Autonomous Robots manuscript No.
(will be inserted by the editor)

Localization aware sampling and connection strategies for incremental motion planning under uncertainty

Vinay Pilania · Kamal Gupta

Received: date / Accepted: date

Abstract We present efficient localization aware sampling and connection strategies for incremental sampling-based stochastic motion planners. For sampling, we introduce a new measure of localization ability of a sample, one that is independent of the path taken to reach the sample and depends only on the sensor measurement at the sample. Using this measure, our sampling strategy puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining adequate samples in regions where uncertainty reduction is poor. This leads to a less dense roadmap and hence results in significant time savings. We also show that a stochastic planner that uses our sampling strategy is probabilistically complete under some reasonable conditions on parameters. We then present a localization aware efficient connection strategy that uses an uncertainty aware approach in connecting the new sample to the neighbouring nodes, i.e., it uses an uncertainty measure (as opposed to distance) to connect the new sample to a neighboring node so that the new sample is reachable with least uncertainty ("the closest"), and furthermore, connections to other neighbouring nodes are made only if the new path to them (via the new sample) helps to reduce the uncertainty at those nodes. This is in contrast to current incremental stochastic motion planners that simply connect the new sample to all of the neighbouring nodes and therefore, take more search queue iterations to update the paths (i.e., uncertainty propagation). Hence, our efficient connection strategy, in addition to eliminating the inefficient edges that do not contribute to better localization, also reduces the number of search queue iterations. We provide simulation results that show that a) our localization aware sampling strategy places less samples and find a well-localized path in shorter time with little compromise on the quality of path as compared to existing sampling techniques, b) our localization aware connection strategy finds a well-localized path in shorter time with no compromise on the quality of path as compared to existing connection techniques, and finally c) combined use of our sampling and connection strategies further reduces the planner run time.

Keywords Sampling strategy · Connection strategy · Planning under uncertainty · Incremental · Localization

1 Introduction

Safe execution of motion plans is of critical importance for many robotic tasks. As a result of uncertainty associated with a robot's motion and its sensory readings, the true robot state is not available. Therefore, a planning method must account for these uncertainties for safe and collision-free execution of motion plans. Partially observable Markov decision process (POMDP) (Kaelbling et al, 1998) is a general framework to deal with motion and sensing uncertainty, however due to its significant complexity, solving realistic problems with large state spaces remains a challenge, even though progress has been made on the efficiency issues of these approaches (Bai et al, 2014, 2015; Kurniawati et al, 2009, 2012; Pineau et al, 2003). A class of methods that carries robot state and associated uncertainty is an approximation to POMDP. Among them, a

Vinay Pilania · Kamal Gupta
Robotic Algorithms & Motion Planning (RAMP) Lab
School of Engineering Science, Simon Fraser University
Burnaby, BC V5A1S6, Canada
E-mail: vpilania@sfu.ca

Kamal Gupta
E-mail: kamal@sfu.ca

sub-class (Bouilly et al, 1995; Fraichard and Mermond, 1998; Lazanas and Latombe, 1995) assumes the presence of landmark regions in the environment where accumulated motion uncertainty can be "reset". Another sub-class (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Bry and Roy, 2011; Huang and Gupta, 2008; Lambert and Gruyer, 2003; Melchior and Simmons, 2007; Prentice and Roy, 2009) uses sampling-based methods (graph-based and tree-based) where uncertainty is propagated from start to goal. We call this type as sampling-based stochastic motion planners.

The sampling-based stochastic motion planners can be implemented either in an incremental (graph-based (Bry and Roy, 2011) or tree-based (van den Berg et al, 2011; Huang and Gupta, 2008; Melchior and Simmons, 2007)) or in a non-incremental way (graph-based (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Lambert and Gruyer, 2003; Prentice and Roy, 2009)). These planners are computationally demanding as compared to their counterparts that do not consider uncertainty (deterministic motion planners). This is because the paths no longer have the "optimal substructure" property, i.e., the incurred costs on different edges depend on each other. Note that the direct transformation of deterministic sampling-based methods to incorporate uncertainty breaks this property, because the underlying space is still the configuration space and the uncertainty is propagated from one configuration to another (Agha-mohammadi et al, 2014; van den Berg et al, 2011). To compute the cost of an edge emanating from a node, the full knowledge of belief (robot pose and associated uncertainty) at the node is required, this in turn requires full knowledge of the history of observations and actions leading up to the node. (Agha-mohammadi et al, 2014) is an exception in the sense that the incurred costs on different edges do not depend on each other. This comes at the cost of some simplifying assumptions including holonomic robot and Gaussian belief for robot states with trivial dynamics. The computational cost further increases if an edge cost in these planners uses collision probability (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Huang and Gupta, 2008), computation of which depends on the beliefs along that edge. Furthermore, this cost will go up drastically if collision checks are carried out in 3D (for example, for mobile manipulators). Since the time consuming step in stochastic motion planners arises from the uncertainty propagation along the edges, incremental stochastic planners can be computationally more demanding as compared to non-incremental ones where search mechanism is carried out only once while in former, search mechanism is repeated every time a new sample is added to the roadmap. For real time applications, for instance to facilitate anytime planning (Karaman et al, 2011), it is important to reduce this run time. At least part of this run time reduction can be achieved by "smart" sampling and connection strategies. Current stochastic motion planners (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Bry and Roy, 2011; Huang and Gupta, 2008; Lambert and Gruyer, 2003; Melchior and Simmons, 2007; Prentice and Roy, 2009) use traditional sampling and connection strategies which are designed for deterministic motion planners and address the issue of uncertainty at path search phase. These strategies add unnecessary nodes and edges that do not contribute to better localization. This leads to a dense roadmap which in turn increases the computational cost. We propose localization aware sampling and connection strategies to bring down the computational cost so that the incremental stochastic motion planners can be used for closer to real time applications.

The localization aware sampling strategy avoids putting large number of samples by considering the "localization ability" of a new sample relative to its neighbouring nodes in the roadmap. It puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining an adequate number of samples in regions where uncertainty reduction is poor. This leads to a less dense roadmap that results in significant time savings in the path search phase. Note that localization of a robot at a point depends on 1) the path taken to reach the point and 2) on the update based on sensor model. However, at the sampling stage the path taken to a node is not available. We develop a new measure of "localization ability of a sample" that "extracts" how well a sensor observation at a sample point reduces uncertainty without explicitly knowing the path leading to it and use this measure to design a localization aware sampling strategy.

A key reason we use reduction in uncertainty as a measure is that higher uncertainty is more detrimental and hence has higher cost for many tasks. Nevertheless, one possible consequence of our sampling technique is that path quality (we use true localization uncertainty along the path as a quality metric) may suffer, if the path passes through regions where uncertainty reduction is poor. Via simulation results, we show that, at least empirically, there is little compromise in path quality. Furthermore, note that since at the sampling stage, true localization uncertainty is not available, a cost function metric using it can not be computed, hence can not be used. The best one can do is to use the uncertainty reduction ability of the sensor at the sample point, as we do. Note that in the search phase (where

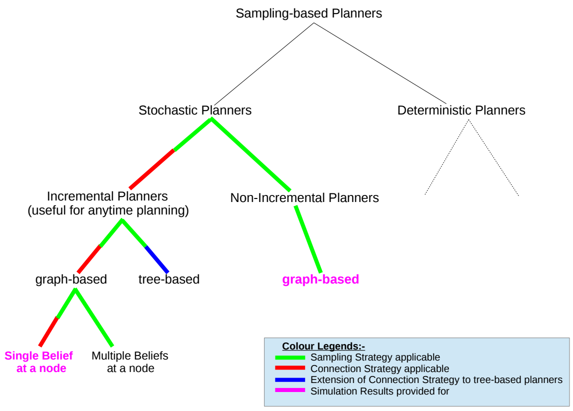

Fig. 1 It shows the class of planners for which our sampling and connection strategies can be used. For example, while going down the line from incremental planners to graph-based - our both approaches are applicable. Similarly, from incremental planners to tree-based - our sampling strategy and extension of connection strategy to tree-based planners are applicable. It also shows the type of planners for which we provide simulation results.

edges are added and uncertainty is propagate along the path), appropriate cost function is still minimized.

The localization aware connection strategy first connects the new sample to a nearest node (chosen based on an uncertainty metric and not on distance metric) and then to other neighbouring nodes. Connection from new sample to a neighbouring node is made only if the new path to that node reduces the uncertainty. Our efficient connection strategy eliminates the inefficient edges that would be created in current connection schemes (see Sec 2.2) but do not contribute toward better localization. As a result, it also reduces the number of search queue iterations needed to update the paths. This helps to find a well-localized path in shorter time with no compromise on the quality of path. Note that our strategy applies to graph-based incremental stochastic planners that maintain a single belief at a node and is not applicable to planners with multiple beliefs. Multiple beliefs at a node are needed for planners that optimize multiple objective functions since multiple paths to a node can not be completely ordered (as is the case for a single objective function, which can be a weighted sum of multiple costs) and need to be kept so as not to prematurely prune an optimal one, although domination criteria can be used to do some pruning (see (Bry and Roy, 2011; Huang and Gupta, 2009), the later is more specific to manipulator planning for fixed base). Of course, tree-based methods, by definition, have single belief since they have a unique path to any given node.

We provide simulation results by comparing incremental stochastic planners (we used RRBT (Bry and Roy, 2011)) with our sampling and connection strategies that show that a) our localization aware sampling strategy places less samples and finds a well-localized path in shorter time with little compromise on the quality of path as compared to existing sampling techniques, b) our localization aware connection strategy finds a well-localized path in shorter time with no compromise on the quality of path as compared to existing connection techniques, and finally c) combined use of our sampling and connection strategies further reduces the planner run time. Separately, we show that our localization aware sampling strategy is also helpful for non-incremental stochastic planners. Fig 1 clearly shows the class of planners for which we provide simulation results and where our contributions are applicable. The probabilistic completeness issues with our approaches are also discussed. We show that a stochastic planner that uses our sampling strategy is probabilistically complete under some reasonable conditions on parameters. Our connection strategy is also trivially complete.

2 Related work

In this section we review the related work and place our research work in context. First, we review the related work on sampling strategies, and then the work concerning connection strategies for sampling-based motion planning under uncertainty.

2.1 Sampling strategies

A large number of sampling schemes have been used with the standard (without uncertainty) sampling based planners (RRT or PRM) such as, sample around and near the obstacles, or in narrow corridors, medial axis sampling to sample far away from the obstacles, use visibility to reduce the number of samples, adaptive strategies such as restrict sampling to size-varying balls around nodes, entropy guided approaches, etc. (Hsu et al, 2006) and (Knepper and Mason, 2012) provide a survey of recent work in non-uniform sampling for PRMs. Above mentioned sampling approaches do not consider the uncertainty associated with robot and its sensors.

Missiuro and Roy (2006) proposed an approach where the sampling strategy incorporates mapping uncertainty (they do not consider localization uncertainty that we consider in this paper) in which the decision to accept or reject a sample is based on its collision probability (computed using each of the possible world model). However, the issue of “how good a sample would be in localizing the robot?", which we explicitly consider does not arise in their problem context. As mentioned earlier, computing the collision probability in the presence of localization uncertainty of a sample right at the sampling stage, i.e., before connecting it to the roadmap is not possible. Note that at sampling stage we consider only sensing uncertainty while for path search (where uncertainty is propagated from start) we consider both motion and sensing uncertainty. To the best of our knowledge, we are not aware of any other sampling approach that considers uncertainty. All sampling-based stochastic motion planners (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Bry and Roy, 2011; Huang and Gupta, 2008; Lambert and Gruyer, 2003; Melchior and Simmons, 2007; Prentice and Roy, 2009) use one of the sampling techniques from deterministic motion planners and address the motion, sensing, and mapping uncertainty at query phase where a search algorithm searches the roadmap by propagating uncertainty from start to goal.

Although not directly related to motion planning (or sampling techniques), the notion of uncertainty has been used in the past to select the best sample (the next best goal of robot) for search and exploration. For example: Stachniss et al (2005) first plans for each of the possible goal candidates and selects the one (as next best goal) which in addition to information maximization (unknown region), also has good localization along the path.

2.2 Connection Strategies

Connection strategies used in sampling-based deterministic motion planners simply connect the new sample to the neighbouring nodes within in a fixed size ball or size varying ball. A thorough discussion on these strategies can be found in (Choset et al, 2005a; LaValle, 2006) while for more recent updates we refer to (Hsu et al, 2006; Karaman and Frazzoli, 2011). These approaches do not account for uncertainty associated with robot and its sensors.

All (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Bry and Roy, 2011; Huang and Gupta, 2008; Lambert and Gruyer, 2003; Melchior and Simmons, 2007; Prentice and Roy, 2009) of the sampling-based stochastic motion planners that consider uncertainty inherit the connection strategy from deterministic motion planners. Among incremental planners, Bry and Roy (2011) is obliged to use traditional connection strategy as they optimized multiple objective functions, hence are required to maintain multiple paths to (hence multiple beliefs at) a node in order to guarantee not to prune an optimal path, although some pruning can be done via domination criteria. To the best of our knowledge, the work of Bry and Roy (2011) is the only roadmap (graph) based stochastic motion planner that works in an incremental fashion. Although it is designed for a set of beliefs, the same strategy also works for the case of single belief at a node. We call their algorithm (RRBT) with single belief as RRBT type framework (RRBT-TF). It minimizes the uncertainty at goal while respecting the chance-constraints (threshold on uncertainty) along the path. Planners in (Agha-mohammadi et al, 2014; van den Berg et al, 2011; Huang and Gupta, 2008; Lambert and Gruyer, 2003; Melchior and Simmons, 2007; Prentice and Roy, 2009) also use single belief at a node but they do not incrementally construct the roadmap.

