2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Efficient Mobile Robot Exploration with Gaussian Markov Random Fields in 3D Environments

Chaoqun Wang[1], Teng Li[2], Max Q.-H. Meng[1*] and Clarence de. Silva[2]

***Abstract*— In this paper, we study the problem of autonomous exploration in unknown indoor environments using mobile robot. We use mutual information (MI) to evaluate the information the robot would get at a certain location. In order to get the most informative sensing location, we first propose a sampling method that can get random sensing patches in free space. Each sensing patch is extended to informative locations to collect information with true values. Then we use Gaussian Markov Random Fields (GMRF) to model the distribution of MI in environment. Compared with the traditional methods that employ Gaussian Process (GP) model, GMRF is more efficient. MI of every sensing location can be estimated using the training sample patches and the established GMRF model. We utilize an efficient computation algorithm to estimate the GMRF model hyperparameters so as to speed up the computation. Besides the information gain of the candidates regions, the path cost is also considered in this work. We propose a utility function that can balance the path cost and the information gain the robot would collect. We tested our algorithm in both simulated and real experiment. The experiment results demonstrate that our proposed method can explore the environment efficiently with relatively shorter path length.**

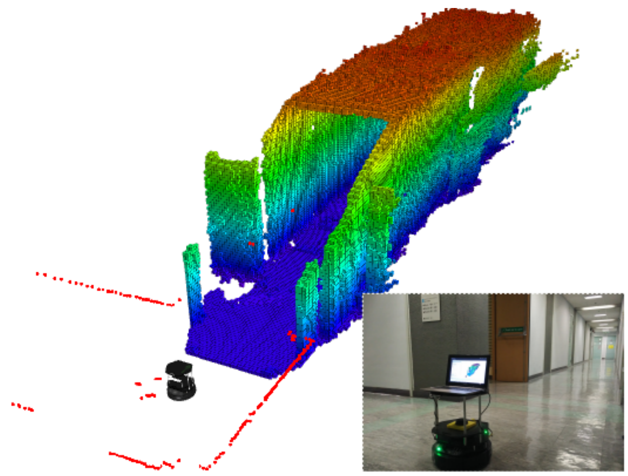


Fig. 1. Autonomous exploration of mobile robot in 3D corridor environments.

I. INTRODUCTION

Active exploration is the ability of collecting the information autonomously without human intervention.The robot actively selects the sensing location so as to collect more information and improve its navigation performance at the same time. This ability is one of the key parts for robot to perform its task autonomously. It has many applications in environment monitoring, surveillance and many other areas. In order to acquire more information within a certain time budget, robot need to make decisions to determine where to go. In addition to the traditional algorithm, for example, frontier-based method that always guides the robot to the nearest frontier [1][2][3], there are also some work that use information-gain based methods [4][5]. Information-gain based method tries to guide the robot to the areas that can collect more information.

A target environment is generally unknown to a robot in autonomous exploration applications. It is a challenge to collect information at all sensing locations by a robot due to the computational burden of ray casting. In [4], a mutual information-based method has been proposed to perceive an environment actively. GP was employed to predict the mutual information in each candidate sensing location. Similar method has been adopted in [5]. They utilized the Bayesian Optimization (BO) to facilitate the selection of sensing actions. As indicated in [6], due to the computational issues, a GP model is not efficient in comparison to a GMRF model. A GMRF model can approximate a GP model quite well with less computational burden by implementing a precision matrix.

In the present work, we consider the problem of autonomous exploration using a mobile robot in an indoor environment, as Fig.1 shows. We establish a GMRF model to characterize the distribution of MI that is utilized to evaluate the informative sensing locations. Training samples are generated randomly in free space. Instead of using sensing locations separately, we utilize a sensing location and its neighbors as a sensing patch. Furthermore, the sensing patch can extend the area that is rich in information. We obtain the MI of the sensing patches by the ray casting method. Then the sensing patches are used to estimate the hyperparameters of the model. The established model is then utilized to predict the MI of other candidate locations that are not sampled. The MI of the unsampled locations are inferred by the environmental model and the training samples. The computational burden of GMRF is less than that of GP. The efficient computation of the hyperparameters of the GMRF model further speeds up this process. We use this proposed pipeline to explore a 3D environment with mobile robots in an indoor environment.

This paper will organized as follows. In section II, we

[1]Chaoqun Wang and Max. Q. H. Meng are with The Chinese University of Hong Kong, China.{cqwang, qhmeng}@ee.cuhk.edu.hk
[2]Teng Li and Clarence W. de Silva are with The University of British Columbia, Vancouver, BC, Canada. {tengli,desilva}@mech.ubc.ca
*Corresponding author

will introduce some related work in the area. Nextly we would give the problem formulation and some definitions in section III. In section IV, we will present our proposed method for solving this problem. Then we would introduce the experiment and results in V. Finally in section VI, we would give the conclusion and future work.

