# Uncertainty Reduction via Heuristic Search Planning on Hybrid Metric/Topological Map


Qiwen Zhang and Gregory Dudek
*School of Computer Science*
*McGill University*
*Montreal, Canada*
*[qzhang32,dudek]@cim.mcgill.ca*

Ioannis Rekleitis
*Computer Science and Engineering Department*
*University of South Carolina*
*Columbia, United States*
*yiannisr@cse.sc.edu*



***Abstract*—This paper presents an extension of our previous work on hybrid metric/topological maps to enable uncertainty reduction planning through the map, taking into account both map uncertainty and distance. An enhancement of the edge structure which enables the simulation of bidirectional edge propagation through an extended Kalman filter is proposed in our heuristic search planning algorithm to plan for maximal map uncertainty reduction. This work expands on the heuristic search framework proposed in [1] to apply in hybrid metric/topological maps instead of more constrained camera sensor networks. Experimental results from realistic simulations and deployment on a real robotic system are presented to show the efficacy of the proposed algorithm and validate our approach for uncertainty reduction.**



***Keywords*-Generalized Voronoi Graph; SLAM; Uncertainty Reduction; Heuristic search; Planning**


Figure 1. Our main robotic platform, a TurtleBot 2, is equipped with a Hokuyo laser range-finder used for navigation. Several experiments performed with this configuration validate the Ear-based exploration, the construction of the hybrid Metric/Topological map, and the heuristic path planning.

## I. Introduction

This paper presents a heuristic search planning framework which results in reducing the uncertainty in a map created by an exploring robot. The proposed approach is applied in the context of hybrid metric/topological maps as an extension to our previous work in ear-based exploration, but the methods proposed can be applied to any generic mapping technique that maintains a metric state representation of the environment. This work is a continuation of the Simultaneous Localization and Uncertainty Reduction (SLURM) framework, proposed in the context of exploring a camera sensor network [2], utilizing a single [3] or multiple robots [4]. In [5] a hybrid metric/topological map was proposed to facilitate the generation of an efficient and accurate environment representation. Further improvements in the accuracy of the resulting map are feasible with the utilization of the proposed heuristic-search path planning algorithm. Under the assumptions of a static GPS-denied world the exploration strategy is solely based on laser-scan data and odometry information.

The topological representation of the environment has the advantage that it gives us a simple way to encode all the distinct locations in the environment. Since the Generalized Voronoi Graph [6] (GVG) is used as the graph-based representation, the distinct locations in the environment correspond to the points that are equidistant to more than two obstacles. Naturally, the edges of the graph represent the connectivity of the distinct locations, commonly termed meetpoints. During the exploration process, the robot processes all the information available and maintains an extended Kalman Filter (EKF) to keep track of the state of the robot and the pose of the visited meetpoints. As the robot navigates further and further through unknown territory, the uncertainty of the robot and the map will inevitably grow. If the uncertainty grows too large, then further exploration is futile; any newly discovered nodes will have its location easily confused with other landmarks regarding the map. Thus, in order to reduce map uncertainty, the robot must return to previously visited nodes, also known as re-localization. One such re-localization strategy is to always return to the node with the current least uncertainty in the map via the shortest path. This strategy can be appealing due to its simplicity, but this paper demonstrates that such a strategy is not optimal in reducing map uncertainty.

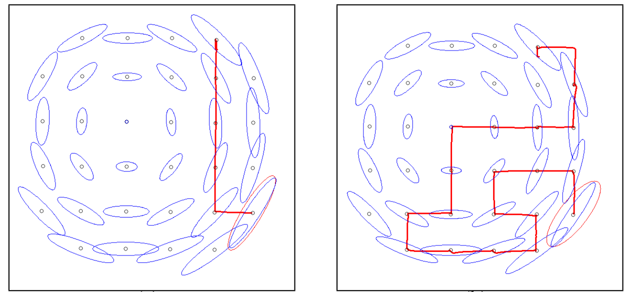
In the proposed framework, a heuristic search planning method seeks to maximize the uncertainty reduction in the map, with respect to the amount of distance that the robot has to travel. Therefore, at any point during the exploration the robot is capable of re-localizing efficiently at a minimal cost possible in terms of traveled distance and time. The heuristic search algorithm is based on the well-known A* search algorithm and takes into account the accumulated uncertainty of the map in the form of the trace of the covariance of the extended Kalman filter; a common metric [7], see Eq. (1). However, it is important to note that there is no proof of optimality in this case since the trace of the covariance measure is not a sufficient statistic. The cost function of the priority queue which describes the search order is dictated by the following variables and parameters:

$$trace(\mathbf{P}^t) = \sum_{i=n}^{n} \mathbf{P}^t_{ii} \tag{1}$$

- **Current distance traveled**: As in the formulation of classic A* algorithm, the proposed method keeps track of the distance already traveled from the start for each possible path solution, allowing it to choose paths that have minimal combined current cost and heuristic.
- **Distance to target**: In the context of re-localization, the robot has access to the explored map of the environment. In order to satisfy the condition of a consistent and admissible heuristic, this measured distance to the target will be the shortest path distance computed from available map information from the current location to the goal.
- **Uncertainty measure**: As mentioned previously, the trace of the covariance matrix was chosen as our map uncertainty measurement. Thus, the cost function takes into account the trace combined with the current distance traveled and shortest path distance to the goal to determine which search state to consider next. As a result, if the current solution path estimates a significant uncertainty reduction, then this path is more likely to be considered as the final re-localization path.
- **$\alpha$ free parameter**: Finally, there is $\alpha \in [0,1]$, a free hyper-parameter that allows the user to control how much bias to put towards considering uncertainty reduction over time efficiency. Fig. 2 demonstrates the effect of changing $\alpha$ on the resulting solution path. The two figures differ in path length due to the choice of $\alpha$, a parameter set by the user. All of these parameters will be covered in greater details in Section V.

## II. Related Work

Topological maps were first described in the seminal paper by Kuipers and Byun [8]. Further results with topological maps were presented in [9] and [10]. Different groups have previously proposed combining hybrid topological maps



(a) (b)

Figure 2. Effect of $\alpha$ on the resulting solution path starting from the same source node to target node. (a) $\alpha = 1.0$, which is equivalent to Dijkstra's shortest path (b) $\alpha = 0.0$, heuristic search considering only map uncertainty reduction.

and metric representations to achieve loop-closure, such as Werner et al. [11] using a particle filtering approach in a GVG representation to determine the location of the robot using a sequence of signatures of the GVG meetpoints recently visited. Work by Choset and Nagatani [12] is closest in ideology to our previous contributions in ear-based exploration as they rely on metric information at the meetpoints overlayed on top of a GVG representation. Multiple heuristics are considered to eliminate candidate GVG meetpoints when the robot performs loop-closure.

Estimating the state of a map and its entropy is commonly done using an extended Kalman filter, which is described in [13] for SLAM. The EKF keeps track of a mean and covariance for each map landmark. Many researchers, [14] and [15], have proposed attempts to reduce the uncertainty in the map estimates. Another group proposed in their work [16] a simulation-based approach, in which multi-step paths would be generated to determine the most uncertainty reducing solution, however it comes at the expense of high computational costs. Their work is most in common with our proposed framework.

By contrast, our approach takes the exploration strategy into account, hence the application of our heuristic search based on A* can be applied to path plan to frontier nodes during exploration but also when re-localizing to reduce map uncertainty estimates. The algorithm depends on the map uncertainty measure; the trace of the covariance is one of the proposed measures in [7]. The cost function, which combines both distance measures and the map uncertainty measure, is based on a weighted linear combination proposed by Makarenko et al. [15].

## III. GVG Exploration

The underlying exploration strategy, which is the focus of our previous contributions, uses the Generalized Voronoi Graph (GVG) as its navigation map. In a two dimensional environment, the GVG can be computed online efficiently using only range information from a laser range finder mounted on the robot. Any location in space which is equidistant to three or more obstacles is determined to be a meetpoint (graph node). To avoid any mislabeled locations as nodes, the range data is filtered to remove any spurious edges. An important aspect of the GVG is its reactive nature, hence once the edges between the meetpoints are discovered, it becomes trivial to return to previously explored nodes by following the relative orientations between the bearings of the meetpoint. The overall procedure of the navigation system is outlined as follows:

- **Finding Meetpoints/Endpoints**: The robot constantly checks the filtered laser data during navigation to detect meetpoints or endpoints. From there, the robot decides towards which direction to explore next. If all the directions have been previously explored, then the robot selects a meetpoint (node) with unexplored edges to proceed with exploration. Nodes with unexplored edges are termed frontier nodes. Note that in the proposed exploration strategy, the robot systematically chooses the next counter-clockwise direction to explore next. This strategy is termed ear-based exploration, and results in systematic frequent loop-closures; for more information please refer to [5].
- **Edge Following**: When the robot is not at a meetpoint, it has to be along an edge between two nodes in the graph. Since an edge corresponds to only two distinct obstacles in the environment, then the robot can compute and follow the center-line. The robot's steering control can efficiently be dealt with using PD control. After traveling along an edge, edge information, such as its length and points along the line are stored for further use in the heuristic search.

