Robotica (2017) volume 35, pp. 1176–1191. © Cambridge University Press 2016
doi:10.1017/S0263574715001101

Safe manipulation in unknown, crowded environments via sensor-based interleaving planner: interleaving software and sensitive skin hardware

Dugan Um†* and Dongseok Ryu‡

† *Texas A&M University - Corpus Christi 6300 Ocean Dr. Unit 5797 Corpus Christi, TX 78412, USA*
‡*Korea Atomic Energy Research Institute 989-111 Daedeok-daero, Yuseong-gu Daejeon, 305-353, South Korea Email: sayryu@kaeri.re.kr*

(Accepted December 8, 2015. First published online: February 11, 2016)

SUMMARY

As various robots are anticipated to coexist with humans in the near future, safe manipulation in unknown, cluttered environments becomes an important issue. Manipulation in an unknown environment, however, has been proven to be NP-Hard and the risk of unexpected human–robot collision hampers the dawning of the era of human–robot coexistence. We propose a non-contact-based sensitive skin as a means to provide safe manipulation hardware and interleaving planning between the workspace and the configuration space as software to solve manipulation problems in unknown, crowded environments. Novelty of the paper resides in demonstration of real time and yet complete path planning in an uncertain and crowded environment. To that end, we introduce the framework of the sensor-based interleaving planner (SBIP) whereby search completeness and safe manipulation are both guaranteed in cluttered environments. We study an interleaving mechanism between sensation in a workspace and execution in the corresponding configuration space for real-time planning in uncertain environments, thus the name interleaving planner implies.

Applications of the proposed system include manipulators of a humanoid robot, surgical manipulators, and robotic manipulators working in hazardous and uncertain environments such as underwater, unexplored planets, and unstructured indoor spaces.

KEYWORDS: Sensor-based planning; point cloud; unknown environment motion planning; collision avoidance.

1. Introduction

Imagine that you command a robot to bring a milk bottle from your refrigerator. Imagine you command a robot to bring a hammer from your storage, yes, from a messy storage. Robots function well in a well-defined workspace, but not in an uncertain and crowded environment. Robots, nowadays, can climb up a stair, can avoid obstacles, move on an uneven terrain such as mountain, unpaved road, etc. However, manipulation in a crowed environment without causing collision and yet achieving search completeness is still an unsolved problem. Therefore, for human–robot coexistence, challenges are to impart safe autonomy in manipulation for cluttered environments. Several recent works exemplifies planning in cluttered environments.[1–5] For the planning strategy in ref. [1], a skin type sensor may be needed to make it feasible for whole-body reactive planning in a cluttered environment. In ref. [2], a point robot guided by a greedy search technique in a cluttered environment successfully demonstrated path planning capability, taking advantage of mechanical simplicity. A fast obstacle avoidance algorithm in cluttered environments is demonstrated in ref. [3] by approximating obstacles to convex shapes thereby no local minima exists. Other examples in this regard[4,5] demonstrate escalating interests in the area of manipulation in cluttered environments.

* Corresponding author. E-mail: dugan.um@tamucc.edu

Unknown environment motion planning is one of the most daunting tasks especially for a manipulator type robot such as a humanoid or multiple arm robots. Sensor-based approaches have been the most dominant trend in the study of unknown environment planning for decades. When it comes to unknown environment planning, a planner calls for continuous perception and execution. Due to the nature of planning in unknown environments, probabilistic search in configuration space becomes the most buyable approach. Therefore, the dilemma of perception in a workspace and execution in the corresponding configuration space has to be resolved.

SLAM (Simultaneous Localization and Mapping) is the well-known approach for mobile robot navigation and planning in unknown environments. Since a mobile robot is treated as a point automaton, a probabilistically complete planner is applicable directly without interleaving between a workspace and the corresponding configuration space. However, for a manipulator, since perception takes place in the robot's workspace and probabilistic planning needs the corresponding configuration space for search completeness, interleaving between the workspace and the configuration space is inevitable.

While several AI planning techniques such as hierarchical planning and action–reaction planning have been introduced, interleaving planning has been evidenced in offering solutions to otherwise difficult or impossible planning problems such as planning in a dynamic environment. The concept of interleaving planning was coined in the 1970s and is exemplified by Ambros-Ingerson and Steel's work.[6] Nourbakhsh and Reza classified interleaving strategies into two categories: subgoaling and simplification.[7] As the concept has been further developed with advanced computational techniques, numerous studies have leveraged the adaptive manner of interleaving planning. For example, Kaelbling and Lozano-Perez showed hierarchical replanning and execution with geometric reasoning.[8]

Furthermore, interleaving planning extended its domain to various areas over the last decade. In ref. [9], planning in a dynamic environment is challenged via interleaving planning between sensing and execution. Motion planning of a robot with five simultaneously moving grippers in a carousel is tackled via interleaving planning. Planning and execution are interwoven to each other to ensure near-optimal performance with random requests in a dynamic environment. In ref. [10], interleaving between a symbolic planner and a geometric planner takes place to bridge the gap between higher level abstract planning and motion planning in a real world. To develop an interwoven symbolic–geometric domain, backtracking is used as an important interleaving mechanism. The model predictive planner in ref. [11] exemplifies the interleaving planning in that two sequentially executed planners: a task-space and a joint-space planner are interwoven for contact force sensing and robot execution.

One disadvantage of such interleaving planners, if performed between perception and actuation, is that there is always a time delay between stages since perception and actuation cannot take place simultaneously. Albeit the disadvantage due to time delay, interleaving planning is the most pragmatic methodology for a manipulator planning problem in a cluttered environment since a probabilistically complete planner solves a path problem if a tractable path exists. Sensing completeness has to be guaranteed for an interleaving planner though (see Section 3). In order to apply a probabilistic motion planner such as PRM[12] or RRT[13] at each step of global planning, a partial c-space has to be revealed. Although some attempts are made in analytic c-space computation for a simple case,[14] the construction of a c-space from a sensed workspace requires lengthy simulation since no analytic functional mapping is available for a complex, clustered environment. Therefore, there are four issues to solve a safe manipulator planning in a cluttered, completely unknown environment.

(1) Complete sensing of surrounding workspace in real time.
(2) Creating a partial workspace from sensed data.
(3) Creating a partial c-space via simulation.
(4) Probabilistic motion planning in the partially created c-space.

As mentioned earlier, between step 2 and 4, partial c-space creation takes a non-zero time period thus the robot may have to be stationary until the simulation for c-space creation finishes. We use probabilistic sampling based simulation to create partial c-spaces. This will lead to frequent interruptions of the overall motion since the planner needs partially created c-spaces. Therefore, a unique challenge of the proposed solution is to minimize motion interruptions to streamline the overall robot motion assuring safe manipulation. Our strategy is to minimize each step cycle, or motion resolution for minimally disrupted continuous motion. In addition, we use real-time OS to run

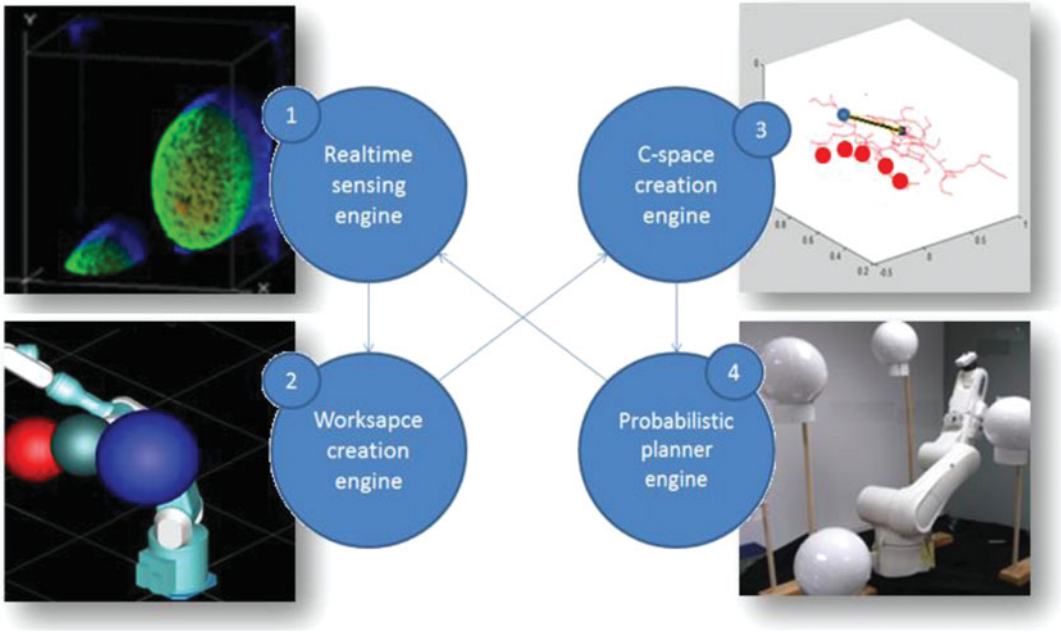

Fig. 1. A framework of sensor-based interleaving planner.

four engine threads simultaneously (See Fig. 1). Data exchange between each engine is only allowed via status flags for real-time execution. In the next section, we discuss each stage of the proposed solution in more detail.