II. Related Work

A large amount of work has been done to solve the exploration problem. The seminal work in [1] use frontier to guide the exploration. Frontier region is the area between explored and unexplored regions. Robots always tend to move to the nearest frontier during the exploration. However, this method is myopic and do not consider the long term reward. Sometimes robots move to the regions that are the nearest but contribute little information. Recently information gain based exploration has gained much attention. In the earlier work in [6], they proposed to maximize the mutual information among sensing locations during the exploration. Charrow et al. proposed a newly Cauchy-Schwarz Quadratic Mutual Information (CSQMI)[7], which is more computationally efficient than MI. The candidate regions whose CSQMI is high would be selected as the target. In [8], the knowledge of environment that is previous seen is taken into consideration in order to make better decision.

Along with the evolution of simultaneously localization and mapping (SLAM) in robotics area, the information based exploration method is prevalent. Robot seeks to maximize the information so as to reduce the mapping and localization uncertainty. In [9], they tried to maximize the knowledge about the map while reduce the localization uncertainty at the same time. Stachniss et al. use Rao-Blackwellized particle filters to compute map and pose posteriors. They balance the cost of executing the path to the target and the information it would collect in the target regions. In [10], they proposed to generate a path for perceive the environment while reduce the localization uncertainty concurrently.

In [11], Shi et al. proposed an information-theoretic exploration method. Under this framework, the work aims to move to the location whose MI is the biggest. They model the environment using GP. The mutual information on the grid is predict using the GP optimized by Bayesian Optimization. Considering the computational efficiency issue of GP model, Linh V. et al. proposed to use GMRF to model the environment [12]. They generated several points with the maximum MI to represent the whole environment using the GMRF model. Then they use a stochastic partial differential equations to get the coefficient of GMRF model. This work emphasize more on the sensor deployment problem in wireless sensor network in environment modeling. In [13], they also consider using MI for guiding the exploration. They use GP to predict a gradient field of occupancy probability so as to provide frontier boundaries. Javier G. et al. proposed to model the gas distribution using Gaussian Markov Random Fields [14]. The process involve some prior knowledge of the environment and the obstacles that affect the distribution of the gas.

III. Problem Formulations

A. Notations

During exploration, robots can actively determine where to go according to the information gain in order to get the entire environment model. The 3D space to be explored is defined as $V \subset R^3$. The entire space V is supposed to be unknown at the beginning, i.e. $V = V_{unknown}$. The space that the robot has already explored is set either V_{free} or V_{obs}, both are regarded as V_{known}. Our goal is to increase the number of voxels that belong to V_{known} whilst reducing the number of unknown voxels $v \subset V_{unknown}$ as much as possible. OctoMap is utilized to represent the 3D environment [15], which is very efficient tool. We use entropy H to describe the uncertainty of the map [16]:

$$H(V) = -\sum_{k=1}^{N} p(v_k) log(p(v_k)) \tag{1}$$

Here $p(v_k)$ is the occupancy probability of voxel v_k in the OctoMap, N represents the total number of voxels in the OctoMap. We aim to reduce the map uncertainty H within limited time. The mutual information MI is proposed to represent the reduced entropy after the robot at certain location l.

$$MI(V) = H(V) - H(V|l) \tag{2}$$

We use path length L to represent the path cost in the exploration. We aim to reduce both path cost L and map uncertainty H. Thus location l shall be chosen according to the utility function determined by the two requirements mentioned above:

$$l = argmin\ \alpha MI \oplus \beta L \tag{3}$$

here α and β are coefficients that balance MI and H. These two quantities is banlanced by operator $\oplus$. We propose a new utility function to balance these two quantities, which would be introduced in detail in the next section.

We use ray casting method to calculate the mutual information MI at a certain location l. Bresenham algorithm is utilized to calculate the voxels the sensor ray scanned. The voxels that $v \in V_{unknown}$ would be used to evaluate the MI at that location. The ray casting method is performed on the current map m_p that the robot already explored. At every sampling location that performs ray casting, we would get the training pairs $\mathbf{Tr} = \{\mathbf{Lr}, \mathbf{Mr}\}$ where $Tr_i = \{L_i, M_i\} \in \mathbf{Tr}$ represent a robot location and its corresponding MI. Since ray casting is computational expensive, we proposed to model the environment using GMRF and predict the MI at other locations that is not observed. In order to make the training samples more representative, we proposed a new sampling method.

