Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991
Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O.Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

On the number of links and placement of telescoping manipulators in an environment with obstacles

Krasimir Kolarov, Bernard Roth
Mechanical Engineering Dept., Design Div.,
Stanford University, Palo Alto, CA 94305
E-mail: K.KRASI@Macbeth.Stanford.EDU

Abstract--In this paper we consider the following problem: Given an environment with obstacles, what are the minimum number of telescoping links that will allow a manipulator operating in this environment to reach every point in the environment. A geometrical algorithm is presented for solving this problem in a two-dimensional planar case when the obstacles are polygonal. Solutions to some generalizations of this problem are outlined, including simultaneous design of a manipulator and its environment, design of a moving robot and/or obstacles, and design for a three-dimensional workspace. Some theorems on the bounding limits for the number of links of the manipulator are formulated .

1.Introduction

The problem we are interested in is the following: given an environment containing obstacles, what are the minimum number of links and the placement of a manipulator operating in this environment that will allow it to reach any point in the environment?

The primary joint type we employ in this study is a one degree of freedom (dof) prismatic joint which folds into itself like a telescope to some minimal length. The manipulator is formed by connecting each telescopically jointed link to a revolute joint at its other end. The revolute axes are perpendicular to the prismatic axes, (see fig.1). This design is fully realizable in practice and parts of it exist in different forms in different robots. Strictly speaking, each telescoping joints connects two links. So in fig.1, we have 7 links (a,b,c,d,e,f,g) - including the ground link but not the end-effector. For the sake of the discussion, it will be more convenient to think of a link in a different way. Since the only function of the prismatic joint is to change the link's length, we consider the telescoping joint to be part of a link's internal structure. Hence, we have in Fig.1 only 3 moving links (1,2,3). This design gives more flexibility than a

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

traditional revolute - jointed manipulator. It also makes it easier to extend our analysis to other types of environments, such as ones where the obstacles can change their place and shape and the robot can move around rather than being on a fixed base. Our analysis can easily be generalized to 2´D by considering planar layers .

The organization of this paper is as follows: After giving some preliminary definitions in Section 2, we present an algorithm in Section 3 that solves the optimal design problem for a manipulator arm consisting of telescopic links and working in a 2D environment with convex polygonal obstacles. This algorithm classifies every point in the environment according to the number of telescopic links needed to reach this point from a fixed base point. From this classification we can determine the minimum number of links for the robot, the best places for the fixed base, the minimal and maximal lengths of the links, etc.

In Section 4 we outline an algorithm for finding the link center of the environment. (The link center, a term from computational geometry, is the set of points that have minimal link distances to all other points in the environment.)

In Section 5 we consider the case in which we are simultaneously designing the robot and the environment. We also consider a moving robot as well as 2´- and 3-dimensional workspaces.

In section 6 we use the art gallery problem to present some theorems providing upper and lower limits on the minimum number of telescopic links.

Finally some future work is outlined in Section 7.

2. Definitions

In this paper the obstacles are approximated by simple polygons. By "simple polygon" we understand a closed, not self - intersecting chain of straight segments. We also consider the outside boundary of the environment to be a quadrangle, in 2D, or a parallelepiped, in 3D. This simplification makes the examples clearer. The reader should note however that the algorithms and the methods are written in general terms. We will use the term "environment" to denote the whole scene, i.e., the outside boundary as well as the obstacles. By links we mean revolute - jointed telescopic links because these are the only types treated in this paper.

The set of points that belong to the outside bounding polygon EN and do not belong to the obstacles represents all the points that are accessible to the manipulator and is called free space E. We will denote the vertices of the polygons by P_{ij}, where the first index, $i = 0,1,2,...,m$ is the index of that obstacle (0 for EN) and the second index, $j = 1,2,...,m_i$ numbers the vertices of the obstacle in increasing order as we move clockwise, and the vertices of

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

EN, in increasing order as we move counterclockwise. Let n be the total number of vertices in the environment.

Given a point P in the environment E we define the visibility region V(P) as the set of all points Q in E that can be reached from P with one (telescoping) link without collision with the obstacles. For example, V(P) is indicated by the shaded area in Fig.2.

