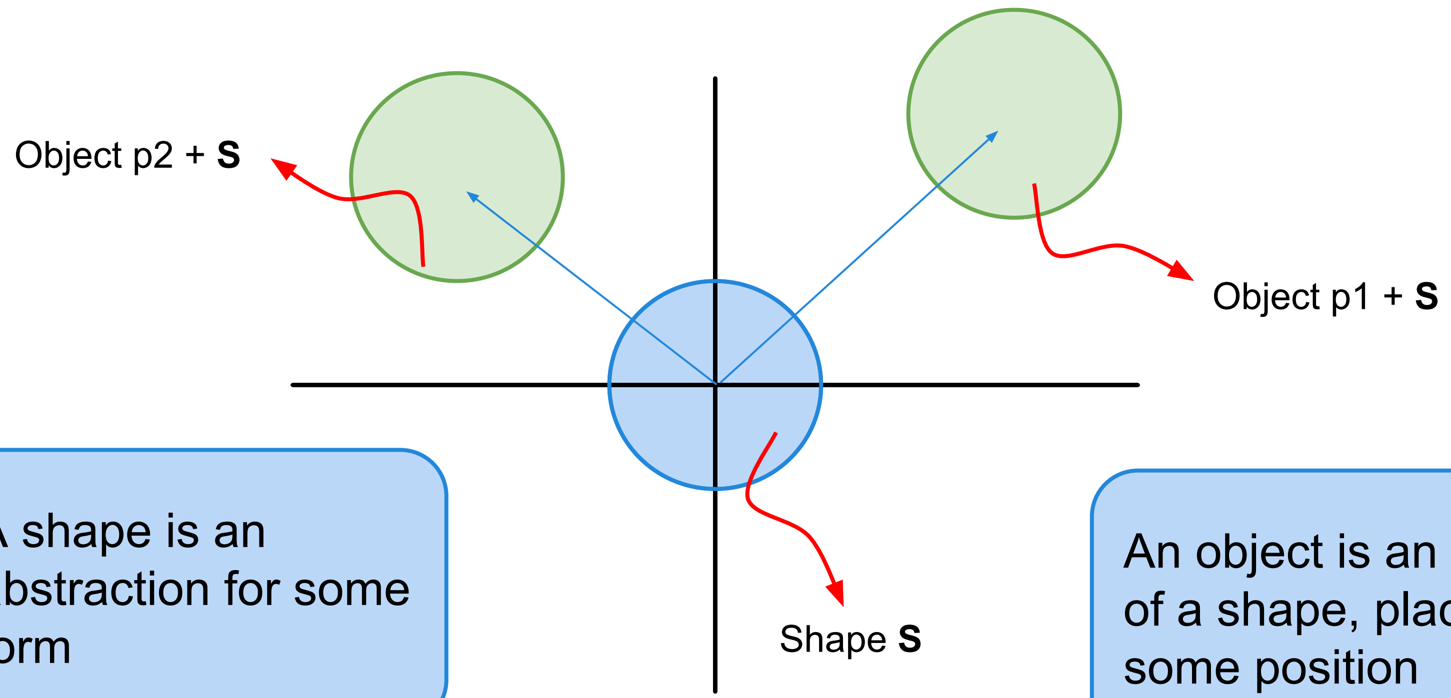
The Contractor

Experiments

Summary

Definitions

About Objects and Shapes

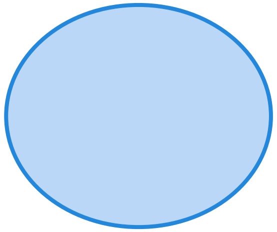
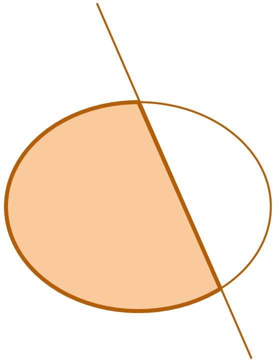
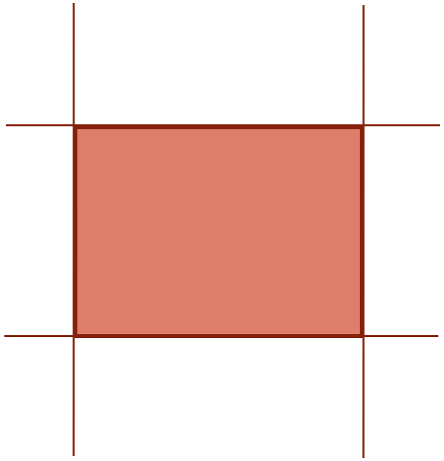
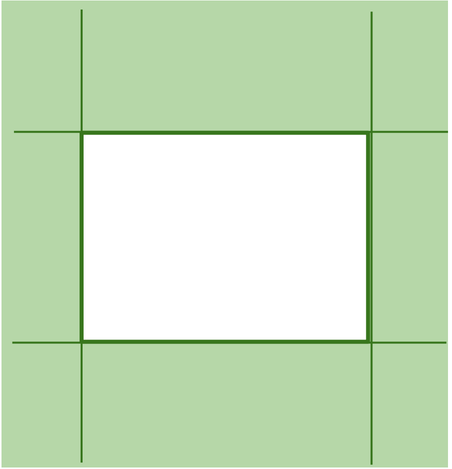


A shape is an abstraction for some form

An object is an instance of a shape, placed at some position

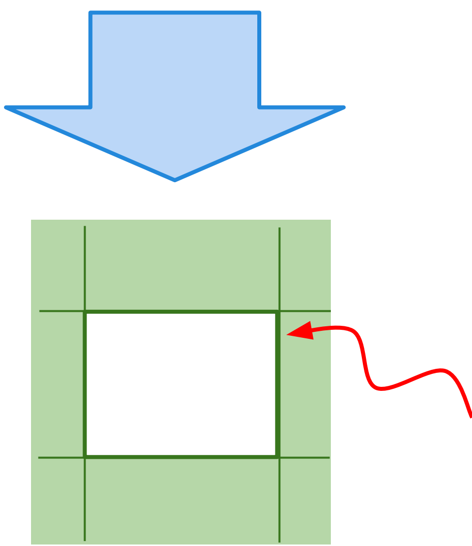
Definitions

- Our target are the **curved shapes**
- That can be represented by **non-linear inequalities**
- In a general way with **disjunctions of system of non-linear inequalities**

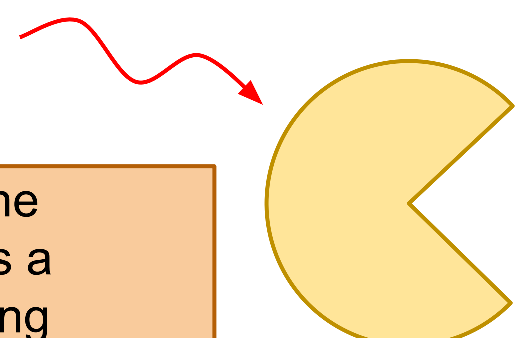





Definitions

$$\text{System}_1 \cup \text{System}_2 \cup \text{System}_3 \cup \text{System}_4$$