B. Gaussian Markov Random Fileds

In sampling process , we get pairs $\mathbf{Tr} = \{\mathbf{Lr}, \mathbf{Mr}\}$, we would like to predict the MI of other sensor locations, i.e. $\mathbf{Ts} = \{\mathbf{Ls}, \mathbf{Ms}\}$, using Gaussian model for all the data and hence we get:

$$\begin{bmatrix} Tr \\ Ts \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_{Tr} \\ \mu_{Ts} \end{bmatrix}, \begin{bmatrix} \Sigma_{TrTr} & \Sigma_{TrTs} \\ \Sigma_{TrTs}^T & \Sigma_{TsTs} \end{bmatrix}\right) \quad (4)$$

let θ to be the parameter for covariance and mean, we have:

$$Tr \sim N(\mu(\theta), \Sigma(\theta)) \quad (5)$$

We could get the parameter θ using maximize likelihood method, then the value to be predict can be written as:

$$E(Ts|Tr, \theta) = \mu_{Ts} + \Sigma_{TsTs}\Sigma_{TsTr}^{-1}(Tr - \mu_{Ty}) \quad (6)$$

Using this formulation and given training dataset, we could predict the MI of unmeasured locations. While this formulation using GP encountered "Big N" problem. The computation complexity would be $O(N^3)$ given N observations in the training dataset. In order to reduce the computational complexity, we proposed to use GMRF to model our environment. In our exploration problem, we assume the independence of voxels that are far away from each other. Suppose we have locations L_i and $L_j \in N$ represent the neighbors that are close to L_i according to some evaluation metric. A GMRF can be defined as:

$$P(L_i|L_j, j \neq i) = P(L_i|L_j, L_j \in N) \quad (7)$$

which means that the MI value of a certain location is just related to some neighbors instead of all the neighbors. This could help with reducing the computational complexity in our algorithm. The precision matrix $Q = \Sigma^{-1}$, with this property , we can rewrite Equation (6) as:

$$E(Ts|Tr, \theta) = \mu_{Ts} + Q_{TsTr}Q_{TrTr}^{-1}(Tr - \mu_{Ty}) \quad (8)$$

The term $E(Ts|Tr, \theta)$ represents the MI of locations that needed to be predicted. The key to predict these values is to get the precision matrix Q, here we proposed to use efficient algorithms to compute the matrix efficiently. With GMRF, we model the environment as:

$$y(L) = \mu + Z(L) + \epsilon(L) \quad (9)$$

where μ is a parameter that is known and random, $Z(L)$ is the GMRF model with zero mean vector and precision matrix Q. $\epsilon(L)$ is independent identically distributed with $N(0, \sigma^2)$.

IV. Proposed Method

A. *Efficient computation of Q*

We use GMRF to model the environment to facilitate exploration. The key to represent the data using GMRF is the computational of precision matrix Q. The stochastic partial differential equations (SPDE) proposed by Lindgren [17] is a useful tool to compute matrix Q. Here in our proposed framework we follow the framework proposed in [18], which is more efficient and accurate.

As Fig. 2 shows, we define $S_\theta(x)$ as $\{y \in Z^2 : \theta(x, y) \neq 0\}$. For $1 \leq i \leq n$, if $S_\theta(x_i) \subset \{x_1, ..., x_n\}$, then we say x_i is a fully neighbored location. $Z(x_i)$ is a fully neighbored observation. If not, then we say x_i is partially neighbored. Suppose there are m_n partially neighbored observations in $\{x_1, ..., x_n\}$. Define a matrix P that $PZ = (Z_1', Z_2')'$ in which Z_1 and Z_2 contains the partially and fully neighbored observations separately.

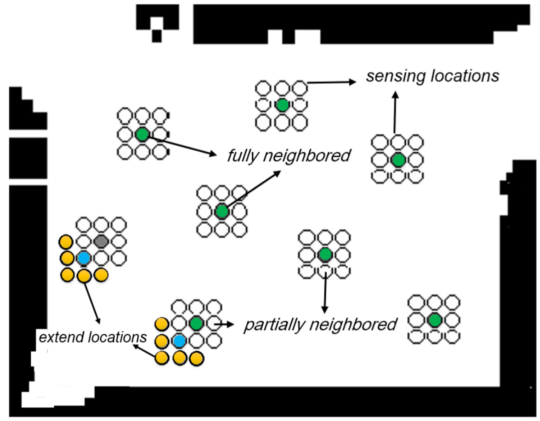


Fig. 2. The circles represent the sensing locations. Circles with green color interior represent the fully neighbored observation locations.Blank circles represent the partially neighbored observation locations. The blue color filled circles represent the central locations that have more MI.The yellow filled circles represent the added locations

