IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 December 2023, accepted 24 January 2024, date of publication 30 January 2024, date of current version 7 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3360209

APPLIED RESEARCH

Autonomous Inspection of High-Rise Buildings for Façade Detection and 3D Modeling Using UAVs

PRAYUSHI MATHUR[1], CHARU SHARMA[2], AND SYED AZEEMUDDIN[1,3], (Senior Member, IEEE)

[1]Center for VLSI and Embedded Systems Technologies (CVEST), International Institute of Information Technology Hyderabad, Hyderabad 500032, India
[2]Machine Learning Laboratory (MLL), International Institute of Information Technology Hyderabad, Hyderabad 500032, India
[3]Department of Electrical and Computer Engineering, University of Mississippi, Oxford, MS 38677, USA

Corresponding author: Prayushi Mathur (prayushi.m@research.iiit.ac.in)

ABSTRACT Given the current emphasis on maintaining and inspecting high-rise buildings, conventional inspection approaches are costly, slow, error-prone, and labor-intensive due to manual processes and lack of automation. In this paper, we provide an automated, periodic, accurate and economical solution for the inspection of such buildings on real-world images. We propose a novel end-to-end integrated autonomous pipeline for building inspection which consists of three modules: i) Autonomous Drone Navigation, ii) Façade Detection, and iii) Model Construction. Our first module computes a collision-free trajectory for the UAV around the building for surveillance. The images captured in this step are used for façade detection and 3D building model construction. The façade detection module is a deep learning-based object detection method which detects cracks. Finally, the model construction module focuses on reconstructing a 3D model of a building from captured images to mark the corresponding cracks on the 3D model for efficient and accurate inferences from the inspection. We conduct experiments for each module, including collision avoidance for drone navigation, façade detection, model construction and mapping. Our experimental analysis shows the promising performance of i) our crack detection model with a precision and recall of 0.95 and mAP score of 0.96; ii) our 3D reconstruction method includes finer details of the building without having additional information on the sequence of images; and iii) our 2D-3D mapping to compute the original location/world coordinates of cracks for a building.

INDEX TERMS Autonomous, UAV, façade, detection, building.

I. INTRODUCTION

For decades in the infrastructure industry, the economic setting has promoted new construction favouring suburban growth. Now, the main focus is on the maintenance, surveillance [1] and rehabilitation of existing structures [2]. Building structures suffer from bending, buckling, compressive and tensile failures. There are high chances of such failures in buildings due to their age, density and altitude. To prevent structural collapse, casualties and economic loss, periodic maintenance is essential for the safety of buildings to increase their durability and lifespan. The inspection requires heavy machinery, lifts, field professionals and people rappelling from dangerous heights, which is labour-intensive and time-consuming. Additionally, professionals traditionally inspect the buildings using climbing gear, swing stages and access tools. They further spend on insurance and labour, which increases the inspection cost. Even after all these efforts and expenses, some surfaces' comprehensive details are unavailable due to the inaccessibility to some parts of surfaces.

In the current era, Unmanned Aerial Vehicles (UAVs) can accurately, efficiently and economically perform a wide range of surveying applications [2] such as transmission-line inspection [3], power-line inspection [4], underwater area inspection [5], tree cavity inspection [6], industrial plants [7], bridge inspection [8], dam inspection [9], aircraft skin cracks [10], turbine blades [11] and generalised for civil structures [12] for Structural Health Monitoring (SHM). To keep professionals out of danger, UAVs can provide aid for

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaogang Jin.

building inspection. This can reduce operational costs, human error, safety risks and time taken for the inspection. UAVs can fly and scan the entire structure for evaluation, owing to their mobility and ability to capture footage by applying appropriate path planning for the whole structure [13].

Classical methods [14] comprise detecting cracks manually which is painstakingly time-consuming and is biased by the subjective judgment of the inspectors. With the advent of computer vision in the past decade, the traditional surveying task can be aided by such techniques for insightful and accurate inspections [15]. Crack detection [16] has gained popularity due to the adverse effects of buildings not being monitored periodically. Computer Vision methods [17], [18], [11], and [19] and physical interaction methods [7], [20], [21] have been assisting in inspection to have economic and periodic maintenance. The footage captured by the UAV can also be used in Building Information Modeling (BIM) [22] for 3D rendering [13]. This serves as an aid to the Architecture, Engineering and Construction (AEC) industry. The benefit of working on a 3D model over traditional drafting is to allow the building inspectors to understand the condition of the façades.

