

# A Methodology for BIM-enabled Automated Reality Capture in Construction Inspection with Quadruped Robots

Srijeet Halder<sup>a</sup>, Kereshmeh Afsari<sup>a</sup>, John Serdakowski<sup>b</sup>, and Stephen DeVito<sup>b</sup>

<sup>a</sup>Myers-Lawson School of Construction, Virginia Tech, USA

<sup>b</sup>Procon Consulting, USA

Email: [srijeet@vt.edu](mailto:srijeet@vt.edu), [keresh@vt.edu](mailto:keresh@vt.edu), [jserdakowski@proconconsulting.com](mailto:jserdakowski@proconconsulting.com), [sdevito@proconconsulting.com](mailto:sdevito@proconconsulting.com)

## Abstract –

Construction inspection is an important part of the construction management process to ensure that the project is compliant with the requirements, regulations, and standards. Due to time and cost constraints, inspectors representing owners, as well as architects, designers, and other stakeholders might not be able to visit and inspect the project site in person as often as needed. Inspections performed by mobile robots can allow more frequent inspections and can help communicate project status with all stakeholders more regularly. This study explains the development of a methodology to capture the site reality with an autonomous quadruped robot through integrated visualization in a BIM-enabled 3D virtual environment. The proposed methodology can potentially facilitate construction inspectors to perform their day-to-day inspection tasks more efficiently. The proposed methodology consists of a 3D virtual environment integrated with the robot control. The methodology is further implemented in Unity Engine with a software application to visualize the space being inspected and control the robot simultaneously. The application captures 360° images of the space autonomously and visualizes the images in the virtual environment as discrete points of reality. This study can be useful to construction managers and inspectors to potentially reduce manpower in inspections of construction projects. The findings of the study can help develop more solutions to facilitate remote construction inspection using autonomous mobile robots.

## Keywords –

legged robot, construction inspection, virtual environment, reality capture

## 1. Introduction

Construction projects are full of uncertainties. Regular and frequent inspection is indispensable for reducing uncertainties in projects [1]. Multiple people

take part in the inspection of construction projects. It involves them traveling to the site, walking around the site, observing the as-built state of the project, and comparing the as-built status with the documented as-planned requirements of the project. Traveling to the site and walking around the site consumes a lot of time [1] that becomes non-trivial when multiplied by the number of people involved in the inspection process.

Currently, many contractors, project owners, and owner representatives use digital jobsite documentation tools such as Cupix, StructionSite, and Holobuilder [2]. These tools link 360° spherical images to 2D floor plans for spatial visualization of as-built conditions. This approach provides a significant advantage over regular unorganized 2D images by providing a platform for communication and collaboration through annotations and organized as-built documentation [3]. The current challenge is that these 2D floor plans are limited to representing a single discipline (e.g. architectural floor plan). It is difficult to show various construction disciplines, such as architecture, structural, mechanical, electrical, etc. in one floor plan without making it too cluttered and difficult to read. We propose to extend this approach to visualize the as-built conditions on a 3D model of the building enabled by the Building Information Model (BIM) to be used in the reality capture process. 3D visualization is more intuitive and can allow visualization of models of multiple disciplines in one view along with the reality capture.

Automated inspection of construction status has been studied before [1], [4], [5]. However, the data collection part in these previous studies has been predominantly manual, which is time-consuming, costly [6], and inaccurate. This study proposes an automated data capturing and visualization process for site inspection using quadruped robots. The process of data collection in reality capture for construction inspection is a repetitive and tedious task [7] that requires human inspectors to persistently and accurately walkthrough inherently risky construction environments.

Robots can play an important role in automating the inspection process [6]. Robots can be programmed to

autonomously navigate through the construction site and take regular images at points of interest, thus can save productive hours for humans that can be utilized for intelligent and high-level decision-making [6].

The main contribution of this paper is the introduction of a methodology for automated reality capture integrated with BIM-enabled 3D visualization and robot control using a quadruped robot. The proposed methodology does not rely on heavy post-processing of reality capture data and provides a rapid and frequent update of project status to all stakeholders in near-real-time within an improved visualization.

