CCD* Lite coverage path planning with landmark weighted path

Rahul Moghe[1], and Meredith Pitschm[2], Tiffany Yang[3] and Peter Stone[4]

Abstract— **This paper presents a practical CCD* Lite algorithm for complete coverage path planning which handles dynamic environment constraints efficiently. The wavefront method and distance transform is used to compute a path from a given initial state. Our novel method of generating wavefronts assign priority to landmarks present in the environment which help the robot localize itself on the map. D* Lite path planning algorithm is then used to dynamically plan a path around the map and also re-plans the path in presence of an unknown obstacle online. As a result, our algorithm is useful for implementation on a real system. The effectiveness of this algorithm is tested on a humanoid robot walking in the 2D environment with dynamic obstacle constraints and beacons as landmarks. The robot was found to handle dynamic obstacles and recompute a path around them in real time.**

I. INTRODUCTION

Complete coverage path planning (CCPP) techniques have been developed for real world applications like vacuum cleaning robots [1], [2], [3], painting robots [4], autonomous underwater vehicles [5], demining application [6], [7], [8], agriculture [9] and inspection of structure [10] to name a few. However, their performance has only been tested in a simulated environment. Existing sensor-based CCPP algorithms do not factor in the robots ability to follow the path by using feedback from landmarks in the environment. Current algorithms also do not efficiently handle dynamic environments. This paper focuses on designing a coverage path for dynamic obstacles while maintaining view of the landmarks that improve the sensor feedback of the robot.

The earliest coverage path planning (CPP) methods include randomized algorithms where the actions of the agent are chosen randomly. CCPP algorithms can be classified as offline, where the map is assumed to be perfectly known, or online, where sensor data is used online to determine the map and cover it. Online algorithms are also known as sensor-based coverage algorithms. Planning algorithms using genetic algorithms, neural networks, cellular decomposition, spanning trees, spiral filling paths and ant colony method fall into the offline category.

This paper focuses on online algorithms where a partially known map is covered using a single robot. The design of a coverage path in this case is done in the following steps. First, the environment is decomposed into smaller regions called cells to effectively cover the area. Each of these cells is then covered by choosing a sweeping direction. This direction may be chosen for example to reduce the number of turns or to traverse the cell in circular paths. The final step is to chose the order in which the cells are covered. This step is also known as the optimal backtracking mechanism. CPP methods with focus on decomposition methods is surveyed in [11], [12] and a recent one with focus on optimal backtracking algorithms and smoothing is studied in [13]. Our work focuses on the optimal backtracking techniques used during CCPP.

Once the region has been decomposed into cells, an adjacency graph is created where each node represents a cell and transitions denote a way to traverse between cells. The optimal backtracking sequence are found using methods involving greedy algorithms like depth first search (DFS) [14], Dijkstra's algorithm [15], A* algorithm, D* algorithm [16] and Theta* algorithm. Dynamic programming and evolutionary methods like the ant colony optimization mentioned before have also been used. Of all these algorithms, only the D* algorithm is equipped to handle dynamic environment constraints. An extension of the D* algorithm called the complete coverage D* (CCD*) was presented in [16]. This algorithm took into consideration the dimension of the robot for floor cleaning problem. The authors have also extended their CCD* algorithm for demining applications [8]. The algorithm presented in this paper applies an extension of the D* called the D* Lite algorithm [17] to the CCPP problem which efficiently handles dynamic environment constraints.

Many CCPP algorithms have been tested in a simulation environment. However, a robot covering an environment often employs control algorithms to follow the paths and has to localize itself on the map. One of the ways of ensuring better coverage and control is to plan the coverage paths so that the robot gets the best measurements to localize itself. For example, existing landmarks or beacons on the map might help the robot better understand where it is on the map. Incorporating this idea in the path planning stage itself has practical implications for CCPP problems.