The resulting shape is not necessarily convex



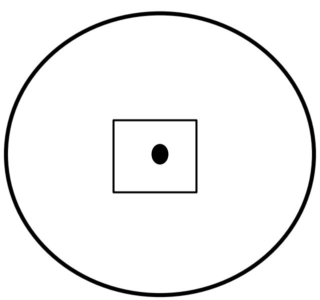
And this allows to represent the avalaible space for packing as a regular shape (called “enclosing shape”)

Definitions

Moving and Reference Objects

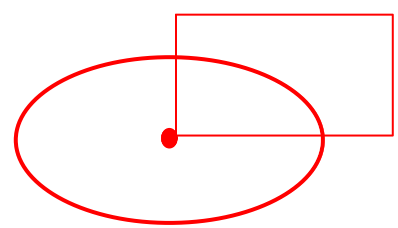
All the objects that we consider must take a **rol** at some moment

Reference Object



It is an object fixed, and it is used as reference for to contract the **origin box** of other object

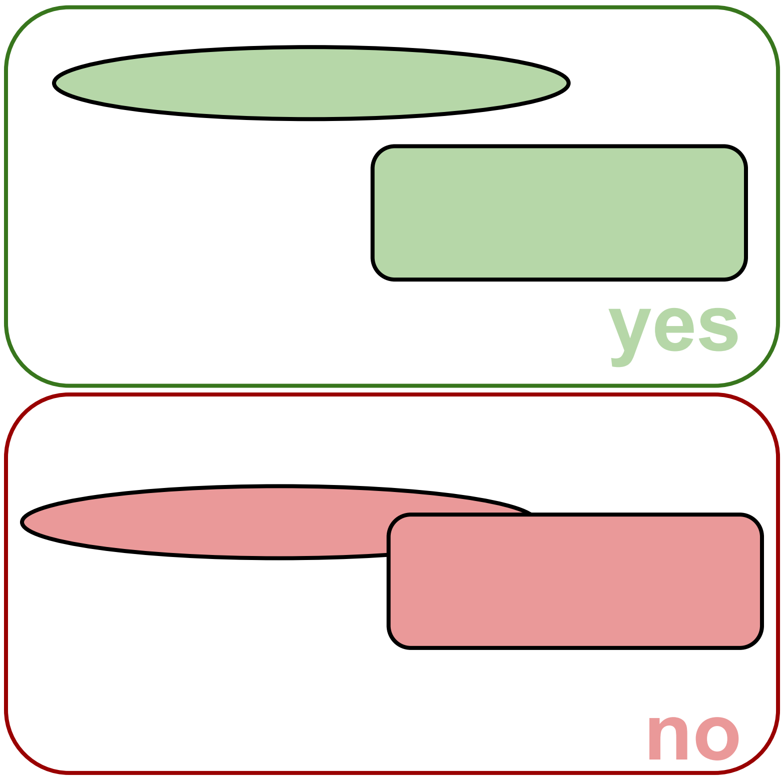
Moving Object



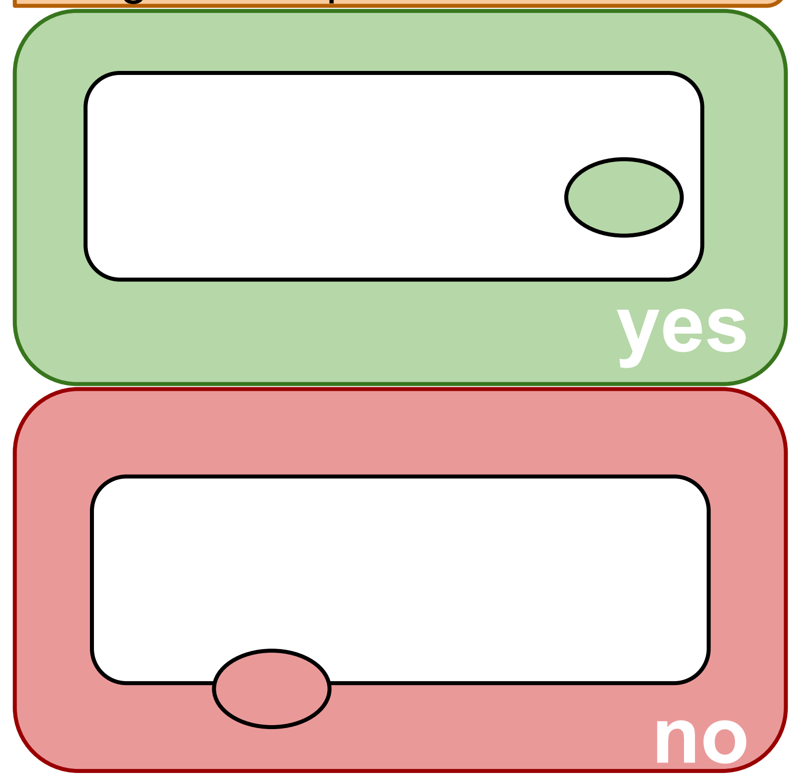
It is an object which **origin box** we want to contract

The Non-Overlapping Constraint

Between Regular Shapes



Between an Enclosing Shape and a regular Shape



The overlapping with the enclosing shape is handled as with any other shape, since the description of the objects allows a transparent process.

The Non-Overlapping Constraint

Domain of the positions of the object **a**

Domain of the positions of the object **b**

$$\forall p_a \in [\boldsymbol{O}_a], \forall p_b \in [\boldsymbol{O}_b],$$

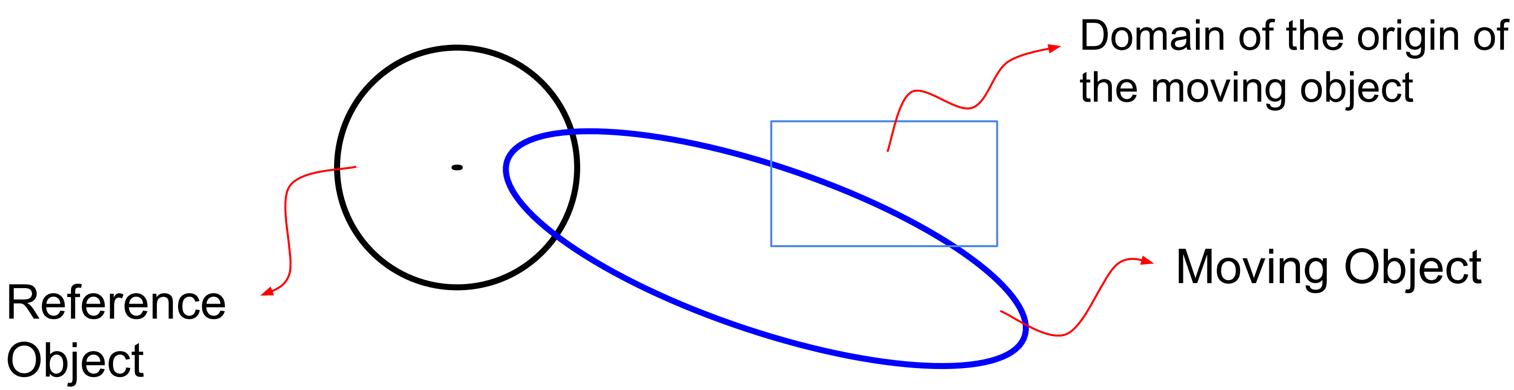
$$(S_a + p_a) \cap (S_b + p_b) = \emptyset$$

