Proceedings of the 2018 IEEE/ASME International
Conference on Advanced Intelligent Mechatronics (AIM),
Auckland, New Zealand, July 9-12, 2018

Active Planning of Robot Navigation for 3D Scene Exploration

Wenzhou Chen[1], Yong Liu[1]

***Abstract*— This work addresses the active planning of robot navigation tasks for 3D scene exploration. 3D scene exploration is an old and difficult task in robotics. In this paper, we present a strategy to guide a mobile autonomous robot equipped with a camera in order to autonomously explore the unknown 3D scene. By merging the particle filter into 3D scene exploration, we address the robot navigation problem in a heuristic way, and generate a sequence of camera poses to coverage the unknown 3D scene. First, we randomly generate a bunch of potential camera pose vectors. Then, we select the vectors through our criteria. After determining the first camera pose vector, we generate the next group of vectors based on the former one. We select the new camera pose vector and thereafter. We verify the algorithm theoretically and show the good performance in the simulation environment.**

I. Introduction

The research of mobile robot navigation has recently been developed as a focus within the mobile robotics community. 3D scene exploration is important in many practical applications, such as surveillance [1], prospection [2] and active vision [3].

Most navigation algorithms address the 3D scene exploration problem by simultaneous localization and mapping(SLAM) [4]–[6]. This kind of method can get the scene map and the robot pose through the vision or lidar sensor data. One of the famous SLAM algotithm is ORB-SLAM [4] which address the localiztion and mapping task based on the feature points and optimization method. There is no doubt that the SLAM algorithm performs well in mapping the unknown 3D scene. But that is't an efficient way. The sparse feature based SLAM algorithm [4], [6] mainly focus on getting accurate pose of robots on the 3D scene and their map is also sparse. The algorithms [5] which build dense map for 3D scene always accompanied with the high computational cost, while it is not necessary in the 3D scene exploration problem.

In this paper, we consider the task of guiding a mobile autonomous robot carrying a vision sensor, in order to efficently explore the unknown 3D scene. We present a exploration algorithm to generate the camera poses which would optimal covered the exploration environment. The basic idea of our strategy is quite familiar with the view planning problem(VPP) [7]–[9], both of the task want to get the optimal view point to cover the target object or environment. The view planning algorithm is well studied and the paper [9] address the VPP with reinforcement learning also get good results.

However, the view planning problems always assume the 3-D CAD model is available before view planning. On the contrary, that is impossible in scene exploration tasks. We can only get a weak prior of unknown environment by the robot observation.Inspired by the color-based particle filter [10] in tracking task, we present a heuristic algorithm based on the particle filter frame and the coverage criteria. We use the prior information to initial our strategy and start the heuristic algorithm to get the local optimal navigation pose sequence one by one.

The remainder of this paper is organized as follows. In Section II, we introduce the necessary preliminary background and define the basic notation. The visual coverage criterias and constrains are given in the Section III. The overall framework of the navigation strategy is described in Section IV. Simulation results and the strategy performance are presented in Section V and the paper is concluded in Section VI.

II. Problem Formulation and Basic Notations

The problem we considered in this paper is the robot navigation strategy for 3D scene exploration, which is usually encoutered in practical tasks. For the specified problem, it is assumed that the robot don't know the exact model of the environment. The robot need to explore the environment by using the sensor observation. Here we use a RGB-D camera as the only sensor on our robot.

A. Camera Model

We assume the mobile robot explore the unknown environment with a fixed RGB-D camera. And the mobile robot explore the environment in a plane and the camera is fixed with the mobile robot without relative movement. We simply regared our RGB-D camera as the combination of a pinhole camera model and a depth sensor model. Generally, practical camera models are always highly nonlinear, and the pinhole camera model which we adopted may lead some modeling errors. However, We mainly focus on the exploration strategy and the coverage performance in this work, the pinhole camera model is more efficient and convenient in our tasks.

We use $f \in \mathbb{R}$ to express the focal length(mm), and $o = (u_0, v_0)$ represents the principle point in pixels. In addition, we denote the image size $w \times h$(pixel $\times$ pixel) to represent the width and height of the image in pixel where $w \in \mathbb{Z}^+$ and $h \in \mathbb{Z}^+$.

[1]Wenzhou Chen is with the Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China.

[1]Yong Liu is with the State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China (Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn).

