Motion Planning and Visibility Problems using the Polar Diagram

C.I. Grima and A. Márquez[†] L. Ortega[‡]

Abstract

Motion planning and visibility problems are some of the most important topics studied in Computer Graphics, Computational Geometry and Robotics. There exits several and important results to these problems. We propose a new approach in this paper using a preprocessing in the plane, the polar diagram. The polar diagram can be considered as a plane tessellation with similar characteristics to the Voronoi Diagram. The Euclidean distance criterion is changed by the minimal angle criterion in this new approach. The advantage of using polar diagrams is an optimal computing preprocessing time and their immediate applications to angle problems as visibility or motion planning problems.

1. Introduction

The solution to many important problems in Computer Graphics requires angles processing of the data input. In Figure 1 a simple visibility problem is presented. The maximum visibility angle from any point *x*, can be easily found by the computation of angular scanning towards the objects *A* and *B*.

Nevertheless, the angular scanning performed in this example is not a practical method when the calculation is repetitive. Every angular sweep needs a linear processing time.

† Dpto. Matemática Aplicada I, University of Seville, Spain
‡ Dpto. Informática, University of Jaén, Spain

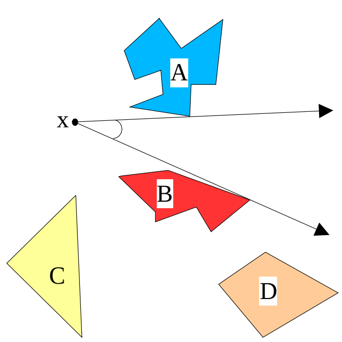

Figure 1: *Vision angle from the pointx.*

The locus approach we propose, the polar diagram, constructs a tessellation in the plane that used as preprocessing, avoid making exhaustive searches to find sites with minimum angle characteristics. The information the polar diagram maintain is intrinsic to its data structure in a similar way to the proximity problem resolution using the Voronoi diagram [16].

But the aim of this paper is not only the polar diagram definition or the study of optimal algorithms construction. This tessellation have important and interesting angle properties. Every polar region is the locus of the points with similar visibility angle with respect to any other site. The problem presented above can be solved efficiently, not only for a determined point *x* but for any point in the plane.

This characteristic allows us to face up to the path planning problem as well. A possible path free of obstacles can be found solving local visibility problems and joining the results.

This paper present the polar diagram of a set of points in the plane in Section 2. Its extension to any other geometric object is studied in Section 3. In Section 4 we study visibility problems and a new method for the path planning computation.

2. Polar diagram Definition

In this section we define the polar diagram of a set of points in the plane assuming that this definition can be generalized to any other geometric object.

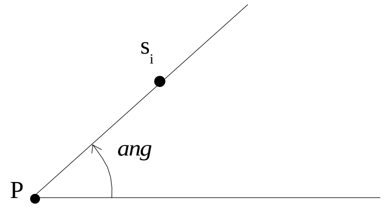

Figure 2: *Polar angle.*

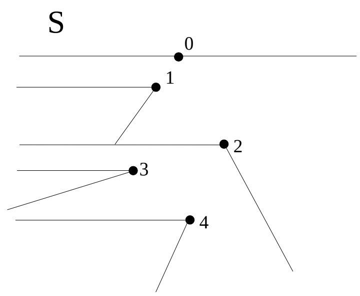

Figure 3: *Polar diagram of S.*

We introduce the polar diagram as a plane partition with similar features to the Voronoi diagram. Let us define all polar diagram elements.

The polar angle of the point p with respect to s_i, denoted as $ang_{s_i}(p)$, is the angle formed by the positive horizontal line of p and the straight line joining p and s_i, as it is shown in Figure 2. As the polar angle must be lower than π, p has less y-coordinate than s_i.

Given a set S of n points in the plane, the loci of points with least positive polar angle with respect to $s_i \in S$ is called *polar region* of s_i, denoted as $\mathcal{P}_S(s_i)$. Thus, $\mathcal{P}_S(s_i) = \left\{(x,y) \in E^2 \mid ang_{s_i}(x,y) < ang_{s_j}(x,y), \forall j \neq i\right\}$. The plane is divided in different regions in such a way that if the point $(x,y) \in E^2$ lies in $\mathcal{P}_S(s_i)$, it is known that s_i is the first site found performing an angular scanning starting from (x,y). We can draw an analogy between this angular sweep and the behavior of a radar [9, 5].