2. Sensor-Based Interleaving Planner (SBIP)

In this section, we discuss the framework of the proposed interleaving planner for safe manipulation in crowded environments. Important issues in each stage will be addressed and discussed.

2.1. Safe sensing of a partial surrounding workspace in real time

For safe manipulation in an unknown, crowded environment, real-time sensing of a surrounding workspace is the essential enabling technology. Important issues in this stage are safe sensing (none-contact based) and sensing completeness (see Section 3 for more detail). In other words, the planner should provide a solution as to how to reveal a partially complete workspace without collision. Partially complete workspaces are needed for probabilistically complete planning in the corresponding c-space. Several workspace sensing or mapping techniques have been reported. The eye-in-hand sensor system has been developed to reduce c-space Entropy for planning and exploration.[15] Natural planning and expanding steps of c-spaces are repeated in the paper for unknown environment path planning. However, the eye-in-hand sensor is limited in reporting collision or in surrounding workspace mapping due to visual occlusion.

Another study includes a moving camera on the ceiling for redundant manipulator planning.[16] The collision avoidance or environment mapping ability is, to some extent, limited due to limited visibility. Therefore, for safe and complete sensing, it is inevitable to have omnidirectional and full body sensing capability. We believe that a sensitive skin type sensing technique is the only remedy for safe and complete sensing. To that end, we propose a fish-eye lens equipped 3D sensor system distributed around the entire body of the robot (see Fig. 2). Unlike other 3D depth technologies,[17,18] our sensing technology is based on real-time infrared photometry.[19,20] The monocular camera-based sensing device is tuned to be able to detect objects from 1 cm to 40 cm range. The ambient light rejection technique enables the sensor to react only to the light source of its own. The firing mechanism of each sensor has to be orchestrated to prevent crosstalk, but doable via frequency modulation. The configuration of sensor installation is another important issue for complete coverage of the robot

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

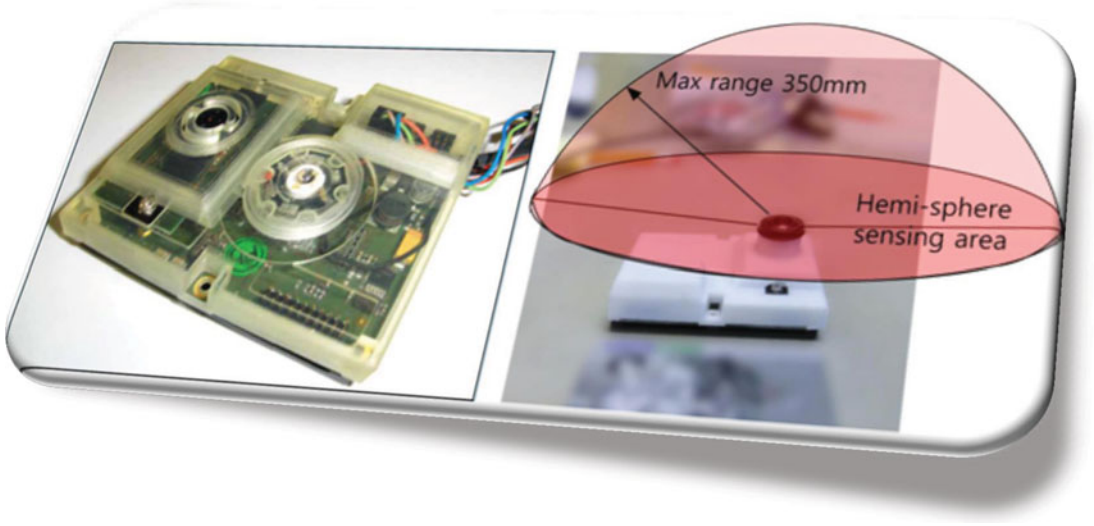

Fig. 2. 3D Depth sensor & sensing range.

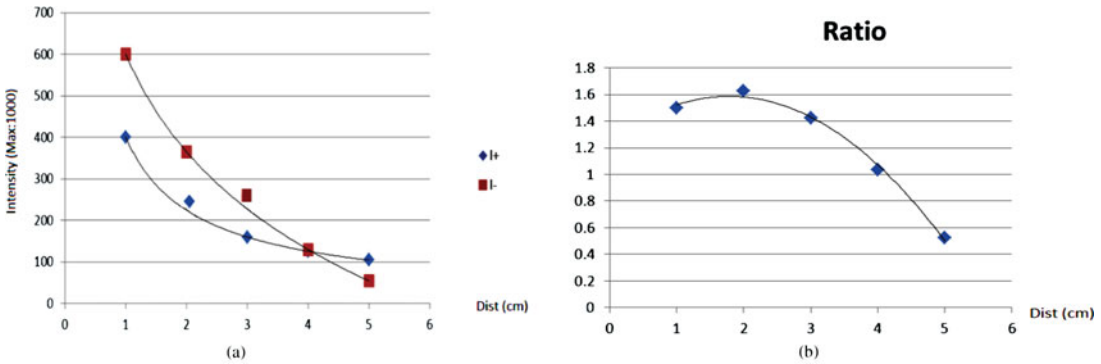

Fig. 3. Light intensity measure for two different light sources (a) and the intensity ratio curve (b).

body. In our experiment, we installed five sensors on the second link just for simplicity in experiments assuming no obstacle will be first faced by the first link.

The infrared intensity ratio-based 3D depth measurement technique we invented is, in principle, different to that of the time of flight (TOF)-based 3D depth measurement technique. First, two light sources with different characteristics are used to produce two different regression curves between the intensity ratio and the distance to an object. For instance, one light source with smaller half intensity angle (HIA) with higher intensity can produce a relatively linear regression curve, while another source with larger HIA with lower intensity produces an exponentially degrading curve as distance increases (see Fig. 3 (a)). If the intensity pattern, D, from respective intensity characteristics, I, and the distance to the object, d, is as below,

$$D_1(x, y, z) = \frac{K_{REFT}(x, y) \cdot I_1(x, y, z)}{d^2(x, y)} \tag{1}$$

$$D_2(x, y, z) = \frac{K_{REFT}(x, y) \cdot I_2(x, y, z)}{d^2(x, y)}, \tag{2}$$

then, by dividing two equations, the effect of local reflectibility $K_{REFT}(x, y)$ is cancelled, resulting in the following relation.

$$\frac{D_1(x, y, z)}{D_2(x, y, z)} = \frac{I_1(x, y, z)}{I_2(x, y, z)}. \tag{3}$$

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

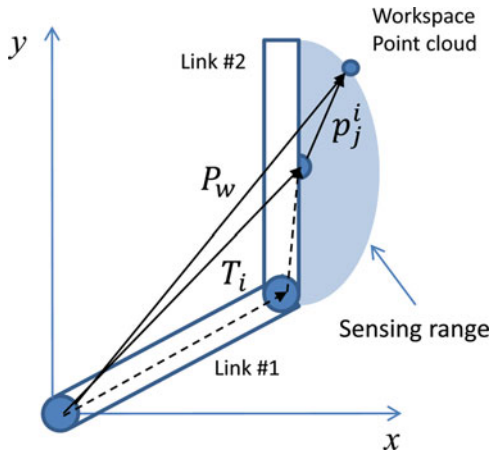

Fig. 4. Relationship between P_j^i and P_W.