Define block matrix as:

$$P\Sigma_\theta P' = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, PQ_\theta P' = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad (10)$$

The spectral representation of the covariance function of a stationary random field could be represent like the the first equation in Equation (11). Based on the assumption that the observation locations falls on a grid (n_1, n_2), the equation can be adjust into the second equation in Equation (11).

$$\begin{aligned} K_\theta(x, y) &= \int_{[0,2\pi]^2} \frac{exp(i\omega'(x-y))}{\Sigma_{h\in Z^2}\eta(h)exp(i\omega' h)} d\omega \\ K_\theta(x, y; J, n) &= \frac{4\pi^2}{n_1 n_2 J^2} \sum_{j \in F_j} f_\theta(\omega_j) exp(i\omega_j'(x-y)) \end{aligned} \quad (11)$$

In [18], they give a proof that given a bounded above function $f(\omega)$ as the spectral density of a Markov Random Field, then the following equations exists:

$$|K_\theta(x, y) - K_\theta(x, y; J, n)| \leq \frac{C_p}{(n_1 J)^p} \quad (12)$$

We use the following formulation to calculate the MI at certain location as:

$$E(Ts|Tr) = \mu_{Ts} + \Sigma_{TsTs}\Sigma_\theta^{-1}(Tr - \mu_{Ty}) \quad (13)$$

We would get the following:

$$(a) det(\Sigma_{\theta)} = det(\Sigma_{11})/det(\Sigma_{22})$$

$$(b) \Sigma_\theta^{-1} = \begin{bmatrix} I & -\Sigma_{11}^{-1}\Sigma_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Sigma_{11}^{-1} & 0 \\ 0 & Q_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\Sigma_{21}\Sigma_{11}^{-1} & I \end{bmatrix} \quad (14)$$

The entries of Σ_{11} can be obtained by Equation(11). Q_{22} is known analytically. Using circulant embedding techniques, multiplication of Σ_{21} or Σ_{12} can be complete. We refer the reader to read [18] for more retails.

B. Resampling

We proposed a resampling method for generate sensing patches randomly in V_{free}. Instead of generating sensing locations singly and randomly, in the first round, we generate several random sensing locations in V_{free}. Then in order to use the method proposed in the former section, we propose to use the eight neighbors around every sensing location generated in the first round. We call the sensing group a sensing patch. In the second round, we calculate the MI of every sensing location. For every sensing patch, we compare the MI of every sensing location. The location which gain much mutual information would be selected as another central location whose eight neighbors would be selected as sensing locations.

As Fig.2 shows, we calculate the MI of green circles and their eight neighbors. For every sensing patch, the one that have much information would be selected as another sensing center, as the blue circles show. Then the eight neighbors of the blue circle would be added to the sensing patch as the training sample for the GMRF model.

C. Balance of Path Cost and information gain

In this section, we would introduce our proposed method for goal point selection so as to balance the path cost and map entropy in the long run. Instead of computing the entropy of the map in every move step of the robot, we choose several candidate regions f_i that the robot expect to explore according to the distribution predicted by GMRF model. I_i represent the information gain corresponding to a certain f_i. To get a target point, we consider both the path cost and the information gain we would get at that point. Here we proposed a utility function to evaluate the candidate goal regions:

$$U = \delta_1 \frac{\lambda_1 I_i}{\sum_{i=1}^{N} I_i} + \delta_2 \frac{\lambda_2 l_i}{\sum_{i=1}^{N} l_i} \tag{15}$$

We normalize these two quantities I and l so as to evaluate them together. Instead of just sum them together with two adjustable coefficient λ_1, λ_2. We consider the divergence of these variables. For example, for the information gain I_i at different regions, we use standard deviation to describe the divergence of I_i with $i \in 1...N$. Hence we get:

$$\delta_1 = dev(I_i), \delta_2 = dev(I_i), i \in 1...N \tag{16}$$

This is meaningful because in some cases, the divergence of the variable would influence the performance obviously. This is based on our observation that when the divergence of one variable is significantly large, that means the value of this decision variable is of great importance. While we can not achieve the same performance singly rely on the fixed coefficient λ_1, λ_2 when they have been set in a certain task.

Our aim is to find the target that have the minimum value of utility function, that is:

$$\begin{cases} min. \;\; U(I_i, l_i) \\ s.t. \;\; l_i \in L, I_i \in I \end{cases} \tag{17}$$

Firstly we compute the mutual information of candidate regions and the path length it would take. Then we could get the target that have the minimum value of utility function.