## 2. Background

Previous studies investigated automating construction inspection [6], [8], [9]. These studies aimed to increase the frequency and accuracy of inspection by capturing visual data using semi-autonomous or autonomous systems [6]. These studies either focused primarily on the data capture alone [6], [8], [10] and not on presenting the information in a way that is intuitive for inspection, or used heavy post-processing that requires skilled personnel with specialized training [9], [11]. Another challenge with heavy post-processing of inspection data (i.e. as-built reality capture) is the delay in the availability of inspection data. By the time the reality capture is available to inspectors and project stakeholders, the construction work might have progressed further, and either the identified construction issues might be resolved already or new issues might have arisen. Our study aims to automate the process of reality capture and its visualization process to make key decision-making information available in near-real-time to remote stakeholders.

Capturing 360° images of construction status is a recent trend in reality capture because it makes the capturing process faster [12], by replacing multiple pictures with one 360° spherical image. They are easy-to-use construction inspection tools with low computation costs [12], [13]. They are proven to be effective for general inspection when accuracy and precision are not as important as ease of availability and faster processing [12]. 360° images provide a “sense of presence” [13], which is absent in 2D images. 360° images are also used as a reality backdrop which is augmented with a 3D model on top of them [12]. The proposed methodology in this paper uses a 360° camera to capture visual data of the jobsite. This study automates the process of capturing 360° images through using quadruped robots. Previous studies on automation of construction inspection use either manual data capturing or heavy post-processing [3], [11], [12]. Automated as-built data capturing has been performed before, using Unmanned Aerial Vehicles (UAVs) [14]. BIM was used

as a preloaded map for path planning of the UAV [14]. However, UAVs have many inherent restrictions. First and foremost, legal restrictions prevent UAVs or drones from being used out of sight of the pilot and in populated areas [15], which defeats the purpose of automation. UAVs also need large clearance and are too noisy to be used in indoor spaces [10]. They also have short flight times [10]. Cables, ladders, and scaffoldings add more challenges.

The use of autonomous robots has been studied for construction progress monitoring and inspection applications [6], [7], [16]–[18]. Our previous research [16] studied the fundamentals of quadruped robot applications in construction monitoring and inspection. Additionally, our prior study [19] indicated that existing quadruped robots can potentially enhance the accuracy and speed of reality capture compared to the manual and human-based process. To make robots more autonomous, various path-finding algorithms are developed to enable mobile robots to navigate the job site and reach a target from a specific starting point. A\* search is one of such algorithms [20]. Other algorithms are modified A\* search, genetic algorithm, potential field algorithm, and roadmap algorithm [20].

Construction inspection is an important process and there is a need for fast and frequent access to project information for better inspection of the project and avoid deviations from the planned project requirements. 360° images have been an effective tool for construction inspection, automating the capturing process of these 360° images can improve the efficiency of the inspection process. Therefore, in this study, we used an autonomous robot to capture as-built site information and present it to construction inspectors rapidly through an integrated 3D virtual environment that contains 3D as-planned project data.

## 3. Proposed Methodology for Automated Robotic Reality Capture and Visualization

This section introduces the proposed methodology for automated reality capture and visualization. The methodology relies on the availability of the BIM model for the building. It also uses marker-based localization. We used AprilTag fiducials for this purpose as explained in Section 4. The methodology is applicable to buildings. Inspection of infrastructure projects like dams, bridges, etc. is out of the scope of the current study.

As shown in Figure 1, the BIM model is used to create a 3D virtual environment. The 3D virtual environment serves two purposes. First, it works as a map for the robot by providing geometric data of the space as well as “walkable” surfaces (explained in section 4.2) for the robot’s navigation. Second, it provides the users a 3D

virtual environment to conduct a virtual walkthrough of the building and observe the model as well as the reality capture of the actual job site alongside each other. This allows the inspectors to switch between the model view and reality capture conveniently and compare the two to better identify discrepancies. Traditionally, this process is done by looking at the 2D images or personally visiting the site and comparing the visuals with the site drawings. In more recent solutions, this process is done by using 2D floor plans of one discipline (e.g. architectural floor plan) with marked 360° images.