B. Coordinate System Description

The object we consider here is a mobile robot equipped with a fixed RGB-D camera. As shown in Fig. 1, the x_c- and y_c-axis belongs to the camera coordinate system *C*. We build the robot coordinate system at the origin of the orthogonal coordinate system *C* and let them coincide with another. We use the camera coordinate system *C* to represent the robot coordinate system for the camera is fixed with the robot. The x_w-axis and y_w-axis represent the world coordinate system. The z_w-axis of world coordinate system is paralleled with the z_c-axis, and the z_c-axis of *C* is perpendicular to the plane of motion. Thus, the constant translation value of camera in z_w-axis can be ignored when we focus on the robot exploration task. We can simplify our robot navigation problem from 3D scene to the 2D plane and define the motion constrain to accomplish the simplification.

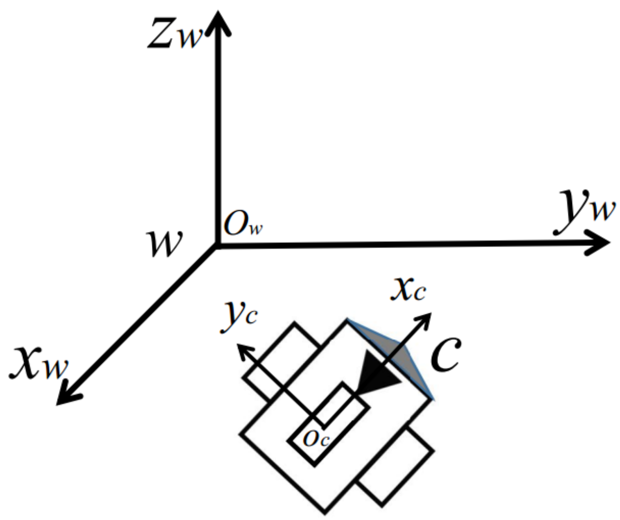

Fig. 1. Coordinate relationship between the mobile robot with fixed camera and the 3D scene.

C. Camera Field of View

As shown in Fig. 2 , the camera has a field of view with rectangular pyramid shape. In our 3D scene exploration task, we need to ensure that our exploration strategy could get the valuable robot navigation poses with the high quality images of the unknown 3D scene. Because it lays a good foundation for later extensions, such as 3D scene reconstruction [11]–[13]. So we define the length value d as the valid performance distance for the camera has a fixed focal length. We assume the camera performs well near a small range of that distance value.

III. Coverage Criteria and Overall Scheme

Based on the aforementioned motion constrain, we reformulate our task in the 2D plane. We use pose vector $P = [x, y, \theta]$ to represent the pose of the camera fixed in the mobile robot by default in the rest parts of our paper. The (x, y) term determine the position of the vector and the θ determine the rotation of the pose vectors. In this section, we formulate the coverage criteria and introduce it in details.

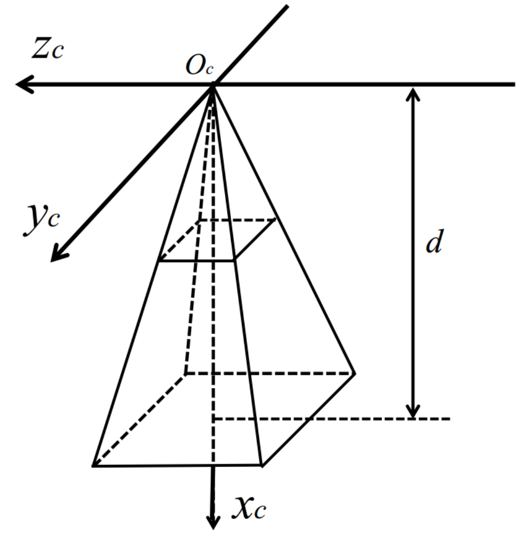

Fig. 2. Field of view and the relative parameters.

A. Depth Error and FOV Constrain

The rectangular pyramid shape camera field of view degrade to an triangular in the 2D plane because of the motion constrain.As shown in the Fig.3, the valuable range of our camera model is defined as $[d + \delta, d - \delta]$ for convenience. The δ value is a parameter that determined by the real camera when we use in practice. We get the depth image for each camera pose from the RGB-D camera, and preprocess the depth image through a bialteral filter [14]. It is assumed that the gray level on depth image is linear correlated with depth value in the 3D scene. So we can get an depth value range $[D + \Delta, D - \Delta]$ correspond with the distance range $[d + \delta, d - \delta]$. By summing up each column of the depth image, the distance between camera and the unknown environment is defined as the FOV constrain. To make use of the depth image, we define the depth image as I and formulate the $Des(x)$ as the depth image array in horizontal. The $error$ function evaluate the relative rotation between the camera and the environment.