Shape **a**

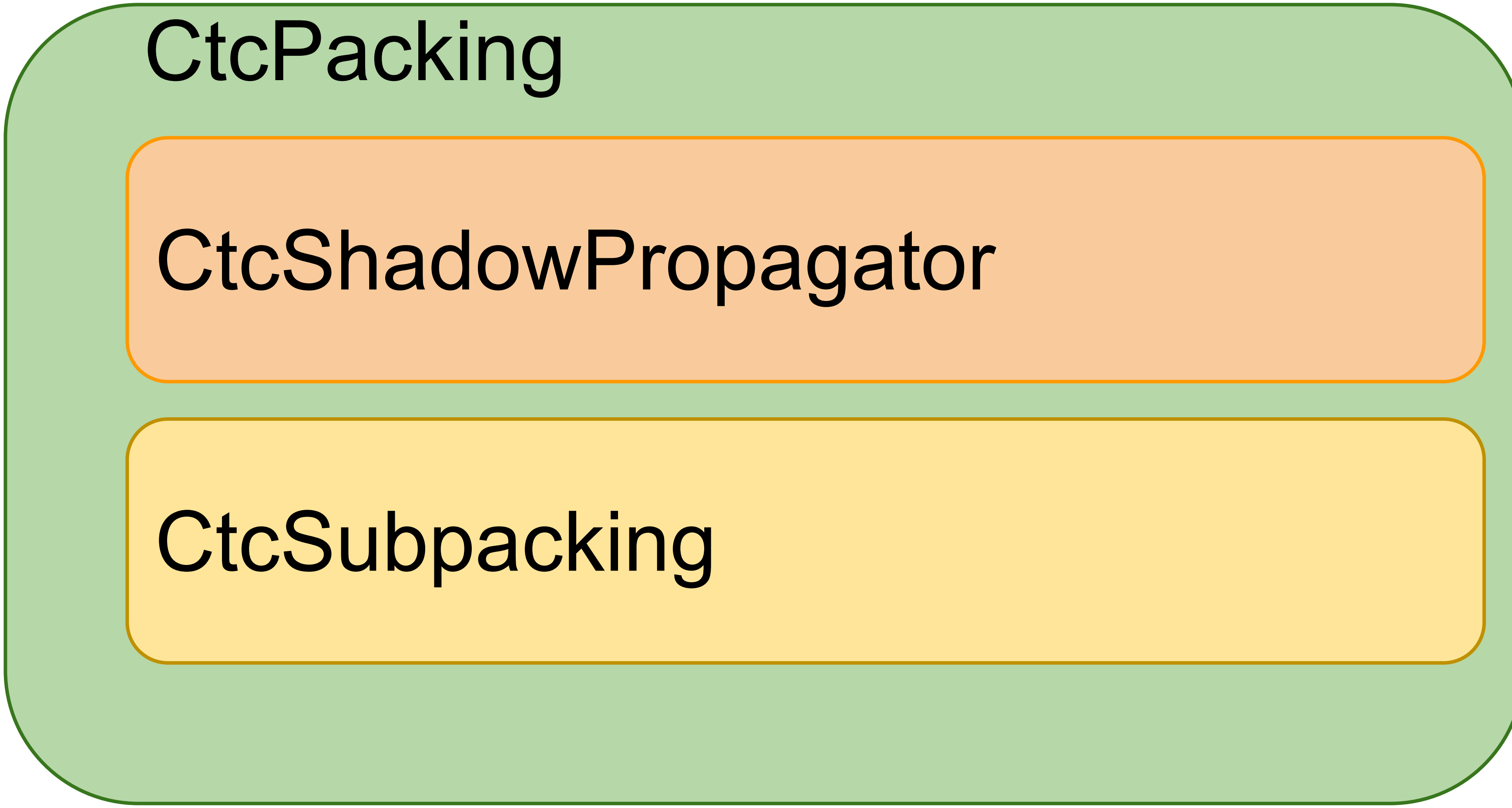
Shape **b**

The Contractor

- **Target**: Remove all the parts that violate the non-overlapping constraint
- The contraction operates over 2 objects
  - The reference object
  - The moving object



