Proceedings of the 10th Mediterranean Conference
on Control and Automation - MED2002
Lisbon, Portugal, July 9-12, 2002.

Application of Linguistic Geometry to Real Time 3D Navigation of Multiple Robots

S. Khanmohammadi*, A. Aghagolzadeh, M. A. Badamchi Zadeh†, S. Ghaemi††**

Email: * khan@tabrizu.ac.ir ** aghagol@tabrizu.ac.ir
† mbadamchi@tabrizu.ac.ir †† sgaemi@tabrizu.ac.ir

Keywords: Linguistic geometry, Real time 3D path planning, Multi robot

Abstract
Linguistic Geometry (LG) is a new method for decreasing the number of branches in a search tree. It is applicable for solving a variety of search problems such as path planning for a moving robot. In this paper the Linguistic Geometry is used to reduce the calculation time in real time 3D navigation of multiple robots in industrial plants.

1. Introduction

Linguistic Geometry (LG) is introduced by PIONEER team in early 80 [1].This methodology is developed by Boris Stilman for military and industrial applications [1-3]. Linguistic Geometry is a procedure for converting human intelligence in search problems to mathematical formulas and computer programs. The main idea of LG is to divide a global problem to several subproblems and to use a hierarchical methodology for solving the individual subproblems. LG allow to discover the inner properties of human expert heuristics that are successful in a certain class of games.
In a two-player opposing game, such as chess, by applying the brute force search algorithm we have to generate a search tree of size T.

$$B + B^2 + ... + B^L = T \quad (1)$$

or

$$\frac{B^{L+1} - 1}{B - 1} = T \quad (2)$$

The branching factor B is a parameter representing the average breadth of the search tree. It shows how many moves (on the average) should be included in this tree at each node. T is the total number of positions generated and L is the depth of the search (assuming all the branches are terminated at the depth L). Greater values of B correspond to a non-selective search; indicating a high exponential growth.
Various search algorithms, such as dynamic programming and branch-and-bound algorithms, were constructed in order to reduce the branching factor.
Alpha-Beta search algorithm can reduce the number of terminal nodes to be visited as follows[1]:

$$\begin{cases} 2B^{L/2} - 1 & \text{if } L \text{ is even} \\ B^{(L+1)/2} + B^{(L-1)/2} - 1 & \text{if } L \text{ is odd} \end{cases} \quad (3)$$

However the grows is still exponentially, albeit with a reduced exponent. The perfect ordering can theoretically double search depth (during the same time frame) employing the reduced branching factor $\sim\sqrt{B}$.

The main advantage of LG is its ability to decrease the number of searches in the large scale problems for getting the optimal or suboptimal solution. For example in a typical three-dimensional path planning problem the number of searches decreases from one billion branches to 56 only [1].

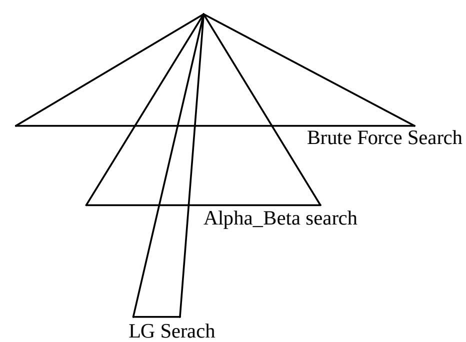

Fig 1. Comparison of searches for the same processing time

In the basic definition of LG two types of elements are considered where two sets react in two independent (with a bit of overlap) areas. While in

[1] Slagle and Dixon, 1969, Knuth and Moore, 1975

real world applications several areas may be considered with some overlaps [1-8].
The object of this paper is the development of LG to multi 3D robotic environments. Some basic definitions of LG are presented in the next sections.

