(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2012 (01.11.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/148013 A1

(51) International Patent Classification:
G06F 19/00 (2011.01) *G06F 17/10* (2006.01)

(21) International Application Number:
PCT/KR2011/003034

(22) International Filing Date:
26 April 2011 (26.04.2011)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **KOREA INSTITUTE OF SCIENCE AND TECHNOLOGY** [KR/KR]; 39-1, Hawolgok-dong, Seongbuk-gu, Seoul, 136-791 (KR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **HAQUE, Tareq, Rezaul** [BD/KR]; Cognitive Robotics Center, L8424, KIST, 39-1, Hawolgok-dong, Seongbuk-gu, Seoul 136-791 (KR). **KIM, Chang Hwan** [KR/KR]; #201-301, Sanggye Buram Daelim Apt., 1289, Sanggye-dong, Nowon-gu, Seoul 139-899 (KR).

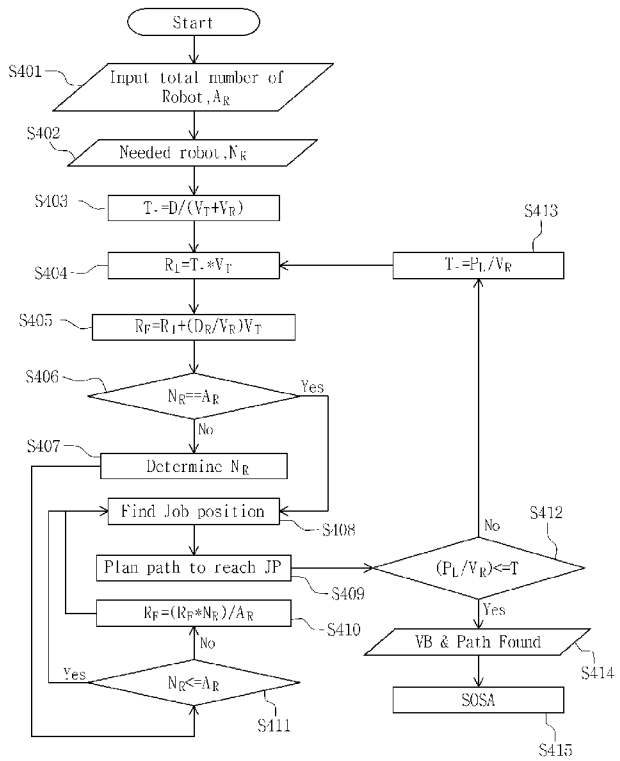
(74) Agent: **KIM, Sun-young**; Korea Coal Center, 10th Floor, 80-6, Susong-Dong, Chongro-Ku, Seoul 110-727 (KR).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[Continued on next page]

(54) Title: METHOD OF SEARCHING INVADER IN ROADMAP BASED ENVIRONMENT

[Fig. 4]



(57) Abstract: A method of searching invader in roadmap based environment includes determining virtual boundary which is defined based on the average velocity of searching robot, estimated invader's velocity, initial position of invader and environment constraint; along with virtual boundary the number of searching robots and their job positions on the edges of the virtual boundary are determined; Path planning for the searching robots to reach their job positions; and to search the virtual boundary appropriate job node is selected according to cost function till searching of all the nodes inside virtual boundary is over.

WO 2012/148013 A1

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

Description

Title of Invention: METHOD OF SEARCHING INVADER IN ROADMAP BASED ENVIRONMENT

Technical Field

[1] The present invention relates to multi-robots systems; and, more particularly, to co-operative search and path planning by multi-robots.

Background Art

[2] Recently, researchers have shown much interest in multi-robot systems and have frequently employed multi-robot teams to complete tasks that cannot be accomplished by a single robot. In cases of search and coverage algorithms, researchers have preferred multiple robots because of their robustness and quick completion of tasks. Various solutions have, thus, been proposed considering different environments, vehicles and resources.

[3] McGee and Hendrick derived optimal strategies to guarantee capture under certain conditions when the target velocity is assumed to be bounded and sensing occurs within a finite limit. But, the bounding strategy can only be ensured in circular obstacle-free environments (T. G. McGee and J. K. Hedrick, "Guaranteed strategies to search for mobile evaders in the plane," in Proceedings of the American Control Conference, Minneapolis, MN, 2006, pp. 2819 - 2824.). LaValle and Guibas developed pursuit-evasion strategies for multiple pursuers in polygonal environments (L. Guibas, J. Latombe, S. LaValle, D. Lin, and R. Motwani, "Visibility based pursuit-evasion in a polygonal environment," Intl Journal of Computational Geometry and Applications, 9(5):471494, 1999.). Their algorithm discretizes polygonal environments into conservative visibility regions and then uses an information space approach to develop complete algorithms that guarantee capture in simple environments. Gerkey et al. extended these methods to cases where the pursuer has a limited field-of-view (B. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," Intl Journal of Robotics Research, 25(4):299315, 2006.). For a single pursuer, these algorithms are guaranteed to find a solution if one exists. But in the presence of multiple pursuers, however, they lose this property. Additionally, these algorithms are difficult to extend to complex environments because of the total number of cells necessary in a conservative visibility discretization. These algorithms are also not applicable to complex environments in which capture cannot be guaranteed. Sarmiento et. al. present a framework for searching polygonal environments with multiple robotic pursuers using a one-step cost heuristic(Sarmiento, R. Murrieta-Cid, and S. Hutchinson, "A Multi-robot Strategy for Rapidly Searching a Polygonal En-

vironment," in Proc. 9th Ibero-American Conf. on AI, Puebla, Mexico, 2004.). But they did not extend this work to mobile evaders, and they did not present results in large-scale environments.

[4] T. Balch and R. Arkin presented the analysis of the effects of different kinds of communication on the performance of teams of mobile robots that performed tasks like searching for objects or covering a terrain (T. Balch and R. Arkin, "Communication in reactive multi-agent robotic systems," J. Autonomous Robot., vol. 1, no. 1, pp. 2752, 1994.). A family of robust multi-robot ant-based algorithms, which used approximate cellular decomposition, was presented by I. Wagner et. al (Wagner, M. Lindenbaum, A. Bruckstein, "Mac vs. pc determinism and randomness as complementary approaches to robotic exploration of continuous unknown domains," International Journal of Robotics Research 19 (1) (2000) 1231.). The algorithms are provably robust, but are not necessarily efficient: In contrast to the minimal capabilities of ant-robots (other than pheromone use), our algorithms make use of the robots' memory and communicate to carry out efficient path.

