View Planning via C-space Entropy for Efficient Exploration with Eye-in-Hand Systems

Yong Yu	Kamal K. Gupta
yongyu@cs.sfu.ca	kamal@cs.sfu.ca
School of Engineering Science	School of Engineering Science
Simon Fraser University	Simon Fraser University
Burnaby, B.C. V5A 1S6 Canada	Burnaby, B.C. V5A 1S6 Canada

Abstract: We present an implemented sensor-based planner for motion planning and exploration for eye-in-hand systems. A model-based motion planner is used to plan paths within the known part of the environment to further sense the unknown part of the environment. Each sensing action is viewed as gaining information about the status of configuration space. We introduce the notion of C-space entropy as a measure of ignorance or lack of information of C-space. The next view is planned so as to maximize expected entropy reduction (MER), or equivalently, expected information increase. Experimental results demonstrate that MER criterion results in efficient exploration of unknown environments and that the planner can make a robot arm move around safely (without collisions) while carrying out exploratory and purposive tasks in unknown environments.

1. Introduction

In sensor-based motion planning, the task for a robot, equipped with a sensor, is to sense and explore its unknown environment (and reach a given goal) while avoiding collisions with any obstacles in the environment. A variety of sensors have been used in the past, mostly with mobile robots, with some exceptions as in [1]. This paper falls in the category of eye-sensor based motion planning for robots with non-trivial geometry/kinematics. An "eye" sensor is essentially a distance or range sensor that senses distances from a single vantage point (or a reference frame); non-trivial[1] geometry/kinematics implies that the physical space and C-space for the robot are distinctly different. This class of robots is broad and includes robots ranging from a simple polygonal mobile robot to complex articulated manipulators, and humanoid robots [2]. In particular, our concern here is motion planning and exploration for an eye-in-hand system – a manipulator arm (called robot from here on), equipped with an eye type sensor as shown in Figure 1. The robot is required to plan and execute collision-free motions in an environment initially unknown to it. The decision to mount the sensor (eye) on the robot end-effector (hand) was motivated by the additional

[1]Idealizations such as point or circle robots, made for most mobile robots, are considered to have trivial geometry/kinematics.

maneuverability for the sensor. Our framework, however, is general and could be extended to other types of sensor+robot mechanisms, for instance humanoid robots, where the eye sensor(s) is located on the head [2]. See [3] for some interesting issues for sensor-based planning with eye sensors. Figure 2 shows

Figure 1. The experimental eye-in-hand system — a PUMA 560 with a wrist mounted area-scan laser range finder. Inset shows an enlarged view of the sensor with the camera on the left and the laser striper on the right.

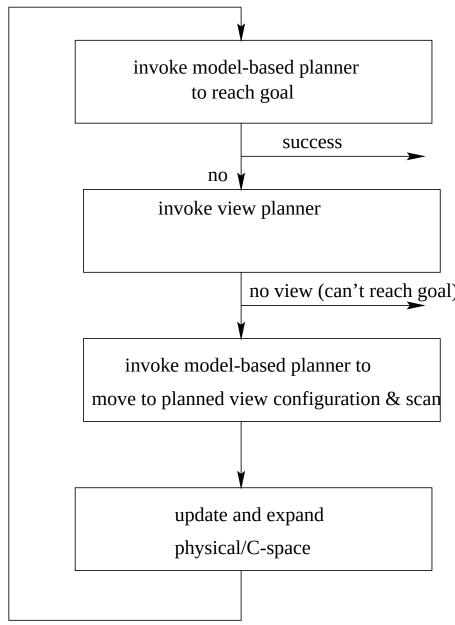


Figure 2. A general framework for eye-sensor based MP.

our "high-level" framework for eye-sensor based MP for robots with non-trivial geometry/kinematics. The sensing action is assumed to be discrete, i.e., it take place at discrete time instants. This framework assumes that a model-based[2] planner is available. This underlying model-based planner is used to plan paths within the known part of the environment to further sense the unknown part of the environment. We call this problem of determining the next sensing action the view planning problem. It has two component sub-problems: a) determine which region (in physical space) to scan and b) determine a reachable robot configuration from where the scan can be taken. The robot invokes the model-based planner to move to the planned view configuration and takes a scan. Having taken the new scan, the robot "updates" its internal representation (of physical and configuration space) and re-invokes the model-based planner to reach the goal. If it succeeds, it terminates with success. Otherwise, this process is repeated until the given goal is declared reachable by the model-

[2]We use the term model-based motion planning when the environment (model) is assumed known [4].

based planner, or the goal is declared unreachable[3]. The two key sub-problems that sensor-based MP therefore needs to solve are: (i) *view planning*, and (ii) *an "incrementalized" model-based planner* that repeatedly updates its physical and configuration space representation.