The aforementioned methods implement a particular stage or a few stages of the building inspection problem. We propose a novel end-to-end pipeline for high-rise building inspection, as shown in Fig. 2. Our first module generates the trajectory for the UAV. Then, collision avoidance ensures that the UAV reaches the destination despite encountering obstacles by recomputing its trajectory on the fly. During the flight, the UAV captures images of the building based on camera and UAV parameters. The images captured are used for crack detection and to reconstruct a 3D mesh model of the building and mark the detected crack correspondences on the model. Fig. 1 shows an example of a 3D mesh model of the building constructed with our proposed pipeline using real-world images, the trajectory of drone navigation and a detected crack. Our major contributions are as follows: i) To the best of our knowledge, we are the first to propose an automated pipeline for building inspection; ii) We propose a novel end-to-end framework for high-rise infrastructure inspection for façade detection; iii) We also integrate the entire pipeline for autonomous inspection using UAVs; iv) We create a dataset and empirically study real-world building data for crack detection, 3D reconstruction and respective mapping.

The rest of the paper explains the method in Section II, implementation details in Section III, analysis of the crack detection module in Section IV and conclusion in Section V.

II. METHOD

This section combines multiple modules to create a unified and cohesive solution for high-rise building inspection. Firstly, we identify various modules involved in the building inspection, such as autonomous drone navigation, façade detection and model construction. Our proposed pipeline integrates these modules as shown in Fig. 2 and explains each framework in the following sections and supplementary material.

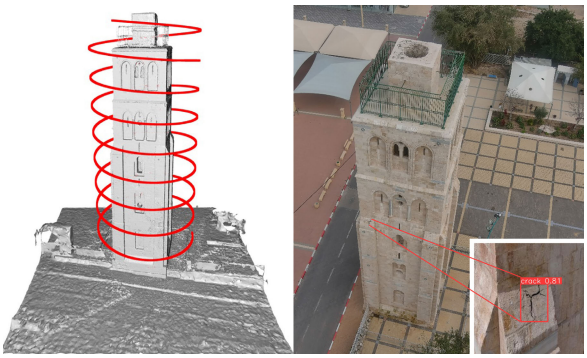

FIGURE 1. Our reconstructed 3D mesh model (left) of the high-rise building (right) with the detected crack (bottom right). The proposed pipeline reconstructs the 3D model from the real-world images captured by UAV, where the red spiral path marks the trajectory followed by the UAV.

A. AUTONOMOUS DRONE NAVIGATION

1) TRAJECTORY PLANNING

For inspection, we develop an autonomous trajectory plan to scan the building under inspection from bottom to top. Our algorithm takes the corner coordinates, $\{(x_i, y_i)|i = 1, 2, 3, 4\}$ and the height, h_b of the building as input to plan the trajectory. The UAV trajectory is planned in an elliptical spiral path around the building. In Fig. 3, the inner rectangle represents the building ground plane. We take a distance, s from the building, as the drone should be at a minimum safety distance to capture images. The rectangle $ABCD$ maintains this distance, s from the building. For minimum distance, we plan an ellipse spiral trajectory passing through $ABCD$. Fig. 3 represents the 2D projection of the trajectory. Equation (1) shows the equation of an ellipse with center (h, k) and radii a and b.

Consider a circle around a square, where the circle touches all four corners of the square. Here, the radius, r of the circle, is the same as the semi-diagonal of the square. If we squeeze the square from either of the sides, we get a rectangle. As an effect, the circle gets squeezed into an ellipse keeping the ratios the same. Therefore, in equation (2), we get the radii, a and b, of the base ellipse for the spiral trajectory path.

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \tag{1}$$

$$a = \frac{x_{AB}}{\sqrt{2}} \qquad b = \frac{y_{BC}}{\sqrt{2}} \tag{2}$$

$$x_{AB} = (x_2 - x_1) + 2s \qquad y_{BC} = (y_3 - y_2) + 2s \tag{3}$$

$$a = \frac{(x_2 - x_1 + 2s)}{\sqrt{2}} \qquad b = \frac{(y_3 - y_2 + 2s)}{\sqrt{2}} \tag{4}$$

Combining equations (1) and (4), we get the equation of an ellipse which is used by the UAV to form a spiral path till the

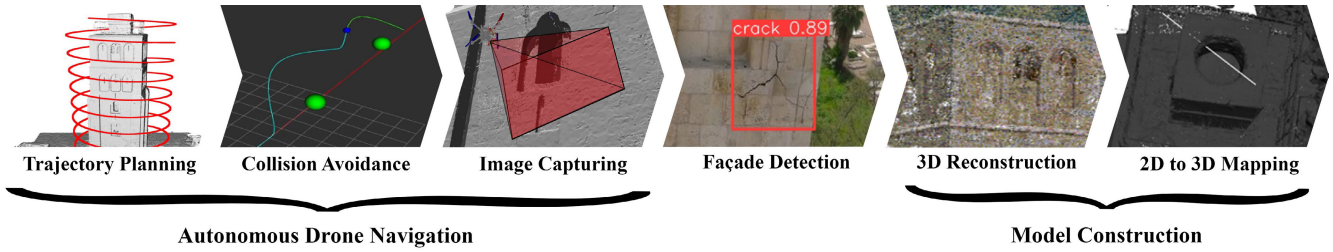

FIGURE 2. Proposed building inspection pipeline consists of three modules: (1) Autonomous Drone Navigation containing (a) Trajectory Planning, (b) Collision Avoidance and (c) Image Capturing; (2) Façade Detection; (3) Model Construction containing (a) 3D Reconstruction and (b) 2D to 3D Mapping.