The problem with the use of traditional connection strategy for incremental stochastic planners is that it even considers those edges which do not contribute toward better localization. With the inclusion of such edges, the planning time increases, however, the same quality of path can be find in lesser time if we eliminate these edges. This is exactly what our localization aware connection strategy does. It eliminates those edges which do not contribute toward better localization.

Similar to graph-based incremental stochastic planners, current tree-based stochastic motion planners (van den Berg et al, 2011; Huang and Gupta, 2008; Melchior and Simmons, 2007) also inherit the connection strategy from tree-based deterministic motion planners. There the EXTEND step simply connects the sample to nearest node (distance based) and then propagate the uncertainty to it. However, this does not provide the least uncertain path to the sample. Instead, our connection strategy will connect the sample to a neighbouring node (within a ball) the uncertainty propagation from which gives minimal uncertainty at the sample. Furthermore, we use the additional "rewiring" notion of RRT* (Karaman and Frazzoli, 2011), albeit with uncertainty metric, to rewire the connections to the neighbouring nodes. We extend RRT* to handle the uncertainty associated with a robot's motion and its sensors.

3 Contribution and problem statement

In this paper, we suggest more efficient localization aware sampling and connection strategies that are applicable to classes of planners as stated in Fig 1. We use an existing planner, RRBT basically as a tool (i.e., as a base planner) to demonstrate the efficiency gained by our strategies for incremental stochastic planners. We present three planners: i) RRBT-LAS, that demonstrates efficiency gained solely due to our sampling strategy, ii) RRBT-LAC, that demonstrates efficiency gained solely due to our connection strategy, and iii) RRBT-LASC, that demonstrates the efficiency gained due to combined effect of our sampling and connection strategies. The objective of these planners is to minimize the cost function while respecting the chance-constraints (threshold on uncertainty),

$$\min_{x_{0:T}} \left\{ \sum_{t=1}^{T} J(x_t) \right\}$$

subject to

$$P(x_t \in \chi_{obs}) < \delta, \forall t \in [0, T]$$

where x_t is robot state, χ_{obs} represents the states where robot is in collision with obstacles, δ is a user defined threshold that defines the chance-constraint and $J : x \rightarrow \Re^+$ is the cost function. The cost function we use for simulation results is the trace of covariance matrix.

4 Localization ability of a sample

In this section, we describe how to compute the localization ability of a sample. For this, we first briefly explain the extended Kalman filter (EKF) (Prentice and Roy, 2009) and then develop an expression for the localization ability of a sample.

Applying a control input u_t at time t brings the robot from state x_t at time t to state x_{t+1} at time $t+1$ according to a given stochastic dynamics model:

$$x_t = f(x_{t-1}, u_{t-1}, w_t), \qquad w_t \sim \mathcal{N}(0, W_t) \tag{1}$$

where w_t is the process noise at time t drawn from a zero-mean Gaussian distribution with variance W_t that models the motion uncertainty. After each motion, the robot receives noisy sensor readings z_t at time t that provide us with partial information about the state according to a given stochastic observation model:

$$z_t = h(x_t, q_t), \qquad q_t \sim \mathcal{N}(0, Q_t) \tag{2}$$

where q_t is the measurement noise drawn from a zero-mean Gaussian distribution with variance Q_t that models the sensor uncertainty.

We assume that the robot state is represented by Gaussian (μ, Σ) - μ being the mean and Σ being the covariance. The systems in our case are generally non-linear, therefore, the EKF linearizes f and h functions at each step. The EKF estimates the state at time t from the estimate at time $t-1$ in two separate steps: process step to propagate the applied control input u_{t-1}, and a measurement step to incorporate the obtained measurements z_t. The process step follows as:

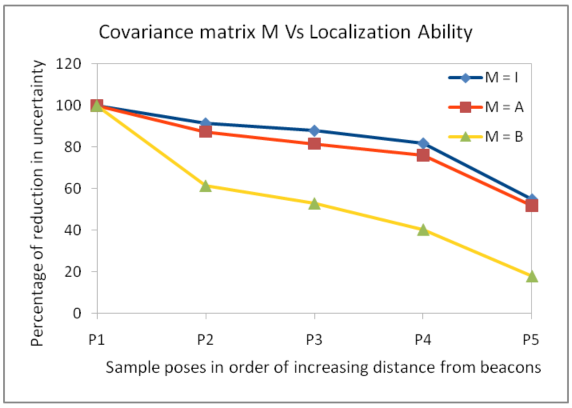

Fig. 2 Covariance matrix (M) Vs Localization ability (L_n) for five sample points (P1 to P5). I = [1, 1, 1], A = [1, 0.15, 1], B = [0.15, 0.15, 1].

$$\overline{\mu}_t = f(\mu_{t-1}, u_{t-1}) \tag{3}$$

$$\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t W_t V_t^T \tag{4}$$

where G_t and V_t are the Jacobian matrices of f with respect to x and w. Similarly, the measurement step follows as:

$$\mu_t = \overline{\mu}_t + K_t(h(\overline{\mu}_t) - z_t) \tag{5}$$

$$\Sigma_t = \overline{\Sigma}_t - K_t H_t \overline{\Sigma}_t \tag{6}$$

where H_t is the Jacobian of h with respect to x and K_t is known as the Kalman gain,

$$K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1} \tag{7}$$

Equation 4 propagates the uncertainty in robot state from Σ_{t-1} (at time $t-1$) to $\overline{\Sigma}_t$ (at time t) after incorporating control input and associated motion noise. Equation 6 further propagates it from $\overline{\Sigma}_t$ to Σ_t after incorporating sensor measurements and the associated sensor noise. It is Equation 6 that reduces the uncertainty with the help of meaningful measurements and it is this reduction in uncertainty, from $\overline{\Sigma}_t$ to Σ_t, that we are interested in capturing.

This can be achieved if we assume an a priori uncertainty at each sample point, say a covariance matrix M. Therefore Equations 6 and 7 will change to:

$$\Sigma_n = M - K_n H_n M \tag{8}$$

$$K_n = M H_n^T (H_n M H_n^T + Q_n)^{-1} \tag{9}$$

where subscript n stands for newly sampled robot pose.

To capture the localization ability, or reduction in uncertainty we need a norm on the covariance matrix. We used trace of the matrix, similar to Prentice and Roy (2009). Please note that the trace of a matrix is basically the sum of its eigenvalues (Ribeiro, 2004). Furthermore, the eigenvalues of the covariance matrix are proportional to the principal axes of equiprobability ellipsoids assuming Gaussian probability distribution. Please note that other metrics such as the determinant could also have been used. Once the metric is defined, the localization ability of a sample n is then given by $L_n = \frac{\|M\|-\|\Sigma_n\|}{\|M\|} \times 100$. The numerator part captures the reduction in uncertainty while the denominator acts as normalization factor. This measure (L_n) is applicable to Gaussian models, however, in Section 9, we extend it to multimodal distribution. Since we are using trace, therefore, we use a diagonal matrix for M, in fact an identity matrix.

Clearly, L_n in general depends on M. However, Fig 2 empirically shows that the L_n monotonically reduces irrespective of specific M (we chose three arbitrary M's) as the distance of samples from beacons increases, i.e., ability of sensor data to reduce uncertainty is reduced for all three different M. It is this trend that is important.

As mentioned in the introduction, L_n reflects just the sensor's ability to gather accurate information and not the actual localization uncertainty at the sample. The latter also depends on the path chosen and the accuracy of process model and can not be computed at sampling stage. Once the sample is connected to the roadmap, the true belief will be computed by search mechanism (uncertainty propagation from start). L_n is just a measure to accept or reject a sample.

5 Rapidly-exploring random belief tree with localization aware sampling strategy

In this section, we provide a Rapidly-exploring Random Belief Tree with Localization Aware Sampling Strategy (RRBT-LAS) algorithm where we replace uniform sampling of RRBT (Bry and Roy, 2011) with our localization aware sampling strategy. Please note that we did not consider linear-quadratic Gaussian (LQG) controller in RRBT-LAS simply because our focus is on showing efficiency of our sampling scheme and if RRBT-LAS is efficient (without LQG) then it will be more efficient after incorporating LQG that requires additional computation along the edges of the roadmap.

The algorithm operates on a set of nodes V and edges E, that define a roadmap in state space. Each node $v \in V$ has a state $v.x$, state estimate covariance $v.\Sigma$, a parent node $v.parent$, and localization ability v.loc. The state covariance prediction and chance-constraint checking (Bry and Roy, 2011) is implemented by a Propagate(e, v_{start}) routine that takes as arguments an edge and a starting node for that edge, and returns a covariance matrix at the ending node for that edge. If the chance-constraint is violated by the uncertainty at ending node, the function returns no covariance matrix. The comparison of partial paths at a node v is implemented by UpdateBelief(v, Σ) routine that updates the covariance matrix and parent node at v if the new path is less uncertain.

Algorithm 1: RRBT-LAS Algorithm

```
$v.x := x_{start}$; $v.\Sigma := \Sigma_0$; $v$.parent := NULL
$v$.loc := $tr(M)$
$V := \{v\}$; $E := \{\}$
while $i < P$ do
    ($x_{rand}$, loc_ability) := LocalizationBiasedSample()
    $v_{nearest}$ := Nearest($V, x_{rand}$)
    $e_{nearest}$ := Connect($v_{nearest}.x, x_{rand}$)
    if Propagate($e_{nearest}, v_{nearest}.\Sigma$) then
        $v_{rand}$.loc := loc_ability; $v_{rand}.x := x_{rand}$
        $V := V \cup v_{rand}$
        $E := E \cup e_{nearest}$
        $Q := Q \cup v_{nearest}$
        $V_{near}$ := Near($V, v_{rand}$)
        for all $v_{near} \in V_{near}$ do
            $E := E \cup$ Connect($v_{near}.x, x_{rand}$)
            $Q := Q \cup v_{near}$
        end
        while $Q \neq \emptyset$ do
            $u$ := Pop($Q$)
            for all $v_{neighbor}$ of $u$ do
                $\Sigma'$ :=Propagate($e_{neighbor}, u.\Sigma$)
                if UpdateBelief($v_{neighbor}, \Sigma'$)
                then
                    $Q := Q \cup v_{neighbor}$
                end
            end
        end
    end
    $i := i + 1$
end
```

Algorithm 2: LocalizationBiasedSample()

```
1  $x_{rand}$ := Sample()
2  $L_{x_{rand}}$ :=ComputeLocalizationAbility($x_{rand}$)
3  if $L_{x_{rand}}$ < LocAbilityTH then
4      $V_{neighbor}$ := Neighbor($V, x_{rand}$, DistTH)
5      for all $v_{neighbor} \in V_{neighbor}$ do
6          if $v_{neighbor}$.loc > $L_{x_{rand}}$ then
7              reject sample, go to step 1
8          end
9      end
10 end
11 return ($x_{rand}$, $L_{x_{rand}}$)
```

We also require the following routines: Sample() returns i.i.d. uniform samples, Nearest(V, v_{new}) takes the current set of nodes as an argument and returns the node in V that minimizes euclidean distance to v_{new}, and Near(V, v_{new}) returns every node within some ball centered at v_{new} of radius $\rho \propto (\log(n)/n)^{1/d}$ where n is

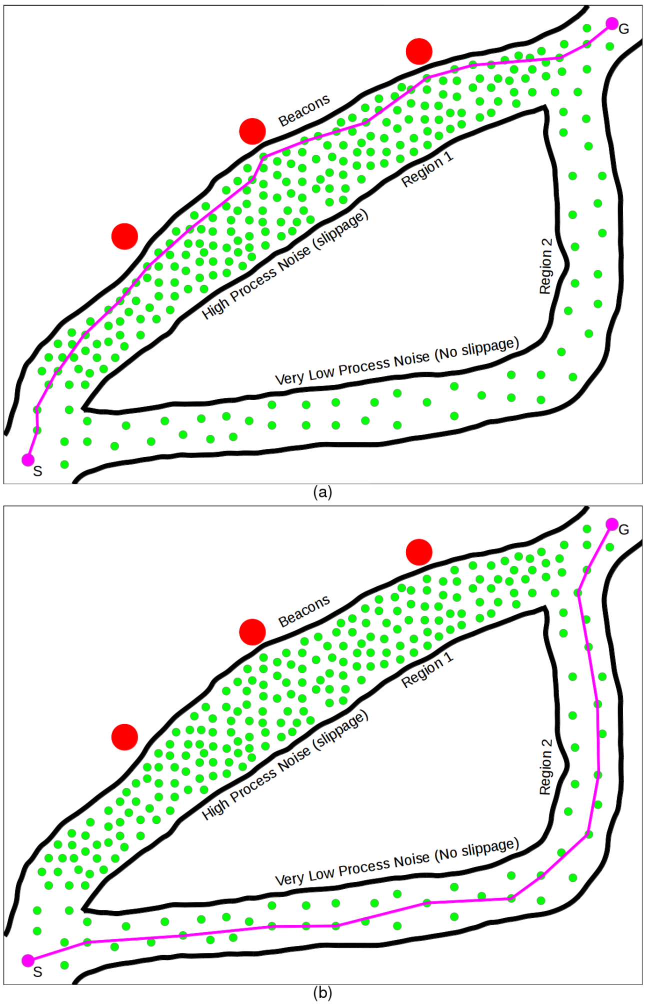
Beacons
Region 1
High Process Noise (slippage)
Region 2
Very Low Process Noise (No slippage)
S
G
(a)
Beacons
Region 1
High Process Noise (slippage)
Region 2
Very Low Process Noise (No slippage)
S
G
(b)

Fig. 3 Regions with high uncertainty reduction (better localization ability of sample points) also have high process noise. However, the path search phase mitigates this when the cost function is minimized. (a) a stochastic planner knows only one Gaussian process noise model, i.e., it does not know that Region 1 has high process noise and Region 2 has low process noise, therefore, a path is computed which passes through Region 1 (near beacons), (b) planner knows two Gaussian process noise models, one corresponding to each Region, the path search phase that minimizes uncertainty finds a path that passes through Region 2.

the number of nodes and d is the state dimension (See (Karaman and Frazzoli, 2010)).

