2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

2022 30th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-0673-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MED54222.2022.9837166

Experimental Evaluation of a Geometry-Aware Aerial Visual Inspection Framework in a Constrained Environment

Vignesh Kottayam Viswanathan, Sumeet Gajanan Satpute, Björn Lindqvist,
Christoforos Kanellakis and George Nikolakopoulos

***Abstract*— This article aims to present an experimental evaluation of an offline, geometry-aware aerial visual inspection framework, specifically in constrained environment, established for geometrically fractured objects, by employing an autonomous unmanned aerial vehicle (UAV), equipped with on-board sensors. Based on a model-centric approach, the proposed inspection framework, generates inspection view-points around the geometrically fractured object, subject to the augmented static bounds to prevent collisions. The novel framework of visual inspection, presented in this article, aims to mitigate challenges arising due to the spatially-constrained environment, such as limited configuration space and collision with the object under inspection, by accounting for the geometrical information of the vehicle to be inspected. The efficacy of the proposed scheme is experimentally evaluated through large scale field trials with a mining machine.**

***Index Terms*— geometry-aware, visual inspection, constrained environment, autonomous unmanned aerial vehicle,geometrically fractured object**

I. Introduction

The application of aerial robotic vehicles have been actively studied by the academia towards navigation, exploration and mapping in constrained environments in the last decade [1]–[5]. Robotic solutions to perform structural monitoring of 3D infrastructures, such as employing Unmanned Aerial Vehicles (UAVs) to perform visual inspection of high-rise structures such as dams [6]–[8], bridges [9]–[11], civil buildings [12], [13], remote wind-turbines [14], [15] power lines [16], [17] and rail tracks [18], [19] has received lot of attention due to their inherent versatility and low-operational cost. Due to their inherent versatility to be rapidly configured for the task at hand and being a cost-effective method in recent years, aerial robots have been actively incorporated into inspection and diagnosis of large ageing structures. With rising autonomy in resource-mining industry, autonomous UAVs are the ideal companion to enable a quick inspection task and coverage.

The primary aim, while performing the inspection task, is to obtain an accurate and information-rich 3D reconstructed mesh from the on-board optical sensor, such as an *RGB-D* camera. The quality of the mesh can be mainly affected by the generated inspection trajectory, the observed view-orientations and the degree of photogrammetric constraints considered. In view of coverage of structures, [20] presents a novel multi-agent cooperative coverage path planning (C-CPP) approach to experimentally inspect a wind-turbine. The proposed work utilizes a geometry-based inspection path derived from processing the point cloud model of the structure. In [21], the author presents a sampling-based generation of view-points and path primitives for model-based structural inspection of buildings using UAVs. In [22], the author implements a meta-heuristic based optimization of waypoints that satisfies multiple outlined inspection objectives for a sloped dam. In view of past works pertaining to visual inspection in constrained environments, [23] presents a use-case of an aerial inspection of thermal boiler plants. In their work, the authors proposed a resilient motion estimation architecture that is based on the on-board stereo camera and an Inertial Measurement Unit (IMU) sensor. In [24], the authors presented a visual inspection in a GPS-denied environment, such as that of dam penstocks, with main focus on fused six Degree-of-Freedom (DoF) pose estimation performance, obtained from visual odometry and range estimated using Micro Aerial Vehicles (MAVs), while their prime consideration was towards the manually-controlled remote operation of MAV that can be safely deployed towards performing visual inspection of long penstocks. In [25], the authors proposed a close-range aerial visual inspection platform to be used in an indoor environment, where the focus was towards building an experimental platform to validate inspection path-planning techniques in a controlled set-up that can be easily replicated for real-life scenarios.

In contrast to the current trend in visual inspection in constrained environments, the main contributions of this article are as follows: a) a novel model-centric geometry-aware visual inspection framework is proposed to cover a geometrically fractured object in a constrained environment, b) a novel experimental evaluation of the proposed visual inspection framework in an underground mine and with a full scale mining machine is presented.

The remainder of the article is structured accordingly. In Section II the experimental framework is presented and explained along with the augmented static bound and the utilized view orientation method. In Section III the implemented control architecture is presented including the modelled non-linear system dynamics and the applicable communication framework for the experiments. In Section IV, results from the utilized framework and their corresponding discussions

This work has been partially funded by the illuMINEation Horizon 2020 project from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 869379.