D. Implementation Details

We use Octomap to represent the 3D environment. With this representation, we compute the MI of several sampling sensing locations using ray casting method. The sampling locations are set with random sampling which we guarantee the distribution of sampling points is uniformly. As we all know, ray casting method is relatively a computational inefficient method. It would be a heavy computational burden to compute all the MI of every sensing location. In our proposed framework, we use GMRF model to model the distribution of MI. Using this model, we could predict the MI of other locations that do not perform ray casting, as Fig.3 shows. In order to speed up the computation of precision matrix Q, we use a novel method that can get the MI value of unmeasured locations efficiently.

To simplify the problem, we choose several top M candidate sensing locations as the candidate sensing location. Since MI has been taken into consideration in the whole framework. Most work that employ the GP or GMRF model to represent the environment do not consider the path cost. We use a utility function which consider the MI and the path cost. In the meanwhile, the SLAM module produces the localization information and map for the exploration process.

Algorithm 1: Exploration Process

Input: Partial Map
Output: Goal
$L_s = Sampling(Map)$;
for $i < size(L_s)$ **do**
  for $j < 8$ **do**
    $MI \leftarrow raycast(L_s(i))$;
    $MI_n \leftarrow raycast(L_s.neighbor_j)$
  end
  $M \leftarrow rank(MI_n)$;
  $L1 = L_s.neighbor_{M(1)}$;
  $L2 = L_s.neighbor_{M(2)}$;
  $Lss.pushback(L_s(i), L_s(i).neightbor$
   $L_1, L2, L1.neihtbor, L2.neighbor)$;
end
$L_r = GMRF(Map, L_{ss})$;
$P = samplingPlanner(L_{ss})$;
for $i < size(L_{ss})$ **do**
  $U = utility(L_{ss}, P)$
end

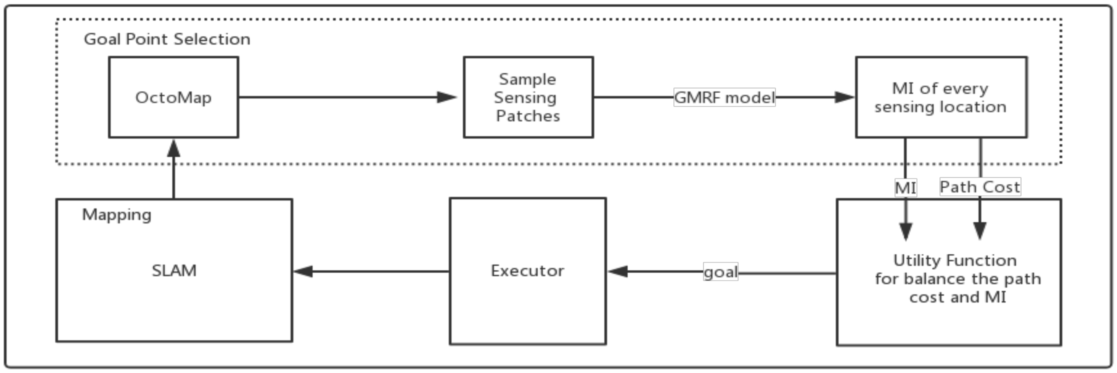


Fig. 3. The framework of our proposed exploration method. Octomap is employed to represent the 3D world. Sampling module produces several training samples in the environment. GMRF help with predict the MI of every voxel. The voxels has two attributes: 1) MI 2) Path cost to reach there. We proposed a function to balance these two attributes so as to shorter the exploration time. The SLAM module is responsible for localization and providing the environment map.

Algorithm 1 shows the pipeline of our proposed method. With $Sampling$ module we can get random sampling locations separately in V_{free}. This is the base step in our proposed framework. Instead of calculating the MI of the sampling locations, we calculate MI of every patch, i.e. the sampling points and their eight neighbors. Line 2-11 in Algorithm 1 shows the process of generating sensing patches. For a sensing patch, after we get the MI of the sampling points and their neighbors, we choose the first two neighbors which have more MI than the others as new sensing centers. These two points along with their eight neighbors would also be added in our sampling set. $GMRF$ model the distribution of MI in the environment using training dataset L_{ss}. Line 14-16 shows the balance of path cost and information gain using our proposed utility function. We use our planner proposed in work [19] as the path planner to get the path cost.

V. Experiment and Result

1) Environment Setup: We use both simulation environment and real experiment to demonstrate the efficiency of our proposed method. In simulation environment, we use three different kinds of environments, the office like environment, clustered environment and maze like environment. We use Gazebo as the simulation engine, all the physical parameter of the robot and sensors in simulation environment is set according to their values. The real experiment employs a real turtlebot robot with a asus laptop with Intel Core i7-4510U at 2GHz, 8GB memory (without GPU). The robot is equipped with a low-cost rplidar laser scanner with range 0.2m to 6m, and 360° area scanning range for localization, and Asus Xtion 3D depth camera, which can provide 30Hz RGB and depth images, with 640x480 resolution and 58 HFV.