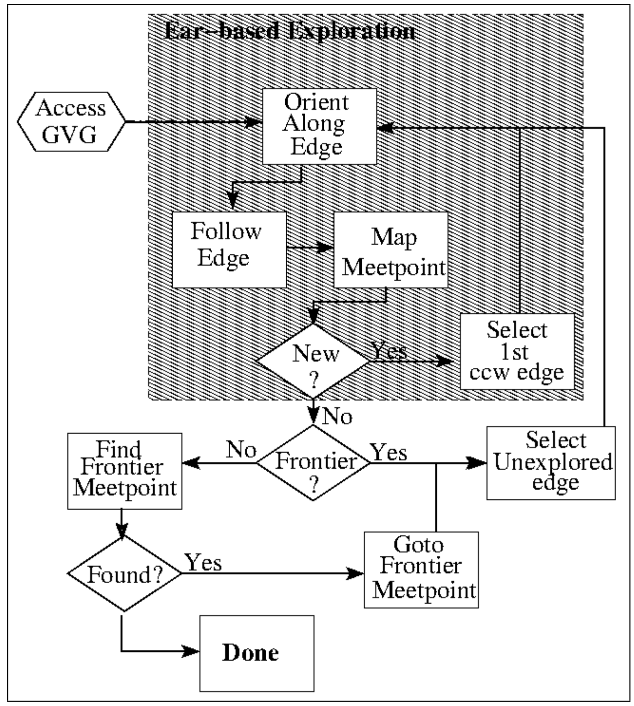



Figure 3. Ear-based exploration using the GVG representation.

## IV. Localization

During the exploration of the unknown environment, an extended Kalman filter is used to estimate the state and propagation uncertainty of the robot. The accumulated uncertainty is encoded in the covariance matrix **P**. The structure of the matrix is in 2:

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{P}_t^r & \mathbf{P}_t^{rl} \\ \mathbf{P}_t^{lr} & \mathbf{P}_t^{map} \end{bmatrix} \tag{2}$$

where $\mathbf{P}^r$ encodes the uncertainty of the robot's pose; $\mathbf{P}^{map}$ encodes the uncertainty of the map; and $\mathbf{P}^{rl}$,$\mathbf{P}^{lr}$ records the cross-correlation between the robot and the map. The EKF only uses odometry information to track the motion of the robot through the corridors, by applying the following robot propagation equations Eq. (3):

$$\begin{aligned}
\mathbf{X}_{t+1} &= \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_{t+1} dt \cos\theta_t \\ y_t + v_{t+1} dt \sin\theta_t \\ \theta_t + \omega dt \end{bmatrix} \\
\mathbf{P}_{t+1}^r &= \Phi_{t+1} \mathbf{P}_t^r \Phi_{t+1}^T + G_{t+1} Q G_{t+1}^T \\
&\text{where} \\
\Phi_{t+1} &= \begin{bmatrix} 1 & 0 & -v_{t+1} dt \sin\theta_t \\ 0 & 1 & v_{t+1} dt \cos\theta_t \\ 0 & 0 & 1 \end{bmatrix} \\
G_{t+1} &= \begin{bmatrix} -dt\cos\theta_t & 0 \\ -dt\sin\theta_t & 0 \\ 0 & -dt \end{bmatrix} \\
Q &= \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix}
\end{aligned} \tag{3}$$

During propagation, the estimated pose and uncertainty of the mapped meetpoints does not change. As the robot encounters meetpoint $i$ for the first time, the robot adds the pose of the meetpoint into the EKF and keeps track of the state $X_i = [x_i, y_i, \theta_i]$. Whenever meetpoint $i$ is re-visited, an update is performed using the following update equations:

$$
\begin{aligned}
S &= H\mathbf{P}H^T + R \\
H &= [-C^T(\theta) - C^T(\theta)J(\mathbf{X}_i^{mtp} - \mathbf{X}_{t+1})] \\
C(\theta) &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \\
J &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\
K &= \mathbf{P}H^T S^{-1} \\
\mathbf{X} &= \mathbf{X} + Kr \\
\mathbf{P} &= \mathbf{P} - KSK^T
\end{aligned} \tag{4}
$$

where $r$ is the difference between the recorded value of the meetpoint and the latest estimate and $R$ is the noise covariance of the laser sensor.

## V. Heuristic Search Planning

During the exploration of the environment, the robot needs to path plan to return to frontier nodes to continue exploring unknown territory. When path planning in a GVG/metric map to go from one node to another, the proposed heuristic method can be utilized in order to improve the quality of the map. The heuristic search method does so by computing a cost function dependent on map uncertainty and traveled distance. Thus, the robot can perform this search for all frontier nodes and selecting to return to the node which offers the best improvement. Naturally, the same path planning method can also be applied to when the robot requires re-localization to reduce map uncertainty. In that case, searching is not restricted to only frontier nodes but allowed to return to any node in the current known map.

