

# AUTONOMOUS UAV-BASED 3D-RECONSTRUCTION OF STRUCTURES FOR AERIAL PHYSICAL INTERACTION

Beril Sirmacek, Ramy Rashad\*, Patrick Radl

University of Twente, Enschede, The Netherlands  
b.sirmacek@utwente.nl and r.a.m.rashadhashem@utwente.nl

**KEY WORDS:** UAV, Path planning, SLAM, 3D segmentation, Autonomous navigation

## ABSTRACT:

We introduce a fully automated only path planning approach especially for drones. This novel method relies on usage of a stereo camera mounted at the bottom of a hexagonal drone for real-time point cloud reconstruction and localization. The real-time point cloud is analyzed in a software loop where the entropy of the point cloud and the surface normals are calculated. The low entropy positions (which indicate the 3D areas with less point density and less information) and the surface normals are used for calculating the next inspection point which can be targeted by the drone in order to enhance the point cloud best. Path planning to these automatically selected target points is done during the flight (quite real-time) and automatically. The initial experiments are performed on Gazebo simulation environment within the ROS system using realistic parameters of our real drone and real stereo camera.

## 1. INTRODUCTION

Unmanned aerial vehicles (UAVs) have become a successful cost-effective tool for various civilian applications such as visual inspection, surveillance, and mapping. Most of the current operational applications are restricted to passive tasks. However, UAVs can be also used in active tasks like contact-based inspection, aerial grasping, and aerial maintenance. This has stimulated researchers to investigate interaction control of aerial robots (Ruggiero et al., 2018). Impedance control is a widely used method for interaction control, where virtual springs and dampers are used to control the behavior of a robot. Thus, allowing an aerial robot to perform an interaction task, like grinding or polishing, with a complexly shaped body (Fasse and Broenink, 1997). In practice, the principal stiffness directions of the impedance controller are aligned with the surface normal of the desired body to be handled. In most of the scenarios which are addressed by the literature, there is at least a rough prior-knowledge about the body which needs to be investigated. This prior-knowledge might be known from a CAD/CAM model of the ideal body, or GIS data or point clouds acquired by laser imaging or photogrammetry. To allow UAVs to autonomously perform interaction tasks on unknown objects, it needs to visually build a 3D reconstructed model of the object. Rough knowledge about the structure can give opportunity to apply a pre-segmentation to the 3D environment and already design a flying path before starting inspection with drone (Mansouri et al., 2018). Even when the drone can make a path re-planning during the flight depending on the details inspected on the structure, the algorithms are still not totally unaware of what to expect approximately where.

Herein, we introduce a framework which is able to make path planning on the flight without having any prior information about the environment and the structure which needs to be investigated. Stereo cameras placed on the drone enable us to create a rough representation of the 3D environment in real-time, in order to determine the object of interest which must eventually have a dense and accurate 3D model.

\*Corresponding author

We have conducted our research and experiments in order to answer the following main question;

*In order to generate a complete 3D visual model of an unknown object, is it possible to determine a suitable flight direction, considering the completeness of the model on-the-fly?*

In addition to that, during our experiments, we have kept the following question in our minds;

*If it is possible to generate path online, would it help to create a better point cloud or would it make the path more time/energy saving for drone?*

We have conducted our experiments on Gazebo environment in ROS using a hexagonal drone with a stereo camera. We have selected the drone and the camera parameters similar to our real drone and camera system in order to be able to perform the experiments on the real environment after experiments on the simulation platform. Our simulation experiments indicate high potential of our framework in order to perform fully automated 3D reconstruction and inspection work on environments without prior information. In the future, such 3D reconstructed visual model will allow the drone to perform the interaction task with any complex shaped body with more autonomy without the prior knowledge of the geometry of it. Next we explain our method and we provide our experimental results in detail.

## 2. METHOD

For generating the rough real-time 3D environment models, we benefit from the robust approach ORBSLAM2 which extract visual features (ORBs) and finds their 3D positions based on a stereo camera observation (Mur-Artal and Tardós, 2017). ORBSLAM2 algorithm results with generating a point cloud where each point corresponds to the ORB features located in the correct position within the 3D environment.

The workflow of our novel, fully automated and active drone path planning algorithm is represented in Figure 1. The mathematical implementation details of each algorithm step are listed below.