Once the ratio curve is tabulated with a calibration data, the distance can be found easily with a polynomial equation from the ratio and the distance relationship. Depending on the light sources, there is a dead zone occurring in the sensing range. For instance, in Fig. 3, the ratio curve (b) is not a monotonic function in terms of distance. Therefore, it is important to create a unique combination of two different light sources to minimize the dead zone in the sensing range. Introduced sensing mechanism successfully demonstrated 3D depth sensing for non-contact-based collision checking with any object within the range of 45cm hemisphere range (see Fig. 4). The proof of sensing completeness of the proposed sensor is discussed in Section 3.

2.2. Creating a partial workspace from sensed data

The proposed sensing system is capable of mapping the surroundings of the manipulator in real time to a group of point clouds. Point cloud data sets corresponding to the sensed obstacle have to be transformed into global coordinates by forward kinematics such as

$$P_w = \bigcup_{i=1}^{N} p_j^i \cdot T_i, \tag{4}$$

where p_j^i is a point cloud data from sensor j situated at a link i, and T_i is the transformation matrix. N is the number of links of the manipulator. Therefore, P_w manifests the workspace in point cloud format revealed by 3D sensors.

In order to facilitate and expedite the c-space point cloud generation process, we assumed that workspace obstacles are all in spherical shape since the datum estimation of a sphere is simple and accurate. For instance, profile fitting for sphere surface datum for rotation is unnecessary. One easier fitting method of a sphere to point clouds is introduced in ref. [22]. That is, from analytic geometry, the equation of a sphere becomes

$$c1(x^2 + y^2 + z^2) + c2(x) + c3(y) + c4(z) + c5 = 0. \tag{5}$$

Using the following determinant equation, for instance, for four point cloud data sets, the sphere equation is found by resolving the determinant.

$$\begin{vmatrix} (x^2 + y^2 + z^2) & x & y & z & 1 \\ (x_1^2 + y_1^2 + z_1^2) & x_1 & y_1 & z_1 & 1 \\ (x_2^2 + y_2^2 + z_2^2) & x_2 & y_2 & z_2 & 1 \\ (x_3^2 + y_3^2 + z_3^2) & x_3 & y_3 & z_3 & 1 \\ (x_4^2 + y_4^2 + z_4^2) & x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0. \tag{6}$$

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

2.3. Creating a partial c-space via simulation

For a given partial workspace point cloud set, P_w, at an instance, we create a local c-space map using RRT (Rapid expanding Random Tree), one of probabilistic sampling algorithms, such that

$$B_{robot}, W \xrightarrow{RRT} P_c := \{p_m \in C^k \,|m = 1, \ldots, M\}, \tag{7}$$

where C is the c-space for the robot, k is the degree of freedom of the c-space, and M is the number of point cloud data produced by the mapping process. B_{robot} is the kinematic model of the robot and W is the workspace revealed up to the current cycle. In order to collect c-space point clouds, RRT takes B_{robot} and W as inputs and expands RRT tree to collect 'q (c-space configuration)' that causes collision with obstacles. The robot has to stop its motion during partial c-space creation by the RRT technique with the partially created virtual workspace. In order to expedite the mapping process and to minimize abrupt discontinuities in motion due to lengthy online simulation, we use a simplified two linkage model of the robot for collision check. In Section 5, an experiment using a six-DOF robot with only the first three-DOF enabled is introduced. Algorithm 1 below represents a method as to how to map a partial c-space by the RRT technique. First, for an empty point cloud set, Λ_n, RRT grows a tree collecting 'q' of all collision free configurations. During the pocess, 'q' that causes collision is explicitly collected as a c-space point cloud. P_c, the c-space point cloud set, grows as such 'q' is added into the set. We check the Gaussian density distribution of the c-space point clouds to terminate the mapping process at each stage by Eq. (8). q_{mean} in Eq. (8) is the mean value of all configurations.

$$\sigma_d = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|q_i - q_{mean}\|^2}. \tag{8}$$

Algorithm 1 Partial c-space mapping

q: c-space configuration
Λ_n: collection of c-space point clouds
T_{RRT}: RRT Tree
$B_{ROBOT}(q)$: workspace robot model for given configuraiton by q
W: Partial workspace
P_W: workspace point clouds
P_C: c-space point clouds
δ_Λ: threshold value of the Gaussian density distribution
$\Lambda_n = \{\}$;
do while $\sigma_d(\Lambda_n) > \delta_\Lambda$
$q \leftarrow$ Grow T_{RRT} /* by RRT (qn, qgoal)−Planning*/
$\Lambda_n = \Lambda_n \bigcup q$ /*Planning*/
$W \leftarrow P_W$ /* by base-level posture estimation */
if $B_{ROBOT}(q) \cap w \neq 0$ /* if robot collides with P_W − c-space creation */
$P_C = P_C \bigcup q$ /* then collect c−space point cloud−c−space creation*/
end if
end do
return (T_{RRT}, P_C)

2.4. Probabilistic motion planning in a partial c-space

We use biased greedy search as motion planning strategy. SBIP is related to online replanning algorithms such as Dynamic A* Lite by Koening and Likhachev where an incremental heuristic search algorithm is presented.[23]

Algorithm 1 in the previous section shows the mapping algorithm, while Algorithm 2 represents the motion planning strategy. The function $d(,)$ returns n-dimensional Euclidian distance. The planning strategy in Algorithm 2 is a simple greedy search algorithm biased toward the goal. If the goal configuration is still far in distance, then the planner requests newly collected workspace point clouds. After updating the workspace point clouds, the planer calls for mapping of a partial c-space

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

Algorithm 2 SBIP path planning

q_i: initial c-space configuration
q_{goal}: final c-space configuration
q_{random}: random c-space configuration
$T_{RRT} = \{\}$; /*Initialize RRT Tree*/
$P_W = \{\}$; /*Initialize workspace point cloud*/
$P_C = \{\}$; /*Initialize configuration space point cloud*/
$q_i = \emptyset$;
do while $d(q_i, q_{goal}) > \varepsilon$
Collect P_s^i /*using sensitive skin − workspace creation*/
$P_W = P_W \cup \sum_{i=1}^{N} \left(P_s^i \cdot T_i\right)$/*update workspace point cloud−workspace creation*/
Call Algorithm 1(T_{RRT}, P_W, P_C, q_i, q_{goal})/* update RRT Tree and P_C */
Pick q_i:$d(q, q_{goal}) > d(q_i, q_{goal})$, $q \in P_C$ /*greedy global search planning*/
If (*local minima*)
Pick q_i:$q_i = q_{random} \in T_{RRT}$/*planning*/
end if
end do

by collecting c-space point clouds with RRT expansion. The function 'Local minima' returns positive if the same q is multiply selected. A random configuration will be picked up from the current tree to avoid local minima.

Rusu et al. proposed a 3D perception-based replanning architecture using a voxel-based collision map constructed from a laser range-finder.[24] For the replanning, they showed an offline replanning method using a sampling based planner. In contrast, SBIP combines an online replanning method with a probabilistic search-based planner to enable real-time planning in unknown environments.

3. Search Completeness of SBIP

Now we discuss the probabilistic completeness of the SBIP planner, specifically to show that SBIP is guaranteed to eventually find a path from an initial configuration to a final configuration, if a tractable path exists.

Definition:
q: a random configuration or a node in the expanded tree.
W_{Total}: a 3D occupancy grid for the workspace with discretization at resolution r. Therefore, an element voxel of the grid is occupied when a random configuration, q, belongs to the element.

$$W_{Current} := \{W_1, W_2, \ldots, W_{i-1}, W_i, \}.$$

We also define,

$$Q_T = \text{RRT}(W_{Total,} q_{init}, q_{goal}),$$

where RRT is a probabilistic planner with proven completeness, where Q_T is a c-space trajectory that, when followed, takes the robot from q_{init} to q_{goal} without collision for a given W_{Total}.

