View Planning for Exploration via Maximal C-space Entropy Reduction for Robot Mounted Range Sensors

Pengpeng Wang and Kamal Gupta

Robotic Algorithms and Motion Planning (RAMP) Lab
School of Engineering Science
Simon Fraser University
Burnaby, BC, CANADA, V5A 1S6
{*pwangf, kamal*}*@cs.sfu.ca*

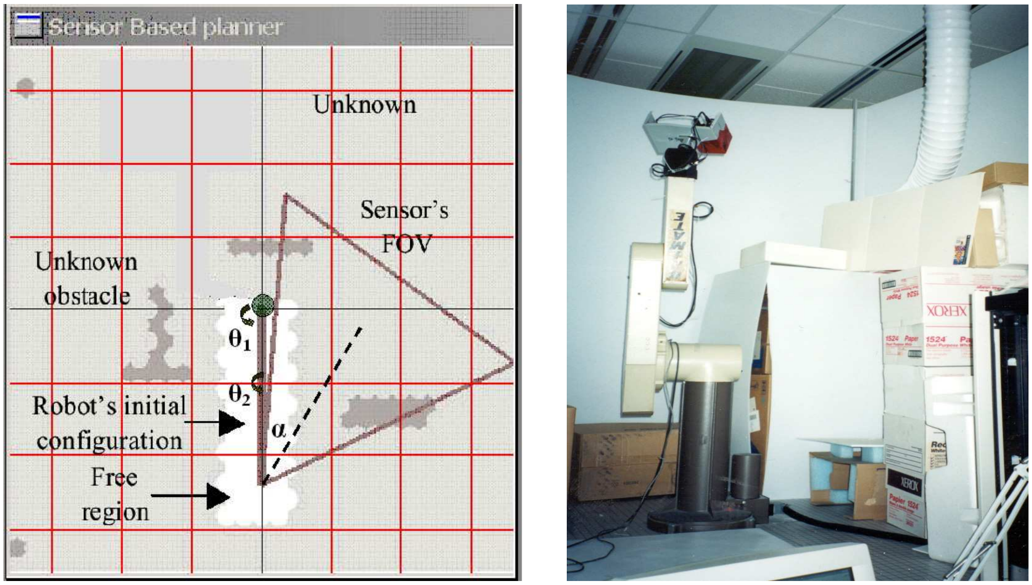
Abstract

We introduced the concept of C-space entropy recently in [9] as a measure of knowledge of C-space for sensor-based exploration and path planning for general robot-sensor systems. The robot plans the next sensing action to maximally reduce the expected C-space entropy, also called the Maximal expected Entropy Reduction, or MER criterion. The resulting view planning algorithms showed significant improvement of exploration rate over physical space based criteria. However, this expected C-space entropy computation made two idealized assumptions. The first was that the sensor field of view (FOV) is a point; and the second was that there are no *occlusion (or visibility) constraints*, i.e., as if the sensor can sense through the obstacles. We extend the expected C-space entropy formulation where these two assumptions are relaxed, and consider a range sensor with non-zero volume FOV and occlusion constraints, thereby modeling a realistic range sensor. Planar simulations and experimental results on the SFU Eye-in-Hand system show that the new formulation results in further improvement in C-space exploration efficiency over the point FOV sensor based MER formulation.

keywords: sensor-based robot path planning, robot mounted range sensor, view planning, configuration space, configure space entropy

1 Introduction

In sensor-based path planning, a robot, equipped with one or more sensors, senses and explores its environment as it plans its motions while avoiding collisions with the obstacles in its environment [1]. Most research in sensor-based path planning and exploration has concerned itself with mobile robots [2, 3, 4, 5, 6, 7], which are often modeled as points in the physical space, hence considered to be trivial geometry and kinematics. Our recent work has concentrated on general robot-sensor systems, where a range sensor is mounted on a robot with non-trivial geometry and kinematics [8, 9]. See also [10, 11, 12]. The range sensor (also called an "eye" type sensor) provides distances from a given vantage point (actual implementation may be a laser range scanner, passive stereo vision, etc.). Fig. 1 shows two examples of Eye-in-Hand systems [8, 9]. (i) shows a simulated planar two link robot equipped with a triangle field of view (FOV) range sensor on its end-effector. The white, light grey, and dark grey regions in the figure are the free region known to the robot, free but still unknown region, and unknown obstacles respectively. The robot starts from its initial configuration, the vertical line in the middle of the figure, and its task is to explore its environment while avoiding collisions with the obstacles, known or unknown. (ii) shows the SFU Eye-in-Hand system, a six degree of freedom (dof) PUMA 560 robot arm and a wrist mounted area scan laser range finder. The robot is shown at its start configuration, with unknown obstacles around it; its task is to explore the environment while avoiding collisions with obstacles. Sensor-based planning and exploration for such general robot-sensor systems is of interest from several perspectives. The first and foremost is that the problem is of intrinsic interest and leads to several novel fundamental issues [13]. It is also of practical interest since it results in a high degree of manoeuvrability for the sensor, necessary in constrained situations such as exploring under a table.



(i) 2-D simulation

(ii) SFU Eye-in-Hand system

Figure 1: Two Eye-in-Hand systems: (i) 2D simulation: a planar 2-link robot with a triangle FOV range sensor (ii) SFU Eye-in-Hand system: 6 dof PUMA 560 with a wrist mounted area scan laser range finder.

Unlike simple mobile robots, for robots with non-trivial geometry and kinematics, such as the Eye-in-Hand systems, where the robot can move (path planning) and what it should sense (view planning), has a much more complex relationship [8]: the robot must find additional physical space to manoeuvre itself, taking into account its own "size" and "shape" and not simply any additional physical space. What this implies is that the effect of the sensing action, which obviously senses physical space, must be (implicitly or explicitly) transformed to and viewed from the configuration space (C-space) of the robot.

In [9], we showed that, for efficient C-space exploration, the next view should be planned to give maximum information (whether a configuration is free or in collision with an obstacle) of the robot C-space. Treating the unknown environment stochastically, we introduced the notion of C-space entropy as a measure of the robot's (lack of) knowledge of C-space. The next best view is the one that maximizes the expected entropy reduction, called Maximal expected Entropy Reduction (MER) criterion. In [9], under a Poisson point process model of (obstacle distribution in) the environment, we derived a closed-form expression of MER criterion. We also showed that point FOV sensor based MER criterion is far more efficient in C-space exploration than physical space based criteria, such as to randomly choose the next view or to choose the next view with the maximum unknown physical space volume (MPV) in the senor FOV [11].

However, the above result is based on two idealized assumptions: (i) the sensor has a point FOV, i.e., it senses a single point and (ii) no visibility constraints were taken into account, i.e., as if the sensor would "see" (get range measurement) through the obstacles. In this paper[1], we relax the above two assumptions and derive closed form expressions for the MER criterion computation for generic range sensors with non-zero volume FOV while respecting occlusion constraints. These closed form expressions are a key contribution of the paper. Simulations and experiments show clear improvements of exploration efficiency with the new formulation over point FOV sensor based MER. This also implies that the generic sensor FOV based MER outperforms the physical space criteria mentioned above. The improvements come with a computation overhead for each iteration, the cumulative effect of which mitigates due to the reduced number of iterations. See Section 5 for a detailed discussion. Since real and commercially available sensors fall in the generic FOV sensor category, the results in this paper make the MER criterion applicable to real systems.

A detailed presentation of other related works and comparisons with our general approach is given in [9]. Here we simply mention Choset *et al.* [16] for sensor-based path planning for a planar rod robot; Kruse *et al.* [11] and Renton *et al.* [12] for previous work on Eye-in-Hand systems; survey by Scott *et* al. [17] on a variety of view planning problems (and correspondingly a variety of criteria) in computer vision area; and Banos and Latombe [2] for work on NBV problem for a point robot in the presence of obstacles and visibility constraints. In addition, information theoretic concepts such as Shannon's entropy are used in learning tasks [18] and in mobile robot exploration [19, 20, 21, 22]. As opposed to the focus in this paper, these works do not incorporate any robot C-space information.

[1] As an intermediate result, and also applicable in its own right (certain range sensors sense along a beam), we derived closed form expressions for MER for beam FOV sensors, sensors sensing along a beam of finite length, [14, 15].