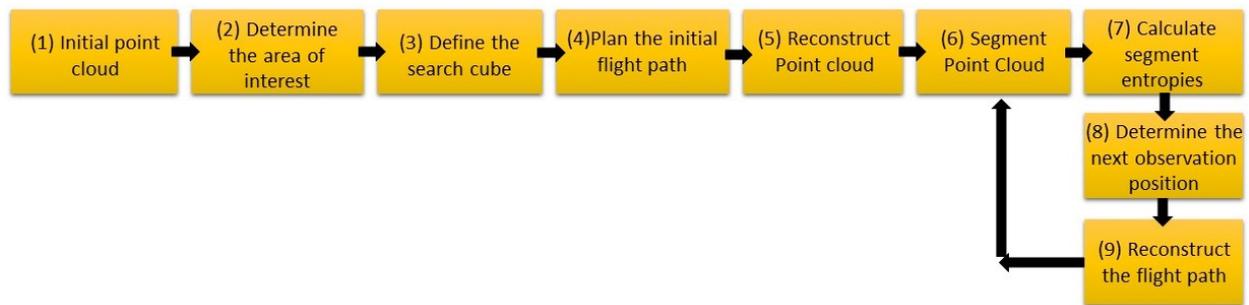


Figure 1. Work flow of the proposed autonomous UAV path planning algorithm.

(1) At this first step, the drone starts moving upwards while not moving in any angular direction. During this lifting period, the stereo camera captures frames and the ORBSLAM2 algorithm already starts building a point cloud.

(2) An initial rough voxel segmentation is used to determine the most dense area (approximately). We assume that the object of interest stands within this very dense area and the rest of the points are probably coming from the outliers, shadows, neighbour objects or calculation errors.

(3) The most dense voxel is assumed as the search cube. The center of mass in this voxel (based on the point locations), is taken as the center where the drone will use as the origin when it is designing circular paths.

(4) The initial path is planned as a circle around the search cube. This circle assumes the center as the 'center of mass' which is calculated in the earlier step. The radius is taken as equal to 1 meter added to the half of one voxel edge ( $r_{flight} = 1 + r_{object}$  where one voxel edge is  $2 * r_{object}$ .)

(5) While drone is flying through the initial path (a circle with a  $r_{flight}$  radius around the origin), ORBSLAM2 generates the initial point cloud of the object of the interest.

(6) The point cloud is segmented into 4 equal pieces around the origin point. (Fig. 4 shows the segments of an example point cloud, in order to represent how they are divided in 3D space.)

In our earlier experiments, we have applied segmentation on our point cloud by looking at 3D point cloud normal vectors and calculating the differences of normals (DoN) (Y. Ioannou and Greenspan, 2012) as a clue of the 3D object separation. The automatic segmentation of the point cloud could lead to precise calculation of the segments which have low entropy values. However, the time requirement of the DoN computation made us look for simplified solution in order to be able to choose the segments real-time during the flight.

(7) Again at this step, our initial idea was to calculate 3D local entropy values by calculating 3D local point cloud density and finding its 3D local minimums. After implementing this mathematical method, we noticed that the work flow became really slow for online autonomous planning. Especially, as the point cloud becomes denser, with time the workload increased and the real-time operation became unrealistic. We believe that the 3D local density minimum calculation could be done realistically if it is implemented on a GPU with multiple workers. However, for this initial experiment, we have decided to simplify the entropy

calculation. We have calculated one entropy value for each segment which is equal to the number of points within the segment. This simple calculation gave us opportunity to select the segment with the lowest entropy value in milliseconds time frame, while still finding out a meaningful position which needs to be targeted by our drone to get more information.

(8) For the next flight position, only the segment which has the lowest entropy value is taken into account. The  $(x_T, y_T, z_T)$  values of the target position is calculated as  $r_{flight}$  meter further away from the points which are in the half height of the segment.

(9) The flight path is computed as a circular route (with a  $r_{flight}$  meter radius) from the last position  $(x(t), y(t), z(t))$  of the drone to the  $(x_T, y_T, z_T)$  position. While the drone is flying to the next target position, the ORBSLAM2 algorithm continues adding more points to the point cloud and denser representation of the other object regions are generated on the way. When the drone reaches to the target point, the segmentation and entropy calculation algorithms are repeated. In this way, the drone continues finding the lowest entropy areas of the generated point cloud and finding the next target position in order to make more observations and continue filling the point cloud. In this way, we can densify the point cloud until a satisfactory density value (a pre-determined threshold) is reached. This fully automated path planning algorithm helps us to guarantee to not have holes or poor density information in some local areas of the 3D model.

In the next section, we compare the proposed autonomous path planning with the classical approach for building point cloud of an object (turning around the object in a certain flight radius with a certain speed while increasing its height with a constant speed).