The Contractor

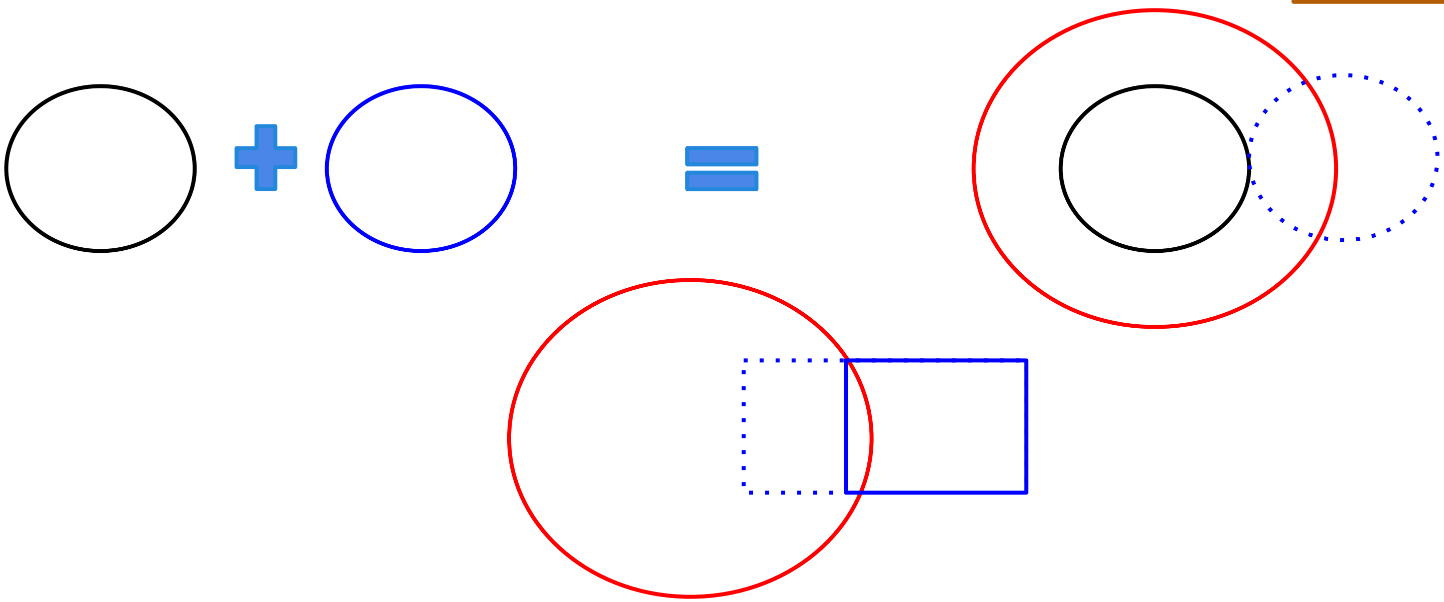


The Contractor

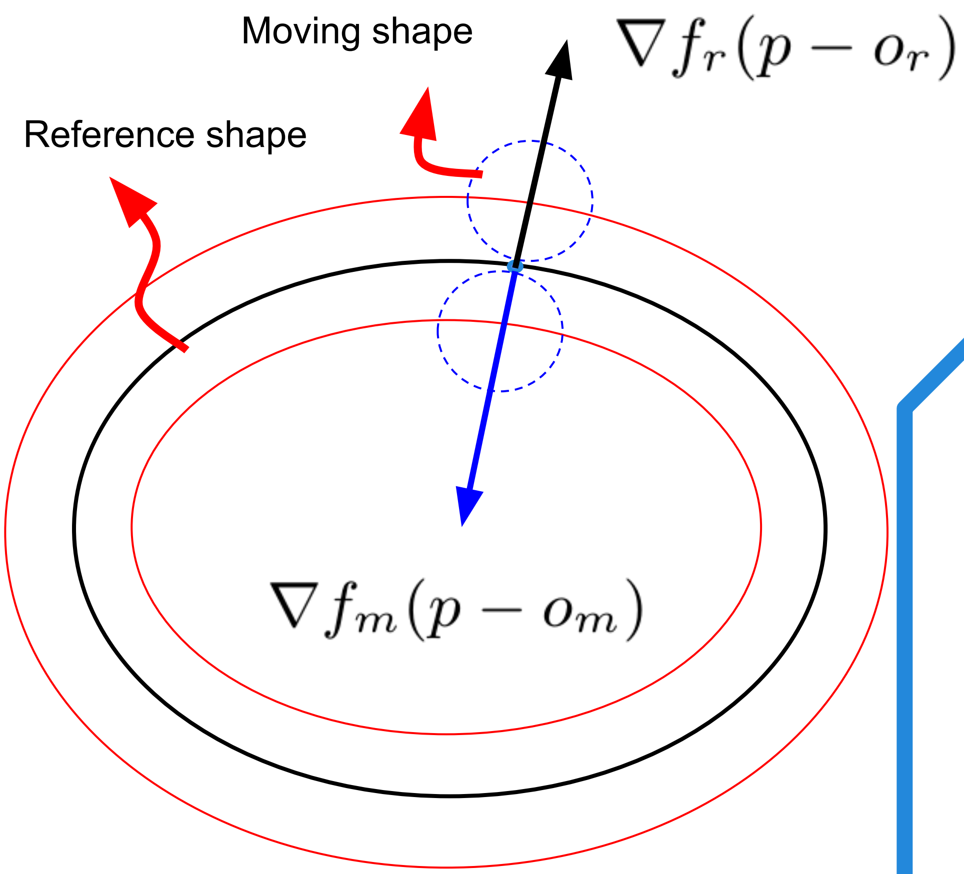
CtcShadowPropagator

It is important not to lose solutions and perform the contraction in an efficient way

The Shadow



The Contractor



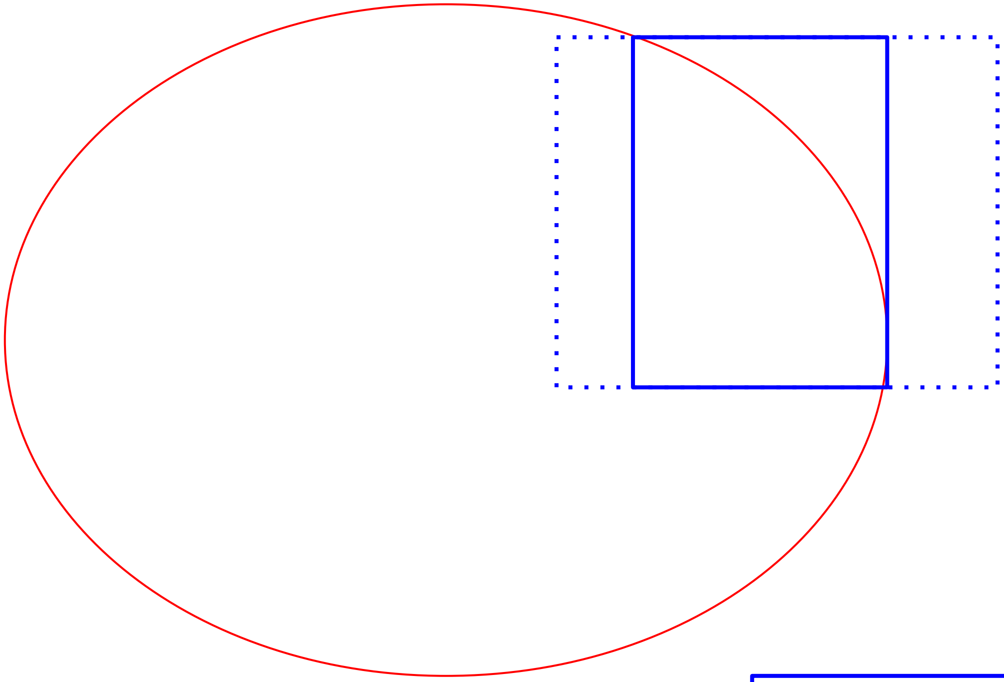
How to calculate the bound of the shadow?

$$f_r(p - o_r) = 0$$

$$f_m(p - o_m) = 0$$

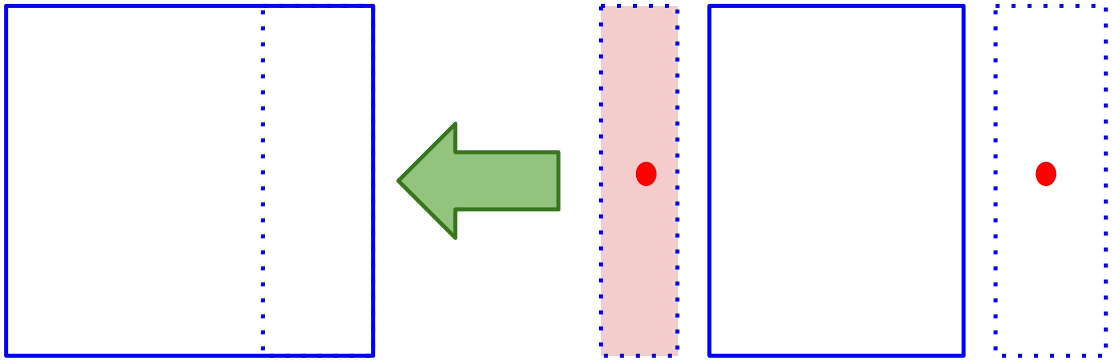
$$\frac{\nabla f_r(p - o_r) \bullet \nabla f_m(p - o_m)}{\|\nabla f_r(p - o_r)\| * \|\nabla f_m(p - o_m)\|} = -1$$

The Contractor



How to obtain the final contracted box for each pair of objects?