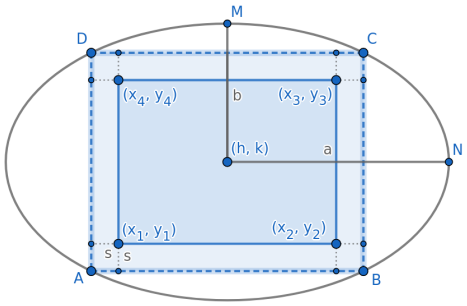

FIGURE 3. 2D projection of trajectory planning on the XY plane. The top view of the building is represented by the inner rectangle and ellipse trajectory followed by the UAV maintaining a safe distance, *s*, from the building.

height, h_b. Along with the radii of the ellipse, the pitch of the spiral path is needed to decide the final trajectory around the building. The pitch is decided based on the distance, s and camera parameters, which include the field of view, aperture, focal length, image resolution and ISO of the camera.

2) COLLISION AVOIDANCE

To consider different types of protrusions of a building, like balconies, antennas and other possible obstacles, we need collision avoidance [23]. Such obstacles must first be detected for complete autonomous drone navigation, and the UAV needs to be guided through a new optimized path to reach its goal. Thus, collision avoidance can be divided into Obstacle Detection and Trajectory Optimization.

a: OBSTACLE DETECTION

A depth camera is attached to the UAV to detect upcoming obstacles [24]. We capture the positions of the obstacles in the environment as point clouds obtained from a depth camera. The point clouds are then down-sampled to reduce the number of points for processing and to denoise the data. Object clusters are extracted from the reduced point cloud, and their positions are determined.

b: TRAJECTORY OPTIMIZATION

This module is responsible for computing a collision-free trajectory towards a given goal position. The motion model of the UAV is given as follows:

$$\dot{x_t} = v_{x_t} \quad \dot{y_t} = v_{y_t} \quad \dot{z_t} = v_{z_t} \tag{5}$$

$$\ddot{x_t} = a_{x_t} \quad \ddot{y_t} = a_{y_t} \quad \ddot{z_t} = a_{z_t} \tag{6}$$

Here, $(v_{x_t}, v_{y_t}, v_{z_t})$ represents the velocity of the UAV in x, y, and z directions and $(a_{x_t}, a_{y_t}, a_{z_t})$ represents the acceleration of the UAV in x, y, and z directions. Given the motion model, we assume that there is a low-level controller for the UAV, which takes the next waypoint, velocity and acceleration as input and generates control commands for the UAV to reach that position. Our trajectory optimizer can be defined as follows:

$$\min_{a_{x_t}, a_{y_t}, a_{z_t}} \sum C_g + C_a \tag{7}$$

$$a_{min} \leq a_{x_t}, a_{y_t}, a_{z_t} \leq a_{max} \tag{8}$$

$$(x_t - x_{o,t})^2 + (y_t - y_{o,t})^2 + (z_t - z_{o,t})^2 > (\lambda_o)^2 \tag{9}$$

$$f_l(x_t, y_t, z_t) > 0 \tag{10}$$

$$C_g = (x_{t_f} - x_{goal})^2 + (y_{t_f} - y_{goal})^2 + (z_{t_f} - z_{goal})^2 \tag{11}$$

$$C_a = (a_{x_t})^2 + (a_{y_t})^2 + (a_{z_t})^2 \tag{12}$$

Here, the cost function, C_g given by equation (11), minimizes the distance of the trajectory end position to the goal, and the cost function, C_a given by equation (12), minimizes the acceleration magnitude at each time instant, for maintaining the smoothness of the trajectory. The inequality constraint in equation (8) limits the accelerations to their maximum and minimum bounds, equation (9) enforces collision avoidance of the i^{th} obstacle with the UAV by enforcing the euclidean distance between them to be greater than the threshold λ_o and equation (10) ensures that the trajectory does not cross the lane boundaries defined by the building walls, which is represented by the function, f_l.