[5] In any search and coverage problem, most of the solutions are solved by decomposing the area into grid cells. This way of solution is applicable for an open space such as a forest and for sweeping purposes. But to provide security to some secure area like a housing society, a park, or an industrial farm, it cannot be thought as composed of grids because the whole area may not be reachable or some physical constraints may be present. For example, robots are only allowed to move through specified roads or more importantly, most of the times it is very difficult for wheeled mobile robots to move around in uneven terrain which imposes constraints on their mobility, and hence, we need a different solution. For these kind of scenarios, we propose the newly developed Simultaneous Online Search Algorithm (SOSA).

Disclosure of Invention

Technical Problem

[6] The present invention provides a method which defines a Virtual Boundary to bound the invader and big enough to search by the searching robots.

[7] Further the present invention provides Simultaneous Online Search Algorithm (SOSA).

[8] Further the present invention provides the method of searching invader to reduce redundancy.

Solution to Problem

[9] In accordance with the present invention, there is provided a method of searching invader in roadmap based environment includes determining virtual boundary which is defined based on the average velocity of searching robot, estimated invader's velocity,

initial position of invader and environment constraint; along with virtual boundary the number of searching robots and their job positions on the edges of the virtual boundary are determined;

[10] Path planning for the searching robots to reach their job positions; and to search the virtual boundary appropriate job node is selected according to cost function till searching of all the nodes inside virtual boundary is over.

Advantageous Effects of Invention

[11] In multi-robot system collision may occur. While using the present invention collision chance is less because searching robots always search for unsearched nodes or nodes those are not booked by other robots. This also minimizes the occurrence of overlap. And minimum occurrence of overlap will reduce the search time.

[12] The present invention ensures maximum use of all searching robots because it is an online search path planner. Distance between job nodes usually varies. In case of offline implementation of SOSA, if planner plans one job node for each searching robot in each iteration then path length of each searching robot will be different from each other. As a result some searching robot will stop searching before completion of searching entire virtual boundary. In the present invention when a searching robot reaches its job node then plan its next job node. And it continues planning till all the job nodes inside virtual boundary are searched.

[13] The present invention does not have any invader position to plan search path. The present invention always looks for unsearched or uncovered node. It looks for n next nodes in all reachable directions. After adding this cost with other costs the present invention determine next job node.

[14] The present invention does not depend on the shape of the virtual boundary. Because while planning for search path the present invention only consider the nodes inside the virtual boundary. So it does not matter what is the shape of the virtual boundary.

Brief Description of Drawings

[15] The objects and features of the present invention will become apparent from the following description of preferred embodiments given in conjunction with the accompanying drawings, in which

[16] Fig. 1 describes a diagram showing the environment representation denoting polygons (p), vertices (v), and nodes (n);

[17] Fig. 2 describes a diagram showing a scenario when the searching invader is first detected;

[18] Fig. 3 describes a diagram showing virtual boundary representation;

[19] Fig. 4 describes a flowchart showing how to determine virtual boundary and the number of searching robots;

[20] Fig. 5 describes a flowchart of simultaneous online search algorithm (SOSA);

[21] Fig. 6 describes a flowchart showing how to find minimum cost job node

[22] from a searching robot's current node;

[23] Fig.7 describes a diagram of reachable node representation that means from a node which other nodes searching robot can move;

[24] Fig.8 describes a diagram of tree view of reachable and booking nodes according to Fig.7.

Mode for the Invention

[25] The present invention now will be described more fully hereinafter with reference to the accompanying figures, in which embodiments of the invention are shown. This invention may, however, be embodied in many alternate forms and should not be construed as limited to the embodiments set forth herein.

[26] Accordingly, while the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that there is no intent to limit the invention to the particular forms disclosed, but on the contrary, the invention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the claims. Like numbers refer to like elements throughout the description of the figures.

[27] The present invention is described below with reference to block diagrams and/or flowchart illustrations of methods and/or display driver chips according to embodiments of the invention. It is understood that some blocks of the block diagrams and/of flowchart illustrations, and combinations of blocks in the block diagrams and/or flowchart illustrations, may be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, and/or other programmable data processing apparatus, create means for implementing the functions/acts specified in the block diagrams and/or flowchart block or blocks.

[28] These computer program instructions may also be stored in a computer readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instructions which implement the function/act specified in the block diagrams and/or flowchart block or blocks.

[29] The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-

implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions/acts specified in the block diagrams and/or flowchart block or blocks.

[30] As will be appreciated by one of skill in the art, the present invention may be embodied as a method, data processing system and/or computer program product. Thus, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects, which may be collectively referred to herein as a "circuit" or "module".

[31] It should also be noted that in some alternate implementations, the functions/acts noted in the blocks may occur out of the order noted in the flowcharts. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved.

[32]

[33] Environment representation

[34] Environment is represented by Graph G and G = (V, P) Where V is a set of r vertices and all the vertices v_i (i = 1 ~ r) will contain coordinate information. Roads are represented by polygon P_i (i = 1 ~ k)} where k is the number of polygons to represent the roads within the environment and $P_i = (v_1 \sim v_m)$. Here m is the number of vertices of i^{th} polygon. The polygon edges which are common to two polygons are portal edges and the midpoint of portal edge is denoted as a node. Nodes are actually used by a path planner. Another additional table is also generated while initialization i.e. reachability table. It contains the reference of all reachable nodes for each node.

[35] Referring to Fig. 1, P is the set of polygons, V is the set of vertices (white circles) and n is the set of nodes (red circles). Edge (v_3, v_4) is the common edge of polygon P_1 and P_2. So midpoint of edge (v_3, v_4) is n1 which is a node. Again edge (v_3, v_6) is the common edge of polygon P_2 and P_6. So n_2 is a node. This way all the nodes are determined. After finding nodes next job is to find reachable nodes from a node. So from Fig. 1 we can see for node n_1's reachable nodes are n_2 and n_3. Node n_2's reachable nodes are n_1, n_3 and n_7. This is the way to determine the reachable node for each node.

[36] Environment representation has big impact on our search method. So while designing new environment, we need to make sure there is enough node inside virtual boundary to cover entire area. Actually, a node's distance from its reachable node should not be more than the sensor coverage range.

[37]

[38] Defining virtual boundary

[39] Virtual boundary will confine the searching robot not to search the area with no chance of invader's presence. While determining virtual boundary, job positions and

the number of searching robots are also determined. Job positions are appropriate positions on the edge of virtual boundary for searching robots to start searching. Referring to Fig.4, Iterative process is employed to determine the virtual boundary and number of searching robots.

