2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), 16-19 October 2016, Abu Dhabi, UAE

GPU Accelerated Coverage Path Planning Optimized for Accuracy in Robotic Inspection Applications

Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, Jorge Dias, Guowei Cai
Khalifa University Robotic Institute
Khalifa University, Abu-Dhabi, U.A.E
Email: {randa.almadhoun,tarek.taha,lakmal.seneviratne,jorge.dias,guowei.cai}@kustar.ac.ae

***Abstract*—In this paper, we introduce a coverage path planning algorithm for inspecting large structures optimized to generate highly accurate 3D models. Robotic inspection of structures such as aircrafts, bridges and buildings, is considered a critical task since missing any detail could affect the performance and integrity of the structures. Additionally, it is a time and resource intensive task that should be performed as efficiently and accurately as possible. The method we propose is a model based coverage path planning approach that generates an optimized path that passes through a set of admissible waypoints to cover a complex structure. The coverage path planning algorithm is developed with a heuristic reward function that exploits our knowledge of the structure mesh model, and the UAV's onboard sensors' models to generate optimal paths that maximizes coverage and accuracy, and minimizes distance travelled. Moreover, we accelerated critical components of the algorithm utilizing the Graphics Processing Unit (GPU) parallel architecture. A set of experiments were conducted in a simulated environment to test the validity of the proposed algorithm.**

I. Introduction

The demand for robotic inspection has been growing recently in different sectors such as aviation, marine, and construction sectors. Different types of robotic systems are employed in the application of robotic inspection including: Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), and maritime robots, all of which could be utilized for various inspection operations. Structures inspection is particularly important in applications that require maintenance, fault traceability, anomaly and defects detection, and model digitizing. UAVs are one of the robotic systems which simplifies the reconstruction process that is essential for inspection by providing the flexibility of capturing visual information of the structure regions that are hard to reach.

Technically, various robotic capabilities are required to preform inspection such as: localization in the environment where the structure exists; path planning and navigation in order to compute a set of achievable routes; sensing and perception in order to gather information about the structure from different viewpoints along the route. As such, the robot should be equipped with intelligent sensing capabilities that facilitates gathering information with high quality in order to reconstruct the structure of interest accurately for inspection. Different structure inspection algorithms have been documented in the literature addressing various challenges as comprehensively reviewed in [1].

Due to application criticality (inspection), we propose a coverage path planning algorithm that explicitly targets both the model coverage, and accuracy requirements during aircraft inspection applications. The proposed algorithm utilizes an existing mesh model of the complex structure and the models of the sensors mounted on the UAV. It consists of three main components: viewpoints generation, path planning, and coverage evaluation. Our algorithm integrates the sensors Field of View (FOV), range limitations, and measurement errors in order to generate an optimized path encapsulating viewpoints that achieves the maximum coverage. The new algorithm is developed with a heuristic reward function that preforms viewpoints selection based on the distance, turning angle, coverage percentage and sensor accuracy. In addition to this, the speed of the proposed algorithm is accelerated utilizing the GPU parallel architecture. Finally coverage evaluation is evaluated based on different criteria including the achieved coverage percentage, and the produced model resolution through these set of viewpoints. The proposed algorithm is resolution complete [2] which provides efficient 3D reconstruction of complex shapes with certain coverage guarantees, in addition to an estimation of resolution accuracy with a resolution driven reward function.

We will address each of the main components of the algorithm as follows: viewpoints generation in section II, path computation and reward estimation in section III, and coverage evaluation in section IV. The simulated experiments used to verify the proposed algorithm are presented in section VI. Finally, conclusions and future enhancements are given in section VII.

II. Viewpoints Generation

The main aim of the viewpoints generation process is to generate a set of viewpoints from which the structure of interest is visible. Our viewpoints generation method starts by preforming position based discretization using a specific grid resolution which generates a set of sample UAV waypoints defining an xyz position in 3D space. Next, a set of yaw angle orientations are generated by performing an orientation based discretization on each position sample. The generated set of sample waypoints are represented by xyz coordinates and ψ yaw angle. Each sensor viewpoint is then generated by applying a 4×4 transformation matrix that defines the sensor location with respect to the UAV body frame.

