On the Existence of a Strategy to Maintain a Moving Target within the Sensing Range of an Observer Reacting with Delay

Rafael Murrieta E.C.E **Alejandro Sarmiento** C.S. **Seth Hutchinson** E.C.E.

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
{ murrieta, asarmien, seth } @ uiuc.edu

Abstract—
This paper deals with the problem of computing the motions of a robot observer in order to maintain visibility of a moving target. The target moves unpredictably, and the distribution of obstacles in the workspace is known in advance. Our algorithm computes a motion strategy based on partitioning the configuration space and the workspace in non-critical regions separated by critical curves. In this work is determined the existence of a solution for a given polygon and delay.

I. Introduction

We consider the problem of tracking a target moving among obstacles. Specifically, a robot equipped with a visual sensor must keep visibility of a moving evader (the target). The observer and the target are in the same workspace which contains static obstacles. The obstacles generate motion constraints as well as visibility constraints. A key distinction with previously considered tracking problems, like missile control or pure visual tracking, is the presence of obstacles combined with the sensor ability to move. This paper focus on computing the motions of a robot observer in order to maintain visibility of a moving target. Low level motion control and visual data processing issues are not addressed in this paper.

The target-tracking problem has been traditionally addressed with a combination of vision and control techniques [3]. Purely control approaches, however, do not take into account the complexity of the environment. The basic question which has to be answered is where should the robot observer move in order to maintain visibility of a target moving in a cluttered workspace. Both visibility and motion obstructions have to be considered. Thus, a pure visual servoing technique will fail because it ignores the geometry of the workspace.

Maintaining visibility of targets is related to the *art-gallery problem* [11], where the goal is to compute the locations of a minimal number of guards such that all points in the workspace (the art gallery) are visible to at least one guard. In tracking, we are interested in guarding a moving point (the target) using a mobile guard (the observer).

A. Previous Work

Previous works have studied the motion planning problem for target tracking. Game theory [2] is proposed in [7] as a framework to formulate the tracking problem and an online algorithm is presented. This algorithm operates by maximizing the probability of future visibility of the target.

The approach presented in [9] computes a motion strategy by maximizing the *shortest distance to escape* – the shortest distance the target needs to move in order to escape the observer's visibility region. This planner has been integrated and tested in a robot system which includes perceptual and control capabilities.

In [5], a technique is proposed to track a target without the need of a global map. Instead, a range sensor is used to construct a local map of the environment, and a combinatorial algorithm then computes a differential motion for the observer at each iteration.

The problem of planning observer motions to maintain visibility of a moving target has received a good deal of attention in the motion planning community over the last few years. Several techniques have been reported in the literature, and a variety of strategies have been proposed to perform the tracking. However, complete algorithms [10] have been rarely proposed.

The main goal of this work is to proposed a complete motion strategy for the problem of maintaining visibility of a moving target.

II. Problem definition

The problem consists in **always maintaining visibility** of a mobile evader (**the target**) by using a mobile robot equipped with sensors (**the observer**). The target and the observer are represented as points. The environment where they are moving is modeled as a polygon. The visibility between the target and the observer is represented as a line segment and it is called **the bar**. This bar is emulating the visual sensor capabilities of the observer.

A. Assumptions

It is possible to think about several variants of this problem. There are two main factors to be considered: The speeds of the target and the observer, and the existence of delay between the target and the observer motions.

The simplest case consists in assuming that there is no delay between the target and the observer motions, and that the observer's speed is infinite. Even with these simplistic assumptions it is important to establish the existence of a solution for a given polygon and a given bar length; as this is used for solving the more complex cases.

In [10] we have proposed a complete and optimal algorithm for this case. In the present work, it is assumed that **there is a delay between the target's motions and the observer's**.

In a real robotic system this delay in the observer's reaction exists because of the execution time of perception and motion planning algorithms. The main implication of the existence of delay is that the observer computes where to move based on a target position that may have changed. Therefore, the exact position of the target is unknown.

The target moves continuously, its trajectory is unknown but its maximal speed is known and finite. On the other hand, the observer is able to move with infinite speed. Other than this, no kinematic nor dynamic constraints are imposed on the observer or the target motions.

III. Problem Modeling

The basic idea for solving the problem in a polygon consists **in partitioning the configuration space and the workspace in non-critical regions separated by critical curves**.

Delay is an important factor in the problem modeling. In order to deal with it the sensor must have range.