Within the framework described above, we have implemented a sensor-based planner for the real eye-in-hand test-bed system shown in Figure 1. The model-based planner component used in our overall algorithm is an incremental version of the probabilistic roadmap planner [5], adapted to deal with unknown environments. We call it Sensor-based Incremental Construction of Probabilistic Roadmaps (SBIC-PRM). Since a roadmap is constructed without explicitly representing the C-obstacles, it is particularly useful as a representation for high dimensional configuration spaces, our main motivation behind using this approach. While other roadmap type approaches could also be used, a key advantage of PRM is that it naturally handles disconnected components in the roadmap. See [6] for the case when ACA is used as the underlying model-based planner within our framework.

We have reported various aspects of SBIC-PRM in [7, 8]. This paper integrates a novel, systematic and more efficient information-theoretic approach to the view planning sub-problem within the overall SBIC-PRM planner. In this approach each sensing action is viewed as gaining information about the status of configuration space. It uses a novel notion of C-space entropy as a measure of ignorance or lack of information of C-space. The next view is planned so as to maximize expected entropy reduction (MER), or equivalently expected increase in information. The details of this information theoretic view planning approach are described in [9, 10]. The emphasis in this paper is on its integration within the overall SBIC-PRM planner and experimental results showing the efficacy of MER based view planning compared to other criteria.

Figure 3 shows SBIC-PRM running on a simulated 2-link eye-in-hand system. The robot has a range sensor on the end-effector. The sensor has a field of view of 40°, indicated by the triangular region. The sensor also has an additional rotational degree of freedom at the wrist. The left image in each column shows the physical space and the right image shows the corresponding C-space. In both physical space and C-space, white region is known free space, black region is known obstacles and gray region is unknown. In physical space, dark gray region is obstacles, but unknown to the robot. White region around the initial robot configuration (pointing vertically down and shown in gray) is known to be free at the start. The robot in the desired goal configuration is shown in black. SBIC-PRM distinguishes between three types of landmarks: *White* landmarks, similar to the nodes in [5] belong to $\mathcal{C}_{free}$, i.e., the robot in this configuration lies completely in $\mathcal{P}_{free}$; *Black* landmarks correspond to robot being in collision with a *(sensed or known)* obstacle at the corresponding configuration; and, *Gray* landmarks correspond to those configurations where the robot does not intersect with any sensed obstacles, and does not lie com-

[3]This termination condition would depend on whether it is a start-goal problem or exploration problem. This one is for start-goal problem.

pletely in $\mathcal{P}_{free}$ either. In Figure 3 the dots in gray (unknown) region[4] are the gray landmarks, the dots in the white (known) region are the white landmarks, and the dots in the black region (can't see them) are black landmarks. The nodes of the roadmap (a graph) are the white landmarks and two nodes are connected if a local planner — that simply tries to execute a discretized straight line in configuration space — returns a collision-free path between the nodes. Edges between nodes are not shown to avoid clutter. There are several connected components in the evolving roadmap (recall that the C-space is a torus, shown as a rectangle). The start node belongs to the one on lower left corner.

The robot carries out its motion within this evolving roadmap to further sense the physical environment. Each sensing action, as mentioned earlier, is chosen to maximize the expected entropy decrease (while satisfying reachability constraints), resulting in efficient and fast exploration of the environment. With each sensing action, as new free space (and obstacles) is sensed in the physical world, the roadmap is expanded by placing new landmarks and checking the status of all gray landmarks (old and new ones). The status of some may change to white (or black), thereby expanding the roadmap. This process is repeated until either the final goal is reachable from one of the nodes in the graph (as in this case), or the goal is declared unreachable.

The core computation in this process is that of collision detection with known obstacles/unknown region. The geometry of the robot and the sensor is modeled by a collection of polyhedra and the environment is modelled as spatial occupancy octrees. Each scan gives a range image which is used to update the octree(s) representing the environment. Details of these rather straightforward geometric computations are in [11].

