Simultaneous Planning and Mapping (SPAM) for a Manipulator by Best Next Move in Unknown Environments

Dugan Um[1], Marco A. Gutiérrez[2], Pablo Bustos[2] and Sungchul Kang[3]

Abstract— **In this paper, we propose a SPAM (Simultaneous Planning and Mapping) technique for a manipulator type robot working in an uncertain environment via a 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. We assume no a priori knowledge of either the obstacles or the rest of the environment exits. For rapid map building and path planning, we use a skin type setup based on 3D depth camera sensors that completely encompass the entire body of a manipulator. The 3D sensors capture the point clouds used to create an instantaneous c-space map whereby a Best Next Move algorithm directs the motion of the manipulator. The Best Next Move algorithm utilizes the gradient of the density distribution of the k-nearest-neighborhood sets in c-space. It has tendency to travel along the direction by which the point clouds spread in space, thus rendering faster mapping of c-space obstacles.**

The proposed algorithm is compared with several sensor based algorithms for performance measurement such as map completion rate, distribution of samples, total nodes, etc. Some improved performances are reported for the proposed algorithm. Several possible applications include semi-autonomous tele-robotics planning, humanoid arm path planning, among others.

I. INTRODUCTION

Motion planning in unknown environments is a challenging problem in path planning. Sensor based approaches have been the dominant trends in the study of unknown environment planning for decades. When it comes to unknown environment planning, a planner calls for continuous perception and planning, thereby closing the loop between sensation and actuation. Due to the limited sensing distance of most of the depth sensor or visual occlusion, only the local area is known to the robot for local path planning. No optimum global path generation idea is reported so far due to the uncertainty innate by an unknown environment. However, if a planner can produce a global map rapidly, optimum path planning is feasible in unknown environment.

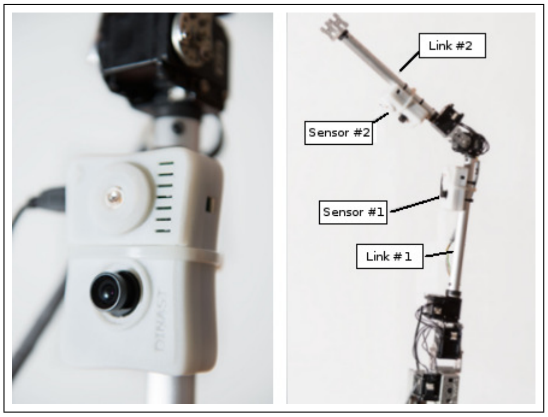


Fig. 1. IPA sensor installation on a three Degrees Of Freedom (3-DOF) robotic linkage

Sequential mapping of a local area and path planning is a natural step for sensor based motion planning. In [1], a novel framework for an unknown environment path planning of a manipulator type robot is proposed. The framework described in [1] is a sensor based planner composed of a sequence of multiple MBPs (Model Based Planners) in the notion of cognitive planning using realtime rehearsal. C-space Entropy is examined in [2] for planning and exploration for a robot with an eye-in-hand sensor system. Natural planning and expanding steps of the c-space is repeated in the paper for an unknown environment path planning algorithm. However, the eye-in-hand sensor has limitation in reporting collision or c-space mapping in realtime due to visual occlusion.

Other studies in sensor based planning of manipulators include [3] whereby an avoidance ability of a redundant manipulator is studied with a moving camera on the ceiling. In [4], trajectory planning for a redundant mobile manipulator is studied using avoidance manipulability concept. Manipulability is the point of study for best path selection in multiple available configurations with singularity and manipulability ellipsoid. In [5] and [6], kinematic analysis in local or global motion planning for a manipulator has been studied in the notion of singularity for a redundant robot as well. Topological analysis in conjunction with singularity concern for a redundant manipulator is dealt with the study on critical point surfaces in configuration space. In summary, the result in the paper implies that a manipulator has to stay in a continuous c-sheet to avoid singularity, which means that a motion planning with inverse kinematic concern is neither robust nor efficient due to the limited utility of a given c-space.