2) Results: Fig.4 demonstrates the performance of our proposed sampling algorithm and prediction result using GMRF model. Fig.4(a) shows the partial map of the environment. Fig.4(b) shows the ground truth. We get several

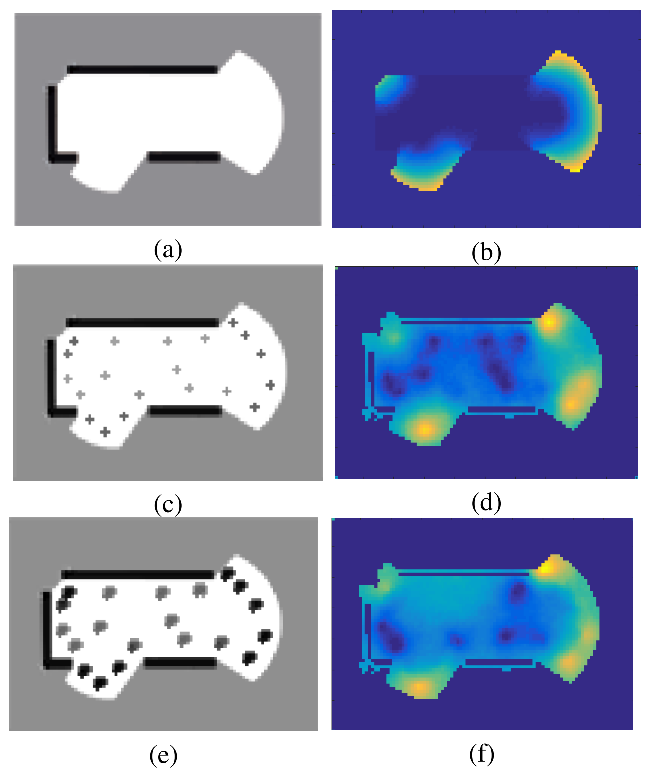
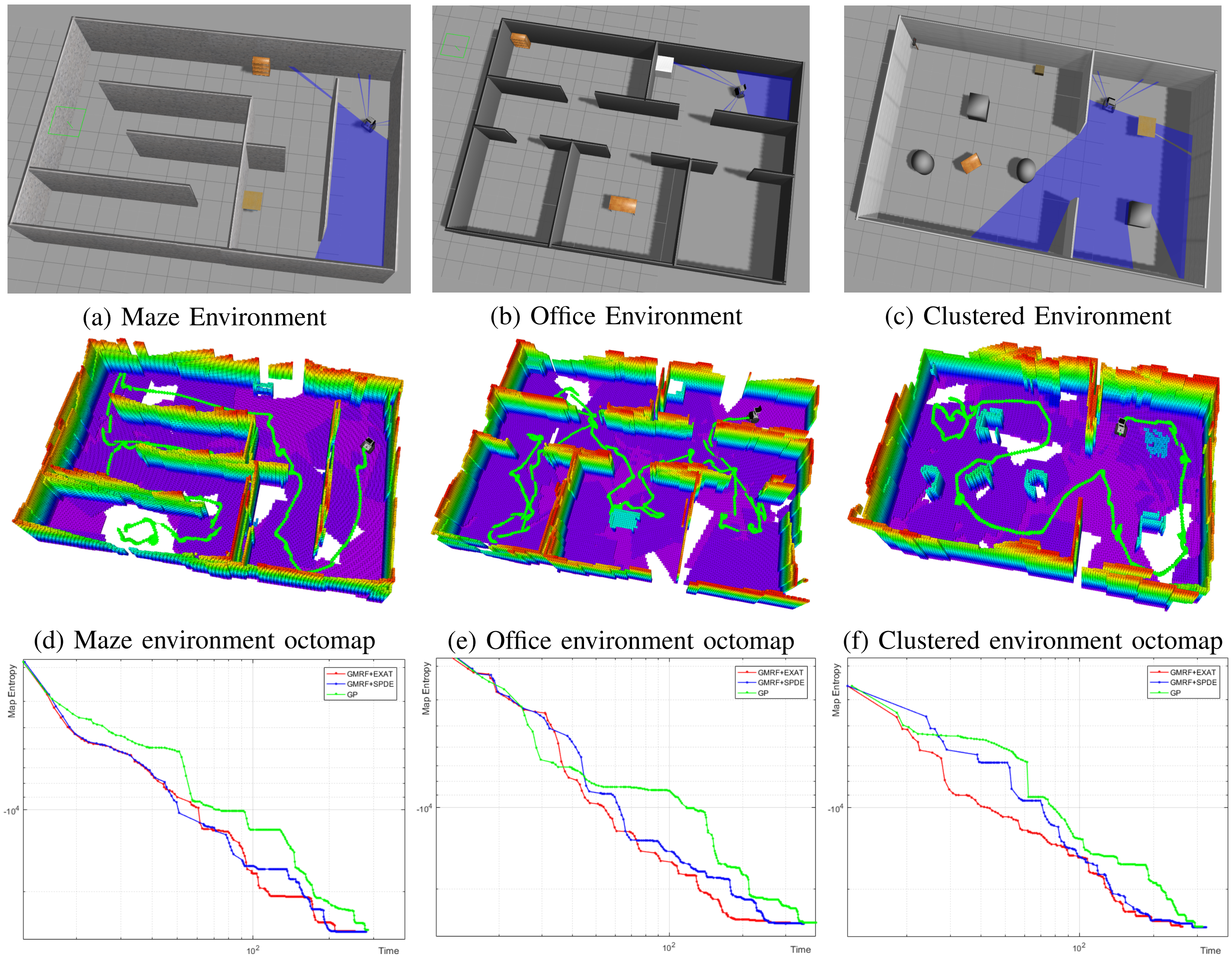