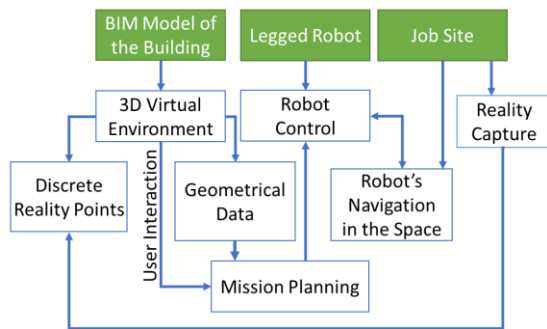


Figure 1 Conceptual overview of the proposed methodology

The proposed methodology in this study is based on a 3-step process – Robot’s Mission Planning, Reality Capture, and Visualization that are integrated into one platform. In the first step, the user creates Goalpoints in the virtual environment. We define Goalpoints as the locations in the space where the robot will navigate autonomously and capture a 360° image for inspection purposes. These Goalpoints and the geometrical data of the space are used to create autonomous missions for the robot. A mission, as we define it, consists of Goalpoints and Waypoints. Waypoints are intermediate locations that the robot will navigate through to reach the Goalpoints while avoiding obstacles. This process is explained in more detail in Section 4.2.

In the second step, i.e. Reality Capture, the robot’s mission information generated in the first step is used by the application to navigate the robot to the Goalpoints by walking on the planned path. Once reached the Goalpoint, the robot triggers the 360° camera to capture a spherical image.

In the third step, the 360° images are captured and transferred from the camera and visualized in the 3D virtual environment. The images are spatially linked to the 3D virtual space. The user or the inspector can perform a virtual walkthrough in the 3D virtual environment and compare the captured images with the 3D model for regular inspection.

## 4. Implementation

The proposed methodology is implemented using the Unity engine. Unity’s 3D visualization and scripting capabilities have been used previously by researchers for inspection and monitoring of buildings [21], [22].

The hardware used in this study included Spot, an autonomous quadruped robot by Boston Dynamics, and a Ricoh Theta V 360° camera mounted to the top of the robot. Spot’s Python SDK provides many high-level functions to control the movement of the robot and its navigation through the space and it has been used in this study. However, Unity uses C# programming language as its default scripting language. To enable data exchange between the two languages, the two programs use transfer control protocol (TCP) socket programming, similar to [23]. Robot Operating System (ROS) can be an alternative to TCP in future implementation.

The following subsections explain the implementation process of the proposed methodology in more detail.

### 4.1. Overall Workflow

Figure 2 shows the overall workflow of the proposed methodology. As discussed in Section 3, this study uses a 3D model of the building for both visualizing the site images and navigating the robot. The 3D model is extracted from the BIM model as an FBX file using the TwinMotion plugin in Revit. The FBX file also contains some of the component properties including the materials and textures of the building elements in the model.

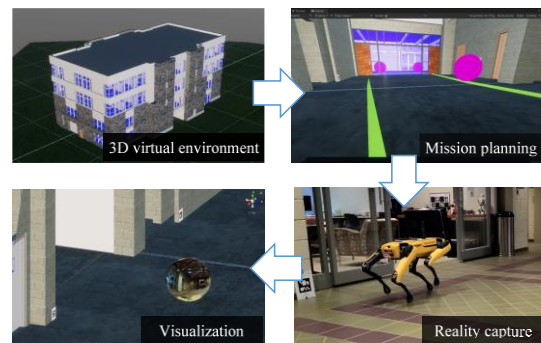


Figure 2 Workflow of the proposed methodology

The first step in the process is Mission Planning. Once the Goalpoints are created by the user where the images should be captured, the optimum path for the robot is calculated in Unity. This process is explained in more detail in Section 4.2. The mission information generated in the first step is used to control and navigate the robot. The robot walks to each Goalpoint on the planned path and captures 360° images. This is explained in more detail in Section 4.3. Finally, the captured images are downloaded from the camera using the camera’s SDK and visualized inside the 3D virtual environment. The user can then, do a virtual walkthrough in the 3D virtual

environment integrated with 360° images and observe the space in both as-built and as-planned conditions. This step is explained further below in Section 4.4.

## 4.2. Mission planning

Mission planning starts with localizing Spot's initial position in the 3D virtual environment. This study uses fiducial-based localization [24]. Spot has the built-in capability to detect AprilTag fiducials, an example of which is shown in Figure 3. The application maintains a list of all fiducials observed through the robot's cameras in the memory along with their position relative to the robot. At any point in the space, the relative position of the robot with respect to the fiducials is read from the robot using the robot's SDK.