As a result, majority of the manipulator planning schemes we investigated are either hindered by the sensor configuration or by motion constraints for a realtime manipulator motion planning especially for a crowded unknown environments. More flexible sensor configurations for maximum

[1] D. Um is with Texas A&M University Corpus Christi, USA `dugan.um at tamucc.edu`

[2] M. A. Gutiérrez and P. Bustos are with Robolab, University of Extremadura, Spain `marcog at unex.es`

[3] S. Kang is with Korea Institute of Science and Technology (KIST) `kasch@kist.re.kr`

coverage in conjunction with a supportive planning algorithm is, therefore, in essential need.

To that end, we propose a skin type sensor made out of cameras with 3D depth sensing capability to tackle an unknown environment manipulator motion planning problem (see 1 for a 3-DOF robot example). Our approach is a probabilistic path planning with Simultaneous Planning and Mapping, thus SPAM. For 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 realtime point clouds of obstacles from any posture of the robot; thereby a realtime local c-space construction becomes feasible. However, an appropriate guidance algorithm of the robot in global motion planning is of utmost importance for maximum exploration and convergence capability. To that end, we envision a 3D point cloud registration method a possible guidance algorithm for manipulatorś motion.

3D point cloud registration calls for various descriptors for object recognition [7]. We take advantage of such registration processes to propose a guidance method of a manipulator in a partially constructed c-space environment. Amongst many registration methods, is the Group Averaging Features, an invariant feature for point cloud registration [8], in which a gradient of the density distribution is used to log essential points for 3D shape identification.

We discuss rationale of why and how we utilize the steps of the invariant feature extraction method for sensor based motion planning in Section II. We discuss results of the comparison between simulations of the proposed algorithm and another sensor based planning algorithm in Section III. Map completion rate, distribution of samples, and total nodes are measured for comparison. Section IV shows a real device used to validate the proposed method. Finally in section V improved performances are reported for the proposed algorithm along with some lines of interesting future work.

II. BEST NEXT MOVE PLANNER

Due to the higher order complexity of the manipulator type robot path planning, probabilistic sampling based search is common in general. Several sampling based path planners are reported to be a complete planner so that they either find a path or terminate otherwise. Manipulator path planning in unknown environment, however, is challenging in that neither the path optimality nor the plannerś completeness can be guaranteed.

Gradual but rapid construction of the c-space map, if feasible, allows a planner to complete a mission in path search with higher probability. C-space mapping especially in a crowded environment is the most daunting task in manipulator path planning though. In [9], Best Next View (BNV) in conjunction with a sensor-based roadmap planner is used for object mapping in unknown environment. Utilization of BNV in the object recognition in an unknown environment is through the concept of detecting key events in the set of range data such as discontinuity of the range data in the scene. These key events are used to drive the global motion of the manipulator to reduce the ignorance level of the given workspace.

Similarly, we propose the Best Next Move algorithm as a guidance strategy of the robotś global motion in uncertain environment. By the BNM algorithm, the local motion in each step is designed to reveal the maximum environmental map possible. We use the point cloud registration scheme in [8] as the best next move strategy since it calls for rapid point cloud identification and collection of a 3D shape.

When it comes to unknown environment manipulator motion planning, two subjects have to be addressed in parallel: map construction and navigation for convergence. We consider map construction and goal convergence as two separate tasks for unknown environment manipulator planning. The more complete c-space map a planner generates, the better chance of convergence to the goal achieved. To that end, objectives of motion strategy are set such that:

1) Rapid map construction stage: steering global motion to build a maximum environment map
2) Goal convergence stage: achieving search completeness

For the first objective 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 global motion. Best Next Move is the direction possibly to collect maximum information of c-space obstacles.

Group Average Feature (GAF) method, one of the 3D point cloud registration methods is designed to search point cloud sets to register the uniqueness of a 3D object as on [8]. With the sensitive skin type sensor, we can construct a workspace of the robot in realtime and map out the c-space obstacle instantaneously. That workspace will then correspond to the point cloud data obtained by the collision shield developed around the robotic manipulator at a certain point of time. First we collect point cloud data by 3D depth sensors attached on the manipulator.

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 similar to the kernel function to extract features. To that end, for a given P_c (workspace point cloud) at an instance, 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 hyper conical sensing range in n-DOF c-space (see Figure 2).