For brevity, we omit a comparison with other similar systems [12, 13], except to note that our approach is more systematic, effective and efficient for both view planning and model-based planning components. See [11] for details. For example, the next view in [12] is chosen to maximize the unknown physical space volume. We show (see Section 3.5) that the MER criterion based view planning results in more efficient exploration. We also note there is vast view planning literature in machine vision [14], however, these works assume that the sensor (camera) can move arbitrarily and do not consider the geometric and kinematic constraints — critical for eye-in-hand systems — on where the sensor can be positioned.

2. Notation

Let $\mathcal{A}$ denote the robot. We assume that the sensor body is "absorbed" into the robot $\mathcal{A}$. Hence, we can treat the physical sensor as an abstract reference frame without any physical body. Let $\mathcal{P}$ denote the physical space (R^p, $p = 2$ or 3) and $\mathcal{C}$ denote the configuration space. $\mathcal{A}(q) \subset \mathcal{P}$ denotes the physical space occupied by the robot at configuration $q \in \mathcal{C}$. Let $\mathcal{V}_{\mathcal{S}}(q) \subset \mathcal{P}$ denote the sensed region when robot scans at configuration q. Subscripts $free$, obs and

[4] Note that the regions are not computed by the planner, only the corresponding landmarks are computed. Regions are shown for visualization only.

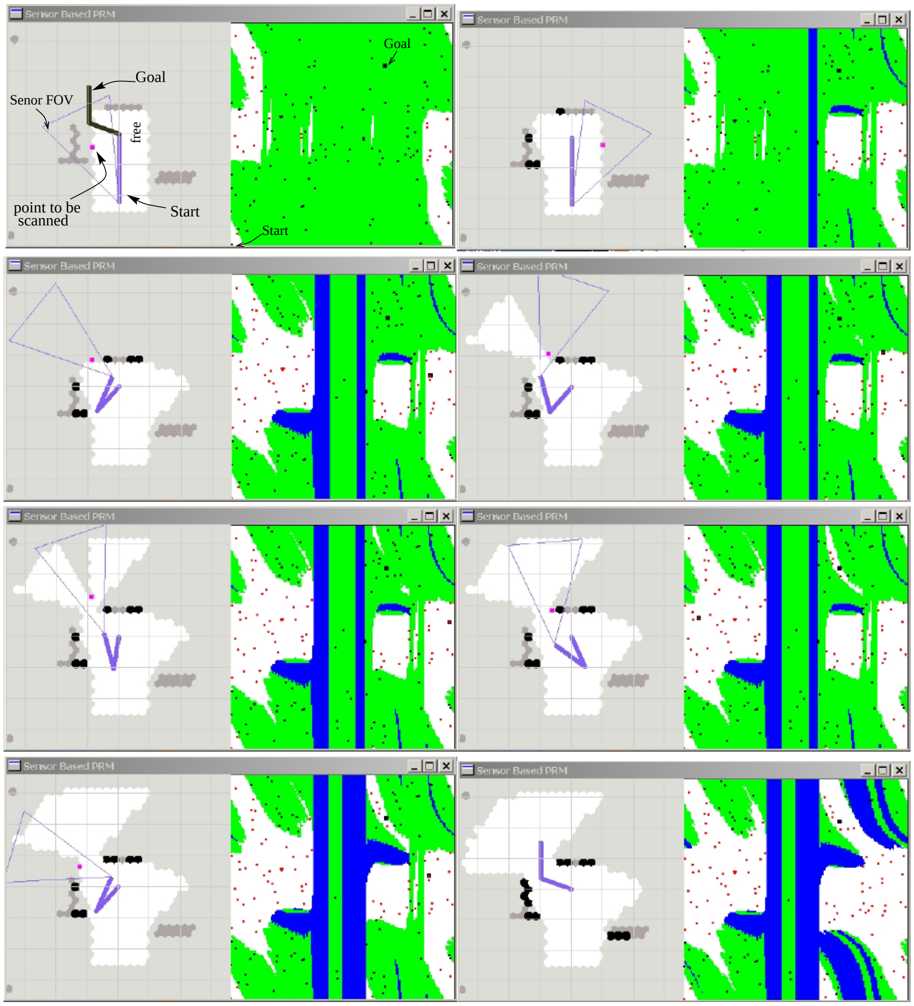


Figure 3. SBIC-PRM illustrated for a 2-link eye-in-hand system — a 2-link robot arm with an eye sensor (a range camera) mounted at the wrist. Sensor field of view is indicated by the triangular region. See text for further explanation.

unk are used to indicate the free, obstacle and unknown portions of physical or configuration space. For instance, $\mathcal{A}_{unk}(q)$ describes the part of the robot that lies in unknown physical space at configuration q. $\mathcal{C}_{free}$ is characterized by a *roadmap* $\mathcal{R}$. We use $\mathcal{R}(l_0)$ to represent the connected component of $\mathcal{R}$ that contains l_0, the landmark corresponding to the start configuration q_0. $\mathcal{C}_{free}(q_0)$ denotes the connected component of $\mathcal{C}_{free}$ containing q_0.