2 Background: C-space Entropy

2.1 Notation and Problem Statement

Let $\mathcal{P}$ denote the 2D or 3D physical space, R^2 or R^3. Let $\mathcal{A}$ denote the robot, and $\mathcal{C}$ denote its configuration space (C-space). $q \in \mathcal{C}$ denotes a robot configuration, and $\mathcal{A}(q) \in \mathcal{P}$ denotes the physical region occupied by the robot at q. Subscripts $free$ and *obs* denote the free regions and obstacles in both physical and C-space. $\mathcal{P} = \mathcal{P}_{free} \cup \mathcal{P}_{obs}$, where $\mathcal{P}_{free}$ denotes the free physical space (known and unknown), and $\mathcal{P}_{obs}$ denotes the set of obstacles (known and unknown). $\mathcal{C}_{free}$ and $\mathcal{C}_{obs}$ are similarly defined. Subscripts u and *known* denote the unknown and known quantities (in physical and C-space) and superscript i denotes the iteration number, i.e., the number of scans (or views) that the sensor has already taken ($i = 0$ at the very start). For notational brevity, rather than using subscripts *known-free* and *known-obs*, we simply use *free* and *obs*. So $\mathcal{P}^i_{free}, \mathcal{P}^i_{obs}$, and $\mathcal{P}^i_u$ denote the known free physical space, known obstacles, and unknown physical space after iteration i. Hence $\mathcal{P}^i_{known} = \mathcal{P}^i_{free} \cup \mathcal{P}^i_{obs}$, and $\mathcal{P} = \mathcal{P}^i_{known} \cup \mathcal{P}^i_u$. Furthermore, $\mathcal{A}^i_u(q)$ denotes the unknown part of the robot at iteration i, i.e., $\mathcal{A}^i_u(q) = \mathcal{A}(q) \cap \mathcal{P}^i_u$. Similarly $\mathcal{A}^i_{known}(q) = \mathcal{A}(q) \cap \mathcal{P}^i_{known}$.

$\mathcal{C}^i_{free}$, $\mathcal{C}^i_{obs}$, and $\mathcal{C}^i_u$, respectively denote the known free C-space, known C-obstacles, and unknown C-space, after iteration i. A configuration q is: free ($\in \mathcal{C}^i_{free}$), if $\mathcal{A}^i_u(q) = \emptyset \ \wedge \ \mathcal{A}(q) \cap \mathcal{P}^i_{obs} = \emptyset$; obstacle ($\in \mathcal{C}^i_{obs}$), if $\mathcal{A}(q) \cap \mathcal{P}^i_{obs} \neq \emptyset$; and unknown ($\in \mathcal{C}^i_u$), if $\mathcal{A}^i_u(q) \neq \emptyset \ \wedge \ \mathcal{A}(q) \cap \mathcal{P}^i_{obs} = \emptyset$.

Let $\mathcal{B} \subset \mathcal{P}^i_u$. The set of unknown robot configurations at which the robot overlaps $\mathcal{B}$, is called the unknown C-zone of $\mathcal{B}$, denoted by $\chi^i_u(\mathcal{B})$, i.e., $\chi^i_u(\mathcal{B}) = \{q \in \mathcal{C}^i_u : \mathcal{A}^i_u(q) \cap \mathcal{B} \neq \emptyset\}$. The definition also applies to the special case, when $\mathcal{B}$ is a point x, i.e., $\chi^i_u(x) = \{q \in \mathcal{C}^i_u : x \in \mathcal{A}^i_u(q)\}$ [9]. The key notion of the unknown C-zone of $\mathcal{B}$ essentially captures the part of C-space that will be affected if $\mathcal{B}$ were to be sensed by the sensor.

Let $\mathcal{S}$ denote a range sensor attached to the robot. Let s denote the sensor's configuration, the vector of parameters that completely determine the sensor's coordinate frame we attach to its origin. For the eye-in-hand systems, $s = (x, y, \theta)$ for planar case; $s = (x, y, z, \alpha, \beta, \gamma)$ for 3D case. $\mathcal{V}(s) \in \mathcal{P}$ denotes the sensor's field of view (FOV) at configuration s. We discuss two different sensor FOV's: (i) an idealized point, i.e., $\mathcal{V}(s)$ is a point, and (ii) a generic non-zero volume FOV sensor, such as an area scan range sensor, i.e., $\mathcal{V}(s)$ is an open set. Finally, we use "\" to denote the set difference operator.

2.2 C-space Entropy and MER Criterion

Following subsections are a brief recap of the results in [9].

Assume a stochastic model of obstacle distribution in the unknown physical space $\mathcal{P}^i_u$, such as the Poisson Point Process described later. As the function $\mathcal{A}(q)$ relates C-space to physical space, this stochastic model induces a probability distribution over C-space. The entropy of the resulting distribution, called the C-space entropy, $H(C|\mathcal{P}^i_{known})$, measures the robot's (lack of) knowledge of its own C-space. The conditional $\mathcal{P}^i_{known}$ denotes "given the knowledge of the physical space after iteration

i", the prior before sensing.

Assume C-space is represented as a set of N configurations, q_1, ..., q_N. This could arise either by discretizing the C-space at a certain resolution, or by randomly sampling it (for high dimensional C-spaces). Q, a binary random variable (r.v.) that takes a value of 0(free) or 1(obstacle), denotes the collision status of configuration q. The marginal entropy of Q, $H(Q)$, is given by [23],

$$H(Q) = \Pr(Q)\log\Pr(Q) - (1-\Pr(Q))\log(1-\Pr(Q)) \tag{1}$$

C-space entropy is then given by,

$$H(\mathcal{C}|\mathcal{P}^i_{known}) = -\sum_{Q_1=0,1} .. \sum_{Q_n=0,1} \left(\Pr[Q_1,\ldots,Q_n|\mathcal{P}^i_{known}]\cdot\log\Pr[Q_1,\ldots,Q_n|\mathcal{P}^i_{known}]\right) \tag{2}$$

The next best view, s^{i+1}_{max}, is chosen to maximize the expected C-space entropy reduction (denoted by $ER_{\mathcal{C}}$), where the expectation is computed over all possible sensing results for sensing action s. (Note that the number of possible sensing results can be exponential.) This is called MER (Maximal expected Entropy Reduction) criterion. Formally, the criterion is defined as,

$$\begin{aligned} s^{i+1}_{max} &= \arg\max_s ER_c(s) \overset{def}{=} \arg\max_s \; -E\{\underset{s}{\triangle}H(\mathcal{C}|\mathcal{P}^i_{known})\} \\ &\overset{def}{=} \arg\max_s(E\{H(\mathcal{C}|\mathcal{P}^i_{known}) - H(\mathcal{C}|\mathcal{P}^i_{known},\mathcal{V}(s))\}) \end{aligned} \tag{3}$$

$\underset{s}{\Delta}$ denotes the difference operation in a quantity (in this case, entropy) before and after sensing at s. The conditional $\mathcal{P}^i_{known}, \mathcal{V}(s)$ denotes one possible sensing result, given the knowledge of $\mathcal{P}^i_{known}$; and E denotes the expected value carried over all possible sensing results. For notational simplicity, we will omit the conditional $\mathcal{P}^i_{known}$ with the implicit assumption that all probabilistic quantities, unless otherwise mentioned, are conditional on $\mathcal{P}^i_{known}$.

2.3 Stochastic Model of Physical Space: Poisson Point Process

We use a simple probabilistic model of obstacle distribution in the physical space — the Poisson point process, which is essentially characterized by uniformly independently distributed points in space [24]. From a motion planning point of view, these points, denoted by pt, are obstacles. Given the density parameter λ for Poisson point process, the probability of an *unknown* region $\mathcal{B}_u \subset \mathcal{P}^i_u$ being free of obstacles, denoted by $p(\mathcal{B}_u|\mathcal{P}^i_{known})$, called the void probability, is the same as the unconditional void probability of $\mathcal{B}_u$, $p(\mathcal{B}_u)$, and is given by, (Here "| |" denotes the volume of a bounded set.)

$$p(\mathcal{B}_u|\mathcal{P}^i_{known}) = p(\mathcal{B}_u) = e^{-\lambda\,|\mathcal{B}_u|} \tag{4}$$

The void probability of an unknown robot configuration q is,

$$p(q|\mathcal{P}^i_{known}) = \begin{cases} e^{-\lambda\,|\mathcal{A}^i_u(q)|} & q \in \mathcal{C}^i_u \\ 0 & q \in \mathcal{C}^i_{obs} \\ 1 & q \in \mathcal{C}^i_{free} \end{cases} \tag{5}$$

Eq. (5) maps the probability distribution of physical space to C-space and induces a stochastic process onto it.

Although the Poisson point process model is a simplifying assumption because obstacles, in general, are not points in the real environments, nevertheless, it matches well our reasonable intuition that the more the robot (at a given configuration) is in unknown region, the less is the chance that it would be collision free. It is an unbiased uniform distribution assuming no obstacle shape information, and hence reasonable when no *a priori* information is known about the environment. As we show in this paper, Poisson point model allows one to derive efficiently computed closed form expression for expected C-space entropy reduction (for both point and generic FOV sensor models). The resulting algorithms, as shown via simulations and real experiments in Section 5, drastically improve the efficiency of C-space exploration when compared to pure physical space based criteria, such as MPV. Furthermore, the closed form expressions give us insights into the interplay between geometry (C-space changed as the result of a sensing action) and stochastic (entropy) component. This is beyond the scope of the current paper and is further explored in [25].