[40] Eq.(1) $T_* = D/(V_T + V_R)$

[41] Eq.(2) $R_I = T_* * V_T$

[42] Here, T_* is the time when searching robot meet the invader if they move towards each other. Referring to Fig.2, D is the straight distance between invader and the closest searching robot. Referring to Fig.3, R_I is the initial circular virtual boundary (ICVB) radius, V_R is the searching robot's average velocity and V_T is the invader's approximated velocity.

[43] The time the farthest searching robot takes to reach the closest intersection point between road and virtual boundary at the same time the distance traveled by the invader is R_I. Equations Eq. (1, S403) and Eq. (2, S404) are used to find R_I. Referring to Fig.3, Eq. (1) is only used to calculate T_* at the first iteration of the algorithm.

[44] Referring to Fig.6, final circular virtual boundary (FCVB) radius - If invader moves other than towards searching robot, then by the time searching robot reach virtual boundary, invader will be out of virtual boundary. So R_I needs to be increased to make sure invader is inside virtual boundary. To make sure boundness, distance travel by invader within the time searching robot takes to search virtual boundary need to add with R_I (S405).

[45] Eq.(3) $R_F = R_I + (D_R/V_R)V_T$

[46] Here, R_F is final circular virtual boundary radius. D_R is the straight distance from searching robot's job position to other end of virtual boundary. But for the first iteration while virtual boundary is not defined then $D_R = 2R_I$.

[47] Referring to Fig.3 virtual boundary (VB) is to be determined. Because the searching robots are only able to move through the roads, so it's better to think virtual boundary as a rectangular area, whose length and width is equal to $2R_F$. If circular virtual boundary is not fully inside the secure area then length and width of virtual boundary will be different. The number of searching robots (N_R) is determined based on the coverage of edge length. That means how many searching robots need to cover one edge of virtual boundary by its sensor.

[48] Eq.(4) $N_R = \lceil R_F/R_s \rceil$

[49] After determining the number of searching robots (N_R), if NR is greater than the total available number searching robots (A_R) then R_F needs to be increased to make the

boundary boundable for less number of searching robots to search.(S406, S407, S410, S411)

[50] Eq.(5) $$R_F = (R_F * N_R)/A_R$$

[51] Next problem is to choose the job position for searching robots. Job positions are chosen from a set of candidate job positions (S408). Candidate job positions are the positions where roads are intersected by the edges of virtual boundary. To choose job positions, searching robots are assigned priority based on distance from invader's detection point. The farthest searching robot is assigned the most priority. After finding job position A^* path finder is used to find the shortest path for searching robot's current position to its job position. Fig. 4 describes the whole process of determining virtual boundary and the number of searching robots.

[52] Eq.(6) $$(P_L/V_R) \leq T_*$$

[53] Eq.(7) $$T_* = P_L/V_R$$

[54] Here, P_L is the max path length among the paths from searching robot's current position to its job position and in Eq. (7), T_* is the time taken by searching robot to reach job position.

[55] When all the selected searching robots satisfy Eq. (6), they can reach their job positions and search virtual boundary. If Eq. (6) is unsatisfied then Eq. (7) is used to set new time limit to define inner circular virtual boundary radius (R_I). And then based on new R_I and previous iteration virtual boundary information we will again define new virtual boundary and find job position (S412, S413).

[56] In any case, if the number of searching robots becomes more than the total number of searching robots then the number of searching robots will be fixed to the total number of searching robots. And the algorithm will increase the size of virtual boundary to ensure boundless for the available searching robots. After using all the available searching robot, if Eq. (6) is not satisfied then entire area will be virtual boundary. After reaching job position, SOSA plans the path to search virtual boundary.

[57]

[58] Simultaneous online search Algorithm (SOSA)

[59] SOSA is a search path planner which ensures effective and minimum redundant search path in roadmap based environment. SOSA's job is to search all the nodes inside virtual boundary. While search planning a searching robot chooses its most feasible reachable node as job node and book n nodes as the future job node. The node which can be the job node after a searching robot reaches its current job node is a future job node. And when the next searching robot will look for its job node, it will consider other searching robots' job nodes and booking nodes.

[60] Fig. 5 shows the flow chart of SOSA. In Fig.5, nRobot is the number of searching robot, i is the index of the searching robot and j is the index of reachable node from a searching robots current node. All the searching robots will move towards their job nodes, their first job is to search for invader. If invader is detected, search is completed. If invader is not detected, SOSA will check for each searching robot has reached its job nodes If not reached then searching robot index will increase and process will continue with next searching robot. And if a searching robot reaches its job node, SOSA will check if the search of entire virtual boundary has completed or not. If search completed without invader detection then it's an unsuccessful search. The term 'Search completed' means all the nodes inside virtual boundary should be searched. If search is not completed then final cost function will be called for all the reachable nodes of a searching robot's current job node and the minimum cost node will be chosen as next job node. This process will continue till all the nodes are searched.

[61] SOSA's basic idea is to find out the minimum cost node of all reachable nodes from the current job node. From these reachable nodes, the node with minimum cost is considered as the next job node. Fig. 6 presents the whole process of finding the minimum cost job node. The final cost function to determine a reachable node's cost is described below

[62] Eq.(8) $$T = R_c + \alpha C_c + \beta N_c + K_c$$

[63] Eq.(9) $$R_c < L_c$$

[64] Eq.(10) $$C_c = 2L_c$$

[65] Eq.(11) $$C_c + L_c + nr < N_c$$

[66] Here, T is the total cost of a reachable node to choose (S603). R_c is the cost for a searching robot to take turn to move to next reachable node and its maximum cost is π. L_c is an integer constant, a node returns L_c if it is booked by some searching robot. C_c is an integer constant, a node returns C_c if the node already considered as job node or it can be searched from a job node. A reachable node returns N_c if it is outside of virtual boundary and it's an integer constant. n is the number of job nodes to book by a searching robot and r is number of searching robots. From eqs. (9) ~ (11), we can see R_c returns the minimum value among cost terms and N_c returns the maximum value to influence the searching robots to search only inside virtual boundary. L_c and C_c are greater than R_c because searching robot should look for unsearched and not booked node. R_c is only useful when there is more than one node has same cost. C_c is greater than L_c because no need to search already searched node.

[67] α is 1 if the reachable node is searched by the searching robot, 2 if the reachable node is the previous job node of the searching robot, 3 if the reachable node is the previous job node of the searching robot, 0 otherwise.

[68] β is 1 if the reachable node within sensor range from the job node and if anyone of its reachable node is outside of the virtual boundary, 2 if the reachable node is outside, 0 otherwise.