The generated set of waypoints and viewpoints are then filtered using collision, distance and coverage based filtering approach. Collision based filtering eliminates the sample viewpoints that are inside the model or collides with the model. Distance based filtering keeps the sample viewpoints available within a defined distance from the structure model determined based on the sensor's minimum and maximum effective range.

978-1-5090-0916-9/16/$31.00 ©2016 IEEE

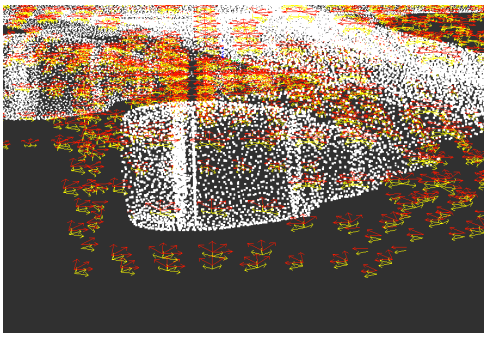


Fig. 1: Visualization of a set of filtered waypoints in red and the corresponding sensor viewpoints in yellow applying 1.5m position based resolution and $pi/4$ orientation based resolution

To eliminate viewpoints that provides no coverage, the visible surface at each viewpoint is extracted by performing frustum culling and occlusion culling. Frustum culling extracts the structure part that lies inside the sensor FOV frustum, and occlusion culling extracts the visible surface from the extracted frustum. Figure 1 shows the final filtered sensor viewpoints.

III. Path Computation and Reward Estimation

Coverage path planning represents the next significant component. At this stage, the previous component generated a discretized sample space which consists of a set of filtered sample waypoints W, and a corresponding sensor viewpoints V. The discretized sample space is then used to generate the search space by graphically connecting samples with their neighbors based on a pre-defined connection radius r. Then the path planning is performed using a modified A^* [3] algorithm, with a heuristic reward function to search this search space for an optimised path that achieves the desired coverage percentage.

The proposed heuristic R targets the point cloud accuracy which is calculated based on the used sensor model. It minimizes travel distance δd and turning angle δa, and maximizes the coverage C and the point cloud average accuracy $\bar{\sigma_z}$ computed at each step. The reward function is defined in equation 1, the first term is proportional to the distance travelled, and the average accuracy $\bar{\sigma_z}$ of the visible point cloud encapsulated in the FOV at the viewpoint. The average accuracy $\bar{\sigma_z}$ of the point cloud is evaluated by computing the standard deviation of error in depth at each point in the point cloud and taking the average of the resulting values. When the next waypoint involves only a rotation, then the reward is proportional to δa and $\bar{\sigma_z}$ as shown in the second part of the equation.

$$R = \begin{cases} \frac{1}{\delta d} \times (1 - \frac{\bar{\sigma_z}}{\max \sigma_z}) \times C, & \text{if } \delta d > 0 \\ (1 - \frac{\delta a}{2\pi}) \times (1 - \frac{\bar{\sigma_z}}{\max \sigma_z}) \times C, & \text{if } \delta d = 0 \end{cases} \quad (1)$$

IV. Coverage Evaluation

Coverage evaluation component was presented in literature as a critical part of the planning algorithm iterations, and as a significant performance criteria [4]. In this component, we evaluate the completeness of the generated coverage path and the accuracy across this path.

An assessment of the coverage path planning completeness is performed by quantifying the percentage of the covered volume of the structure compared to the predicted 3D structure volume across the generated path. The covered volume represents the actual volume of the surface reconstructed by following the trajectory produced by the coverage path planning algorithm, and collecting data along the path. The predicted volume, however, is measured by performing frustum culling on the reference model at each trajectory waypoint, and accumulating the volume along the trajectory. Each of the covered and the original volumes of the structure is represented by a grid of voxels. These voxel grids are then used to calculate the coverage percentage as described in 2.

$$Coverage\ \% = \frac{Covered\ Volume_{\ Voxel\ Grid}}{Original\ Volume_{\ Voxel\ Grid}} \times 100 \quad (2)$$