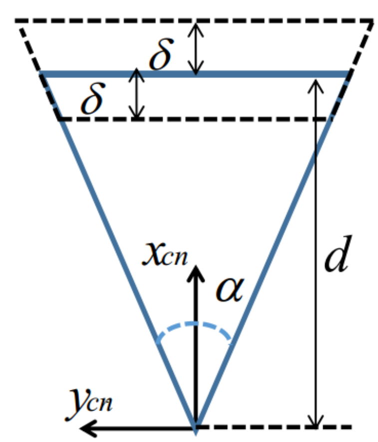

Fig. 3. Field of view and the relative parameters.

$$Des(x) = \sum_{y=0}^{h} I(x,y) \tag{1}$$

$$error(I) = \sum_{x=0}^{\frac{w}{2}} \sum_{y=0}^{h} I(x,y) - \sum_{x=\frac{w}{2}}^{w} \sum_{y=0}^{h} I(x,y) \tag{2}$$

B. Overall Scheme

With the aforementioned constrains and models in mind, we describe the overall scheme for the proposed robot exploration strategy in this section.

The question that mainly concerned in this work is exploring the unknown environment efficiently. Inspired by the visual coverage problem solutions [7], [15], [16], we transfer our exploration problem into a 3D scene coverage problem. In order to generate the essential camera poses which can cover the 3D scene in a effective way, we use these poses to guide the robot explore in the 3D scene. Now our exploration problem could be written as following optimization problem:

$$\min \quad \mathbf{n} \tag{3}$$

$$s.t. \quad Area = f(p_1,p_2) \cup f(p_2,p_3) \cup ... \cup f(p_{n-1},p_n) \tag{4}$$

The function $f(p_i, p_{i+1}), (i = 1,2,...,n)$ represent the coverage area summation for each adjacent pose. The $Area$ is the area of the whole 3D scene, and the n represent the total number of camera poses.

We could explain the optimization formulation more clearly. The purpose of our task is getting the minimal number of camera poses when the camera field of views can cover the unknown 3D scene with a proper pose. It is hard to compute the minimal number directly without any conditions and constrains. We rewrite our optimization problem to a much more solvable expression.

$$\max \quad \sum_{n=1}^{N} \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2} \tag{5}$$

$$\begin{cases} f(p_{n-1},p_n) \geq \sigma \\ Area = f(p_1,p_2) \cup f(p_2,p_3) \cup ... \cup f(p_{n-1},p_n) \end{cases} \tag{6}$$

The σ is defined as a area threshold to represent the mimimal overlap area between two adjacent poses. Because we need the overlap area to address the subsequent tasks such as sparse mapping.

IV. Particle-Based Pose Determination

On the basis of the previous assumption and definition, we combine the robot navigation problem with the visual coverage scheme proposed in [15] to build a heuristic navigation strategy by using the particle-based pose determination algorithm.

Our paper present an algorithm to determine the optimal poses in a heuristic way. As shown in Fig.4, we use aforementioned coverage criterion and build our algorithm based on the particle filter frame [10]. The X value represent the pose vector sets of P in our algorithm. We initial the position value by two-dimentional Gaussian distribution in the 2-D plane. And the rotation value θ is initiated by the uniform distribution in the range of $[0, 2\pi]$.

By following the particle-based pose determination algorithm, we can determing one optimal coverage pose according to the depth error and the field of view constrain. That pose is the seed of our strategy. We generate the next group of pose vectors beside the first pose vector. The mean location value of the next pose vector group is in the line which is parallel to the first local optimal pose vector. The (x, y) of the new poses is Gaussian distributed. And the θ term is Guassian distributed in the range of $[0, 2\pi]$ with their mean coordinated with the θ term of last local optimal pose. The distance value between the mean (x, y) term of the new poses and the old optimal pose (x, y) term is $d \times \tan\left(\frac{w}{2}\right)$. By recurrently running our particle-based pose determination algorithm, we can get the optimal poses one after one. However, it is only the local optimal pose for our exploration. To get the global pose value, we can simply change the distance term to a sequence of parameters and solve it through equation 5 and 6.