Other more complex models do not lend themselves to such closed form expressions and therefore would tremendously increase computational cost of entropy computations. For example, if we were to use occupancy grid maps [26] for the physical space, the expectation computation in Eq. (3) is to be carried out over all possible combinations of the grid statuses, thus potentially having an exponential (in the number of unknown cells in the sensor FOV) computational complexity. Existing exploration approaches (for exploring physical space) for mobile robots ignore this complexity and do not compute the true expectation. For example, some assume that the unknown area to be sensed is completely free, i.e., only one sensing result is possible [5, 7]. This is clearly an oversimplifying assumption. Others use an ad-hocly defined information function [22].

2.4 MER for Point FOV Sensor Model

For a point FOV sensor model, Fig. 2, which only senses a point (or an infinitesimal ball centered at a point) in physical space, rather than computing the expected entropy reduction, it is more appropriate to compute the corresponding density function,[2] i.e., expected entropy reduction per unit volume. Furthermore, ignoring mutual entropy terms for computation efficiency, the entropy reduction density function ($\widetilde{ERD}_{\mathcal{C}}(x)$) for a point FOV sensor is given by

$$\widetilde{ERD}_c(x) = \sum_{q \in \mathcal{X}_u^i(x)} -\lambda \log(1 - e^{-\lambda |\mathcal{A}_u^i(q)|}) \tag{6}$$

We interpret the above equation as follows. If a point $x \in \mathcal{P}_u$ were to be sensed, it affects the C-space entropy via each *unknown* configuration q that it "touches". The term inside the summation is the marginal contribution via configuration q. It depends on the volume of the unknown portion of the robot at q; the larger this volume, the higher is the marginal entropy reduction.

[2] While the absolute entropy reduction due to a single point becoming free is zero almost everywhere, the corresponding density function is non-zero and finite. Thus, it gives a more complete picture of the entropy reduction.

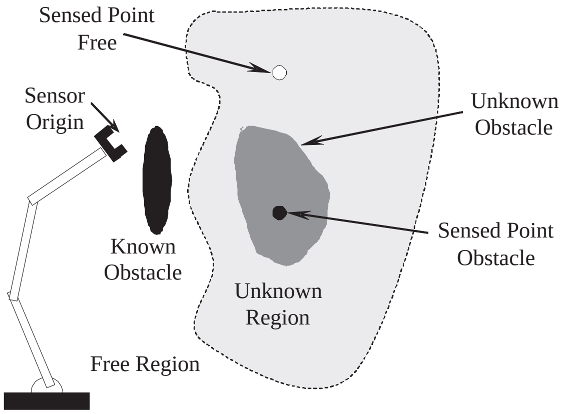


Figure 2: The point FOV sensor model. It senses if a given point x (or a small region around it) is obstacle or free. No occlusion constraints are modeled: a point can be sensed even if it is occluded by an obstacle. Two different potential sensing actions are shown. x_1 would be sensed free and x_2 would be sensed obstacle (even if it is occluded by known obstacle from the sensor).

Two key limitations in the above derivation are: (i) no occlusion constraints are taken into account, and (ii) the sensor FOV has zero volume. In the following, we relax these assumptions and derive an analytic expression for a generic nonzero volume FOV sensor, our main theoretical result.

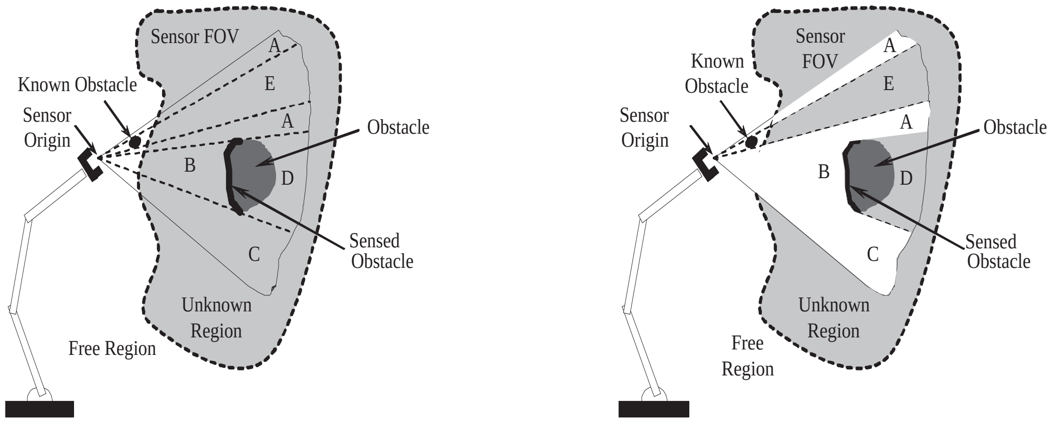
3 MER for Generic FOV Sensor Model

We now consider the general case, a range sensor with non-zero volume FOV and the volume sensed is governed by occlusion constraints. Most commercially available range sensors that provide range images (such as the area scan laser range sensor used in SFU Eye-in-Hand system) fall into this category. Fig. 3 shows a schematic diagram as the sensor senses an unknown region within its FOV. Let $\mathcal{V}_u(s)$ denote the unknown portion of the FOV that is not occluded by known obstacles. In the figure, $\mathcal{V}_u(s)$ consists of regions A, B, C and D (region E is excluded from $\mathcal{V}_u(s)$ since it is occluded by a known obstacle). After sensing, regions A, B and C become free; region D remains unknown because it is occluded by the sensed obstacles (shown in dark). Of course, the sensor also provides the distances from the sensor's origin to the sensed obstacles.

Since the volume of the sensor is not zero, clearly it is the entropy reduction (ER) function (as opposed to the density in the point sensor case) that is relevant. Making similar approximations as earlier, i.e., to approximate C-space entropy as sum of marginal entropies of unknown configurations, we have,

$$\widetilde{ER}_{\mathcal{C}}(s) = -\underset{s}{E}\{\triangle \widetilde{H}(\mathcal{C})\} = \sum_{q \in \chi_u(\mathcal{V}_u(s))} er_q(s)$$

As in the point sensor case, the summation is carried over the unknown C-zone of $\mathcal{V}_u(s)$, the set of unknown robot configurations at which some unknown part of the robot could be sensed. $er_q(s)$, the



Before scan. After scan.

Figure 3: A generic range sensor. After this sensing action, regions A, B and C become free, the black contour is a sensed obstacle and region D, occluded by the sensed obstacle remains unknown. Region E also remains unknown, but it is occluded by an already known obstacle.

marginal entropy reduction of q, is given by:

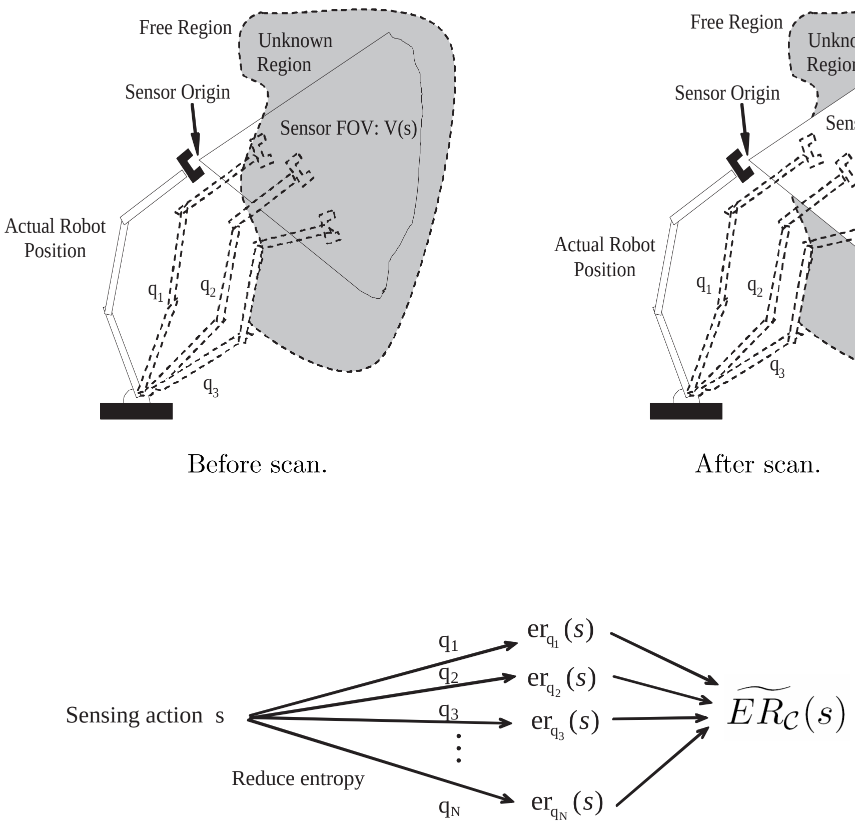
$$er_q(s) = -E\{\underset{s}{\triangle} H(Q)\} \tag{7}$$

As in Fig. 4, a sensing action s reduces the expected entropy of each unknown configuration, and the sum of the reductions approximates the expected C-space entropy reduction.