2. Complex System

Definition: A Complex System is presented by the following eight-tuple [2]:

$$< X , P , R_p , \{ON\} , v , S_i , S_t , TR > \quad (4)$$

where:
$X = \{x_i\}$ is a finite set of points (locations of elements);
$P = \{p_i\}$ is a finite set of elements (a union of two nonintersecting subsets P_1 and P_2);
$R_p(x, y)$ is a set of binary relations of reachable in X (x and $y \in X$ and $p \in P$) ; Element p can move from point x to point y if point y is reachable from x , i.e. $R_p(x, y)$ holds.
$ON(p) = x$, where ON is a partial function of placement from P into X;
v is a function on P with positive integer values describing the values of elements;
S_i , S_t are the descriptions of the initial and target states. The Complex System searches the state space, which should have initial and target states;
TR is a set of operators , $TRANSITION(p, x, y)$ of transitions of the system from one state to another.

The problem of the optimal operation of the system is considered as a search for the optimal sequence of transitions leading from the initial state S_i to a target state S_t based on a certain criteria for optimality.

3. Autonomous Robots as Elements of the Complex System

A robot control model can be represented as a Complex System.
X represents the operational district [1];
P is the set of robots or autonomous vehicles;
$R_p(x, y)$ represent moving capabilities of different robots;
$ON(p) = x$ if robot p is at point x;
$v(p)$ is the weighting value of robot p;
S_i is a set of arbitrary initial or starting states;
S_t is the set of target states;
$TRANSITION(p, x, y)$ represents movement of robot p from x to y; if there is a robot on point y, it must be removed before movement of robot p from point x to point y.

4. Distance Between Elements

Definition: A *MAP* of the set X with respect to point x and element p for the Complex System is the mapping [3]:

$$MAP_{x,p} : X \rightarrow Z_+ \quad (5)$$

(where $x \in X, p \in P$ and Z_+ is the set of all non-negative integer numbers).
MAP is constructed as follows:
Consider a family of reachable areas from the point x, i.e. a finite set of the following nonempty subsets $\{M^k_{x,p}\}$ of X (as show in figure 2):
$k = 1: M^k_{x,p}$ is a set of points m reachable in one step from x when $R_p(x,m)=True$;
$k > 1: M^k_{x,p}$ is a set of points reachable in k steps and not reachable in $k-1$ steps, i.e. points m are only reachable from points $M^{k-1}_{x,p}$.
Let:
$MAP_{x,p}(y) = k$, for y from $M^k_{x,p}$, denotes the number of steps from x to y .

For other points we have

$$MAP_{x,p}(y) = 2n \quad \text{if } y \neq x \quad \text{and} \quad (6)$$
$$MAP_{x,p}(y) = 0 \quad \text{if } y = x$$

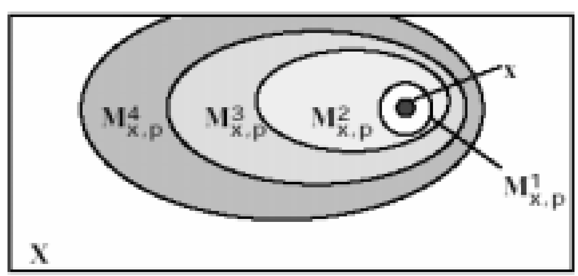

Fig 2. Interpretation of the family of reachable areas

5. Trajectories

Definition: A trajectory for an element $p \in P$ with the origin at $x \in X$ and the destination at $y \in X$ $(x \neq y)$ with a length l is the following formal string of symbols $a(x)$ with points of X as parameters:

$$t_0 = a(x)a(x_1)...a(x_l) \qquad (7)$$

Where $x_l = y$ and each successive point x_{i+1} is reachable from the previous point x_i, i.e. $R_p(x_i, x_{i+1})$ holds for $i = 0,1,...,l-1$; element p stands at the point x: $ON(p) = X$ [4].

$t_p(x,y,l)$ is the set of all trajectories for element p, originated at x with destination y and length l.

Definition: the shortest trajectory t of $t_p(x,y,l)$ is the trajectory of minimum length for the given origin x, destination y, and element p.

The above and following Properties of Complex System allow us to construct formal grammars for generating the shortest trajectories.