The sensitive skin-setup of sensors covers the entire body of a manipulator, meaning that we can obtain 3D point data all around the robot manipulator. Thanks to that, a collision shield is formed and point clouds of all the obstacles in the workspace at a time t will be collected such that:

$$P_s^t := \left\{ p_j^i \in R | j = 1, ..., N \right\} \qquad (1)$$

where, p_j^i is the point cloud from sensor j situated at

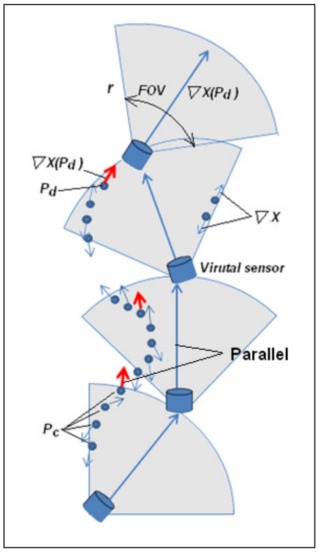


Fig. 2. Algorithm 1&2 at a glance

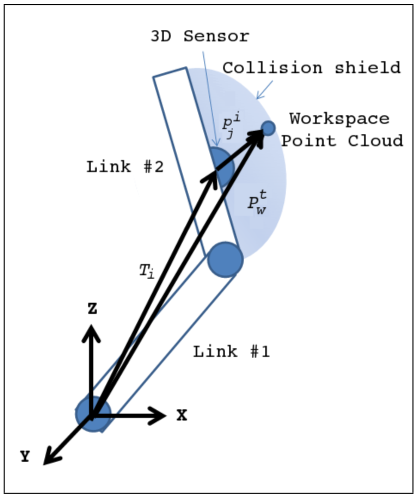


Fig. 3. Two link manipulator showing the relationship between the set of point clouds P_s^t and the final resulting one, P_w^t

link i, R is the workspace, N is the total number of point clouds collected by all sensors. Therefore P_s constitutes the set of all point clouds obtained by the robotic manipulator installed sensors at time t. For a given set of joint space variables $\theta_1, \theta_2, ..., \theta_n$, that define a certain configuration of the manipulator, the forward kinematic model provides transformation matrix such that:

$$T_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \tag{2}$$

where, R_i is the rotational matrix and t^i is the translational vector for each i_{th} link respectively. Then by the forward kinematics, P_w^t , the point cloud of the collision shield at a certain time t in the workspace coordinate, becomes:

$$P_w^t = \bigcup_j^N p_j^i * T_i \tag{3}$$

Finally, P_w^t becomes the set of points formed by all the points from the sensors point clouds at time t translated and rotated by its corresponding $i_t h$ link T_i matrix to the workspace generic coordinates. An example of this in a two link manipulator is shown at Figure 3

For a given point cloud set, P_w^t, at an instance, we create a local c-space map in realtime using RRT (Rapid expanding Random Tree), one of probabilistic sampling algorithm, such that $P_w^t \overset{RRT}{\rightarrow} P_c$, so that:

$$P_c := \left\{ p_n \in C^k | n = 1, ..., M \right\} \tag{4}$$

where, C is the c-space for the robot, k the degree of freedom of the c-space, and M is the number of point clouds produced by the mapping process.

Now we define an 'intensity function' $X : C^k \rightarrow C$ indicating the presence of the point in c-space. We choose to represent the point set P_c as the sum of overlapping Gaussian distributions. The function X at point $p \in P_c$ is defined as:

$$X(p) = \sum_i exp^{-\left(\frac{\|p_i - p\|}{\sigma_G}\right)^2} \tag{5}$$

The gradient of the X is then:

$$\triangledown X(p) = \frac{-2}{\sigma_G^2} \sum_i (p_i - p) exp^{-\left(\frac{\|p_i - p\|}{\sigma_G}\right)} \tag{6}$$

To get an intuition for the meaning of the density gradient, please refer to [8]. For point cloud registration, they further develop kernel functions for object identification. We use the most dominant gradient vector as a guidance to direct the global 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 searches c-space obstacles. The planner may look similar to potential field planner 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, thus faster mapping of c-space obstacles possible.