In the following, we are going to show how to compute the marginal entropy reduction term $er_q(s)$. Due to lack of space, we give only intuitive explanations. [15] gives detailed proofs.

3.1 $er_q(s)$ Computation

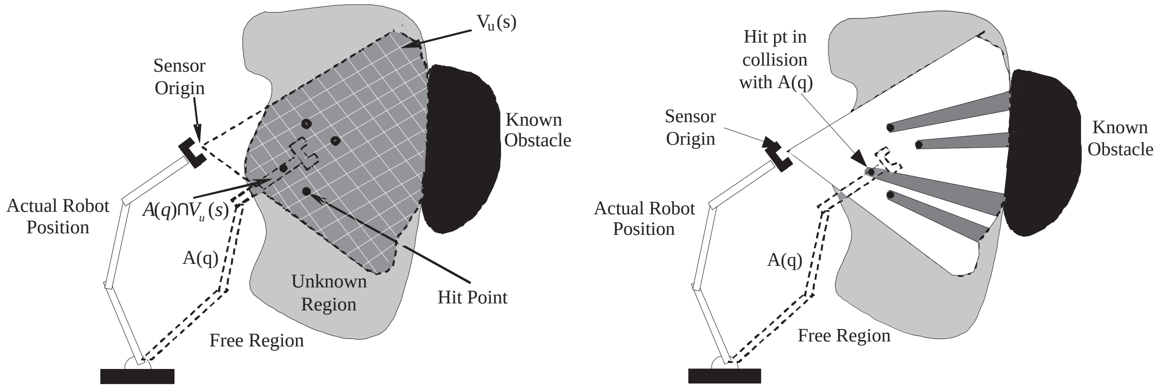
Recall that the environment is composed of free space and point obstacles, called *pt*. With occlusion constraints, the sensor only detects the very first point obstacle, called a hit point and denoted by *hpt*, along each sensing ray. Note the subtle but important difference between a *hpt* and a *pt*. For a robot configuration q, the possible outcomes for a sensing action s, can be grouped into event 1 and event 2. Event 1, Fig. 5, corresponds to cases where the sensor would sense at least one *hpt* inside $\mathcal{A}(q) \cap \mathcal{V}_u(s)$, the unknown region occupied by the robot (if it were at configuration q) that lies inside the sensor FOV, if the sensor were to be placed at sensor configuration s, i.e., $\exists hpt \in \mathcal{A}(q) \cap \mathcal{V}_u(s)$. Event 2, Fig. 6, corresponds to a set of outcomes in which there does not exist any *hpt* inside $\mathcal{A}(q) \cap \mathcal{V}_u(s)$. We accordingly decompose the entropy reduction computation into two parts: $er_q(s) = (er_q)_1 + (er_q)_2 = E\{\underset{event\,1}{\Delta} H\} + E\{\underset{event\,2}{\Delta} H\}$.



Before scan.

After scan.

Figure 4: A sensing action at configuration s. Assuming that the sensed region becomes free, the knowledge about each of the three robot configurations shown above is changed. q_1 is known free while q_2 and q_2 are still unknown. $er_q(s)$ provides a (expected) measure of this knowledge gained for each q. The expected C-space entropy reduction is simply a sum of these marginal terms.



(a) Before Scan

(b) After Scan

Figure 5: Computation of $(er_q)_1$. Event 1 refers to the cases where at least one hpt lies inside $\mathcal{A}(q)\cap\mathcal{V}_u(s)$. If event 1 were to occur, q would be in collision with a sensed obstacle (sensed hpt).

3.1.1 $(er_q)_1$ Computation

If event 1 were to happen, configuration q would be in collision with a sensed obstacle (the sensed *hpt*). So $H(Q \mid \text{event } 1) = 0$ (since $p(q|\text{event } 1)$, the probability of q being free conditional on event 1 is 0) and $\underset{event\,1}{\triangle} H(Q) = H(Q \mid \text{event } 1) - H(Q) = -H(Q)$.

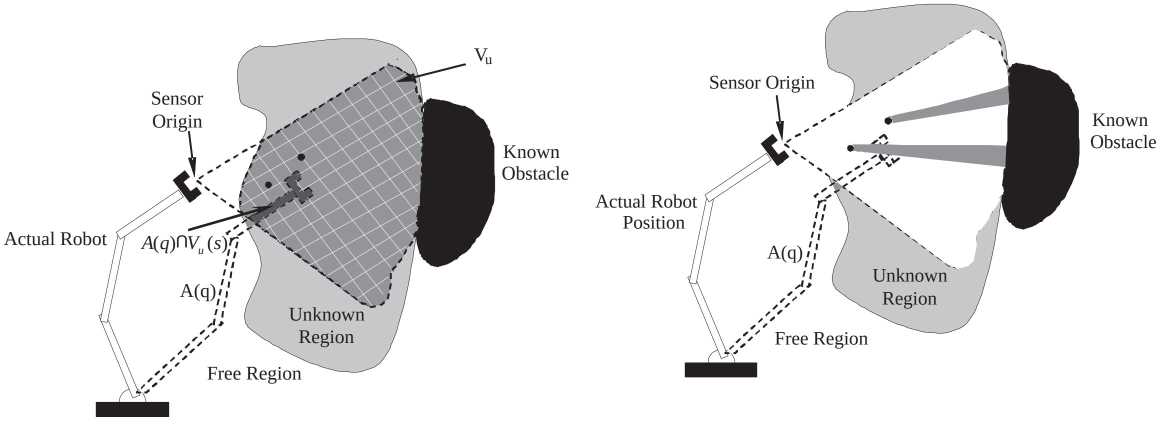
It turns out that the probability of event 1, $\Pr[\text{event } 1] = \Pr[\exists hpt \in \mathcal{A}(q) \cap \mathcal{V}_u(s)]$, is the same as $\Pr[\exists pt \in \mathcal{A}(q) \cap \mathcal{V}_u(s)]$, as if occlusion does not matter. Theorem 1 states this result formally.

Theorem 1: $\Pr[\exists hpt \in \mathcal{B}] = \Pr[\exists pt \in \mathcal{B}] = 1 - e^{-\lambda \cdot |\mathcal{B}|}$, where $\mathcal{B} \subseteq \mathcal{V}_u(s)$ is an open set.

See [15] for a rigorous proof. An intuitive explanation is as follows. For a *pt* not to be sensed, thus not becoming a *hpt*, it must be occluded by another *pt*, which requires exact alignment of points along the sensing ray, and the chances of that happening are rare, and in fact approach zero in the limit. Thus the probability that a $hpt \in \mathcal{B}$ is equal to the probability that a $pt \in \mathcal{B}$, which is easily computed using Eq. (5). And hence,

$$\begin{aligned} (er_q)_1 &= \Pr[\exists hpt \in \mathcal{A}(q) \cap \mathcal{V}_u(s)] \cdot H(Q) \\ &= \Pr[\exists pt \in \mathcal{A}(q) \cap \mathcal{V}_u(s)] \cdot H(Q) \\ &= (1 - e^{-\lambda \cdot |\mathcal{A}(q) \cap \mathcal{V}_u(s)|}) \cdot H(Q) \qquad (8) \end{aligned}$$

3.1.2 $(er_q)_2$ Computation



(a) Before Scan

(b) After Scan

Figure 6: Computation of $(er_q)_2$. Event 2 refers to cases where *hpt* (if it exists) lies in $\mathcal{V}_u(s) \setminus \mathcal{A}_u(q)$, the grid shaded region excluding the region in (a) occupied by the robot $\mathcal{V}_u(s) \cap \mathcal{A}_u(q)$. If event 2 were to occur, the whole $\mathcal{V}_u(s)$ except the regions occluded by sensed obstacles (*hpt*) would be sensed free, shown as white region in (b).

In this case, as shown in Fig. 6, the status of q would either remain unknown, albeit the unknown portion of $\mathcal{A}(q)$ may have decreased, or it may be known free; in either case, its collision status will not be obstacle.

For event 2, the closed form expression computation for entropy reduction is quite complicated; however, a closed form entropy reduction expression for a sub-event of event 2, called event $2'$, is easily calculated. Fortunately, it turns out that the difference in the entropy reduction between the two events is negligible in a probabilistic sense, and hence the closed form entropy reduction expression for event 2 is derived. Event $2'$ corresponds to those outcomes in which there does not exist any *pt* in $\mathcal{A}(q) \cap \mathcal{V}_u(s)$, or equivalently the region $\mathcal{A}(q) \cap \mathcal{V}_u(s)$ is free, as shown in Fig. 7. Note that this event $2'$ is fictitious since occlusion is ignored and event $2'$ is used purely for derivation purposes. As an immediate consequence of Theorem 1, event 2 and event $2'$ have the same probabilities, as stated in Lemma 1 below. The expression on the right hand side follows directly from Eq. (4).