[69] According to Fig. 7 when a searching robot's job node is A and reachable node is D then β is 1 because one reachable node R of D is outside of virtual boundary. If D is a job node and it's reachable node R is out of virtual boundary so in this case β is 2. When A is a job node then it has three reachable nodes B, C and D. For these three reachable nodes, the final cost will be calculated and the minimum cost node will be chosen as the next job node.

[70] Now, we discuss about fourth term (K_c) of the final cost function. K_c books n future job node and returns the cost to book n future job nodes. K_c's job is to go forward and collect the status of the nodes available in different direction (i.e. different reachable node). And calculate the cost for choosing a direction. All the cost will be added with final cost function (T). And reachable node with the minimum final cost is the next job node. Now we will see how K_c works. K_c uses B^* to book n future job node.

[71] Eq.(12) $$B^* = R_C + \alpha C_C + \beta N_C + \gamma$$

[72] Eq.(13) $$\gamma = \eta C_C + \lambda L_C$$

[73] Here, B^* is the cost of a node to book while booking n future job nodes. γ returns the cost of choosing node which is already booked by itself. η is 1 when a chosen node is already booked itself else 0. And λ is 1 when it is within the sensor range from a node chosen by itself else 0. Three functions (C_1, C_2 and C_3) are used to compute the cost to book n nodes for a reachable node.

[74] Eq.(14) $$C_1 = \sum_{i=1}^{n} C_c x_i \quad for\ x_i = \begin{cases} 1 & if\ b_i\ is\ searched \\ 0 & otherwise \end{cases}$$

[75] Here, C_1 is the cost to book searched node. b is the booking list and b_i is the i^{th} booking node for a reachable node.

[76] Eq.(15) $$C_2 = \sum_{j=1}^{r} \sum_{i=1}^{n} (L_c + n - i + 1) y_i$$

[77] Here, C_2 is the cost to book nodes which are already booked node by other searching robots and cost will more to book other searching robot's booking list's front node than rear node. If b_i is booked but not by the current searching robot then y_i is 1 else 0.

[78] Eq.(16)

$$C_3 = \sum_{i=1}^{n} N_c z_i$$

[79] Here, C_3 is the cost to book nodes out of virtual boundary. If b_i is outside of virtual boundary then z_i is 1 otherwise 0.

[80] So $K_c = (C_1 + C_2 + C_3)$ is the cost from a reachable node to book n future job nodes. Here we want to give an example of term Kc according to Figs. 7 and 8. If a searching robot has reached its job node (A) then it will look for its next job node. For this example let n = 3. Node A has 3 reachable nodes (B, C and D). Now in B direction we will determine booking node using Eq. (12). For 1st booking reachable nodes of B are A, E, C and D. Among these four node E is the best choice because A is previous job node, D has a reachable node (R) outside VB and between E and C, to reach E searching robot does not require any angular velocity. For 2nd booking E's reachable nodes are B, I, H and G and here I will be the minimum cost reachable node from E. For 3rd booking, I's reachable nodes are H, J, E and G and here J is the best choice. So for node B chosen nodes to book are E, I, J. In the same way, for node C chosen nodes are Q, N, M. Now, using Eqs.(14) and (16) K_c will be calculated for booked nodes and will be added to final cost function of B, C and D. Among B, C and D which return the minimum will be considered as future job node. If C is the node with minimum cost node then BookingList (b) will be updated with the node Q, N, M for the current searching robot.

[81] While the invention has been shown and described with respect to the preferred embodiments, it will be understood by those skilled in the art that changes and modification may be made without departing from the spirit and scope of the invention as defined in the following claims.

Industrial Applicability

[82] "Patrolling and Searching" by human are tedious, hard and sometimes hazardous. These problems can cause unsecure situations in secure area. So an alternative approach is to use technology to assist in these tasks through the use of multiple mobile robots to guard the grounds of secure area from intrusion and also search for intrusive presence if detected inside secure area.

[83] But algorithm is necessary to control multiple mobile robots, so we developed Simultaneous Online Search Algorithm (SOSA) to control multiple mobile robots in searching and patrolling.

[84] So, there is enormous applicability for our algorithm to be implemented in real environments like housing society, industrial area, museum, and government's important places.

Claims

[Claim 1] A method of searching invader in roadmap based environment includes determining virtual boundary which is defined based on the average velocity of searching robot, estimated invader's velocity, initial position of invader and environment constraint; along with virtual boundary the number of searching robots and their job positions on the edges of the virtual boundary are determined;
Path planning for the searching robots to reach their job positions; and
Searching the virtual boundary appropriate job node is selected according to cost function till searching of all the nodes inside virtual boundary is over.

[Claim 2] The method of claim 1, further comprising
determining the number of searching robots based on the virtual boundary.

[Claim 3] The method of claim 1, wherein determining virtual boundary includes
Driving T_*, which satisfies the following relationship:

$$T_* = D/(V_T + V_R)$$

wherein T_* denotes the time when the searching robot meets the invader, D denotes straight distance between invader and the closest searching robot, V_T denotes the invader's approximated velocity and V_R denotes the searching robot's average velocity;
Driving R_I, which satisfies the following relationship:

$$R_I = T_* * V_T$$

Wherein R_I denotes the initial circular virtual boundary; and
Driving R_F, which satisfies the following relationship:

$$R_F = R_I + (D_R/V_R)V_T$$

Wherein R_F denotes final circular virtual boundary radius, D_R denotes the straight distance from the searching robot's job position to other end of virtual boundary.

[Claim 4] The method of claim 2, wherein determining the number of searching robots includes
driving N_R, which satisfies the following relationship:

$$N_R = \lceil R_F/R_s \rceil$$

Wherein N_R denotes the number of the searching robots, R_s denotes the

radius of sensor in the searching robot.

[Claim 5] The method of claim 4, after the driving New N_R, further comprising Driving New R_F if $N_R > A_R$, which satisfies the following relationship:

$$\text{New } R_F = (R_F * N_R)/A_R$$

Wherein A_R is the number of available searching robot.

[Claim 6] The method of claim 5, after the driving New R_F, further comprising Driving New T_* if $P_L/V_R > T_*$, which satisfies the following relationship:

$$\text{New } T_* = P_L/V_R$$

[Claim 7] The method of claim 1, searching the virtual boundary appropriate job node includes

driving the cost function T, which satisfies the following relationship:

$$T = R_c + \alpha C_c + \beta N_c + K_c$$

wherein T denotes the total cost of a reachable node to choose, R_c denotes the cost for a searching robot to take turn to move to next reachable node and its maximum cost is π, C_c denotes an integer constant, a reachable node returns C_c if the node already considered as job node or it can be searched from a job node, a reachable node returns N_c if it is outside of the virtual boundary and it's an integer constant, K_c returns the cost to book n future job nodes,

whrerin α is 1 if the reachable node is searched by the searching robot, 2 if the reachable node is the previous job node of the searching robot, 3 if the reachable node is the previous job node of the searching robot, 0 otherwise,

wherein β is 1 if the reachable node within sensor range from the job node and if anyone of its reachable node is outside of the virtual boundary, 2 if the reachable node is outside, 0 otherwise.

