

Research Article

SPAM for a Manipulator by Best Next Move in Unknown Environments

Dugan Um and Dongseok Ryu

Department of Engineering and Computing Sciences, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA

Correspondence should be addressed to Dugan Um; dugan.um@tamucc.edu

Received 15 January 2013; Accepted 4 February 2013

Academic Editors: Z. Bi and G. C. Gini

Copyright © 2013 D. Um and D. Ryu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a SPAM (simultaneous planning and mapping) technique for a manipulator-type robot working in an uncertain environment via a novel Best Next Move algorithm. Demands for a smart decision to move a manipulator such as humanoid arms in uncertain or crowded environments call for a simultaneous planning and mapping technique. In the present work, we focus more on rapid map generation rather than global path search to reduce ignorance level of a given environment. The motivation is that the global path quality will be improved as the ignorance level of the unknown environment decreases. The 3D sensor introduced in the previous work has been improved for better mapping capability and the real-time rehearsal idea is used for c -space cloud point generation. Captured cloud points by 3D sensors, then, create an instantaneous c -space map whereby the Best Next Move algorithm directs the local motion of the manipulator. The direction of the Best Next Move utilizes the gradient of the density distribution of the k -nearest-neighborhood sets in c -space. It has a tendency of traveling along the direction by which the point clouds spread in space, thus rendering faster mapping of c -space obstacles possible. The proposed algorithm is compared with a sensor-based algorithm such as sensor-based RRT for performance comparison. Performance measures, such as mapping efficiency, search time, and total number of c -space point clouds, are reported as well. Possible applications include semiautonomous telerobotics planning and humanoid arm path planning.

1. Introduction

Manipulator motion planning in unknown environments is a challenging task in path planning study. For a manipulator path planning with a complete environmental model, the notion of completeness and optimality often becomes a main issue of discussion. For unknown environments, however, optimal path finding becomes difficult due to the uncertainty in environments. The focus of question lies rather in the probabilistic convergence to the goal point or planning in difficult areas to deal with high level of uncertainty. From the standpoint of implementation, however, both sides rigorously use configuration space approach to benefit from mathematical advantages. Sensor-based motion planning is the most common approach when it comes to unknown environment planning problems. Among many attempts in this group is the Lumelsky, [1–3] which demonstrated an algorithm that guarantees convergence if a path exists or terminates in a limited time for up to 3 DOF Cartesian robot. Monotonicity

plays an important role in the proof of convergence. Various heuristic strategies have been introduced in unknown environment planning, while Voronoi graph search or polygon manipulation for a closed form solution have been studied in the model based planning [4]. Among many in the latter is the Schwartz and Sharir's work [5] to obtain a closed-form path planning solution in 3-DOF workspace. For simplicity, polygon models are used as a real-world representation. The complexity of the exact solution, however, becomes $O(n^{2^d + b})$ when n is the number of obstacle edges and d is the number of degrees of freedom. Due to the exponential increase in complexity as the number of obstacles grows, it only served as a proof of existence of the solution. As a result, manipulator motion planning in higher-order DOF is known as either PSAPCE-hard or NEXSPACE-hard, and only-probabilistic-sampling based approaches demonstrated success for path search and convergence. Among those that are successful is the random preprocessing [6], which paved a road for a fast and practical path solution in higher DOF

manipulator planning problems. In random preprocessing, the preprocessor distributes random seeds in free c -space followed by a query session to check connectivity between each seed. Only a few algorithms especially with probabilistic-sampling-based approaches are reported successful for more than 3 DOF revolutionary joint robots due to the level of complexity in unknown environments. However, for a planning with a complete model, discovering a shortest path or an optimal path has been the primary subject of study. For example, optimal path planning strategies of a mobile robot or a group of robots have been studied in [7, 8]. In addition, in [9], an optimal configuration of dynamically moving vehicles to maintain connectivity has been tackled.

However, due to the limited sensing range or visual occlusion caused by today's sensing devices, an optimal path planning for a manipulator in unknown environments is either hindered or limited. Therefore, trends in this area are moving toward a rapid map generation strategy in local motion and, thus, achieve optimality in global path search with better environmental information. That is, if a planner can produce a global map rapidly with an appropriate local strategy, an optimal path planning would be more attainable in unknown environments. Therefore, we focus, in this paper, on an expeditious mapping strategy of an unknown environment rather than on a global path search strategy alone.