3) IMAGE CAPTURING

While the UAV moves around the building, as shown in Fig. 1, a camera facing the building is dedicated for capturing images. These images are captured by keeping the camera settings constant throughout the flight of the UAV. This helps in keeping the intrinsic parameters of the camera identical for all the images of a particular building. While capturing the images, the frame rate of the camera is set such that the overlap in the consecutive images is at least 60% to 70%.

FIGURE 4. A random sample of images captured of the high-rise building by the UAV.

A sample of the images used for simulation is shown in Fig. 4. The total number of images captured of the building is 2123 for the 3D model construction, of which 175 images contain cracks.

The resolution of the images is chosen such that the façades of the building are clearly visible. The visibility is based on the distance of the UAV from the building, which is considered during the trajectory planning of the UAV (Section II-A1). The camera used for this purpose should not have an active auto-focus facility as the camera's intrinsic parameters change when the camera auto-focuses separately for all the images. Our pipeline requires the camera intrinsics for all the images of a particular building to be the same for 3D construction, elaborated in Section II-C1.

B. FAÇADE DETECTION

A building has various types of façades on the walls. These façades can be cracks, de-laminations, paint deteriorations, moulds and stains. In this paper, we focus on detecting cracks using our model so that those which need attention don't get neglected due to probable human error. To build an autonomous crack detection system for high-rise buildings, various issues occur, such as the texture of the building, lighting conditions and severity of the crack. We use YOLOv5 [25] deep learning model for object detection to detect cracks with bounding boxes in our pipeline.

We use YOLOv5-Large among the YOLOv5 variations available. It is 89MB containing 267 layers with 46.5 million parameters. It is originally trained on the COCO dataset [26].

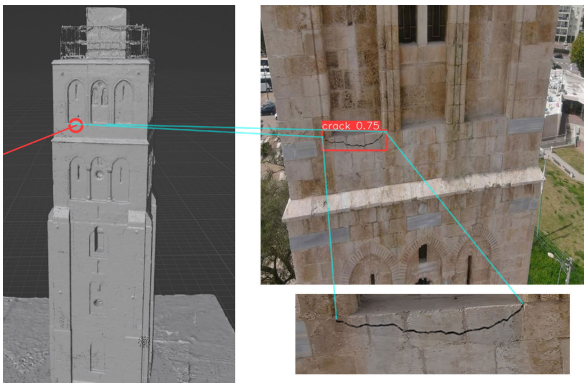

FIGURE 5. 3D mesh model of the building (left) with marked crack using 2D to 3D mapping shown using the red ray and red ring; Captured image of the building (top-right) using UAV showing the crack predicted using YOLOv5 [25]; Zoomed-in Crack (bottom-right); Correspondences of all the three images are shown using blue lines.

Further, we apply transfer learning to this model using real-world images. The train, validation and test set split is 80%, 10% and 10%, respectively. The base model is fine-tuned on the training images for the model to understand the crack structures on the building. The predictions of the final fine-tuned model are shown in Fig. 7 on the test set.

C. MODEL CONSTRUCTION

1) 3D RECONSTRUCTION

Now, we have the images captured from the UAV. We use these images for constructing a 3D model of the building.

This model helps in the visualisation of the building and its detected façades (Section II-B).

a: CAMERA PARAMETER EXTRACTION

For constructing the 3D model, the intrinsic and extrinsic camera parameters are required. These are calculated using a standard traditional method called COLMAP [27], [28]. COLMAP is a Structure-from-Motion (SfM) and Multi-View Stereo (MVS) method which takes multi-view images as input. It uses the overlap patches in distinct unordered images for feature matching. This gives the camera settings while capturing the images (intrinsic camera parameters) and the camera locations for all the images (extrinsic camera parameters). These camera parameters are used for the reconstruction of the building.

b: RECONSTRUCTION

Firstly, the camera parameters generated above are used to reconstruct a dense point cloud using COLMAP. Then, Poisson-disk sampling [29] is applied to sub-sample the generated point cloud. Normals are then computed for each of the remaining points for constructing the 3D mesh model of the object using Marching Cubes APSS algorithm [30], [31]. This gives a fine triangular mesh of the building. This fine high-density mesh is used to produce accurate results as it has more number of faces per unit area. Different views of the reconstructed 3D mesh model are shown in Fig. 1, Fig. 2, Fig. 5 and Fig. 6.