Fig. 4. Experiment result of our proposed sampling method. (a-b) is the original partial map and the ground truth that compute the MI of every location.(c) is simpling in several separate locations. (d) is the corresponding MI distribution predicted by GMRF. (e-f) shows the result of our proposed sampling patches and distribution of MI.

random locations in partial map. Using ray casting algorithm, we get the MI of the sampling points. The amount of MI is represent by the markers with different shades. Fig.4(d) shows the predicted MI corresponding to the map. Fig.4(c) shows the sampling result using our proposed method. Instead of using several separate grids as sensing locations, our proposed sensing patches gained better performance. As



(a) Maze Environment (b) Office Environment (c) Clustered Environment

(d) Maze environment octomap (e) Office environment octomap (f) Clustered environment octomap

(g) Entropy of the maze map (h) Entropy of the office map (i) Entropy of the clustered map

Fig. 5. Simulated result of our proposed exploration method. We test our algorithm in three different kinds of environment: (a) maze like environment (b) office like environment (c) clustered environment. (d-f) shows the 3D map we built with our proposed pipeline. The green line shows the trajectory during exploration. (g-i) shows the entropy during exploration comparing with GP based method

Fig.4(f) shows, comparing with ground truth, our proposed sampling method can gain much better performance than singly using separate sensing locations in Fig.4 (e).

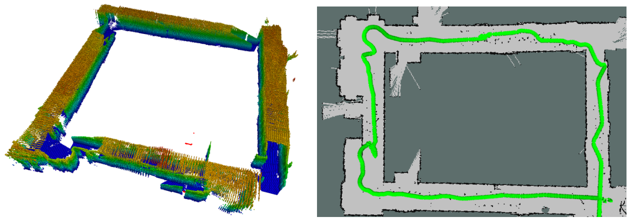
We test our algorithm in three different kinds of environments: maze like environment, office like environment and clustered environment. All the environment is set to be $20m \times 10m$. The robot is equipped with a laser scanner for localization and kinect for 3D sensing. Fig.5 shows our experiment results. In all these environments, our proposed method can explore the environment more efficiently than the other methods. For maze like environment, the behavior is little same with that of wall following algorithm. While the difference exists when the robot meet several walls. Wall following algorithm can only follow the wall without considering the information the area contained. Compared with GP based algorithm, our proposed is more efficient in all these environments. The result also demonstrate the efficiency using our proposed method to calculated the coefficients of GMRF. Fig.5 (g)-(i) illustrate the performance of our proposed method. Green lines represent the entropy change of the map using GP model to predict the MI distribution in the environment. The other lines represent the exploration result using GMRF model. The red line is the performance of our proposed method that get the exact likelihood of GMRF model, while the blue line represent the traditional way to solve the GMRF using SPDE. The graphs demonstrate that our proposed method is overwhelming than the other methods.

We also test our algorithm in real experiment. The robot is employed to explore an indoor hallway environment and office room environment. The time spent and path length is shown in table I together with the simulation result in three

	GMRF modeling (our method)		**GP modeling**	
	Time cost(s)	Path length(m)	Time cost(s)	Path length(m)
Simulated				
1	260	**78**	281	**86**
2	313	**98**	405	**106**
3	350	**130**	398	**132**
Real				
1	480	**107**	500	**103**
2	280	**70**	310	**93**

TABLE I

COST EVALUATION IN SIMULATED AND REAL EXPERIMENT.