Apply the **Ibex** function **diff** with the original box, and check the center of the resulting boxes



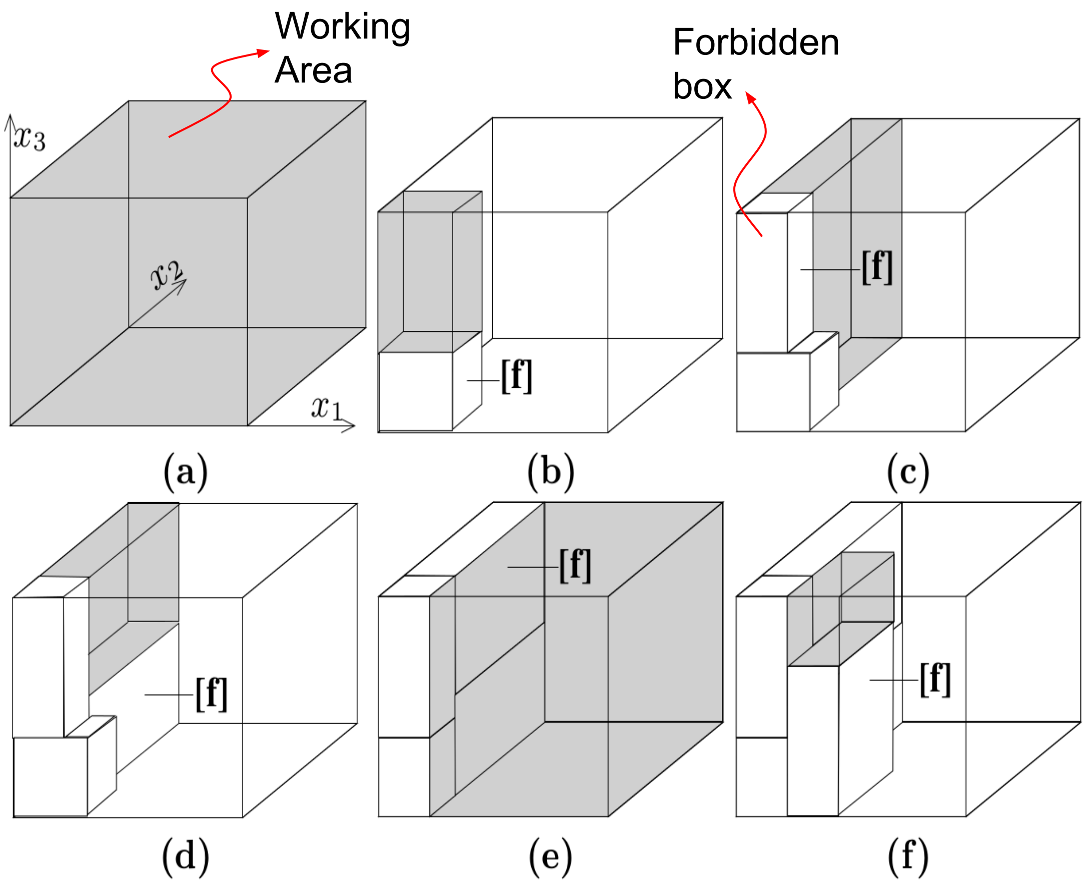
The Contractor

CtcSubpacking

The origin box of each moving object is contracted with **Sweep**.

The Contractor

The Sweep algorithm



The objective it is to find all the **forbidden boxes** in a **working area**, such that it will be possible **prune** a section of the box

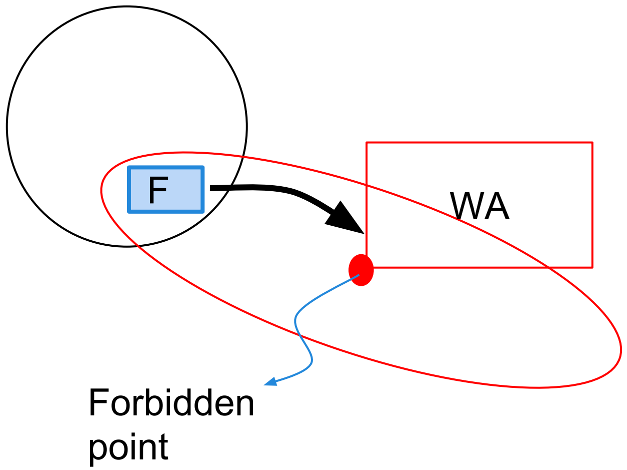
This forbidden boxes are obtained with an **inflator**, that considers one constraint at each time

The Contractor

The Inflator

Our main objective it is to find the biggest box in the intersection of the object

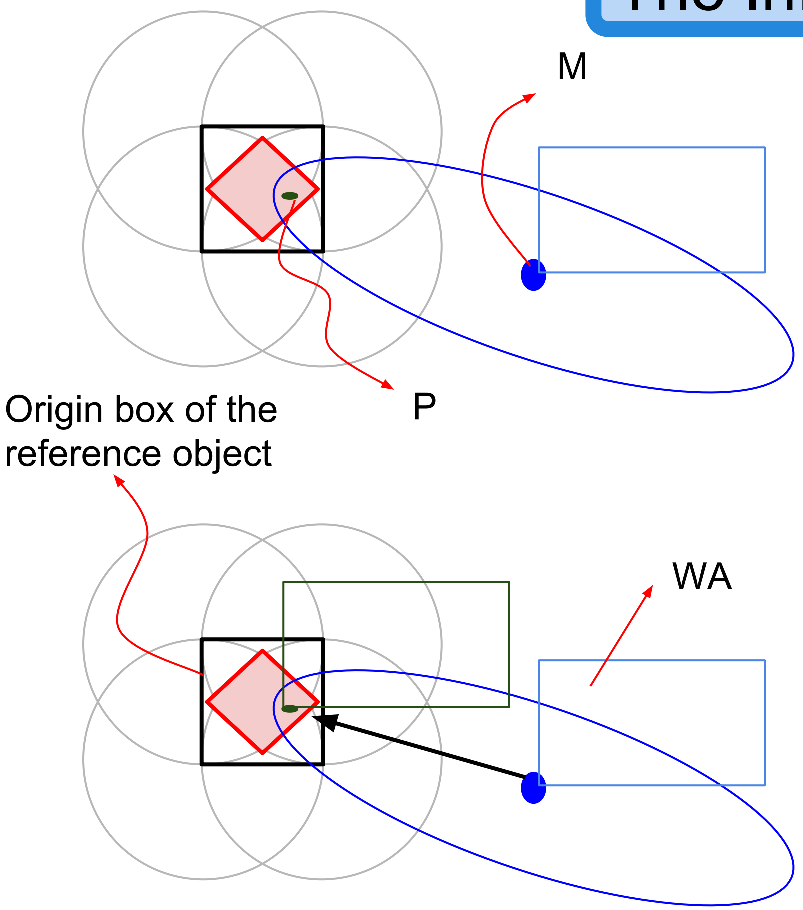
And respect the dimensions of the **Working Area (WA)**