The contributions of our work are summarized as follows. The CCD* Lite algorithm is presented which is an efficient implementation of the existing CCD* algorithm. Another contribution of our work is to weight the landmarks accordingly in the path planning stage to ensure better localization for path following. This can be done in the segmentation phase by ensuring that each cell has at least one landmark in it. This algorithm is implemented on the Aldebaran Nao

[1]Rahul Moghe is a PhD student in the Aerospace Engineering and Engineering Mechanics Department at the University of Texas at Austin, TX, USA. `rahul.moghe@utexas.edu`

[2]Meredith Pitschm is a PhD student in the Mechanical Engineering Department at the University of Texas at Austin, TX, USA. `pitschm1@utexas.edu`

[3]Tiffany Yang is a Masters student in the Computer Science Department at the University of Texas at Austin, TX, USA. `tiffanyyang@utexas.edu`

[4]Peter Stone is a Professor at the Computer Science Department at the University of Texas at Austin, TX, USA. `pstone@utexas.edu`

robot with a generated map and a set of beacons to help localize the robot.

II. METHODS

This section is divided into the following subsections; problem formulation, coverage algorithm, control algorithm, and evaluation methods.

A. Problem Formulation

The CCPP described in this paper uses the uniform grid-based decomposition of a known planar environment into cells which form the set $\mathcal{G}$. In the high-level planning phase, action set $\mathcal{A}$ consists of discrete actions, $\mathcal{A} = \{UP, DOWN, LEFT, RIGHT\}$. All the actions are deterministic, i.e., a performed action results in a unique transition that the agent will undergo. This results in a finite state machine: $(\mathcal{G}, \mathcal{A}, g_0, \delta, F)$, wherein $\mathcal{G}$ is the set of states of the system, $\mathcal{A}$ is the set of input actions, g_0 is the set of initial states, and $\delta : \mathcal{G} \times \mathcal{A} \longrightarrow \mathcal{G}$ denotes the state-transition function, and F are the (possibly empty) final accepting states. The environment also contains known and unknown obstacles. Hence, this paper presents an algorithm to cover an environment with given obstacles while robustly re-planning for unknown but static obstacles.

B. Coverage Algorithm

Many studies on CCPP focus on evaluating the algorithm in a simulated environment. A major challenge in coverage problems on a real platform is localization. In order to tackle this problem, landmark-based decomposition methods developed in the literature [18], [19], [20] use the slice decomposition method to decompose the environment. Since, we use a uniform grid decomposition, we employ the wavefront method derived in [21]. However, in our wavefront method, we increase the value of a cells closest to the edge using a wave factor so that the cells near the edges to have a higher priority. This is done to factor in the better localization that the edges have to offer. In this way, the path generated by the distance transform weighs the path to follow the edge of the environment so that the pose estimation has the least covariance.

The environment contains a set of known obstacles and unknown obstacles. In order to re-plan around the dynamic obstacles, a CCD* Lite algorithm is implemented. This, although similar to the CCD* algorithm presented in [16], is an efficient implementation of the Dynamic A* (D*) algorithm.

C. Control Algorithm

This subsection describes the control and estimation algorithms used on the robot. Although the actions $\mathcal{A}$ are discrete in the high-level planner, the states and measurements of the robot are continuous. The state of the robot consists of its global x and y position, and the global orientation θ, with respect to an inertial stationary frame. This section is divided into two subsections; the state estimation and the control input provided to the robot.

1) Estimation: Let the state of the system be $\boldsymbol{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$. A particle filter is used to estimate the state $\boldsymbol{x}$ of the robot. A 1000 particles are used to localize using 6 beacons and 6 colored markers kept on the L and T intersections of the soccer field. The standard deviation in the distance measurement was chosen to be linearly increasing with distance since beacons farther away had more noisy measurements. The resampling step was based only on weights of the particle.

The beacon detection was performed using heuristics on the segmented image. Run length encoding and Union find algorithms enabled us to find contiguous blobs of the same color in the image. In order to improve the beacon detection for tilted beacons appearing on the edges of the image frames the tilt of the beacon was calculated using the two blobs constituting the beacon. The x and y shift between the blobs was then used to calculate the angle of tilt of th blobs. This angle was then used to use an updated height of the beacon in the beacon detection so that more accurate estimates of the beacon distance and bearing could be achieved.

The markers were made up of post it notes kept one next to the other and of different colors. Heuristics were used to estimate the distance and bearing of the markers on the 6 locations of L and T intersections of white lines on the field.