3. View Planning with Kinematic/Geometric Constraints

Recall that there are two sub-problems within the view planning problem: a) to determine where in the physical space should the sensor scan, and b) determine a robot configuration from where the scan should be taken. We first present our approach to solve a). In general, there are two main objectives in choosing which region to scan: (i) to gain information about the environment (exploratory component) and (ii) to preferentially "expand" toward the goal (goal component).

3.1. Where to Scan: C-space Entropy and MER Criterion

Recall that from a pure exploration perspective, we may view each sensing action as gaining information about the status of configuration space. The status of C-space is viewed as an n-dimensional discrete stochastic signal, in which each discretized configuration takes a value in a binary set {0(free), 1(obstacle)}. The entropy of this stochastic process, denoted by $H(\mathcal{C})$ and called C-space entropy, provides a measure of information about C-space. $H(\mathcal{C})$ can be written as $H(\mathcal{C}) \sum_{i=0}^{N-1} H(C(q_i)) - \sum_{i \neq j} I(C(q_i), C(q_j)) + \sum_{i,j,k \text{ different}} I(C(q_i), C(q_j), C(q_k)) - \cdots$, where $H(\mathcal{C}(q_i))$ is the entropy of the random variable $\mathcal{C}(q_i)$, the status of the configuration q_i, and $I(\cdot)$ is the mutual information of multiple random variables. The next sensing action is chosen to maximize the *expected* value of the change in entropy[5]. We call this criterion the *maximal entropy reduction (MER)*. Formally, the view planning problem is to choose a scan, such that $-E(\Delta H(\mathcal{C}))$ is maximized, i.e.

$$\max_{\substack{\text{all possible}\\ \text{scans}}} -E(\Delta H(\mathcal{C})) = \max_{\substack{\text{all possible}\\ \text{scans}}} \left(H(\mathcal{C}_{\substack{\text{before}\\ \text{scan}}}) - E(H(\mathcal{C}_{\substack{\text{after}\\ \text{scan}}}))\right)$$

3.2. Implementing MER criterion: Information Gain Density

$H(\mathcal{C})$ is defined as a function of C-space. But it is ultimately determined by the status of physical space, and indeed the the probability that a given region, $\mathcal{B} \subset \mathcal{P}$ is free, i.e., $p(\mathcal{B} \subset \mathcal{P}_{free})$. When a small neighborhood of a point $x \in \mathcal{P}$, denoted by $\mathcal{B}(x)$ is scanned, it would change the entropy. We could compute this (expected) change in entropy per unit volume around every point $x \in \mathcal{P}$, essentially a histogram of $G_c(x) = \lim_{vol(\mathcal{B}(x)) \to 0} \frac{-E(\Delta H)}{vol(\mathcal{B}(x))}$. The best point to be scanned is the one that maximizes $G_c(x)$, the *information gain density function* (IGDF). It[6] can be expressed as $\sum_q g_q(x)$ where

$$g_q(x) = \lim_{vol(\mathcal{B}(x)) \to 0} \frac{-E\Delta H(C(q))}{vol(\mathcal{B}(x))}$$

An interpretation of the above expression is as follows. When $\mathcal{B}(x) \subset \mathcal{P}$ is scanned, it affects the C-space entropy via each configuration q. $g_q(x)$ is the expected contribution of q to the resulting C-space entropy reduction (per unit volume). $G_c(x)$ is the summation of $g_q(x)$ over all the configurations in the

[5]For simplicity, we discuss the case of a single scanning action. This is easily generalized to a sequence of sensing actions.