In this section, the relation between range and delay is established, the curves used to partition the work and configuration spaces are described and, some particular regions defined in the workspace are introduced.

A. Delay and Range

The delay in the reaction of the observer is due to the execution time of perception and motion planning algorithms. The observer does not have information about the target position continuously.

However, if the delay can be estimated or an upper bound established, and the maximal target speed is known, then the possible target positions can be defined by a disk. The radius of the disk is proportional to the delay. Maintaining target visibility under these conditions is equivalent to maintaining visibility of this disk.

The visibility between the target and the observer is represented by a bar. In order to maintain visibility of the disk, the bar must vary its length between a maximum and a minimum value. The bar is also able of rotating around the observer or around the target. Thus, the bar is emulating the sensor range and field of view. The target controls the bar's position (x, y) and the observer controls the bar's orientation θ and length.

$Smax$ is the maximal target speed and dt is the delay time, r is the maximal distance that the target can travel during the delay, thus $r = Smax \times dt$.

Let us call the maximal bar length $Lmax$ and the minimal bar length $Lmin$. The bar length at steady state is called $Lss = Lmin + 2 \times r$.

Let us established the minimal range mor able to keep target visibility for a given r.

Observation: The shortest path for the target to escape the observer range is over the line passing through the points representing the observer and the target.

Proposition III.1 *The minimal observer range* **mor** *to maintain target visibility for a given* r *is* $mor = 4 \times r$ *(see figure 1).*

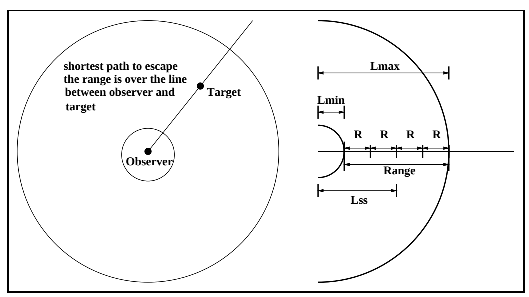

Fig. 1. Minimum range

Lemma III.2 *Target must be at* Lss *distance from the observer at the moment of the first sensing* t_0.

Proof To prove the lemma, consider that if the target at time t_0 is at $Lss \pm \epsilon$ distance from the observer then in the next sensing at time t_1 the target will appear, in the worse case, at $Lss \pm (r + \epsilon)$ form the observer. Therefore the target may actually be at distance $> Lmax$ or $< Lmin$ from the observer, and then outside of the range.

Proof To prove the proposition, consider that if the target at time t_0 is exactly at Lss distance from the observer then in the next sensing at time t_1 the target will appear, in the the worse case, at $Lss \pm r$ form the observer. Therefore the target may actually be at distance $Lmax$ or $Lmin$ from the observer, and then inside of the range. $mor = Lmax - Lmin = 4 \times r$. $\blacksquare$

The result above shows that the range must be greater or equal than $4 \times r$. For a given delay radius (target speed), the model that allows to solve the largest number of cases is the one where the range extends from the minimal observer range $Lmin$ to 4 times the delay radius (provided that this lees or equal than $Lmax$), as shown in figure 1. This model is used throughout this paper.

There are three interesting corollaries that will be used later to establish the existence of a solution.

Corollary III.3 *The observer motion that guarantees having the target within the range must be made, in the worst case, either at distances $Lmin + r$ or $Lmax - r$ from the previous target position.*

Corollary III.4 *The observer must move if the target appears in any sensing at distance different of Lss.*

Corollary III.5 *Any new sensing must be made at distance Lss between the current observer position and the previous target position.*

B. Partitioning the work and configuration spaces

The curves used to model the problem correspond to those used to establish the cell decomposition for ladder motion planning [13], [1]. The present approach uses 4 sets of such curves. Two of the sets use all the types of curves. One of these is calculated at distance Lss from the obstacles, the other at $Lim + r$.

The third set of curves are of type 2 only. These curves are defined at $2 \times r$ distance from the reflex vertices (those with interior angle larger than π). This type of curve is an arc of circle centered in every reflex vertex and limited by the segments touching the vertex.

The fourth set of curves is composed of straight lines emerging from every reflex vertex and having the direction of the segments associated to the vertex. These lines correspond to the inflection rays in an aspect graph based on perspective projection [12]. These lines also correspond to curve type 3, if the bar had infinite length.