2) Control: The control input to the robot was its x, y and θ velocities, namely, v_x, v_y, and v_θ in its own local frame. Hence, going from the local to global frame, the dynamics of the robot in a state space formulation can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_x cos\theta + v_y sin\theta \\ v_x sin\theta - v_y cos\theta \\ v_\theta \end{bmatrix} \tag{1}$$

This equation can be rewritten using simple algebra as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix} \tag{2}$$

The $R \in SO(3)$ matrix defined above is a rotation matrix which is a part of the special orthogonal group of order 3. Hence, this matrix is always invertible and has a determinant of 1 at all times. The current orientation of the robot is used to find the counter rotation matrix R^{-1} and used in the control law to find a stabilizing controller. We propose the following control law,

$$\begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix} = \underbrace{\begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R^{-1}} \begin{bmatrix} -K_x e_x \\ -K_y e_y \\ -K_\theta e_\theta \end{bmatrix} \tag{3}$$

wherein $e_x = x - x_{des}$ and $e_y = y - y_{des}$. The value e_θ is calculated based on the value of θ diagonally opposite the current θ. This decides the current direction that the robot should turn in. We set the $e_\theta = \theta_{des} - \theta$. Now, if $sgn(\theta_{des}) * e_\theta - \pi > 0$, then $e_\theta = e_\theta - 2sgn(\theta_{des})\pi$. This ensure a smooth control of theta to the desired value of θ.

Hence, the closed loop dynamics of the system in the global coordinates is given as

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} -K_x e_x \\ -K_y e_y \\ -K_\theta e_\theta \end{bmatrix} \tag{4}$$

where, it should be noted that $\dot{e}_x = \dot{x}$ and so on since x_{des} is constant for the phase where the robot goes from one cell to another. This controller is proved to be Uniformly exponentially stable using the Lyapunov function $V(\boldsymbol{x}(t)) = \frac{1}{2}(e_x^2 + e_y^2 + e_\theta^2)$. Differentiating the function we get,

$$V(\boldsymbol{x}(t)) = \frac{1}{2}(e_x^2 + e_y^2 + e_\theta^2) \tag{5}$$

$$\dot{V}(\boldsymbol{x}(t)) = e_x.(-K_x e_x) + e_y.(-K_y e_y) + e_\theta.(-K_\theta e_\theta) \tag{6}$$

$$\dot{V}(\boldsymbol{x}(t)) = -K_x e_x^2 + -K_y e_y^2 + -K_\theta e_\theta^2 \tag{7}$$

where, the derivative $\dot{V}(\boldsymbol{x}(t))$ of the Lyapunov function is negative definite and hence, the error in all the three states is guaranteed to converge exponentially with rates given by the P gains K_x, K_y, and K_θ. In addition to that,q to ensure a smooth transient response, we use a full PID control instead of just the P controller used above.

$$v_x = -K_{p,x} e_x - K_{i,x} \int_{t_i}^{t_f} e_x(t)dt - K_{d,x}\dot{e}_x \tag{8}$$

$$v_y = -K_{p,y} e_y - K_{i,y} \int_{t_i}^{t_f} e_y(t)dt - K_{d,y}\dot{e}_y \tag{9}$$

$$v_\theta = -K_{p,\theta} e_\theta - K_{i,\theta} \int_{t_i}^{t_f} e_\theta(t)dt - K_{d,\theta}\dot{e}_\theta \tag{10}$$

The PID gains were tuned on the robot by first tuning the P gain and setting the rest to 0. The I and D gains were then gradually added to improve the convergence and transient response mentioned above. This nonlinear controller is similar to a feedback linearization technique, the only difference being that instead of canceling out the nonlinear dynamics, the nonlinear part appears in the form of a rotation matrix and is canceled using a counter rotation matrix.