[6]This limit may not always exist. In practice, one could always compute an approximation for a small region $\mathcal{B}(x)$. Also, two simplifying assumptions are made here. First, mutual entropy terms are neglected. Second, we assume that the status (free or obstacle) of $\mathcal{B}(x)$ can be determined after the scan, i.e. no obstacle occludes the view.

entire C-space. However, note that only certain configurations contribute to $G_c(x)$. $g_q(x)$ equals 0 when $\mathcal{B}(x) \cap \mathcal{A}(q) = \emptyset$, because changing the status of $\mathcal{B}(x)$ will not change the contribution to C-space entropy from such a configuration q. The set of configurations that do contribute belong to what we call the C-zone of x, denoted by $\mathcal{X}(x)$. Formally, $\mathcal{X}(x) = \{q : x \in \mathcal{A}(q)\}$. Points outside the C-zone of x do not contribute to $g_q(x)$. Therefore, we explicitly write $G_c(x) = \sum_{q \in \mathcal{X}(x)} g_q(x)$

We have derived an expression for $G_c(x)$ for Poisson distribution of (point) obstacles. Recall that a stationary Poisson point process Φ is characterized by uniformly distributed points in space [15]. From a motion planning point of view, these points are obstacles in the physical space of the robot. The density parameter, λ, gives us the ability to set the density of obstacles in the physical space. For a more dense distribution, λ is higher. We simply state the result here, see [11] for details.

$$
\begin{aligned}
p(\mathcal{B} \subset \mathcal{P}_{free}) &= e^{-\lambda\, vol(\mathcal{B})} \\
p(q) &= p(\mathcal{A}_{unk}(q) \subset \mathcal{P}_{free}) = e^{-\lambda\, vol(\mathcal{A}_{unk}(q))} \\
G_c(x) &= \sum_{q \in \mathcal{X}(x)} \lambda\Big(\log \frac{p(q)}{1-p(q)} - p(q)\log p(q) - (1-p(q))\log(1-p(q))\Big)
\end{aligned}
$$

These expressions completely determine the IGDF for a physical space with a Poisson point distribution.

3.2.1. IGDF: Practical Implementation

The calculation of IGDF can be approximated as follows. Eq. 1 tells us that computing $IGDF(x)$ requires a summation over C-zone $\mathcal{X}(x)$. The computational cost of finding the C-zone $\mathcal{X}(x)$ is high for a high dimensional C-space. One way out would be to approximate the summation over randomly selected configurations $q \in \mathcal{C}_{unk}$ (unknown part of C-space) when $x \in \mathcal{A}(q)$.

3.3. Overall Objective Function

The exploratory component is achieved by maximizing the IGDF, as we just discussed above. For the goal-directed component, a reasonable choice is to find the gray landmark closest to the goal (call it l_G) subject to the condition that $\mathcal{A}(q_{l_G})$ has at least a certain fraction of its volume in $\mathcal{P}_{free}$. $\mathcal{A}_{unk}(q_{l_G})$ is the region that should be sensed. Formally,

$$G_g(x) = \delta(\mathcal{A}_{unk}(q_{l_G})) = \begin{cases} 1, & x \in \mathcal{A}_{unk}(q_{l_G}) \\ 0, & otherwise \end{cases} \tag{1}$$

The combined objective function is then a weighted sum of G_c and G_g, i.e. $G(x) = w_e G_c(x) + w_g G_g(x)$. $G(x)$ can be considered as a generalized IGDF with a larger weight placed on the points (in physical space) belonging to configuration q_{l_G}. The point x to be scanned is the one that maximizes $G(x)$, i.e. $x_{max} = \{x : \max_{x \in \mathcal{P}_{unk}} \{G(x)\}\}$. In Figure 3, for each snapshot, the small dot in the centre of the sensor FOV (triangle) corresponds to x_{max}. w_e and w_g, the weights for exploratory and goal components, can be chosen to emphasize the exploratory or goal-directed behaviour as needed by the task at hand.

3.4. Determine View Configuration To Scan From:

Given a point x_{max} to be scanned, the next step is to determine a reachable configuration from which to scan it, i.e. to determine a view configuration, say q_v. Obviously, $q_v \in \mathcal{C}_{free}(q_0)$, since the robot needs to reach q_v in order to scan. We search $\mathcal{R}(l_0)$, the connected component of the roadmap to find the view node[7]. From all the white landmarks which satisfy the collision (between robot and known obstacles) and visibility (of x_{max} from the sensor) constraints, we choose the one whose view node configuration results in the center of the sensing region $\mathcal{V}_{\mathcal{S}}(q_v)$ being closest to x. This is the the view node. Note that a q_v may not exist for a given x_{max}. In this case, the planner chooses the next best x_{max} to scan. If the view planner has exhausted all the observable physical space, it will stop and report a failure. In Figure 3, the robot is at the chosen view node in each snapshot.