The Authors are with the Robotics and AI Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå

Corresponding Author's email: vigkot@ltu.se

The corresponding author would like to convey his thanks to Christoforos Kanellakis, Björn Lindqvist and Ilias Tzetvidis for their support towards field trial

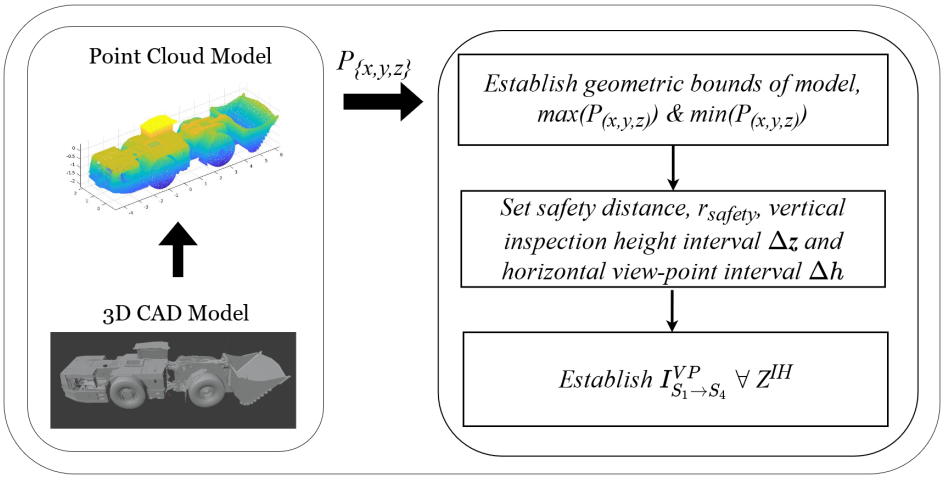

Fig. 1: The proposed Geometry-Aware Visual Inspection Framework

are presented. Finally, in Section V the conclusions are drawn.

II. Visual Inspection Framework

The overarching representation of the proposed novel visual inspection framework is outlined in Fig. 1. The evaluated method is offline and utilizes an *a-priori* knowledge about the inspected geometrically fractured object. From the available 3D Computer Aided Design (CAD) model of the target object, a dense point cloud representation ($\mathbf{P_{(x,y,z)}} \in \mathbb{R}^3$) of the target structure is obtained using *CloudCompare*, an open-source 3D point-cloud processing software.

As the presented use-case is considered to be in a spatially-constrained environment like underground mines or voids, a safe and stable flight profile is pertinent to have a successful inspection mission. To construct the static bounding box around the object, we first consider [X_O Y_O Z_O]$\in \mathbb{R}^3$ as the object-centered coordinate frame $\mathbb{O}$, in which the point cloud $P_{(x,y,z)}$ is defined. Next, we obtain the *min* and *max* values along the respective axes of the point-cloud from (1). This enables the framework to take into account the geometry of the object being inspected in order to prevent the framework from generating view-points immediately above the object or located on the surface of the object.

$$\begin{bmatrix} x_{max} \\ y_{max} \\ z_{max} \end{bmatrix} = max \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \\ \begin{bmatrix} x_{min} \\ y_{min} \\ z_{min} \end{bmatrix} = min \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{1}$$

In (1), P_x, P_y and P_z represent the corresponding points of the point cloud along the X_O, Y_O and Z_O axis respectively and utilizing the `alphashape` command in *MATLAB* generates a bounding-box region, encapsulating the geometrically fractured object. Given a set of points $S(p)$ where $p \in \mathbb{Z}^+$ is the number of points available and $S \subset \mathbb{R}^d$ where d is the dimension of points in S, `alphashape` generates the enclosed α-hull from the intersection of closed discs formed with a radius of $1/\alpha$ that contains the said set of points [26]. Figure. 2 provides a pictorial depiction of the aforementioned restricted region constructed from (1).

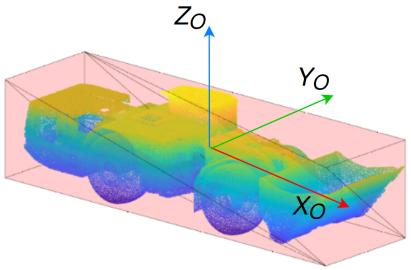

Fig. 2: Graphical representation of model-driven geometric bounding box encapsulating the object

At this point, it should be highlighted that the proposed framework assumes that the knowledge of the constrained environment is unavailable *a-priori*. Hence, an estimate of an offset safety distance ($r_{safety} \in \mathbb{R}$) is initially considered. Defining r_{safety} enables us to tackle the issue of a limited configuration space to later generate view-points based on *in-situ* information on the available space around the object. Incorporating the applicable safety distance to the known geometric bounds of the object gives rise to the augmented static bounds. Equation (2) provides the mathematical representation of the integrated stand-off distance with the geometric bounds considered in (1).

$$X_O^{SD} = \begin{bmatrix} x_{max} + r_{safety} \\ x_{min} - r_{safety} \end{bmatrix} \\ Y_O^{SD} = \begin{bmatrix} y_{max} + r_{safety} \\ y_{min} - r_{safety} \end{bmatrix} \tag{2}$$

Here in eq. (2), $\mathbf{X_O^{SD}} \in \mathbb{R}^{2\times1}$ becomes the final operating stand-off distance along the X_O-axis. Along the Y_O-axis, $\mathbf{Y_O^{SD}} \in \mathbb{R}^{2\times1}$ is designed to complete the portion of the proposed framework thus leading towards the modelling of

a safe and collision-free inspection behaviour. This allows us to design a primitively-shaped bounds encapsulating the geometrically fractured object, thus enabling the framework to safely generate view-points around the object of interest based on a simplified representation of the target object.