Definition: An admissible trajectory of degree k is the trajectory t_0 which can be divided into k shortest trajectories; more precisely there exists a subset $\{x_{i_1}, x_{i_2}, ..., x_{i_{k-1}}\}, i_1 < i_2 < ... < i_{k-1}, k \leq l$ such that corresponding substrings $a(x_0)...a(x_{i_1}), a(x_{i_1})...a(x_{i_2}),...,a(x_{i_{k-1}})...a(x_l)$ are the shortest trajectories (as shown in figure 3).

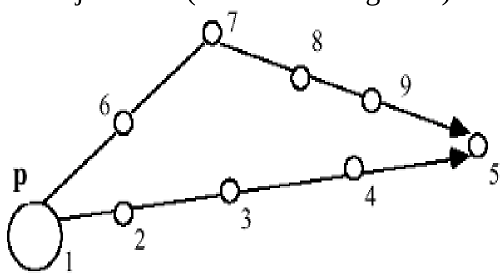

Fig 3. An interpretation of shortest and admissible trajectories

6. Control Grammar

Consider the following control grammar for the Complex System with symmetric relation of reachable R_p, as summarized in table 1 [5].

Table 1. summary of Grammar of shortest trajectories $\mathbf{G_t}$

L	Q	Kernel, π_k	π_n	F_T	F_F
1	$\mathbf{Q_1}$	$S(x,y,l) \rightarrow A(x,y,l)$		two	ϕ
2_i	$\mathbf{Q_2}$	$A(x,y,l) \rightarrow A(x, med_i(x,y,l), lmed_i(x,y,l))$ $A(med_i(x,y,l), y, l - lmed_i(x,y,l))$		three	three
3_j	$\mathbf{Q_3}$	$A(x,y,l) \rightarrow a(X) A(next_j(x,l), y, f(l))$		three	4
4	$\mathbf{Q_4}$	$A(x,y,l) \rightarrow a(y)$		three	5
5	$\mathbf{Q_5}$	$A(x,y,l) \rightarrow \mathbf{e}$		three	ϕ

Pred=**{Q₁, Q₂, Q₃, Q₄, Q₅}**

$\mathbf{Q_1}(x,y,l) = (MAP_{x,p}(y) \leq l < 2MAP_{x,p}(y)) \wedge (l < 2n)$

$\mathbf{Q_2}(x,y,l) = (MAP_{x,p}(y) \neq l)$

$\mathbf{Q_3}(x,y,l) = (MAP_{x,p}(y) = l) \wedge (l \geq 1)$

$\mathbf{Q_4}(y) = (y = y_0)$

$\mathbf{Q_5}(y) = (y \neq y_0)$

($n = |X|$, number of points in X)

$f(l) = l-1$ $\qquad$ $D(f) = Z_+ / \{0\}$

$\mathbf{L} = \{1,4\} \cup two \cup three$ $\qquad$ two=$\{2_1, 2_2, ..., 2_n\}$, three=$\{3_1, 3_2, ..., 3_n\}$

where **L** is a finite set called the set of labels;

$\mathbf{Q_i}$ represents the condition of applicability of productions (antecedents)

If $\mathbf{Q_i} = \boldsymbol{T}$ then $\boldsymbol{F_T}$ (a permitted subset of **L**) is reachable in the next step;

$\boldsymbol{F_F}$ is analogous to $\boldsymbol{F_T}$ in the case of $\mathbf{Q} = \boldsymbol{F}$.

Where $\boldsymbol{T}$ is true and $\boldsymbol{F}$ if false.