[Claim 8] The method of claim 7, wherein the driving the cost function T satisfies the following relationship:

$$R_c < L_c$$

Wherein L_c is an integer constant and the reachable node returns L_c if it is booked by some robot.

[Claim 9] The method of claim 8, wherein the driving the cost function T satisfies the following relationship:

$$C_c = 2L_c$$

[Claim 10] The method of claim 8, wherein the driving the cost function T satisfies

the following relationship:

$$C_c + L_c + nr < N_c$$

[Claim 11] The method of claim 8, wherein the driving the cost function T further comprising Driving B^* for each booking position, which satisfies the following relationships:

$$B^* = R_C + \alpha C_C + \beta N_C + \gamma \quad ;$$

$$\gamma = \eta C_C + \lambda L_C$$

Wherein γ returns the cost of the booking node which is already booked by itself, η is 1 when the booking node is already booked itself else 0, and λ is 1 when it is within the sensor range from the booking node chosen by itself else 0.

[Claim 12] The method of claim 11, wherein driving K_c by using the minimum B^* of all the booking nodes, which satisfies the following relationship:

$K_c = (C_1 + C_2 + C_3)$

Wherein C_1 is the cost to book searched node, C_2 is the cost to book nodes which are already booked by other robots and cost will more to book other robot's booking list's front node than rear node, and C_3 is the cost to book nodes out of virtual boundary.

[Claim 13] The method of claim 12, wherein driving K_c, which satisfies the following relationships:

$$C_1 = \sum_{i=1}^{n} C_c x_i \quad for\ x_i = \begin{cases} 1 & if\ b_i\ is\ searched \\ 0 & otherwise \end{cases}$$

Wherein b is the booking list and b_i is the ith booking node for the reachable node,

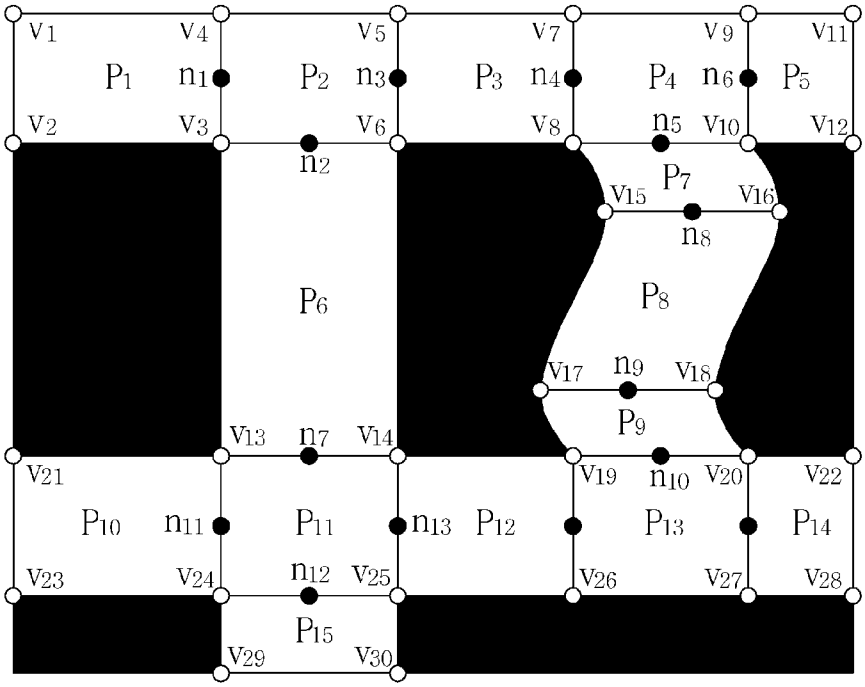
$$C_2 = \sum_{j=1}^{r} \sum_{i=1}^{n} (L_c + n - i + 1) y_i$$

Wherein If b_i is booked but not by the searching robot then y_i is 1 else 0

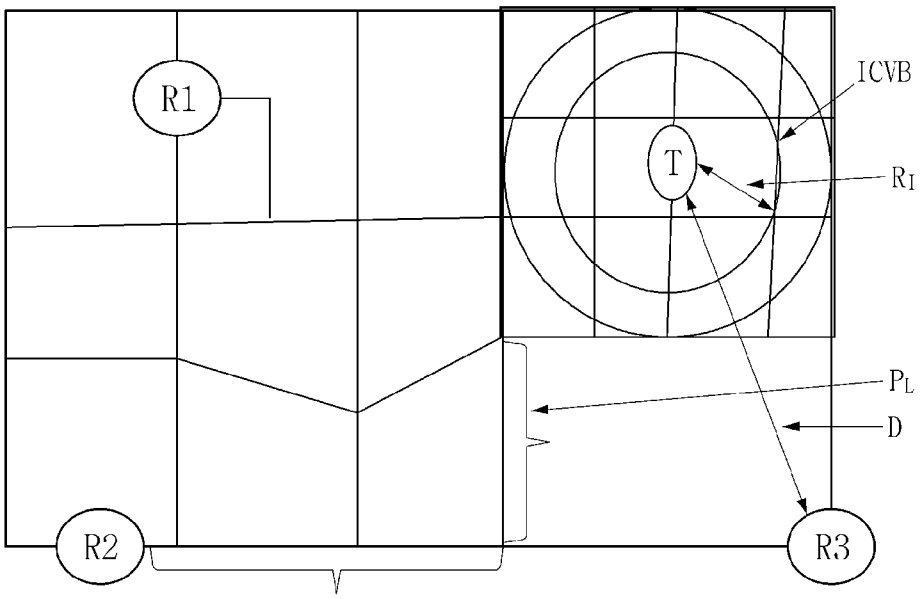
$$C_3 = \sum_{i=1}^{n} N_c z_i$$

Wherein b_i is outside of virtual boundary then z_i is 1 otherwise 0.