The five types of curves used to establish cell decomposition for ladder motion planning are in some way capturing the notion of visibility, note that if the curves were defined in a polygon of finite size by a bar of infinite length then the resulting curves would correspond exactly to those of an aspect graph based on perspective projection.

B.1 Cell decomposition for ladder motion planning

The curves defined in [13], [1] are the set of points where the structure of the C-obstacle region above the xy-plane undergoes a qualitative change. Indeed, when such a curve is crossed, either the C-obstacles's faces which are intersected by a line perpendicular to the xy-plane at the current position changes, or the number of intersection points changes [8].

Two sets of curves are defined for determining necessary conditions for the existence of a solution (see section IV).

Thanks to these curves it is possible to divide and represent the configuration space with a connectivity graph G. G is a non-directed graph whose nodes are all the C-space cells. There is an edge connecting any two nodes only if the corresponding cells are adjacent (see below).

The cells are non-critical regions. A region is a maximal subset of admissible positions of the bar which intersect no critical curve.

Roughly speaking, the definition of a non-critical region is based on stops. Consider a non-critical region R and define $F(x, y) = \{\theta/(x, y, \theta) \in C_{free}\}$.

If all orientations of the bar are collision free at (x, y) then $F(x, y) = [0, 2\pi)$, else $F(x, y)$ contains a finite number of open, maximal intervals. The center of the bar rotation (point P) corresponds in our problem to the target position.

A stop is the vertex or edge that the bar contacts at an endpoint of an interval in position (x, y). Thus, each interval has a clockwise and counterclockwise stop associated to it. $\sigma(x, y)$ denotes the pair of stops associated with $F(x, y)$. If $F(x, y) = [0, 2\pi)$, then we write $\sigma(\Omega, \Omega)$, where Ω designates no stop. For any two points (x, y) and (x', y') both in the same non-critical region R: $\sigma(x, y) = \sigma(x', y')$. $\lambda(x, y, s) = \theta$ denotes the orientation θ at which the bar hits stop s.

Cells are define as follows: $Cell(R, S_1, S_2) = \{(x, y, \theta)/(x, y) \in R \text{ and } \theta \in (\lambda(x, y, s_1), \lambda(x, y, S_2)\}$. $Cell(R, \Omega, \Omega)$ denotes a cell with no stops. $Cell(R, \emptyset, \emptyset)$ denotes a cell where the bar is always in collision. Note, that this situation can occur for a given bar length and polygon.

The adjacent condition is:
Two cells $k = cell(R, S_1, S_2)$ and $k' = cell(R', S_1', S_2')$ are adjacent if and only if:
- The boundaries of R and R' share a section of the critical curve β
- $\forall (x, y) \in int(\beta)$
$(\lambda(x, y, S_1), \lambda(x, y, S_2)) \cap (\lambda(x, y, S_1^{'}), \lambda(x, y, S'_2)) \neq \emptyset$

If two cells, k and k' are adjacent, any configuration in k can be connected to any configuration in k' by a free path whose projection onto xy-plane crosses β transversely, with constant orientation in some neighborhood of the crossing point. From here on, regions denote sets of points in the xy-plane and cells denote sets of configurations in the configuration space.

B.2 Target escape region

Target escape regions $ter's$ are regions in the neighborhood of a reflex vertex. Every reflex vertex has an associated ter. A ter is the region contained inside curves type 2. The curve is established at distance $2 \times r$ thus, the observer can always detect the target at least r distance from the vertex, even in the worst case. Because of the delay, when the target is detected at distance r from the reflex vertex (after processing) the target can actually be on the vertex at the present time, exactly before it escapes behind the corner. This gives the observer the opportunity to react and move to a cgr (see below). It is assumed that the observer can move with infinite speed. When the target is inside one of these regions, the observer must be in a cgr and must see all the ter through which the target can escape.

B.3 Corner guard regions

Corner guard regions $cgr's$ are used to prevent the target from escaping behind a reflex vertex. This is equivalent to determining if the corresponding target escape region ter is visible by a bar having length Lss.

Every reflex vertex has an associated cgr. The cgr is the region contained inside the two inflection rays of every vertex. These lines would also correspond to curves type 3, if the bar had infinite length.

If the target is inside a ter region then the observer must move to the cgr from where it is possible to see the complete ter region through which the target can escape.