The "link distance" between two points is the minimal number of edges in a polygonal path between the two points that lies in the environment and does not intersect the obstacles. In Fig.2 the link distance between P and Q is 1, while the link distance between P and R is 2.

3. Algorithm for visibility from a point

In this section we assume that the position for the fixed base has been chosen and we show how to determine the minimum number of links necessary to cover the entire environment. We also determine the minimum and maximum lengths of the links. The algorithm is simple: starting with an arbitrary base point B, we build the visibility region V(B), the locus of points that can be reached with one telescoping link from B. Then we add a second link, and determine a second visibility region $V_2(B)$ (assume $V_1(B)=V(B)$) of points that could not be reached with one link, but that can be reached with two links from B (i.e., those that can be reached with one link from the points in $V_1(B)$), and so on. The algorithm stops when we have covered all the points in the free space E.

The basic approach for building the visibility regions is drawing tangents from a generating point towards the vertices of the environment and looking for intersections of these tangents with the edges of the environment. In Fig.3 the generating point on level 1 is the base point B. Then in a clockwise direction we obtain the generating points G_{11}, G_{12} and G_{13} by realizing that points like P_{11} do not contribute to the next visibility region and that points like P_{01} and P_{12} are dominated by G_{11} (also P_{14} by G_{12} and P_{02}, P_{24} by G_{13}), i.e., whatever is visible from P_{01} or P_{12} on the next level is also visible from G_{11}.

In fact the set $V_1(B)$ will consist of the triangles $P_{01}BG_{11}$, $G_{11}BP_{12}$, $P_{12}BP_{11}$, $P_{11}BP_{14}$, $P_{14}BG_{12}$, $G_{12}BP_{24}$, $P_{24}BG_{13}$, $G_{13}BP_{02}$ and $P_{02}BP_{01}$. Some of those are just edges, not real triangles, but for the sake of generality and continuity the algorithm considers all of them. On the next level we start with generating points G_{11}, G_{12} and G_{13}, excluding the free space that has already been covered on the previous levels. The algorithm terminates when the union

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

of the visibility regions $V_i(B)$ (i=1,2,...,k), covers the whole free space with a minimal link distance k. This k is the minimal number of links necessary to cover the whole environment starting from the base point B. In the case of the example seen in Fig.4 this number is 3, since the two unshaded regions are parts of $V_3(B)$ that are not covered by $V_1(B)$ or $V_2(B)$.

In order to decide whether we have covered all points in E for a certain level of visibility we formulate the following theorem:

Theorem 1. If all edges of the environment E can be reached from some point P with k revolute - jointed telescopic links, then all points in E can be reached from P with k revolute jointed telescopic links.

In practice, an edge AB will be reached from a point C with k links if the segment of the line AB that is contained in E can be reached with one link from the generating points for C on level k-1. This condition can easily be checked algorithmically.

We emphasize that in addition to the tangents from the generating points on each level we need also to consider the mutual tangents between the obstacles that intersect the sets of reached points on the previous level. For example the unshaded area on Fig.5 is actually visible from B with 2 telescoping links, i.e., it belongs to $V_2(B)$ but can not be included there by using the generating points G_{11} and G_{12}. This example actually corresponds to visibility of the unshaded area from the whole segment $P_{16}G_{12}$, rather than just from the vertex G_{12}. In the literature, this is referred to as a link visibility.

We also find as a by-product the minimum and the maximum lengths of the links for each level.

In our problem we are not concerned with the complexity of the algorithm (which is generally the main issue in computational geometry) because we will be building these regions only once, off-line for design purposes. By comparison, the algorithm of Suri and O'Rourke [2] has a complexity that can be as high as $\frac{1}{2}(n^4)$ since on each level it builds the vertex and edge visibility regions using a triangulation approach. Mitchell, Rote and Woeginger have used similar techniques in [3] and they have recently reported an algorithm with complexity O(Elogn) for building the region illuminated from all the edges, where E is the number of visible pairs among the obstacle vertices.

Our visibility regions also consist of triangles, but we generate these triangles as we go and we do not have to generate an initial triangulation of the environment.