5.1 Algorithm description

The RRBT-LAS algorithm is described in Algorithm 1. The roadmap is initialized with a single node with state x_{start}, covariance Σ_0 and its localization ability $tr(M)$ (trace of matrix M) from lines 1-3. At each iteration of the while loop, the roadmap is updated by sampling a new state using our localization aware sampling strategy (line 5), described in Sec 5.2, and then adding edges to the nearest and near nodes as in the RRG algorithm (Karaman and Frazzoli, 2010). Whenever an edge is added from an existing node to the new node, the existing node is added to the queue (lines 12 and 16). It should be noted that the new node is only added to the roadmap (along with the appropriate edges) if the chance-constraint can be satisfied by propagating an existing belief at the nearest node to the new sampled node as shown on line 8. After all the edges have been added, the queue is exhaustively searched from lines 18-26 using UPDATEBELIEF() routine. Note that the true belief of a node is computed during search mechanism (uncertainty propagation) from lines 20-25 which is different from its localization ability (line 5).

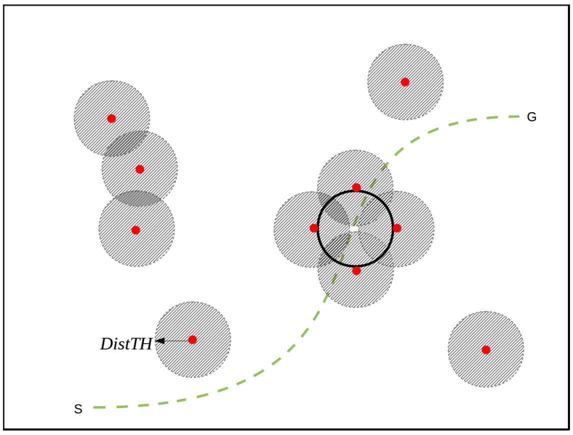

Fig. 4 Black color (bold) circle denotes one of the balls of radius r that is used to tile a path, red color dots represent randomly placed samples, and hatched region (of radius $DistTH = r$) around each sample denotes the restricted region where samples can not be placed according to heuristic used in localization aware sampling. This figure shows the situation where none of the sample has yet been placed inside the black ball, but some samples are placed at a distance d from its center such that $r < d < r + \epsilon$. Even in this worst case scenario, the probability of generating a sample, given by the ratio of volume of white region inside the black ball (after excluding the hatched region) and total volume of white region, is greater than 0. This ratio approaches one as more and more samples are placed outside the black ball.

5.2 Localization aware sampling strategy

Our localization aware sampling strategy puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining adequate samples in regions where uncertainty reduction is poor. The regions are decided based on a threshold “LocAbilityTH”, i.e., if the localization ability of a new uniformly sampled point is above this threshold, then the sample lies in regions with high uncertainty reduction and is simply added as a node. If the localization ability of a sample is below the threshold, the sample lies in regions with low uncertainty reduction, and the decision to accept or reject is governed by the localization ability of neighbouring nodes. If any neighbouring node within a ball of radius "DistTH" centered at the new sample has a localization ability above that of the new sample, the new sample is simply rejected, otherwise it is accepted as a node. This localization aware sampling strategy is implemented by routine LocalizationBiasedSample() as described in Algorithm 2.

A main motive behind accepting all the samples which lie within regions with high uncertainty reduction is to favour paths through such regions because they will likely result in high path quality. In the worst case, i.e., when the only way for a robot is to pass through regions of low uncertainty reduction (as explained in Fig 15 in Sec 8), path quality will be compromised since there are less samples in these regions (as validated in simulations, compromise is small), but at the same time we gain significant savings in planning time. Do note that if we decrease "DistTH" or "LocAbilityTH", our localization aware sampling strategy will converge to uniform sampling. Correspondingly, a higher DistTH results in faster run time, however, to retain probabilistic completeness, there is an upper bound (see Sec 5.3).

Furthermore, as we have mentioned before, localization uncertainty also depends on process noise at the sample point, information that is not available at sampling stage since it requires knowledge of path to the sample point. Assuming that the process noise is similar within the neighbourhood of a sample, accepting or rejecting a sample based on L_n (even though we note that localization depends on both steps of EKF) is defensible. It is indeed possible that regions with high uncertainty reduction ability of sensor may also have high process noise, hence the overall uncertainty may still be high. This is mitigated by the search phase of the planner where actual uncertainty is computed and the cost function (based on uncertainty) is minimized. See Fig 3 for such an example. In the figure, Region 1 has high process noise but our localization aware sampling strategy puts more samples because the localization ability of sample points is higher in that region as compared to Region 2 (which gets lesser samples). In (a) a stochastic planner does not know that Region 1 has high process noise and Region 2 has very low process noise, it just knows one Gaussian process noise model from which it samples the process noise while propagating the uncertainty from start to goal at path search phase. This is the case with all the stochastic planners. Therefore, a path is computed which passes through Region 1. However, it is not difficult to adapt the planner to different process noise models. In (b) the planner knows two Gaussian process noise models and which one to apply for a region at the path search phase. Therefore, it mitigates the overall uncertainty and finds a path which passes through Region 2.

5.3 Probabilistic completeness and optimality

First, we explain why we show probabilistic completeness for RRBT-LAS as generally the notion of completeness is defined useful for deterministic planners. The class of sampling-based stochastic planners that we are concerned with in this paper (including RRBT and thereby RRBT-LAS) first requires the construction of roadmap (or tree) in configuration space (C-space) and then followed by an uncertainty propagation step. This class has underlying collision checks with respect to nominal paths. Therefore, if a sampling strategy (as in our case for some value of DistTH) does not allow to connect the different regions of the C-space then the next step of uncertainty propagation can not be done and the planner will not find a path. That is why it is important to discuss the completeness of sampling strategy in exploring the C-space prior to uncertainty propagation step.

The probabilistic completeness is along the lines of (Choset et al, 2005b) and is essentially proved by assuming that a collision free path with clearance $\rho \geq 2r$ exists (where $2r$ is the radius of the largest inscribed circle within the robot), and then tiling it with a set of carefully chosen balls of radius r such that generating a sample in each ball ensures that these samples can be connected with collision-free edges and therefore, a collision-free path will be found. As shown in Fig 4, we can show that for $DistTH \leq r$ the probability of generating such samples approaches 1 as the number of samples increases. We show that RRBT-LAS is probabilistically complete under this reasonable restriction of DistTH. The complete proof is available in Appendix A.

As a result of the sampling strategy, RRBT-LAS, in general, will not be optimal.

6 Localization aware connection strategy

RRBT-TF inherits the connection strategy from the Rapidly-exploring Random Graph (RRG) (Karaman and Frazzoli, 2011) which is a deterministic motion planner. Fig 5 demonstrates how RRBT-TF uses this connection strategy for incremental stochastic motion planners. For a newly sampled point x_{new}, RRBT-TF propagates the uncertainty from nearest node (selected based on distance metric) to the new sample and if the uncertainty obtained satisfies the chance-constraints then the new sample and corresponding edge ($e_{nearest}$)

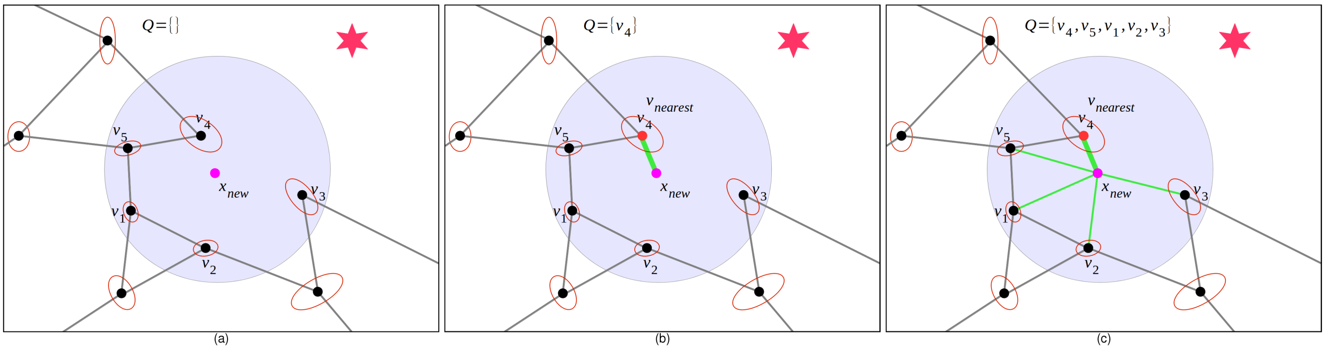

Fig. 5 These snapshots show the connection strategy of RRBT-TF. Red color star denotes beacon, red color ellipses denote belief nodes and Q denotes search queue. Furthermore, red color node denotes $v_{nearest}$, thick green line denotes $e_{nearest}$, and thin green lines denote e_{near}. (a) newly sampled point x_{new}; (b) $v_{nearest}$ (based on distance metric) is connected to x_{new} only if chance-constraints are satisfied and then inserted into Q, so far no belief update at x_{new}; (c) connecting x_{new} to all other neighbouring nodes and inserting them into Q, again no belief update at x_{new}. RRBT-TF now uses the search queue Q to iterate through the inserted nodes and update the beliefs.

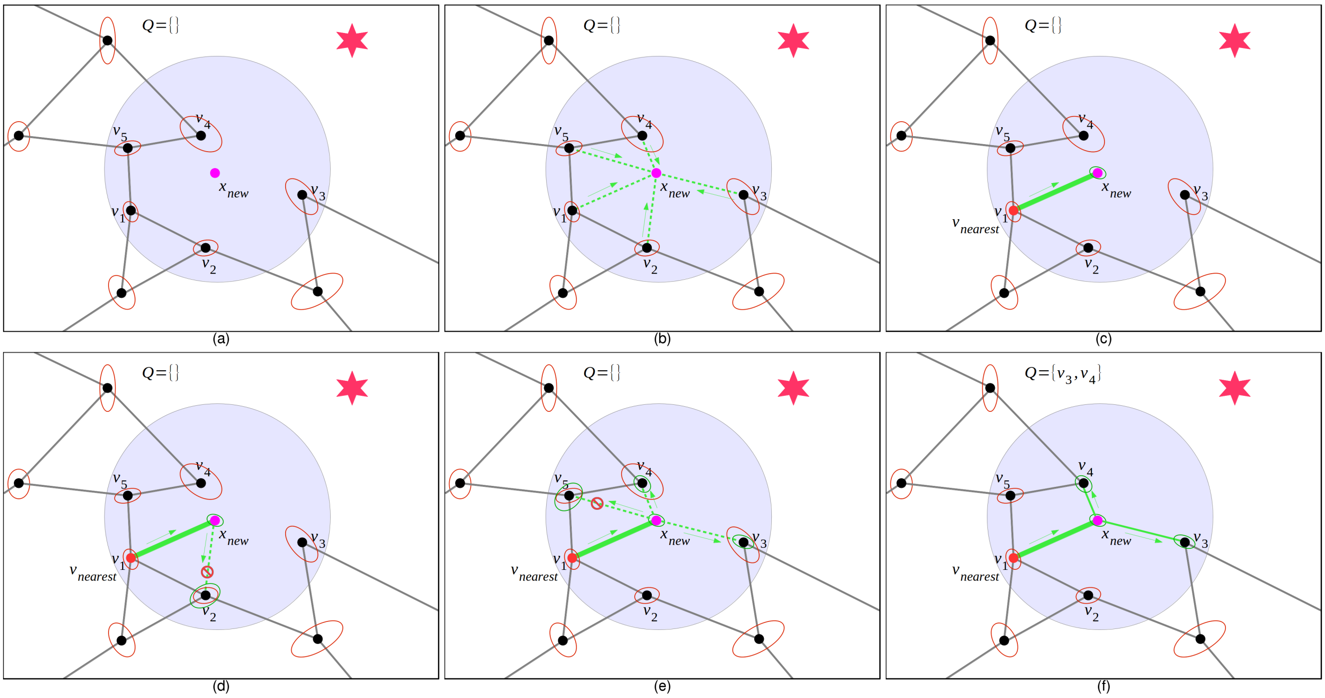

Fig. 6 The sequence of snapshots show our localization aware connection strategy. All the notations stated in Fig 5 also hold true in this figure. In addition, green color dashed lines denote the uncertainty propagation without actually adding those edges. The direction of uncertainty propagation is shown by green arrow. (a) newly sampled point x_{new}; (b) to search for $v_{nearest}$, uncertainty is propagated from neighbouring nodes to x_{new}; (c) $v_{nearest}$ is found (different from RRBT-TF), if chance-constraints are satisfied then $e_{nearest}$ is added and belief at x_{new} is updated (shown by green ellipse); (d) uncertainty propagation from x_{new} to v_2, new path to v_2 has more uncertainty, therefore, no update of belief at v_2 (also shown by red color cross mark); (e) similar attempts to connect x_{new} to other neighbouring nodes v_3, v_4, v_5; (f) successful edges (to nodes v_3, v_4) that reduce uncertainty are added along with belief updates at corresponding nodes. Note that only nodes v_3, v_4 are inserted into Q.

are added to the roadmap as shown in Fig 5 (b). Note that the nearest node is now inserted into search queue Q , however, there is no belief update at x_{new}. Only after successful connection of x_{new} to $v_{nearest}$, it proceeds further to connect x_{new} to other neighbouring nodes and insert them into Q as shown in Fig 5 (c). RRBT-TF then uses Q to iterate through inserted nodes and update the paths (uncertainty propagation).