It is important to note at this point that re-localization physically moves the robot to reduce the map uncertainty. In contrast, it is possible to anticipate how the state and covariance estimates in our EKF will update to re-visiting previously explored nodes, without physically moving the robot. This work strides to achieve that with this heuristic search planning algorithm by picking the path which will maximally reduce the map uncertainty through re-localization.

### A. Search Cost Function

The proposed heuristic search algorithm is based on the A* search algorithm; given a source node $s$ and a target node $t$, the algorithm maintains a closed and an open set to keep track of nodes visited. As nodes are dequeued from the open set, those nodes are added to the closed set to avoid duplicate nodes in the resulting path. The neighbours of the currently processed node are added to the open set (if they do not belong to the closed or open set). In the case where the neighbour is already in the open set, then the partial path is preserved to that node which results in the least cost of the two solutions. The algorithm is guaranteed to terminate and, given an admissible and consistent heuristic, also guaranteed to be optimal.

In the discussed framework, the uncertainty measure used is the trace of the covariance matrix $\mathbf{P}^{map}$; see Eq. (2), recording only the uncertainty of the mapped meetpoints. Taking into account the covariance of the robot would result in the introduction of a location bias since the covariance of the robot depends on its current pose. This component contributes towards the current cost $g$ of the solution path. The other component of the current cost $g$ is the distance traveled. Finally, the chosen heuristic cost $h$ is the straight-line distance between the current node and the target node. Given that edge information is stored for the known map, the shortest path distance can be computed using Dijkstra's algorithm and still satisfy the requirements for heuristic admissibility.

The cost of a solution path up until node $i$, $c(i)$ is then

$$
c(i) = g(i) + h(i) \tag{5}
$$

where $g(i)$ is the combined cost from $s$ to $i$ and $h(i)$ is the distance-only heuristic cost from $i$ to $t$. We also have

$$
g(i) = d(s,i) + tr(\mathbf{P}_i^{map}) \tag{6}
$$

with $tr(\mathbf{P}_i^{map}) = tr(\mathbf{P}_i) - tr(\mathbf{P}_i^r)$, the trace of covariance matrix after traveling to node $i$, and $d(s,i)$ is the traveled distance from $s$ to $i$. The issue with the current formulation of Eq. (6) is that distance and uncertainty measures have no common grounds of scale. As proposed in [15], the distance components are scaled by the shortest path distance $1/d^*(s,t)$ and the uncertainty component by $1/tr(\mathbf{P}_s^{map})$, the trace of covariance matrix at the start location. Finally, a free hyper-parameter $\alpha$ is included, which decides whether to prioritize uncertainty reduction or minimizing traveled distance. The updated cost function then becomes:

$$
c(i) = \alpha \frac{d(s,i)}{d^*(s,t)} + (1-\alpha)\frac{tr(\mathbf{P}_i^{map})}{tr(\mathbf{P}_s^{map})} + \alpha \frac{h(i)}{d^*(s,t)} \tag{7}
$$

with $\alpha \in [0,1]$. Setting $\alpha = 1$ would then give you Dijkstra's shortest path, whereas the other extreme, $\alpha = 0$, will completely disregard distance to maximize uncertainty reduction. The normalization allows for $\alpha$ to vary smoothly, and the user has complete control over the search bias through the choice of the free parameter $\alpha$.

### B. Solution Path Simulation

The previous subsection described how the search planning algorithm creates parallel solution paths and decides which direction to explore next based on the proposed cost function. Each solution path maintains its own state and covariance in order to simulate the motion of the robot to obtain accurate estimates of the trace of the covariance. Those solution paths are created by adding the neighbours of the node $i$ currently under consideration into the open set. In other words, the solution path from $s$ to $i$ is already determined and the neighbours of $i$ are all potential candidates in the solution path from $s$ to $t$. Those new paths prolong the current path from $s$ to $i$, hence the state and covariance $\mathbf{X}$ and $\mathbf{P}$ are copied from the path from $s$ to $i$. During the simulation and propagation of the motion of the robot through the environment, the only modifications to the local state and covariance information are for that specific solution path; it does not affect at all the information stored in the EKF that keeps track of the state of the physical robot, since the robot is not moving during the heuristic search simulation.

To simulate the motion of the robot from one node to another, the edge information stored during exploration is used. One issue that arises from using the stored edge information is that the robot has only traveled most of the edges in a single direction. The proposed approach reverses the information of the edge completely, including the orientation of the robot at every point along the line segment of the edge. Using those bidirectional edges, the path between any pair of nodes can be simulated by repeatedly propagating the points along the line through the EKF and calling an update operation on the EKF for all edges along that path, as outlined in IV. Through this process, the uncertainty estimate gets updated through re-localization simulation, leading to find a solution path to the target that maximizes uncertainty reduction. Running this search algorithm to all other nodes in the environment means that the best target node to physically re-localize the robot to can be found.