Suppose W_i is a subset of W_{Total}. At each iteration of SBIP, the robot uses its current map to plan a trajectory as follows:

$$Q_i = \text{RRT}(W_i, q_i, q_{goal}),$$

where q_i is a node closest to the goal by global greedy search algorithm picked up at each sequence.

Let us also define

$$Q_0 = \text{RRT}(W_0, q_0, q_{goal}),$$

where $q_0 = q_{init}$ and $W_0 = \emptyset$, whose cardinality is equal to zero, or $|W_0| = 0$. W_0, however, becomes W_i as the workspace is partially created by the sensed point clouds.

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

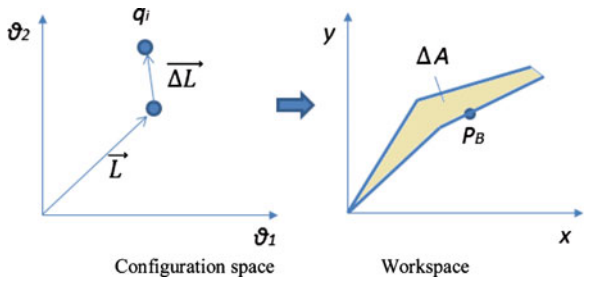

Fig. 5. Mapping between configuration space and workspace. Configuration space Workspace.

Assumption:
W_i is the subset of W_{Total}, therefore the maximum cardinality of $W_{Current}$ is equal to that of W_{Total}.

Each W_i is connected.

Lemma 1. SBIP is a complete planner.

Proof. *RRT* is guaranteed to return a trajectory, Q_i, from the robot's current configuration, q_i, to the goal, q_{goal}, if a tractable path exists, otherwise, halts if the point cloud density by Eq. (9) for a given W_i exceeds a limit. Since W_i is the element of W_{Total}, and each W_i is connected, the robot can retune to q_{init} anytime necessary. In addition, even if the planner halts at a boundary of $W_{Current}$, it adds new voxels from the sensed point clouds, thus the cardinality of $W_{Current}$ increases after each iteration and maximum cardinality of $W_{Current}$ becomes $|W_{Total}|$. Then following is true.

$$\text{If } |W_{Current}| = |W_{Total}|\,, \text{ then } {}^{\exists}Q\,|Q = \text{RRT}(W_{Current}, q_{init}, q_{goal}).$$

Therefore, SBIP is guaranteed to reach q_{goal}. Since the proof did not use the fixed resolution, it holds for any resolution for which a solution trajectory exists. □

One more assumption necessary to ensure the SBIP a complete planner is that the robot perfectly senses and reveals the complete partial workspace at each step. That is, W_i, the workspace constructed at each step of the search operation, should be a complete model matching the current environment, otherwise the condition '$|W_{Current}| = |W_{Total}|$' cannot be met and RRT may not be able to connect q_{init} and q_{goal} although there is a tractable path exists. Therefore, sensing completeness at each stepwise motion has to be guaranteed for probabilistically complete manipulation in a cluttered, unknown environment.

Let us consider a mapping between a workspace and a configuration space for a two bar linkage robot. The planner RRT grows a tree by selecting an existing node and grows a branch in random direction by $\overrightarrow{\Delta L}$, a non-zero stepwise movement, in c-space, while $\overrightarrow{L}$ is the location vector of the current node in the tree.

As shown in Fig. 5, a step motion in c-space causes the robot to sweep a non-zero area (ΔA) in workspace, where P_B is an arbitrary point on the boundary or

$$S_B = \{P_B\,|P_B \in \partial \Delta A\}.$$

Therefore, if q_i is a collision free configuration in c-space, then the robot creates non-zero swept area by the robot in workspace. That is,

$$\overrightarrow{\Delta L} = 0 \leftrightarrow \text{ obstacle is on the path and } \Delta A = 0,$$
$$\overrightarrow{\Delta L} \neq 0 \leftrightarrow \text{ path is free of obstacle and } \Delta A \neq 0.$$

In reverse mapping, if $\Delta A \neq 0$, then the planner can create a corresponding vector, $\Delta \overrightarrow{L}$ in c-space, thus RRT can expand the tree. The reverse mapping is the primary mechanism for RRT expansion by

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

collision checking in workspace. However, if $\Delta A = 0$, then following is true.

$$\Delta A = 0 \leftrightarrow {}^{\exists} P_B \,|P_B \in !W_F,$$

where W_F is collision free workspace. Now we define P_R, a set of points on the surface of the robot body, R, such that,

$$P_R \in \partial R(\theta_1, \theta_2) \text{ or } \quad P_R \in \partial R(\vec{L}).$$

And assume

$$S(P_R) \text{ returns '0' if } \quad P_R \in W_F, \text{ otherwise '1'}.$$

That is,

$$S(P_R) = \text{'0'} \quad \text{if } P_R \in W_F, S(P_R) = 1 \text{ if } \quad P_R \in !W_F.$$

Then, following lemma is true.

Lemma 2. ${}^{\exists}\overrightarrow{\Delta L}|\overrightarrow{\Delta L} \neq 0$ *iff* $\{S(P_R)|P_R \in \partial R(\overrightarrow{L} + \overrightarrow{\Delta L})\} = \emptyset$.

Proof. If we define,

$S_R = \{P_R|\ P_R \in \partial R\}$

then

${}^{\exists}P_R$: $P_R = P_B$ if $\theta_1, \theta_2, ..., \theta_n = \varepsilon$, where ε is non-zero, infinitesimally small value.

and

$S_B \leftrightarrow S_R$ for small changes in $\theta_1, \theta_2, ..., \theta_n$

That is, S_B and S_R are in bijective functional relationship for a small motion. Since,

$${}^{\exists}P_B \,|P_B \in !W_F \rightarrow \Delta A = 0\ .$$

Therefore,

$${}^{\exists}P_R \,|S(P_R) = 1 \rightarrow \Delta A = 0\ .$$

Furthermore, since

$$\Delta A \neq 0 \rightarrow \Delta \vec{L} \neq 0.$$

Therefore,

$$\text{For} \forall P_R \left| P_R \in \partial R(\vec{L} + \overrightarrow{\Delta L})\ , \quad \text{if } {}^{\exists}P_R \,|S(P_R) = 1 \rightarrow \overrightarrow{\Delta L} = 0. \right.$$

This proves Lemma 2. □

In conclusion, for the SBIP to be able to expand a tree in certain direction in an unknown environment, entire body of the robot has to be free of collision at the configuration designated by $\overrightarrow{L} + \overrightarrow{\Delta L}$, in c-space. This implies that every point on ∂R has to be sensitive.

Definition: A sensor has sensing completeness if,

$$\text{For} \forall P_R \left| P_R \in \partial R(\vec{L} + \overrightarrow{\Delta L})\ , \quad S(P_R) = 0 \text{ if } P_R \in W_F, \quad S(P_R) = 1 \text{ if } P_R \in !W_F. \right.$$