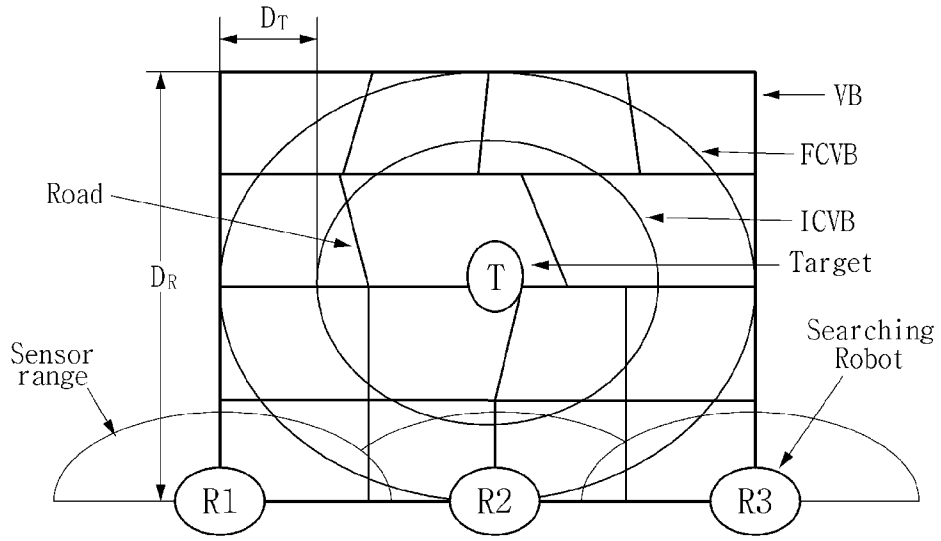
[Fig. 1]


V1
V4
V5
V7
V9
V11
P1
n1
P2
n3
P3
n4
P4
n6
P5
V2
V3
V6
V8
n5
V10
V12
n2
P7
V15
V16
n8
P6
P8
V17
n9
V18
P9
V13
n7
V14
V21
V19
n10
V20
V22
P10
n11
P11
n13
P12
P13
P14
V23
V24
n12
V25
V26
V27
V28
P15
V29
V30

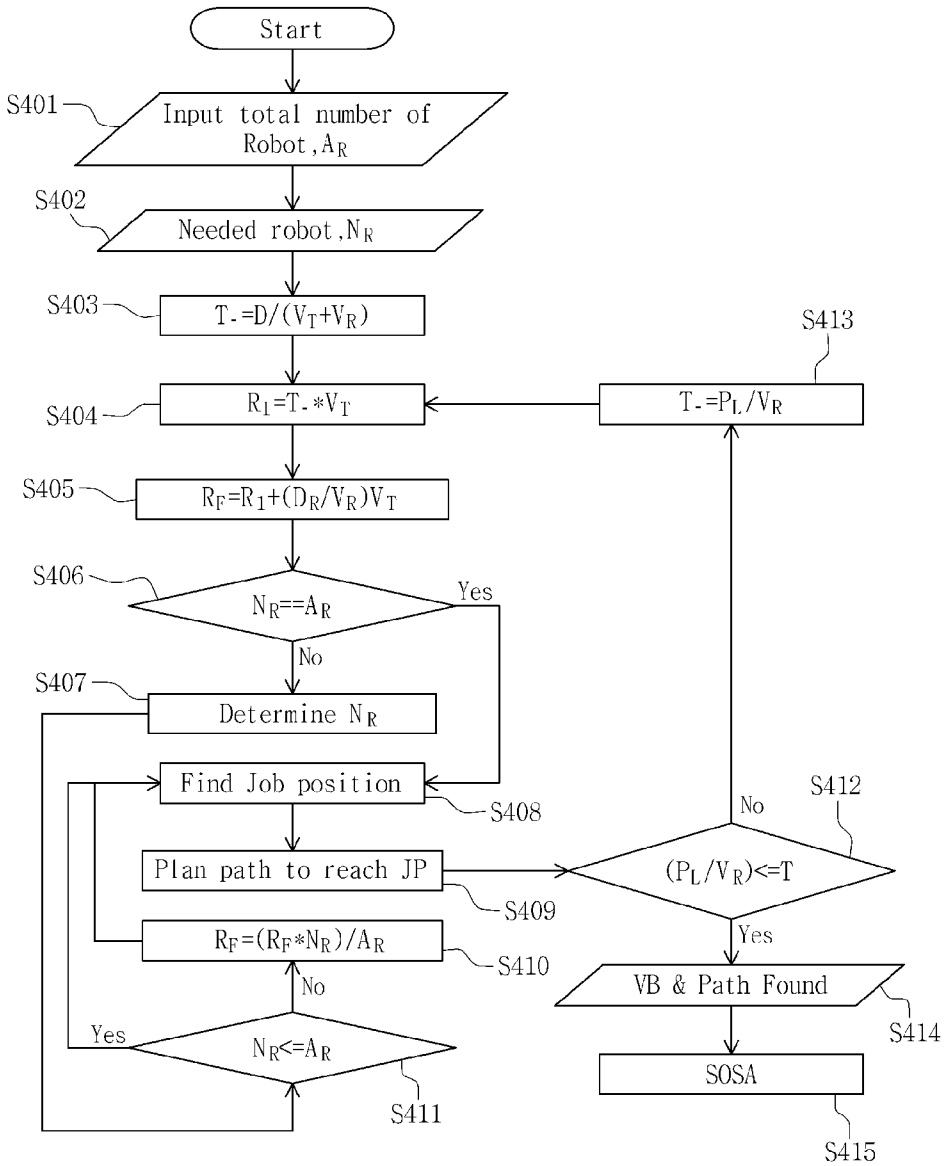
[Fig. 2]