Critical curves and non-critical regions for ladder motion planning are used to reach a cgr with an appropriate bar configuration (see traveling condition IV-A and motion strategy V).

IV. Conditions for solving the target tracking problem

There are four necessary conditions to establish the existence of a solution and therefore the completeness of the strategy. This section explains the first three, the fourth will left for the motion strategy section due to their relationship.

The firsts three conditions are: a) The observer must travel at least at distance $Lmin + r$ between two sensings (traveling condition), b) the observer must always sense the current target position being at Lss distance from the previous target position (steady state condition), and c) for each reflex vertex there must exist a cgr from which ter can be completely seen (visibility condition).

A major difference between the problem without delay and with delay is that, with delay the observer must be stationary some time after sensing. The observer cannot move immediately after sensing because it needs time to process this information and determine where the target was and decide where the observer must move. Since it is assumed that the observer motion takes zero time, sensing while moving is pointless. Because of the delay, the observer may need to guard one o more ter's at a time. Therefore a strategy able to guard all the ter's through which the target can escape is required (see section V).

A. Traveling condition

Satisfying this condition is equivalent to determining the existence of a solution for the case where the observer speed is infinite and, there is no delay between the target and the observer motions [10].

In a polygon (closed curve) this condition will always depend in the initial bar configuration. If the bar starts in an unappropriated configuration (escapable cell, see below) the target can break the bar. Otherwise, this condition is satisfied if the target can never bring the bar to an unappropriated configuration.

The cell decomposition for ladder motion planning is used for determining whether or not this condition is satisfied.

This set of curves is defined at $Lmin + r$ distance from the obstacles. The cells in the configuration space defined by these curves are used to determine if the observer can change the bar configuration (by rotating around the target) at the minimal distance that guarantees the target being inside the range (see III.3). Cells in the configuration space where the target can break the bar must be eliminated. We call those cells escapable cells.

Definition For cell $K \subset \Re^2 XSO(2)$ above region $R \subset \Re^2$, if $\exists$ R' adjacent to R such that there is not a K' adjacent to K projecting onto R' then cell K is an escapable cell.

All the escapable cells must be eliminated. This rule must be recursively applied to all the cells on the configuration space until either no cell K is eliminated (the condition is satisfied) or all the cells K_i corresponding to a single region R are eliminated (the condition is not satisfied).

Proposition IV.1 *If $\exists$ R such that all its corresponding cells K_i are escapable, then the target can get outside the observer range.*

Proof If $\exists$ R such that all its corresponding cells K_i are escapable cells then by definition there exists at least one R' adjacent to R such that it does not have any cell K' adjacent to any cell K_i. Therefore the target can move between regions R and R' but the observer will not be able to bring the bar from a configuration in K_i to an adjacent configuration in K'. ∎

The following examples illustrate the previous condition. In all the examples, the edges are denoted by E_i and the vertices by V_i.

Figure 2 shows an example of a very narrow and long non-convex corner. There are 15 regions in the xy-plane and 22 cells in the configuration space. The graph representing the environment does not contain a single connected component. This means that the bar cannot completely rotate between region 4 and region 10.

The rule used to detect non escapable cells is applied to all the cells until no more escapable cells are detected. Red rectangles indicate the escapable cells. One of the components contains only escapable cells. The other component contains all the region in the xy-plane. Therefore, if the bar does not start in a non escapable cell a solution exist for the case of no delay. This also means that the traveling condition is satisfied for the case of delay.

Figure 3 shows a polygon (a rectangle). The rectangle has two parallel segments smaller that 2 times the bar length. There are 18 regions in the xy-plane and 32 cells in the configuration space. The rule used to detect non escapable cells is recursively applied to all the cells until all the cells corresponding to a single region are eliminated. Red rectangles indicate the escapable cells. The graph in the figure only contains the cells after elimination of escapable cells. The region 8 is not in the graph. If the target is in region 8, it can leave the region toward an adjacent region (i.e region 9) that does not have a cell adjacent to the bar configuration in region 8. Therefore, a solution does not exist with or without delay.

For more details on this condition see [10].

B. Steady state condition

In this case, the set of curves for the ladder motion planning are defined at distance Lss. These other curves are used to determine if the observer can sense the current target position being at Lss distance from the previous target position. This is a condition to ensure the target always being inside the observer range (see III.5).