## VI. Experimental Results

All of our software framework is implemented in the Robot Operating System (ROS) environment [1]. Furthermore, the experiments are run using Stage [2], a simulation engine that takes into account odometry drift and wheel slip to create more realistic robotic simulations. In addition, the heuristic search planning algorithm was deployed on a TurtleBot 2, and several experiments of exploring and mapping the corridors of our building were performed.

### A. Heuristic Search Validation

*1) Simulated Environment:* The first experimental setup tests the validity and efficiency of our heuristic search algorithm itself. In a simulated environment shown in Fig. 4a, which consists of a symmetrical grid-like world, heuristic search simulations were executed planning a path from all meetpoints in the map as the source node to all other meetpoints in the environment as the target node. It is important to note that in the interest of preserving the consistency of the robot state and covariance, the robot needs to navigate to the start node before running the heuristic search

[1]http://wiki.ros.org/

[2]http://wiki.ros.org/stage

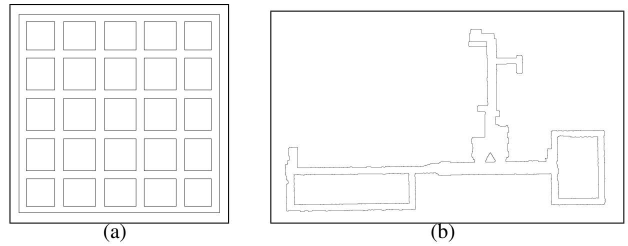


Figure 4. (a) The grid-like world environment used to compare the effect of $\alpha$ in our heuristic search method. (b) The floorplan of the Center for Intelligent Machines, McGill University, used to carry out the relocalization experiments.

simulations. In order to avoid unintentional re-localization while positioning the robot at the source node, it would be required to restart the robot at the same location it finished exploring the environment for the first time to keep each set of heuristic search simulations from different meetpoints to all be on an even playing field. This was achieved by incorporating a map saving/loading feature in our system to allow the user to repeat experiments consistently. Finally, this procedure was repeated 4 times in total, comparing the effect of $\alpha$ on the uncertainty reduction from re-localization and on the total distance traveled by the robot. Specifically, the experiments ran with $\alpha = 0.0, 0.001, 0.01$ and 1.0.

Our results in Fig. 5 show the effectiveness of our heuristic search approach. The top figures demonstrate the effect of $\alpha$ on the total distance of the solution path offered by our heuristic search algorithm, for all pairs of nodes in the environment. The bottom figures show the uncertainty reduction after performing the re-localization path proposed by the algorithm, for different values of $\alpha$.

The error bar plots (a)-(d) show the increased solution path length $\alpha$ changes. In Fig. 5, as expected, the graph shows a $y = x$ plot since our algorithm computes the shortest path for $\alpha = 1.0$. Interestingly enough, our data points for small distances $d^*(i, j)$ (meaning that the nodes in the pair $(i, j)$ are close) show larger variances than for data points between pairs $(i, j)$ at larger distances. The reason for this is that when $i$ and $j$ are nearby, the cost function will often result in selecting the direct path since traveling further away can often lead to increased robot uncertainty. When the two nodes are further apart, the heuristic search has more leeway to lengthen the solution path, if that suits it with increased uncertainty reduction.

The same analysis can be made about the plots (e)-(h) which show the uncertainty reduction with respect to the shortest path for the same values of $\alpha$. Picking the shortest path as the solution does not perform very well since the algorithm constrains the choice of paths. Hence, even if a small deviation from the shortest path could allow for a large uncertainty reduction, the algorithm doesn't allow for this deviation, leading to poor performance. Another