(a) (b)

Fig. 6. Real experiment of our proposed method (a) The 3D map of the environment (b) The corresponding 2D map and the trajectory of the robot during exploration

environments. Comparing with the GP based exploration method, our proposed method can explore the environment with relative less path length and less time cost.

In real experiment, the robot explore the environment autonomously using our proposed pipeline. The area of the environment is about 320 m^2 and it takes 490 seconds to explore the whole environment. As Fig.6 shows, Fig.6(a) shows the 3D map of the corridor environment. The following 2D map shows the trajectory of the robot during exploration. As Fig.6(b) shows, when the robot is traveling in a narrow corridor, the path is relatively smooth. While when the robot is in some open areas, the trajectory become back and forth. This is because the range of the 3D sensor can not cover the area at one time, what's more, here we use a global planner that do not consider the local information the robot would collect.

VI. Conclusion and Future Work

In this paper, we propose to use GMRF model to represent the distribution of MI in the unknown environment. Compared with GP model, GMRF model is more efficient. We present a newly sampling strategy that can represent the truly distribution of the MI as much as possible. Based on the random sampling locations in the map, we utilize the neighbors of sampling points. Hence we use sampling patches instead of separated sampling locations. We use MI to evaluate the information we would get at certain sensing location. With the GMRF model, we can get the MI of all sensing locations. A novel method that can accelerate the computation of precision matrix Q of GMRF model is used in this paper. Compared with the traditional SPDE method, the new method for getting the hyperparameter of GMRF model is more accurate and efficient. What's more, we proposed a novel utility function to balance the path cost and the information gain the robot would get instead of just considering the information during the exploration process. In the future, we would test our proposed algorithm in more complex real environments. What's more, we would exploit the dependency of explored regions and unexplored regions so as to make better decision during exploration.

VII. Acknowledgement

This project is partially supported by the Hong Kong RGC GRF grant # 14205914, ITC ITF grant ITS/236/15, and Shenzhen Science and Technology Innovation project c.02.17.00601, awarded to Max Q.-H. Meng.

References

[1] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.

[2] Matan Keidar and Gal A Kaminka. Efficient frontier detection for robot exploration. *The International Journal of Robotics Research*, 33(2):215–236, 2014.

[3] Jan Faigl, Olivier Simonin, and Francois Charpillet. Comparison of task-allocation algorithms in frontier-based multi-robot exploration. In *European Conference on Multi-Agent Systems*, pages 101–110. Springer, 2014.

[4] Francesco Amigoni and Vincenzo Caglioti. An information-based exploration strategy for environment mapping with mobile robots. *Robotics and Autonomous Systems*, 58(5):684–699, 2010.

[5] Luca Carlone, Jingjing Du, Miguel Kaouk Ng, Basilio Bona, and Marina Indri. An application of kullback-leibler divergence to active slam and exploration with particle filters. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 287–293. IEEE, 2010.

[6] Peter Whaite and Frank P Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997.

[7] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4791–4798. IEEE, 2015.

[8] Daniel Perea Ström, Igor Bogoslavskyi, and Cyrill Stachniss. Robust exploration and homing for autonomous robots. *Robotics and Autonomous Systems*, 90:125–135, 2017.

[9] Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 540–545. IEEE, 2002.

[10] Gabriele Costante, Christian Forster, Jeffrey Delmerico, Paolo Valigi, and Davide Scaramuzza. Perception-aware path planning. *arXiv preprint arXiv:1605.04151*, 2016.

[11] Shi Bai, Jinkun Wang, Fanfei Chen, and Brendan Englot. Information-theoretic exploration with bayesian optimization. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1816–1822. IEEE, 2016.

[12] Linh V Nguyen, Sarath Kodagoda, and Ravindra Ranasinghe. Spatial sensor selection via gaussian markov random fields. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(9):1226–1239, 2016.

[13] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. Mutual information-based exploration on continuous occupancy maps. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6086–6092. IEEE, 2015.

[14] Javier G Monroy, Jose-Luis Blanco, and Javier Gonzalez-Jimenez. Time-variant gas distribution mapping with obstacle information. *Autonomous Robots*, 40(1):1–16, 2016.

[15] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[16] Claude E Shannon, Warren Weaver, and Arthur W Burks. The mathematical theory of communication. 1951.

[17] Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

[18] Joseph Guinness and Ilse CF Ipsen. Efficient computation of gaussian likelihoods for stationary markov random field models. *arXiv preprint arXiv:1506.00138*, 2015.

[19] Chaoqun Wang and Max Q-H Meng. Variant step size rrt: An efficient path planner for uav in complex environments. In *Real-time Computing and Robotics (RCAR), IEEE International Conference on*, pages 555–560. IEEE, 2016.