Proposition IV.2 *If $\exists K = Cell(R, \emptyset, \emptyset)$ in the polygon then the observer cannot maintain the target within the sensor range and therefore this condition is not satisfied.*

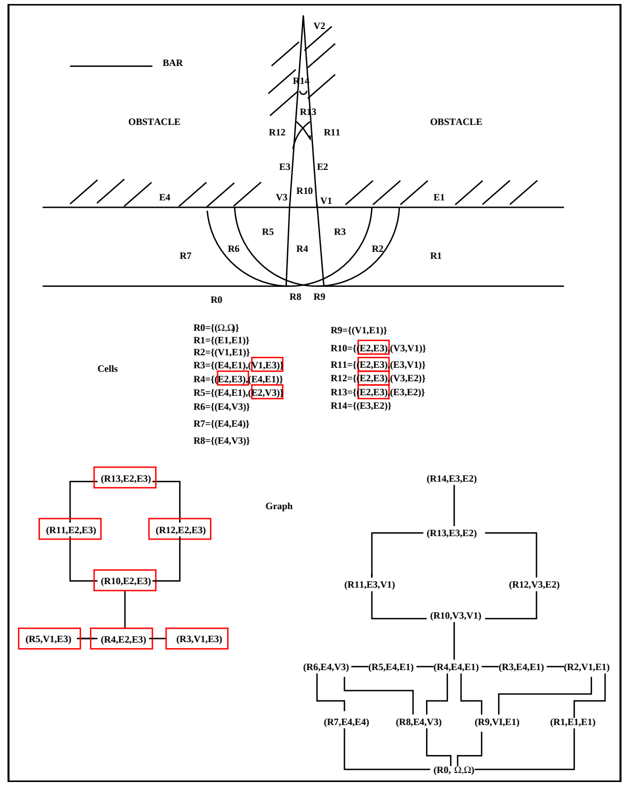

Fig. 2. No convex corner

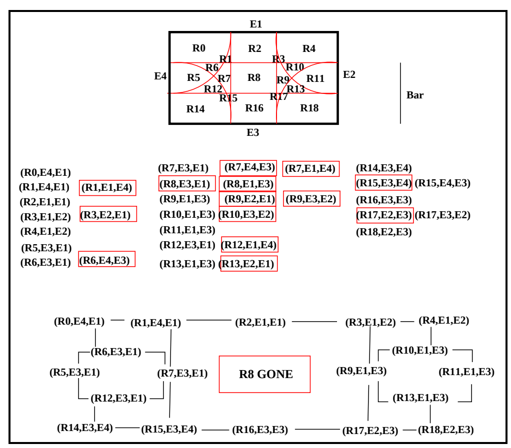

Fig. 3. Rectangle

Proof If the target at time t_{n-1} is inside a region R such that $\exists K = Cell(R, \emptyset, \emptyset)$ then the observer cannot be at distance Lss from the target at time t_n. Therefore by III.5 the target at time t_{n+1} could be outside of the range. ∎

C. Visibility condition

The visibility condition consists in verifying if all the *ter* regions in a polygon can be guarded by the observer, for a given length of the bar.

There can only be two reasons why the observer cannot see a single *ter* region being in a *cgr*: 1) a portion of *ter* is occluded, b) the end point of a bar of Lss length cannot sweep all *ter* being inside *cgr*. these reasons are called non-shadow and coverage conditions. The visibility condition is satisfied when the non-shadow and coverage conditions are satisfied.

Of course, the non-shadow and coverage conditions must only be verified in polygons with reflex vertices.

Besides, if *ter* regions intersect, then the observer must move to the intersection of the *cgr* regions associated with the *ter* regions. In section V the motion strategy to reach these *cgr*'s will be discussed in more detail.

C.1 Non-shadow condition

The non-shadow condition can be characterized by verifying whether or not some reflex vertex is inside a *ter*. If a reflex vertex is inside a *ter* then there could be a shadow.

If this is true then the only region where the whole *ter* can be seen, is the region corresponding to the intersection of the *cgr*'s associated to the vertices. If this intersection exists then the whole *ter* can be guarded by being in a single *cgr* and then the condition is satisfied. If the intersection of all the *ter*'s exists then the boundary is a convex polygon. For simplicity when this intersection exists it is also called *cgr*.