For mobile robot navigations in unknown environments, SLAM (simultaneous localization and mapping) is a well-known approach for path planning. In SLAM, mobile robot tracking by probabilistic approaches is the point of study. Kalman filter has been proved to be a useful tool in SLAM study. Kalman gain optimization by minimizing the error in the a posteriori state estimation has been largely studied. The most common form is the EKF by which a nonlinear system estimation becomes feasible [10]. When it comes to a probabilistic path planning for a manipulator-type robot, localization is not an issue, but mapping is still important for planning purpose since planning can only take place as the ignorance level of the environment reduces over time. Sensor-based motion planning is the main stream in unknown environment planning studies. Sequential mapping of a local area followed by a local path planning is a natural step in sensor-based motion planning algorithm. In [11], a novel framework for an unknown environment path planning of manipulator type robots is proposed. The framework proposed is a sensor-based planner composed of a sequence of multiple MBPs (model based planners) in the notion of cognitive planning using real-time rehearsal.

In [12], c -space entropy is examined for planning and exploration to guide a robot with an eye-in-hand sensor system. Natural planning and expanding steps of the c -space are repeated in the paper for an unknown environment path planning. However, the eye-in-hand sensor has limitations in reporting collisions or c -space mapping capability in real-time due to visual occlusion. Other studies on sensor based planning of manipulators include [13], where an avoidance ability for redundant manipulators is studied with a moving camera on the ceiling. Trajectory planning for a redundant

mobile manipulator is studied using avoidance manipulability concept [14]. In the paper, manipulability is the point of study for collision avoidance to select the best path from a multitude available configurations depending on the singularity and manipulability ellipsoid. A kinematic analysis of local or global motion planning for manipulators has been also studied in the notion of singularity for redundant robots as well [15, 16]. Topological analysis in conjunction with a singularity concern for a redundant manipulator deals with the study on critical point surfaces in c -space. The study implies that a manipulator has to stay in a continuous c -sheet to avoid a singularity, which means that a motion planning with inverse kinematic concern is neither robust nor efficient due to the singularity problem as well as limited utilization of a given c -space.

As a result, in a real-time manipulator motion planning especially for unknown environments, the majority of the manipulator planning schemes we investigated are either hindered by sensor configurations or by motion constraints due to singularity problem. That is, inverse kinematics, if incorporated in the process of motion planning, renders the overall planner unstable or less robust. A more flexible sensor configuration for maximum sensibility is, therefore, in essential need.

To that end, we proposed a skin-type sensor made out of a camera with 3D depth sensing capability [17]. Compared to the previous work, the sensor quality in depth measurement has been improved [18] and the real-time rehearsal idea is used in this work for c -space cloud point generation. However, the effort to advance the previous work is focused primarily on rapid map generation via simultaneous planning and mapping (SPAM) capability. For a rapid map building and path planning, we use the skin-type sensors that completely encompass the entire body of a manipulator. Such sensor can generate real-time point clouds of obstacles, thereby making a real-time local c -space construction feasible. However, an appropriate guidance algorithm that extends the capability of the skin-type sensor for rapid map generation is an essential need. To that end, we use a 3D point cloud registration method as a possible guidance algorithm for manipulator's local motion.

3D point cloud registration calls for various descriptors for object recognition [19]. We take an advantage of the benefit of such registration processes as a guidance method of a manipulator in a partially constructed c -space environment. Amongst many registration methods, is the group averaging features (GAFs), an invariant feature for registration [20], in which a gradient of the density distribution is used to log essential features for point cloud registration.

We discuss the rationale of why and how we utilize the steps for the invariant feature extraction method for sensor-based motion planning in Section 2. We discuss the results of comparison of the proposed algorithm with other sensor-based planning algorithms as well. Performance indices used for comparison include mapping efficiency, search time, and total number of c -space point clouds. Improved performances are reported for the proposed algorithm.