The point cloud data, then, becomes following:

$$p_i = [x_i, y_i, z_i, \triangledown X_i]^T \tag{7}$$

Total framework of the proposed SPAM cycle is shown in Figure 4. Note that no inverse kinematic solution is necessary for the planner, thus the algorithm is simple and robust. Detail algorithm of c-space mapping is shown in Algorithm 1.

σ_d in Algorithm 1 is the Standard Distance Distribution as shown in Equation 8. Λ_n, occupied c-space or a collection of c-space point clouds at step n will be added to the c-space point clouds, P_c^n, at the end of each mapping loop.

Algorithm 1: c-space mapping

```
// Initialize RRT tree for expansion:
$\Lambda_n \leftarrow \varnothing$;
$T_{RRT}^n \leftarrow \varnothing$;
// workspace point clouds:
$P_w^n \leftarrow P_w^{n-1} \cup \sum_{i=1}^n \left(P_s^i \cdot R^i + t^i\right)$;
do while $\sigma_d(\Lambda_n) > \delta_\Lambda$ do
    grow $T_{RRT}^n$ forward $\nabla X_n(P_d)$;
    $p_c \leftarrow$ a branch grown from $T_{RRT}^n$;
    // if robot collides with workspace
       point cloud:
    if $B_{ROBOT}(p_c) \cap P_w^n \neq 0$ then
        // Collect c-space point cloud:
        $\Lambda_n \leftarrow \Lambda_n \cup p_c$;
        remove $p_c$ from $T_{RRT}^n$;
    end
end
// Update c-space point cloud:
$P_c^n = P_c^{n-1} \cup \Lambda_n$;
return $(P_c^n, T_{RRT}^n)$;
```

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

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

The condition of the while loop provides the search completeness of the algorithm via space filling. Search continues until either the goal is found, or SDD of the free c-space is too dense ($< \sigma_T$), thus no more exploration is meaningful.

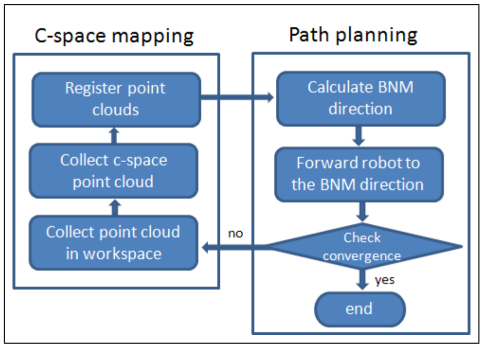


Fig. 4. SPAM cycle

The SPAM cycle is composed of two parallel processes with 5 main functions (See Figure 4). In c-space mapping, main area of work is to convert the workspace point clouds into c-space point clouds by RRT (Rapid growing Random Tree). In a virtual environment, a robot will move by RRT algorithm and whenever collision occurs with a point cloud, the robotś configuration will be sampled and stored as a c-space point cloud data. Based on the registered c-space point clouds in the mapping process, the BNM path planner will guide the robot to the direction obtained by the gradient of the intensity function. Intuitive operation of the algorithm is shown in Figure 3. As shown in Figure 3, the BNM algorithm has a tendency of directing the point automaton to glide along the surface of c-space objects. This tendency allows rapid search of an object surface so that a systematic unknown environment explorations is feasible.