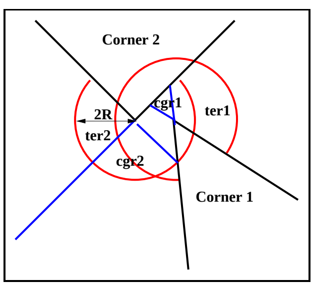

Fig. 4. Non-shadow condition

In the example shown in figure 4 the non-shadow condition is not satisfied.

C.2 Coverage condition

For determining if the observer can guard a *ter* region being in an *cgr* region, it is necessary to know if one end point of the bar of length Lss can sweep all the *ter* region while the observer is inside *cgr*.

The shape of the region swept by the bar end point must be computed. This shape corresponds to all the possible configurations of the bar lying inside and on the border of a specific *cgr* (the *cgr* is treated as a closed set).

We call $A(cgr|Lss)$ the region swept by the end point of a bar of length Lss being inside a *cgr*.

A *cgr* could be the intersection of several single *cgr* regions. In this case the intersection of the *cgr* regions must cover the intersection of the *ter* regions associated to every *cgr*.

Without any lost of generality the intersection of several *cgr* regions and the intersection of several *ter* regions are treated as single *ter* and *cgr* for computing $A(cgr|Lss)$.

C.3 Computing $A(cgr|lss)$

The computation of $A(cgr|Lss)$ is done using the boundary of *cgr*.

$A(b|Lss)$ is the area swept by one end point of the bar of length lss that is rotating around while the other end point is moved along the boundary of *cgr*. The boundary of *cgr* is composed of line segments. Every segment is processed independently. The area swept by the bar along a segment is called $A(s|Lss)$.

The computation of $A(s|Lss)$ is done by drawing a circle centered at each one of the 2 end points of the segment with radius Lss (curve type 2) and 2 parallel lines at Lss distance from the segment (curve type 1). The lines are drawn on both sides of the segment and are called *plr* and *pll*. The regions inside curve type 2 are called *ct* and *cb* $A(s|Lss)$ is equal to the union of the polygon defined by the end points of *plr* and *pll* with regions *ct* and *cb* minus the intersection of *ct* and *cb* (see figure 5). $A(b|Lss) = \bigcup A(s|Lss)\forall_{s \in b}$.

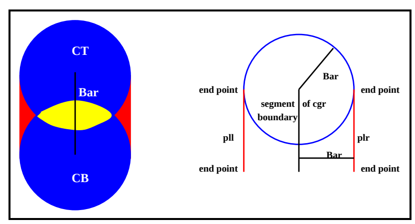

Fig. 5. Computing A(s/Lss)

If $cgr - A(b|Lss) = \emptyset$ then $A(cgr|Lss) = A(b|Lss)$ (figure 6 b), else there is a hole inside $A(b|Lss)$ and there are two cases. When all the curves type 2 related to the vertices of the *cgr* intersect, the boundary of the hole is composed by arcs of circle. This hole region is the intersection of all the curves type 2. In this case $A(cgr|Lss) = A(b|Lss)$ as well (figure 6 c). However, if this hole is composed by straight line segments then $A(cgr|Lss) = A(b|Lss) \bigcup cgr$ thus this hole will disappear (figure 6 a). This method of computing $A(cgr|Lss)$ only works for convex polygons and every *cgr* is a convex polygon. The coverage condition is satisfied if $ter \subseteq A(cgr|Lss)$. This condition

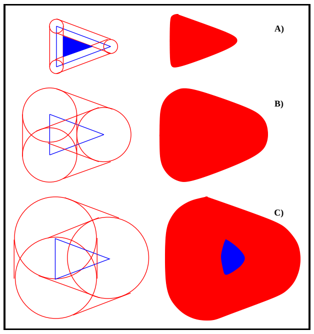

Fig. 6. Computing $A(cgr|Lss)$

is equivalent to the covering set concept used in topology, for more details see [6].

V. The Motion Strategy

The motion strategy is as follows. The observer is required not to start target pursuit in an escapable cell. The observer must move if the target appears in any sensing at distance different from Lss. In this case, it must move to a position at distance Lss form the current target sensing (previous target position). If the target is inside a ter region the observer must move to the corresponding cgr.

At all times the observer must move to a position that respects the sensor range. In the free space, there may be many positions that satisfy such a constraint. Given that the target motion is unpredictable and that we assume that moving the shortest distance is optimal, the best strategy is the one that moves the observer as little as possible. This implies a movement in the direction of the bar (toward or away from the target) because the minimum distance between two points is a straight line segment. We call this combination of bar rotation around the observer and translation to maintain Lss, the reactive motion rm.