Now we explain our localization aware connection strategy which is demonstrated in Fig 6. For a newly sampled point x_{new}, instead of connecting it to the distance metric based nearest node (v_4 in case of RRBT-

TF, Fig 5), we connect it to one of the neighbouring nodes such that the uncertainty propagation from that node to x_{new} gives minimal uncertainty at x_{new}. We call that neighbouring node as our nearest node. Fig 6 (b) and (c) demonstrate it: (b) shows the uncertainty propagation from neighbouring nodes to x_{new}; (c) shows that v_1 is selected as nearest node and it is connected to x_{new} (i.e., x_{new} and $e_{nearest}$ are added to the roadmap given that uncertainty obtained at x_{new} satisfies the chance-constraints). In contrast to RRBT-TF, we update belief at x_{new} and do not insert v_1 into Q. This is because a least uncertain path to x_{new} is already computed. Only after successful connection of x_{new} to $v_{nearest}$ (v_1), we proceed further to connect x_{new} to other neighbouring nodes (v_2, v_3, v_4, v_5). Different (and efficient) from RRBT-TF, which simply connect x_{new} to all other neighbouring nodes as in deterministic planner RRG, we connect x_{new} to only those nodes for which the new path (through x_{new}) reduces the uncertainty. Fig 6 (d) explains it better. In the figure, the uncertainty propagation from x_{new} to v_2 increases the uncertainty at v_2 (green ellipse is bigger than red ellipse), therefore x_{new} can not be connected to v_2. Similarly, in Fig 6 (e), the uncertainty propagation from x_{new} to v_3 and v_4 reduces the uncertainty at v_3 and v_4, therefore, x_{new} can be connected to these nodes. Fig 6 (f) shows the edges that are finally added to the roadmap along with the belief updates at corresponding nodes (shown by green ellipse). It also shows the nodes that are inserted into Q. Similar to RRBT-TF, the planner then uses Q to iterate through inserted nodes and update the paths.

In summary, our localization aware connection strategy does not add those edges along which localization is poor and also inserts less number of nodes into search queue. The combined effect of these two factors reduces the run time in our case with no impact on path quality. Also, the roadmap obtained using our connection strategy is a sub-roadmap of the RRBT-TF roadmap. These points can be verified by comparing Fig 6 (f) with Fig 5 (c).

6.1 Rapidly-exploring random belief tree with localization aware connection strategy

We provide a Rapidly-exploring Random Belief Tree with Localization Aware Connection Strategy (RRBT-LAC) algorithm where we replace connection strategy of RRBT-TF with our localization aware connection strategy.

Algorithm 3: RRBT-LAC Algorithm

```
1  $v.x := x_{start}$; $v.\Sigma := \Sigma_0$; $v$.parent := NULL
2  $V := \{v\}$; $E := \{\}$
3  while $i < P$ do
4      $x_{rand} :=$ SAMPLE(); $\Sigma_{rand} := \emptyset$
5      $V_{near} := \{\text{NEAR}(V, x_{rand})\}$
6      for all $v_{near} \in V_{near}$ do
7          $e_{near} :=$ CONNECT$(v_{near}.x, x_{rand})$
8          $\Sigma' :=$PROPAGATE$(e_{near}, v_{near}.\Sigma)$
9          if $tr(\Sigma') < tr(\Sigma_{rand})$ or $tr(\Sigma_{rand}) = 0$ then
10             $v_{nearest}{=}v_{near}$; $e_{nearest}{=}e_{near}$;
               $\Sigma_{rand}{=}\Sigma'$
11         end
12     end
13     if $tr(\Sigma_{rand}) \neq 0$ then
14         $V := V \cup v_{rand}(x_{rand}, \Sigma_{rand}, v_{nearest})$
15         $E := E \cup e_{nearest}$
16         for all $v_{near} \in V_{near} \setminus v_{nearest}$ do
17             $e_{near} :=$ CONNECT$(v_{rand}.x, v_{near}.x)$
18             $\Sigma' :=$PROPAGATE$(e_{near}, v_{rand}.\Sigma)$
19             if UPDATEBELIEF$(v_{near}, \Sigma')$ then
20                 $E := E \cup e_{near}$
21                 $Q := Q \cup v_{near}$
22             end
23         end
24         while $Q \neq \emptyset$ do
25             $u :=$ POP$(Q)$
26             for all $v_{neighbor}$ of $u$ do
27                 $\Sigma' :=$PROPAGATE$(e_{neighbor}, u.\Sigma)$
28                 if UPDATEBELIEF$(v_{neighbor}, \Sigma')$
                   then
29                     $Q := Q \cup v_{neighbor}$
30                 end
31             end
32         end
33     end
34     $i := i + 1$
35 end
```

The RRBT-LAC algorithm is described in Algorithm 3. The roadmap is initialized with a single node with state x_{start} and covariance Σ_0 from lines 1-2. At each iteration of the while loop (line 3), the roadmap is updated by sampling a new state and then adding edges to the neighbouring nodes using our localization aware connection strategy. Lines 6-12 correspond to Fig 6 (b) where we propagate the uncertainty from neighbouring nodes to the new sample in order to search for the nearest node. In lines 14-15 we actually add the new sample and the corresponding edge (connecting new sample to the nearest node) to the roadmap which corresponds to Fig 6 (c). Then from lines 16-23 we try to connect the new sample to the other neighbouring nodes and adding them to the search queue Q only if the corresponding connections reduce the uncertainty at these nodes as shown by the check on line 19. This corresponds to Fig 6 (d), (e) and (f). After all the successful edges have been added, the search queue Q is exhaustively searched from lines 24-32 where the queue iterates through all the inserted nodes and updates the paths.

6.2 Probabilistic completeness

The planner (RRBT-LAC) that uses our localization aware connection strategy uniformly sample a new state. Moreover, the connection (to a node) is avoided only if the new path does not contribute to better localization at that node as compared to its old path. Basically, the first connection to a node (whenever possible, i.e., collision-free) is always established. Therefore, RRBT-LAC is probabilistically complete and the proof is same as mentioned in (Choset et al, 2005b).

6.3 Asymptotic optimality

Unlike our sampling strategy, our connection strategy does not compromise on the path quality (as measured by path cost) under the assumption that the cost function used in the connection strategy (we have used trace of the covariance matrix) and the cost function used in the path search phase, i.e., path cost (we have used trace here as well) are the same. It eliminates only those partial paths for which there exists a better substitution path with respect to the cost function, i.e. smaller trace of the covariance matrix. However, if one replaces our cost function in the planning objective with a different cost function, our connection strategy also must use the same cost function to decide the connection of new sample to nearest node and other neighbouring nodes. Under this assumption, RRBT-LAC is asymptotically optimal on the similar lines as mentioned in RRBT.

7 Extending our localization aware connection strategy to tree-based planners

The tree version of our localization aware connection strategy uses the uncertainty metric to connect to the "nearest" node and then "rewires" (borrowing the "rewiring" notion of RRT* (Karaman and Frazzoli, 2011)). More formally, we propose the following key modifications in RRT* to handle stochasticity associated with a robot's motion and its sensory readings:

a) Instead of connecting the new sample to the distance metric based nearest node, we propose to search for a nearest node (and connect it to the new sample) as we demonstrate in Fig 6 (b) and (c) for incremental roadmap. This ensures a least uncertain path for the new sample.
b) To connect the new sample to other neighbouring nodes, we propose to use the similar strategy as we demonstrate in Fig 6 (e), however, it should use the "rewiring" notion in order to maintain tree structure. For example: after connecting the new sample to the nearest node (as described above), the uncertainty is propagated from the new sample to a neighbouring node, if it reduces the uncertainty at that node then the edge connecting that neighbouring node to its parent node is removed and the new edge connecting the new sample to that neighbouring node is retained. This ensures a least uncertain path from the root of the tree to a node.

8 Results

In this section, we provide simulation results for RRBT-LAS, RRBT-LAC and RRBT-LASC (RRBT with our localization aware sampling and connection strategies).

We used the motion model and sensor model from (Prentice and Roy, 2009) for all the planners. We experimented with two different sensor models: RangeModel 1 - where beacons have a limited range (we used 2 meters) and RangeModel 2 - where beacons have range that spans the entire map. For both sensor models, the sensor data has a distance varying Gaussian noise. First we report our results with RangeModel 2. Please note that our localization aware sampling and connection strategies also hold for complex measurement models, for example: range sensors.

We used 30 different seeds, each seed generating a set of 10000 pseudo random collision-free input samples. We ran all the planners on each set by varying the number of input samples from 100 to 10000 in incremental manner and provided our simulation results by averaging the outcome over 30 sets. Our implementation is in C++ under linux and runs on a Pentium dual core 2.5 Ghz computer with 4GB memory.

8.1 Simulation results for RRBT-LAS

We compared RRBT-LAS algorithm with RRBT-TF. We demonstrate our approach in two ways: (i) through visualization in Fig 7 - 10, we show the efficacy of our localization aware sampling strategy in judiciously placing the samples, and (ii) we use plots in Fig 13 and 14 to show that our localization aware sampling leads to saving in planning time with little compromise on the quality of path.

Fig 7 shows the placement of nodes in the roadmap (edges are not shown) for uniform sampling as we increase the number of input samples from a seed. The uniform sampling strategy is not aware of sensor model, therefore, the actual number of nodes in the roadmap are equal to the number of input samples added from a seed. Big red color balls (7 of them) in the snapshots denote the beacons which were used for localiza-

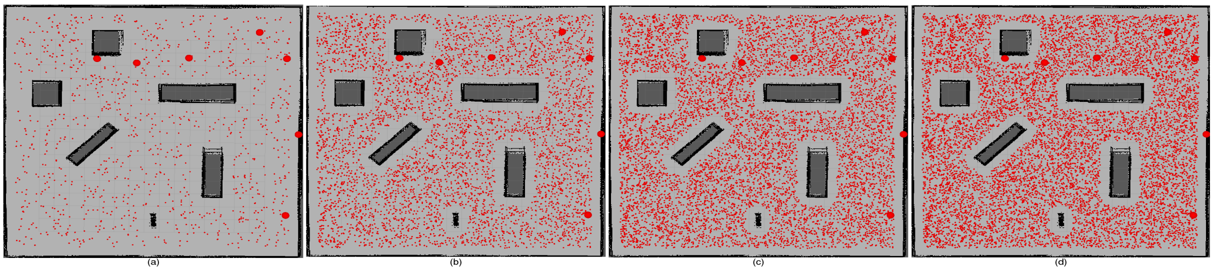

Fig. 7 RRBT-TF [uniform sampling]. # input samples from a seed [# actual nodes in the roadmap] : (a) 1000 [1000], (b) 5000 [5000], (c) 8000 [8000], (d) 10000 [10000]

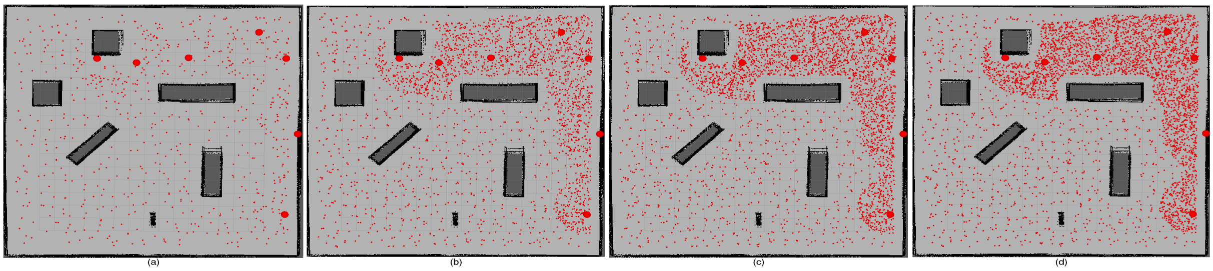

Fig. 8 RRBT-LAS [localization aware sampling] using RangeModel 2. # input samples from a seed [# actual nodes in the roadmap] : (a) 1000 [680], (b) 5000 [2331], (c) 8000 [3347], (d) 10000 [4060] . Here we used DistTH = 30 cm and LocAbilityTH = 90% (reduction in uncertainty).

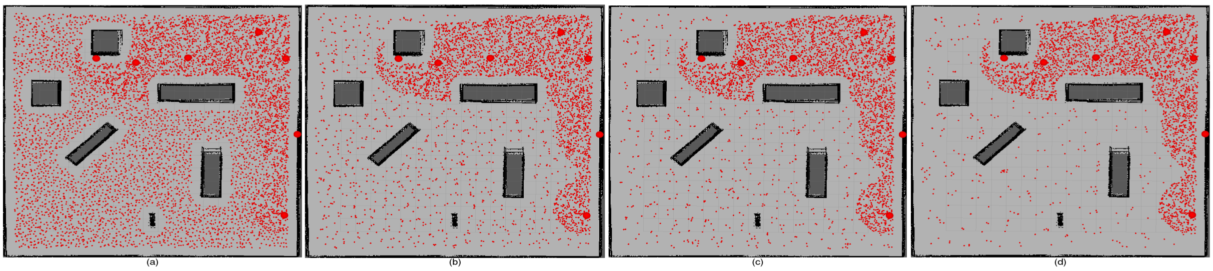

Fig. 9 Effect of varying DistTH (while keeping LocAbilityTH = 90% and the number of input samples from a seed as 10000) in RRBT-LAS using RangeModel 2. DistTH in (a) 10 cm, (b) 30 cm, (c) 40 cm, (d) 60 cm.