2) 2D TO 3D MAPPING

The façades detected in Section II-B are mapped on the 3D model generated in Section II-C1 for better inference and study of the detected façades by the experts described as follows:

a: IMAGE CORRECTION

The 2D images captured in section II-A3 as shown in Fig. 4 have image distortion where images appear to be curved or deformed due to lens aperture. This is known as image distortion, which is majorly of two kinds: tangential distortion and radial distortion. Tangential distortion is caused due to misalignment of the camera lens with respect to the parallel image plane. This results some areas to appear nearer than expected and can be corrected using equations (13) and (14). In radial distortion, the straight lines in an image appear to be curved and as the point in the real world is farther from the image center, the radial distortion increases. This needs to be corrected before working on images, and can be corrected using equations (15) and (16).

$$x_{tangential} = x + [2p_1xy + p_2(r^2 + 2x^2)] \tag{13}$$

$$y_{tangential} = y + [p_1(r^2 + 2y^2)2p_2xy] \tag{14}$$

$$x_{radial} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \tag{15}$$

$$y_{radial} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \tag{16}$$

TABLE 1. Specifications of RGB and depth camera mounted on Pelican Quadrotor used for simulation.

Specifications	
Image Resolution	752×480
Frame Rate	30 fps
FOV	80°
Min. Depth	0.02m
Max. Depth	30m

So, here we find that five parameters are required for distortion correction. Distortion coefficients are given by $coef_{distortion}$ as follows:

$$coef_{distortion} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \tag{17}$$

The distortion coefficients are obtained from camera intrinsic parameters, which we extract in Section II-C1 from the images captured in Section II-A3. Camera intrinsic parameters also include information like focal length (f_x, f_y) and optical centers (c_x, c_y) given in equation (18). There are camera extrinsic parameters which provide the rotation parameters, r_{ij} and translation parameters, t_i of the images as shown in equation (18). Thus, the final projection of the image after correcting the image distortion is given by λ as follows:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{18}$$

b: RAY TRACING

The façades detected on the 2D images in Section II-B are to be marked on the 3D model generated for a better understanding of the façade locations and their criticality. This is implemented using Ray Tracing. This method requires the camera intrinsic and extrinsic parameters extracted using COLMAP in Section II-C1 and the corresponding images containing the detected façades. Considering the origin of the rays to be at the location of the camera, the rays are extrapolated to the points in the 3D model through the façade in the 2D image. Multiple rays are projected from the camera location towards the scene. The ones which pass through our desired pixel in the image are visualised from the camera to the detected façade in the 3D model of the building.

Finally, our pipeline integrated with all the modules helps to autonomously monitor building inspection, as shown in Fig. 6.

III. IMPLEMENTATION DETAILS

We use the Gazebo framework for simulation on a 2.25 GHz system with 16 GB RAM, an Intel i7 processor and Nvidia RTX 3060 GPU. The 3D mesh model constructed in Section II-C1 from the images captured using UAV, is imported into Gazebo [32].

In the simulation environment, a Pelican Quadrotor is spawned with two cameras to follow the trajectory planned