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

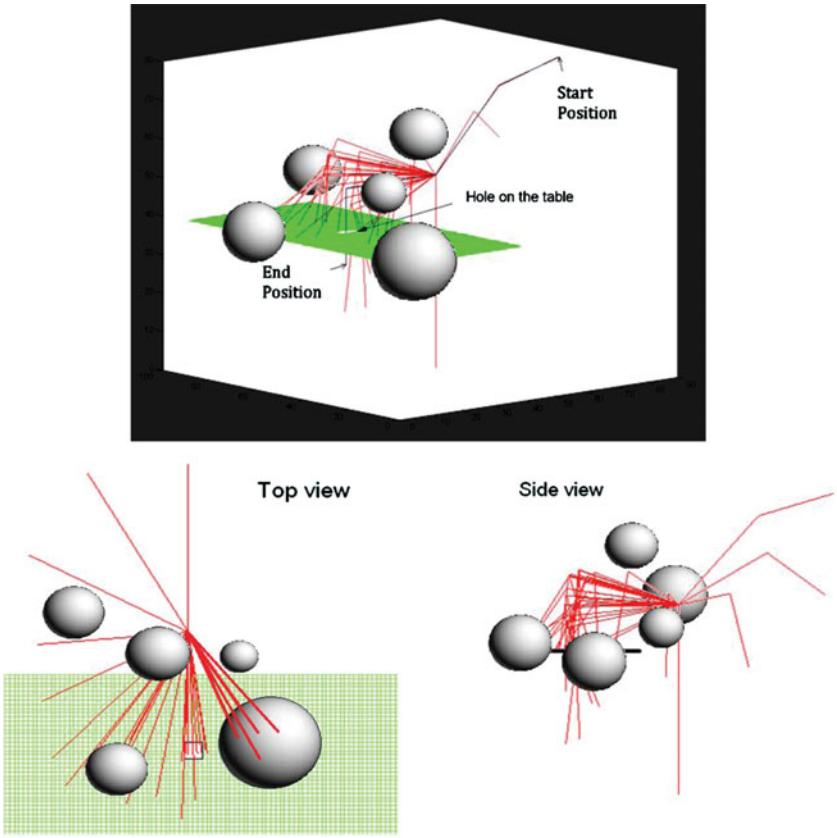

Fig. 6. SBIP for three-DOF Manipulator (RRR) planning.

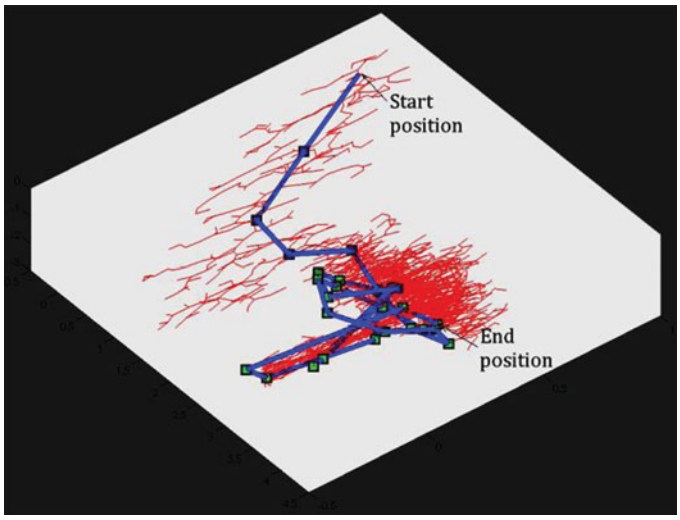

Fig. 7. Corresponding 3D C-space with planning trajectory.

Above definition implies that the robot should be able to estimate collision with any obstacle throughout the entire body, otherwise tree expansion for online c-space creation is conclusively impossible. There are only a few sensor types that are possibly claimed to have sensing completeness. One is a sensitive skin type sensor[25,26] and another is a haptic skin sensor introduced in.[11,27] As for the haptic skin sensor though, the nature of contact-based sensing may be used with reservation in certain applications. Other single sensor-based approaches such as eye-in-hand, ceiling cameras, or a laser depth sensor cause visual occlusion, resulting in loss of spatial information of the workspace. As

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

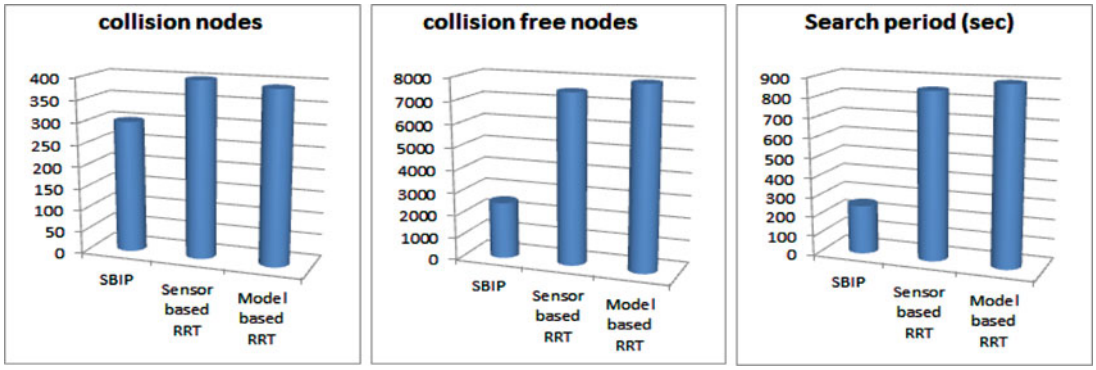

Fig. 8. Performance indices.

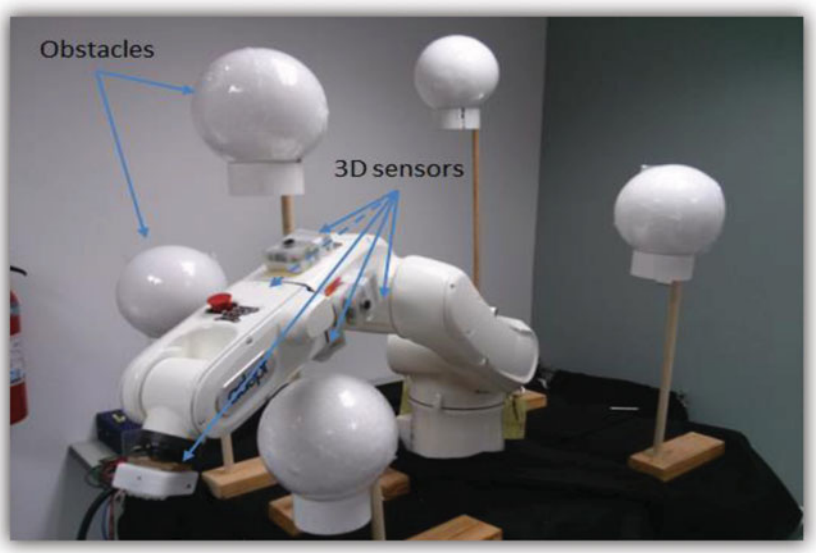

Fig. 9. 3D sensor installation on a three-DOF robotic linkage.

a result, for search completeness of SBIP, sensing completeness has to be guaranteed. This justifies the use of sensitive skin type non-contact sensors for manipulators operating in a complex, clustered environment.

4. Simulation Results

In order to test the proposed algorithm, we setup a three-DOF revolutionary joint robot as a testbed for simulation (see Fig. 6). The robot needs to avoid collision with all floating obstacles and reach the goal under a table with a hole, through which is the only way to access the goal. For simplicity in simulation, we applied non-zero thickness on the robot's links to mimic depth sensing to detect collision. Figure 7 shows the result of the path search in joint space by the SBIP.

For comparison, a sensor-based RRT planner and a model-based RRT planner[13] are put into simulation with the same environment. For 30 runs of each simulation, some statistics are shared in Fig. 8. Total search time is the time in seconds from the beginning to the end of the search operation. Number of P_c stands for the total number of c-space point clouds. The number of collision free nodes is dramatically reduced for SBIP while not much of gain on the total collision nodes is reported. This is primarily due to the nature of greedy global planning strategy. Crowded but relatively smaller size of obstacles in the workspace would be another reason. As far as the number of collision free nodes is concerned, balance between random search and goal-oriented greedy search should be considered in case of complex environments with multiple local minima.