All $s_i \in S$ constructs a polar region and these n regions divide the plane defining a tessellation we have named *polar diagram* of S, denoted as $\mathcal{P}(S)$. Lines and half-lines constructing these polar regions are called *polar edges*.

To summarize, we construct a plane tessellation with the polar angle criterion. Actually, the polar diagram constructs a partition of the lowest semi-plane. The boundary is the straight horizontal line crossing the highest site of S. In Figure 3 is depicted the polar diagram of a set of points in the plane and the final division constructed using the least polar angle criterion.

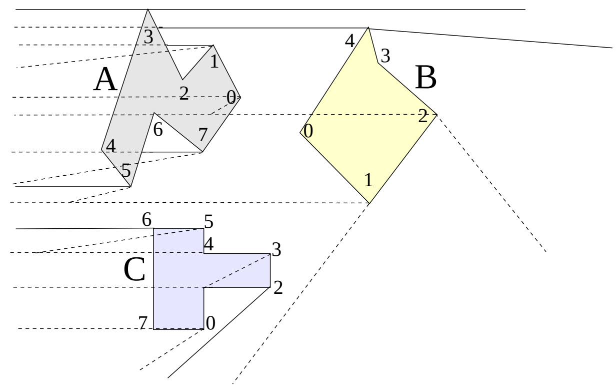

Figure 4: *Example of polygons polar diagram.*

3. Polar diagram of geometric objects

In [5, 9] we give optimal algorithms for the polar diagram construction using the sweep line and the divide and conquer methods. Otherwise, there is no justification for a plane preprocessing. The fastest method, the plane sweep, constructs the polar region of s_i, ones the $i-1$ points with greater y-coordinate have been processed.

As it has been mentioned before, plane partitions have found lots of applications fields. However in many real problems, tessellation generators can not be considered elementary points. Thus, reality is represented using geometric objects as segments, polygons and circles. Some of these problems are proximity problems, path planning and visibility or illumination problems. Classical examples of tessellation in the plane are the Voronoi diagram or the trapezoidal maps. Polar diagram of geometric objects is in fact, a new partition of the plane with similar characteristics to the polar diagram of a set of points [7, 8], and its definition is really similar to the given for points in the plane. Let O be a set of geometric objects in the plane, the polar region associated to o_i, $\mathcal{P}_O(o_i)$ is the locus of points with least polar angle with respect to o_i than with respect to any other object of O, in a positive angular scanning starting from the zero angle.

There is an important property associated to the polar diagram of polygons and segments: it is contained into the polar diagram of the set of points made up of their vertices or endpoints.

An optimal method for the polar diagram construction of a set of geometric objects can follow a sweep incremental algorithm for a set of points, but adding restrictions in order to discard some polar edges or portions of them. Following the following rules allows us to eliminate certain edges: (1) if there is any obstacle to the right of an endpoint, (2) if it splits two sectors of the same polar region, (3) if an edge portion lies inside another polar region and (4) if a polar edge lies inside the object it belongs. We show an example of poly-

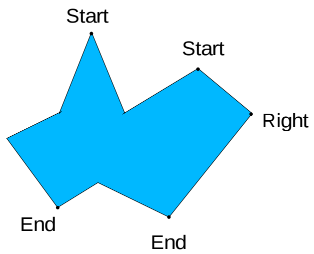

Figure 5: *Processed vertices.*

gons polar diagram in Figure 4 drawing the discarded edges with striped lines, the rest of edges remain.

Algorithm 1 describes the technique for polar diagram construction of a set of polygons in the plane. The algorithm extension for line segments is obvious, however the polar diagram for a set of circles needs some other comments to reach to similar conclusions as we see in [14].

Nevertheless, the polar diagram computation of polygons can be improved again because not all vertices belong to a polar edge. There are not edges associated to reflex vertices and neither to those in the left side of a polygon. Only the following vertices, illustrated in Figure 5, are taken into account:

Start: Vertex v_i is a Start vertex if v_{i-1} and v_{i+1} have less y-coordinate.

End: Vertex v_i is an End vertex if v_{i-1} and v_{i+1} have greater y-coordinate.

Right: Vertex v_i is a Right vertex if v_{i-1} has less y-coordinate and v_{i+1} has greater y-coordinate than v_i.

