An Automated 3D Model Generation Framework for Construction Equipment Images using Edge Detection Algorithm

Jae-Kun Lee
Dept. of Architecutre & Architectural Engineering
Yonsei University
Seoul, Republic of Korea
jake102518@yonsei.ac.kr

Hye-Min Lee
Dept. of Civil & Environment Engineering
University of Science and Technology
KICT School
Goyang, Republic of Korea
hyemin@kict.re.kr

Hyoun-Seok Moon*
Dept. of Future & Smart Construction Research
KICT
Goyang, Republic of Korea
hsmoon@kict.re.kr
(Corresponding Author)

***Abstract*—Recently, the interest in construction equipment simulation has increased, and thus the necessity for rapid generation of 3D BIM construction equipment library for virtual space simulation at a low cost has also been raised. However, utilizing the existing BIM equipment libraries is inefficient as they are costly and require manual model making. As a solution, this paper suggests a framework for generating 3D construction equipment library based on the images of equipment elements using edge detection. Furthermore, we examined the practicality of the method for automatic 3D model generation that we present. The framework suggested in this study is expected to improve the productivity of the construction industry by enhancing the efficiency of sharing status information and facilitating the equipment interferences review within a virtual environment.**

Keywords—Building Information Modeling, Construction Equipment, BIM Library, Edge Detection Algorithm, Computer Vision, Framework.

I. Introduction

In the construction industry, various technologies that incorporate smart construction techniques have been increasingly employed. These include Building Information Modeling (BIM), Digital Twin, drones, collision avoidance sensors for construction equipment, and more [1]. Recently, the adoption of unmanned and automated construction equipment technologies is increasing aligning with the intelligent construction era. Furthermore, alongside artificial intelligence models, construction equipment using machine control (MC) and machine guidance (MG) algorithms has been developed to address the shortage of skilled workers, enhance construction site safety, and improve productivity in the construction industry [2]. Meanwhile, large construction projects are usually carried out through the participation and the division of roles among various construction companies which makes the process highly complex and sometimes hinders the smooth exchange of construction status information. To address these issues, especially construction work simulation proves to be valuable for forecasting construction timelines and creating effective schedule plans for management purposes [3]. The utilization of a Four-dimensional (4D) model with a high Level of Detail (LOD) to schedule and visualize the construction progress is anticipated to positively impact safety, productivity, and constructability on construction sites [4]. With this approach, researchers and managers can utilize a specific simulation language to design a construction operation, consider the predicted construction schedule, and efficiently evaluate various construction schemes [5].

As the demand for construction equipment simulation increases [6], there is a growing requirement to develop construction equipment libraries that can be utilized in sharing construction situation information and enabling real-time interference control on large construction sites. A BIM library is a collection of object information, consisting of individual components that form a facility. It allows extensive sharing and utilization of the model information across multiple users and projects [7]. One of them is the construction equipment library, which is used for construction simulation such as material handling planning, movement planning, and interference review, thereby evaluating actual risk factors in construction execution. In BIM Modeling software like Autodesk Revit, it is possible to download and utilize BIM object libraries as open-source resources from The NBS National BIM library [8]. It is an online environment that provides BIM libraries made by diverse manufacturers. These BIM libraries have also gained acknowledgment of the availability of BIM models from leading producers of CAD platforms, including Bentley, Autodesk, Tekla, and others [9].

In the case of the current acquisition methods for construction equipment libraries, there are few models available, even though a wide variety of construction equipment is required at construction sites. While there are dynamic modeling tools available for equipment interference review, such as Fuzor [10] and Smart Planner [11], the operational costs of using these programs are significant, and the equipment libraries provided by websites are limited in number. For this reason, field practitioners are compelled to either manually create the required equipment libraries or incur significant expenses by outsourcing the task to a software company. The existing methods of utilizing equipment libraries present challenges rapidly constructing and efficiently utilizing desired libraries. Utilizing BIM libraries at the online storage is an efficient way of finding the necessary data without having to redraw and reproduce them. In addition, it can suggest an effective solution and reduce costs when creating repetitive model information for large projects [9]. In the field of construction equipment simulation, we must actively utilize the advantages of these BIM libraries. Therefore, we propose a framework for automatically generating a 3D construction equipment library based on images of construction equipment using a computer vision approach. Using this framework, it is possible to secure operability in equipment simulation and generate equipment libraries based on minimal image and information resources, without incurring substantial costs.