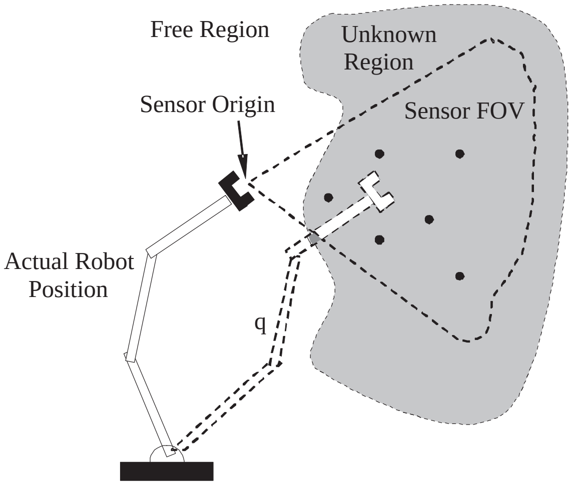


Figure 7: Event $2'$ corresponds to the cases for which there does not exist any *pt* in $\mathcal{A}(q) \cap \mathcal{V}_u(s)$. Thus, $\mathcal{A}(q) \cap \mathcal{V}_u(s)$, the white region inside the sensor FOV, becomes free.

Lemma 1: $\Pr[\text{event } 2] = \Pr[\text{event } 2'] = e^{-\lambda \cdot |\mathcal{A}(q) \cap \mathcal{V}_u(s)|}$.

Next, Theorem 2 states that the expected entropy of q given event 2 is the same as that given event $2'$, i.e., occlusion does not affect expected entropy.

Theorem 2: $E\{H(Q|\text{event } 2)\} = \Pr[\text{event } 2']H(Q \mid \text{event } 2') = e^{-\lambda \cdot |\mathcal{A}(q) \cap \mathcal{V}_u(s)|} \cdot H(Q \mid \text{event } 2')$.

See [15] for a rigorous proof. An intuitive explanation is difficult, but an outline is as follows. We first show that the expected newly sensed robot volumes (for robot at q) for event 2 and $2'$ respectively are the same. Intuitively, this is again due to the nature of point obstacles, i.e., that the probability of occluding nonzero volume approaches zero. Then properties of entropy functions — being bounded and concave — are then used to prove the above theorem.

Thus, combining Lemma 1 and Theorem 2, we have,

$$\begin{aligned} (er_q)_2 &= \Pr[\text{event } 2'] \cdot H(Q) - \Pr[\text{event } 2'] \cdot H(Q \mid \text{event } 2') \\ &= e^{-\lambda \cdot |\mathcal{A}(q) \cap \mathcal{V}_u(s)|} \cdot (H(Q) - H(Q \mid \text{event } 2')) \qquad (9) \end{aligned}$$

Summing the two components together, we have,

$$er_q(s) = (er_q)_1 + (er_q)_2 = H(Q) - e^{-\lambda \cdot |\mathcal{A}(q) \cap \mathcal{V}_u(s)|} \cdot H(Q \mid \text{event } 2') \tag{10}$$

$H(Q \mid \text{event } 2')$ and $H(Q)$ in the above equation are determined using Eqs. (1), (5) and $p(q \mid \text{event } 2') = e^{-\lambda \cdot |\mathcal{A}_u(q) \setminus \mathcal{V}_u(s)|}$.

Finally, summing over the unknown C-zone,

$$\widetilde{ER}(s) = \sum_{q \in \chi_u(\mathcal{V}_u(s))} H(Q) - e^{-\lambda|\mathcal{A}(q) \cap \mathcal{V}_u(s)|} H(Q|\text{event } 2') \tag{11}$$

With simple algebraic manipulations, we can easily show that Eq. (11) reduces to Eq. (6) as the sensor FOV $\mathcal{V}(s)$ approaches a single point in the limit [15].

4 View Planning Algorithm: Generic FOV Sensor based MER

Now that we have computed an expected entropy reduction expression for a given sensor action, we can use the MER criterion to decide the next scan. The algorithm then is as follows:

```
for every s    /* according to a certain resolution */
    determine $\mathcal{V}_u(s)$
    $\widetilde{ER}(s) = 0$    /* initialize */
    for every q    /* according to a certain resolution */
        if ($q \in \mathcal{C}_u^i \wedge \mathcal{A}_u(q) \cap \mathcal{V}_u(s) \neq \phi$)
            compute $er_q(s)$ /* Eq. (10) */
            $\widetilde{ER}(s) = \widetilde{ER}(s) + er_q(s)$
$s_{max} = \arg\max_s (\widetilde{ER}(s))$
```

Determining quantities such as $\mathcal{A}_u(q)$ and $\mathcal{V}_u(s)$ involves geometrical computations. For instance, determining $\mathcal{V}_u(s)$ corresponds to computing the intersection of the sensor FOV with $\mathcal{P}_u$ while excluding portions of $\mathcal{P}_u$ occluded by known obstacles (before sensing action). For simple geometries (polyhedral, for example), these are relatively simple. In case of complex environments, discretized representations such as grids or octrees can be used, as was done in [27]. $er_q(s)$ is therefore easily computed. The iteration over q, i.e., summation over C-space to determine $ER(s)$, may be prohibitive for robots with many degree of freedoms. In this case, the summation can be carried out over a large enough number of random samples to approximate the entropy reduction [9]. There is a vast literature on monte carlo evaluation of multi-dimensional integrals [28]. We simply used empirically determined large enough number of samples[3]. See Section 5 for specific resolutions and sampling parameters, respectively for sensor's configuration space and C-space, used in our experiments. Note that this approximation error is of order $1/\sqrt{N}$, where N is the number of samples [28]. Thus the number of samples needed, for a given error bound, is independent of dimensionality of the robot C-space. The iteration over s, i.e., search over all sensor configurations for s_{max}, is proportional to the number of discretized sensing configurations.

[3] One can also estimate the number of samples needed for a given error bound. See [28] for details.

5 Experimental Results

We now compare generic FOV sensor based MER criterion with the point FOV sensor based MER criterion, for both the simulated 2-link Eye-in-Hand system and the real six-dof SFU Eye-in-Hand system. In [9], it was shown, via extensive experiments and simulations that view planning based on point FOV sensor MER criterion outperforms random view planning (to randomly choose a sensor configuration) and MPV in terms of C-space exploration efficiency. Here we simply compare the point FOV sensor based MER criterion, Eq. (6), with generic FOV sensor based MER criterion derived in this paper, Eq. (11).

The overall sensor-based planner used is SBIC-PRM (sensor-based incremental construction of probabilistic road map) reported in [8]. It consists of an incrementalized model-based PRM [29], that operates in the currently known environment; and a view planner that decides a reachable configuration within the currently known environment from which to take the next view. The two sub-planners operate in an interleaved manner. Following is a brief description of SBIC-PRM.

Before each scan, the planner places uniformly random samples in C-space and checks their collision status within known physical space. If more than a certain percentage of the samples are known, either free or in collision with known obstacles, the exploration succeeds. If not, the status-unknown samples are added to a list $\mathcal{L}$. Then the view planner is called to plan a sensor's configuration, which is reachable (within the current probabilistic roadmap) from the current robot configuration. The robot moves to this sensor configuration and take the next view. After the scan, the physical space known to the robot is updated. The collision status of all the samples in list $\mathcal{L}$ are checked. The obstacle ones are removed from the list. The free ones are made nodes in the probabilistic roadmap (PRM) [29] and checked for connectivity with their neighboring nodes. In this way, the roadmap is incrementally expanded.

Note that the number of samples in $\mathcal{L}$ whose status changes (from unknown to free/obstacle) can be regarded as a measure of the known C-space percentage. We use this to capture various view planning algorithms' efficiency in real experimental results for SFU Eye-in-Hand system.

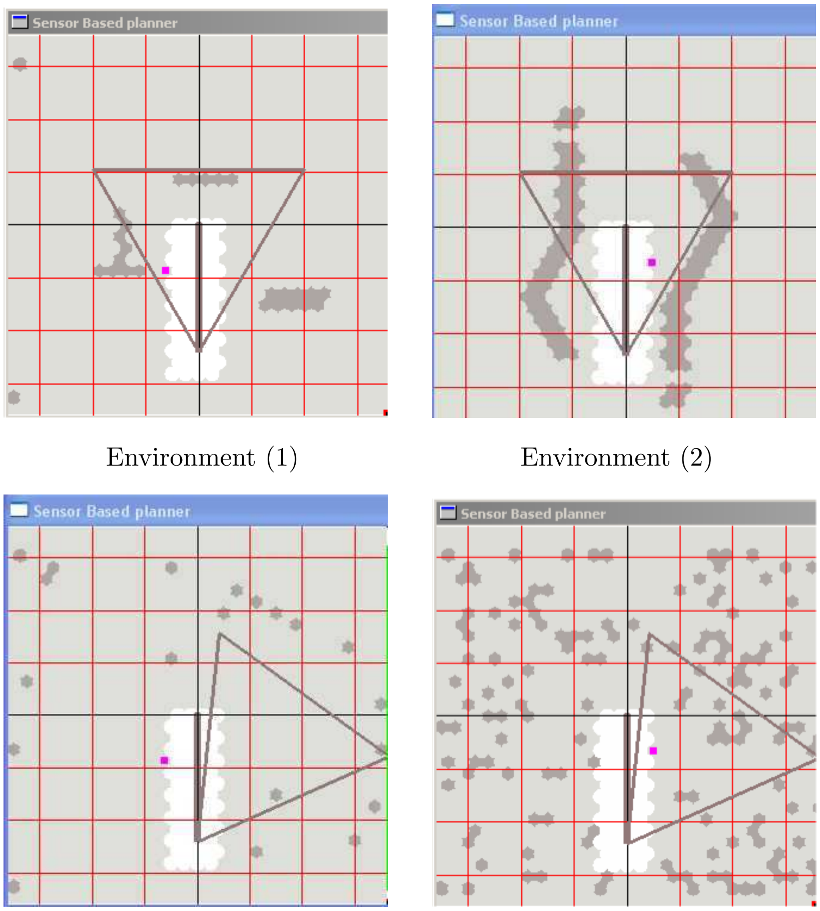
We implemented the following two different view planning criteria to compare their efficiencies to explore the C-space:

1. point FOV sensor based MER criterion: the view planner computes ERD function, Eq. (6), over the unknown physical space and choose x_{max} of the maximum value. It then places the center of the actual sensor FOV at x_{max}.