2. Best Next Move Planner

Due to the higher-order complexity of a manipulator-type robot path planning problem, a model-based probabilistic sampling approach is the most common in planning society. Several sampling-based path planners are reported to be complete so that they either find a path if one exists or terminate otherwise in a limited time. Manipulator path planning in unknown environments, however, is challenging in that neither the path optimality nor the planner's completeness can be guaranteed. Gradual but rapid construction of a c -space map, if feasible, seems to be the only promising strategy in unknown environments.

C -space mapping especially in a crowded environment is the most daunting task in manipulator path planning though. In [21], Best Next View in conjunction with a sensor-based roadmap planner is used for object recognition. Object recognition by BNV is done by the concept of detecting key events in the set of range data such as discontinuity of the range data of the scene. This key event is used to drive the local motion of the manipulator to reduce the ignorance level of the given space. Similarly, we propose the Best Next Move algorithm as a guidance of the robot's local motion in uncertain environments.

Our definition of the Best Next Move in the notion of manipulator motion planning is the move of a point automaton along the direction that enables further collection of environmental information rather than finding a path to the goal. The motivation is that the more complete c -space map a planner can generate, the better the chance for convergence to the goal. Since the 3D point cloud registration method calls for rapid point cloud identification and collection of a 3D shape, it leads to the rapid mapping of a workspace environment, thus enabling the Best Next Move in each step of the motion planning.

When it comes to an unknown environment manipulator motion planning, two subjects have to be addressed in parallel: map construction and navigation for convergence. The objective of each subject is set as shown below:

- (1) map construction: local navigation for maximum mapping capability.
- (2) goal convergence: probabilistic search completeness.

For the first task, we propose a combinatory motion planning of sampling-based search and the point cloud registration inspired approach to determine the Best Next Move (BNM) of the local motion. In this paper, we define the BNM as a direction that enables obtaining maximum geometric information of c -space obstacles, thus maximizing the mapping capability. Group Average Feature method, one of the 3D point cloud registration methods provides a strategy to navigate around a set of point cloud to register the uniqueness of a 3D object, thus providing an effective object recognition. We generate the BNM path for a local motion using the navigation strategy of the GAF method. To that end, first, we collect point cloud data by 3D depth sensors attached on the manipulator as shown in Figure 1. Second, we generate c -space point clouds by real-time rehearsal using a model-based planning algorithm. Finally, we apply the GAF

navigation strategy to determine the BNM path for a local motion.

For the first step, a skin-type sensor (see Figure 2(a)) is developed and tested for its usefulness to verify the proposed planner. The sensor developed is a novel short-range 3D depth sensor based on multiple intensity differentiation method [18]. The sensor is able to simultaneously calculate 3D depth and the surface normal of an object to generate high-quality 3D surfaces with an illumination intensity matrix obtained from two adjacent light sources. A 3D representation of a human hand is presented in the Figure 2(b) as an example.

In order to maximize the benefit of the GAF point cloud registration scheme, we propose a directional navigation of the point automaton in c -space in a way similar to that of how the kernel function extracts 3D features. As shown in Figure 1, collision shield is formed so that point clouds of all the obstacles in the workspace will be collected at an instance such that

$$P_s^i := \{p_n \in R \mid n = 1, \dots, N\}, \quad (1)$$

where R is the workspace, N the number of points collected by all sensors, and i the sensor identifier. For a given set of joint space variables $\theta_1, \theta_2, \dots, \theta_n$, the forward kinematic model provides a transformation matrix such that

$$T_0^i = \begin{bmatrix} R^i & t^i \\ 0 & 1 \end{bmatrix}, \quad (2)$$

where R^i is the rotational matrix and t^i is the translational vector for each i th link, respectively. One example is shown for a two-link manipulator in Figure 3. Then by the forward kinematics, P_w^i , the i th point cloud vector in work space coordinate becomes

$$P_w^i = P_s^i \cdot T_0^n. \quad (3)$$

Now, for a given point cloud set, P_w^i , at an instance, we create a local c -space point cloud, P_c , in real time using RRT (rapid expanding random tree), one of model-based planning algorithms, such that $P_w \xrightarrow{\text{RRT}} P_c$, so that,

$$P_c := \{p_n \in C^k \mid n = 1, \dots, M\}, \quad (4)$$

where C is the c -space for the robot, k , the degree of freedom of the c -space, and M , the number of point clouds produced by the mapping process. For the RRT mapping process and to obtain P_c , we propose a virtual c -space sensor model with which the point automaton in c -space senses c -space obstacles. The virtual sensor in c -space is assumed to have a sensing range, r , and FOV (field of view), θ thus it forms a hyperconical sensing range in n -DOF c -space (see Figure 5).