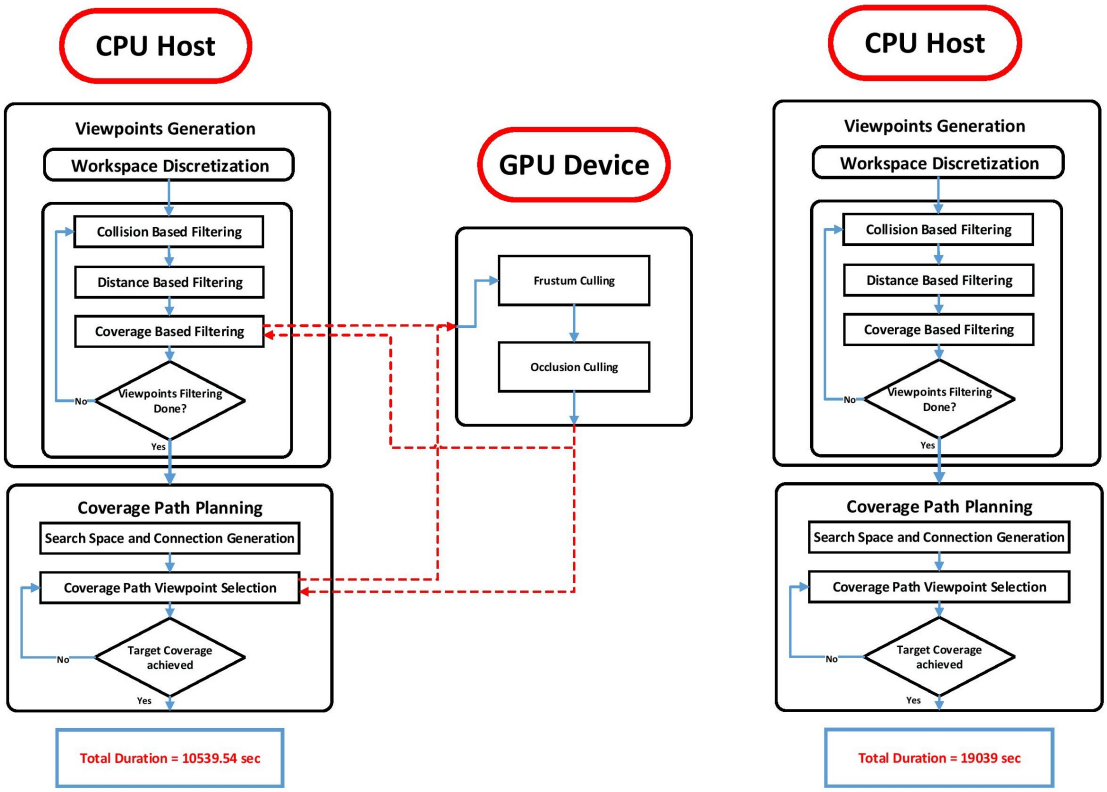
The accuracy of the model generated is evaluated based on the used sensor model by following the generated coverage path. The evaluation of accuracy helps in identifying and improving the regions with the lowest accuracy in order to provide highly accurate 3D reconstruction.

V. Algorithm Optimization

Recently, GPUs are becoming popular as general computing devices due to their massively parallel architectures. Nvidia GPUs can be accessed through a Compute Unified Device Architecture (CUDA) framework API, which is a parallel computing platform running on NVIDIA GPUs [5]. CUDA builds and runs CUDA C codes which are programmed using extended C/C++ language that supports the communication between CPU and GPU.

The proposed coverage path planning include two computationally intensive and time consuming processes which are frustum culling and occlusion culling. These two processes are considered time consuming since they are used iteratively in the coverage path planning algorithm to find a critical part of the heuristic reward function which is the extra coverage. In addition to this, they process 3D point cloud data which is considered computationally expensive. Both processes are optimized by preforming parallel implementation using Nvidia CUDA framework. Preforming algorithmic parallelization consists of dividing the intensive tasks into small tasks that can run in parallel on the available GPU resources instead of executing them sequentially on a CPU. The diagram shown in Figure 2 illustrates the general architectural difference between the proposed algorithm in it's parallelize and non-parallelized form.

A set of experiments were performed using the optimized coverage path planning algorithm and the original algorithm. A set of target coverage percentages were applied and the coverage path planning duration result of each was recorded in order to observe the difference between the optimized algorithm performance and the original algorithm performance. In order to test the performance of the optimized proposed algorithm in CUDA, we selected a GPU Nvidia Quadro K4200 and a CPU Intel(R) Xeon(R) E5-2650 v3. The GPU chip include 1344 CUDA cores in total and the Memory clock rate is 2700 MHz. The used CUDA Toolkit 7.5 contains the latest compiler (nvcc) which includes libraries, and code samples.