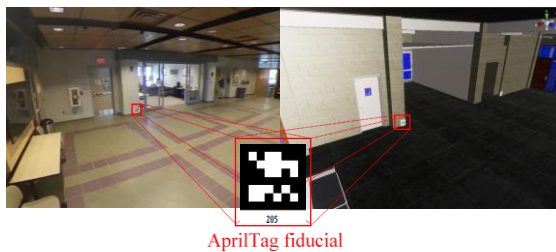


Figure 3 AprilTag fiducials in the job site (left), and on the corresponding location in the 3D model (right)

This study uses Unity's built-in NavMesh module, which in turn uses the A\* heuristic algorithm to find the best path [25]. Once the robot is localized in the space, the application iterates over the predefined Goalpoints and finds the optimum path covering all the points. This operation is done by calling Unity's "NavMeshAgent.CalculatePath" function. The Goalpoints are traversed in the order they are created by default and the application does not try to minimize the travel distance. However, the points can be reordered by the user before starting the mission.

The Unity NavMesh requires the "walkable" and "non-walkable" spaces to be explicitly defined in the application using the 3D model of the space. All the elements from the floor and door families in the BIM model are set as walkable, while everything else is non-walkable. The safe distance from the non-walkable surfaces is set to 0.3m, which is slightly more than half of the width of the robot. After setting the parameters, Unity creates a mesh of walkable points in a process called 'baking'. Figure 4 shows the final baked NavMesh. The light blue areas on the floor shown in the figure are the walkable areas.



Figure 4 Walkable spaces defined in Unity

The application packages the mission data into a JavaScript Object Notation (JSON) string to be sent to the Python application for further processing. The mission data consists of the coordinates of the points on the path. The points on the path are either Goalpoints or Waypoints as explained in Section 3. The images are captured only at the Goalpoints, and the Waypoints only help with navigating the robot around the obstacles, such as walls. Waypoints are created by Unity at each location where the direction of the path changes. The number of Waypoints depends on the obstacles between the starting point of the robot and the following Goalpoints. Figure 5 shows a snippet of a sample JSON string representing a mission. The value 'isGoal' for the points states whether it is a Goalpoint or not.

```
{
  "mission": {
    "fiducials": [
      {
        "id": "205",
        "points": [
          {
            "x": -0.11544030904769898,
            "y": 0.3089998662471771,
            "z": 2.905001163482666,
            "isGoal": false
          },
          {
            "x": -0.11544030904769898,
            "y": -10.651000022888184,

```

Figure 5 Sample mission JSON data

## 4.3. Execution and Reality capture

The mission data in the form of JSON string is received in Python for further processing. First, the fiducials are identified from the JSON data. The robot's internal system is queried to find all the fiducials that are detected by the robot's perception system. As soon as a fiducial is observed by the robot, its location is stored in a list called 'known\_fiducials'. During the initiation of the mission, there must be at least one fiducial in the 'known\_fiducials', otherwise, the mission fails with an error. The closest fiducial from the robot's current



location is identified by finding the magnitude of the position vector of the fiducial. Then, the coordinates of the next point are read from the mission data. The coordinates are transformed to the robot's body frame of reference. The application uses the Spot's SDK to control the robot and iteratively checking its new position until the robot reaches the next Goalpoint or Waypoint. Spot's SDK abstracts lower-level controls of the limbs. Only the vector representing the target coordinates from the robot's current position are required to move the robot. Robot's built-in obstacle avoidance system prevents collision with any dynamic or static obstacles that are not represented in the model. However, the robot cannot re-route itself if the intended path is completely blocked. This is currently the limitation of the project and will be addressed in the future.

A radial tolerance of 0.5m is allowed in the application. This means the robot will start moving to the next Goalpoint (or Waypoint) as soon as it reaches within 0.5m of the current Goalpoint (or Waypoint). If the current point is a Goalpoint, the robot will stop and trigger the 360° camera. The camera must be connected to the robot's internal Wi-Fi network. The program waits for the camera to process the image by reading the camera's state using the camera's SDK. Once the image is processed and stored, the robot continues its mission. Figure 6 shows the robot executing its mission and presents its real-time virtual representation in the 3D virtual environment.



Figure 6 The robot in the job site (top), and the 3D virtual environment (bottom) visualized in real-time simultaneously

#### 4.4. Visualization

Once the whole mission is complete, i.e., the robot

reaches the last point in its path, all captured images are downloaded from the camera. The images are applied as a texture to spheres in the 3D virtual environment representing the Goalpoints. These spheres represent bubbles as Goalpoints that contain the reality capture data. The user can then use the 3D virtual environment and navigate through the building model virtually to enter any of the bubbles/Goalpoints to view the 360° reality images from that location. When the user moves out of the Goalpoint in the 3D virtual environment, the 3D building model (i.e. as-planned BIM data) will be visible.