3) Uncertainty Analysis: Now let us assume that the robot doesn't have perfect measurements and the inverse rotation matrix is uncertain. Let us say that the $\theta_m = \theta + v$. Where v quantifies the uncertainty in the measurement and θ_m is the measured orientation of the robot. Hence, the new closed loop dynamics of the robot is given as,

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} cos(\theta-\theta_m) & sin(\theta_m-\theta) & 0 \\ sin(\theta-\theta_m) & cos(\theta-\theta_m) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -K_x e_x \\ -K_y e_y \\ -K_\theta e_\theta \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} cosv & sinv & 0 \\ -sinv & cosv & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -K_x e_x \\ -K_y e_y \\ -K_\theta e_\theta \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} \approx \begin{bmatrix} 1-\frac{v^2}{2} & v & 0 \\ -v & 1-\frac{v^2}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -K_x e_x \\ -K_y e_y \\ -K_\theta e_\theta \end{bmatrix} \tag{13}$$

wherein we used the small angle approximations to quantify the uncertainties in the measurements. We choose to ignore the v^2 terms assuming that v is small. This results in the occurrence of cross terms which affect other channels of control, for example

$$\dot{e}_x = -K_x e_x - K_y e_y v \tag{14}$$

$$\dot{e}_y = -K_y e_y + K_x e_x v \tag{15}$$

$$\dot{e}_\theta = -K_\theta e_\theta \tag{16}$$

and so on. Even if there is uncertainty in the measurement, this still ensures that the θ controller converges exponentially. Choosing the same Lyapunov function, we get

$$V(\boldsymbol{x}(t)) = \frac{1}{2}(e_x^2 + e_y^2 + e_\theta^2) \tag{17}$$

$$\begin{aligned} \dot{V}(\boldsymbol{x}(t)) &= e_x.(-K_x e_x - K_y e_y v) \\ &\quad + e_y.(-K_y e_y + K_x e_x v) + e_\theta.(-K_\theta e_\theta) \end{aligned} \tag{18}$$

Note that if we choose $K_x = K_y$, the cross terms cancel and we get a negative definite Lyapunov function which ensures uniform exponential convergence for small angle deviation. Hence, we used equal gains for the x and y channels.

D. Evaluation Methods

The algorithm was evaluated on the Nao Robot and the soccer field as the environment. The 6 beacons and 6 markers placed on the field helped with the localization. Orange colored cut outs of the size of a grid cell were used as obstacles. There were two obstacles on the field whose positions were known beforehand. The size of the obstacles was two grid cells. We used one or two unknown obstacles of size of either one or two grid cells. The robot starts at one of the corners of the field to cover the area. The robot has some initial idea about the map, however, new obstacles are introduced to change the map. The robot is expected to re-plan its path when the obstacle is detected to successfully cover the map. Figs. (1) and (2) shows the top view of the environment along with the grid cells which the robot is trying to cover.

III. RESULTS

The coverage algorithm presented in this paper was run on the Nao Robot. The entire algorithm could not be run in one go because of the temperature of the joints. Hence,

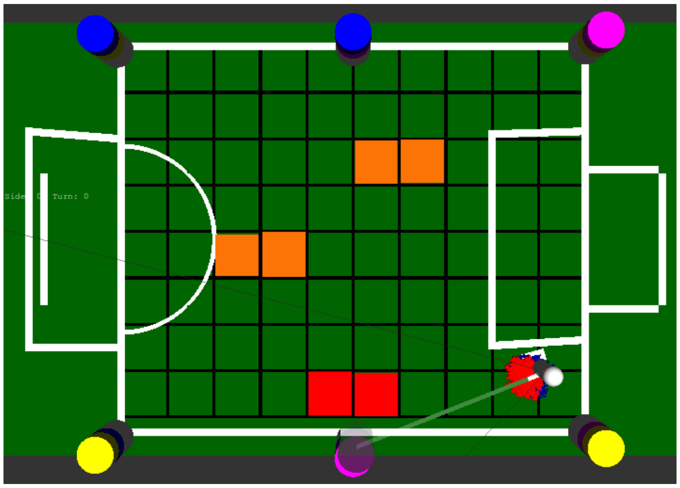

Fig. 1. The top view of the environment with the grid cells and beacon locations and the landmarks made up of L and T intersections on the soccer field. The orange grids represent the known obstacles locations and the red grid cells denote the unknown obstacles on the map. The 1000 particles of the Particle Filter are shown by blue arrows with red heads. The estimate of the position of the robot is shown in white.