At the beginning of derivation: $x = x_0$, $y = y_0$, $l = l_0$, $x_0 \in X$, $y_0 \in X$, $l_0 \in Z_+$, $p \in P$

med

$\boldsymbol{med_i}$ is defined as follows:

$DOCK = \{v | \, v \in X, MAP_{x_0,p}(v) + MAP_{y_0,p}(v) = l\}$

IF

$DOCK_l(x) = \{v_1, v_2, ..., v_m\} \neq \{\}$

THEN

$med_i(x, y, l) = v_i$ $\qquad$ for $1 \leq i \leq m$ and

$med_i(x, y, l) = v_m$ $\qquad$ for $m < i \leq n$

OTHERWISE

$med_i(x, y, l) = x$

$\boldsymbol{lmed_i}$ is defined as follows:

$lmed_i(x,y,l) = MAP_{x,p}(med_i(x,y,l))$

$\boldsymbol{next_i}$ is defined as follows:

$SUM = \{v | \, v \in X, MAP_{x_0,p}(v) + MAP_{y_0,p}(v) = l_0\}$

$ST_k(x) = \{v | \, v \in X, MAP_{x,p}(v) = k\}$

$$MOVE_l(x) = ST_1(x) \cap ST_{l_0-l+1}(x_0) \cap SUM$$

IF

$$MOVE_l(\mathrm{x}) = \{ \mathrm{m}_1, \mathrm{m}_2, \ldots, \mathrm{m}_r \} \neq \{\}$$

THEN

$$next_i(\mathrm{x}, l) = \mathrm{m}_i \quad \text{for } i \le r \text{ and}$$

$$next_i(\mathrm{x}, l) = \mathrm{m}_r \quad \text{for } r < i \le n$$

OTHERWISE

$$next_i(\mathrm{x}, l) = \mathrm{x}$$

Theorem: All the admissible trajectories $t_p(x_0, y_0, l_0)$ of degree 2 from point x to point y of the length l_0 for the element p on x, $ON(p) = x$, can be generated by the grammar $\mathbf{G_t}$ [5].

In this procedure the trajectories of robots are calculated off-line. The following algorithm is introduced in this paper to change this algorithm to an on-line version for application on a multi robot system.

8. On line control grammar

The flowchart of this algorithm is shown in figure 4.

This algorithm may be used sequentially or in parallel to calculate the next point for each robot. In this algorithm the $next_i(x,l)$ is calculated by means of LG method. If the $next_i(x,l)$ is empty, the robot moves to this point; otherwise the $DOCK_l(\mathrm{x})$ of adjacent points of the robot are calculated and the robot will move to the point with minimum (sub minimum) $DOCK_l(\mathrm{x})$.

In the original control grammar with off-line calculations, if the robot is inhibited by other robots a new path must by calculated from the start point and the trajectory traveled by the robot till its current position will have no use [5], while in on-line applications the robot will never return to its origin for choosing a new path [9, 10].

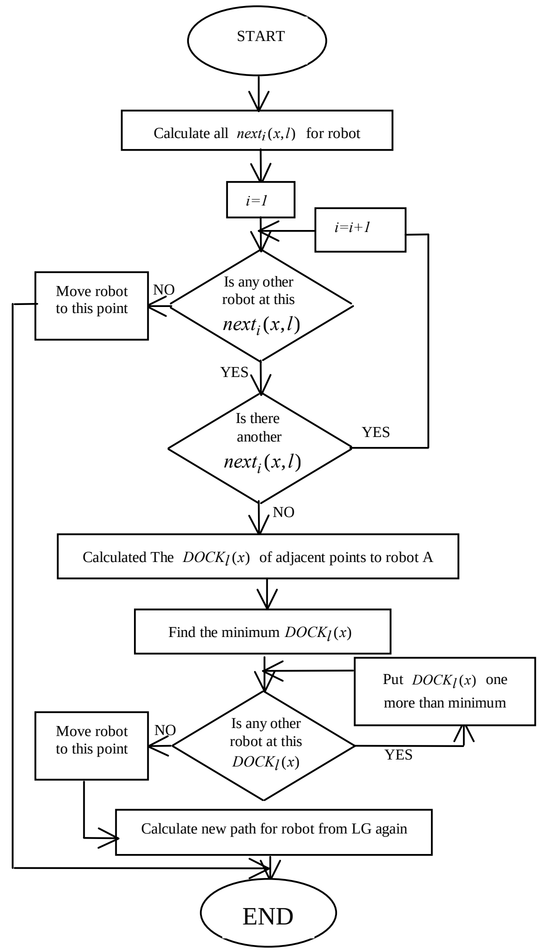

Fig 4. Flowchart of on-line control grammar

Example:

Consider the plant of figure 5. with two robots A and B and their targets TA and TB respectively. Robot A stands at (1, 1, 1) with its target TA in (9, 2, 2) and robot B stand at (9, 1, 1) with its target TB at (1, 2, 2). Both robots have the same importance (weighting) and each robot can move to one of its adjacent points in each step. The bold points of figure 5 represent the static obstacles. Each robot tries to attain its target by moving step by step via its adjacent points toward the target. The robots are considered to move asynchrony in the order A and B. This order of movements is fixed and may not change during the movements. If a robot is inhibited by other robots to move towards its target, it can move backward to modify its situation for the next movement.

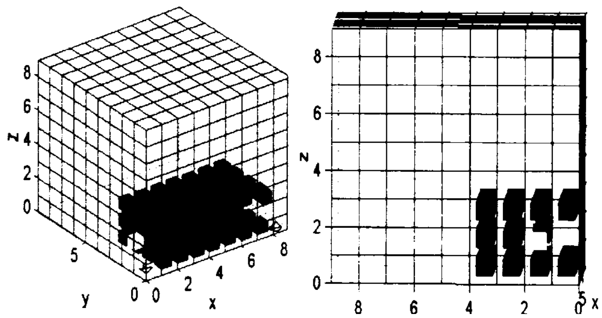

Fig5. Initial state of robots (attention in row Y=2 and Z=2 only X=5 is obstacle)

Here the on-line control grammar is applied for the system of figure 5.

Figures 6 and 7 show the SUM_A and SUM_B for robots A and B respectively. Each *SUM* represents the set of all shortest paths from each robot to its target.

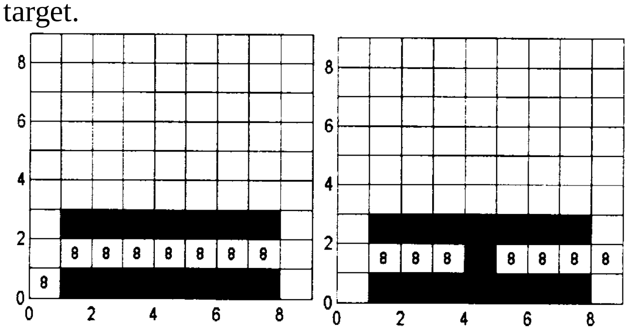

Fig6. SUM_A (projection to XZ plane) left for Y=1 and right for Y=2

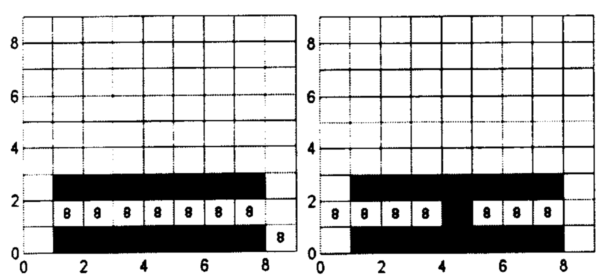

Fig7. SUM_B (projection to XZ plane) left for Y=1 and right for Y=2

Now consider the following paths for robots:

t_A= a(1, 1, 1) a(2, 1, 2) a(3, 1, 2) a(4, 1, 2)
t_B= a(9, 1, 1) a(8, 1, 2) a(7, 1, 2) a(6, 1, 2)

Figure 8 shows the trajectory of two robots.