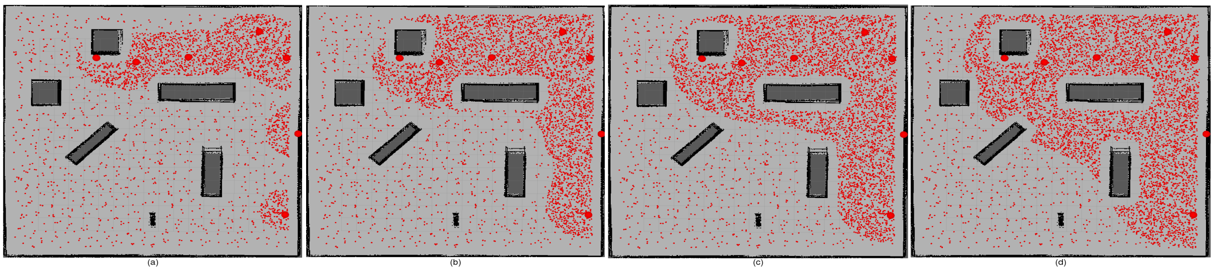

Fig. 10 Effect of varying LocAbilityTH (while keeping DistTH = 30 cm and the number of input samples from a seed as 10000) in RRBT-LAS using RangeModel 2. LocAbilityTH in (a) 93.3%, (b) 86.6%, (c) 83.3%, (d) 76.6%.

tion. Compared to uniform sampling (Fig 7), the actual number of nodes in our localization aware sampling strategy are significantly reduced as shown in Fig 8. From the figures, we observe that our localization aware sampling strategy places more samples in regions with high uncertainty reduction (near beacons) and eliminates unnecessary samples from regions with low uncertainty reduction. We also show the effect of varying two thresholds (DistTH and LocAbilityTH) in our localization aware sampling strategy. In Fig 9, we varied only DistTH and observed that the sparsity of nodes in regions with low uncertainty reduction increases with the increase of DistTH. However, the nodes in regions with high uncertainty reduction remain unchanged with the variation of DistTH. Similarly, in Fig 10, we varied only LocAbilityTH and observed that the area under regions

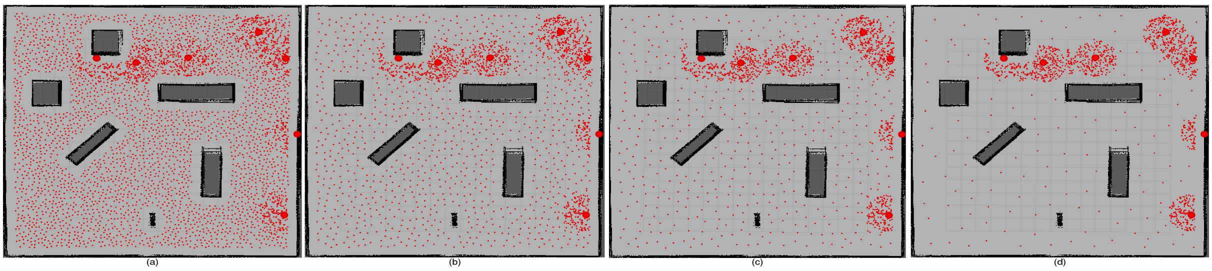

Fig. 11 Effect of varying DistTH (while keeping LocAbilityTH = 86.6% and the number of input samples from a seed as 10000) in RRBT-LAS using RangeModel 1. DistTH in (a) 20 cm, (b) 40 cm, (c) 50 cm, (d) 70 cm.

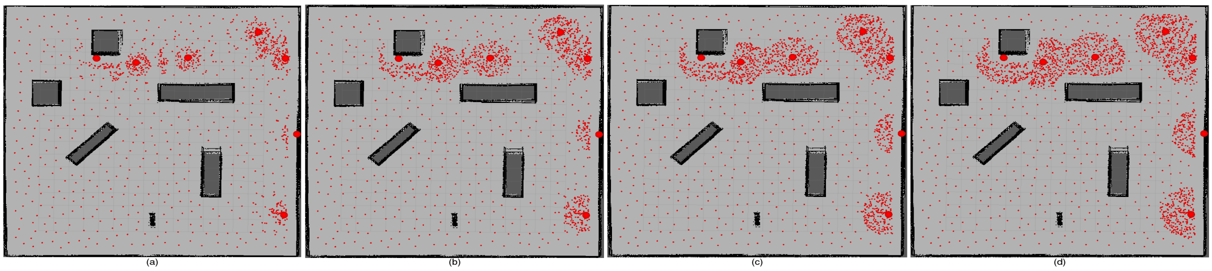

Fig. 12 Effect of varying LocAbilityTH (while keeping DistTH = 30 cm and the number of input samples from a seed as 10000) in RRBT-LAS using RangeModel 1. LocAbilityTH in (a) 93.3%, (b) 86.6%, (c) 83.3%, (d) 76.6%.

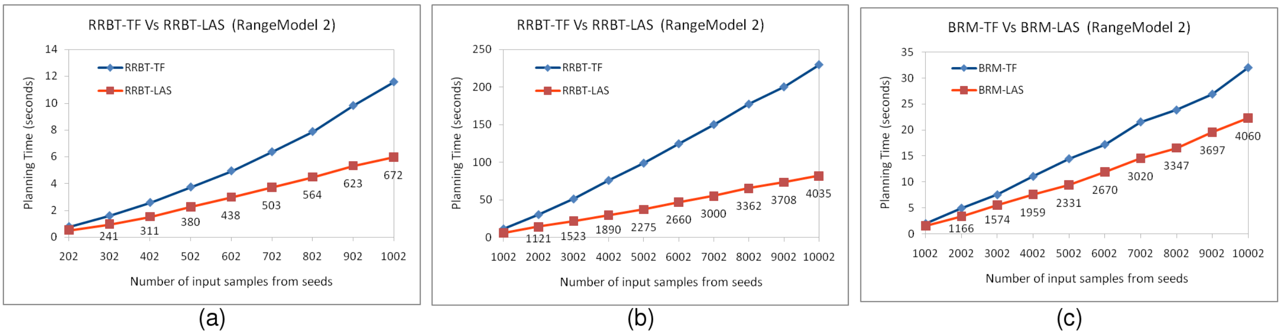

Fig. 13 Plots show the comparison of RRBT-TF Vs RRBT-LAS for incremental motion planning (a, b) and of BRM-TF Vs BRM-LAS for non-incremental motion planning (c). Data labels for each data point along red curves in RRBT-LAS and BRM-LAS show the actual number of nodes in the roadmap. For RRBT-TF and BRM-TF, actual number of nodes and number of input samples from seeds are the same, therefore, data labels are not shown along their corresponding curves. For the plots we used DistTH = 30 cm and LocAbilityTH = 86.6%. Also note that the saving in planning time is for $DistTH \leq r$ (where $2r$ is the inscribed radius of robot).

deemed as high uncertainty (higher than the threshold) reduction increases with the decrease of LocAbilityTH. In Fig 9 and 10, we kept the number of input samples from a seed as 10000, therefore, the comparison should be done with Fig 7(d).

In Fig 13 (a and b), we observe that RRBT-LAS reduces the planning time significantly as a result of our localization aware sampling strategy. This can be seen from the graphs as we move from 200 input samples to 10000 input samples, split over two sub-plots due to range of horizontal scales. We also observed that the run time savings increase supra linearly with the number of input samples. We talk about plot in Fig 13 (c) at the end of this section where we discuss the utility of our sampling strategy for non-incremental stochastic planners.

Furthermore, we compared the quality of paths generated by RRBT-TF and RRBT-LAS. We used two comparison metrics: (a) trace of covariance matrix at goal, (b) normalized sum of trace of covariance matrices along path. For worst case scenario, where the only way for the robot is to pass through regions with low uncertainty reduction (as shown in Fig 15), we observed little compromise on the quality of paths generated by RRBT-LAS as compared to uniform sampling of RRBT-TF. Plots in Fig 14 show the comparison. At 1000 input samples, RRBT-LAS has 10% more uncertainty along path and 5% more uncertainty at goal.

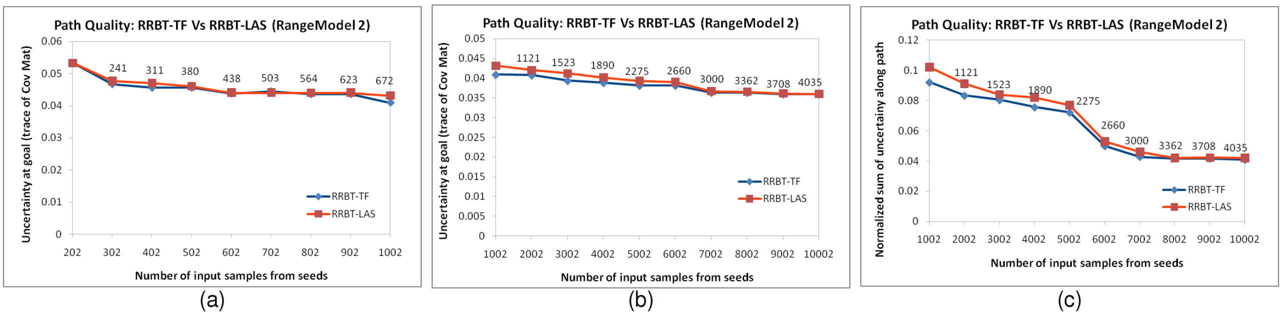

Fig. 14 Comparison of path quality between RRBT-TF and RRBT-LAS for a scenario where the only path to goal passes through regions with low uncertainty reduction, essentially a worst case scenario for path quality for our sampling scheme (see Fig 15). Y-axis denotes the trace of covariance matrix at goal in plots (a, b) and normalized sum of trace of covariance matrices along path in plot (c).

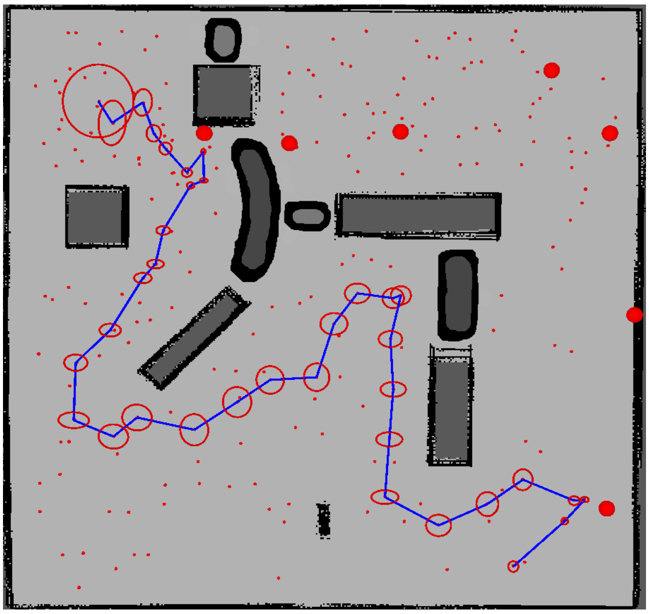

Fig. 15 It demonstrates the importance of maintaining an adequate number of samples in regions with low uncertainty reduction. In the figure, the regions with high uncertainty reduction are obstructed by obstacles, therefore a path was found which most of the time passes through regions with low uncertainty reduction (away from beacons). The red color ellipses show the uncertainty at waypoints along the path.

This is the highest degradation in path quality that we observed as we move from 200 input samples to 10000 input samples. Note that the path quality saturates at about 7000 samples for the considered case. The compromise (even if it is relatively small) on path quality comes from the fact that, for this scenario, the entire path passes through regions with low uncertainty reduction where we reduce the number of samples. It is reasonable to expect that for a large majority of scenarios, only portions of a path will pass through regions with low uncertainty reduction, hence the path quality compromise would be even smaller. This is the key reason that our localization aware sampling strategy simply accepts all the samples within regions with high uncertainty reduction (as shown in Fig 8 to 10).

We also implemented a non-incremental planner, Belief Roadmap (BRM) (Prentice and Roy, 2009) with our Localization Aware Sampling - we call the resulting planner BRM-LAS. However, our BRM implementation used the uncertainty propagation approach of van den Berg et al (2011) to propagate uncertainty along the edges (instead of using one-step transfer function approach of BRM). This is because the transfer function approach, although more efficient, assumes maximum likelihood observations along the path and therefore, can not infer the true a-priori probability distributions along the path. Since it is a straight forward modification of BRM, the corresponding pseudo code is not provided in the paper. We compared BRM-LAS algorithm with BRM-TF (original BRM with uniform sampling and uncertainty propagation approach of van den Berg et al (2011)). Fig 13 (c) shows the simulation results. From the plot, we observed that our localization aware sampling strategy reduces the planner run time for non-incremental planners as well (although the saving is not as large as for incremental planners).

8.2 Simulation results for RRBT-LAC

We carried out simulations to compare our RRBT-LAC algorithm with RRBT-TF. We demonstrate our approach in two ways: (i) through visualization in Fig 16, we show the efficacy of our localization aware connection strategy in reducing the number of edges in the roadmap, and (ii) we use plots in Fig 17 to show that our localization aware connection strategy leads to saving in planning time.