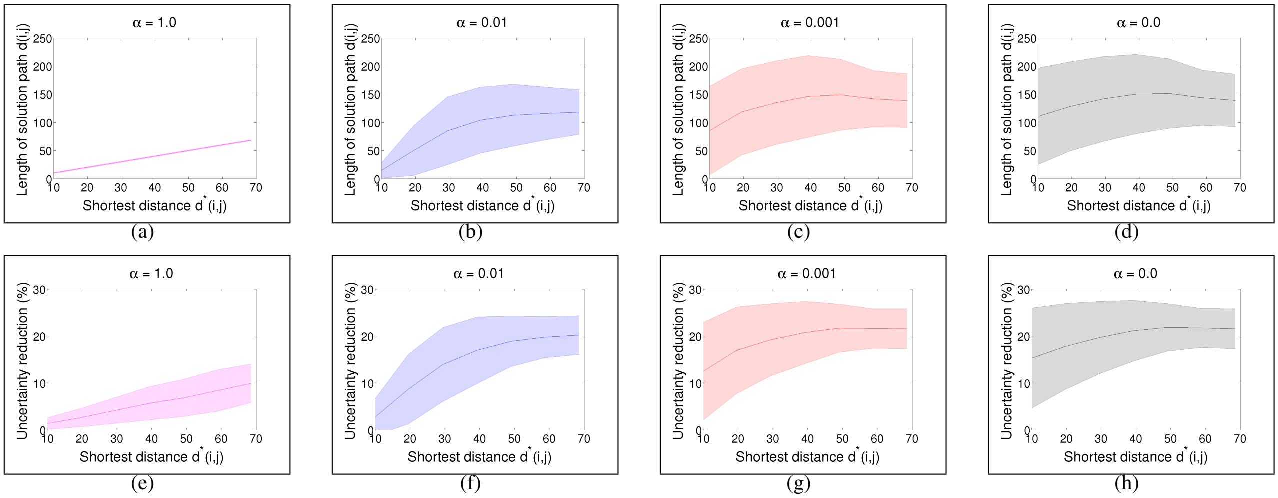



Figure 5. (a)-(d) Error bar plots showing length of the solution path $d^*(i,j)$ compared to the shortest path distance using $\alpha = 1.0, 0.01, 0.001$ and 1.0 for (a) through (d), respectively. (e)-(h) error bar plots showing total uncertainty reduction (in %) from single re-localization for all pairs $(i,j)$ again using $\alpha = 1.0, 0.01, 0.001$ and 1.0 for (e) through (h), respectively.

interesting observation is that all the other three variants seem to converge roughly to the same uncertainty reduction for larger distances. However, for $\alpha = 0.01$ at small distances, it is apparent that the strategy is not performing optimally. The logical explanation here is that at larger distances, the shortest path solution provides access to all other uncovered nodes within small deviations. In that case then, allowing the solution path to deviate slightly as in $\alpha = 0.01$ becomes equivalent to allowing large deviations such ($\alpha = 0.0$) because the algorithm cannot provide any larger deviations to obtain better results. This information is extremely resourceful for maximizing the efficiency of our proposed solution as our heuristic search planning method looks through all possible target nodes for re-localization.

*2) Real robotic environment:* We repeated the previous setup but deployed it on a real robotic system, the TurtleBot 2, instead of running in simulated environment. The experiments were conducted at the Center for Intelligent Machines at McGill University; see Fig. 4b. We started the robot from three different locations in the environment and let it explore the entire environment. After the environment was fully explored, the robot would compute the same heuristic search algorithm with every other meetpoint as the target node. The same values for $\alpha$ were chosen, in order to show the effect of varying $\alpha$ on the produced final map uncertainties.

Fig. 6 shows the results from aggregating the three trials for each value of $\alpha$. Unsurprisingly, our uncertainty reduction method is validated by the uncertainty reduction floating around 30%. However, what we notice is that the resulting uncertainty reduction percentages as well as length of solution paths are almost identical for all the values of $\alpha$. The explanation for this is quite simple; if we look at the environment, we see that there are hardly any different paths possible from any two junctions in the environment. In other words, the only variation possible is based at the triangular set of meetpoints in the middle of the environment. Therefore, using $\alpha = 0$ compared to $\alpha = 1$, the robot can only take the longer side of the triangle rather than the shortest path. This minimal difference is insignificant in terms of uncertainty reduction, which shows in the results. Nonetheless, this set of experiments still validates that our system works in real environments and demonstrates the value of relocalization in producing higher accuracy maps.

### B. Relocalization Experiments

The second set of experiments demonstrate the advantages of performing re-localizations using our heuristic search algorithm during the exploration phase of the unknown environment. These experiments were done in simulation using Stage, but this time in a more realistic environment; Fig. 4b shows the map for part of the Center for Intelligent Machines at McGill University, the chosen environment. In one set of experiments, the robot explored the entire environment without re-localizing. At the end of its exploration, the robot performed re-localization once using our heuristic search planning method. In the other set of experiments, the robot was set to have a probability $\varepsilon = 0.1$ of re-localizing whenever it reached a meetpoint. Heuristic search on the current partial exploration map was used to determine which node to return to. In addition, the robot also performed re-localization once to further reduce its map uncertainty.