P_c is a collection of configurations $(\theta_1, \theta_2, \theta_3, \dots, \theta_n)$ that causes collision between a robot and P_w^i by RRT mapping at each step. Now we define an "intensity function" $X : C^k \rightarrow C$ indicating the presence of the points in c -space.

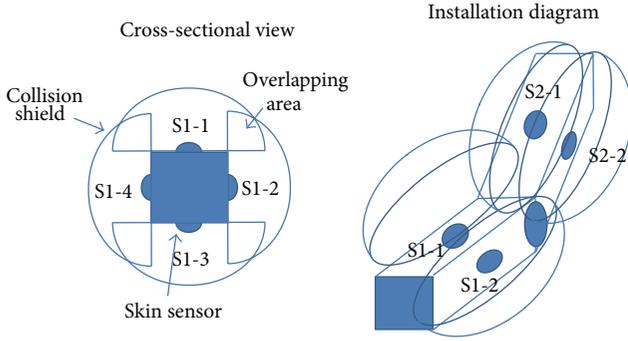
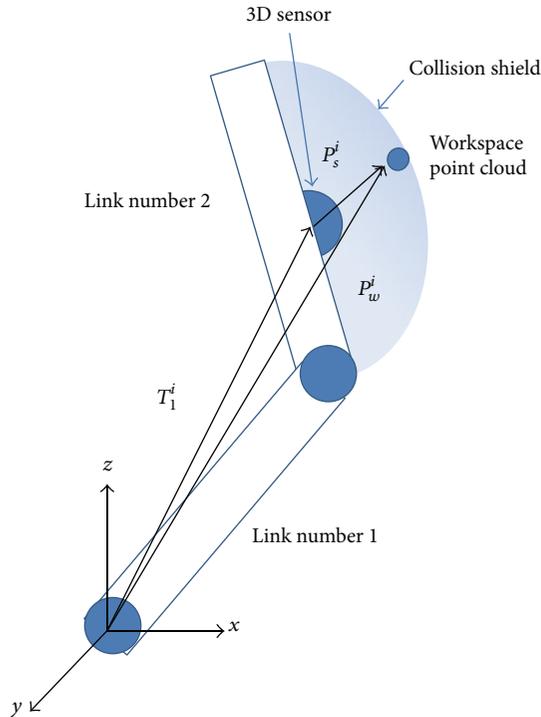


FIGURE 1: The concept of collision shield.



FIGURE 2: (a) Skin-type 3D sensor; (b) 3D construction.

FIGURE 3: Relationship between P_s^i and P_w^i .

We choose to represent the point set P_c as the sum of overlapping Gaussian distributions. The function X at a point $p_c^i \in P_c$ is defined as

$$X(p_c^i) = \sum_i \exp^{-\|p_c^i - p_c\|/\sigma_G}. \quad (5)$$

The gradient of the function X is then:

$$\nabla X(p_c^i) = -\frac{2}{\sigma_G^2} \sum_i (p_c^i - p_c) \exp^{-\|p_c^i - p_c\|/\sigma_G}. \quad (6)$$

To get an intuition for the meaning of the density gradient and the effect of the magnitude of σ_G , please refer to [19]. For point cloud registration, they further develop kernel functions for object identification. We use the most dominant gradient vector as a BNM path to direct the local motion of the point automaton in c -space. Then, the planner will steer the point automaton along the most dominant gradient to maximize exploration capability, thus rapidly navigating around c -space obstacles for map generation. The planner may look similar to the potential field planner to some degree because it steers the robot along the gradient of the cloud density. However, it is different in that it does not always move away from the obstacle, but it has tendency to travel along the direction by which the point clouds spread in space. This feature renders a rapid mapping of a c -space obstacle possible.