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

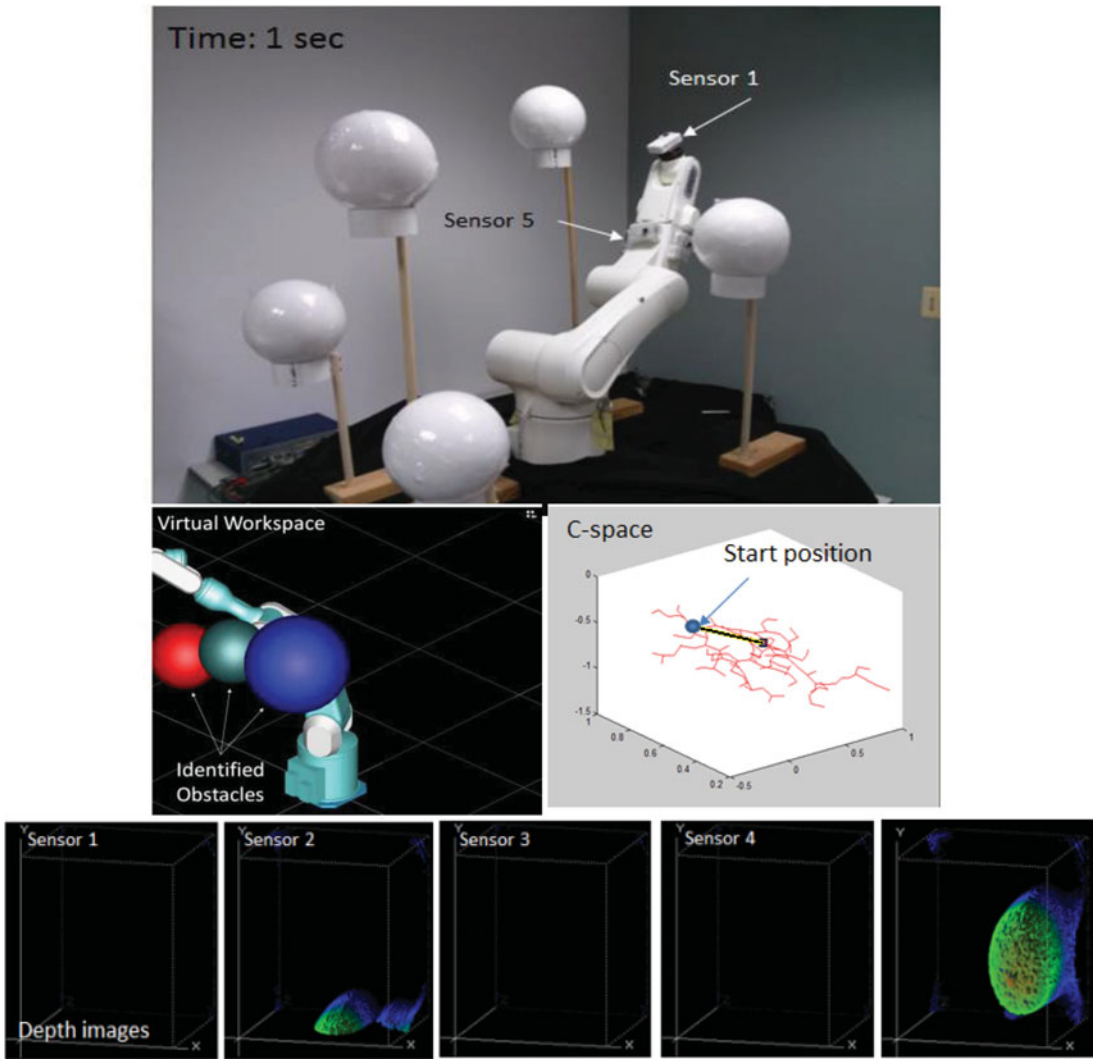

Fig. 10. Robot configuration, virtual workspace, and c-space generation at 0.1 s elapsed.

The sensor-based RRT has a tendency of following the surface of obstacles by calculating the density gradient of *k*-nearest neighborhood (See ref. [28] for more detail). Therefore, the number of collision nodes exceeds that of model-based RRT. Another result noticeable is the significant performance gain of the SBIP in search time compared to that of both sensor-based RRT and model-based RRT. Again, the greedy global planning strategy and convex shape of obstacles in conjunction with successful local minima avoidance nature of probabilistic planning are the main reasons.

5. Experiments

The target system for experiments is shown in Fig. 9 with five 3D sensors covering the second joint for collision shield formation. The robot used for experiments is a six-DOF Adept viper manipulator which is chosen primarily because of high-speed motion capability by low-inertia harmonic drives and a lightweight arm to achieve maximum acceleration in each step motion. 8 kHz servo update rate via its integrated 10/100 Base-T Ethernet for improved path following and control is another reason. Multiple 3D sensors used to form sensitive skin around the body of manipulator are from DYNAST. Cyclops II 3D sensors offer better ambient light noise control in short range for both indoor and outdoor, so it provides reliability in indoor experiments. Each sensor describes point clouds in its own local coordinate. Therefore, all of the point clouds have to be transformed to the global coordinate to describe a partially revealed workspace. Transformation from the local coordinates to the global coordinate for each point cloud is performed by Eq. (1).

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

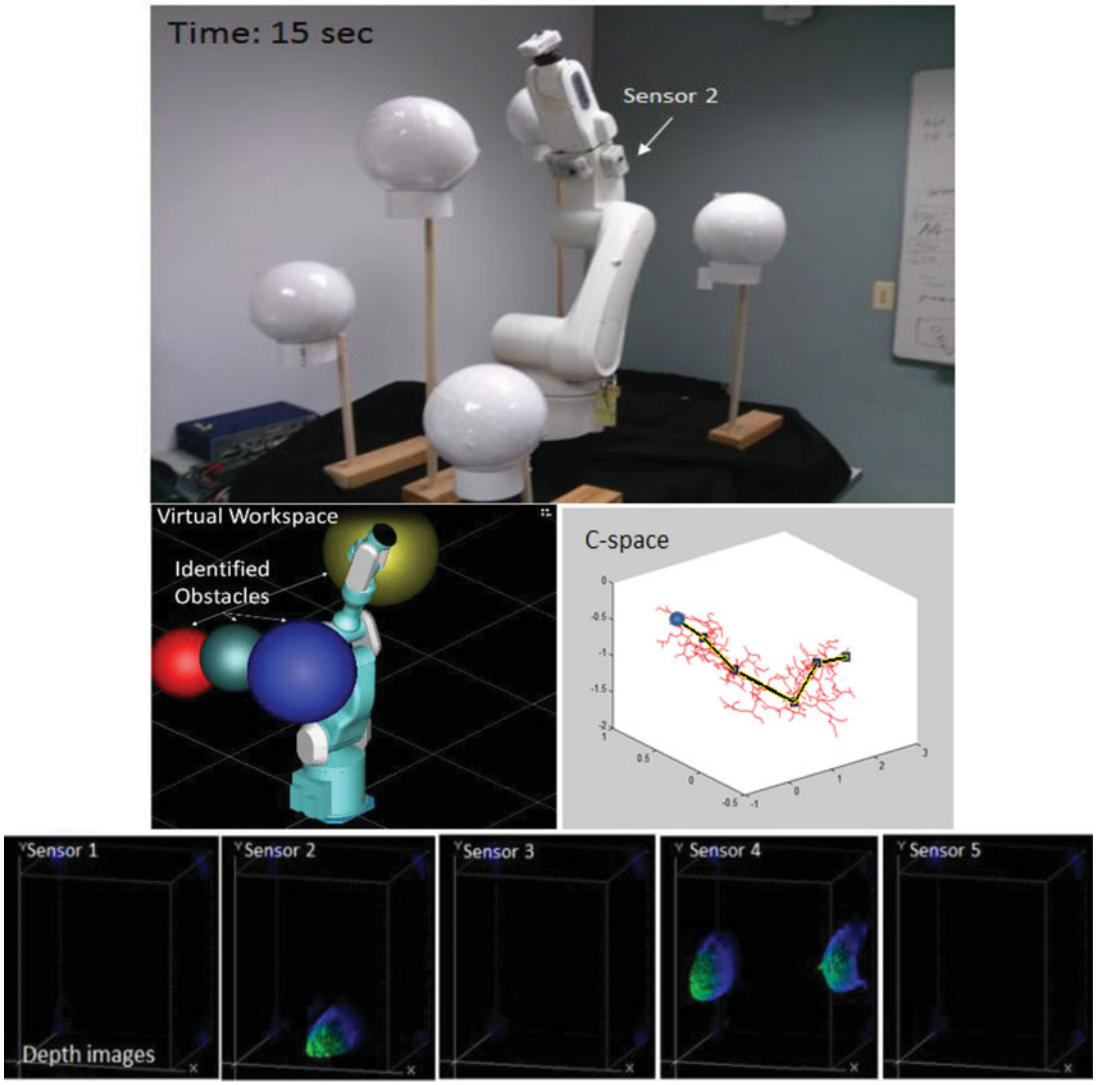

Fig. 11. Robot configuration, virtual workspace, and c-space generation at 15 s elapsed.