Taking into account the limited vertical space available, i.e from the ground till the roof of the constrained environment in consideration, the upper and lower vertical bounds are derived from the model information that will constrain the framework from generating view-points that can endanger the safety of the UAV. This work implements a loop-based approach at various altitudes to inspect the object. The process of modelling the inspection heights is described as:

$$Z_k^{IH} = z_{min} + z_{safety} + (k-1)\Delta z \ < z_{max} \tag{3}$$

where $\mathbf{Z^{IH}} \in \mathbb{R}^{k\times 1}$, the modelled inspection heights at which the UAV performs a loop-based visual inspection based on the horizontal stand-off distance described beforehand, $z_{safety} \in \mathbb{R}$ is the minimum approachable vertical safety height, $k \in \mathbb{R}$ refers to the number of vertical increments required for inspection with $k = [1, 2...]$ and Δz can be the minimum required vertical distance to maintain an image overlap.

Assuming a clock-wise direction of motion for the inspection task, for $k = 1$ the first point of inspection route can be written as the point of intersection of three planes given as:

$$\begin{aligned} X &= x_{max} \\ Y &= {Y_O^{SD}}_{(2,1)} \\ Z &= z_{min} + z_{safety} \end{aligned} \tag{4}$$

where $(X, Y, Z) \in \mathbb{R}^3$. As the implemented static bounds around the object is of a primitive shape, decomposing the inspection region around the object based on the direction of motion along the respective axis into four sections ($S_1 \rightarrow S_4$) in the X_OY_O- plane results in a straightforward method to model the required inspection view-points. Figure 3 graphically conveys the intended sequence of generation of view-points obtained for a single pass around the object.

In S_1, the set of inspection view-points $\mathbf{I_{S_1}^{VP}} \in \mathbb{R}^{3\times j}$ can be written from (4) as:

$$\begin{aligned} X &= x_{max} + (j-1)\Delta h \ \leq x_{min} \\ &\text{for constant,} \\ Y &= {Y_O^{SD}}_{(2,1)} \\ Z &= z_{min} + z_{safety} \end{aligned} \tag{5}$$

where, Δh is the required horizontal view-point interval to satisfy inspection parameters and $j = [1, 2...]$ is the number of horizontal increments required for inspection along $X_\mathbf{O}$ axis. In the same fashion, for S_2, the set of inspection view-points $\mathbf{I_{S_2}^{VP}} \in \mathbb{R}^{3\times m}$ can be written as:

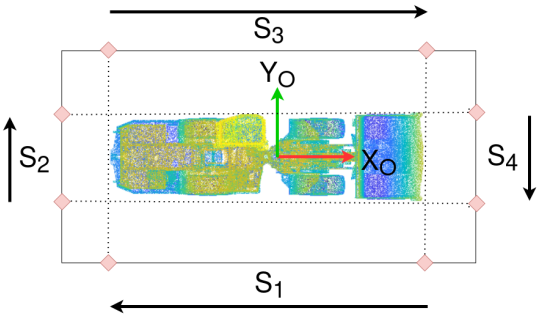

Fig. 3: Depiction of the decomposed 2D inspection region around the object. The *black arrows* indicate the direction of motion followed by the framework to generate the view-points. The *diamonds* represent the extreme bounds in the X_OY_O plane from (1) within which the view-points are generated.

$$\begin{aligned} Y &= y_{min} + (m-1)\Delta h \ \leq y_{max} \\ &\text{for constant,} \\ X &= {X_O^{SD}}_{(2,1)} \\ Z &= z_{min} + z_{safety} \end{aligned} \tag{6}$$

where, $m = [1, 2, ...]$ is the number of horizontal increments required for inspection along $Y_\mathbf{O}$ axis. Similarly, the above approaches can be applied to S_3 and S_4 respectively to obtain the final integrated three dimensional inspection view-points. Now, $\xi_{\mathbf{ref}}^{\mathbf{V}} \in \mathbb{R}^3$ for a single pass at $k = 1$ can be written in the accrued form as,

$$\xi_{\mathbf{ref}}^{\mathbf{V}} = \mathbf{I}_{\mathbf{S_1 \rightarrow S_4}}^{\mathbf{VP}} \tag{7}$$