If exists some box **F**, it is possible **translate** it to the **forbidden point**

The Contractor

The Inflator



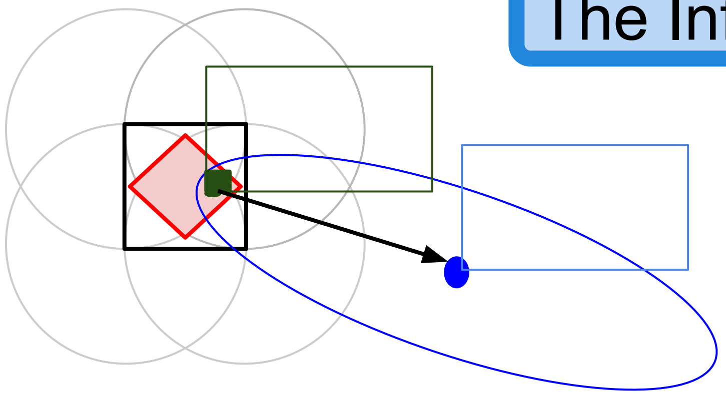
- Center the moving object at M
- Search a point **P** that belongs to the intersection of the two objects

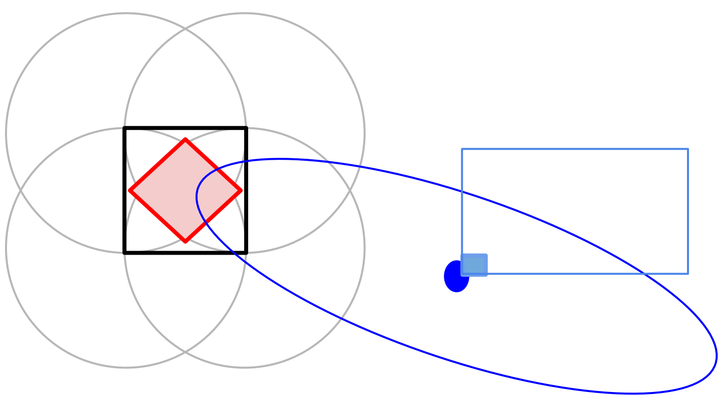
To be explained further

- Translate the WA to **P**

The Contractor

The Inflator





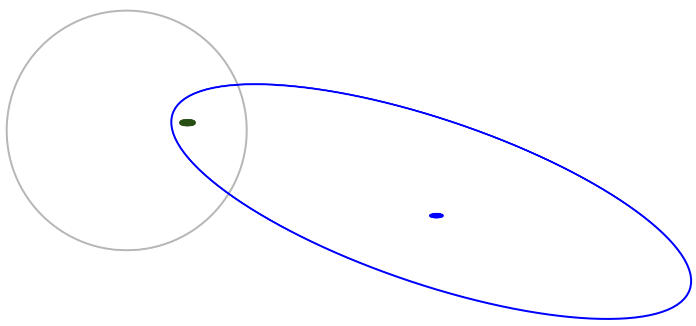
- *Inflate* **P** *to* an inner box of the reference shape

To be explained further

- Translate this box back to the forbidden point

The Contractor

How to obtain the point **P**?



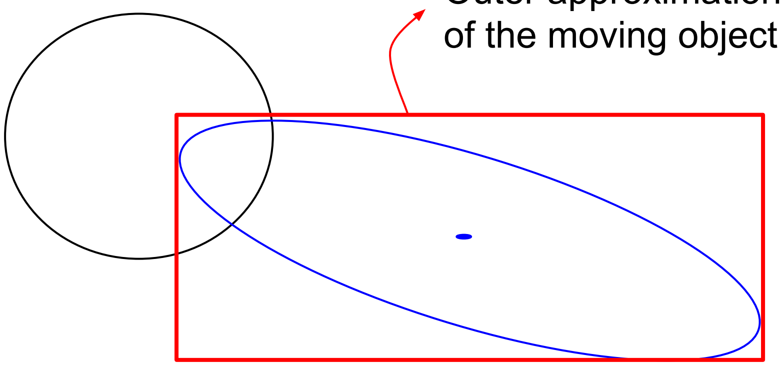
Our objective is find a point that belongs to the moving object and to the **compulsory part** of the reference object

Where always will be possible to find the reference object, given an origin box

The Contractor

How to obtain the point **P**?

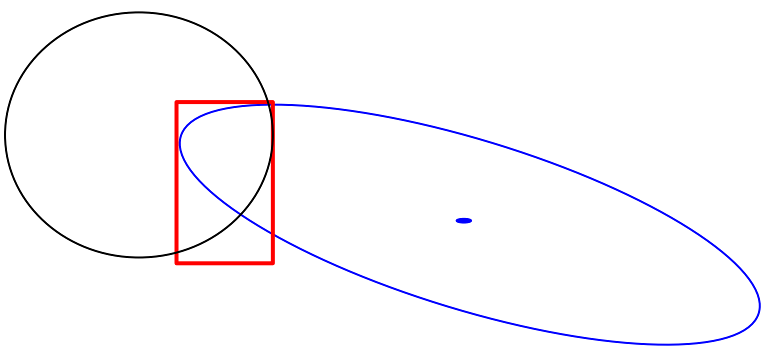
Outer approximation
of the moving object