### 3. EXPERIMENTS

We have implemented our algorithms within the ROS framework and the image acquisitions are done with a simulation of the stereo camera mounted on a drone in Gazebo environment as presented in 3 (Furrer et al., 2016). In order to make the assessment easier, we have started to our experiments by using a cylindrical structure for 3D reconstruction. However, usage of the cylindrical structure also helps us to consider our framework for a future work on 3D reconstruction and inspection of wind turbines.

The simulation environment includes the object of interest which is a cylinder with 1m radius ( $r_{object}$ ), 3m height ( $h$ ). The start position of the drone is always on the x/y plane at the  $y_0 = 0$  and  $x_0 = r_{flight}$  (at the boarder of segment0 and segment 3, and the flight radius  $r_{flight}$  is absolute to the object center). The stereo camera is placed at the bottom of the drone in a way where

$n_{rotations}$	$n_{points}$	$n_{rotations}$	$n_{points}$
5	5047	5	5200
10	6982	10	4565
15	7573	15	4477
20	7944	20	4255
25	8802	25	3993
50	8637	50	3856
100	8846	100	3931

Table 1. Column 2: Amount of points captured when flying a helix with a radius of  $2m$  ( $1m$  distance to the object surface) with  $\omega = 0.25rad/s$ . Column 4: Amount of points captured when flying a helix with a radius of  $3m$  ( $2m$  distance to the object surface) with  $\omega = 0.25rad/s$ .

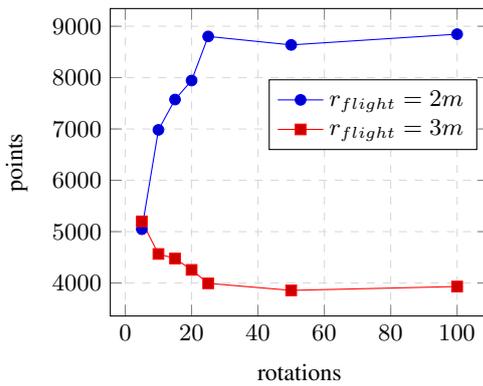


Figure 2. Plot showing the number of points captured when flying a helix with different radius and rotations at  $\omega = 0.25rad/s$

the propellers cannot occlude the field of view. The field of view angles offer the following angles;  $HFOV = 66^\circ$  and  $VFOV = 49.5^\circ$ . In order to make the environment realistic, the cylinder object is textured with a real photo and a directed sunlight is used. The sunlight is directed at segment 0's center, therefore segment 2 is kept fully in the shadow, while 1 and 3 are in the transition.

We first started with hard-coding a helix shape flight path with the formula provided at Equation 1. Flying in a helix shape path is commonly used in many inspection applications. In Table 1, we have provided a list of number of points when different flight radius and different number of rotations are chosen for this hard-coded setup. For better analysis, the table values are plotted at Fig. 2.

$$\begin{aligned}
 x(t) &= -r \cdot \cos(\omega t) + x_0 \\
 y(t) &= r \cdot \sin(\omega t) + y_0 \\
 z(t) &= v_z \cdot t + z_0 \\
 \psi(t) &= -\omega t + \psi_0
 \end{aligned} \tag{1}$$

Next, we provide the analysis from the autonomous experiment using the proposed work flow. For this experiment, we have assumed that the mass center of the point cloud is determined already. Therefore the flight radius  $r_{flight}$  is provided to the algorithm by user. The angular velocity  $\omega$  in positive rotational direction at a constant speed. After the drone made the initial

step	$n_{total}$	$n_{seg0}$	$n_{seg1}$	$n_{seg2}$	$n_{seg3}$
0	2463	1090	393	325	655
1	2262	686	364	325	887
2	2277	438	375	753	711
3	2495	790	687	398	620
4	2514	894	445	396	779

Table 2. Amount of points  $n$  captured at each step when flying the autonomous segment inspection with a  $r_{flight}$  of  $2m$  ( $1m$  distance to the object surface) with  $\omega = 0.25rad/s$

step	$n_{total}$	$n_{seg0}$	$n_{seg1}$	$n_{seg2}$	$n_{seg3}$
0	2433	248	494	882	809
1	2579	383	494	882	820
2	3394	829	932	808	825
3	3631	811	983	1011	826
4	3581	811	882	894	994

Table 3. Amount of points  $n$  captured at each step when flying the autonomous segment inspection with a  $r_{flight}$  of  $3m$  ( $2m$  distance to the object surface) with  $\omega = 0.25rad/s$

turn around the origin on a circular path with  $r_{flight}$  radius, the point cloud is segmented into four major pieces and the segment with the lowest entropy is selected. After the segment with the lowest entropy is determined and the shorter path either in positive or negative rotational direction on this circle is taken. At this step, also the neighbour segment with the lower entropy is determined and designated as the following segment for step 2. The remaining segments of steps 3 and 4 are visited in the same rotational direction as the segment of step 2 was compared to the one of step 1. This keeps the total path and therefore the flight time shorter.