Fig. 7 shows the state of the map covariance at two different times during the mapping process; in Fig. 7a and 7c, the robot has just finished exploring the entire environment whereas in Fig. 7b and 7d the robot has completed a re-local-

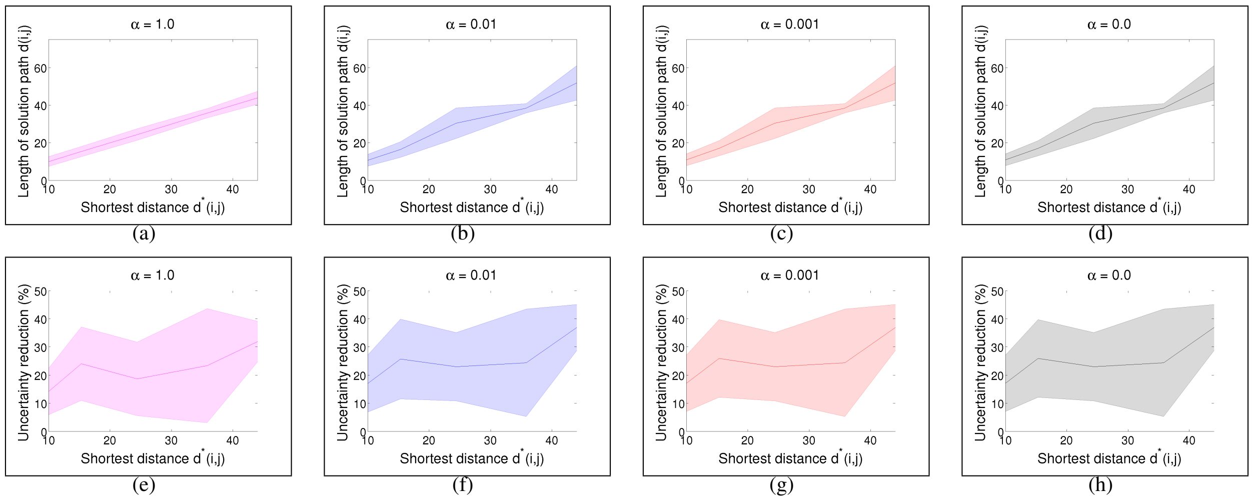



Figure 6. (a)-(d) Error bar plots showing length of the solution path $d^*(i,j)$ compared to the shortest path distance using $\alpha = 1.0, 0.01, 0.001$ and $1.0$ for (a) through (d), respectively. (e)-(h) error bar plots showing total uncertainty reduction (in %) from single re-localization for all pairs $(i,j)$ again using $\alpha = 1.0, 0.01, 0.001$ and $1.0$ for (e) through (h), respectively.

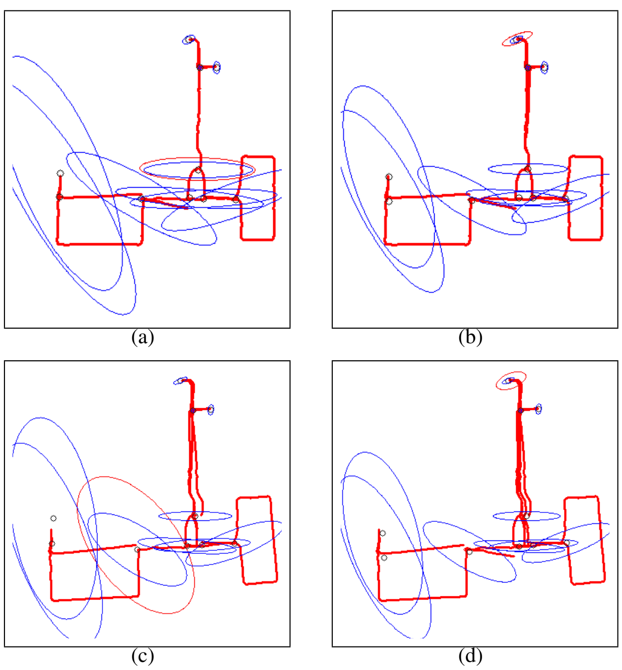



Figure 7. (a) shows the map 2-$\sigma$ uncertainty ellipses after exploring the entire environment without re-localizing during exploration (b) shows the map uncertainty reduction after performing re-localization (c) and (d) show the same maps as in (a) and (b), except that re-localization was performed during exploration of the environment (prior to the map shown in (c)).

ization procedure. The red uncertainty ellipse is the current covariance of the robot, while the blue uncertainty ellipses are the covariances of the meetpoints in the map. Notice that the 2-$\sigma$ uncertainty ellipses in Fig. 7c and 7d are smaller than in Fig. 7a and 7b. This reduced uncertainty is the result of performing re-localizations prior to completely exploring the environment. The explanation is that when the robot uncertainty has increased, further exploration only results in large uncertainty maps. Re-localizations after exploring the entire environment are still useful, as demonstrated in the contrast between the two figures Fig. 7a and 7b before and after re-localizing. However, maintaining a lower uncertainty during exploration is key to result in more accurate final maps. The numerical results for $tr(\mathbf{P}^{map})$ show that the final map in Fig. 7d is 41.7% more accurate than in Fig. 7b. Even more importantly, the trace of uncertainty in Fig. 7c is only 53.2% that of Fig. 7a. This highlights the importance of frequent re-localizations during the exploration phase.