With the desired inspection view-point matrix $\xi_{\mathbf{ref}}^{\mathbf{V}}$ around the object generated, the applicable view orientation is then addressed. This primarily depends on the chosen sensor configuration. In this work, the UAV is considered to have a front-facing camera, which requires maintaining a desired yaw angle $\psi_d \in \mathbb{R}$ with roll $\phi_d \in \mathbb{R}$ and pitch $\theta_d \in \mathbb{R}$ set as zero. We consider that the UAV maintains a center-look attitude configuration throughout inspection. Based on this, the object of interest is assumed to be at the center with the coordinates $(0, 0)$ as given in (8):

$$\begin{aligned} \phi_d &= 0 \\ \theta_d &= 0 \\ \psi_d &= atan2(\frac{0 - \xi_{ref(2,:)}^V}{0 - \xi_{ref(1,:)}^V}) \end{aligned} \tag{8}$$

In Fig. 4, a graphical plot of inspection view-points that cover the geometrically fractured object is provided. Using a vertical interval distance Δz= 0.6m, a horizontal interval distance Δh = 0.5m, minimum vertical safety altitude z_{safety} =0.86m and r_{safety} = 2m, the visual inspection framework generates three inspection loops at 0.86m, 1.46m and 2.06m respectively.

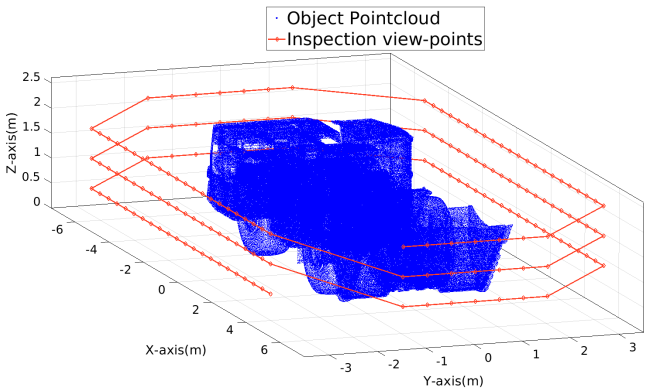

Fig. 4: Sequence of inspection view-points around the object based on the proposed inspection framework

III. Controller Methodology

A. Nonlinear Equations of Motion

In Fig. 5, the coordinate frames employed for the modelling of the UAV equations of motion are given, where [X_W Y_W Z_W] $\in \mathbb{R}^3$ represent the states in the world frame ($\mathbb{W}$) and [X_B Y_B Z_B] $\in \mathbb{R}^3$ represent the states in the UAV body frame ($\mathbb{B}$). (9) gives an second-hand overview of the equations used to represent the UAV states, while a complete and comprehensive explanation on the used equation of motion can be found in [27], [28].

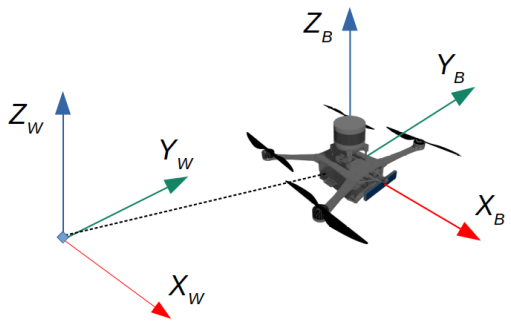

Fig. 5: Representation of the implemented world ($\mathbb{W}$) and body ($\mathbb{B}$) frames of reference

In (9), the six Degrees-of-Freedom (DoF) equation of motion is given for the UAV, where *p(t)* $\in \mathbb{R}^3$ and *v(t)* $\in \mathbb{R}^3$ denotes the translational states $[x, y, z]$ and $[v_x, v_y, v_z]$ in frame $\mathbb{W}$ along their respective axis. The current and commanded rotational states, roll ($\phi, \phi_{ref} \in \mathbb{R}$) and pitch ($\theta, \theta_{ref} \in \mathbb{R}$) angles are along their conventional axis of motion.

$$
\begin{aligned}
\dot{p}(t) &= v(t) \\
\dot{v}(t) &= R(\phi,\theta)\begin{bmatrix}0\\0\\T\end{bmatrix} + \begin{bmatrix}0\\0\\-g\end{bmatrix} - \begin{bmatrix}A_x & 0 & 0\\0 & A_y & 0\\0 & 0 & A_z\end{bmatrix} V(t) \\
\dot{\phi}(t) &= \frac{1}{\tau_\phi}(K_\phi \phi_{ref}(t) - \phi(t)) \\
\dot{\theta}(t) &= \frac{1}{\tau_\theta}(K_\theta \theta_{ref}(t) - \theta(t))
\end{aligned}
\tag{9}
$$

The acceleration component of the UAV is dependent on the direction and magnitude of the thrust vector given by the rotation matrix ($R(\phi,\theta) \in SO(3)$) and thrust ($T \in \mathbb{R}$). The terms A_x, A_y, $A_z \in \mathbb{R}$ stand for the linear damping coefficients considered, $g \in \mathbb{R}$ represent the earth's gravity, the attitude gains K_ϕ,$K_\theta \in \mathbb{R}^2$ and the time constants τ_ϕ,$\tau_\theta \in \mathbb{R}^2$.