Construction equipment pertains to machinery specifically designed for earthmoving tasks, encompassing a range of tools such as excavators, dump trucks, loaders, compaction rollers, graders, scrapers, and more. These machines are primarily utilized for four fundamental earthworks processes: excavating, hauling, spreading, and compacting [12]. In previous research related to construction equipment, various themes have been explored, including maintenance, productivity, optimization, operator's competence, robotics, and other aspects [13]. Recent studies related to construction equipment primarily focus on the construction worksite monitoring system [14][15][16], which includes activity recognition [6][17][18][19], activity tracking [20][21], and performance monitoring [22][23]. The monitoring can lead to enhanced construction efficiency, cost control, and the identification of potential project challenges [24]. To optimize construction site operations, Lou et al. [20] developed a framework that utilizes computer vision and deep learning to automatically estimate the poses of various construction equipment in videos captured on construction sites. Wu et al. [21] proposed a novel differential received signal strength positioning algorithm for precise equipment tracking. Rashid and Louis [12] introduced a framework based on an RNN-based deep learning network, while Langroodi et al. [17] devised a novel machine learning method that integrates a Random Forest classifier with the fractional calculus-based feature augmentation technique to create an accurate equipment activity recognition model. In regard to enhancing the productivity and efficiency of construction equipment, previous research has been studied on various methods for managing the usage of equipment using database [25] [26] and predicting the residual value with machine learning [27]. Furthermore, with regard to construction equipment simulation, Marzouk and Moselhi [28] proposed an automated engine to simulate earthmoving operations for estimating construction time and cost. Additionally, frameworks for predicting construction situations, equipment utilization rate, estimated duration, and cost were developed to visualize the construction process in a virtual space [29][30]. In addition to the aforementioned preceding research, studies have been conducted on various topics, including the detection of operation's mental fatigue for preventing accidents [31], construction equipment safety education utilizing 4D BIM with virtual reality [4][32], the simplification of construction equipment supply using BIM [33], and the implementation of Robotic Autonomous Systems (RAS) in the earthmoving process [34].

After reviewing the previous literature related to construction equipment, studies on construction equipment monitoring using on-site images or videos using computer vision and deep learning have been mainly conducted. In the case of studies on equipment simulation, there have been proposals for automation engines and frameworks but most of the studies have been conducted under the assumption that construction libraries were already provided. Even though research has been conducted on diverse subjects related to construction equipment, the domain of automatically generating a construction equipment library has not been fully explored.

This study presents a framework for automatically generating a construction equipment library based on images of construction equipment units. The main objective of this framework is to facilitate seamless information sharing regarding the construction status and enable the review of construction interferences in virtual space. First of all, we reviewed the previous research related to construction equipment and analyzed the grounded theory approach and the methods of findContours. Based on that, the framework was derived to enhance the productivity of construction equipment-related tasks. The framework is organized into three primary procedures, with comprehensive explanations of the algorithms and technical aspects linked to each step. Following the procedures outlined in the proposed framework, we developed a 3D equipment library using construction equipment unit images. The practical applicability of automatically generating a 3D construction equipment library has been verified. Lastly, we summarize the implications and limitations of this study, as well as delineate potential avenues for future investigation in this field.

II. Theory Approach and Proposed Framework

In this paper, “findCountours()” script function, which is one of the OpenCV algorithms, was used to perform the edge line detection. Other methods, such as Sobel and Canny, determine the edge line based on the image gradient. In contrast, the "findContours()" function detects the boundary between 0 and 1 pixels, which requires a binary image consisting only of black or white pixels. Despite its more complex prerequisite, this function has two advantages compared to other edge detection algorithms in generating the BIM equipment library. Firstly, when the resolution of the input image is low, it is better to ignore the detailed shape of the pixels and instead, draw a clean straight line considering the overall boundary. The "findContours" function can achieve it with an approximation step, while others, like Canny, cannot. Secondly, it is also necessary to detect and select the inner edges inside the outermost edge to specify the coordinates of the connection points. It detects and indexes every discrete boundary in different hierarchies, which is useful when selecting the outline of a specific joint to determine the exact coordinate. The "findContours" function is a Python implementation of the two border-following algorithms proposed by Suzuki and Abe [35]. Among the two algorithms, we used the one that determines not only the borders but also their hierarchical relationship (i.e., border A surrounds border B). The detailed description of the algorithm is as follows:

(a) If the current following border is between the 0-component which contains the pixel (p, q + 1) and the 1-component which contains the pixel (p, q), change the value of the pixel (p, q) to - NBD.

(b) Otherwise, set the value of the pixel (p, q) to NBD unless (p, q) is on an already followed border.

where 0-component means a black pixel, 1-component means a white pixel, and NBD represents a distinct number of the border hierarchy.

This paper suggests a framework, as depicted in Figure 1, to establish the foundation of a program that performs the following two functions: (i) automatically simulating an equipment interference review which examines the working range of equipment and its operation time, (ii) sharing information on the construction progress in a virtual space. The framework enables the simple creation of a new 3D BIM equipment library using images and the specification data of the equipment.

The implementation procedure of the framework consists of three main steps: (i) Image Selection, (ii) Vector Edge Line Detection, and (iii) 3D Model Generation. First, the equipment images that will be used for creating the equipment library are selected based on the specific conditions, which are described in the next part. Next, the outline of the equipment unit part is extracted from the selected image. This extracted outline is then vectorized, and by applying the preset size values including the thickness, a 3D model of the equipment unit can be created. Detailed algorithms and technicalities for each step are explained in the next part.

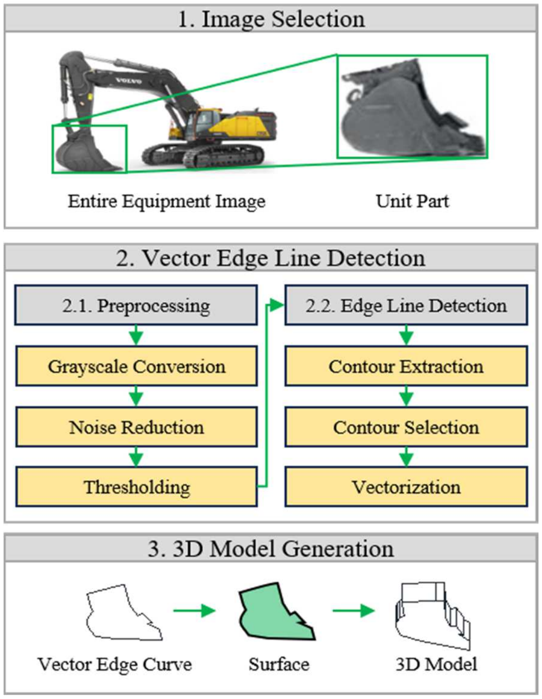


Fig. 1: The process map of the research methodology

III. Framework Implementation

A. Image Selection

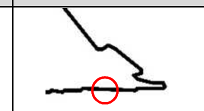
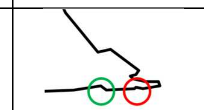
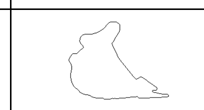
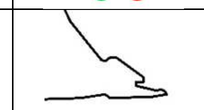
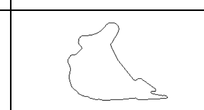
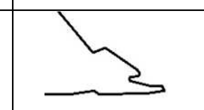
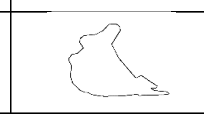
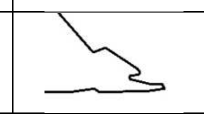
We start by collecting the images of the construction equipment that we opt to generate into a 3D model. In this paper, only the images that satisfy the following two conditions are used: (i) they must be taken from the side of the equipment, and (ii) they must have a white or transparent background. These restrictions are to ensure accurate detection of the edge line and the uniform quality of the 3D model without further processing such as AI-based object detection. Images that meet these qualifications can be found from the official catalogs provided by the manufacturers. Then, each collected equipment image is divided into minimum unit parts that perform uniform translational or rotational motion. This process could be skipped if there exist the official pre-divided images of the unit parts.