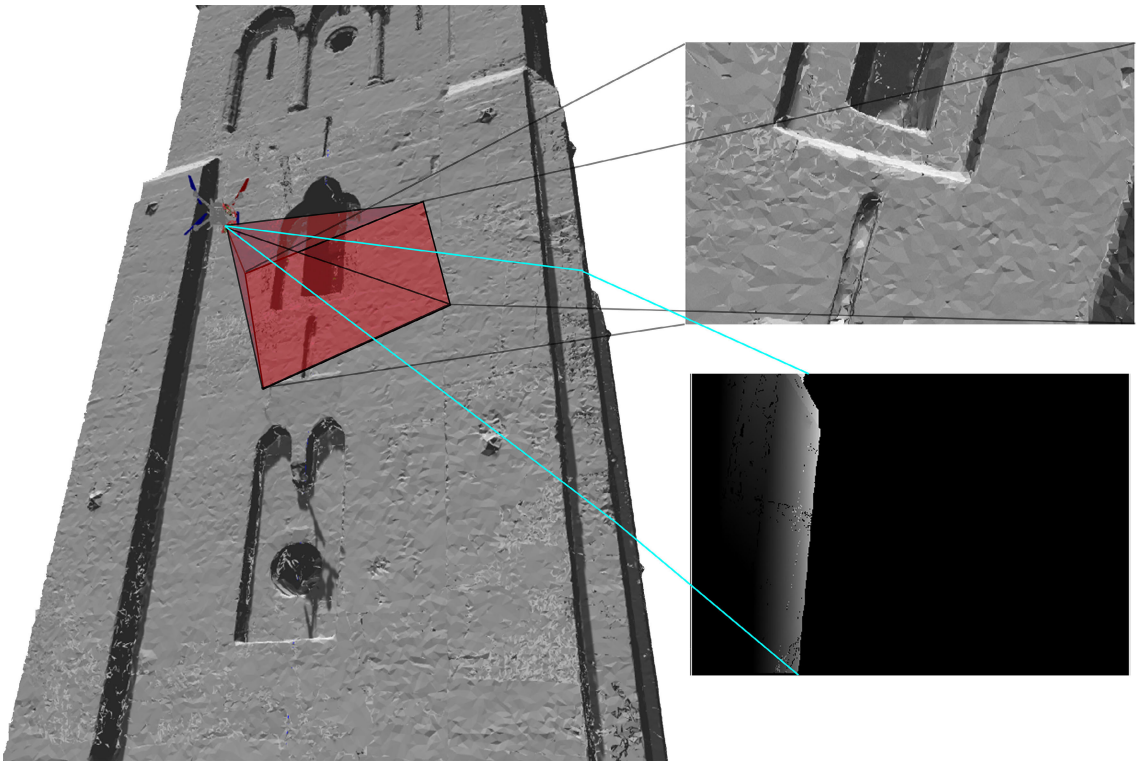

FIGURE 6. The final simulation of the autonomous drone navigation on our high-rise building 3D reconstruction (left); the RGB camera view of the drone (top right); the depth camera view of the drone (bottom right).

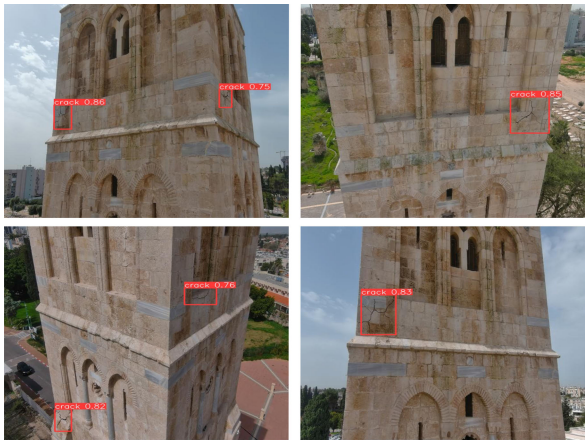

FIGURE 7. The cracks detected in the building images are detected using a fine-tuned model based on Yolov5 [25].

in Section II-A1. The specifications of the both the cameras are mentioned in Table 1. Both the cameras are mounted on the Pelican, orthogonal to each other where the RGB camera faces the building model to capture images and the depth camera faces towards the path to optimize its path for collision-free flight. An example of the view is shown in Fig. 6. Optimal control solver ACADO [33] is used for trajectory optimization in Eq. (7). We use Point Cloud Library (PCL) to obtain obstacle positions from the point cloud.

We annotate the cracks in the captured images for façade detection. The YOLOv5-Large model is fine-tuned on the images with cracks. For the YOLOv5-Large model, all these images are resized to 640 × 640 pixels. The fine-tuning takes approximately 55 minutes to train till 233 epochs with the model size of 92.8MB. The model, initially supposed to train for 500 epochs stops training early at 233 epochs for the best model. This happens as no improvement is observed in the model post 132 epochs.

The captured images are also used for 3D model reconstruction of the high-rise building. Initially, a dense point cloud of 1,384,433 points taking 20.8MB is generated. The generated triangular 3D mesh of the high-rise building contains 864,584 vertices and 871,137 triangles with a size of 126.8MB, which contains triangular faces, vertices, vertex normals and vertex textures of the generated mesh. The detected cracks are mapped on the 3D model as shown in Fig. 5. Hence, the entire proposed pipeline is integrated in an end-to-end manner for high-rise building inspection.

IV. ANALYSIS

The first step in our pipeline is collision-free drone navigation. The depth camera used here has a maximum depth range of 30 metres. This gives the UAV enough visibility to avoid upcoming obstacles. The UAV takes 200-300 milliseconds to compute the distance and the size of the obstacle to avoid them in real time for safety.

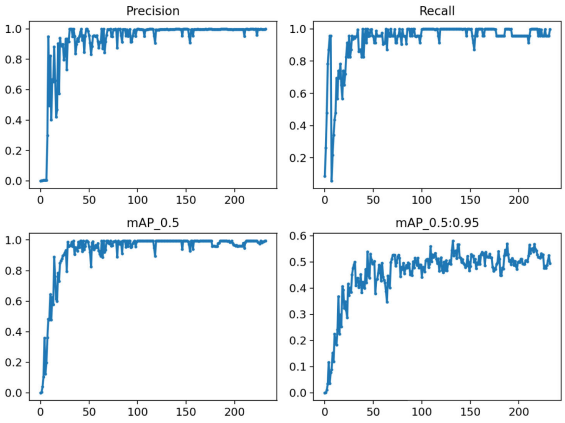

FIGURE 8. Evaluation metrics of the crack detection model.

TABLE 2. Results of the crack detection model.