In this experiment, we assumed that the shape of the obstacles is known to be sphere in the workspace just for the sake of convenience in experiments. This may seem defeating the purpose of the paper on the first hand. However, since the only purpose of assumed shape and size of obstacles is to accelerate workspace creation, collision check and faster visualization, the assumption does not disqualify the proof of usefulness of the proposed planer.

Figures 10, 11, and 12 are snapshots of the search operations by SBIP algorithm. The same robot is created in the virtual workspace for online c-space mapping by the RRT algorithm. In each snapshot, there are five pictures at the bottom to represent the workspace point clouds for the detected obstacles by five 3D cameras. During the c-space mapping stage, the robot becomes stationary since non-zero time is required to grow a RRT for c-space point cloud collection. Each c-space point cloud is collected whenever the virtual robot collides a work space obstacle as the RRT grows in c-space. The black dots in each c-space represent the result of motion planning by the greedy global search strategy.

Time analysis of each thread that constitutes the overall SBIP is performed. Ten to twelve steps are observed during the multiple planning experiments. The frequency of each cycle may be adjusted by spatial density distribution in Algorithm 2. As shown in Fig. 13, the total planning time for each cycle is below 4 s at early stages, but it exponentially grows as the robot faces more obstacles and accumulates more point clouds in c-space.

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

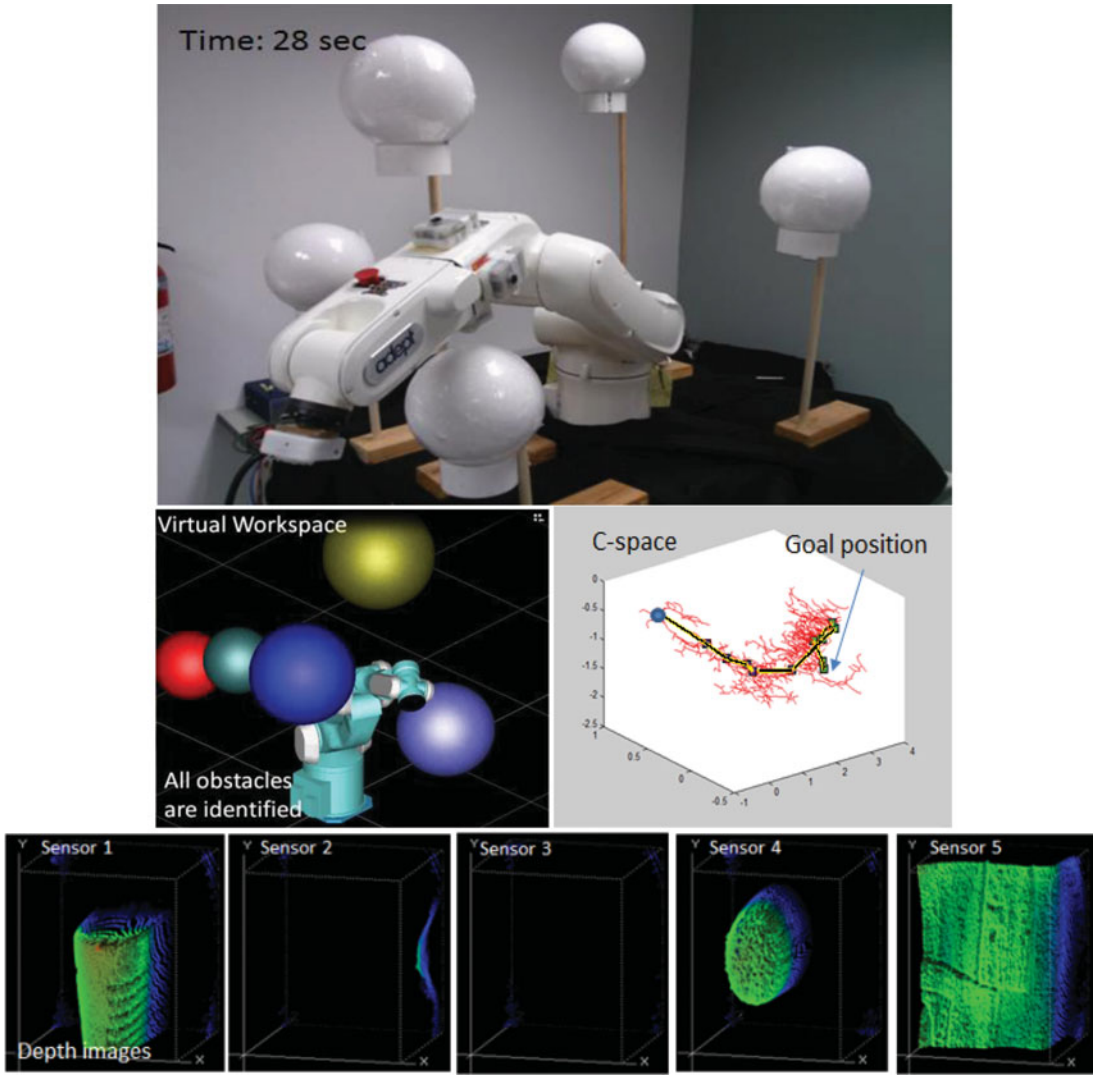

Fig. 12. Robot configuration, virtual workspace, and c-space generation at 28 s elapsed.

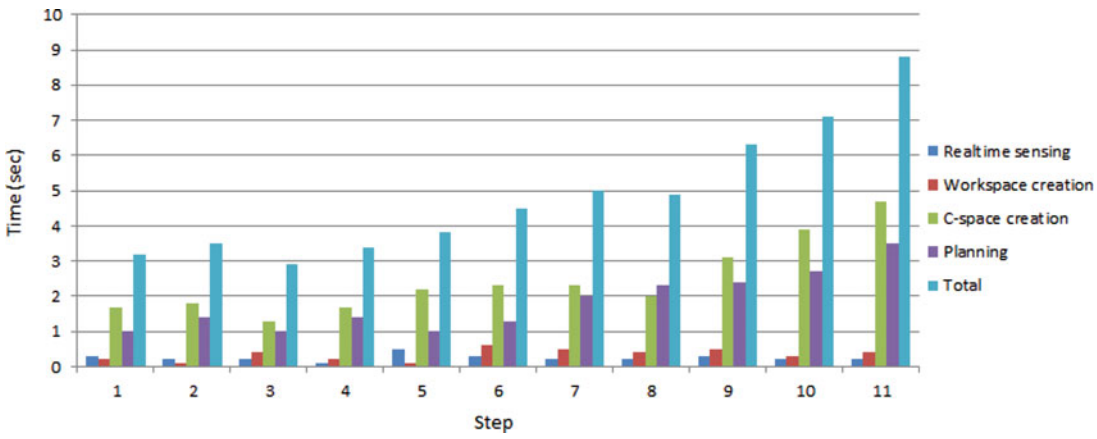

Fig. 13. Time analysis of four threads.

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

Since the bottleneck of the overall SBIP approach proposed in the paper is the real-time simulation for c-space point cloud collection at each stepwise local motion, high end computation capability will further streamline the global motion of the robot. If then, more complex planning problems such as dual arm path planning or humanoid locomotion planning with dynamic constraints in unknown environments could be tackled with the proposed planner and system setup.

One possible improvement to reduce the total search time other than the use of high power computing is to minimize the number of point clouds in workspace creation. At the moment, since we keep adding all of the sensed point clouds to create the workspace, the number of workspace point clouds increases linearly although the sensed number of objects remains the same. For instance, if we only register the point cloud whose closest neighborhood is at certain distance, we can keep the minimum number of point clouds for the workspace without diminishing the spatial resolution, thus the c-space creation cycle time will be minimized. However, calculation for identifying the closest neighborhood for each point cloud may not be trivial.