Figure 7 shows the interface of the application developed in Unity. The spheres in the 3D virtual environment shown in Figure 7 are each a 360° spherical image captured by the robot from the corresponding location inside the job site. The 360° images captured through the robot's autonomous mission are available to inspectors and project stakeholders on the developed platform in real-time after the robot's mission is complete.

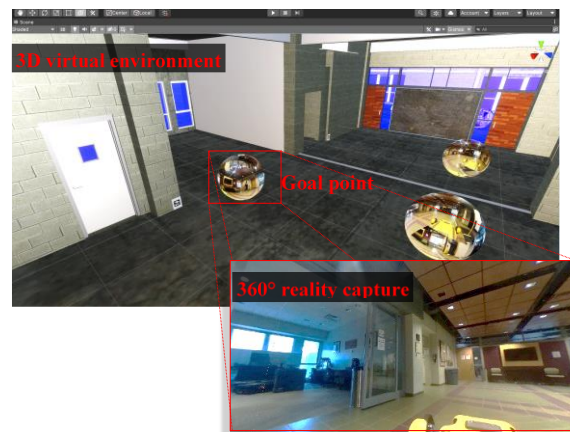


Figure 7 Visualizing reality capture integrated with the 3D virtual environment. On top: 3D virtual environment developed in Unity using the BIM model, on the bottom: a snapshot of the captured 360° image of the reality on site

## 5. Validation and Results

To test the validity and feasibility of the proposed methodology, the developed application is tested in Bishop-Favrao Hall (BFH) on Virginia Tech's campus in Blacksburg, VA. A test mission was executed that involved walking through a space of approximately 4,841 sqf. in size including two classrooms and the entrance lobby and captured 360° images at six Goalpoints as shown in Figure 8. The total length of the robot's path was 42.4m. The robot successfully executed the mission within 3 minutes and 50 seconds with the moving speed set to 0.4m/s. The robot captured and sent back six 360° images for visualization in the 3D virtual environment.

Figure 8 shows the 2D floor plan of BFH with the Goalpoints and the path of the test mission in the experiment.

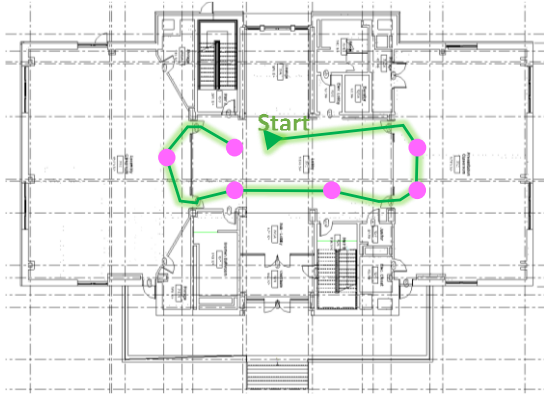


Figure 8 Robot's mission path and Goalpoints on the 2D Floor Plan of BFH

In this experiment, installing the fiducials at the site true to their corresponding position in the 3D model was found challenging. Small residual errors in the orientation of the fiducials created large deviations in the navigation of the robot as it moved farther from the fiducials. Installing more fiducials helped to keep the deviations small. The required number of fiducials depends on the length of the path. Spot's perception has a range of about 4m. Therefore, installing a fiducial every 8m on the robot's path ensures that at least one fiducial is always visible to the robot. For that reason, five fiducials were installed for the whole path at an average interval of 8m.

During the test mission, passing through the doors was found to be challenging. Due to the narrow clearance, the path occasionally overlapped with the walls due to the errors in placement of the fiducials as well as the localization error arising from the motion of the robot. Placing a fiducial near the doors improved the robot's navigation through the doorways.

## 6. Discussion and Conclusion

Regular construction inspection is crucial for project success. However, construction inspection involves many repetitive tasks that are better suited for robots. Delegating repetitive and mundane tasks to robots can potentially make the process of inspection and monitoring more effective and can allow humans to focus on more high-level tasks. For example, legged robots can traverse and scan the jobsite more frequently than humans while humans can undertake more intelligent planning and decision making tasks.

This paper is built upon the authors' previous studies on the use of quadruped robots for construction inspection [16], [19] that presented the potential and

challenges of autonomous quadruped robots in construction inspection. This paper presents the development of an autonomous data capturing and visualization method for construction inspection using a quadruped robot. This study uses the BIM model of the building for (a) mission planning of the robot, and (b) for visualizing both the building and the reality capture in a 3D virtual environment. Robot navigation through the BIM model is the key to reliable autonomous robotic construction site documentation to ensure completing the robot's mission in dynamically changing construction environments.