Fig 16 shows the roadmap and planned path for RRBT-LAC and RRBT-TF as we increase the number of input samples from a seed. The number of edges

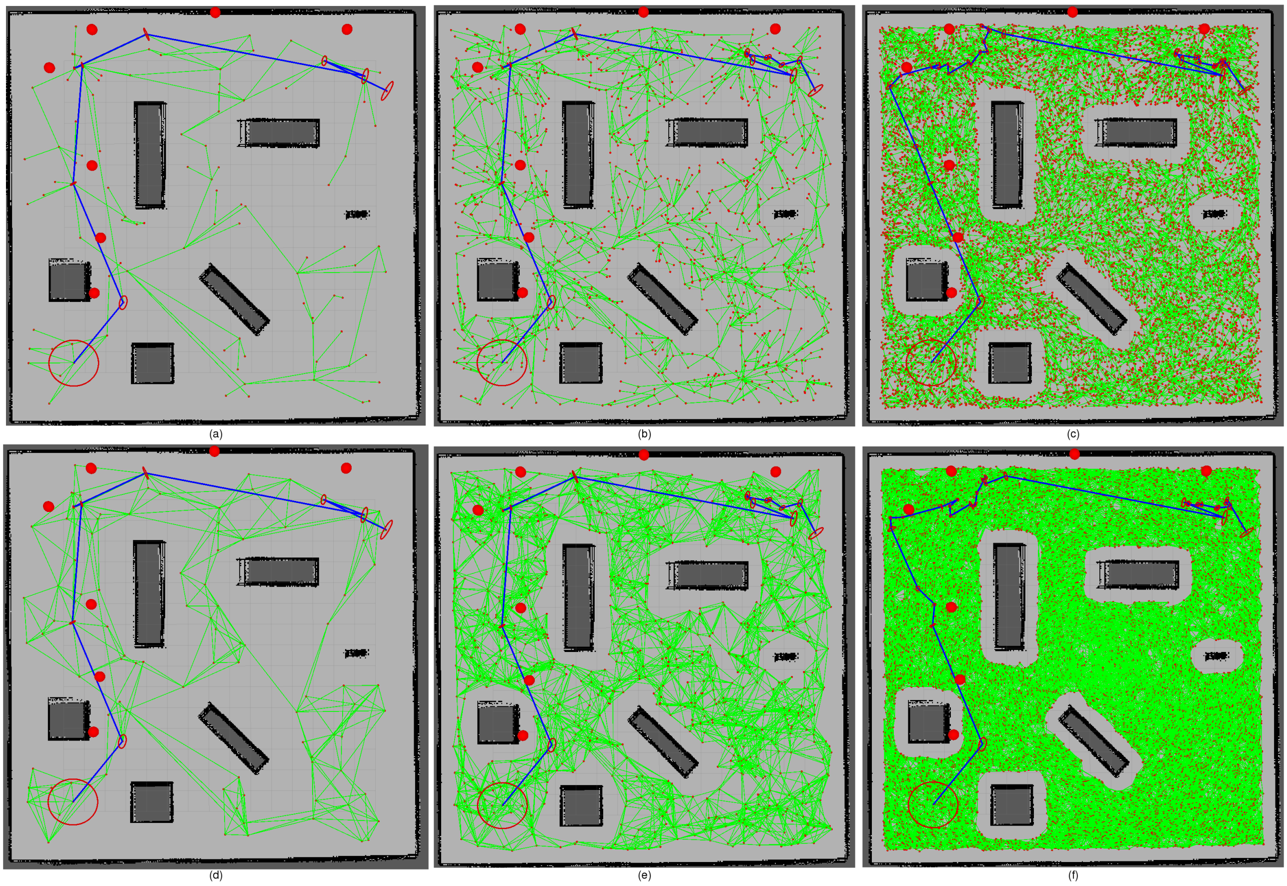

Fig. 16 RRBT-LAC Vs RRBT-TF. Big red color balls of circular shape represent the beacons. Top row (a, b, c) shows the roadmap and planned path (blue color) for RRBT-LAC where the # [nodes] and # [edges] are: (a) [100] and [121], (b) [1000] and [1338], (c) [10000] and [13373]. The bottom row (d, e, f) is for RRBT-TF where the # [nodes] and # [edges] are: (d) [100] and [308], (e) [1000] and [5007], (f) [10000] and [67405]. Note that the nature of well localized path remains same as we reduce the number of edges (which do not contribute toward better localization) in our RRBT-LAC.

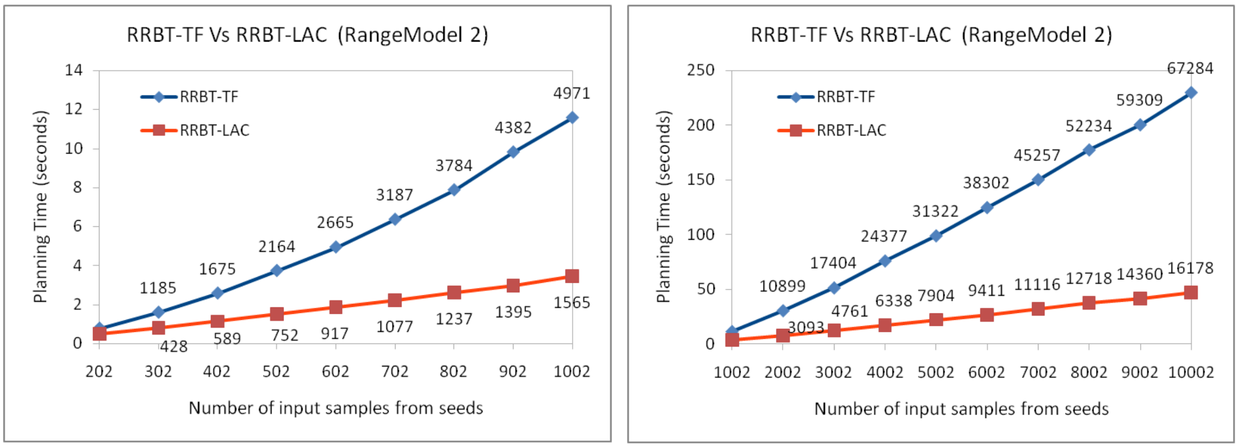

Fig. 17 Comparison of planning time for RRBT-LAC Vs RRBT-TF. Data labels for each data point along red curves in RRBT-LAC and blue curves in RRBT-TF show the number of edges in the roadmap. Please take a note of y-axis scale while comparing different graphs.

in the roadmap constructed by RRBT-LAC are significantly reduced as compared to the roadmap of RRBT-TF. This can be observed by comparing the roadmaps in (a), (d) and (b), (e) and (c), (f). From the figure, it is also important to note that the nature of paths planned by RRBT-LAC and RRBT-TF remains same, i.e., our

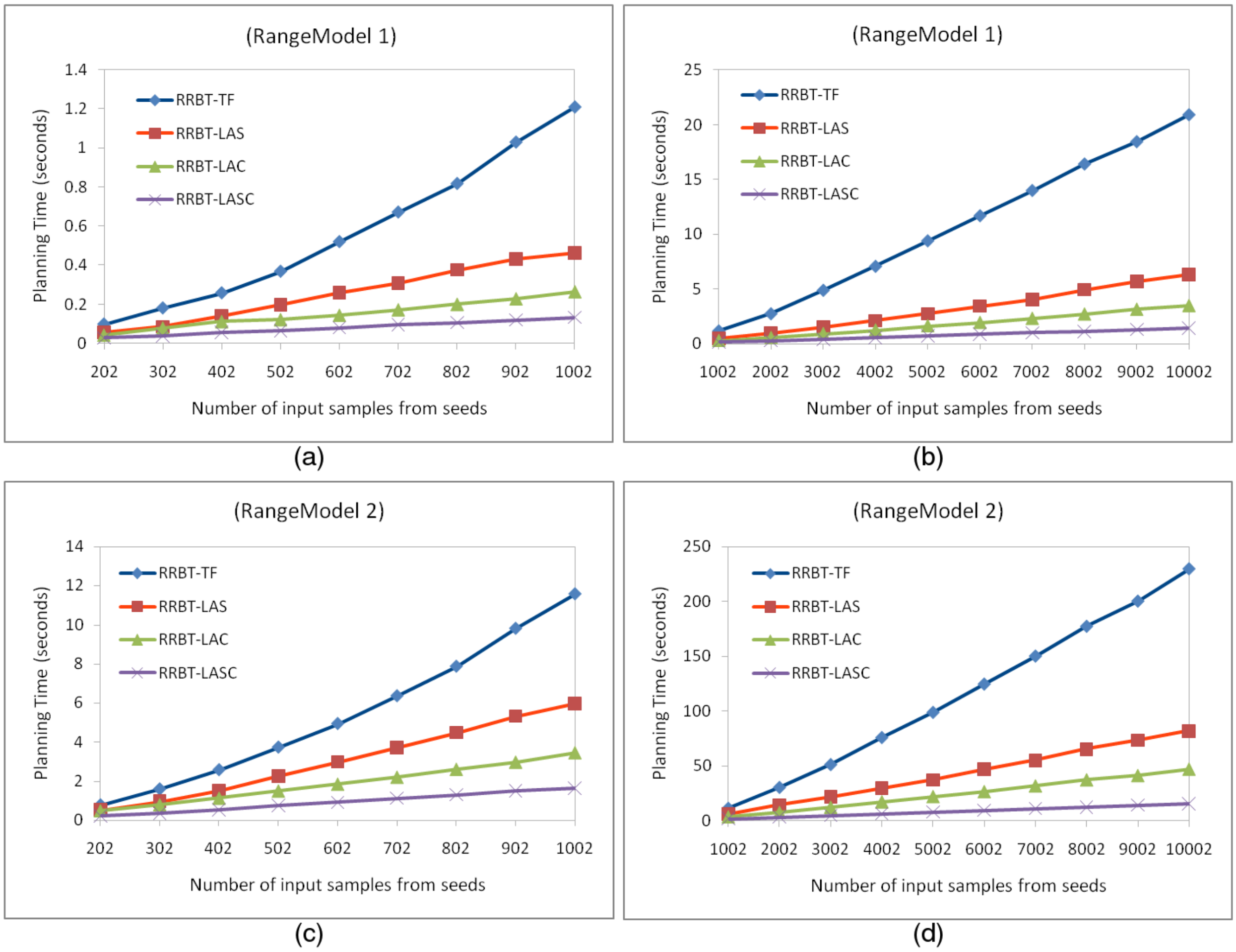

Fig. 18 Comparison of planning time for RRBT-TF Vs RRBT-LAS Vs RRBT-LAC Vs RRBT-LASC for RangeModel 1 and RangeModel 2. Please take a note of y-axis scale while comparing different graphs.

localization aware connection strategy used in RRBT-LAC does not compromise on the quality of path. This is because the edges, which are present in the roadmap of RRBT-TF but not in the roadmap of RRBT-LAC , do not help to reduce the uncertainty along those partial paths and therefore, are removed from the roadmap of RRBT-LAC. Fig 17 shows that RRBT-LAC reduces the planning time as a result of our localization aware connection strategy.

8.3 Simulation results for RRBT-LASC

Fig 18 (c), (d) shows the simulation results for the combined effects of the sampling and connection strategies where we replace the sampling and connection strategies of RRBT-TF with our localization aware sampling and connection strategies and call it RRBT-LASC. We observe that the run time savings increase supra linearly (from RRBT-LAS to RRBT-LASC) with the number of input samples.

8.4 Simulation results with different sensor model

Additionally, we also evaluated all the planners with a different sensor model (RangeModel 1) where beacons have a limited range (we used 2 meters). We observed similar behaviour as with RangeModel 2. Fig 11 and 12 show the effect of varying two thresholds (DistTH and LocAbilityTH) in our localization aware sampling strategy. The run time saving from RRBT-LAS to RRBT-LASC is shown in Fig 18 (a), (b).

8.5 Simulation results with replicated environment

To demonstrate our sampling and connection strategies in a different scenario, we replicated a simple environment used in Bry and Roy (2011) as shown in Fig 19. In the example, the goal is protected by two obstacles, with a narrow gap in between. The initial uncertainty in robot's pose is such that a direct path to the goal may result in a collision. It is therefore necessary for

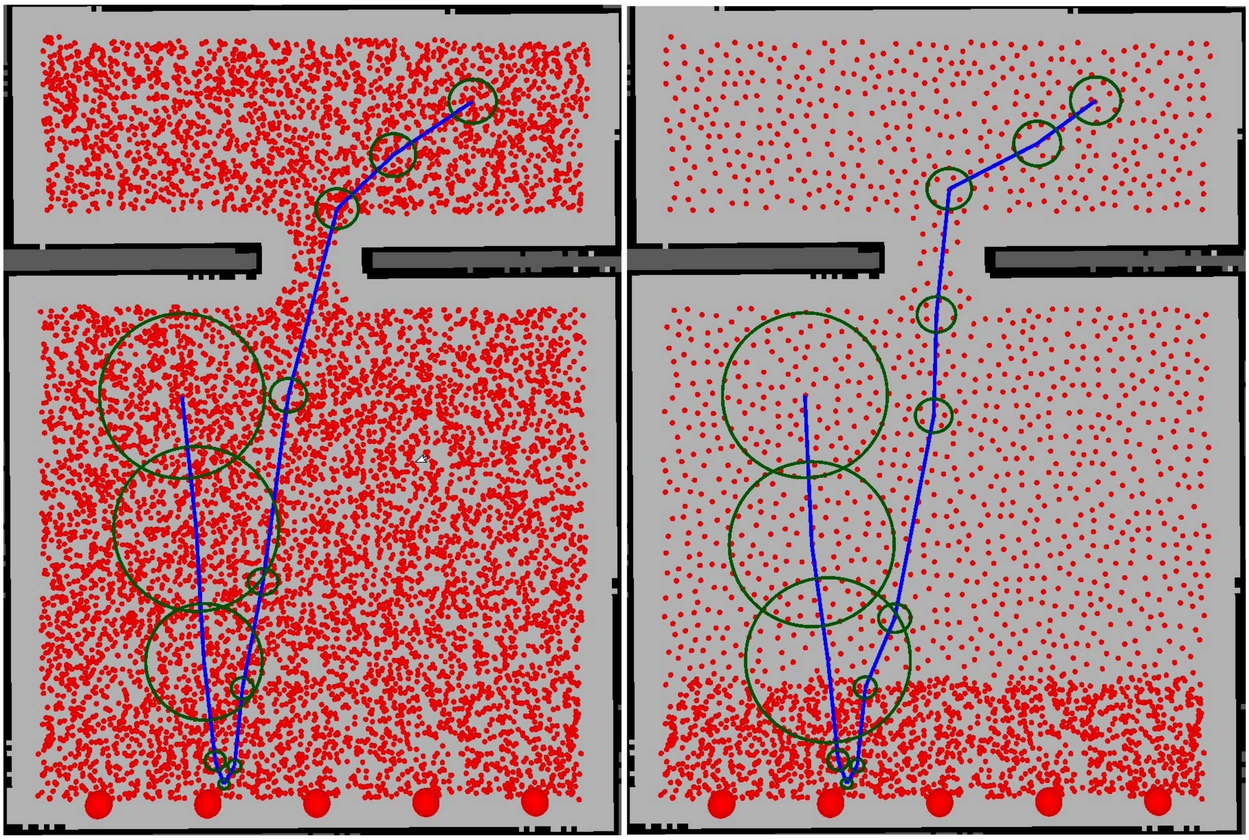

Fig. 19 These figures show the path planned by RRBT-TF (left) and RRBT-LAS (right) with number of input samples from a seed as 10000. For RRBT-LAS, we used DistTH as 10 cm and LocAbilityTH as 76.6%. Note that sensor measurements are available only within a beacon's range (big red color balls, 5 of them), which is 2 meters in this example.