6. Conclusion

In this paper, we introduced a framework of SBIP for safe manipulation in cluttered, unknown environments. The goal of the study was to move a multi-degree of freedom robot safely in a cluttered and uncertain environment such as domestic or unstructured industrial workspaces. The focus of study was on the question as to how to guarantee both *search completeness* and *safe manipulation*. We propose a non-contact-based sensitive skin as a means to provide safe manipulation hardware and an interleaving planner between a workspace and a configuration space as software. We found that real-time simulation for c-space point cloud collection is the bottleneck of the total planning performance. In addition, as it is shown in the proof of the probabilistic completeness of the SBIP, search completeness in an unknown environment cannot be achieved without sensing completeness in each partial workspace. To that end, we proposed a sensitive skin type 3D depth sensor that encompasses the entire body of the robot. Sequential interleaving takes place to gradually but eventually reveal the entire configuration space by means of complete sensing at each cycle thus it is probabilistically complete planner.

Unlike the SLAM, localization is not a critical issue in manipulation planning, but gradual and complete mapping at each step is needed to guarantee search completeness. Interleaving between a workspace and a configuration space is the main idea for mapping and convergence. The proposed algorithm will enable a smart move of a manipulator such as dual arms or a humanoid in uncertain or crowded environments.

References

1. A. Orthey and O. Stasse, "Towards Reactive Whole-Body Motion Planning in Cluttered Environments by Precomputing Feasible Motion Spaces," *Proceedings of* 13*th IEEE-RAS International Conference on Humanoid Robots*, (Humanoids), Atlanta, GA (Oct. 15–17, 2013) pp. 274–279.
2. A. Hussain, I. Qureshi, K. F. Qamar, S. M. F. Islam, Y. Ayaz and N. Muhammad, "Potential Guided Directional-RRT* for Accelerated Motion Planning in Cluttered Environments," *Proceedings of IEEE International Conference on Mechatronics and Automation*, Takamatsu, Japan (Aug. 4–7, 2013) pp. 519–524.
3. Z. Shiller and S. Sharma, "High Speed On-Line Motion Planning in Cluttered Environments," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal (Oct. 7–12, 2012) pp. 596–601.
4. A. Hornung, S. Bottcher, J. Schlagenhauf, C. Dornhege, A. Hertle and M. Bennewitz, "Mobile Manipulation in Cluttered Environments with Humanoids: Integrated Perception, Task Planning, and Action Execution," *Proceedings of the Prof. of 14th IEEE-RAS International Conference on Humanoid Robots*, Humanoids, Madrid, Spain (Nov. 18–20, 2014) pp. 773–778.
5. N. Kitaev, I. Mordatch, S. Patil and P. Abbeel, "Physics-Based Trajectory Optimization for Grasping in Cluttered Environments," *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Washington State Convention Center Seattle, Washington (May 26–30, 2015) pp. 3102–3109.
6. J. A. Ambros-Ingerson and S. Steel, "Integrating Planning, Execution and Monitoring," *Proceedings of in AAAI*, vol. 88, Saint Paul, Minnesota (Aug. 21–26, 1988) pp. 21–26.
7. I. R. Nourbakhsh, *Interleaving Planning and Execution for Autonomous Robots* (Kluwer Academic Publishers, Massachusetts, 1997).

https://doi.org/10.1017/S0263574715001101 Published online by Cambridge University Press

8. L. Kaelbling and T. Lozano-Perez, "Hierarchical Task and Motion Planning in the Now," *Proceedings of Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China (May 9–13, 2011) pp. 1470–1477.
9. C. S. Silva, A. M. Bernardino and C. Pinto-Ferreira, "Interleaving Real-Time Multi-Agent Planning and Execution: An Application," *Proceedings of 6th. Conf. on Tools with Artificial Intelligence*, New Orleans, Louisiana (Nov. 1994) pp. 6–9.
10. L. de Silva, A. K. Pandey and R. Alami, "An Interface for Interleaved Symbolic-Geometric Planning and Backtracking," *Proceedings Int. Conf. Intelligent Robots and Systems (IROS)*, Tokyo Big Sight, Japan (Nov. 3–8. 2013) pp. 3–7.
11. A. Jain, M. D. Killpack, A. Edsinger and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *Int. J. Robot. Res.* **32**(4), (2013) pp. 458–482.
12. L. Kavraki and J.-C. Latombe, "Randomized Preprocessing of Configuration Space for Fast Path Planning," *Proceedings Int. Conf. on Robotics & Automation*, San Diego (May 1994) pp. 2138–2145.
13. S. M. LaValle and J. J. Kuffner, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, San Francisco, CA (Apr. 2000) pp. 995–1001.
14. C. S. Zhao, M. Farooq and M. M. Bayoumi, "Analytical Solution for Configuration Space Obstacle Computation and Representation," *Proceedings IEEE 21st Int. Conf. on Industrial Electronics, Control, and Instrumentation*, vol. 2, Piscataway, NJ (Nov. 6–10, 1995) pp. 1278–1283.
15. Y. Yu and K. Gupta, "C-space entropy: A measure for view planning and exploration for general robot-sensor systems in unknown environments," *Int. J. Robot. Res.* **23**(12), 1197–1223 (Dec. 2, 2004).
16. K. Kieda, H. Tanaka, and T. Z. Zhang, "On-line Optimization of Avoidance Ability for Redundant Manipulator," *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, Beijing, China (Oct. 9–15, 2006) pp. 592–597.
17. F. R. Livingstone, L. King, J.-A. Beraldin and M. Rioux, "Development of a Real-Time Laser Scanning System for Object Recognition, Inspection, and Robot Control," *SPIE Proceedings, Telemanipulator Technology and Space Telerobotics*, vol. 2057, Boston, Massachusetts, (Sep. 7–10, 1993) pp. 451–461.
18. T. Ringbeck and B. Hagebeuker, "A 3D Time of Flight Camera for Object Detection," *Optical 3-D Measurement Techniques*, ETH Zürich Plenary Session 1: Range Imaging, Jul., 2007.
19. T. Malzbender, B. Wilburn, D. Gelb and B. Ambrisco, "Surface Enhancement Using Real-Time Photometric Stereo and Reflectance Transformation," *Proceedings of the 17th Eurographics conference on Rendering Techniques*, Cyprus (Jun. 26–28, 2006) pp. 245–250.
20. D. Um, D. Ryu and M. Kal, "Multiple intensity differentiation for 3D surface reconstruction with mono-vision infrared proximity array sensor," *IEEE Sensors J.* **11**(12), 3352–3358 (Jun. 2011).
21. G. Sergio, "Fitting Primitive Shapes to Point Clouds for Robotic Grasping," *KTH Computer Science and Communication*, Master of Science Thesis, Stockholm, Sweden (2009).
22. S. Koenig and M. Likhachev, "D* Lite," *Proceedings of 14th Conf. on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, Alberta, Canada (Jul. 28–Aug. 1, 2002) pp. 476–483.
23. R. Rusu, I. Sucan, B. Gerkey, S. Chitta, M. Beetz and L. Kavraki, "Real-Time Perception-Guided Motion Planning for a Personal Robot," *Proceedings Int. Conf. Intelligent Robots and Systems (IROS)*, St. Louis, MO (Oct. 11–15, 2009) pp. 4245–4252.
24. D. Um, B. Stankovic, K. Giles, T. Hammond and V. Lumelsky, "A Modularized Sensitive Skin for Motion Planning in Uncertain Environments," *Proceedings Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, Belgium, (May 1998) pp. 7–12.
25. D. Um, "How to tackle sensor based manipulator planning problems using model based planners: Conversion and implementation," *Int. J. Robot. Autom.* 24(2), 137–146 (2009).
26. D. Park, A. Kapusta, J. Hawke and C. C. Kemp, "Interleaving planning and control for efficient haptically-guided reaching in unknown environments," *Proceedings 14th IEEE-RAS Int. Conf. on Humanoid Robots* (Humanoids), Madrid, Spain (Nov. 18–20, 2014) pp. 809–816.
27. D. Um, "How to tackle sensor based manipulator planning problems using model based planners: Conversion and implementation," *Int. J. Robot. Autom.* (Impact Factor: 0.206), 24(2), 137–146 (2009).