Metrics	Train	Validation	Test
Number of images	139	18	18
Number of cracks	172	23	23
Precision	1.00	0.995	0.95
Recall	0.998	1.00	0.957
mAP50	0.995	0.995	0.96

The results of façade detection as shown in Fig. 7 show a bounding box around the cracks detected by the YOLOv5 model. Prediction results with evaluation metrics such as precision, recall and mean average precision (mAP) are shown in Fig. 8 and Table 2. Since our detection model is learning from the annotated cracks on the building, it predicts cracks with a recall of 95.7% and an average precision of 96% for unseen images. This gives confidence to the pipeline that our model performs well on a real-world high-rise building.

V. CONCLUSION

In this paper, a novel autonomous inspection pipeline is proposed for buildings to detect cracks with precision and recall of 0.95 and an mAP score of 0.96. Our method includes three modules with autonomous drone navigation, façade detection and model construction. We successfully demonstrated the effectiveness of all three modules as shown in our results. The inspection pipeline proposed in this paper is an end-to-end pipeline to detect cracks on a high-rise building. To the best of our knowledge, we believe this is the first automated pipeline for building inspection. This is highly impactful in the structural engineering domain. For future work, the proposed approach can also be made automated in other domains, such as the automobile industry, agriculture and underwater environments, to name a few.

REFERENCES

[1] P. Mathur, A. K. Singh, S. Azeemuddin, J. Adoni, and P. Adireddy, "A real-time super-resolution for surveillance thermal cameras using optimized pipeline on embedded edge device," in *Proc. 17th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2021, pp. 1–7, doi: 10.1109/AVSS52988.2021.9663831.

[2] T. Rakha and A. Gorodetsky, "Review of unmanned aerial system (UAS) applications in the built environment: Towards automated building inspection procedures using drones," *Autom. Construct.*, vol. 93, pp. 252–264, Sep. 2018, doi: 10.1016/j.autcon.2018.05.002.

[3] J. Bian, X. Hui, X. Zhao, and M. Tan, "A novel monocular-based navigation approach for UAV autonomous transmission-line inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–7, doi: 10.1109/IROS.2018.8593926.

[4] N. Iversen, O. B. Schofield, L. Cousin, N. Ayoub, G. vom Bögel, and E. Ebeid, "Design, integration and implementation of an intelligent and self-recharging drone system for autonomous power line inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4168–4175, doi: 10.1109/IROS51168.2021.9635924.

[5] G. Picardi, R. Lovecchio, and M. Calisti, "Towards autonomous area inspection with a bio-inspired underwater legged robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 930–935, doi: 10.1109/IROS51168.2021.9636316.

[6] K. Steich, M. Kamel, P. Beardsley, M. K. Obrist, R. Siegwart, and T. Lachat, "Tree cavity inspection using aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4856–4862, doi: 10.1109/IROS.2016.7759713.

[7] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3532–3537, doi: 10.1109/IROS.2012.6385917.

[8] A. E. Jimenez-Cano, J. Braga, G. Heredia, and A. Ollero, "Aerial manipulator for structure inspection by contact from the underside," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1879–1884, doi: 10.1109/IROS.2015.7353623.

[9] N. Palomeras, M. Carreras, P. Ridao, and E. Hernandez, "Mission control system for dam inspection with an AUV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, doi: 10.1109/IROS.2006.281705.

[10] C. Wang, X. Wang, X. Zhou, and Z. Li, "The aircraft skin crack inspection based on different-source sensors and support vector machines," *J. Nondestruct. Eval.*, vol. 35, no. 3, pp. 1–8, Sep. 2016, doi: 10.1007/s10921-016-0359-3.

[11] B. E. Jaeger, S. Schmid, C. U. Grosse, A. Gögelein, and F. Elischberger, "Infrared thermal imaging-based turbine blade crack classification using deep learning," *J. Nondestruct. Eval.*, vol. 41, no. 4, p. 74, Dec. 2022, doi: 10.1007/s10921-022-00907-9.

[12] M. Medhi, A. Dandautiya, and J. L. Raheja, "Real-time video surveillance based structural health monitoring of civil structures using artificial neural network," *J. Nondestruct. Eval.*, vol. 38, no. 3, pp. 1–16, Sep. 2019, doi: 10.1007/s10921-019-0601-x.

[13] S. Song and S. Jo, "Online inspection path planning for autonomous 3D modeling using a micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6217–6224, doi: 10.1109/ICRA.2017.7989737.

[14] G. T. Ferraz, J. de Brito, V. P. de Freitas, and J. D. Silvestre, "State-of-the-art review of building inspection systems," *J. Perform. Constructed Facilities*, vol. 30, no. 5, Oct. 2016, doi: 10.1061/(ASCE)CF.1943-5509.0000839.