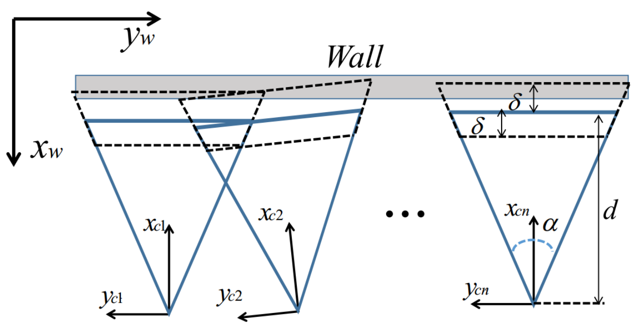

Fig. 4. Coverage iteration.

Algorithm 1 Particle-Based Single Pose Determination

Random **initial** the N pose vectors X by the prior Env and the first observation Z, initial the probability weight $\pi_0^{(n)}$ for $X_0^{(n)}$,$n \in \{0,1,2,...,n\}$
$t \leftarrow 0$
$FOV\_flag \leftarrow 1$
while FOV_flag **do**
    X_t = Resample(X_t, π_t)
    π_t = WeightUpdate(π_t, X_t, I)
    FOV_flag = Evaluate(X_t, π_t)
    $t \leftarrow t + 1$
end while
return P
function RESAMPLE(X_t, π_t)
  Compute the CDF of the weight and resample it. (a)
  1) calculate the normalized cumulative probabilities c_t'
  $c_t^{(0)} = 0$
  $c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)}$

$c_t^{'(n)} = \frac{c_t^{(n)}}{c_t^{(N)}}$

2) Generate a uniformly distributed random number $r \in [0, 1]$
3) Use binary search to find the smallest j, we get a N-d array j for which $c_t^{j[i]} \geq r[i]$
4) Update the particle set,$X_t^i = X_t^{j[i]}$

end function
function WEIGHTUPDATE(π_t, X_t, I_t)
  Update the weight by the scorefun and normalize the weight.

$w_t = ScoreFun(X_t, \pi_t, I_t)$
$\pi_t^{(i)} = \pi_t^{(i)} w_t^{(i)}$
$\pi_t^{(N)} = \sum_{i=1}^{N} \pi_t^{(i)}$
$\pi_t = \frac{\pi_t^{(i)}}{\pi_t^{(N)}}$

  return π_t
end function
function SCOREFUN(X_t, π_t, I_t)
  Init N-d array $error$ and w_t
  $error[i] = error(I_t^{(i)})$
  $w_t^{(i)} = \frac{\pi_t^{(i)}}{error[i]}$
  $w_t^{(N)} = \sum_{i=1}^{N} w_t^{(i)}$
  $w_t = \frac{w_t^{(i)}}{w_t^{(N)}}$
  return w_t
end function
function EVALUATE(X_t, π_t)
  Select the i in $max(\pi_t^{(i)})$
  if $((D - \Delta) < Des(\frac{w}{2}) < (D + \Delta))$
   and $(|error(I)| < 2\Delta)$ **then** $P = X_t^{(i)}$
    return 0
  else
    return 1
  end if
end function

V. SIMULATION AND EXPERIMENTAL RESULTS

As shown in the Fig.6, we build the simulation 3D scene in the blender software environment. The scene is a room with some furneture in it. The purple wall is the given 3D scene which need to be covered by our camera FOVs. The camera pose shown in this figure is the first local optimal pose of our iteration algorithm. By following our particle-based pose determination algorithm, new pose vector sets would be generated and selected with our coverage constrains. And we can generate the camera pose one after one. Fig.5 illustrate the performance of our scheme in blender environment for the outstanding coverage rendering. The FOV of each local optimal camera pose is rendered into rectangular pyramid. The overlap areas of FOVs are rendered into the light color. And the darker area are the FOVs of particular camera poses. The picture clearly shows a sequence of pose iteration results as an example.

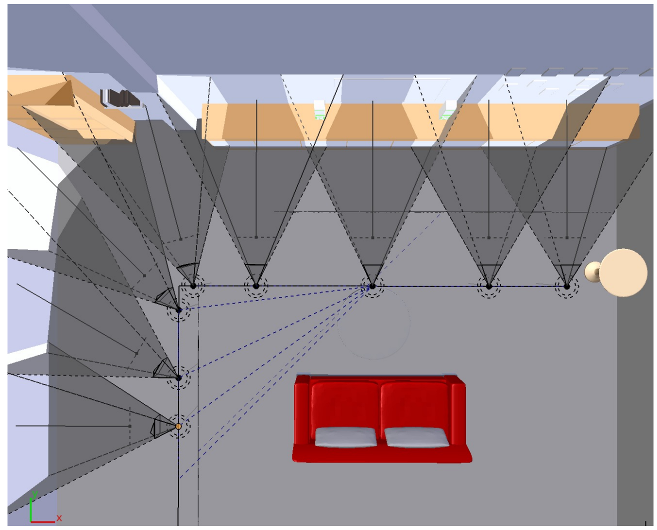

Fig. 5. Coverage strategy illustration.