Algorithm 1 needs an $O(n \log n)$ time to sort all vertices. The incremental approach always work in a similar way: every vertex point is reached from top to bottom and processed according to some conditions. There are n vertices to process, and those preliminary polar regions can be constructed in linear time. For every vertex v_i it is necessary an additional $O(\log n)$ time to find neighbors to left and right in order to discard those edges mentioned before, however the optimal $O(n \log n)$ time is not modified.

To sum up, the polar diagram of polygons can be computed in a $\Theta(n \log n)$ time, being an optimal preprocessing in the plane for visibility and motion planning problems.

4. Visibility problems and Motion Planning

The polar diagram can be considered a new geometric approach to solve angle problems. This new tessellation applications are the convex hull of a set of points and objects in the plane [5], visibility problems and its generalization to the path planning problem. The polar diagram advantages are their robust construction methods in optimal computation time.

Algorithm 1: *Incremental*

Input: A set P of N polygons in E^2
Output: $\mathcal{P}(S)$
BEGIN

1. Sort vertices of P by decreasing order, obtaining $V = \{v_0, v_1, \ldots v_{n-1}\}$
2. Push(stack, 0)
3. Compute v_0 polar edges
4. FOR $i = 1$ to $n-1$ DO
   a. Be such p that $v_i \in p$
   b. WHILE $v_{top(stack)}$ oblique edge intersects with the horizontal of v_i
      i. Pop(stack)
   c. Let p_R and p_L be the nearest polygons to right and left of v_i
   d. IF v_i is a convex and *begin* vertex THEN
      i. IF $\nexists$ another *begin* vertex $v_j \in p$ to the right of v_i
      ii. THEN compute an horizontal edge from v_i reaching to p_L if it exists or to infinite otherwise
      iii. IF $\nexists$ p_R
      iv. THEN compute an oblique edge from v_i with gradient $\overline{v_{top(stack)} v_i}$ if does not cross p
   e. IF v_i is a convex and *right* vertex THEN
      i. IF $\nexists p_R$ and $v_{top(stack)} \notin p$
      ii. THEN compute an oblique edge from v_i with gradient $\overline{v_{top(stack)} v_i}$ if does not cross p
   f. IF v_i is convex and *end* vertex THEN
      i. IF $v_{top(stack)} \notin p$ THEN
         A. IF $\exists p_R$
         B. THEN compute an horizontal edge from v_i reaching p_L if it exists or to infinite otherwise
         C. ELSE compute an oblique edge from v_i with gradient $\overline{v_{top(stack)} v_i}$
5. END_FOR
6. Push(stack,i)

END

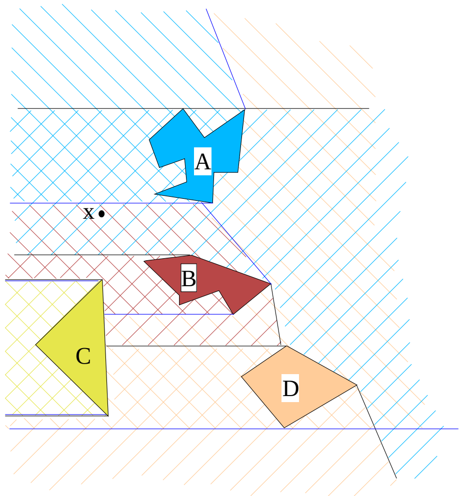

Figure 6: *Point x lies into the polar regions of A and B.*

4.1. Visibility problems

Visibility problems are one of the most important topics in Computational Geometry with important repercussions in Computer Graphics. Some of these classical problems are the Art Gallery [1], Illumination [12] and even the Path Planning problem [13].

We define the problem shown in Figure 1 as the *maximum visibility angle problem in an orthogonal direction*. It can be considered one of the simplest visibility problems, the maximum visibility angle from a point x towards any orthogonal direction, East, North, West or South. This problem can be easily solved in linear time computing an angular scanning with positive and negative criteria. But again, this exhaustive search can be avoided using polar diagrams.

After the polar diagram definition, it is straightforward to understand that this visibility technique can be improved using this tessellation as preprocessing. Object A is known to be the first obstacle found in a positive angular sweep starting from point x. This information is given by the polar diagram in a logarithmic time, the time we need to locate this point into a polar region.