2. generic FOV sensor based MER criterion: the view planner computes the ER function and determines s_{max} (i.e., it executes the view-planning algorithm in Section 4). It then takes the next view at s_{max}.

5.1 Simulation Results for 2-link Robot

In the simulated two-link Eye-in-Hand system shown in Fig. 1 (i), the sensor has an additional dof, denoted by α, that allows it to rotate freely around the wrist. The robot's configuration is given as $q = (\theta_1, \theta_2)$, and the sensor's configuration $s = (q, \alpha) = (\theta_1, \theta_2, \alpha)$. The sensor FOV is a triangle with apex angle of 60 degree. The task for the robot is to explore its environment, starting from its initial configuration, pointing vertically downwards. In the simulation, the data structures for the physical space and the robot is similar to [10].[4] The simulation program, written in C++, runs on a Pentium III 800.



Environment (1)

Environment (2)

Environment (3)

Environment (4)

Figure 8: The three different physical spaces used in the simulation: (1) structured environment, (2,3) random generated environment.

We conducted a series of simulations for the robot in tens of different environments. For lack of space, only the results of four representative environments are presented. The results for the other environments were similar. The four environments, Fig. 8, are: two structured environments and two

[4] Briefly speaking, the physical space is represented by well structured ($30 * 30$ grids in this simulation) overlapping cells of radius r and the robot is represented by a collection of cells of radius $r/2$. While doing collision detection, all the physical space cells that have intersection with robot cells are checked to determine the status of this robot configuration.

unstructured ones with randomly generated obstacles, one with low density of obstacles and the other with a much higher density.

We carried out 10 runs for each environment and computed the averages of these C-space exploration rates and run times due to the random nature of underlying planner (SBIC-PRM). For lack of space we only show snapshots of the simulation results for environment (1) for some of the scans, Fig. 9. The left sub-image in each snapshot shows the physical space, and the right sub-image shows the C-space. In both physical and C-space, "grey" (or "green" in colored version), "white", and "dark" (or "blue" in colored version) regions denote unknown part $\mathcal{P}_u^i$ or $\mathcal{C}_u^i$, free part $\mathcal{P}_{free}^i$ or $\mathcal{C}_{free}^i$, and obstacles $\mathcal{P}_{obs}^i$ or $\mathcal{C}_{obs}^i$, respectively. It is clear that the generic FOV sensor MER criterion based view planning algorithm outperforms that based on point FOV sensor.

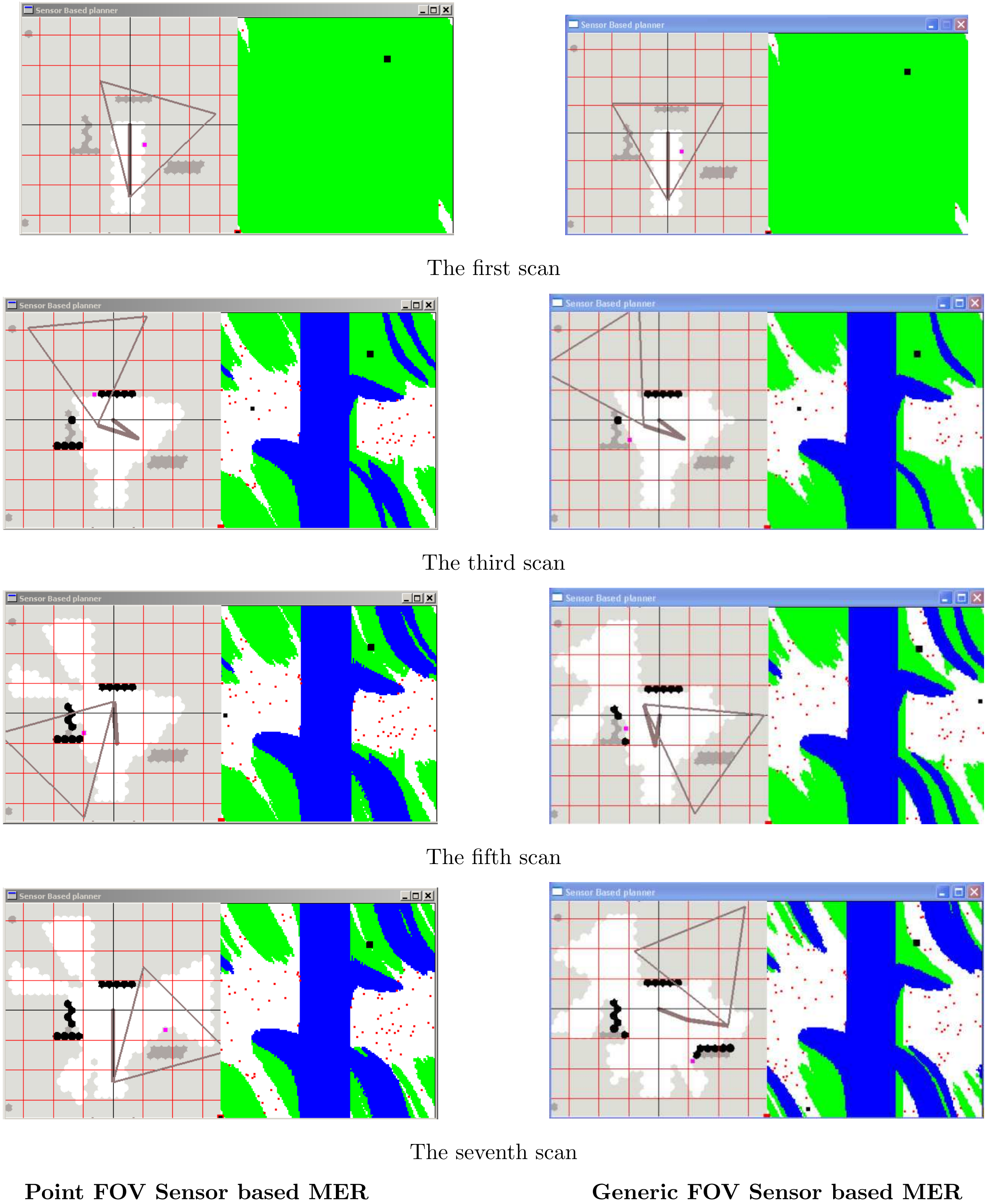
To quantify this improvement, in Fig. 10 we show the percentage of known C-space area vs. number of iterations (i.e., scans taken) for these two criteria for the four simulation environments. Since the C-space is two dimensional, the percentage in C-space is explicitly computed by discretizing the C-space up to a certain resolution (2.4 degrees per joint). Please note that this explicit computation of known C-space is for visualization only, the view planning algorithm does not compute it. Clearly generic FOV MER criterion performs better. For example, for the structured environment in the first plot, to expand the known C-space to 98%, point FOV sensor based MER criterion takes 17 scans while generic FOV sensor based MER criterion takes 10 scans. However, this increased efficiency comes at a computational cost. The average view planning time for generic FOV MER criterion is 51 seconds per iteration, while it is 22 seconds for the point FOV MER criterion.

5.2 Experimental Results on the SFU Eye-in-Hand System

The experimental test-bed is the six-dof SFU Eye-in-Hand system mentioned in Section 1 and shown in Fig. 1(ii). It consists of a PUMA 560 (hand) and an area scan laser ranger finder (eye) mounted on its wrist.

We conducted experiments for comparing the efficiency of C-space exploration using point FOV sensor based MER and generic FOV sensor based MER. The robot starts in the same start configuration shown in Fig. 1 (ii). The measure used for known C-space is the number of samples in $\mathcal{L}$ whose status is known. It is a rough but reasonable measure, given that explicit computation of the 6 dimensional C-space is computationally intensive, and hence impractical. The data structure for the physical space is as in [8], i.e., the physical space is represented by a binary valued (each leaf node is obstacle/free) octree and after each scan, the range image acquired by the robot is integrated into the octree. The parameters involved in the implementation are listed in Table 1. Again we conducted ten runs for the environment shown in Fig. 1 (ii). The view planner and the overall planner are implemented on a Pentium II 450 and the run-time results are based on this.