If a reactive motion rm would cause the bar to collide, the observer can rotate the minimum angle around the target that makes the bar be in a collision free configuration. In those cases the bar will show a compliance motion (keeping the bar in contact with the obstacles) [4]. The above strategy is optimal in the minimum distance traveled.

If a reactive motion rm would cause the bar to be in an escapable cell the observer must rotate around the target to keep the bar configuration in a non escapable cell.

The observer must never cause the bar configuration to be in a escapable cell. This can be accomplished by moving the observer at the minimum distance possible $Lmin + r$ from the target and going to a position at Lss distance from the target.

If there is not delay between the target and observer motions, the observer must avoid that the target brings the bar to an escapable cell exactly at the moment when target is crossing a critical curve [10].

If delay exists, the observer does not have information about the target position continuously. In this case, critical curves defined at distance Lss and $Lim+r$ from the obstacles are used to determine when the observer must avoid that the target brings the bar to an escapable configuration. The region between these two families of critical curves are used as triggers to start observer motions. The regions between the critical curves can be seen as a "thick curve" where it is certain that, even with delay, the target will be sensed. If the target is inside this region ("thick curve"), it indicates the observer to start the rotation before it is too late.

Figure 7 shows an environment with two reflex vertices (black lines). The boundaries of ter's are the arcs of circle in blue, dashed lines show the cgr regions, red lines indicate the critical curves at $Lmin + r$ distance and green lines indicates the critical curves at Lss distance. The red dot indicates the target and the blue dots the observer. A bar of $Lmin + r$ length is indicated in yellow, this shows the observer in motion – not in steady state. If the target is inside the trigger between region $R1$ and $R3$ and the bar is about to be in configuration $(R3, E2, E1)$ then the observer must rotate around the target to bring the bar to configuration $(R3, E1, E2)$. In this environment the traveling condition is satisfied, since the graph after elimination of escapable cells, contains all the region in the xy-plane.

A. No determinable motion for a single pursuer

Because of the delay, there are situations where the observer is not able to determine a motion that guarantees to have the target within range, this is the fourth condition for the existence of a motion strategy.

If two or more ter's intersect and the intersection between cgr's regions does not exist, the observer is not able to make a decision that ensures target visibility. The observer can move only to one of the cgr's and the target can move to the ter region that is guarded by the other cgr (see figure 8). Note that in this case the non-shadow condition is satisfied for both single cgr's. If regions inside the two families of curves (triggers) corresponding to two or more different escapable cells intersect, it means that the target can bring the bar to two or more different escapable cells. If there is not a bar configuration that can keep the bar outside all escapable cells then there is not a solution. The observer can only choose one of them and the target can bring the bar to other.