However, the least positive polar angle criterion is not suitable to find object B. In fact, it is necessary a negative angular scanning instead of a positive one. But again, polar diagrams can be useful in this search, we only need to change the polar diagram criterion of construction to find a different tessellation with similar characteristics. In Figure 6 it has been superimposed these two East polar diagrams. Point x belongs to different polar regions depending on the used criterion. When this circumstance happens, as we observe in Figure 7 with the point x, it always means that the visibility angle is null. Point p is in both cases in B polar regions, so we know this object hinders in a trajectory towards the East direction, in fact, this is the only object we should avoid in a East trajectory.

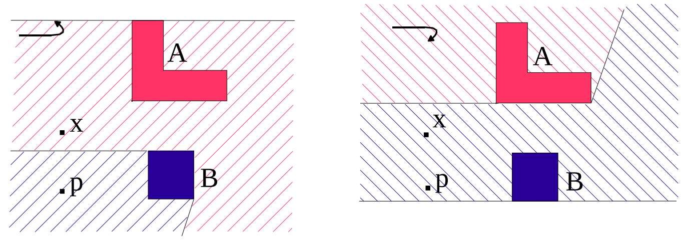

Figure 7: *Polar diagrams in the zero angle.*

Theorem 1
Given a set of n geometric objects in the plane, the polar diagram can be used as preprocessing to find the maximum orthogonal visibility angle problem in $O(\log n)$ time.

Proof
It is straightforward to prove this theorem taking into account that for any orthogonal direction, both polar diagrams can be computed in $O(n \log n)$ time. A point location in a polar region is known to be found in logarithmic time.
□

Nevertheless, not only orthogonal visibility problems can be solved using polar diagrams. The generalization to any other direction is the key to deal with other geometric angle problems. If we pretend to provide a robot with automatic movement, or simulate visits to virtual scenes generated in a random way, or to find a solution to collision detection problem, polar diagrams can be used as preprocessing in the plane to improve computation times. These and other visibility problems applications can be taken into account to some interest areas in Computer Graphics. We give in the next section an introduction to the path planning problem.

4.2. Path planning

The motion planning purpose is to provide a mobile object with the capacity of automatic decision about any kind of movement among different obstacles. This mobile object uses to be a robot, thus, any new proposal in the resolution of this problem can be considered a Computer Graphics and Computational Geometry contribution to the Robotics.

Several algorithms have been developed for path planning problems, we find an exhaustive survey in [13]. One of the most important constructs the visibility graph, in which every two vertices are connected with edges if one vertex is visible from the other. The resulting graph is the input to the Dijkstra algorithm [2].

Using the visibility graph, it is possible to find the minimum path from an origin and a destination. Nevertheless the

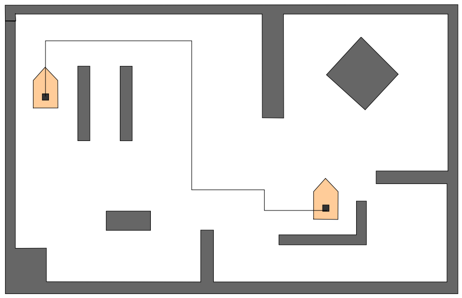

Figure 8: *Path planning in a 2D scene.*

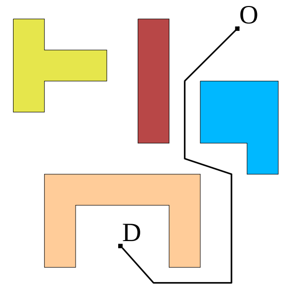

Figure 9: *Path between origin and destination.*

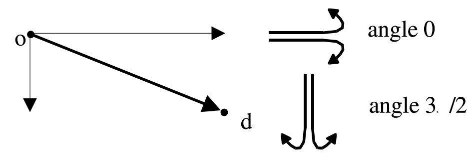

Figure 10: *Vector decomposition into orthogonal components.*

path planning solution using the visibility graph can not be considered a fast method. In some situations it is more desirable a quickest technique than the best one. Some other times, the minimum path does not comply some conditions.

An example of non optimal path planning algorithm is given by Kedem [11]. This proposal performs a trapezoidal map of the free obstacles space, computing vertical lines from every vertex while another obstacle is not intersected. Using this $O(n\log n)$ preprocessing of the plane, a set of free obstacles areas is found, and a path planning algorithm can be computed.