Fig. 11 shows the snapshots of robot taking scans at some of the view configurations planned by the view planner using generic FOV sensor based MER criterion. It takes five scans to reach the goal



The first scan

The third scan

The fifth scan

The seventh scan

Point FOV Sensor based MER **Generic FOV Sensor based MER**

Figure 9: The simulation results for environment (1) with point and generic FOV sensor based MER criteria respectively: the snapshots. The figure shows both the planned next scan (the triangle shown in the figure) and the physical space and C-space expansion after the scan of last iteration.

configuration between two "wall" obstacles. Note that the plan was biased toward goal as in [9] for this example. Fig. 12 compares the exploration efficiency of the two criteria. We can clearly see the planner based on generic FOV sensor based MER outperforms that based on point FOV sensor based MER in

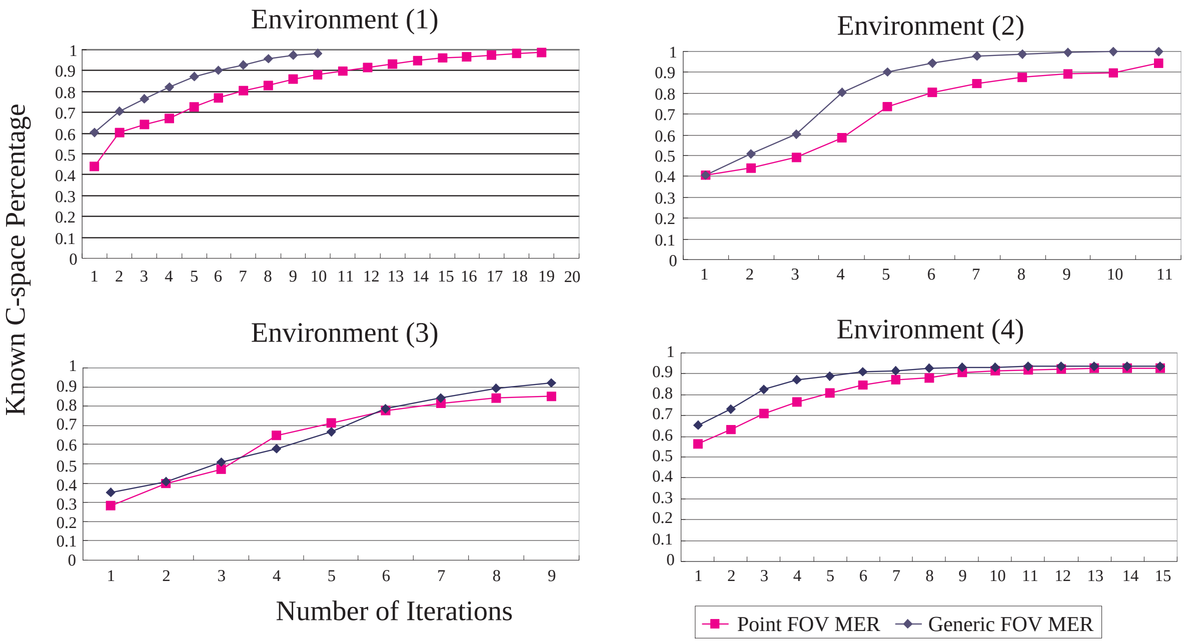


Figure 10: Comparison of C-space exploration efficiency for view planning criteria: point FOV sensor based MER and generic FOV sensor based MER. The data shown are average performance for ten tests conducted.

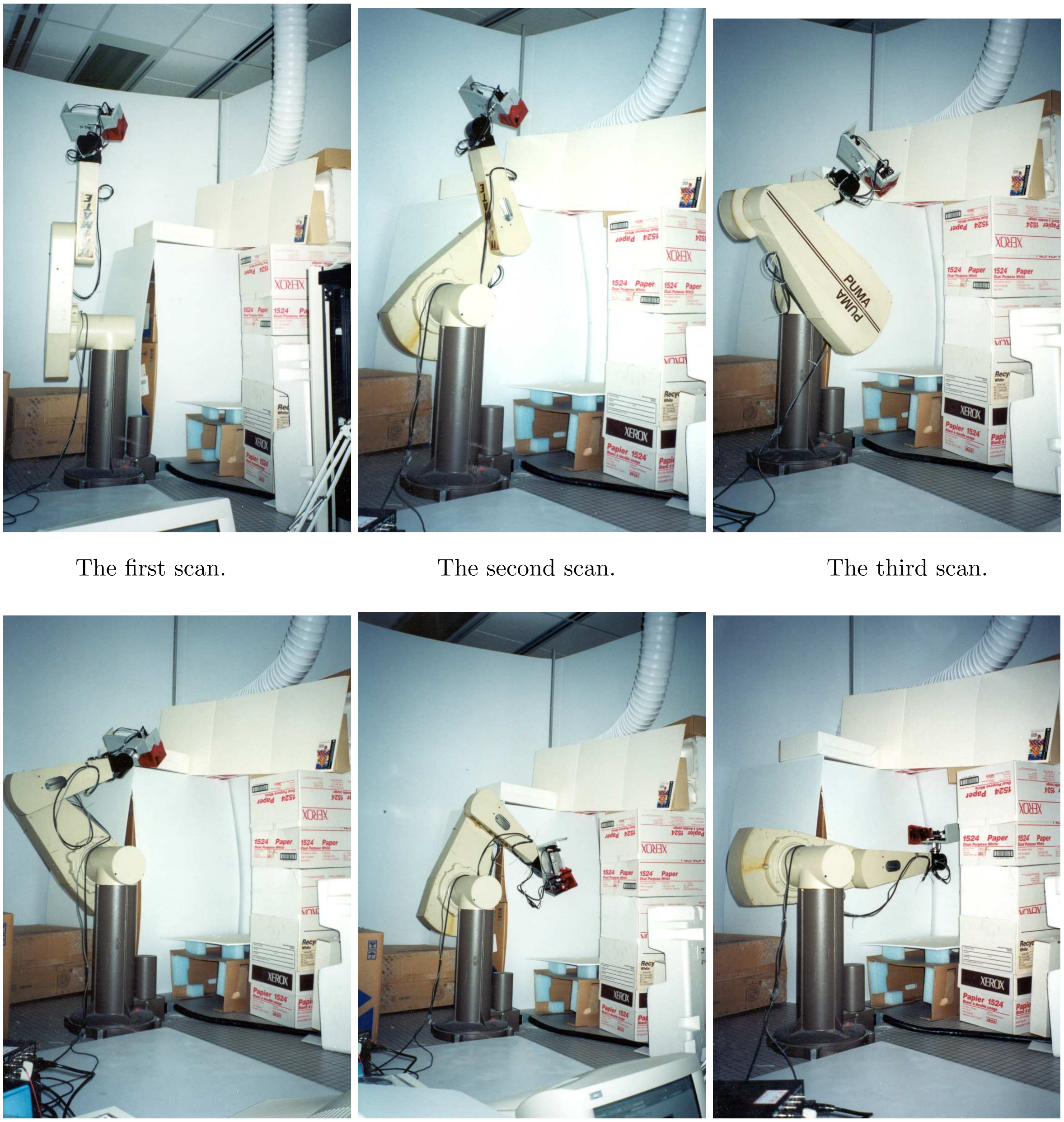
Parameter	Value
Range image resolution	256 × 256
Physical space (octree)	192cm × 192cm × 192cm
Octree resolution	1.5 cm
Number of samples for roadmap	10,000 / iteration
Number of samples for MER	20,000 / iteration
λ: Obstacle density	0.5

Table 1: Implementation parameters for experiments on SFU Eye-in-Hand systems.

that it explores (makes known) a significantly larger portion of the C-space.

As in the simulations, this increased C-space exploration efficiency for each view taken, comes at a computational cost. Table 2 shows the average of the number of iterations, the total view planning, and the total running time for reaching the goal. The total running time includes both the view planning time and the additional time for executing the whole sensor-based planner, which includes time for sensor scanning, known physical space and roadmap update, roadmap searching, and executing the robot movements. This additional time is roughly invariant with respect to the view planning criteria and is roughly 1.5 minutes per iteration in our implementation.

Again as in the 2D simulations, generic FOV sensor based MER is computationally more expensive.



The first scan. The second scan. The third scan.

The fourth scan. The fifth scan. Reach goal after the fifth scan.

Figure 11: Experimental results based on generic FOV sensor based MER criterion. Each snapshot shows robot taking a scan, and after five scans, the robot is able to reach the goal configuration.