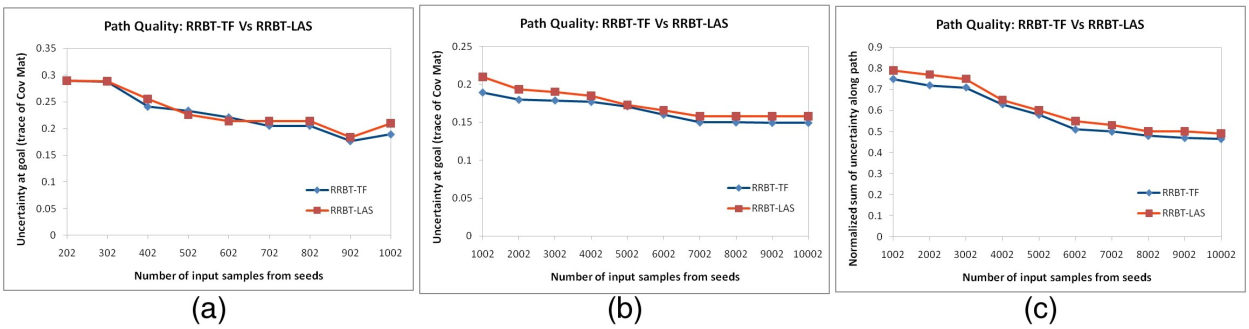

Fig. 20 Comparison of path quality between RRBT-TF and RRBT-LAS for a scenario shown in Fig 19. Y-axis denotes the trace of covariance matrix at goal in plots (a, b) and normalized sum of trace of covariance matrices along path in plot (c).

the robot to drive into the well-localized region near beacons to gain sensor measurements. This will reduce the uncertainty in its own position which in turn will decrease the chance of collision while passing through the narrow gap. We provide our simulation results for this scenario in Fig 19 to Fig 21.

Fig 19 shows one of the trials of RRBT-TF and RRBT-LAS after 10000 number of input samples from a seed. The actual number of samples in case of RRBT-TF remains same (10000) while in case of RRBT-LAS, it was reduced to 1893. It is important to note that the nature of the path remains same where sensor data is able to achieve higher uncertainty reduction (near to beacons). Plots in Fig 20 show the difference in quality of paths generated by RRBT-TF and RRBT-LAS. The quality slightly degrades in case of RRBT-LAS mainly because most of the time the path passes through regions where there are no sensor measurements. Therefore, the sampling strategy used in RRBT-LAS limits the number of samples to only one with in DistTH as in no measurement zone the localization ability of two samples remains same. In such scenarios, if we de-

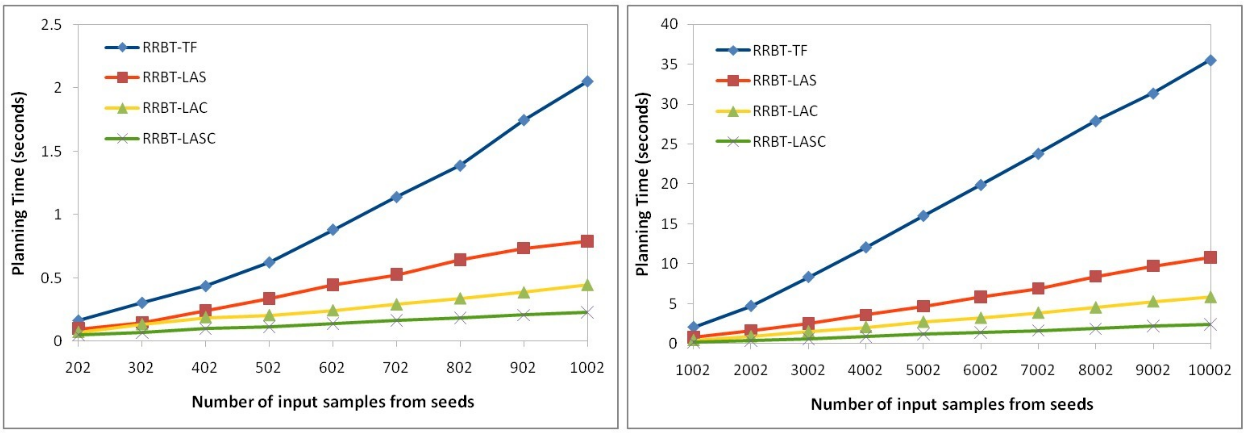

Fig. 21 Comparison of planning time for RRBT-TF Vs RRBT-LAS Vs RRBT-LAC Vs RRBT-LASC for a scenario shown in Fig 19.

crease DistTH then the quality will improve but at the cost of computational time. The comparison of planning time is shown in Fig 21. We observe that the run time savings increase supra linearly (from RRBT-LAS to RRBT-LASC) with the number of input samples. We also investigated how effective the computed paths are in getting the robot to reach the goal region reliably. For that, we executed each computed path ten times by varying the motion and sensor noises and found that the paths generated by planners that use our smart strategies are as good as that of RRBT-TF, i.e., in all the execution trials the robot successfully reached the goal as shown in Fig 22.

9 Conclusion and future work

We presented efficient localization aware sampling and connection strategies for incremental sampling-based stochastic motion planners to reduce the planning time. Our novel sampling strategy judiciously places the samples using a new notion of localization ability of a sample, i.e., it puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining adequate samples in regions where uncertainty reduction is poor. This leads to a less dense roadmap and hence results in significant time savings. An important aspect of our work is a new measure, the “Localization Ability of a sample” that captures the ability of sensor data in reducing the uncertainty at the sample point without actually knowing the path leading to it. We show that a stochastic planner that uses our sampling strategy is probabilistically complete for DistTH less than or equal to half of the inscribed radius of the robot. Robots of larger size will benefit more from the sampling scheme while maintaining probabilistic completeness.

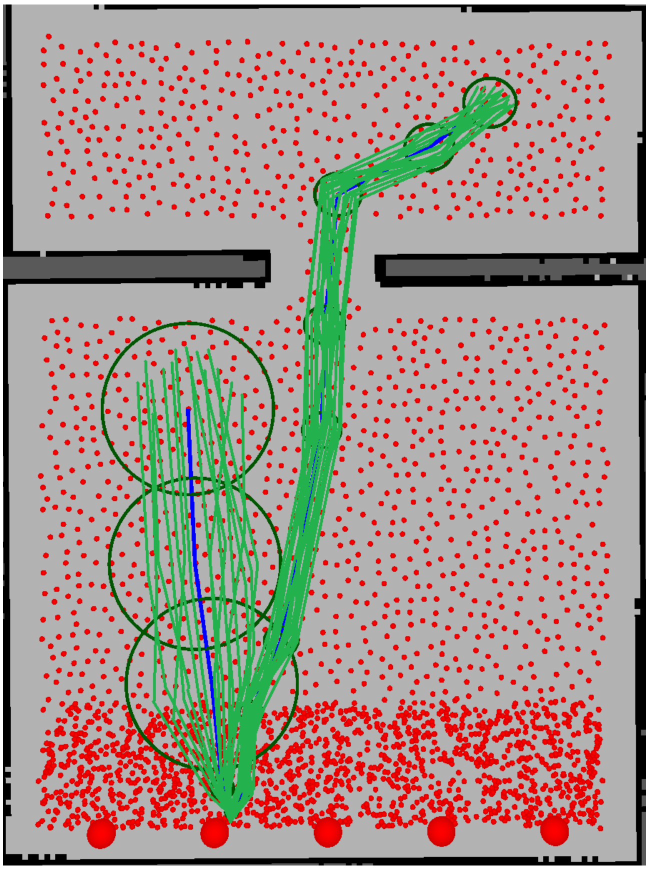

Fig. 22 Light green color shows the trails of path execution. In this particular example, we executed the planned path 20 times.

We then presented a localization aware connection strategy for sampling-based incremental stochastic motion planners that uses an uncertainty aware approach in connecting the newly sampled point to the neighbouring nodes. At each iteration of incrementally adding a new sample, our connection strategy does not consider those edges which result into increase of uncertainty along those partial paths and also reduces the number of search queue iterations (which are required to update the paths) by inserting less number of neighbouring nodes into it. As the roadmap will grow by adding more samples, the number of edges that our connection strategy judiciously removes will also increase and the number of search queue iterations will decrease. The combined effect of these two factors reduces the planner run time significantly.

We implemented incremental stochastic motion planners (RRBT-TF, RRBT-LAS, RRBT-LAC, RRBT-LASC) and demonstrated our localization aware sampling and connection strategies. We empirically show that a) our localization aware sampling strategy places less samples and find a well-localized path in shorter time with little compromise on the quality of path as compared to existing sampling techniques, b) our localization aware connection strategy finds a well-localized path in shorter time with no compromise on the quality of path as compared to existing connection techniques, and finally c) combined use of our sampling and connection strategies further reduces the planner run time. Additionally, we also implemented non-incremental (BRM-TF and BRM-LAS) stochastic motion planners and observed that our localization aware sampling strategy reduces the planner run time for non-incremental planners as well (although the saving is not as large as for incremental planners).

Finally, we showed that our localization aware connection strategy (for incremental roadmap) can also be used for tree-based stochastic planners. For that we extended RRT* to handle stochasticity associated with a robot's motion and its sensory readings.

We believe that L_n can be extended to the multimodal distribution using Monte Carlo localization (MCL) (Fox et al, 1999) that uses a particle filter to represent the distribution of likely states, with each particle representing a possible robot state. The MCL algorithm works in two stages. First, it uses the motion model to shift the particles to predict its new state after the motion and the likelihood (weight) of each new particle is computed using sensor measurements. In the second stage, the particles are resampled based on how well the actual sensed data correlate with the predicted state.

To extend our L_n measure to multimodal distribution, we bypass the prediction of new particles based on the robot motion as we do not know the control commands at the sampling stage. The procedure to compute the localization ability of a sample is then as follows. We could assume a fixed distribution of particles around a sampled point with each particle assigned the same weight. This set of particles essentially serves the same role as M for the Gaussian case. The same distribution is used for all samples by appropriately transforming corresponding to the co-ordinates of the sample points. Sensor measurement step is then used to assign new weights to each particle followed by a resampling step as in standard MCL. This new set of particles essentially serves the same role as Σ_n for the Gaussian case. Kullback-Leibler divergence (Wikipedia, 2015) that measures the information gain between two probability distributions can then be used as localization ability of a sample.

References

Agha-mohammadi, A., Chakravorty, S., Amato, N. (2014). FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research*, *33*(2), 268–304.

Bai, H., Hsu, D., Lee, W. S. (2014). Integrated perception and planning in the continuous space: A POMDP approach. *The International Journal of Robotics Research*, *33*(9), 1288–1302.

Bai, H., Cai, S., Ye, N., Hsu, D., Lee, W. S. (2015). Intention-aware online POMDP planning for autonomous driving in a crowd. In: *Proc.of the IEEE Int. Conf. on Robotics and Automation (ICRA)* (pp. 454–460).

van den Berg, J., Abbeel, P., Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, *30*(7), 895–913.

Bouilly, B., Simeon, T., Alami, R. (1995). A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty. In: *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1327–1332).

Bry, A., Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In: *Proc.of the IEEE Int. Conf. on Robotics and Automation (ICRA)* (pp. 723–730).

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., Thrun, S.*Principles of Robot Motion: Theory, Algorithms, and Implementations.* title, MIT Press, Cambridge, MA.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., Thrun, S.*Principles of Robot Motion: Theory, Algorithms,* *and Implementations.* title, MIT Press, Cambridge, MA, chap 7, pp. 242–246.

Fox, D., Burgard, W., Dellaert, F., Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. In: *Proc.of the National Conference on Artificial Intelligence* (pp. 343–349).

Fraichard, T., Mermond, R. (1998). Path planning with uncertainty for car-like robots. In: *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 27–32).

Hsu, D., Latombe, J. C., Kurniawati, H. (2006). On the probabilistic foundations of probabilistic roadmap planning. *International Journal of Robotics Research (IJRR)*, *25*(7), 627–643.

Huang, Y., Gupta, K. (2008). RRT-SLAM for motion planning with motion and map uncertainty for robot exploration. In: *Proc.of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (pp. 22–26).

Huang, Y., Gupta, K. (2009). Collision-probability constrained PRM for a manipulator with base pose uncertainty. In: *Proc.of the IEEE International Conference on Intelligent Robots and Systems (IROS)* (pp. 1426–1432).

Kaelbling, L., Littman, M., Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

Karaman, S., Frazzoli, E. (2010). Incremental sampling-based optimal motion planning. (In: *Proc.of the Robotics: Science and Systems (RSS)*).

Karaman, S., Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research (IJRR)*, *30*(7), 846–894.

Karaman, S., Walter, M., Perez, A., Frazzoli, E., Teller, S. (2011). Anytime motion planning using the RRT*. (In: *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*).