In the beginning, it is an
unbounded box

Contract some box **C** w.r.t. the moving object

$$o_m + S_m \subseteq C$$

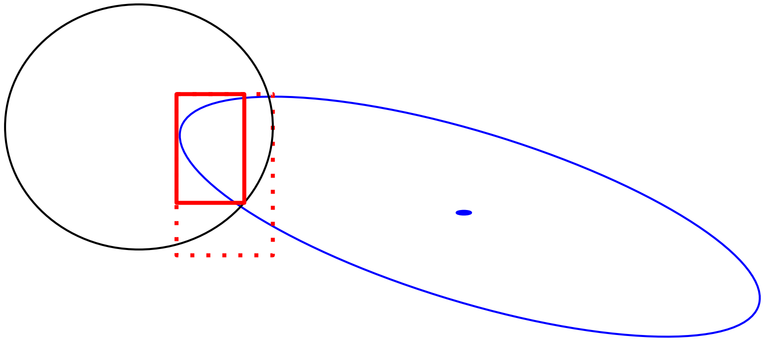


Contract the box **C** w.r.t. the reference
object, centered in the middle point

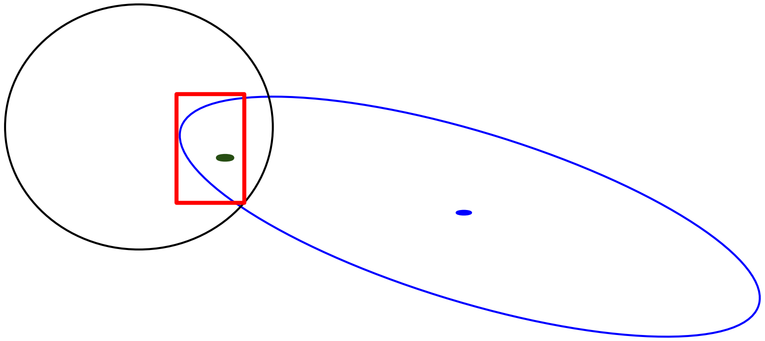
$$o_r + S_r \subseteq C$$

The Contractor

How to obtain the point **P**?



Inflate C w.r.t. the reference object and the middle point of the box **C**



Pick randomly a point in the resulting box and check if it belongs to both objects

Using interval evaluation

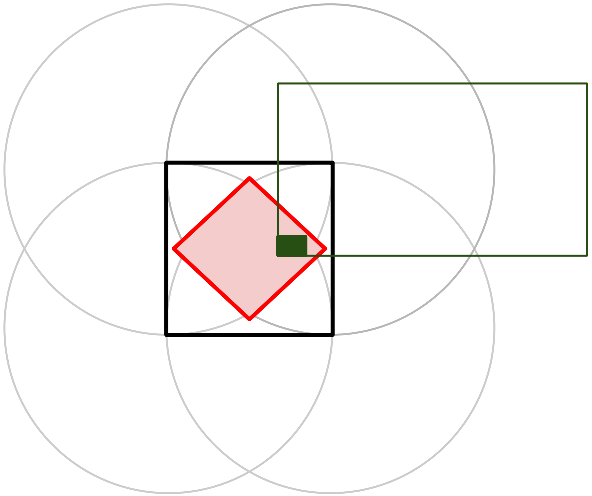
Else, **Bisect**

The contracted box

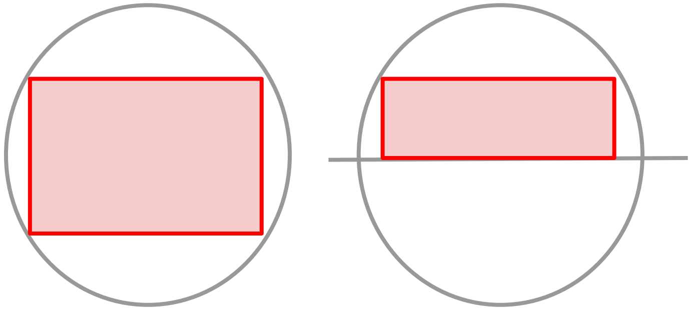
The Contractor

Inflate the point **P**

The point **P** is inflated using the box translated and the compulsory part



The inflators of Ibex can inflate inside inequalities

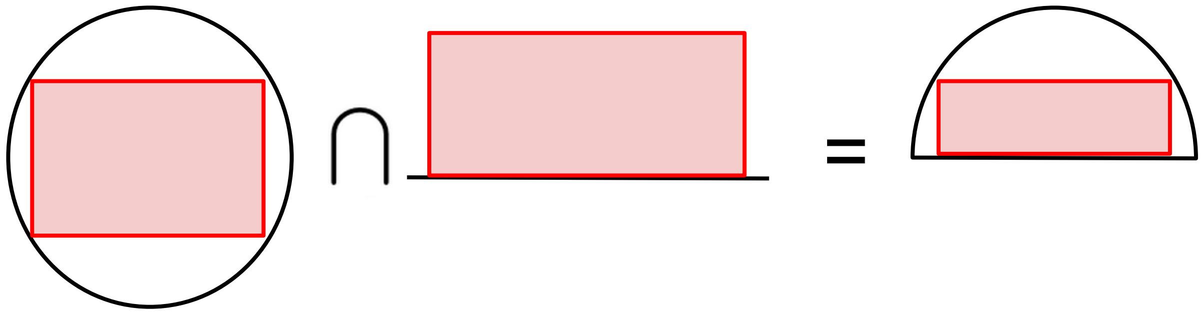


It is possible to inflate inside the same box all the inequalities that are part of the shape

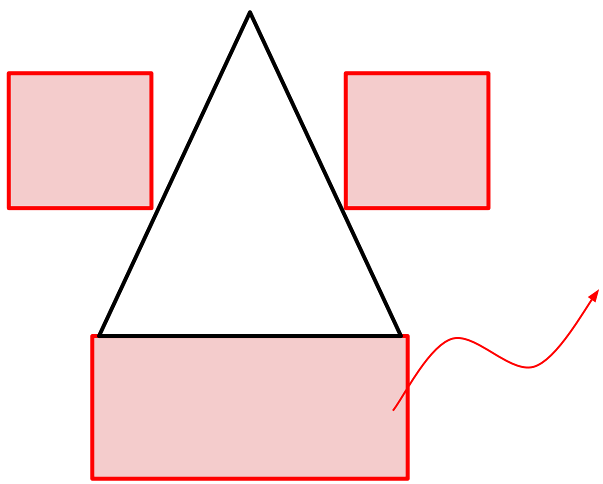
The Contractor

Inflate the point **P**

For this we have defined an **intersection** of inner inflators ...



And a **union** of inner inflators

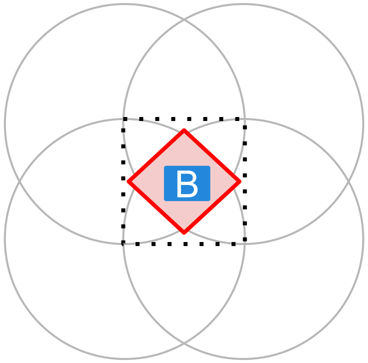


Choose the box with the greatest perimeter

The Contractor

Hence, it is possible to construct a box **B** inside the **compulsory part** of the reference object, i.e.

$$B \subseteq \left(\bigcap_{p_r \in [O_r]} \{p_r + S_r\} \right)$$



The Contractor: overlapping shape

- The overlapping shape it is the region where the reference shape will always collide with the moving shape.
- Can be calculated exactly with polytopes, but we are using curved objects.