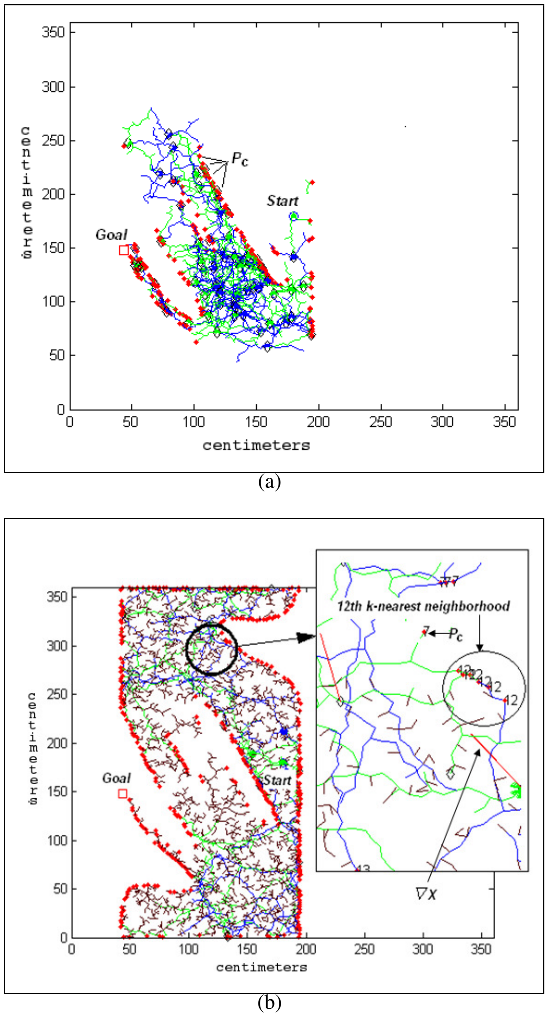
Algorithm 2: BNM path planning

```
$q_i \leftarrow$ initial location for nth expansion;
$T_{RRT} \leftarrow \varnothing$;
$P_c \leftarrow \varnothing$;
do while $\sigma_d(T_{RRT}) > \sigma_T$ do
    if $min\ d(q., P_c) < \varepsilon$ then
        // a path has been found!
        return success
    else
        // N: number of point cloud in $\Lambda_n$
        for $i \leftarrow 1$ to $N$ do
            $\nabla X(p) \leftarrow \frac{-2}{\sigma_G^2}\sum_i (p_i - p) exp^{-\left(\frac{\|p_i - p\|}{\sigma_G}\right)^2}$;
        end
        $\nabla X_n(p_d) \leftarrow \nabla X_n(p) | min_{p \in \Lambda_n} d(q_i, \Lambda_n)$;
    end
    if $\|q_{i+1} - q_i\| < \varepsilon$ then
        // jump to a new free conf with min
           density distribution
        $q_{i+1} = q | min_{q \in C_f} X(q)$;
    else
        // initial location for next expansion
        $q_{i+1} = q | max_{q \in \Lambda_n} d_1(q, q_i)$;
    end
    call_Algorithm1();
    // Update RRT Tree
    $T_{RRT} \leftarrow T_{RRT} \cup T_{RRT}^n$;
    $P_c \leftarrow P_c \bigcup P_c^n$;
end
```

III. SIMULATION RESULTS

In order to test the proposed algorithm, we setup a 2 DOF revolutionary link robot as a testbed for simulation. Two algorithms are tested for comparison: Sensor-based RRT (see [1] for more detail) and BNM algorithm, introduced in this paper. Using the algroithms the system will try to find a path in an unknown environment with different obstacles in order to reach a certain target. The same sensor model and workspace configurations are applied on both algorithms.

Figures 5(a) and 5(b) show the result of the path search for RTT and BNM respectively. The steps that the robot makes in the different algorithms is shown figures 6(a) and 6(b). 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. This measure is useful and important since rapid and complete map generation is the key strategy in SPAM in unknown environments. The more information we obtain about the unknown environment, the better the planner can plan a path converging to the goal



(a)

(b)

Fig. 5. a) Sensor-based RRT algorithm; b) Sensor-Based BNM algorithm. Each red dot reperesent a point cloud obtained by the 3D sensors.

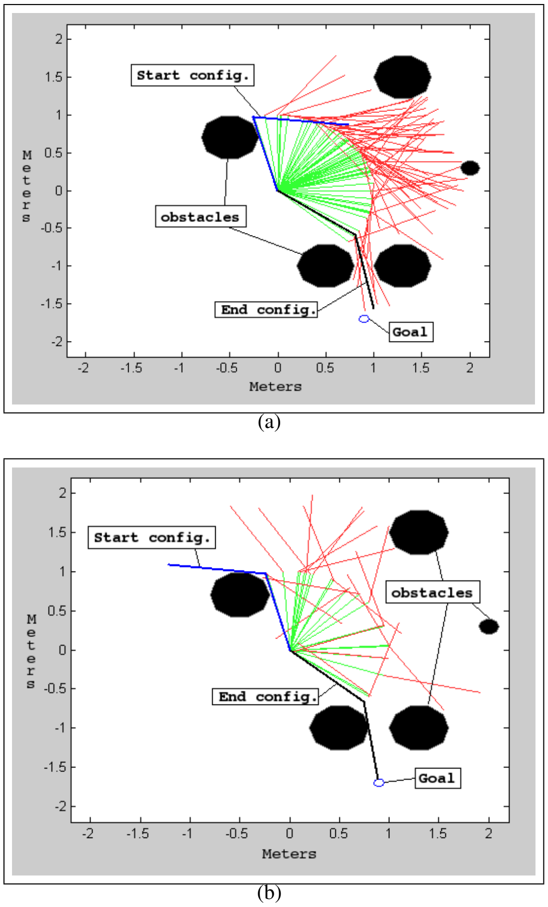
point. We define the mapping efficiency such that:

$$Mapping\ efficiency = \frac{\%\ of\ map\ built}{No.\ of\ point\ clouds\ in\ cspace} \tag{9}$$

With the sensor-based RRT algorithm, about 45% of the c-space map is constructed upon termination, see figure 5(a) for a visual overview and table I for quantified data. Figure 6(a) shows the different positions the robot takes following the RRT algorithm. As milestone in workspace reveals, in the latter figure, overlapping occurs densely in certain areas.

To the contrary, in the second test, BNM algorithm demonstrated about 82% of the c-space construction before the termination (see figure 5(b) and table I). 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. If the window is carefully examined, there are two vectors that show the directions of the gradient of density function. Figure 6(a) shows the milestones in workspace for the robot. You can appreciate that they are more evenly distributed over the entire workspace as a result.

As a summary, overall search time of the sensor-based RRT planner is about twice as much as that of the BNM algorithm. Another thing noticeable, upon the completion of the path search, is the rate of environmental map completion. If you look at figure 5(a) and figure 5(b) for comparison, significantly more environmental map is revealed by the BNM algorithm compared to that of Realtime-RRT planner.



(a)

(b)

Fig. 6. Milestones in workspace for robot movement by Sensor based: a) RRT algorithm; b) BNM algorithm. Position in blue is the starting postion while the black one is the final position, in which the robot reaches the target. Black dots are the obstacles randomly placed in the unknown environment.

TABLE I

SIMULATION RESULTS

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

IV. EXPERIMENTS

Two IPA sensors (see [10]) are installed on a manipulator type robot to generate a collision shield around it following the skin type sensor setup (Figure 1). Transformation matrices for local coordinates to the global coordinate for each camera configuration have been setup using Equation 1. The robot tries to reach an object, set as goal, through the unknown environment where an obstacle (a big white box) has ben placed. In Figure 7, the robot stops and takes depth map from each sensor, by which a workspace point cloud

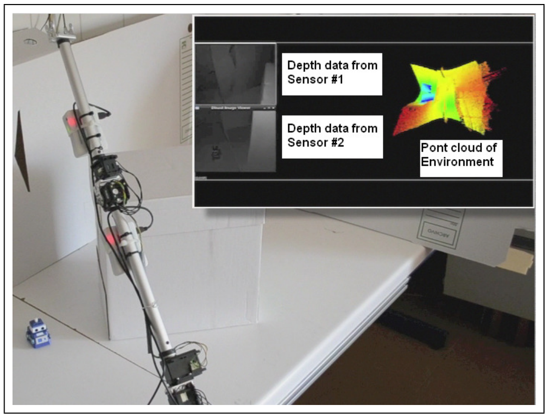

Depth data from Sensor #1
Depth data from Sensor #2
Pont cloud of Environment

Fig. 7. Point cloud registration with depth data from sensor #1 and sensor #2

is generated. With the point cloud map, BNM kicks in to generate a c-space point cloud map as shown in Figure 8. Now the robot is guided along the intensity gradient for the next step generating the maximum environment map until it reaches the goal point (Figure 9).