The contribution of this study is the introduction of a methodology for automated capturing and visualization of the as-built images of a building integrated with a 3D virtual environment of the space enabled by BIM. The study aims to provide a virtual platform to construction management personnel, owners, and their representatives for frequent and remote inspections.

This study uses quadruped robots because they are versatile and can navigate construction sites with more flexibility than wheeled robots. A software solution is developed on Unity Game Engine and Python. The integration of the 3D visualization and robot control was implemented seamlessly in Unity which allowed the development of remote autonomous mission planning. For hardware, the Spot robot from Boston Dynamics was used along with a Ricoh Theta V 360° camera. Spot's SDK provides high-level control functions for the robot that allowed the research team to focus more on mission planning and visualization instead of developing the control system for the robot.

The proposed methodology was validated by running autonomous test missions in Bishop-Favrao Hall on the Virginia Tech campus in Blacksburg, VA. Some of the challenges faced in the initial runs (as explained in Section 5) were addressed in the later revisions of the application and iterative experimental investigations. In this methodology, the only input provided by the user is the location of the Goalpoints where the inspection images are captured on the 3D virtual environment. Future studies will evaluate the developed methodology in multiple scenarios on construction sites with dynamic obstacles. The developed methodology can be further augmented by a more data-rich BIM model of the building to be utilized for creating immersive Virtual Reality (VR) environments to help with inspection tasks in comparing and analyzing the as-built vs. as-planned project data.

### 6.1. Limitations and future studies

The major limitation of the study is that it requires an accurate and exact model of the building for navigation. Additional furniture or closed doors can cause the robot to fail its mission. Our ongoing research is identifying

ways to cater to the additional obstacles encountered by the robot in real-time (e.g., closed doors) and re-routing the mission when required. This feature will make the autonomous missions more robust and capable of handling the highly dynamic nature of construction sites.

Future studies may investigate extending this approach with immersive virtual reality and augmented reality solutions to provide an immersive experience to construction managers and owners of the construction project.

Another limitation of the current study was that the developed application and the robot were connected to the same network. Future studies may separate the user interface and the robot across the internet to allow fully remote inspection. Executing the mission remotely across the internet will not require any complex changes in the developed application but only an additional communication layer running on a cloud server to relay the commands from the controller machine to the robot.

Future studies might also evaluate the usability of the proposed approach in hands-on experiments with inspectors, construction experts, and owners to study the adoption and acceptance of the proposed technology.