Knepper, R. A., Mason, M. T. (2012). Real-time informed path sampling for motion planning search. *International Journal of Robotics Research (IJRR)*, *31*(11), 1231–1250.

Kurniawati, H., Du, Y., Hsu, D., Lee, W. S. (2009). Motion planning under uncertainty for robotic tasks with long time horizons. (In: *Proc. of the International Symposium on Robotics Research*).

Kurniawati, H., Bandyopadhyay, T., Patrikalakis, N. (2012). Global motion planning under uncertain motion, sensing, and environment map. *Autonomous Robots*, *33*(3), 255–272.

Lambert, A., Gruyer, D. (2003). Safe path planning in an uncertain-configuration space. In: *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4185–4190), Roma, Italy.

LaValle, S. M.*Planning Algorithms.* title, Cambridge University Press, Cambridge, UK.

Lazanas, A., Latombe, J. C. (1995). Motion planning with uncertainty: a landmark approach. *Artificial Intelligence*, *76*(1-2).

Melchior, N. A., Simmons, R. (2007). Particle RRT for path planning with uncertainty. In: *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1617–1624), Roma, Italy.

Missiuro, P. E., Roy, N. (2006). Adapting probabilistic roadmaps to handle uncertain maps. In: *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1261–1267).

Pineau, J., Gordon, G., Thrun, S. (2003). Point-based value iteration: An anytime algorithm for POMDPs. In: *International Joint Conferences on Artificial Intelligence* (pp. 1025–1032).

Prentice, S., Roy, N. (2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, *28*(11-12), 1448–1465.

Ribeiro, M. I. (2004). Gaussian probability density functions: Properties and error characterization. Technical report, Institute for Systems and Robotics.

Stachniss, C., Grisetti, G., Burgard, W. (2005). Information gain-based exploration using rao-blackwellized particle filters. In: *Proc.of the Robotics: Science and Systems (RSS)* (pp. 65–72).

Wikipedia (2015). Kullback-Leibler divergence. `https://en.wikipedia.org/wiki/Kullback-Leibler_divergence`, [Online; accessed September 25, 2015].

A Appendix: Probabilistic completeness proof for RRBT-LAS

In this section, we provide a formal proof that a planner with our localization aware sampling strategy is probabilistically complete for DistTH less than or equal to half of the inscribed radius of the robot.

The worst case situation that leads to probabilistic completeness issues with our approach is using RangeModel 1 where the sensors (beacons in our case) have limited range. In that case the heuristic used in our sampling strategy will limit the samples to only one (within ball of radius DistTH) for regions with low uncertainty reduction. This is where the completeness issue arises. If the value of DistTH is large then the planner that uses our sampling strategy may not be able to find a path. We show that if we keep DistTH below half of the inscribed radius of the robot (a reasonable assumption) then if there exists a collision-free path, a planner that uses our sampling strategy will also find one. For the proof we assume that the entire path passes through regions with low uncertainty reduction (a worst case scenario for our sampling strategy). Also note that our proof builds along the lines of Choset et al (2005b), therefore, we follow most of their notations.

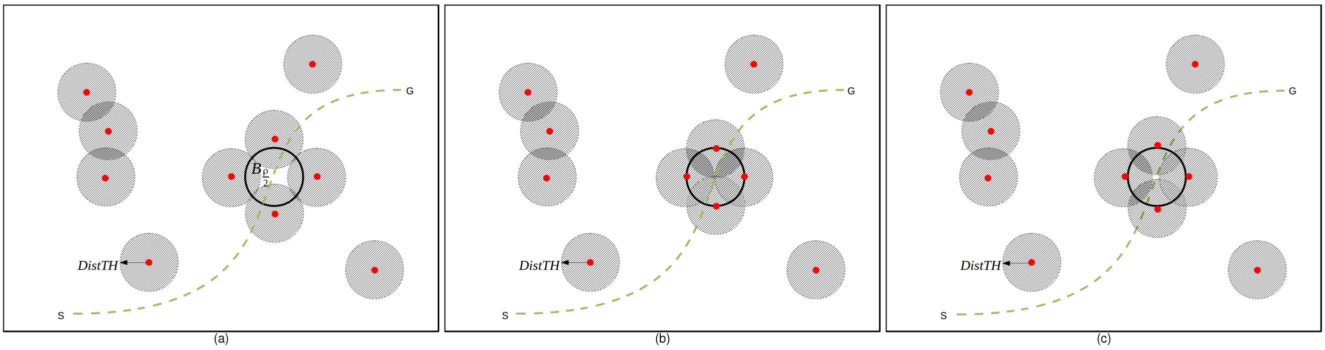

Fig. 23 [Case $DistTH \leq \frac{\rho}{2}$] - black colour (bold) circle denotes one of the balls $B_{\frac{\rho}{2}}(q_i)$ that is used to tile a path, red colour dots represent randomly placed samples, and hatched region (of radius DistTH=$\frac{\rho}{2}$) around each sample denotes the restricted region where samples can not be placed according to heuristic used in localization aware sampling. This figure shows the situation (excluding b) where none of the samples has yet been placed inside the black ball. (a) neighbouring samples around $B_{\frac{\rho}{2}}(\cdot)$ restrict some region (hatched areas inside the black ball) inside the black ball where samples can not be placed, (b) neighbouring samples totally covered $B_{\frac{\rho}{2}}(\cdot)$ but in that case samples lie on the periphery (closed set), (c) samples are placed at a distance d such that $\frac{\rho}{2} < d < \frac{\rho}{2} + \epsilon$, even in this worst case scenario the probability of generating a sample in $B_{\frac{\rho}{2}}(\cdot)$ is greater than 0 (see text for explanation).

Suppose q_s, $q_g \in C_{free}$ (free region of C-space) are two robot configurations that can be connected by a path in C_{free}. RRBT-LAS is considered to be probabilistically complete, if for any given (q_s, q_g)

$$\lim_{n \to \infty} Pr[(q_s, q_g)FAILURE] = 0 \tag{10}$$

where $Pr[(q_s, q_g)FAILURE]$ denotes the probability that RRBT-LAS fails to answer the query (q_s, q_g) after a roadmap in C_{free} with n samples has been constructed. The outline of the probabilistic completeness proof is as follows: First we assume that a path π from q_s to q_g exists. We then tile the path with a set of carefully chosen balls such that generating a sample in each ball ensures that these samples can be connected with appropriate collision-free edges and hence a collision-free path, $\hat{\pi}$ between q_s and q_g will be found by RRBT-LAS and the probability of generating such samples approaches 1 with increasing n.

Assume a path π (of length L) from q_s to q_g exists in d dimensional C-space. The clearance of π, denoted $\rho = clr(\pi)$, is the farthest distance away from the path at which a given point can be guaranteed to be collision-free. Note that $\rho \geq 2r$, where r is the inscribed radius of the robot. The measure μ denotes the volume of a region of space, e.g, $\mu(B_\epsilon(x))$ measures the volume of an open ball $B_\epsilon(x)$ of radius ϵ centered at x. If $A \subset C_{free}$ is a measurable subset and x is a random point chosen from C_{free}, then

$$Pr(x \in A) = \frac{\mu(A)}{\mu(C_{free})} \tag{11}$$

We now tile the path π with balls each of radius $\frac{\rho}{2}$. Let $m = \lceil \frac{2L}{\rho} \rceil$ and observe that there are m points (centers of balls) on the path such that $dist(q_i, q_{i+1}) < \frac{\rho}{2}$, where $dist$ is a Euclidean metric on $\mathbb{R}^d$. Let $y_i \in B_{\rho/2}(q_i)$ and $y_{i+1} \in B_{\rho/2}(q_{i+1})$. Then the line segment $\overline{y_i y_{i+1}}$ must lie inside C_{free} since both endpoints lie in the ball $B_\rho(q_i)$. An illustration of this basic fact is given in Figure 7.17 of Choset et al (2005b). Let $V \subset C_{free}$ be a set of n configurations generated by our localization aware sampling strategy. If there is a subset of configurations $\{y_1, ..., y_m\} \subset V$ such that $y_i \in B_{\rho/2}(q_i)$, then each ball will get a sample and a path from q_s to q_g will be found. Let $I_1, ..., I_m$ be a set of indicator variables such that each I_i witnesses the event that there is a $y \in V$ and $y \in B_{\rho/2}(q_i)$. It follows that RRBT-LAS succeeds in answering the query (q_s, q_g) if $I_i = 1$ for all $1 \leq i \leq m$. If at least one of the indicator variables is 0 then RRBT-LAS would fail. Therefore, the probability of failure (Equation 10) then can be written as

$$Pr[(q_s, q_g)FAILURE] \leq Pr\left(\bigvee_{i=1}^{m} I_i = 0\right) \tag{12}$$

$$\leq \sum_{i=1}^{m} Pr[I_i = 0] \tag{13}$$

where the last inequality follows from the union bound. We now mainly focus on the computation of $Pr[I_i = 0]$ for i^{th} ball, i.e., after placing n samples by our localization aware sampling strategy what is the probability that none of these samples lie in a ball $B_{\rho/2}(q_i)$.

For RangeModel 1, in regions outside the sensor range where there is no sensor information, hence no uncertainty reduction, our localization aware sampling strategy does not allow another sample within the vicinity (DistTH) of an already placed sample point (see Fig 23). Therefore, the probability of failure to generate a second sample in a ball $B_{\rho/2}(q_i)$ depends on where the first sample was placed and so on. Let $I_i^1, ..., I_i^n$ be a set of indicator variables for the i^{th} ball such that each I_i^k, for all $1 \leq k \leq n$, witnesses the event that the k^{th} sample does not lie in ball $B_{\rho/2}(q_i)$. These events are dependent on each other. Below we provide the expressions to compute $Pr[I_i^k = 0]$ that will lead us to the computation of $Pr[I_i = 0]$. For the first two samples, the probability of failure to generate a sample inside ball $B_{\rho/2}(q_i)$ can be written as

$$Pr[I_i^1 = 0] = \left\{1 - \frac{\mu(B_{\rho/2}(q_i))}{\mu(C_{free})}\right\} \tag{14}$$

$$Pr[I_i^2 = 0] = \int Pr(I_i^2 = 0 \mid x^1) Pr(x^1) dx^1 \tag{15}$$

In above expression x^1 denotes the position of first sample. Above expression is just the marginalization over the position

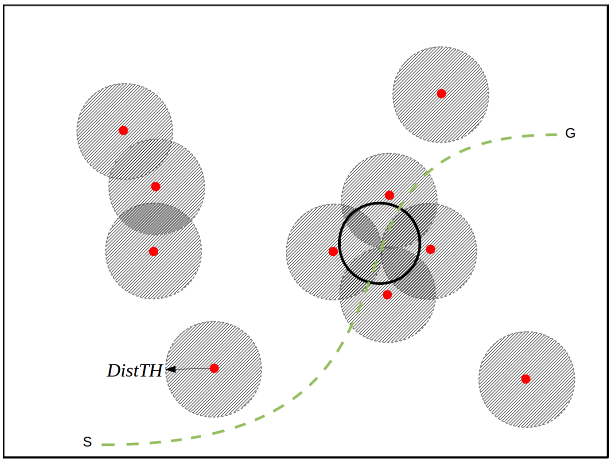

Fig. 24 [Case $DistTH > \frac{\rho}{2}$] - This figure shows that if $DistTH > \frac{\rho}{2}$ then the restricted regions of neighbouring samples may completely block the ball $B_{\rho/2}(q_i)$ and will prevent generation of a sample inside it. This will lead to the failure of a planner that uses our localization aware sampling strategy.

of first sample. Similarly, the expression for the third sample is

$$Pr[I_i^3 = 0] = \iint Pr(I_i^3 = 0 \mid x^1, x^2) Pr(x^2 \mid x^1) \\ Pr(x^1) dx^2 dx^1 \qquad (16)$$

and for n^{th} sample the expression (for $Pr[I_i^n = 0]$) is

$$= \int \cdots \int Pr(I_i^n = 0 \mid x^1, ..., x^{n-1}), ..., Pr(x^2 \mid x^1) \\ Pr(x^1) dx^{n-1} dx^{n-2}, ..., dx^2 dx^1. \qquad (17)$$

Clearly, parameters $\frac{\rho}{2}$ (radius of ball B) and DistTH (restricted region around a sample) are embedded in above expressions. Using Equations 14-17, Equation 13 can now be written as

$$Pr[(q_s, q_g) FAILURE] \leq \left\lceil \frac{2L}{\rho} \right\rceil \left(\prod_{k=1}^{n} Pr[I_i^k = 0] \right) \qquad (18)$$

Note that $Pr(I_i^n = 1 \mid x^1, ..., x^{n-1})$ denotes the probability of generating the n^{th} sample inside ball $B_{\rho/2}(q_i)$ given that $n-1$ samples have been placed. This is nothing but the ratio of volume of white region inside the black ball (after excluding the hatched region) and total volume of white region (with reference to Fig 23). In general, this can be written as

$$Pr(I_i^n = 1 \mid x^1, ..., x^{n-1}) = \frac{\mu({}^B C_{free}^R)}{\mu(C_{free}^R)} \qquad (19)$$

where C_{free}^R denotes the C_{free} left after excluding the restricted regions around already placed samples and ${}^B C_{free}^R$ denotes the same but inside ball $B_{\rho/2}(q_i)$. This ratio approaches one as more and more samples are placed outside the black ball. That implies that $Pr(I_i^n = 0 \mid x^1, ..., x^{n-1})$ approaches zero. The expression in Equation 17 is one of the product terms in RHS of inequality 18. Convergence of $Pr[I_i^n = 0]$ (to 0) will lead to the convergence of RHS of inequality 18. Therefore, $Pr[(q_s, q_g) FAILURE]$ converges to 0 as the number of samples increases, hence showing the completeness of RRBT-LAS. The same completeness can not be guaranteed for $DistTH > \frac{\rho}{2}$ (see Fig 24).