B. Nonlinear Model Predictive Controller

Adopted from [28], let $\mathbf{x_k}$ the state vector of the UAV be given as $\mathbf{x} = [p, v, \phi, \theta]^T$ and the corresponding control action $\mathbf{u_k}$ represented as $\mathbf{u} = [T, \phi_{ref}, \theta_{ref}]^T$ at the k^{th} instant. Let $\mathbf{x}_k = x_{(k+j|k)_j}$ and $\mathbf{u}_k = u_{(k+j|k)_j}$ be the state and control vector obtained throughout the prediction horizon N. Using an explicit integrator, which in this work is the forward Euler method, and a discretizing time-step of value Ts we obtain the discrete prediction model shown in (10),

$$
\mathbf{x_{k+1}} = f(\mathbf{x_k}, \mathbf{u_k}) \tag{10}
$$

The applicable cost function with the positive-definite state, input control and the associated input rate penalty matrices $Q_x \in \mathbb{R}^{8\times8}$, Q_u, $Q_{\Delta u} \in \mathbb{R}^{3\times3}$ is modelled as:

$$
\begin{aligned}
J(\mathbf{x_k}, \mathbf{u_k}, \mathbf{u_{k-1|k}}) = \sum_{j=0}^{N} & \underbrace{||\mathbf{x_{ref}} - \mathbf{x_{k+j|k}}||^2 Q_x}_{state\ cost} \\
& + \underbrace{||\mathbf{u_{ref}} - \mathbf{u_{k+j|k}}||^2 Q_u}_{input\ cost} \\
& + \underbrace{||\mathbf{u_{k+j|k}} - \mathbf{u_{k+k-1|k}}||^2 Q_{\Delta u}}_{input\ smoothness\ cost}
\end{aligned}
\tag{11}
$$

where $\mathbf{x_{k+j|k}}$ and $\mathbf{u_{k+j|k}}$ are the predicted state vector and its corresponding input action vector taken at time-step k for time-step $k+j$. The term *state cost*, addresses the deviation in commanded reference state. As the UAV is commanded to maintain a minimum hovering thrust, $\mathbf{u_{ref}} = [g, 0, 0]$, while the mission is executed, the term *input cost* enforces the controller to meet the required state. Finally, the term *input smoothness cost* ensures that the deviations in the successive inputs are kept to minimum. The resulting NMPC problem can be written as:

$$
\begin{aligned}
&\min_{\mathbf{x_k}, \mathbf{u_k}} J(\mathbf{x_k}, \mathbf{u_k}, \mathbf{u_{k-1|k}}), \\
&\text{s.t. } : \mathbf{x_{k+j+1|k}} = f(\mathbf{x_{k+j|k}}, \mathbf{u_{k+j|k}}), j = 0, 1..N-1, \\
&\qquad \mathbf{u_{min}} \leq \mathbf{u_{k+j+1|k}} \leq \mathbf{u_{max}}, j = 0, 1..N
\end{aligned}
\tag{12}
$$

where $\mathbf{u_{min}} = [5\ -0.5\ -0.5]^T$ and $\mathbf{u_{max}} = [13.5\ 0.5\ 0.5]^T$ is given in their SI-units. The optimization problem (12) is solved by *PANOC* [29], [30]. Full information on the utilized solver can be found at [28], [29], [31] and the references therein. The UAV possesses a low-level attitude controller that considers the required roll, pitch and mass normalized thrust commands to provide corresponding motor speeds.

IV. Experimental Evaluation of Framework

Figure 6 shows the unmanned aerial vehicle used for the experimental evaluation of the visual inspection framework. For the purpose of the experiment, the UAV can be seen equipped with a front-facing Intel Realsense D455 stereo camera and a VL16 Velodyne 3D LiDAR sensor.

Fig. 6: Field-ready unmanned aerial vehicle. [3]

The adopted autonomy architecture on-board the UAV, with full information available in [3], utilizes $LIO-SAM$, a tightly coupled lidar inertial odometry in [32], an artificial potential field (APF) based obstacle avoidance scheme and a reference tracking controller as explained in III-B. In Fig. 7, the adopted autonomy architecture implemented in this work is shown.