## 7. References

- [1] M. Kopsida, I. Brilakis, and P. Vela, "A Review of Automated Construction Progress and Inspection Methods," in *Proceedings of the 32nd CIB W78 Conference on Construction IT*, 2015, no. January, pp. 421–431.
- [2] R. Eadie, J. Proctor, and Y. Ivanov, "Applications for Virtual and Reality Modelling for Construction:," in *XI International Scientific Conference - Civil Engineering Design and Construction*, 2020, p. 9.
- [3] R. Eiris Pereira and M. Gheisari, "360-Degree Panoramas as a Reality Capturing Technique in Construction Domain: Applications and Limitations," 2019, [Online]. Available: <https://par.nsf.gov/biblio/10109436>.
- [4] M. Golparvar-Fard, F. Peña-Mora, and S. Savarese, "D4AR-a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication," *J. Inf. Technol. Constr.*, vol. 14, no. 13, pp. 129–153, 2009.
- [5] C. Gordon, B. Akinci, and J. H. Garrett, "Automated planning support for on-site construction inspection," *Autom. Constr.*, vol. 17, no. 6, pp. 705–718, 2008, doi: 10.1016/j.autcon.2007.12.002.
- [6] K. Asadi *et al.*, "Vision-based integrated mobile robotic system for real-time applications in construction," *Autom. Constr.*, vol. 96, pp. 470–482, Dec. 2018, doi: 10.1016/j.autcon.2018.10.009.
- [7] S. A. Prieto, B. de Soto, and A. Adan, "A Methodology to Monitor Construction Progress Using Autonomous Robots," in *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, Oct. 2020, pp. 1515–1522, doi: 10.22260/ISARC2020/0210.
- [8] H. M. La, T. H. Dinh, N. H. Pham, Q. P. Ha, and A. Q. Pham, "Automated robotic monitoring and inspection of steel structures and bridges," *Robotica*, vol. 37, no. 5, pp. 947–967, 2019.
- [9] P. Kim, J. Chen, and Y. K. Cho, "SLAM-driven robotic mapping and registration of 3D point clouds," *Autom. Constr.*, vol. 89, pp. 38–48, 2018, doi: <https://doi.org/10.1016/j.autcon.2018.01.009>.
- [10] Y. Li and C. Liu, "Applications of multirotor drone technologies in construction management," *Int. J. Constr. Manag.*, vol. 19, no. 5, pp. 401–412, 2019.
- [11] M. Golparvar Fard, F. Peña Mora, and S. Silvio, "D4AR-A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication," *Electron. J. Inf. Technol. Constr.*, vol. 14, no. June, pp. 129–153, 2009.
- [12] R. E. Pereira, H. I. Moud, and M. Gheisari, "Using 360-degree interactive panoramas to develop virtual representation of construction sites," in *Proceedings of the Joint Conference on Computing in Construction (JC3)*, 2017, pp. 775–782.
- [13] R. Eiris, M. Gheisari, and B. Esmaeili, "PARS: Using augmented 360-degree panoramas of reality for construction safety training," *Int. J. Environ. Res. Public Health*, vol. 15, no. 11, p. 2452, 2018.
- [14] A. Ibrahim, D. Roberts, M. Golparvar-Fard, and T. Bretl, "An Interactive Model-Driven Path Planning and Data Capture System for Camera-Equipped Aerial Robots on Construction Sites," in *Computing in Civil Engineering 2017*, 2017, pp. 117–124.
- [15] H. I. Moud, X. Zhang, I. Flood, A. Shojaei, Y. Zhang, and C. Capano, "Qualitative and Quantitative Risk Analysis of Unmanned Aerial Vehicle Flights on Construction Job Sites: A Case Study," *Int. J. Adv. Intell. Syst.*, vol. 12, no. 3, pp. 135–146, 2019.
- [16] K. Afsari, S. Halder, M. Ensafi, S. DeVito, and J. Serdakowski, "Fundamentals and Prospects of Four-Legged Robot Application in Construction Progress Monitoring," in *ASC International*

- Proceedings of the Annual Conference*, 2021, pp. 271–278.
- [17] P. Kim, J. Park, Y. K. Cho, and J. Kang, “UAV-assisted autonomous mobile robot navigation for as-is 3D data collection and registration in cluttered environments,” *Autom. Constr.*, vol. 106, no. 10, p. 102918, Oct. 2019, doi: 10.1016/j.autcon.2019.102918.
- [18] J. H. Lee, J. Park, and B. Jang, “Design of Robot based Work Progress Monitoring System for the Building Construction Site,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2018, pp. 1420–1422, doi: 10.1109/ICTC.2018.8539444.
- [19] S. Halder, K. Afsari, J. Serdakowski, S. DeVito, and R. King, “Accuracy Estimation for Autonomous Navigation of a Quadruped Robot in Construction Progress Monitoring,” 2021.
- [20] N. Sariff and N. Buniyamin, “An overview of autonomous mobile robot path planning algorithms,” in *2006 4th student conference on research and development*, 2006, pp. 183–188.
- [21] C. Brito, N. Alves, L. Magalhães, and M. Guevara, “BIM mixed reality tool for the inspection of heritage buildings,” *ISPRS Ann. Photogramm. Remote Sens. & Spat. Inf. Sci.*, vol. 4, 2019.
- [22] F. P. Rahimian, S. Seyedzadeh, S. Oliver, S. Rodriguez, and N. Dawood, “On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning,” *Autom. Constr.*, vol. 110, p. 103012, 2020.
- [23] C. Kim, T. Park, H. Lim, and H. Kim, “On-site construction management using mobile computing technology,” *Autom. Constr.*, vol. 35, pp. 415–423, 2013.
- [24] B. R. Mantha and B. de Soto, “Designing a reliable fiducial marker network for autonomous indoor robot navigation,” in *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC)*, 2019, pp. 21–24.
- [25] Unity Technologies, “Inner Workings of the Navigation System,” *Unity - Manual*, May 08, 2021.  
<https://docs.unity3d.com/2021.2/Documentation/Mannual/nav-InnerWorkings.html> (accessed May 16, 2021).