The point cloud data format in c -space, then, becomes as follows:

$$p_c^i = [x_c^i, y_c^i, z_c^i, \nabla X^i]^T. \quad (7)$$

Total framework of the proposed SPAM cycle is shown in Figure 4. Note that no inverse kinematics is involved for the planner; thus, the algorithm is simple and robust. A detail algorithm of c -space mapping is shown in Algorithm 1.

σ_d in the above algorithm is the standard distance distribution as shown below:

$$\sigma_d = \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta_i - \theta_{\text{mean}})^2}. \quad (8)$$

Collision check takes place in the virtual workspace by comparing $B_{\text{ROBOT}}(P_c)$, the workspace occupied by the robot model, and P_w^i , the point clouds of workspace obstacles. A detail algorithm of path planning is shown in Algorithm 2.

The ending condition of the “while” loop in Algorithm 2 provides the search completeness via space filling. Search continues until either the goal is found, or SDD of the free c -space is too dense ($< \delta_T$); thus, no more exploration is meaningful (Figure 8). The SPAM cycle is composed of two parallel processes with 5 main functions. In c -space mapping, the main area of work is to convert the workspace point clouds into c -space point clouds by RRT (rapid growing random tree) technique. Based on the registered c -space point clouds in the mapping process, the BNM path planner will guide the robot to the direction along the gradient of the intensity function. An intuitive view of the BNM algorithm is shown in Figure 5.

```

 $\Lambda_n = \{\}; V_n = \{\};$  // initialize RRT tree for  $n$ th expansion,  $c$ -space
 $P_w^n = P_w^{n-1} \cup \sum_{i=1}^n (P_s^i \cdot R^i + t^i)$  //  $n$ : No. of local step
do while  $\sigma_d(\Lambda_n) > \delta_\Lambda$ 
   $p_c \leftarrow$  a branch grown from  $T_{\text{RRT}}^n$  toward  $\nabla X_n(p_d)$ 
  if  $B_{\text{ROBOT}}(p_c) \cap P_w^n \neq \emptyset$  // if robot collides with  $P_w^n$ 
     $\Lambda_n = \Lambda_n \cup p_c$  // collect  $c$ -space point cloud
  else
     $V_n = V_n \cup p_c$ 
  end if
end do
 $P_c^n = P_c^{n-1} \cup \Lambda_n$  // update  $c$ -space point cloud
return  $(P_c^n, T_{\text{RRT}}^n)$ 

```

ALGORITHM 1: c -space mapping.

```

 $q_i$ : initial location for  $n$ th expansion
 $T_{\text{RRT}} = \{\}$ ,
do while  $\sigma_d(T_{\text{RRT}}) > \delta_T$ ,
  if  $(\min d(q, P_c^n) < \epsilon)$ 
    return (success) // A path has been found!
  else
    for  $i = 1 : N$  //  $N$ : number of point could in  $\Lambda_n$ 
       $\nabla X(p^i) = -(2/\sigma_G^2) \sum_i (p_c^i - p_c) \exp^{-\|p_c^i - p_c\|/\sigma_G}$ 
    next  $i$ 
     $\nabla X_n(p_d) = \nabla X_n(p) \mid \min_{p \in \Lambda_n} d(q_i, \Lambda_n)$  //  $d$ : dominant point in  $\Lambda_n$ 
  end if
  if  $\|q_{i+1} - q_i\| < \epsilon$ 
     $q_{i+1} = q \mid \min_{q \in C_f} X(q)$  // jump to a new free conf with min density
    // distribution
  else
     $q_{i+1} = q \mid \min_{q \in \Lambda_n} d_i(q, q_i)$  // initial location for next expansion
  end if
  call Algorithm 1
   $T_{\text{RRT}} = T_{\text{RRT}} \cup T_{\text{RRT}}^n$  // update RRT Tree
end do
return (failure)

```

ALGORITHM 2: BNM path planning.

As shown in Figure 5, the BNM algorithm has a tendency of directing the point automaton to glide along the surface of c -space objects. This tendency of the algorithm allows a rapid search of an object surface thus, rendering a systematic unknown environment exploration feasible. For the global motion of the point automaton, we utilized a simple RRT function in conjunction with the local motion by BNM algorithm. The second “if” condition check in the while loop in Algorithm 2 above is where the global motion takes place.