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

In the implementation of the algorithm, we initially grow the obstacles by the width of the links. At present we choose the base point using simple heuristics, namely, we choose it at a reasonable distance from the edges of the environment (both the outside boundary and the obstacles) and with the maximal sum of distances to the edges that are reachable with one link. The question of choosing the base point is a central one and will be dealt with in more detail in the following sections.

We want to point out that the described algorithm can easily be extended to generalized polygons (to polygons whose edges can be either straight lines or arcs of circles), using the tangents to the circular arcs. Of course any convex shape can be approximated closely by convex polygons, and theoretically this approach extends to all convex obstacles, but in reality it strongly depends on the number of edges since close approximation using a polygon with numerous edges will be impractical. In such more complex cases the problem can perhaps best be attacked topologically.

4. Algorithm for generating the link center

It appears that once we have chosen the base point, finding the minimal number of telescopic links to cover the environment is just a matter of running the algorithm presented in Section 3, but we do not have any guarantee that we have a global optimum. It could be that if we chose another base point relying on some better heuristics we would have a smaller number of links. The problem then is to find the set of points in the environment that are the base points from which we can reach all other points with a minimum number of links. These points form the "link center" of the environment. Using the notion of link distance defined in Section 2 we can introduce, as in [4], the link diameter and the link radius of the environment as respectively the maximal and the minimal link distances between any two points in E. In this context the link center will consist of all points that are at a link distance equal to the link radius from all other points in the environment [4]. Theorem 4 (in [4]) states that the link center is formed by all convex vertices in the environment for the case of a simple polygon. In our problem the environment is a simple polygon with holes (the obstacles). We will cut our environment as many times as the number of holes to obtain a simple polygon. In Fig.6a we have used two cuts – along $P_{01}P_{11}$ and along $P_{03}P_{22}$ thus creating the simple polygon in Fig.6b, in which the convex vertices are P_{01}, P_{01}', P_{02}, P_{03}, P_{03}', P_{04}, P_{11}, P_{11}', P_{22} and P_{22}', i.e., all vertices of the outside boundary EN and the two vertices of the obstacles where we have made the cuts. Of course if we use two other cuts we can get two other vertices of the

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

obstacles to contribute to the link center, and so on. By analogy we can cut anywhere along the edges of E, and we find that all points along the boundary of E will contribute.

Thus we have proven the following:

Theorem 2. The link center for our problem is generated by all edges in the environment, i.e., both those of EN and those of all the obstacles.

This theorem basically says that the link center of E is the intersection, which is not empty, of the same lowest level k visibility regions for all edges of E. Thus Theorem 2 is a corollary of Theorem 1 because all points in this intersection are at a link distance k to all edges of E and consequently at a link distance k to all points in E (Theorem 1). This is the reason why these points form the link center of E. This idea immediately gives us an algorithm for finding this link center, namely:

```
    begin
    k:= 0;
a:  k:= k+1;
    S_k:=E;   {E is the whole free space}
    for i=0,...,m do
       for j=1,...,m[i] do
         begin
         Build visibility region V_k(e_ij);
         {e_ij denotes the edge P_ij P_ij+1}
         S_k = S_k () V_k(e_ij);
         end;
    if S_k is empty then go to a else go to b;
b:  end;
```

The outputs of the algorithmis are the link center S_k and the link radius k of E.

It is important to note that our notion of visibility of an edge is different than the well known concept of visibility from an edge. We are interested in the set of points that can see with certain number of links the whole edge, not just any point on the edge. That is why, for example, the first visibility region of edge e_{ij} will be the simply connected component, containing the edge, of the intersection of the visibility regions for P_{ij} and P_{ij+1}. The result of the application of the link center algorithm for the example on Fig.3 is given on Fig.9 in Section 6.

Once we have found the link center, we can choose any point in it as a base point for the robot. This choice can be made using heuristics similar to those in Section 3 but now we are guaranteed to get the minimum number of links.

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

We want to mention here that with the algorithm in Section 3 extended to non-convex obstacles, all the results for the link center will automatically apply to more general obstacles.