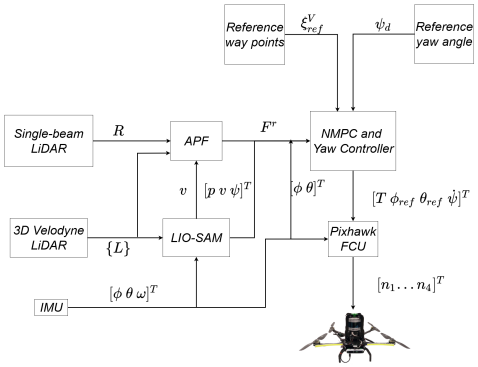

Fig. 7: Adopted controller scheme on-board the UAV [3]

where R is the range measurement along the $Z_{\mathbb{B}}$ axis, $\{L\}$ the local point cloud generated and F^r is the repulsive potential force $[F^r_x\ F^r_y\ F^r_z]$ along the respective axis. Figure 8 shows the communication architecture deployed for the experimental evaluation. The inspection view-points ξ^V_{ref} and it's corresponding yaw orientation ψ_{ref} is generated in *MATLAB* and is passed on to a custom reference pose publisher that over *ROS* communicates with the implemented model predictive controller to command the UAV to follow the desired pose. The UAV is assigned the start position in

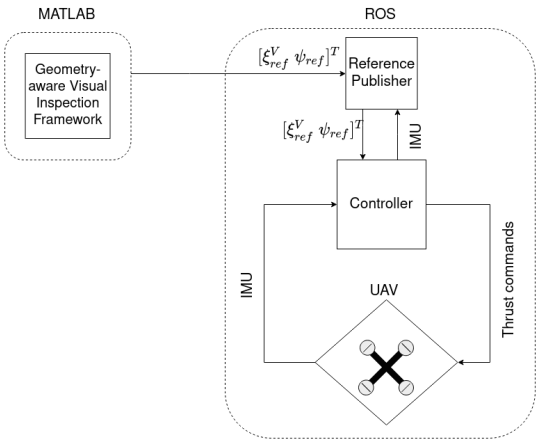

Fig. 8: Graphical depiction of the communication architecture implemented over *ROS*

front of the object, with the inspection route beginning from the *front-left* position. The single pass inspection route is modelled initially in *MATLAB* for a Z^{IH} of 1.6m. As it was observed, having a shorter horizontal interval distance Δh led to an unstable viewing attitude at the inspection view-points as the UAV was in a continuous state of reaching the next view-point, Δh was increased to 1.8m to reduce the oscillation faced. Figure 9 presents the results from the conducted experimental evaluation of the geometry-aware visual inspection framework.

Since the $\mathbb{O}$ frame on the object and the $\mathbb{W}$ frame considered for the UAV do not coincide, the generated view-points in the $\mathbb{O}$-frame are translated to the $\mathbb{W}$-frame with respect to the starting point of the UAV from the center of the object. From the obtained results, it can be seen that due to the absence of the knowledge about the environment in the framework, the generated inspection view-points, though model-centric in nature, have view-points close to the periphery of the constrained environment. Moreover, the offline nature of the inspection framework creates an additional point of concern related to the safety of the UAV. As the view-points are translated according to the start-position of the UAV, a few view-points especially to the back of the object can be seen very close to the machine.

In Fig. 10, the viewing-direction maintained by the UAV during the single pass is shown. It can be seen that while the UAV maintains the required view-orientation, the continuous center-look attitude is not preferred for geometrically fractured objects as the incident viewing angle is not normal to the surface being observed. This also leads to occlusions, caused by the shape of the GFO, leading to gaps in the reconstructed mesh. As such, resulting in an inaccurate 3D reconstruction from the recorded image data-set.

In Fig. 11, the post-processed 3D reconstruction from the LiDAR data, obtained during the experiment, is viewed in *CloudCompare*. From this Figure, it is also obvious the necessity to perform multiple inspection routes to obtain an accurate mesh of the geometrically fractured model. However, due to safety issues, with the UAV's possible

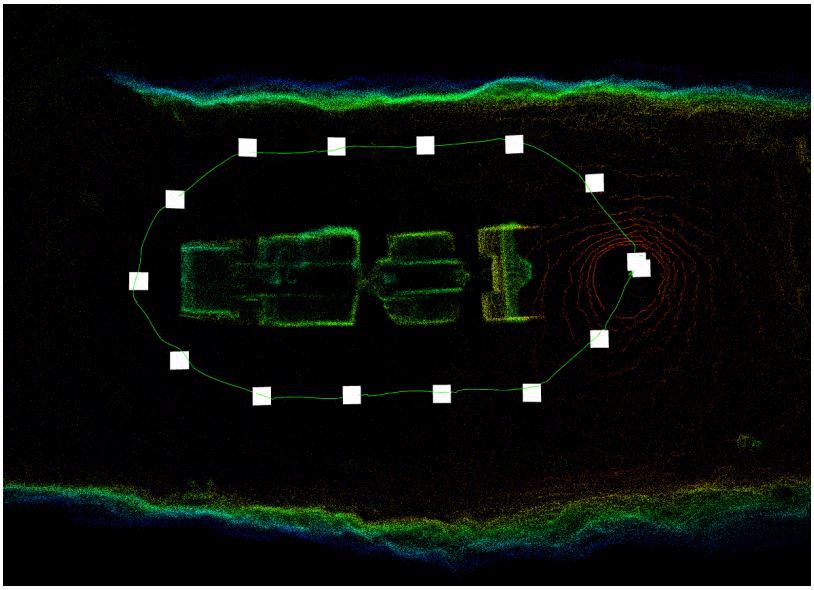

Fig. 9: A single pass inspection route around the geometrically fractured object at $Z^{IH} = 1.6$m viewed in *Rviz*. The *green* line indicates the inspection path followed by the UAV . The *white* square blocks indicate the reference view-points published using the aforementioned communication architecture