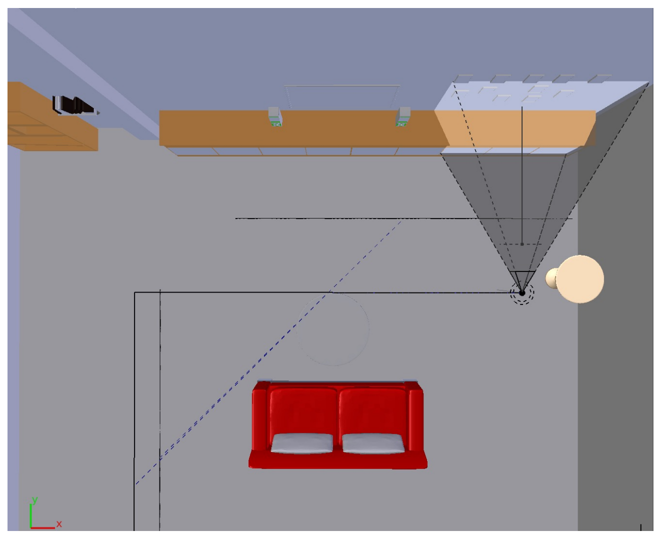

Fig. 6. 3D scene illustration.

To verify the convergence property of our particle-based pose determination algorithm, we build our algorithm in the simulation environment. By generating the specific 3D scene and randomly initial our program, our pose determination algorithm can converge in a good result with the given scene well covered.

VI. CONCLUSION

A novelty robot navigation strategy in an unknown 3D scene is presented in this paper. An exploration approach for covering the unknown environment is implemented by heuristically iterating the particle-based pose determination algorithm. Potential robot poses are generated one after one, with the global optimal coverage problem being novelly degraded as a sequence of more solvable local optimization tasks. As a result, the poses for covering the unknown environment are obtained. We could regard these poses as the key frame and control the robot to efficientlly explore the unknown 3D scene by arriving at the poses.

REFERENCES

[1] M. Masár, "A biologically inspired swarm robot coordination algorithm for exploration and surveillance," in *Intelligent Engineering Systems (INES), 2013 IEEE 17th International Conference on*, pp. 271–275, IEEE, 2013.

[2] G. Ferri, M. V. Jakuba, and D. R. Yoerger, "A novel trigger-based method for hydrothermal vents prospecting using an autonomous underwater robot," *Autonomous Robots*, vol. 29, no. 1, pp. 67–83, 2010.

[3] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," Robotics: Science and Systems, 2015.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, IEEE, 2014.

[7] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys (CSUR)*, vol. 35, no. 1, pp. 64–96, 2003.

[8] F.-M. De Rainville, J.-P. Mercier, C. Gagné, P. Giguere, and D. Laurendeau, "Multisensor placement in 3d environments via visibility estimation and derivative-free optimization," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 3327–3334, IEEE, 2015.

[9] M. D. Kaba, M. G. Uzunbas, and S. N. Lim, "A reinforcement learning approach to the view planning problem," in *Conf. Comput. Vis. Pattern Recognit*, pp. 5094–5102, 2017.

[10] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.

[11] X. Liu, Y. Zhao, and S.-C. Zhu, "Single-view 3d scene reconstruction and parsing by attribute grammar," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 710–725, 2018.

[12] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, ACM, 2011.

[13] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, "Uav-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–10, 2016.

[14] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, pp. 839–846, IEEE, 1998.

[15] X. Zhang, X. Chen, J. L. Alarcon-Herrera, and Y. Fang, "3-d model-based multi-camera deployment: a recursive convex optimization approach," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3157–3169, 2015.

[16] Y. Morsly, N. Aouf, M. S. Djouadi, and M. Richardson, "Particle swarm optimization inspired probability algorithm for optimal camera network placement," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1402–1412, 2012.

978-1-5386-1854-7/18/$31.00 ©2018 IEEE