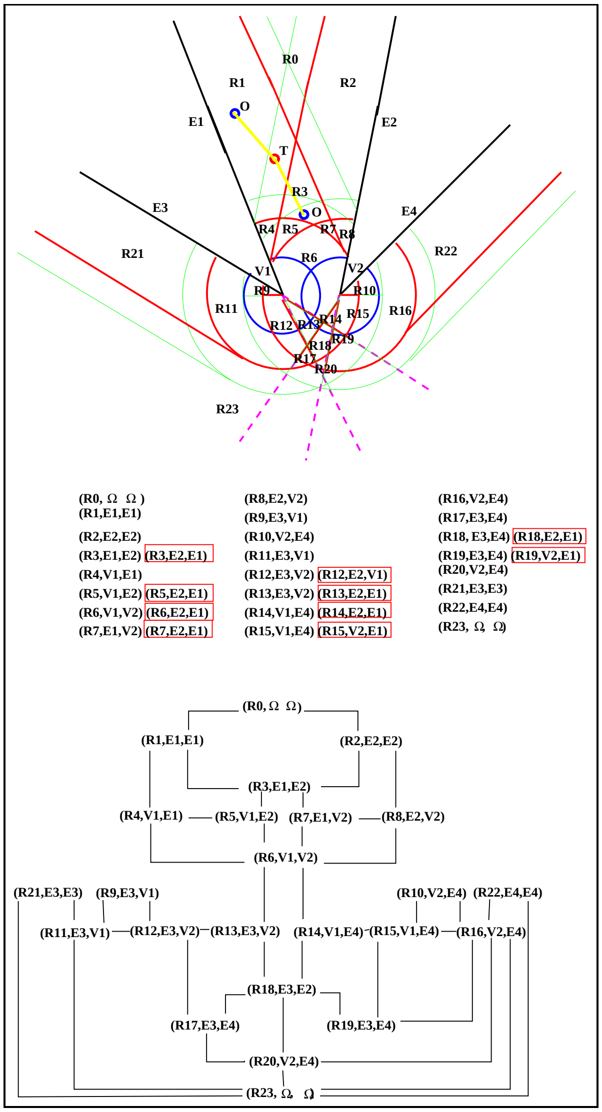

Fig. 7. Two convex corners

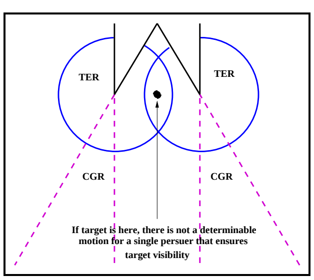

Fig. 8. No determinable observer motion

In these situations more than a single observer is required to guarantee target visibility. If this condition occurs there does not exist a motion strategy that ensures target visibility.

VI. Conclusions

This work proposes an approach to solve the target tracking problem. The target is assumed to move unpredictably and the distribution of obstacles in the workspace is known in advance.

The approach consists **in partitioning the configuration space and the workspace in non-critical regions separated by critical curves**.

The method can determine the existence a solution for this problem. If a solution does exist, a motion strategy that maintains target visibility is proposed.

We conclude that if all the conditions are satisfied and the observer moves (at time t_n) to a location at Lss distance from the previous target position (at time t_{n-1}), never getting closer than $Lmin + r$ from this position, then it is impossible for the target to get outside the observer range.

In this work, it is assumed that the observer speed is infinite. This assumption was done to simplify the analysis, and to better understand the problem. Future work will consist in proposing a solution where the observer's speed is bounded.

References

[1] J. Bañon, Implementation and Extension of the Ladder Algorithm *In Proc IEEE Int. Conf. on Robotics and Automation*, 1990.

[2] T. Başar and G. Olsder, Dynamic Noncooperative Game Theory. *Academic Press. 1982.*

[3] B. Espiau, F. Chaumette, and P. Rives, A new approach to visual servoring in robotics. *IEEE Trans. Robot and Autom.*, 8(3):313-326, June 1992.

[4] A. Fox and S. Hutchinson, Exploiting Visual Constraints in the Synthesis of Uncertainty-Tolerant Motion Plans, *IEEE Trans. Robotics and Automation*, 11(1):56-71, February 1995.

[5] H.H. González, C.-Y. Lee and J.-C. Latombe, Real-Time Combinatorial Tracking of a Target Moving Unpredictably Among Obstacles, *In Proc IEEE Int. Conf. on Robotics and Automation*, 2002.

[6] L. C. Kinsey, Topology of Surfaces. *Springer. 1993.*

[7] S.M. LaValle, H.H. González-Baños, C. Becker and J.-C. Latombe, Motion Strategies for Maintaining Visibility of a Moving Target *In Proc IEEE Int. Conf. on Robotics and Automation*, 1997.

[8] J.-C. Latombe, Robot Motion Planning. Kluwer Academic Publishers, 1991.

[9] R. Murrieta-Cid, H.H. González-Baños and B. Tovar, A Reactive Motion Planner to Maintain Visibility of Unpredictable Targets *In Proc IEEE Int. Conf. on Robotics and Automation*, 2002.

[10] R. Murrieta-Cid, A. Sarmiento and S. Hutchinson, A Motion Planning Strategy to Maintain Visibility of a Moving Target at a Fixed Distance in a Polygon *In Proc IEEE Int. Conf. on Advanced Robotics*, 2003.

[11] J. O'Rourke, Visibility. *In Handbook of Discrete and Computational Geometry, 467-479.* J.E. Goodman and J. O'Rourke Ed. 1997.

[12] S. Petitjean, D. Kriegman and J. Ponce, Computing exact aspect graphs of curved objects: algebraic surfaces. *Int J. Comput Vis., 9:231-255*, Dec 1992.

[13] J.T. Schwartz and M. Sharir, On the Piano Movers' Problem: I. The Case if a Two-Dimensional Rigid Polygon Body Moving Amidst Polygonal Barriers, *Communications on Applied Mathematics*, 36, 345-398, 1987.