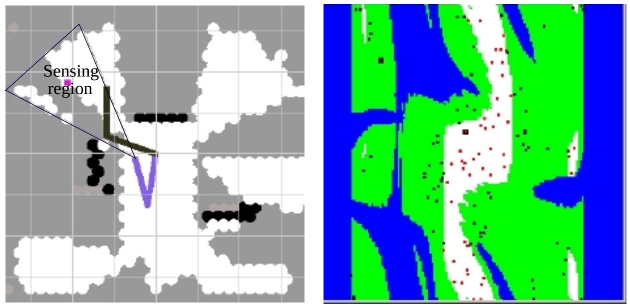
3.5. Effectiveness of MER criterion

We conducted a series of experiments on the simulated two link eye-in-hand system of Figure 3 with three different criteria for view planning: (i) the next view (position and direction) is randomly chosen, (ii) the next view is chosen so that it maximizes the unknown physical space volume, and (iii) the next view is chosen based on MER criterion. The task was to reach the goal configuration (shown in dark); the start configuration being vertically downwards in each of the three cases. Fig. 4 shows the explored physical space and C-space after 14 scans with each of these three different sensing strategies. The MER criterion results in the highest explored C-space (70%), and the maximal unknown physical space volume criterion results in the least explored C-space (33%), with the random scan strategy somewhere in between (47%). In fact the robot was able to reach the goal configuration only with the MER based view planning; it failed to make the goal configuration free with the other two strategies. It is interesting to note that even random strategy does better than the one based on maximizing the unknown physical space volume. One factor is that the overlap between viewing regions tends to be small in the latter strategy, and hence known physical space may tend to consist of several small islands scattered around. On the other hand, only large contiguous regions may make portions of C-space known. In addition, much of the physical space may even lie outside the workspace of the robot, hence contributing nothing to the C-space exploration. Maximizing unknown physical space volume may, therefore be the least efficient view planning strategy from C-space exploration perspective.

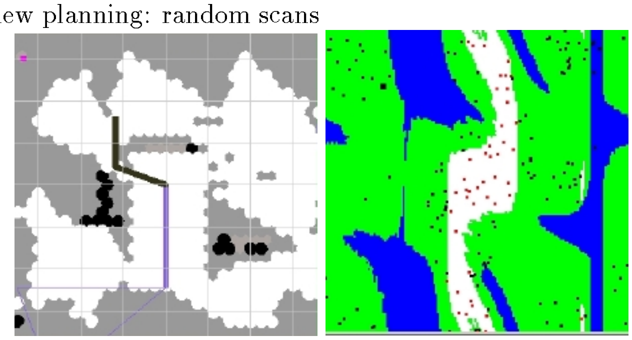
4. Experiments

We now present experimental results with SBIC-PRM running on the real test-bed shown in Figure 1 – a PUMA 560 with a triangulation-based area-scan laser

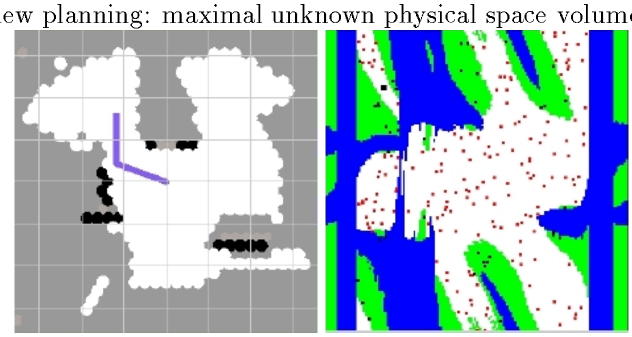
[7] If the sensor has internal degrees of freedom, one can further search over this sub-space to find the "best" viewing position.



view planning: random scans



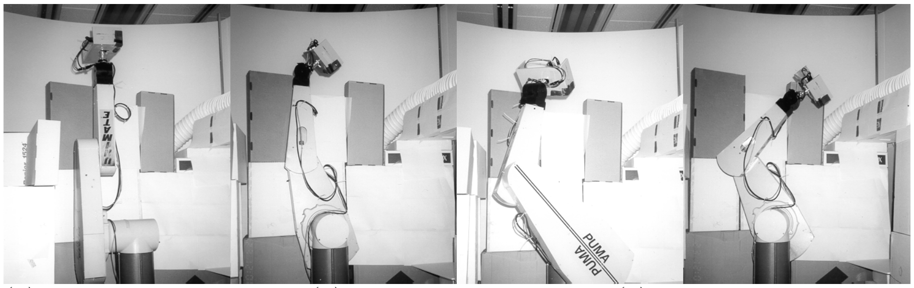
view planning: maximal unknown physical space volume