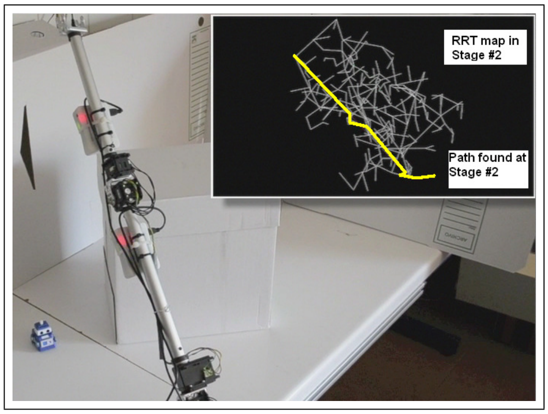


Fig. 8. C-Space point cloud registration by RRT expansion + path planning

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose 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 in that better map construction capability assures higher success ratio for the convergence to the goal. BNM algorithm offers a means for SPAM of the manipulator planning in uncertain environments thus improving mapping and planning at the same time. For rapid map building and path planning, we use a 3D depth camera based skin type sensors setup 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 global motion of the manipulator. We proposed mapping efficiency as a measure of SPAM capability. The proposed BNM algorithm demonstrated up to 82% mapping efficiency in average of 30 runs. As shared 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. We also implemented the BNM algorithm with a sensitive skin sensor setup equipped two linkage manipulator for verification in a real world. Realtime workspace point cloud generation capability from 3D depth sensor data is demonstrated for SPAM technique as well.

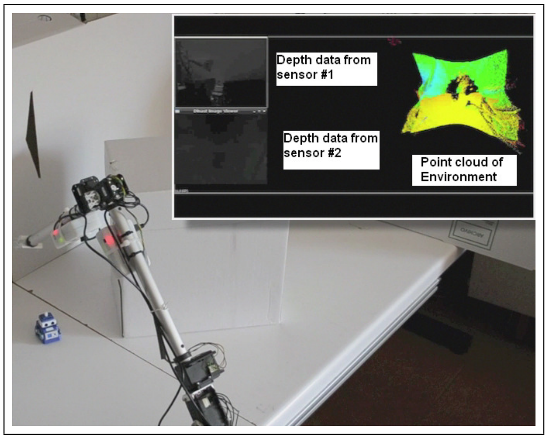

Depth data from sensor #1
Depth data from sensor #2
Point cloud of Environment

Fig. 9. Robot reaches the goal via BNM algorithm

The need for the FOV of the camera to cover the entire range of a link could be avoided. Further development of the algorithm in order to make it able to work with less and more spread sensors might end up in great improvements.

REFERENCES

[1] Dugan Um, Dongseok 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, 2011.

[2] Young Yu, Kamal Gupta, "C-space Entropy: A Measure for View Planning and Exploration for General Robot-Sensor Systems in Unknown Environments," Int. Journal of Robotics Research, Vol. 23, No. 12, pp. 1197-1223, Dec. 2, 2004.

[3] Keiji Kieda, Hiroshi Tanaka, Tong-xiao Zhang, "On-line Optimization of Avoidance Ability for Redundant Manipulator," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, Oct, 9-15, 2006, Beijing, China

[4] Ze Chu, Peng Cao, Yan-Ming Shao, Dong-Hai Qian, Xiang-Can Wang, "Trajectory Planning for a Redundant Mobile Manipulator using Avoidance Manipulability," Proc. of IEEE Int. Conf. on Automation and Logistics, Aug, 2009, Shenyang, China.

[5] Wang Qizhi, Xu De, "On the Kinematics Analysis and Motion Planning of the Manipulator of a Mobile Robot," Proc. on Chinese Control and Decision Conference, 2011.

[6] Joel W. Burdick, "Global Kinematics for Manipulator Planning and Control," Proc. on Signals, Systems and Computers, vol. 2, pp. 1002-1007, 1989.

[7] Paul J. Besl, Neil D. McKay, "A Method for Registration of 3-D Shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, No.2, pp. 239-255, Feb., 1992.

[8] Maja Temerinac, Marco Reisert, Hans Burkhardt, "Invariant Features for Searching in Protein Fold Databases," International Journal of Computer Mathematics, Vol. 84, No. 5, pp 635-651.2007.

[9] Liila Torbi, Kamal Gupta, "Integrated View and Path Planning for an Autonomous six-DOF Eye-in-hand Object Modeling System," IEEE/RSJ Int. Conf. on Intelligent Robots an Systems, Oct. 2010, Taipei, Taiwan.

[10] Dugan Um, Dongsuk Ryu, MyungJoon 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, Jun, 2011.