3. Simulation Results

In order to test the proposed algorithm, we set up a 2-DOF revolutionary link robot as a testbed for simulation. Each link is assumed to be equipped with IPA sensors so that the entire body is covered by a collision shield. Two algorithms are tested for comparison: sensor-based RRT (see [11] for more

TABLE I: Simulation results.

Algorithm	Total search time	No. of P_c	Mapping efficiency
S-RRT	6205 sec	1201	45%
BNM	3134 sec	1144	82%

details) and BNM algorithm introduced in this paper. The same sensor model and workspace configurations are applied. For 30 runs of each simulation, some statistics are shared in Table 1. Total search time is the time in seconds from the beginning to the end of the search operation.

No. of P_c stands for the total number of c -space point clouds generated during the search period. The mapping efficiency in Table 1 is a measure of how efficiently the algorithm generates the c -space map of a given environment.

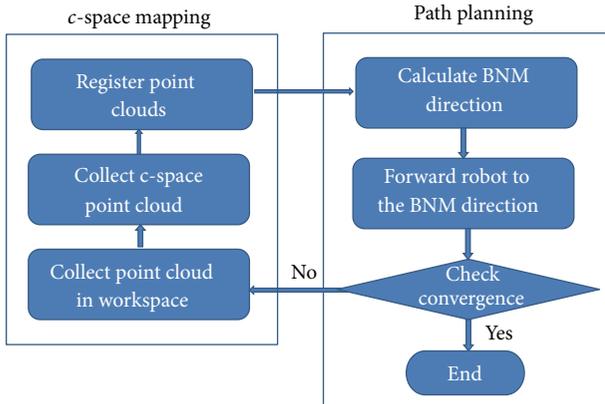


FIGURE 4: SPAM cycle.

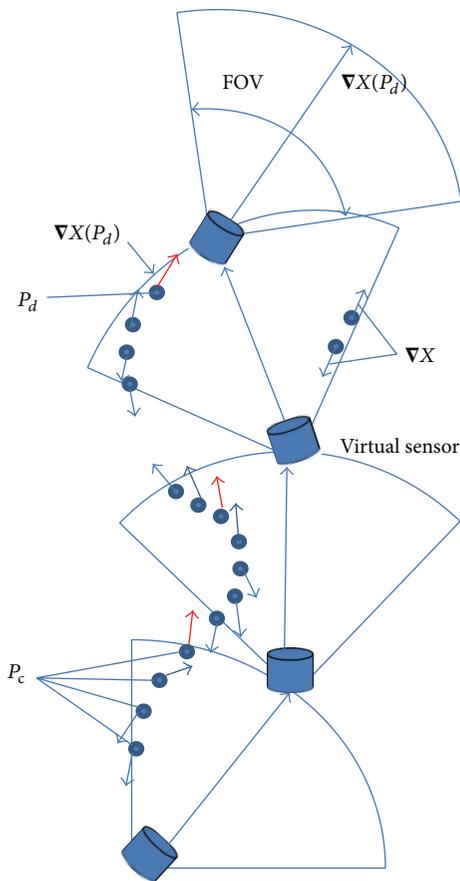


FIGURE 5: Algorithms 1 and 2 at a glance.

This measure is useful since it indicates how fast the algorithm reveals collision-free spaces. The mapping efficiency is defined as below:

$$\text{Mapping efficiency} = \frac{\% \text{ of map built}}{\text{No. of point clouds in } c\text{-space}} \quad (9)$$

Simulations are done by a Pentium dual core desktop with 2 GB memory. With the sensor-based RRT algorithm, about 45% of the c -space map is constructed (Figure 6) until the