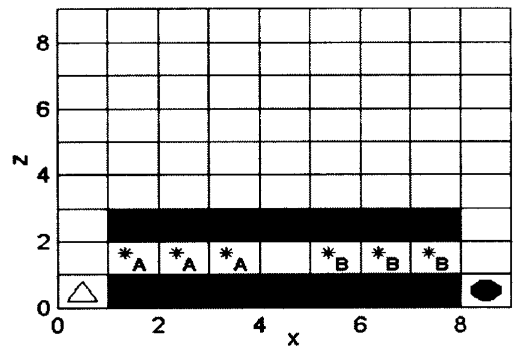

Fig 8. Robots A and B at points (4, 1, 2), (6, 1, 2) (projection to XZ plane)

Suppose it is A's turn to move. $next_i((4,1,2),5)$ for this robot is (5,1,2), hence A moves to this point. Now B will move. $next_i((6,1,2),5)$ for this robot is (5,1,2) but this point is occupied by A. Hence the minimum $DOCK_l(x)$ of B is calculated. It is point (6, 2, 2). B returns to point (6, 2, 2). A new path is calculated. But no change occurs in its path.

Once more, A moves. There is two $next_i((5,1,2),4)$ points for A, (6, 1, 2) and (6, 2, 2). But B is at point (6, 2, 2), hence A moves to (6, 1, 2). Now for robot B $next_i((6,2,2),5)$ is (5, 1, 2). At this moment point (5, 1, 2) is empty and B moves to it.

Continuing this procedure the following paths are passed through by A and B

t_A= a(5, 1, 2) a(6, 1, 2) a(7, 2, 2) a(8, 2, 2) a(9, 2, 2)
t_B= a(6, 2, 2) a(5, 1, 2) a(4, 2, 2) a(3, 2, 2) a(2, 2, 2) a(1, 2, 2)

and two robots get their targets without returning to their start points for searching new paths.

9. Conclusion

The example considered in this paper demonstrates the power of the Linguistic Geometry in transferring the heuristic searching potential of human brain to a computerized calculations domain.

The conventional approaches require a search tree of approximately one billion branches to solve this problem, while the LG tree presented in this paper consists a few moves (branches).

Looking at the complexity of the hierarchy of languages that represent each state in the search process, it is very likely that the growth from the

2D case to 3D is linear with the factor *B* (of search tree) close to one.

A new procedure is introduced for path planning of multi-robot navigation using the Linguistic Geometry. The calculation time is very small for a large-scale problem comparing classic search algorithms. Using the introduced methodology the LG is applicable for real time multi-robot navigation even if there is overlaps in different paths.

References

[1] B. Stilman, "Linguistic Geometry: A New paradigm for intelligent Systems", IEEE proceeding of the 28th annual Hawaii International Conference on Systems Sciences. pp,62-72, 1995

[2] B. Stilman, "Managing Search Complexity in Linguistic Geometry", IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics, Vol. 27, No. 6, pp. 978-998, Dec. 1997

[3] B. Stilman, "Network Languages for Complex Systems", An International Journal: Computer & Mathematics with Applications, Vol. 26, No. 8, pp. 51-80, 1993

[4] B. Stilman, "Linguistic Geometry Tools: Selected Topics",Proc. of the IEEE Int Conf. on Systems, Man, and Cybernetics, La Jolla , Ca , Oct . 1998

[5] B. Stilman, "A Linguistic Approach To Geometric Reasoning ", An international Journal: Computers & Mathematics with Applications, Vol.26, No.7, pp.29-58, 1993

[6] B. Stilman, " Heuristic Networks for Space Exploration, Telematics and Informatics", An Int. Journal on Telecommunication & Information Technology, Vol. 11, NO. 4, pp. 403- 428, 1994

[7] B. Stilman, " A Linguistic Geometry for Control System Design", International Journal of computers and their applications, Vol. 1 (2), pp. 89-110, Dec.1994

[8] B. Stilman, "What Is Linguistic Geometry?", Advances in Systems Science and Applications, Special Issue, pp. 154 – 163, Jun. 1997

[9] Tarokh, M; Hourtash, A, **"**Path planning by robot decomposition and parallel search**",** Robotics and Automation, 1997. Proceedings, 1997 IEEE International Conference on, Vol. 1 ,pp. 562 -568, 1997

[10] Kinugawa, K; Noborio, H., "A comparative study between on-line and off-line navigations on intelligent transportation system", Robot and Human Interactive Communication, 2000. RO-MAN 2000. Proceedings. 9th IEEE International Workshop on pp. 90 -95 ,2000