R1
ICVB
T
RI
PL
D
R2
R3

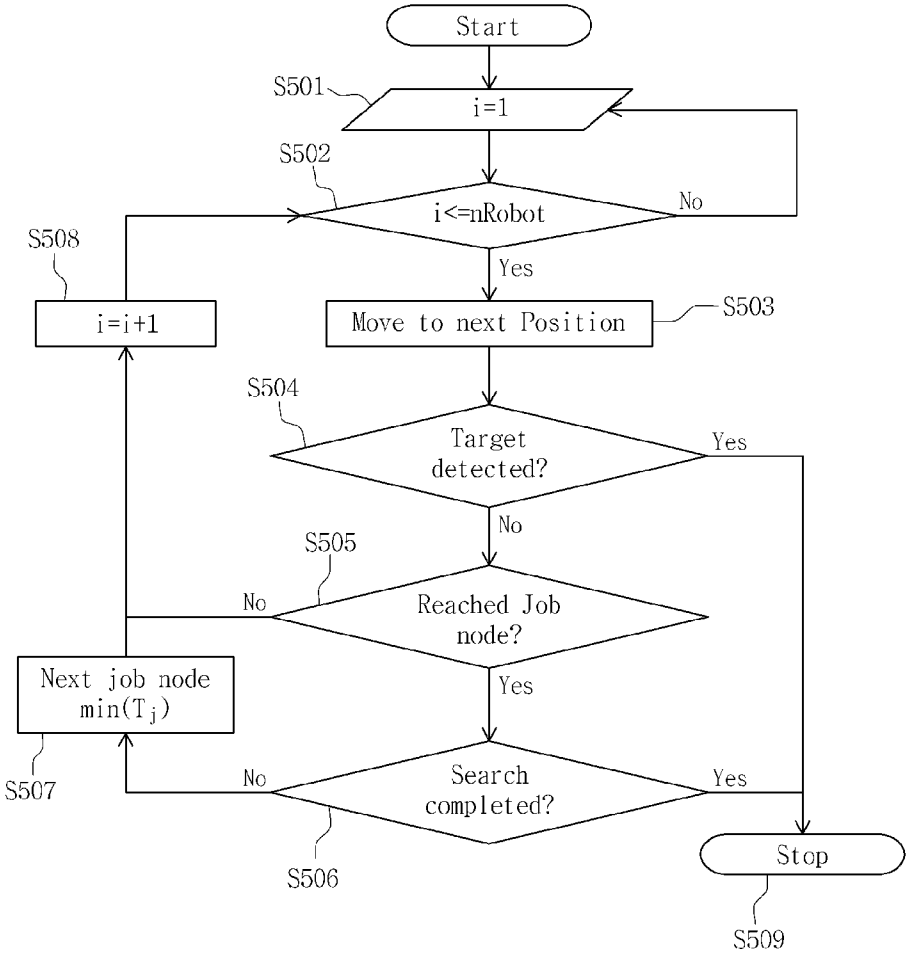
[Fig. 3]


D_T
VB
FCVB
ICVB
Target
Road
D_R
T
Sensor range
Searching Robot
R1
R2
R3

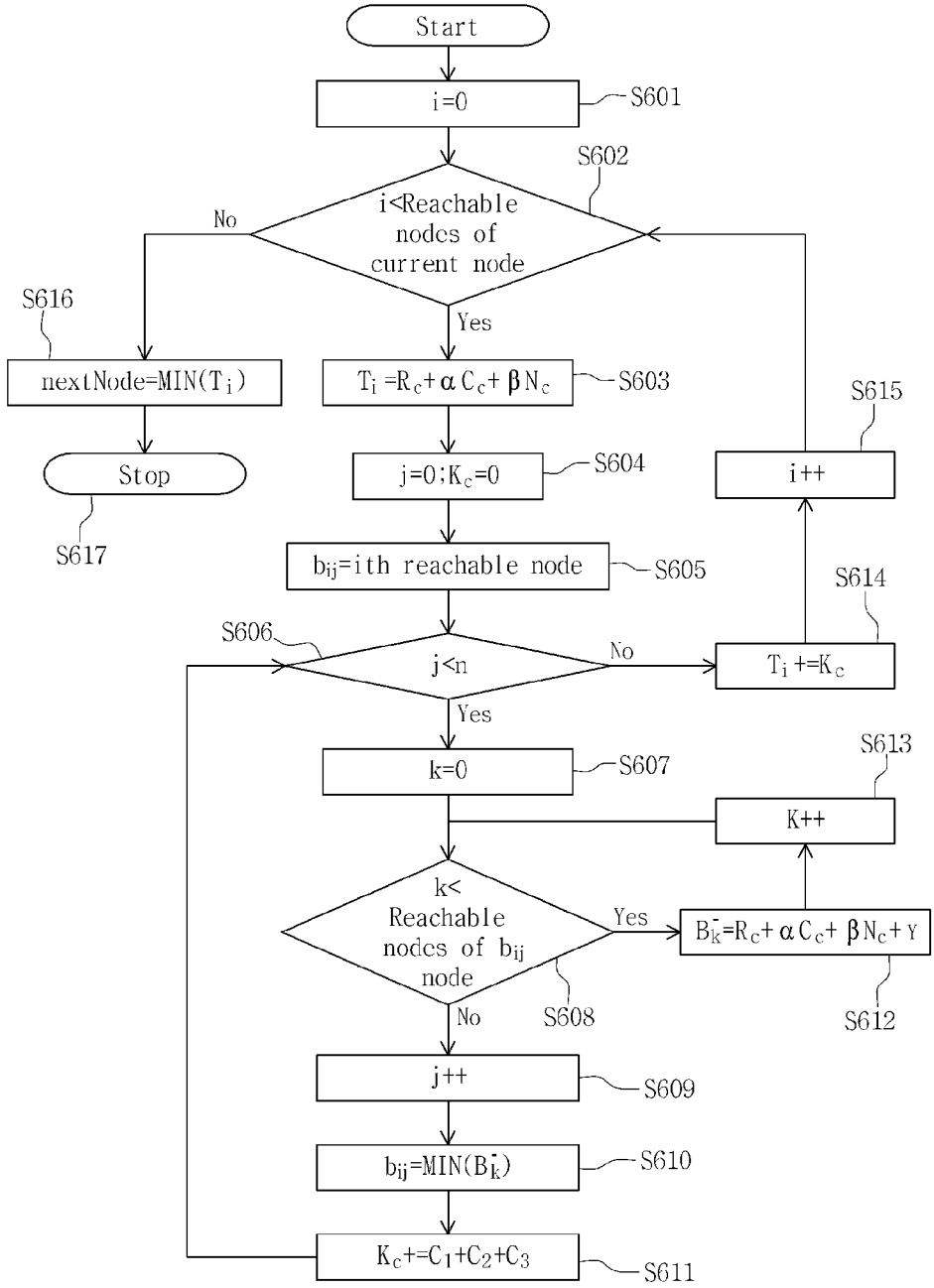
[Fig. 4]


Start
S401
Input total number of Robot, A_R
S402
Needed robot, N_R
S403
$T_-=D/(V_T+V_R)$
S413
S404
$R_1=T_-*V_T$
$T_-=P_L/V_R$
S405
$R_F=R_1+(D_R/V_R)V_T$
S406
$N_R==A_R$
Yes
No
S407
Determine N_R
Find Job position
S408
S412
No
Plan path to reach JP
$(P_L/V_R)<=T$
S409
Yes
$R_F=(R_F*N_R)/A_R$
S410
VB & Path Found
S414
No
Yes
$N_R<=A_R$
SOSA
S411
S415