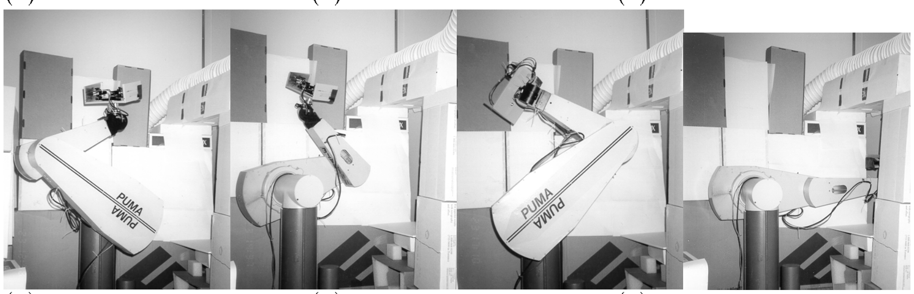
view planning: MER criterion

Figure 4. Comparison of C-space exploration efficiency for three different view planning criteria. White region shows the explored C-space. MER criterion results in significantly more C-space explored.

range finder (the eye) mounted on its wrist. The sensor field of view is a conical region, approximately one metre long with an apex (solid) angle of about 40 degrees. The sensor accuracy is about 1 cm in all three directions. For system and software details, see [7]. Figure 5 shows a real planning task solved by the planner. In this experiment, the robot has its forearm and upper arm inside a hole in a wall at the goal configuration (snapshot goal). The start configuration is robot pointing vertically upwards (snapshot 1). A tight small cuboid region around the initial configuration is assumed to be free at the start. Several boxes were scattered in the robot workspace that are unknown to the robot in the beginning. The robot reached the goal configuration in about 13 scans. Figure shows some of the intermediate viewing configurations that the robot moved to on its way to the goal configuration.



(1) (2) (3) (4)



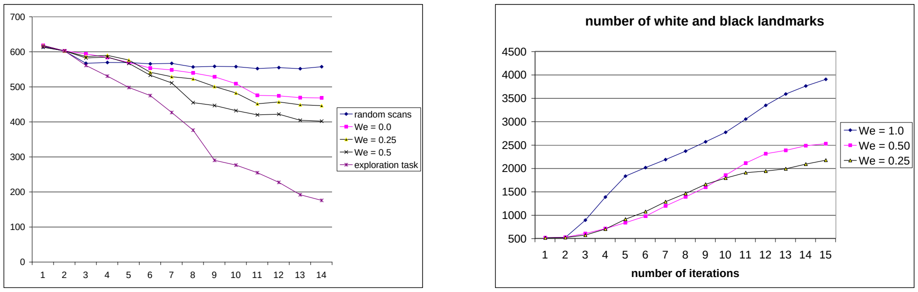
(5) (6) (7) *goal*

Figure 5. An example of a planning task solved by the planner on the real eye-in-hand testbed.

In the following experiments, SBIC-PRM is run 10 times for each case and the results reported are average results. This was done to reduce statistical variations caused by randomized nature of the algorithm.

SBIC-PRM was run with different for different weightings of exploratory and goal components (w_e and w_g). $w_g = 0$ corresponds to a pure exploration task. As a baseline, we also implemented a random view planning strategy in which scans were taken in a random direction from a randomly chosen white landmark. For each run, the C-space entropy is computed[8]. Figure 6(a) shows the resulting plots. The horizontal axis is the number of scans (iterations). The vertical axis is the C-space entropy. The random scan curve is shown with diamond markers. Clearly, larger w_e (exploration weight) results in faster C-space entropy reduction, or equivalently faster exploration. Furthermore, MER based view planning explores the environment significantly faster than random scans. Because of the random approach in generating landmarks, the number of landmarks of different colors (white or black) should be roughly proportional to the volume of C-space of the corresponding status (free or obstacle). Therefore, the number of black and white landmarks essentially provides a measure of the volume of explored (or known) C-space. Figure 6(b) shows the increase in black and white landmarks vs. the number of iterations for different values of exploration weight, w_e. The number of black and white landmarks increases at a faster rate for greater values of w_e. The faster speed of C-space exploration is, on average, at a cost of slightly increasing the number of scans to reach the goal. For instance, with $w_e = .25$, about 14 scans were needed to reach the

[8]For computational reasons, entropy computation was carried out over a set of randomly selected (unknown) C-space configurations.