During all these steps  $r_{flight}$  and  $z = 3m$  stays constant and  $\omega$  is limited. The results for different  $r_{flight}$  at  $\omega = 0.25rad/s$  can be found in Table 2 and 3 and are plotted in Figure 5 and 6.

Our experimental results show that the fully autonomous online path planning algorithm can generate similar number of points to the hard-coded helix path while doing less turns around the object. This means that the autonomous inspection can lead to reconstruct a larger area while using the energy of the drone consciously. Although, our total point cloud shows increase of the points during the inspection period, point numbers of some segments decreased. The reason is that, when ORBSLAM2 observes the same points again, some unreliable points are later removed. Therefore, decreasing point number does not indicate decrease in performance. We expect the performance to increase while the autonomous inspection repetitively finds the next target position to fly and fill the point cloud better. We will continue to our more detailed performance analysis in our future steps.

#### 4. CONCLUSIONS

Herein we have introduced a novel fully automated drone navigation method for 3D reconstruction of the structures when pre-knowledge of the structure shape, size and complexity are not available. Our experimental results indicate high potential of our framework in order to perform fully automated 3D reconstruction and inspection work on environments without prior information.

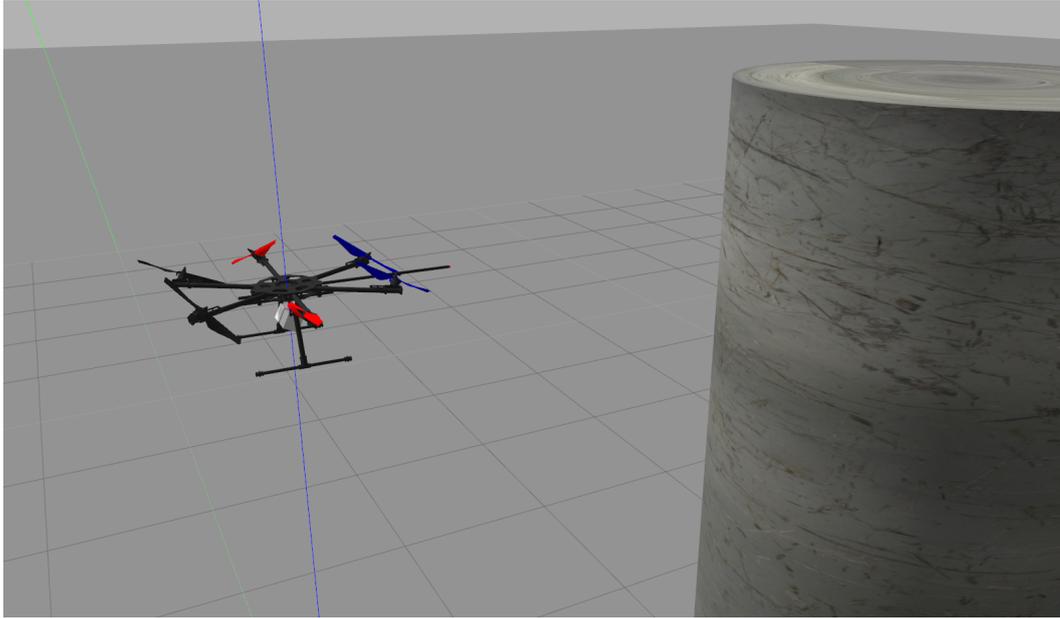


Figure 3. Our simulation environment: A drone with a stereo camera can create its flight path online, in order to make sure that the object of interest can be fully 3D reconstructed.