5. Generalizations of the design problem

In this section we extend our original problem so as to
- deal with 3D environments,
- design the environment and the robot simultaneously,
- deal with a moving robot.

Our algorithm can readily be applied to 2´D. We are given a 3D space with some objects and we want to design and place a telescoping manipulator that can reach all the points in the space. To deal with such spaces, we add a prismatic joint between the manipulator's base, B, and the ground, in such a way that the translation action raises and lowers the entire planar manipulator structure normal to its original plane of action. Now we can

a) divide the space into a reasonable number of horizontal layers;
b) for each layer find the link center;
c) project all link centers of the same link radius (the largest one) on one horizontal plane and find the intersection;
d) for each disconnected part of this intersection check if it can contain a circle with radius equal to the sum of the minimum lengths of the links ; if it can, then we can place the base point there;
e) having chosen the base point for each layer we will also have found the minimum number of necessary links.

For real 3D operations we need to consider spherical rather than revolute joints and the geometry of the problem is different, but the tangential approach from Section 3 can still be applied.

The second problem we discuss in this section is the following: Given a closed space and a set of objects (considered as obstacles), where should the objects be placed in order for the manipulator to reach every point in the free space with the minimum number of telescopic links. This problem allows more flexibility because now we can move the objects around in order to require fewer telescoping links.

If we restrict ourselves to a rectangular EN then in order to simplify the problem, we can approximate each obstacle with its minimal bounding rectangle in 2D or parallelepiped in 3D. If we denote with $A(OB_i)$ the smallest area rectangle enclosing the obstacle OB_i then Toussaint [6] has an algorithm that finds $A(OB_i)$ in linear time. In terms of Fig.7 the algorithm starts with any edge $P_{ij-1}P_{ij}$, and

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

the bounding rectangle with the smallest area is $L_jL_kL_lL_q$, containing the edge $P_{ij-1}P_{ij}$. Then it finds the minimum of the angles a_j, a_k, a_l, $a_q - a_l$ in our case, and builds the next rectangle based on the edge, corresponding to angle $a_l - L_jL_kL_lL_q$. The procedure continues in this manner until it comes back to $P_{ij-1}P_{ij}$, at which point the rectangle with minimum area is the one we are looking for.

Of course this procedure will work for non-convex polygons too. The question of re-orienting the obstacle on the scene has also been taken care of. We just need to place those rectangles for the obstacles within the rectangle of the outer boundary. One way to do this is to sort the rectangles by area and place them in horizontal layers. In that case we can prove that we need only three prismatic links to cover the whole environment. Alternatively, a more formal optimization problem can be formulated.

Having placed all the obstacles, we can apply the procedure from Section 4 to design the optimal manipulator.

With a moving robot in the environment we would most probably save a certain number of links. As a by-product of the program in Section 3, we see that to every point in the environment we have a path from the base point with a minimum number of links. We can take this path and try to move the robot along the path as far as it can go without collision, thus saving one or several links. In 3D we will have to plan the movements of fewer links, eventually just the last one. We can also use moving robot to solve the motion planning problem in our context. Theoretically, if we want to get from one point to another we can start by reaching the first point from the fixed base, then folding the robot into the base, and then reaching out to the second point from the base. While in practice we cannot fold the robot completely, we can always perform this maneauver with a moving robot because we just move as close as possible to the first point, stretch out to reach it, grasp something, return to the original "base" point, then move as close as possible to the second point and reach for it. We are working on implementing these ideas algotithmically.

6. Some upper and lower bounds on the visibility

Most of our results so far are algorithmic, i.e., we apply a certain procedure and at the end we have both the minimum number of telescopic links necessary for E and a classification of the points in E depending on the number of links used to reach them. In this section we show how to find lower and upper bounds on the number of links needed to cover E without having to explicitly build the visibility regions.