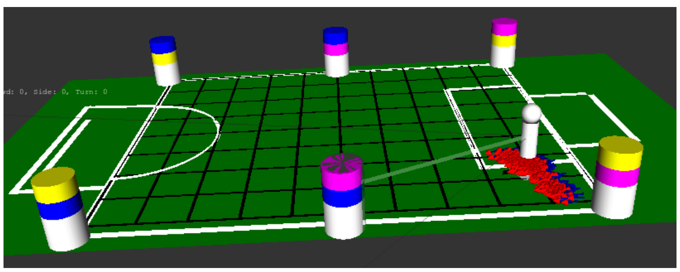

Fig. 2. The side view of the same map given above which shows the beacon colors.

the algorithm was run in parts and the robot joints were rested when their temperature reached above 85 units as calculated by the code base. It was very difficult to compare this algorithm with other existing ones as the entire coverage took a lot of time to complete with many failures in between leading to restarting the entire coverage from the start. We were able to run about two thirds of a path with two rest periods and the robot avoided all the obstacles and re-planned around the obstacles.

ACKNOWLEDGMENT

I was in charge of control, state estimation, tuning of the parameters and running the algorithm on the robot. The derivation of the control law and the particle filter was done by the author. Countless hours of tuning the filter and PID parameters was performed on the robot. Meredith's research and thesis are on coverage and hence, she worked on the wavefront generation and path generation and Tiffany was the software guru and she worked on the D* Lite and everything related to tweaking the code base to match our needs.

REFERENCES

[1] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 801–812, 2011.

[2] H. H. Viet, V.-H. Dang, M. N. U. Laskar, and T. Chung, "Ba*: an online complete coverage algorithm for cleaning robots," *Applied intelligence*, vol. 39, no. 2, pp. 217–235, 2013.

[3] F. Yasutomi, M. Yamada, and K. Tsukamoto, "Cleaning robot control," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*. IEEE, 1988, pp. 1839–1841.

[4] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, "Uniform coverage of automotive surface patches," *The International Journal of Robotics Research*, vol. 24, no. 11, pp. 883–898, 2005.

[5] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an auv," in *Underwater Robots*. Springer, 1996, pp. 17–45.

[6] M. Morin, I. Abi-Zeid, Y. Petillot, and C.-G. Quimper, "A hybrid algorithm for coverage path planning with imperfect sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5988–5993.

[7] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *The International journal of robotics research*, vol. 22, no. 7-8, pp. 441–466, 2003.

[8] M. Dakulovic and I. Petrovic, "Complete coverage path planning of mobile robots for humanitarian demining," *Industrial Robot: An International Journal*, vol. 39, no. 5, pp. 484–493, 2012.

[9] J. Jin and L. Tang, "Optimal coverage path planning for arable farming on 2d surfaces," *Transactions of the ASABE*, vol. 53, no. 1, pp. 283–295, 2010.

[10] B. Englot and F. S. Hover, "Sampling-based coverage path planning for inspection of complex structures." in *Icaps*, 2012.

[11] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[12] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[13] A. Khan, I. Noreen, and Z. Habib, "On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges." *J. Inf. Sci. Eng.*, vol. 33, no. 1, pp. 101–121, 2017.

[14] G. Zuo, P. Zhang, and J. Qiao, "Path planning algorithm based on subregion for agricultural robot," in *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, vol. 2. IEEE, 2010, pp. 197–200.

[15] M. Waanders, "Coverage path planning for mobile cleaning robots," in *15th Twente Student Conference on IT, The Netherlands. Copyright*, 2011.

[16] M. Dakulovic, S. Horvatic, and I. Petrovic, "Complete coverage d* algorithm for path planning of a floor-cleaning mobile robot," in *18th international federation of automatic control (IFAC) world congress*, 2011, pp. 5950–5955.

[17] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[18] S. C. Wong and B. A. MacDonald, "A topological coverage algorithm for mobile robots," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2003, pp. 1685–1690.

[19] ——, "Complete coverage by mobile robots using slice decomposition based on natural landmarks," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2004, pp. 683–692.

[20] S. Wong, "Qualitative topological coverage of unknown environments by mobile robots," Ph.D. dissertation, ResearchSpace@ Auckland, 2006.

[21] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proceedings of international conference on advanced robotics*, vol. 13, 1993, pp. 533–538.