The polar diagram technique for the path planning resolution is also based in the identification of a set of free obstacles regions. In the previous section, we have found a new method based on polar diagrams to solve the maximum visibility angle problem. A new path planning solution can be seen as a chain of points in such a way that each of them can see its successor. All these points are found using the pair of polar diagrams in a determined direction. The result is a polygonal line joining all these visible points.

We use the polar diagram for the following motion planning problem approach. It is considered a system in which there exists a set of planar objects. We assume that the shape and location of these objects are known. Given a initial position o and a destine point d, the path planning problem aim is to find a free obstacles trajectory joining o and d. An example is shown in Figure 8.

We focus our study in a simplification of the problem introduced above: the movement of a point object in a two-dimensional environment. Polar diagrams used in visibility problems are the key to understand how we can move towards a determined direction. For example, whenever point x lies into the polar region of the same object using the two East direction polar diagrams, we do know that this object is the only one obstructing any movement towards this direction. In the opposite case, a East movement is guaranteed and there a free obstacles path exists.

However, a valid path between origin and destination is not always towards an orthogonal direction. Observe Figure 9, even though the vector joining point o and d has a clearly South direction, a West movement is necessary at the end of the route. In the general case, the vector $\overrightarrow{od}$ can be decomposed in two orthogonal components, an horizontal and a vertical component. These new horizontal and vertical vectors determine the direction of the pairs of polar diagrams to use. As it is depicted in Figure 10, vector $\overrightarrow{od}$ has been decomposed giving the pairs of 0 and $3\pi/2$ angle polar diagrams. If any visibility problem in an orthogonal direction is solved using only a pair of polar diagrams, any other direction requires two polar diagrams pairs, the ones given by the direction vector.

The path planning problem uses the technique described above. In order to define a valid path between a origin and destination point, we maintain eight polar diagrams, two in every orthogonal direction. In every step a horizontal, vertical or oblique movement is decided, depending on the $\overrightarrow{od}$ decomposition. A reason to decide one of these types of movements can be the proximity to the destination. Once a valid movement have been processed the origin point changes, and a new trajectory vector is obtained. It is obvious that finally the process finishes when the distance between origin and destination is zero.

Figures 11 and 12 show two different examples of trajectories. Both cases have a common property, the origin and destination vector have a South-East direction, being necessary two pairs of polar diagrams for the path planning resolution. This circumstance and the simplicity of the examples have been chosen for understanding, however the 2D scene complexity does not modify the mechanism followed. When a pair of polar diagrams have provided a portion of trajec-

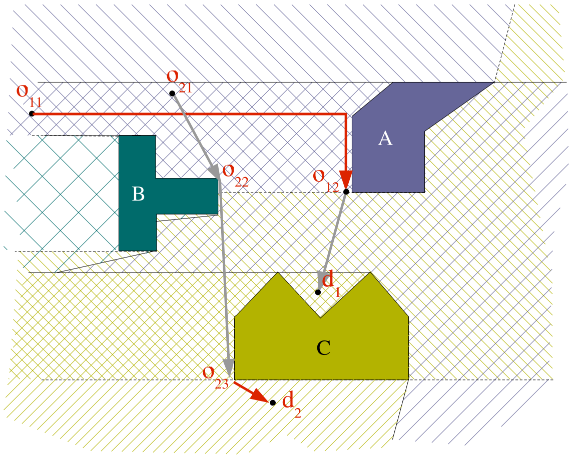

Figure 11: *East direction polar diagrams.*

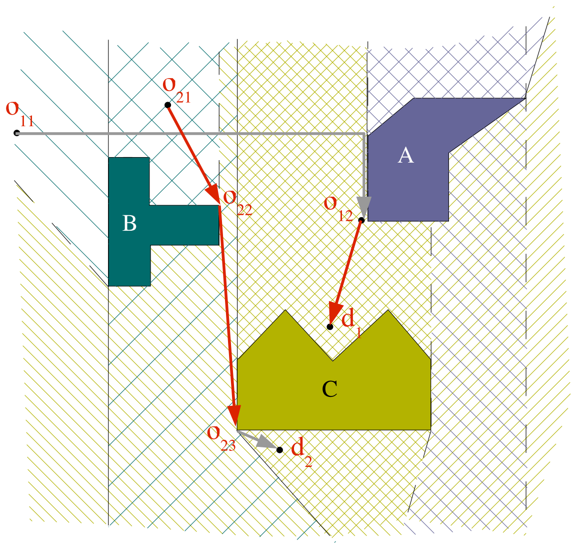

Figure 12: *South direction polar diagrams.*

tory, a red arrow is drawn, otherwise this arrow appears in grey color.