In deriving those results we use the approach used for guard placement in an art gallery, discussed extensively by

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

O'Rourke [1]. In an art gallery (i.e. a simple polygon with or without holes) the question arises how many guards are needed and where they should be placed so that they can see the whole inside (in some generalizations also the outside) of the gallery. The basic approach taken in [1] is first to triangulate E, i.e., to connect all vertices of E with straight edges so that they form triangles that cover E. Then the vertices are 3-colored in such a way that in every triangle the three different vertices have different colors. (It can be shown that there always exist triangulations for which 3-coloring is possible.) Finally the guards are placed at the vertices, having the color that appears the least number of times. This procedure is illustrated in Fig.8 where R, B and G denote red, blue and green respectively. Here it turns out that each color appears at 6 vertices and we can therefore choose any of the three sets. We arbitrarily choose the red vertices.

The main results for polygons with holes (the term used in [1] for environments with obstacles) are

[(n+2*h)/3] vertex guards are sufficient to cover the whole environment E;

[(n+h)/3] vertex guards are sometimes necessary (established through examples) to cover E;

[(n+1)/3] vertex guards are sufficient for one hole;

[(n+h)/3] vertex guards may be sufficient for the general case.

where h is the number of holes (obstacles) and n is the number of all vertices.

For Fig.8 those results will give us [(18+6)/3] = 8 guards as an upper bound, [(18+3)/3]= 7 guards are sometimes necessary. In fact 6 vertex guards are sufficient here.

We can prove

Theorem 3. In the worst case, the visibility of the environment E is equal to the number of guards covering E.

The reasoning behind this theorem is the following: Because of the connectedness of the environment E, every guard has at least 2 neighboring guards in a sense that it is a vertex in a triangle that has a side common to a triangle with a vertex at the other guard. In Fig.8 for example the guard at P_{01} belongs to the triangle $P_{01}P_{15}P_{02}$, which has side $P_{15}P_{02}$ common with the triangle $P_{15}P_{02}P_{14}$, which has as a vertex the guard P_{14}. Thus we can build a graph with vertices representing the guards and edges representing the relationship of neighboring guards (see fig.10).

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

The longest possible chain is one that goes around all the guards. We can think of it as one big obstacle. In that case, to go from any point to any other point, instead of going first to a guard and then following the guards on the chain, we can just go around the obstacle or the outside boundary. Because E is connected, the largest number of moves to go around at most n/2 vertices (either of EN or of the obstacle) to a guard opposite on the chain is n/4. We also need one more link to go from this guard to any point reachable from it. Now we have n/4+1, which for n>7 is less than (n+1)/3 - the number of guards. For n=6 and n=7 the theorem is satisfied trivially. We can also prove this theorem by accounting for the fact that the distance between any two connected guards is at most 2 links.

From this theorem it follows that the upper and lower bounds found for the vertex guards apply also to our problem for the visibility of the region.

However the link center for our example on Fig.9 shows that the link radius is 2, and that 3 telescopic links are enough to cover the whole E without regard to where we put the base point. The unshaded areas constitute the link center - the regions of points that can reach all the other points with a maximum of 2 links (the boundary points are not included). The rest of the points in the free space E need 3 links to reach some of the other points in the environment. Consequently we can say that the visibility of this problem is 3 - the maximum number of telescopic links needed to cover the whole E. As we see, this is half the number of the vertex guards for this problem. The reason is that we have restricted ourselves to vertex guards that are stationary points while with the links we are moving from region to region. Our visibility reflects the optimal solution for this environment while the number of guards is just a solution for the art gallery problem. The problem of minimal guard cover is discussed in [1]. The results there show that this problem is NP-hard.

We will outline here an approach for reducing the number of guards for this environment to ideally the optimal number. We start by pointing out that in several instances triangles that are completely visible by one vertex guard belong to another guard's patrol area. For example in Fig.11 triangles $P_{13}P_{14}P_{24}$ and $P_{13}P_{24}P_{25}$ are both visible to guards P_{14} and P_{25}. It is clear that if all the triangles visible to one guard are also visible to another guard, then this second guard is unnecessary. In our example everything visible to guard P_{25} is visible to P_{14}. Similarly, everything visible to guard P_{31} is visible to P_{01} or P_{33}. Consequently the guards in P_{31} and P_{25} can be eliminated, and we still have complete cover of E, now with only 4 vertex guards.