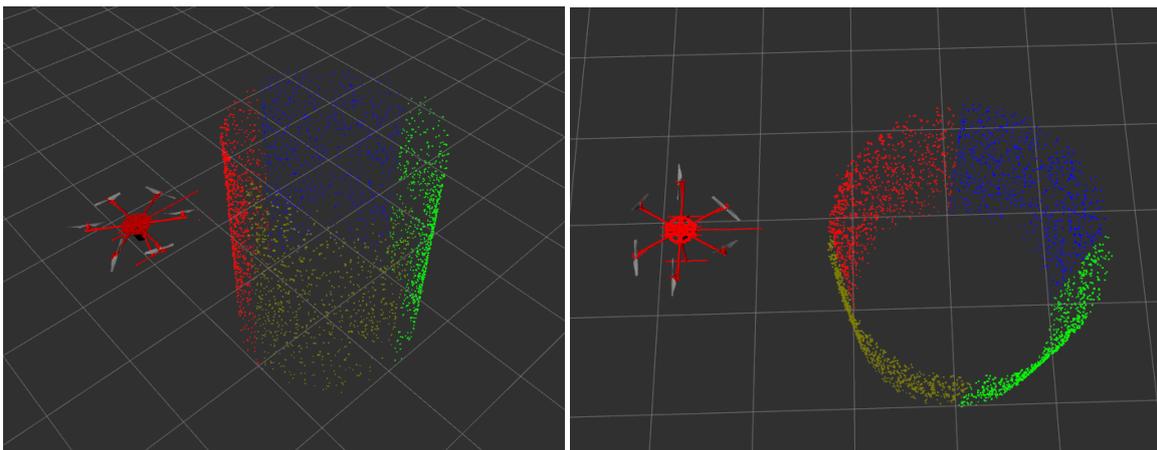


Figure 4. Side and top view of detected points. Point cloud is false colored in order to show the different segments. (Segment 0 = red points, Segment 1 = blue points, Segment 2 = green points, Segment 3 = yellow points).

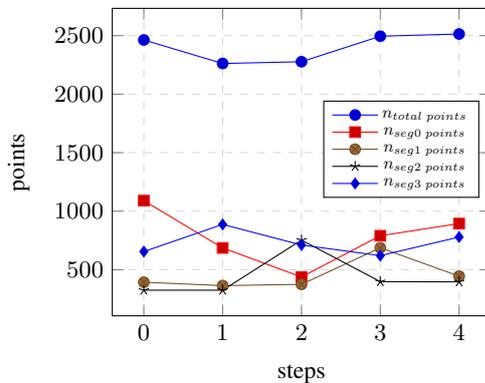


Figure 5. Plot showing the number of points captured at each step when flying the autonomous segment inspection with a radius of  $2m$  ( $1m$  distance to the object surface) with  $\omega = 0.25rad/s$

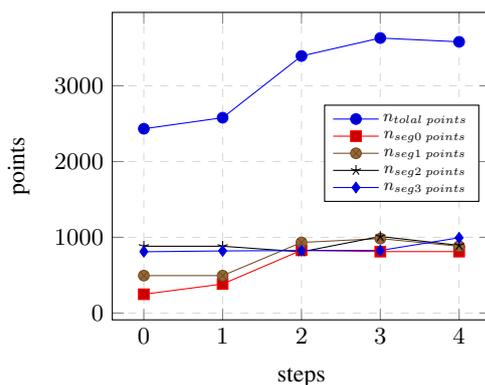


Figure 6. Plot showing the number of points captured at each step when flying the autonomous segment inspection with a radius of  $3m$  ( $2m$  distance to the object surface) with  $\omega = 0.25rad/s$

In our next steps, we will test our algorithms on more complex structures in order to identify the 3D reconstructed challenges of the proposed autonomous framework. Furthermore, we will identify the required steps in order to improve the algorithm for doing inspection by physical interaction with the structure. Last but not least, we will continue our experiments using an actual drone, in addition to our ROS simulations.

## REFERENCES

- Fasse, E. D. and Broenink, J. F., 1997. A spatial impedance controller for robotic manipulation. *IEEE Transactions on Robotics and Automation* 13(4), pp. 546–556.
- Furrer, F., Burri, M., Achtelik, M. and Siegwart, R., 2016. Rotors modular gazebo mav simulator framework. In: *Robot Operating System (ROS)*, Springer, pp. 595–625.
- Mansouri, S. S., Kanellakis, C., Fresk, E. and Nikolakopoulos, G., 2018. Cooperative aerial coverage path planning for visual inspection. *Control Engineering Practice* 74, pp. 118–131.
- Mur-Artal, R. and Tardós, J. D., 2017. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics* 33(5), pp. 1255–1262.
- Ruggiero, F., Lippiello, V. and Ollero, A., 2018. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters* 3(3), pp. 1957–1964.
- Y. Ioannou, B. Taati, R. H. and Greenspan, M., 2012. Difference of normals as a multi-scale operator in unorganized point clouds. *ArXiv e-prints*.