(a) The performance using CPU host and GPU device

(b) The performance using CPU host

Fig. 2: A General overview of the algorithm performance using a combination of the CPU and GPU and using the CPU only

The results of these experiments are shown in Figure 3 which shows the performance increase of the algorithm after the optimization.

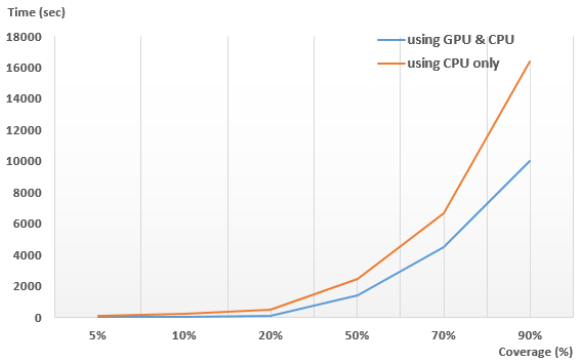


Fig. 3: Results of applying different set of coverages using the algorithm and the optimized algorithm using CUDA. The results show that effect of using parallel implementation on the algorithm performance which highly enhanced

VI. Experiments and Results

A set of experiments were performed in a realistic robot simulator Gazebo [6] in order to evaluate the proposed algorithm. A A340 aircraft mesh model was used to represent a complex aircraft structure that contains 32496 triangular faces. Software-in-the-loop (SITL) simulated experiments were conducted using an Iris quadrotor platform equipped with RGBD sensor. The main critical parameters of the algorithm include the sensor FOV, grid resolution, sensor range, target coverage percentage, and tolerance to the target coverage percentage. An experiment was designed to check the predicted coverage percentage and compare it with the real coverage achieved. In this experiment, the quadrotor is assumed to carry a RGBD sensor mounted at $5.5°$ pitch and $6cm$ under the quadrotor center with a FOV of [58H,45V]° and a maximum depth range of $7m$. The proposed approach was used with a grid resolution of $1.5m$, a sensor distance of (1 to 4)m from the model, a connection radius of $2.5m$, a set of different target coverage percentages and a target tolerance of 1%.

The same experiment was repeated with various target coverage percentages, and their results were evaluated based on metrics that include: the path distance, the search duration, the number of selected viewpoints, the average extra coverage and accuracy per viewpoint, the target coverage percentage and the end coverage percentage. The results of the experiments are summarized in Table I. As evident by our results, our method generated feasible path trajectories that consist of 713 waypoints to achieve the desired 90% coverage percentage. It's worth mentioning that the maximum achievable coverage percentage is 96.22% for this particular setup due to the sensor mounting position, proximity of the aircraft to the ground, and the fact that only stable horizontal hovering of the quadrotor was considered.

Furthermore, the accuracy of the 3D constructed model

TABLE I: Summary of experiments results Applied on A340 aircraft model using heuristic R using the CPU and GPU architectures

Grid resolution = 1.5m, Effective sensor range = 1-4m, Connection radius = 2.5m, Tolerance to coverage percentage = 1%						
Target Coverage	**Path Distance**	**Search Duration**	**Number of selected viewpoints**	**Average extra coverage per viewpoint**	**Average accuracy per viewpoint**	**End Coverage**
20%	86.335m	154.78s	57	0.1062%	6.7249mm	19.4866%
50%	288.445m	1415.33s	185	0.0801%	9.6747mm	49.0591%
70%	610.422m	4525.53s	386	0.0523%	7.8009mm	69.0573%
90%	1185.44m	10481.22s	713	0.0392%	9.005mm	89.177%

was evaluated by computing the standard deviation of error in depth at each point in the point cloud at each viewpoint in the generated coverage path following the equation 2 presented in [7]. The values of $\frac{m}{f_b}$ and σ_d were computed in [7] as 2.85×10^{-5} and $\frac{1}{2}$ pixel by calibrating a Kinect RGBD sensor. Figure 4 illustrates the results of accuracy computations which are visualized as a yellow color gradient that ranges from the lightest (highest accuracy) to the darkest ranges (lowest accuracy).