(a) (b)

Figure 6. Higher exploration weight (w_e) results in faster decrease in entropy (a)and faster increase in number of black+white landmarks (b), thereby indicating faster rate of exploration.

goal; with $w_e = 1$, about 18 scans were needed. In some situations, however, a larger w_e may lead to fewer scans for reaching a goal. For instance, a larger w_e may help the planner to escape dead ends which were initially believed to be a free path towards the goal.

Typically, the planner is able to reach (planning as it senses) the goal configuration in about 7 - 25 scans (depending on the scene complexity), while avoiding collisions with the obstacles throughout. The run time (on a Pentium II 450) for one iteration varies from 1 minute to 2 minutes depending on the number of landmarks generated and the complexity of the scene. More details on run times are reported in [7].

5. Conclusions

We demonstrated that (i) our information theoretic approach to view planning, in particular C-space entropy and the resulting MER criterion provide an efficient way of exploring C-space; (ii) parameters w_g and w_e change the behavior of the algorithm, i.e., the C-space exploration rate increases with the weight of the exploratory component w_e, and (iii) SBIC-PRM is an effective eye-sensor based planner making a robot arm move around safely (without collisions) while carrying out exploratory and purposive tasks in unknown environments.

We have made certain simplifying assumptions in deriving the expression for MER criterion. In particular, (i) we assume Poisson point process, (ii) we ignore occlusion (visibility) constraint, i.e., we assume that a point to be sensed will either become free or obstacle, and (iii) we compute MER for a single point. Our immediate future work is to extend the MER formulation beyond these assumptions. Along another line, several fundamental yet novel issues arise for sensor-based motion planning and exploration for robots with non-trivial geometry/kinematics. We intend to explore them in future. See [3] for some initial thoughts.

Acknowledgements

This research has been jointly funded by the National Science and Engineering Research Council, Canada and National Research Council, Canada. Many thanks to Mehran Mehrandezh for the 2-dimensional implementation of the planar simulated eye-in-hand system. Thanks to Juan Manuel Ahuactzin for

providing the initial simulation code.

References

[1] V.J. Lumelsky and E Cheung. Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, 23(1):194–203, Jan.-Feb 1993.

[2] K. Hirai, M. Hirose, M. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1321–1326, 1998.

[3] Kamal Gupta and Yong Yu. On eye-sensor based path planning for robots with non-trivial geometry/kinematics. accepted for *IEEE International Conference on Robotics and Automation*, 2001.

[4] Kamal K. Gupta and Angel del Pobil, editors. *Practical Motion Planning in Robotics: Current Approaches and Future Directions.* John Wiley, 1998.

[5] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):556 – 580, Aug 1996.

[6] J. Ahuactzin and A. Portilla. A basic algorithm and data structure for sensor-based path planning in unknown environment. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2000.

[7] Y. Yu and K. Gupta. Sensor-based roadmaps probabilistic roadmaps: Experiments with an eye-in-hand system. *Advanced Robotics*, 14(8), August 2000. A version also appeared In *Proceedings of IEEE/RSJ International Conference on Intelligent Robot and System*, pages 1707–1714, 1999.

[8] Y. Yu and K. Gupta. Sensor-based motion planning for manipulator arms: An eye-in-hand system. In *IEEE International Conference on Robotics and Automation video session*, 2000.

[9] Y. Yu and K. Gupta. An information theoretic approach to view point planning for motion planning of eye-in-hand systems. In *Proceedings of 31st International Symposium on Robotics*, pages 306– 311, 2000.

[10] Yong Yu and Kamal Gupta. An information theoretical approach to view planning with kinematic and geometric constraints. accepted for *IEEE International Conference on Robotics and Automation*, 2001.

[11] Yong Yu. An information theoretical incremental approach to sensor-based motion planning for eye-in-hand systems. Ph.D. Thesis. School of Engineering Science, Simon Fraser University. Canada. 2000.

[12] E. Kruse, R. Gutsche, and F. Wahl. Effective, iterative, sensor based 3-d map building using rating functions in configuration space. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1067 – 1072, 1996.

[13] P. Renton, M. Greenspan, H. Elmaraghy, and H. Zghal. Plan-n-scan: A robotic system for collision free autonomous exploration and workspace mapping. *Journal of Intelligent and Robotic System*, 24:207 – 234, 1999.

[14] S. Hutchinson and A. Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Transaction on Robotics and Automation*, 5(6):765 – 783, December 1989.

[15] D. Stoyan and W.S. Kendall. *Stochastic geometry and its applications.* J. Wiley, 1995.