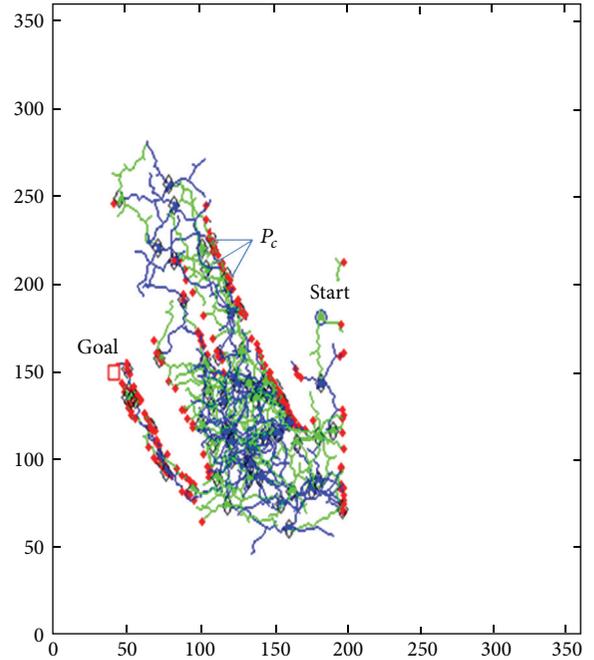


FIGURE 6: Sensor-based RRT algorithm.

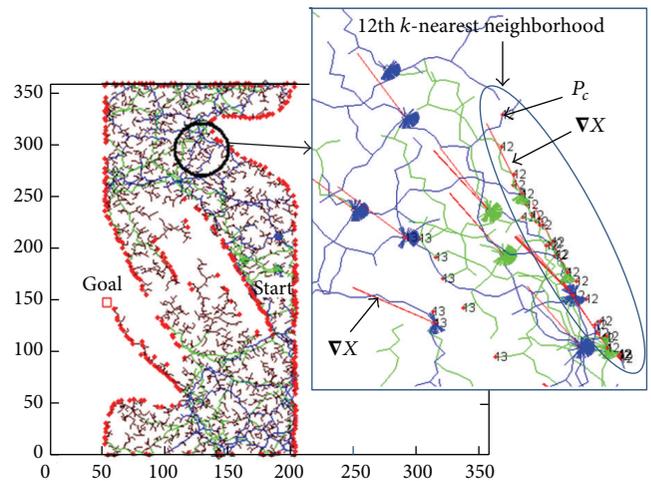


FIGURE 7: Best Next Move algorithm.

target is reached. Notice that the overall search time is about twice as much as the planning by the BNM algorithm. On the contrary, in the second test, BNM algorithm demonstrated about 82% achievement of the c -space construction upon the termination (see Figure 7). In the magnified window on the right, one can see the red dots that depict 12th k -nearest neighborhood by which the surface of the c -space obstacle is identified. Note that the direction of the gradient of the density function goes along the direction parallel to the 12th k -nearest-neighborhood point cloud in c -space.

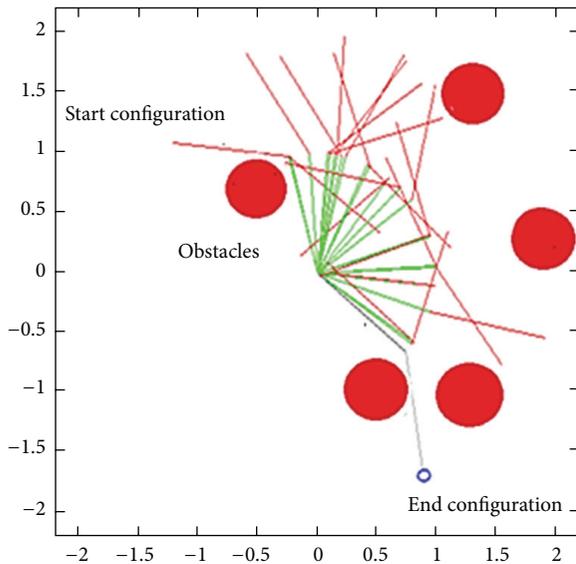


FIGURE 8: Workspace of the robot for milestones during the search operation.