[15] A. M. Paterson, G. R. Dowling, and D. A. Chamberlain, "Building inspection: Can computer vision help?" *Autom. Construct.*, vol. 7, no. 1, pp. 13–20, Dec. 1997, doi: 10.1016/s0926-5805(97)00031-9.

[16] H. S. Munawar, A. W. A. Hammad, A. Haddad, C. A. P. Soares, and S. T. Waller, "Image-based crack detection methods: A review," *Infrastructures*, vol. 6, no. 8, p. 115, Aug. 2021, doi: 10.3390/infrastructures6080115.

[17] C. Eschmann, "Unmanned aircraft systems for remote building inspection and monitoring," in *Proc. 6th Eur. Workshop Structural Health Monitoring*, 2012, pp. 1–8.

[18] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, Apr. 2019, doi: 10.1016/j.neucom.2019.01.036.

[19] S. Tang and Z. Chen, "Scale–space data augmentation for deep transfer learning of crack damage from small sized datasets," *J. Nondestruct. Eval.*, vol. 39, no. 3, pp. 1–18, Sep. 2020, doi: 10.1007/s10921-020-00715-z.

[20] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart, "Hybrid predictive control for aerial robotic physical interaction towards inspection operations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 53–58, doi: 10.1109/ICRA.2014.6906589.

[21] L. Yang, J. Fan, B. Huo, and Y. Liu, "Inspection of welding defect based on multi-feature fusion and a convolutional network," *J. Nondestruct. Eval.*, vol. 40, no. 4, pp. 1–11, Dec. 2021, doi: 10.1007/s10921-021-00823-4.
[22] F. López, P. Lerones, J. Llamas, J. Gómez-García-Bermejo, and E. Zalama, "A review of heritage building information modeling (H-BIM)," *Multimodal Technol. Interact.*, vol. 2, no. 2, p. 21, May 2018, doi: 10.3390/mti2020021.
[23] F. Gao and S. Shen, "Quadrotor trajectory generation in dynamic environments using semi-definite relaxation on nonconvex QCQP," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6354–6361, doi: 10.1109/ICRA.2017.7989750.
[24] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human–robot collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 338–345, doi: 10.1109/ICRA.2012.6225245.
[25] G. Jocher et al., "ultralytics/YOLOv5: v7.0—YOLOv5 SOTA realtime instance segmentation," Tech. Rep., 2022.
[26] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.
[27] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113, doi: 10.1109/CVPR.2016.445.
[28] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 501–518 doi: 10.1007/978-3-319-46487-9.31.
[29] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 6, pp. 914–924, Jun. 2012, doi: 10.1109/TVCG.2012.34.
[30] G. Guennebaud and M. Gross, "Algebraic point set surfaces," in *Proc. ACM SIGGRAPH Papers*, Jul. 2007, doi: 10.1145/1275808.1276406.
[31] G. Guennebaud, M. Germann, and M. Gross, "Dynamic sampling and rendering of algebraic point set surfaces," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 653–662, Apr. 2008, doi: 10.1111/j.1467-8659.2008.01163.x.
[32] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2004, doi: 10.1109/IROS.2004.1389727.
[33] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," *Optim. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, May 2011, doi: 10.1002/oca.939.

PRAYUSHI MATHUR is currently pursuing the M.S. degree (by research) with the International Institute of Information Technology Hyderabad. She completed her internship at the Government of India as an AI Intern and has worked with ISRO on a project. She has worked across diverse domains, such as computer vision, image processing, medical imaging, satellite imagery, 3D graphics, and robotics for the past few years.

CHARU SHARMA is currently an Assistant Professor with the Machine Learning Laboratory, International Institute of Information Technology Hyderabad. She majorly works in the area of geometric deep learning and has been working on problems related to point cloud data, graph-structured data, assignment problems, optimal transport, and omnidirectional images.

SYED AZEEMUDDIN (Senior Member, IEEE) received the B.E. degree from MJCET, Osmania University, Hyderabad, India, in 2003, and the M.S. and Ph.D. degrees from Southern Illinois University, Carbondale, IL, USA, in 2005 and 2008, respectively. He was a Faculty Member with the International Institute of Information Technology Hyderabad, Hyderabad. He is involved in the development and fabrication of integrated optical gyro-scope, CMOS RF LNA, on-chip RF inductors, and bio-sensors. His current research interests include RFICs, devices, and all-optical devices using ring lasers. He is a member of various professional societies, such as OSA, IETE, and the VLSI Society of India. He received the Gold Medal Award from MJCET, Osmania University; the Master's Fellowship Award and Doctoral Fellowship Award from SIUC, USA; and the Visvesvaraya Young Faculty Award from the Government of India. He has been active in the overall development of society for eternal success. He is a reviewer of various conferences and journals.