## VII. Conclusion

This paper presented an extension to our previous work in autonomous exploration and mapping using a hybrid metric/topological representation. The SLURM framework [17] was adapted from a camera sensor network setup to an indoor environment exploration setting. The work presents a SLAM solution based on a combination of the Generalized Voronoi Graph for the topological aspect and an extended Kalman filter to keep track of a metric map.

The results shown in this paper were collected by running in simulated environments on the Stage platform in ROS as well as on a real robotic system, the TurtleBot 2. We are looking to perform further experimentations in different environments with the TurtleBot 2 as well as the Husky[3].

This work extended the exploration framework with a heuristic search planning approach to map uncertainty reduction. The search algorithm exploits the known information about the environment by propagating the stored edge information through parallel hypotheses filters to simulate the

[3] http://www.clearpathrobotics.com/husky/

effects of re-localization on the actual system. The output of the proposed strategy is a solution path for the robot to re-localize to which is estimated to be most beneficial in map uncertainty reduction. Our technique is validated by the results from running simulations in a realistic environment and running on a real robotic system. Moreover, the effects of our hyper-parameter $\alpha$ on the length and the uncertainty reduction of the output solution path are elaborated.

Overall, the results suggest that uncertainty reduction can be incorporated into the SLAM framework to produce even more accurate maps at the cost of added computational overhead. The inclusion of a free parameter $\alpha$ in our heuristic search algorithm makes the uncertainty reduction strategy even more pragmatic since the user has the flexibility of picking $\alpha$ to suit the needs of the mapping task.

## Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the NSERC Canadian Field Robotics Network (NCFRN) for their generous support.

## References

[1] D. Meger, I. Rekleitis, and G. Dudek, "Heuristic search planning to reduce exploration uncertainty," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, France, Sep. 2008, pp. 3392–3399.

[2] I. M. Rekletis and G. Dudek, "Automated calibration of a camera sensor network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton Alberta, Canada, Aug. 2-6 2005, pp. 401–406.

[3] I. Rekleitis, "Simultaneous localization and uncertainty reduction on maps (slurm): Ear based exploration," in *Proc. of IEEE International Conference on Robotics and Biomimetics*, Guangzhou, China, Dec. 2012, pp. 501–507.

[4] ——, "Multi-robot simultaneous localization and uncertainty reduction on maps (mr-slurm)," in *Proc. of IEEE International Conference on Robotics and Biomimetics*, Shenzhen, China, Dec. 2013, pp. 1216–1221.

[5] Q. Zhang, D. Whitney, F. Shkurti, and I. Rekleitis, "Ear-based exploration on hybrid metric/topological maps," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014.

[6] H. Choset and J. Burdick, "Sensor based planning, part ii: Incremental construction of the generalized voronoi graph," in *Proc. of IEEE Conference on Robotics and Automation*. Nagoya, Japan: IEEE Press, May 1995, pp. 1643 – 1648.

[7] R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *Proc. of International Conference in Robotics and Automation*, Apr. 2005, pp. 661–666.

[8] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierachy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, pp. 46–63, 1991.

[9] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Using multiple markers in graph exploration," in *Proc. Symposium on Advances in Intelligent Robotics Systems: Conference on Mobile Robotics*. Philadelphia, PA: International Society for Optical Engineering, 1989.

[10] ——, "Robotic exploration as graph construction," *Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 859–865, December 1991.

[11] F. Werner, F. Maire, J. Sitte, H. Choset, S. Tully, and G. Kantor, "Topological SLAM using neighbourhood information of places," in *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4937–4942.

[12] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 125–137, 2001.

[13] R. Smith, M. Self, and P. Cheeseman, "Autonomous robot vehicles," I. J. Cox and G. T. Wilfong, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 1990, ch. Estimating Uncertain Spatial Relationships in Robotics, pp. 167–193. [Online]. Available: http://dl.acm.org/citation.cfm?id=93002.93291

[14] C. Stachniss and W. Burgard, "Exploration with active loop-closing for fastslam," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

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

[16] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. Castellanos, "Active policy learning for robot planning and exploration under uncertainty," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[17] I. Rekleitis, "Single robot exploration: Simultaneous localization and uncertainty reduction on maps (slurm)," in *Proc. of Nineth Canadian Conference on Computer and Robot Vision*, Toronto, ON, May 2012, pp. 214–220.