We calculate something different

$$Overlapping_{r,m} = \bigcap_{x \in [\mathbf{X}]} (x \oplus S_r \oplus -S_m)$$

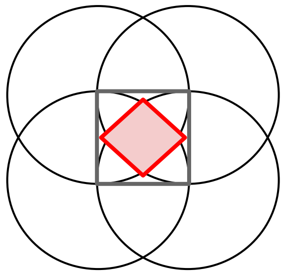
The Contractor: our forbidden box

The forbidden box that we obtain belongs to this expression

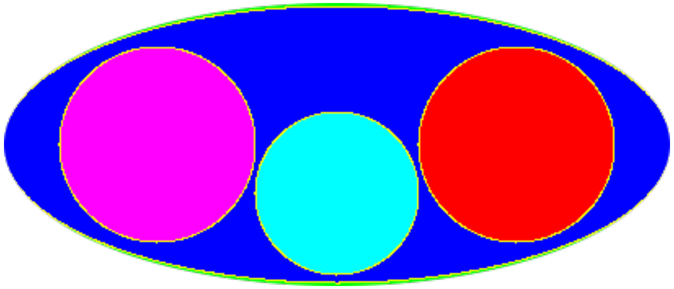
Is an inner box of the reference shape

$$B \subseteq \Big(\bigcap_{p_r \in [O_r]} \{p_r + [iO]_r\} \Big) - (p - o_m)$$

The intersection point



Experiments



We consider 2 of the objects as reference and 1 as moving

Reference/Moving	2^-M	2^-M+1	...
2^-R			
2^-R+1			
...			

Diameter of the box of the moving object

Diameter of the box of each reference object

The contractor that were used to the experiments:

- CtcShadowPropag (S)
- CtcPacking without CtcShadowPropag (P)
- Ctcpacking with CtcShadowPropag (S+P)

Experiments in progress

Summary

It is possible describe a great number of shapes with disjunction of conjunctions of non linear inequalities

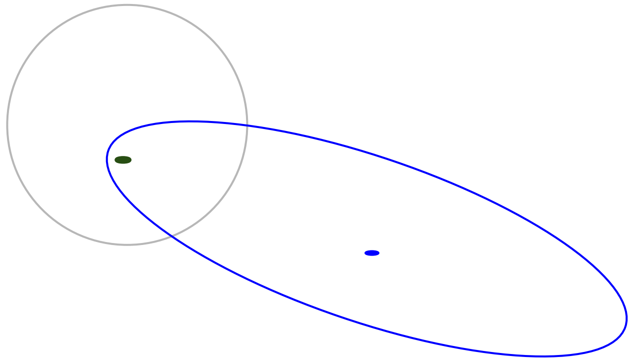
It is possible find the edges of the shadow shape, which give us a first economic contraction

To use an inflator that searches the boxes that belongs to the intersection of the objects, allows us to obtain an approximation of the optimal forbidden region

?

Satisfiability Check

The satisfiability check works similar to the point selector, but only evaluate if the point exists



To find this point, was used a branch & bound algorithm

Contract some box with the moving shape

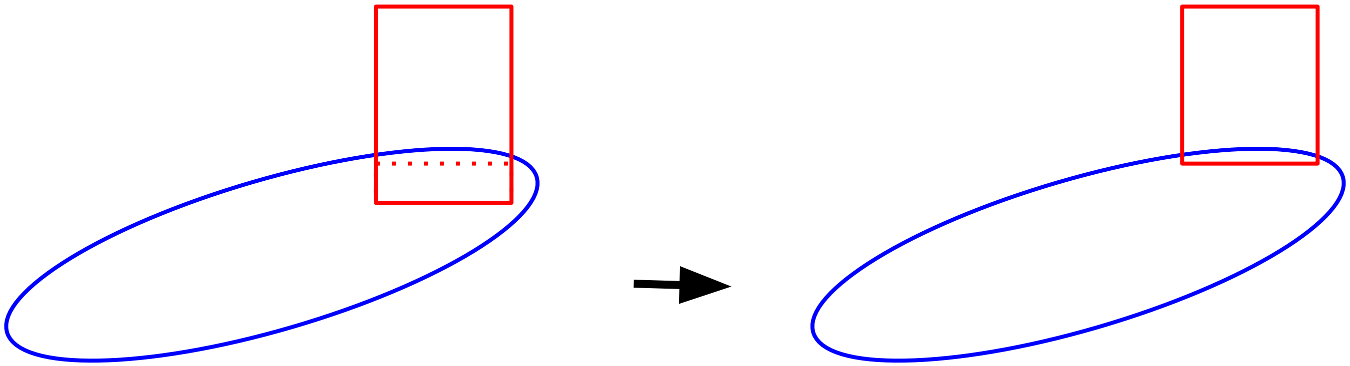
Contract the reference shape centered in the some point

Evaluate the intersection in a random point of the resulting box

Else, **Bisect**

Sweep

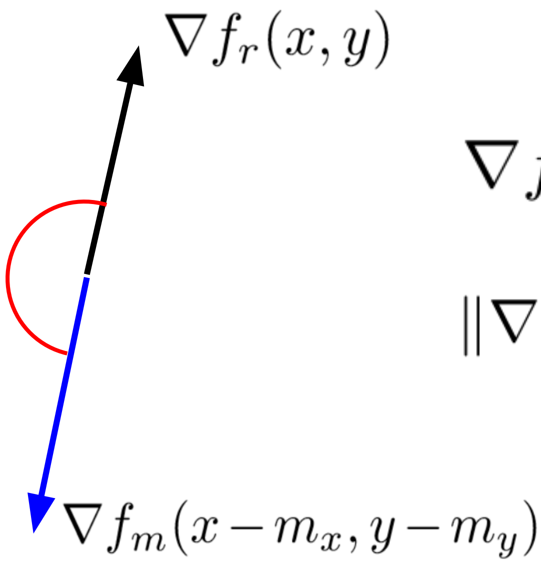
- The target of the sweep algorithm is to prune all the parts of some domain that violate some constraint
- To perform this, we must “saturate” one of the dimensions with forbidden boxes
- This forbidden boxes must consider some forbidden point and the working area



Shadow Calculation

$$\frac{\nabla f_r(x - o_r, y - o_r) \bullet \nabla f_m(x - o_m, y - o_m)}{\|\nabla f_r(x - o_r, y - o_r)\| * \|\nabla f_m(x - o_m, y - o_m)\|} = cos(\theta)$$

$$\frac{\nabla f_r(x - o_r, y - o_r) \bullet \nabla f_m(x - o_m, y - o_m)}{\|\nabla f_r(x - o_r, y - o_r)\| * \|\nabla f_m(x - o_m, y - o_m)\|} = -1$$



$$\nabla f_r(x - o_r, y - o_r) \bullet \nabla f_m(x - o_m, y - o_m) +$$

$$\|\nabla f_r(x - o_r, y - o_r)\| * \|\nabla f_m(x - o_m, y - o_m)\| = 0$$