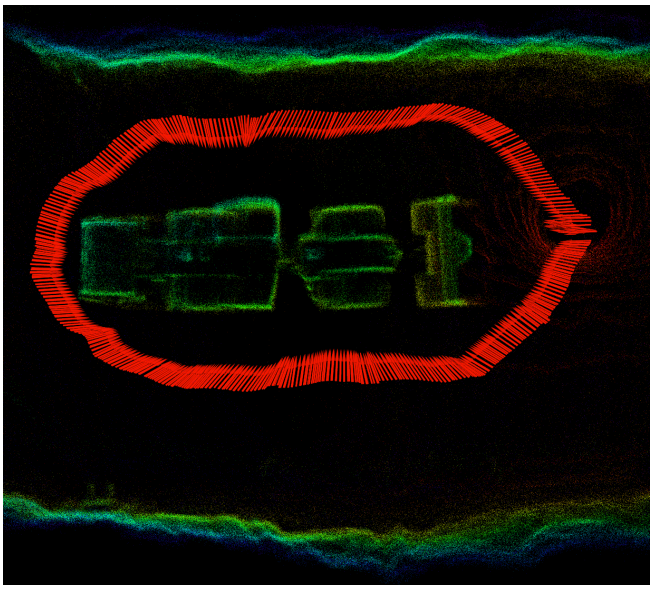

Fig. 10: Experimental view orientation maintained by the unmanned aerial vehicles during a single pass around the geometrically fractured object (mining machine)

collision with the walls of the constrained environment, only a single inspection route could be experimentally evaluated.

V. Conclusions

In this article, an experimental evaluation of the proposed offline geometry-aware visual inspection framework, specifically in a constrained environment by employing an UAV was presented. To enable a collision-free inspection, the framework utilized a primitively-shaped augmented static bounds representation around the object. By implementing a

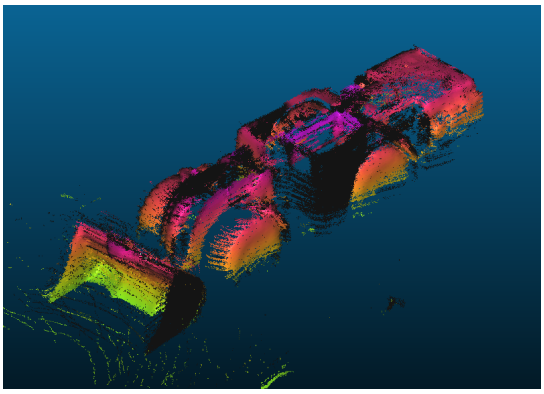

Fig. 11: Point cloud reconstruction of the geometrically fractured object during the single pass inspection

constant horizontal interval distance and a vertical interval distance, the framework generated view-points around the object for the entire height of the object. The performance of the proposed framework was evaluated by conducting field experiments in an underground mine. In view of future work, a comprehensive framework considering system dynamics, camera footprint and adaptive view-point generation methodology should be implemented to stave off the issues faced during experimental evaluation.

References

[1] M. Petrlík, T. Báča, D. Heřt, M. Vrba, T. Krajník, and M. Saska, "A robust uav system for operations in a constrained environmentkanellakis2018towards," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2169–2176, 2020.

[2] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying mavs for autonomous navigation in dark underground mine environments," *Robotics and Autonomous Systems*, vol. 126, p. 103472, 2020.

[3] B. Lindqvist, C. Kanellakis, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Compra: A compact reactive autonomy framework for subterranean mav based search-and-rescue operations," *arXiv preprint arXiv:2108.13105*, 2021.

[4] C. Kanellakis, S. S. Mansouri, G. Georgoulas, and G. Nikolakopoulos, "Towards autonomous surveying of underground mine using mavs," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2018, pp. 173–180.

[5] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund, *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.

[6] A. Khaloo, D. Lattanzi, A. Jachimowicz, and C. Devaney, "Utilizing uav and 3d computer vision for visual inspection of a large gravity dam," *Frontiers in Built Environment*, vol. 4, p. 31, 2018.

[7] S. Zhao, F. Kang, J. Li, and C. Ma, "Structural health monitoring and inspection of dams based on uav photogrammetry with image 3d reconstruction," *Automation in Construction*, vol. 130, p. 103832, 2021.