However, since generic FOV MER based planners take less number of views, the efficiency will be justified where actual sensing (i.e., taking the scan) is relatively slow, as is often the case in practice. Using a faster computer and with more efficient implementations, the computational overhead can be reduced, thereby making the generic FOV based view planner time competitive. Certainly, the generic FOV sensor based MER will result in greater energy efficiency, given that each scan requires the robot to physically move to a view configuration, thereby expanding additional energy.

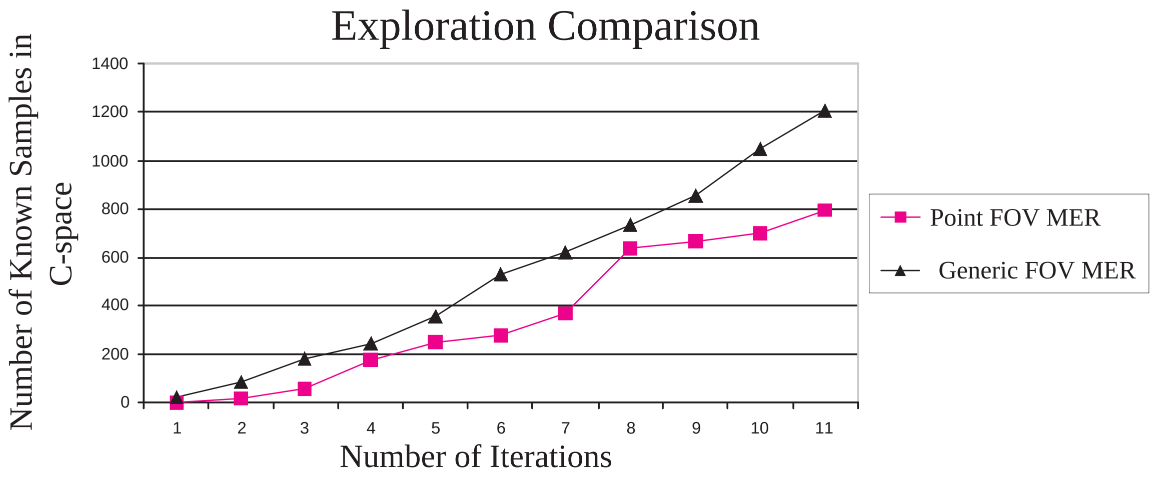


Figure 12: C-space exploration comparison for SFU Eye-in-Hand system.

View Planning Criterion	Point MER	Generic MER
Average View Planning Time (seconds)	123	921
Average Running Time (seconds)	220	1018
Average Number of Iterations	5	8

Table 2: SFU Eye-in-Hand Result: The average view planning and running time (in seconds) for point and generic FOV sensor based MER criteria.

6 Conclusion

This paper is a generalization of our previous work on efficient view planning algorithms based on C-space entropy notion for sensor-based motion planning [8, 9]. It was shown there that MER criterion results in significantly more efficient exploration over former view planning algorithms. However, two unrealistic assumptions were made in getting the closed form expression for the criterion, (1) the sensor FOV is a point, and (2) no occlusion constraints are taken into account. In this paper, we relax these two assumptions and derive MER results based on generic non-zero volume range sensor that is subject to occlusion constraints, thereby modeling a real range sensor. Planar simulations on a 2-link eye-in-hand system, and experiments on real six dof SFU Eye-in-Hand system show that the generic FOV sensor MER based view planning results in more efficient C-space exploration.

7 Future Work

There are a few different directions for further extending our results. The first concerns a more realistic stochastic model for obstacle distribution. One could consider Boolean model [24], which deals with distribution of geometric shapes (rather than points as in Poisson model). However, computing closed form analytical expressions for Boolean model seems difficult due to the increased complexity of probability computations with this model. Another possibility would be to consider occupancy grids [26], which have been extensively used in mobile robot domain, and which have the added advantage that we could incorporate sensor noise into view planning criteria. See [30] for work in this direction. However, as mentioned earlier, these models make the MER computation intractable. We are currently looking into it.

The closed form expression derived for the MER criterion, Eq. (11), merits further scrutiny. It combines two important aspects: (i) effect of sensing action on C-space, and (ii) this effect is measured in an information theoretic sense (using entropy). The former is purely geometric and the latter is stochastic. We have recently looked at it in further detail and tried to delineate the effect of the two components [25].

Current work on MER criteria assumes that the sensor measures exact data, i.e., it is not subject to noise. Although this is reasonable for gross motion planning, as opposed to fine motion planning, using fairly accurate range sensor, e.g. the laser range finder in the SFU Eye-in-Hand system, for other sensors such as the stereo cameras, MER criterion should also incorporate sensory noise into view planning account. See [30] for works in this direction.

REFERENCES

[1] K. Gupta and A. P. del Pobil. *Practical Motion Planning in Robotics: Current Approach and Future Directions*. J. Wiley and Sons, 1998.

[2] H. Banos and J. C. Latombe. Robot navigation for automatic construction using safe regions. In *Proc. of International Symposium on Experimental Robotics*, pp. 395-404, 2000.

[3] K. R. Harinarayan and V. J. Lumelsky. Sensor-based motion planning for multiple mobile robots in an uncertain environment. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1485-1492, 1994.

[4] K. N. Kutulakos, C. R. Dyer and V. J. Lumelsky. Provable strategies for vision-guided exploration in three dimensions. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1365-1372, 1994.

[5] A. Makarenko, S. Williams, F. Bourgault, H. Durrant-Whyte. An experiment in integrated exploration. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 534-539, 2002.

[6] C. J. Taylor and D. J. Kriegman. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Transactions on Robotics and Automation*, 14(3): 417-426, 1998.

[7] S. Thrun, W. Burgard, and D. Fox. Chapter 17: Exploration, *Probabilistic Robotics*. MIT Press. 2005.

[8] Y. Yu and K. Gupta. Sensor-based probabilistic roadmaps: experiments with an eye-in-hand system. *Advance Robotics*, 14(6): 515-537, 2000.

[9] Y. Yu and K. Gupta. C-space entropy: a measure for view planning and exploration for general robot-sensor systems in unknown environments. *International Journal of Robotics Research*, pp. 1197-1223, December 2004.

[10] J. Ahuactzin and A. Portilla. A basic algorithm and data structure for sensor-based path planning in unknown environment. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 903-908, 2000.

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

[12] P. Renton, M. Greenspan, H. Elmaraghy, and H. Zghal. Plan-n-Scan: a robotic system for collision free autonomous exploration and workspace mapping. *Journal on Intelligence And Robotic Systems*, 24:207-234, 1999.

[13] Y. Yu and K. Gupta. On eye-sensor based path planning for robots with non-trivial geometry/kinematics. In *IEEE International Conference on Robotics and Automation*, pp. 265-270, 2001.

[14] P. Wang and K. Gupta. Computing C-space entropy for view planning based on beam sensor model. *IEEE/RSJ IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2389-2394, 2002.

[15] P. Wang. View Planning Via Maximal C-Space Entropy Reduction *M.A.Sc. Thesis*, Engineering Science, Simon Fraser University, 2003.

[16] H. Choset, B. Mirtich, and J. Burdick. Sensor based planning for a planar rod robot: incremental construction of the planar rod-HGVG". In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 3427-3434, 1997.

[17] W. Scott, G. Roth and J. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1): 64-96, 2003.

[18] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, Prentice-Hall.

[19] V. Caglioti. An entropic criterion for minimum uncertainty sensing in recognition and localization — Part I: theoretical and conceptual aspects, *IEEE Transcations on Systems, Man, and Cybernetics — Part B: Cybernetics*, 31, no. 2 (2001): 187-194.

[20] C. Olivier and O. Dessoude. Heterogeneous sensor cooperation for an advanced perception system. In *Proc. of International Conference of Advanced Robotics*, pp. 1275-1280, 1991.

[21] N. Roy, W. Burgard, D. Fox and S. Thrun. Coastal navigation - mobile robot navigation with uncertainty in dynamic environments, In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 35-40, 1999.

[22] V. Sujan and S. Dubowsky. Efficient information-based visual robotic mapping in unstructured environments. *The International Journal of Robotics Research*, 24(4): 275-293, 2005.

[23] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* J. Wiley & Sons, 1991.

[24] D. Stoyan and W.S. Kendall. *Stochastic Geometry and Its Applications.* J. Wiley & Sons, 1995.

[25] P. Wang and K. Gupta. "A configuration space view of view planning". To appear *Proc. IEEE/RSJ IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[26] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation, *IEEE Computer*, vol. 22, no. 6, pp. 46-57, June 1989.

[27] Y. Yu and K. Gupta. An efficient online algorithm for direct octree construction from range images. In *IEEE International Conference on Robotics and Automation*, pp. 3079-3084, 1998.

[28] S. Haber. Numerical evaluation of multiple integrals. *SIAM Review*, 12(4):478-526, 1970.

[29] 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*, pp. 556-580, 1996.

[30] M. Suppa, P. Wang, K. Gupta, and G. Hirzinger. C-space exploration using noisy sensor models. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 4777-4782, 2004.