Now the next step is to introduce non-vertex guards

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

(the so called Steiner points). Consider the guard at P_{33}. In the area covered by it there are two patches that cannot be seen by the rest of the guards. One of these patches is based on the edge $P_{12}P_{13}$ and is depicted on Fig.11 as the shaded area 1.

Now if we can move the guard at P_{01} to a position from which it can see $P_{12}P_{13}$ then from this side guard P_{33} will be redundant. The new guard should also be able to see $P_{11}P_{15}$, consequently we can place it at the intersection of the lines $P_{15}P_{11}$ and $P_{13}P_{12}$. If this point is in E, which in our case is conveniently true, this gives point S_1. Similarly, the other area that is seen by P_{33} but not by the other guards is the shaded area 2 based on the edge P_{32} P_{33}. Therefore we can move the guard in P_{03} to point S_2 – the intersection of the lines $P_{24}P_{23}$ and $P_{32}P_{33}$, which is also inside E. Finally with guards (not necessarily vertices) at S_1, S_2 and $S_3=P_{14}$ we can cover the whole environment E. This is the optimal number of guards because by using arguments similar to those above we can show that it is not possibly to reduce the number of guards by moving them around.

If we apply the result of Theorem 3, we can prove that the visibility of this problem is 3. This method therefore gives us a faster way than the one described in Section 4, to evaluate the visibility of the environment without having to build explicitly all visibility regions.

In Section 3 we proved that if all edges of E are reachable from a given point with k telescopic links then all the points in E can be reached with k links from this point. To find the visibility of E, we need to know this k for all the edges in E. We will use the triangulation of E for this task. Let us take edge $P_{12}P_{13}$ as an example. We can build the graph in Fig.12 based on the connectedness of the vertices of E.

We see from this example that the longest link distance from $P_{12}P_{13}$ to other vertices is 3 (to P_{15}, P_{33}, P_{02}, P_{03}, P_{21}, P_{22} and P_{23}). The last five of those vertices form edges for which the visibility could be 4. But edge $P_{02}P_{03}$ is visible from P_{13} and therefore its visibility is 3. Similarly, $P_{21}P_{22}$ is visible from P_{12} and line $P_{22}P_{23}$ is visible from P_{13} inside E. The conclusion is that all edges and consequently all points in E are visible from the edge $P_{12}P_{13}$ with 3 links. After applying this relatively fast procedure for the other edges, we can conclude that the visibility of the region is 3. To make this procedure even

Copyright 1991 IEEE, Published in the Proceedings of the International Conference on Advanced Robotics, ICAR'91, Pisa, Italy, 1991

faster we can arrange all vertices in a net with all possible connections and references to the corresponding edges, and process this net in parallel to extract the visibility.

7. Future work

We are working on completing the implementation of the algorithms for link center and those in Section 6. We also plan to extend our research to more complex obstacle shapes, to 3D environments, to just revolute links and eventually to robotic systems other than manipulator arms.

REFERENCES

[1] J.O'Rourke, Art gallery theorems and algorithms, Oxford Press, 1987.

[2] S.Suri, J.O'Rourke, Worst-case optimal algorithms for constructing visibility polygons with holes, Proc. 2nd annual ACM Symposium on computational geometry, 1986.

[3] J.Mitchell, G.Rote and G.Woeginger, Minimal link paths among obstacles in the plane, Proc. 6th annual ACM Symposium on computational geometry, 1990.

[4] W.Lenhart, R.Pollack, J.Sack, R.Seidel, M.Sharir , S.Suri, G.Toussiant, S.Whitesides and C.Yap, Computing the link center of a simple polygon, Proc. 3rd annual ACM Symposium on computational geometry 1987

[5] S.Suri, A linear time algorithm for minimum link paths inside a simple polygon ; Computer vision, graphics and image processing, 35, 1986.

[6] T.Toussaint, Solving geometric problems with the 'rotating calipers', Proc. IEEE MELECON'83, Greece

[7] A.Aggarwal, The art gallery theorem : Its variations, applications and algorithmic aspects, Phd thesis, Johns Hopkins University, 1984.