[8] M. F. Pinto, L. M. Honorio, A. Melo, and A. L. Marcato, "A robotic cognitive architecture for slope and dam inspections," *Sensors*, vol. 20, no. 16, p. 4579, 2020.

[9] N. Hallermann and G. Morgenthal, "Visual inspection strategies for large bridges using unmanned aerial vehicles (uav)," in *Proc. of 7th IABMAS, International Conference on Bridge Maintenance, Safety and Management*, 2014, pp. 661–667.

[10] S. Jung, S. Song, P. Youn, and H. Myung, "Multi-layer coverage path planner for autonomous structural inspection of high-rise structures," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[11] E. Jeong, J. Seo, and J. Wacker, "Literature review and technical survey on bridge inspection using unmanned aerial vehicles," *Journal of Performance of Constructed Facilities*, vol. 34, no. 6, p. 04020113, 2020.

[12] J. G. Martinez, M. Gheisari, and L. F. Alarcón, "Uav integration in current construction safety planning and monitoring processes: Case study of a high-rise building construction project in chile," *Journal of Management in Engineering*, vol. 36, no. 3, p. 05020005, 2020.

[13] Y. Tan, S. Li, H. Liu, P. Chen, and Z. Zhou, "Automatic inspection data collection of building surface based on bim and uav," *Automation in Construction*, vol. 131, p. 103881, 2021.

[14] C. Kanellakis, E. Fresk, S. S. Mansouri, D. Kominiak, and G. Nikolakopoulos, "Autonomous visual inspection of large-scale infrastructures using aerial robots," *arXiv preprint arXiv:1901.05510*, 2019.

[15] M. Shafiee, Z. Zhou, L. Mei, F. Dinmohammadi, J. Karama, and D. Flynn, "Unmanned aerial drones for inspection of offshore wind turbines: A mission-critical failure analysis," *Robotics*, vol. 10, no. 1, p. 26, 2021.

[16] S. Shuangchun, L. Yanlei, Y. Zhenxiao, W. Kai, Y. Ping, Z. Kun, and Y. Tao, "Review of autonomous inspection technology for power lines using uavs," in *2021 IEEE International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT)*. IEEE, 2021, pp. 481–484.

[17] N. Ayoub and P. Schneider-Kamp, "Real-time on-board deep learning fault detection for autonomous uav inspections," *Electronics*, vol. 10, no. 9, p. 1091, 2021.

[18] Y. Wu, Y. Qin, Z. Wang, and L. Jia, "A uav-based visual inspection method for rail surface defects," *Applied sciences*, vol. 8, no. 7, p. 1028, 2018.

[19] A. K. Singh, A. Swarup, A. Agarwal, and D. Singh, "Vision based rail track extraction and monitoring through drone imagery," *Ict Express*, vol. 5, no. 4, pp. 250–255, 2019.

[20] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, pp. 118–131, 2018.

[21] W. Jing, D. Deng, Z. Xiao, Y. Liu, and K. Shimada, "Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1472–1479.

[22] I. Z. Biundini, M. F. Pinto, A. G. Melo, A. L. M. Marcato, L. M. Honório, and M. J. R. Aguiar, "A framework for coverage path planning optimization based on point cloud for structural inspection," *Sensors*, vol. 21, no. 2, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/570

[23] M. Burri, J. Nikolic, C. Hürzeler, G. Caprari, and R. Siegwart, "Aerial service robots for visual inspection of thermal power plant boiler systems," in *2012 2nd international conference on applied robotics for the power industry (CARPI)*. IEEE, 2012, pp. 70–75.

[24] T. Özaslan, K. Mohta, J. Keller, Y. Mulgaonkar, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Towards fully autonomous visual inspection of dark featureless dam penstocks using mavs," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4998–5005.

[25] Z. Shang and Z. Shen, "Indoor testing and simulation platform for close-distance visual inspection of complex structures using micro quadrotor uav," *arXiv preprint arXiv:1904.05271*, 2019.

[26] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on information theory*, vol. 29, no. 4, pp. 551–559, 1983.

[27] B. Lindqvist, S. S. Mansouri, J. Haluška, and G. Nikolakopoulos, "Reactive navigation of an unmanned aerial vehicle with perception-based obstacle avoidance constraints," *IEEE Transactions on Control Systems Technology*, 2021.

[28] B. Lindqvist, S. S. Mansouri, and G. Nikolakopoulos, "Non-linear mpc based navigation for micro aerial vehicles in constrained environments," in *2020 European Control Conference (ECC)*. IEEE, 2020, pp. 837–842.

[29] P. Sopasakis, E. Fresk, and P. Patrinos, "OpEn: Code generation for embedded nonconvex optimization," in *IFAC World Congress*, Berlin, 2020.

[30] A. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using panoc," in *2018 European control conference (ECC)*. IEEE, 2018, pp. 1523–1528.

[31] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3556–3563.

[32] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.