4. Conclusion

In this paper, we proposed a SPAM (simultaneous planning and mapping) technique for a manipulator-type robot working in an uncertain environment via a Best Next Move algorithm. Motivation is that better map construction capability assures higher success ratio for the convergence to the goal. BNM algorithm offers a means toward SPAM of the manipulator planning in uncertain environments, thus improving mapping and planning capability at the same time. For rapid map building and path planning, we use a 3D depth camera based skin-type sensors that completely encompass the entire body of a manipulator. Captured cloud points by 3D sensors create an instantaneous c -space map whereby a Best Next Move algorithm guides the local motion of the manipulator. As shown in Table 1, BNM not only creates a c -space map with higher mapping efficiency, but also it directs the point automaton to the goal twice as faster as the sensor-based RRT algorithm in average.

References

- [1] L. Kavraki and J. C. Latombe, "Randomized preprocessing of configuration space for fast path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)*, pp. 2138–2145, San Diego, Calif, USA, May 1994.
- [2] R. Mahkovic and T. Slivnik, "Generalized local voronoi diagram of visible region," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, pp. 349–355, Leuven, Belgium, May 1998.
- [3] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, Eds., pp. 293–308, A. K. Peters, Wellesley, Mass, USA, 2001.
- [4] B. Laroche, P. Martin, and P. Rouchon, "Motion planning for a class of partial differential equations with boundary control," in *Proceedings of the 1998 37th IEEE Conference on Decision and Control (CDC '98)*, pp. 3494–3497, Tampa, Fla, USA, December 1998.
- [5] J. T. Schwartz and M. Sharir, "On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds," *Advances in Applied Mathematics*, vol. 4, no. 3, pp. 298–351, 1983.
- [6] C. Guestrin and D. Ormoneit, "Robust combination of local controllers," in *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI '01)*, pp. 178–185, Seattle, Wash, USA, August 2001.
- [7] N. Bezzo, R. Fierro, A. Singler, and S. Ferrari, "A disjunctive programming approach for motion planning of mobile router networks," *International Journal of Robotics and Automation*, vol. 26, no. 1, pp. 13–25, 2011.
- [8] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: robust sensor-based coverage of unstructured environments and probabilistic methods," *International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 441–466, 2003.
- [9] E. Balas, "Disjunctive programming and a hierarchy of relaxations for discrete optimization problems," *SIAM Journal on Algebraic and Discrete Methods*, vol. 6, no. 3, pp. 466–486, 1985.
- [10] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, Fla, USA, April 1997.
- [11] D. Um and D. Ryu, "A framework for unknown environment manipulator motion planning via model based realtime rehearsal," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 5, no. 1, pp. 30–35, 2011.
- [12] Y. Yu and K. Gupta, "C-space entropy: a measure for view planning and exploration for general robot-sensor systems in unknown environments," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1197–1223, 2004.
- [13] K. Kieda, H. Tanaka, and T. Zhang, "On-line optimization of avoidance ability for redundant manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, Beijing, China, October 2006.
- [14] Z. Chu, P. Cao, Y. Shao, D. Qian, and X. Wang, "Trajectory planning for a redundant mobile manipulator using avoidance manipulability," in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '09)*, Shenyang, China, August 2009.
- [15] W. Qizhi and X. De, "On the kinematics analysis and motion planning of the manipulator of a mobile robot," in *Proceedings of the on Chinese Control and Decision Conference (CCDC ;11)*, Mianyang, China, May 2011.
- [16] J. W. Burdick, "Global kinematics for manipulator planning and control," *Proceedings of the on Signals, Systems and Computers*, vol. 2, pp. 1002–1007, 1989.
- [17] D. Um, "How to tackle sensor-based manipulator planning problems using model-based planners: conversion and implementation," *International Journal of Robotics and Automation*, vol. 24, no. 2, pp. 137–146, 2009.
- [18] D. Um, D. Ryu, and M. Kal, "Multiple intensity differentiation for 3D surface reconstruction with mono-vision infrared proximity array sensor," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3352–3358, 2011.

- [19] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [20] M. Temerinac, M. Reisert, and H. Burkhardt, "Invariant features for searching in protein fold databases," *International Journal of Computer Mathematics*, vol. 84, no. 5, pp. 635–651, 2007.
- [21] L. Torabi and K. Gupta, "Integrated view and path planning for an autonomous six-DOF eye-in-hand object modeling system," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 4516–4521, Taipei, Taiwan, October 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