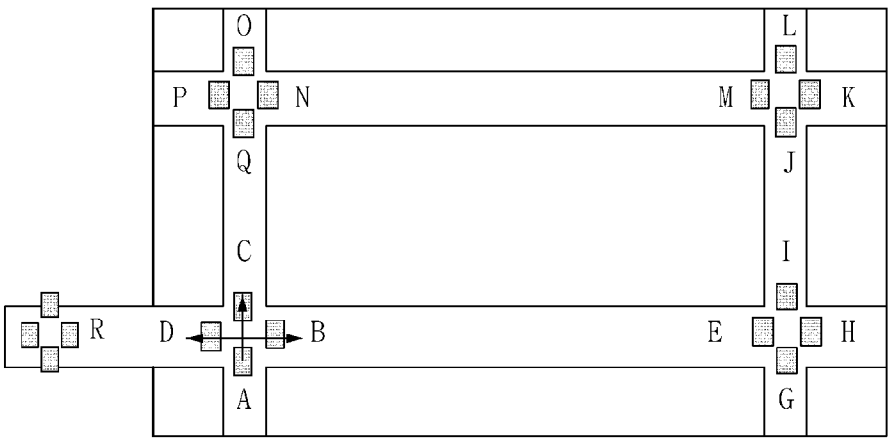
[Fig. 5]


Start
S501
i=1
S502
i<=nRobot
No
Yes
S508
i=i+1
Move to next Position
S503
S504
Target detected?
Yes
No
S505
No
Reached Job node?
Yes
Next job node min(T_j)
S507
No
Search completed?
Yes
S506
Stop
S509

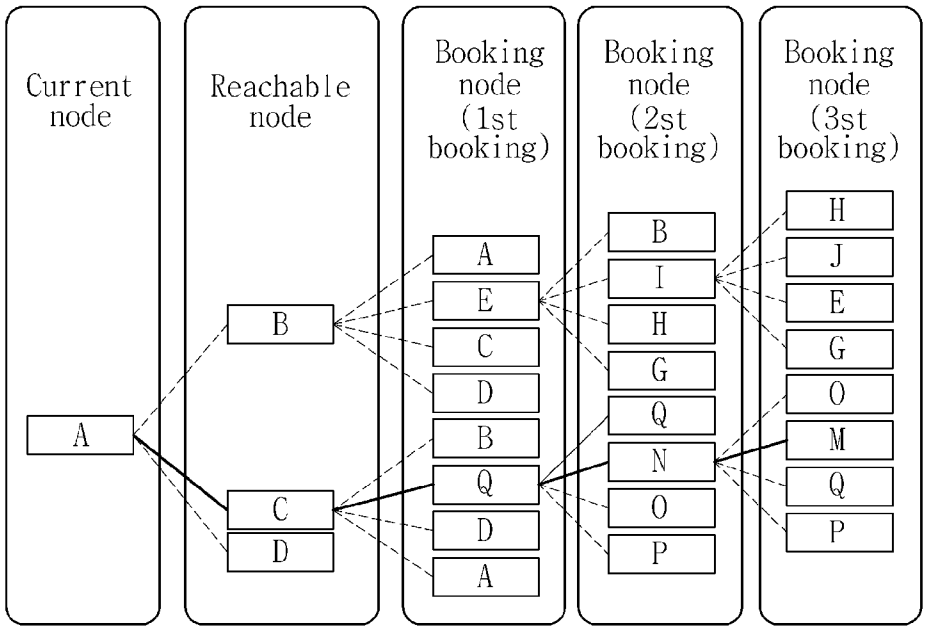
[Fig. 6]


Start
i=0
S601
S602
i<Reachable nodes of current node
No
Yes
S616
nextNode=MIN(Ti)
Ti =Rc+αCc+βNc
S603
S615
Stop
j=0;Kc=0
S604
i++
S617
bij=ith reachable node
S605
S614
S606
j<n
No
Ti +=Kc
Yes
k=0
S607
S613
K++
k< Reachable nodes of bij node
Yes
Bk=Rc+αCc+βNc+γ
No
S608
S612
j++
S609
bij=MIN(Bk)
S610
Kc+=C1+C2+C3
S611

[Fig. 7]


O
L
P
N
M
K
Q
J
C
I
R
D
B
E
H
A
G

[Fig. 8]


Current node
Reachable node
Booking node (1st booking)
Booking node (2st booking)
Booking node (3st booking)
A
B
C
D
A
E
C
D
B
Q
D
A
B
I
H
G
Q
N
O
P
H
J
E
G
O
M
Q
P

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2011/003034

A. CLASSIFICATION OF SUBJECT MATTER

G06F 19/00(2011.01)i, G06F 17/10(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC : G06F; B25J; G08B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: search, invader, robot, grid

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007-0179669 A1 (JAMES MCLURKIN et al.) 02 August 2007 See abstract, figures 1A, 2, 3, 4, paragraphs [0031]-[0061], [0065]-[0111], claim 1.	1-13
A	US 2004-0024490 A1 (JAMES MCLURKIN et al.) 05 February 2004 See figures 1A, 2, 3, 4, paragraphs [0031]-[0061], [0065]-[0112], claim 1.	1-13
A	KR 10-0155899 B1 (SAMSUNG ELECTRONICS INC.) 15 December 1998 See abstract, figures 3-5, claim 1.	1-13
A	JP 2009-295140 A (NATIONAL CHIAO TUNG UNIV.) 17 December 2009 See abstract, paragraph [0039], claims 1, 10.	1-13
A	`SLAM of a Mobile Robot Using IR sensor and Vision sensor` (YONG-JU LEE et al.) 30 November 2006 See abstract, figures 7-10.	1-13

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 DECEMBER 2011 (28.12.2011)	**29 DECEMBER 2011 (29.12.2011)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Cheongsa-ro,
Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KIM, Dong Sung

Telephone No. 82-42-481-8519

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2011/003034

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0179669 A1	02.08.2007	US 2004-0024490 A1	05.02.2004
		US 2006-0079997 A1	13.04.2006
		US 2011-0035052 A1	10.02.2011
		US 7117067 B2	03.10.2006
		US 7254464 B1	07.08.2007
		US 7844364 B2	30.11.2010
US 2004-0024490 A1	05.02.2004	US 2006-0079997 A1	13.04.2006
		US 2007-0179669 A1	02.08.2007
		US 2011-0035052 A1	10.02.2011
		US 7117067 B2	03.10.2006
		US 7254464 B1	07.08.2007
		US 7844364 B2	30.11.2010
KR 10-0155899 B1	15.12.1998	None	
JP 2009-295140 A	17.12.2009	US 2009-0303042 A1	10.12.2009