Example 1: The first example is depicted in Figures 11 and 12 using a polygonal line with 1 as first sub-index. The initial vector $\overrightarrow{o_{11}d_1}$ changes every time that its origin changes. The first movement chosen is horizontal because it obtains a nearer position to the destination d_1. Origin o_{11} lies into the both polar regions associated to A in the East direction. We do know this is the only object that obstruct a horizontal trajectory. We choose the orthogonal movement because object B could intersects with the straight line joining o_{11} and o_{12}. However is it straightforward to improve these orthogonal movements because the polar diagram maintains information about adjacent regions and consequently about adjacent objects.

Once point o_{12} is reached, the South pair of polar diagrams is chosen because o_{12} and the destination d_1 belong to the same pair of regions and a free path between then is guaranteed.

Example 2: In this other example, we firstly choose the South polar diagrams pair to perform a movement just towards the frontier of the polar region where o_{21} lies. Again, the vertical trajectory is more interesting because of the final proximity to d_2. From point o_{22}, a path free of obstacles towards o_{23} is guaranteed because o_{22} belongs to different objects polar regions. Finally, any of the pairs depicted in both figures allows to reach destination d_2. In both cases, polar regions in which points o_{23} and d_2 lies, are exactly the same ones.

The number of steps to finalize the process is as much the number of polar regions crossed whose number uses to be lower than methods like trapezoidal maps. Another issue not discussed at the moment are the local minimum points. At every moment a vertical or horizontal path must be decided, if a local minimum is detected, we always have an alternative path to continue, adding only an inclusion test to the algorithm.

Even when this method does not always find an optimal path, some advantages have to be taken into account: the polar diagram computation is a $\Phi(n \log n)$ preprocessing time that can be computed for different geometric objects and provide a intuitive technique to solve visibility and path planning problems.

References

1. V. Chvátal. A combinatorial theorem in plane geometry. *J. Combin. Theory, Ser.* B. 1975. 4

2. E. W. Dijkstra. A note on two problems in connection with graphs . Number. Math. 1959. 4

3. R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.*, 1:132-133, 1972.

4. C. I. Grima, A. Márquez. *Computational Geometry on Surfaces*. Kulwer Academic Publishers, 2001.

5. C. I. Grima, A. Márquez y L. Ortega. A locus approach to angle problems in computational geometry. *14th European Workshop in Computational Geometry,* Barcelona, 1998. 2, 3

6. C. I. Grima, A. Márquez y L. Ortega. A locus approach to angle problems in computational geometry. *Wet and Discrete*, Darwin (Australia), 1998.

7. C. I. Grima, A. Márquez y L. Ortega. Diagrama polar de objetos geométricos. *IX Congreso Español de Informática Gráfica*, Jaén, 1999. 2

8. C. I. Grima, A. Márquez y L. Ortega. Polar diagrams of geometric objects. *15th European Workshop in Computational Geometry,* Antibes (Francia), 1999. 2

9. C. I. Grima, A. Márquez y L. Ortega. Un preprocesamiento para problemas de ángulos en Geometría Computacional. *VIII Congreso Español de Informática Gráfica*, Ourense, 1998. 2

10. R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Info. Proc. Lett.*, 2:18-21, 1973.

11. K. Kedem y M. Sharir. An efficient algorithm for planning collision-free traslational motion of convex polygonal object in 2-dimensional space amidst polygonal obstcles. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.,* 1985. 5

12. V. Klee. Is every polygonal region illuminated from some point?. *Amer. Math. Monthly*. 1969. 4

13. J. C. Latombe, *Robot Motion Planning*, Academic Publishers, Boston 1991. 4

14. L. Ortega. *El Diagrama Polar.* Tesis Doctoral. Universidad de Sevilla. 2002. 3

15. L. Ortega, A. Rueda, C. Grima, A. Márquez. Camino entre obstáculos usando diagramas polares. *XI Congreso Español de Informática Gráfica,* Gerona, 2001.

16. G.M. Voronoi. Nouvelles applications des paramètres continus à la thèorie des formes quadratiques. deuxième Mémoire: Recherches sur les parallélloèdres primitifs. J. Reine Angew. Math. 1908. 1