$$\sigma_z = (\frac{m}{f_b}) Z^2 \sigma_d \tag{3}$$

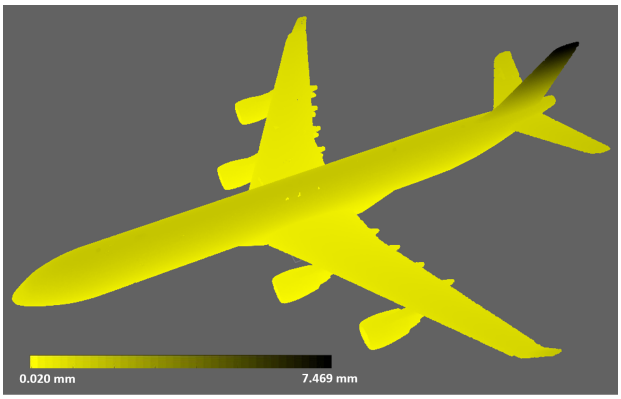
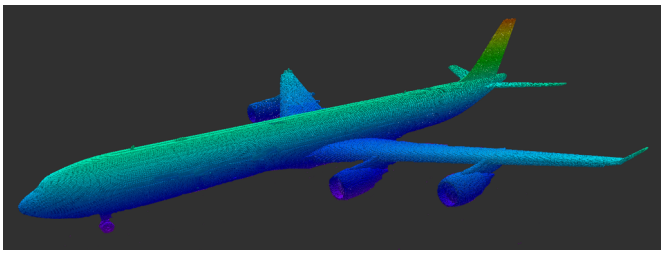


Fig. 4: Colour gradient model representing the accuracy across the generated 90% path ranging from the lightest (highest accuracy) to the darkest (lowest accuracy). A colormap shows the lowest and highest standard deviation of depth.

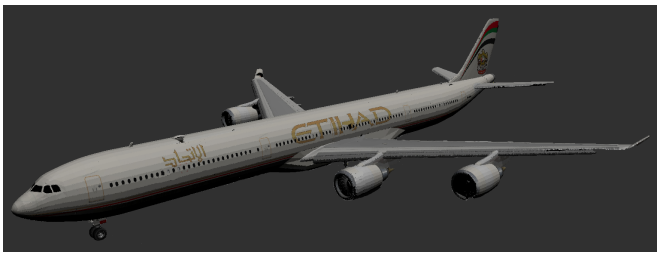
One of the generated coverage paths that targets a coverage of 90% was evaluated in simulation using Gazebo SITL. The result of following that path is shown in Figure 5. The 3D reconstruction was performed using Real Time Appearance Based Mapping (RTAB) [8] and Octomap [9]. The resulting model demonstrates that the generated path are traversable, and generate the desired coverage percentage.

VII. Conclusion

In this paper we introduced a coverage path planning approach that facilitates the inspection of large structures. We integrated sensor models to generate a coverage path offline and provide a prediction of the coverage percentage. The proposed algorithm is developed with a heuristic reward function that evaluates the distance, turning angle and sensor accuracy at each viewpoint. We also accelerated the coverage path planning algorithm by performing parallel implementation for the frustum and occlusion culling processes using Nvidia CUDA framework. The algorithm and the overall approach was verified using a realistic robot simulator where a quadrotor



(a) 3D reconstruction using Octomap



(b) 3D reconstruction using RTAB

Fig. 5: 3D reconstruction models generated by following the 90% coverage path

follows the generated coverage path and generates a 3D reconstructed model. Future work will focus on integrating the surface mesh area and a measure of information gain in the planning heuristics to generate coverage paths that guarantee coverage to a certain percentage efficiently.

References

[1] 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.

[2] Y. Bestaoui Sebbane, *Planning and Decision Making for Aerial Robots*, 2014, vol. 71.

[3] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. Cambridge, Mass: MIT Press, 2005.

[4] A. Wallar, E. Plaku, and D. A. Sofge, "A Planner for Autonomous Risk-Sensitive Coverage (PARC OV) by a Team of Unmanned Aerial Vehicles," *in IEEE Symposium on Swarm Intelligence (SIS)*, 2014.

[5] S. Hong, "Accelerating CUDA Graph Algorithms at Maximum Warp," *in ACM SIGPLAN Annual Symposium on Principles and Practive of Parallel programming*, 2011.

[6] Gazebo. [Online]. Available: http://gazebosim.org/

[7] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 12, pp. 1437–1454, 2012.

[8] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2661–2666.

[9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.