The example image used in this paper is a side image of Cat® Heavy Duty Buckets cut from an official 360-degree image of Cat® 313 GC excavator by Caterpillar INC [36]. The resolution of the bucket image was originally 96 dpi. To find out the minimum resolution that can go through the edge detecting process, we conducted a modeling test with four images, each of which has 25%, 30%, 35%, and 40% of the original resolution. Table I is the result of the edge detection and the approximated point rate (APR) of each line. APR can be calculated as follows:

$$APR = (NPO-NPA) / NPO$$

where NPO stands for the number of points in an original edge line, and NPA stands for the number of points after approximation. Images with low resolution also have low APR because there are insufficient data to distinguish a meaningful detail from a noise. We considered that images should have an APR value higher than 0.75 for a high-quality outcome. However, since APR can only be calculated after the edge detection, it is recommended to check the resolution instead. Although the APR depends on the shape of the object, images with 0.75 or larger APR generally have around 25 dpi. Therefore, the image resolution of a unit part should be at least 25 dpi (about 26% of 96 dpi) to correctly obtain an edge line.

TABLE I. The Result of The Edge Detection and APR

Dpi	Image Result	Enlargement	APR
25% (24 dpi)			0.674
30% (29 dpi)			0.749
35% (34 dpi)			0.794
40% (38 dpi)			0.817
Original (96 dpi)			0.927

B. Preprocessing

Since the retrieved images are already divided into minimum unit parts that translate or rotate as one, detecting the internal components of a unit part is redundant. Therefore, the preprocessing stage is for specifying and separating the entire object from the background within the image. The preprocessing stage includes three steps: (i) Grayscale Conversion, (ii) Noise Reduction, and (iii) Thresholding.

Grayscale Conversion. The official images are usually taken in color space which is unnecessary for edge detection. For that reason, we converted the RGB value of the pixels into grayscale value. This can be done by "COLOR RGB2GRAY() function" from OpenCV. The detailed equation of the conversion is as follows:

$$\text{RGB to Gray: } Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Noise Reduction. The resolution of the example bucket image is 96 dpi, which is a de facto standard image quality value for a 23-inch FHD monitor. However, as the images of the unit parts were created by cutting out the equipment image, they are likely to have a lower resolution than 96 dpi (e.g., 72 dpi) in many cases. As a result, it is difficult to draw an accurate edge line in this low-resolution raster image, because the pixel size representing the line becomes relatively large. The noise reduction step is blurring out the shape of the pixels around the edges to obtain a clear and straight line. This study used "Gaussian Blur" of a kernel size 5 by 5 for denoising the bucket. The kernel size may vary depending on the resolution of the input image. Denoising with Gaussian Blur was done by calling "GaussianBlur()" function from the OpenCV algorithm.

Thresholding. The bucket image includes unnecessary edges such as edges around the joint between a bucket lip and the body or the imprinted logo. Thresholding is for erasing these unnecessary edges and leaving only the shape of the entire unit part. Global thresholding is applied which converts pixels with a grayscale value (Y) higher than a fixed threshold value into black pixels (Y=0) and the others into white pixels (Y=255). The threshold value we used for the sample image was 224, however, it is impossible to predetermine the exact number among the continuous values that can properly threshold any input images. For example, the blurred bucket image contained every grayscale value from 71 to 255. Table Ⅱ presents the results of thresholding with arbitrary values and their differences. As the threshold value influences the quality of the final output, the user should manipulate the value by considering multiple properties of the input image. The properties that should be considered in the process of the bucket image are as follows: (i) the color of the object in grayscale, (ii) some background pixels that are not pure white (Y<255), (iii) the possibility of the shadow from the equipment image not being completely erased and remaining as bright gray. Thresholding was done by calling the OpenCV threshold function.

TABLE II. THE COMPARISON OF DIFFERENT THRESHOLD VALUES

Threshold = 224	Threshold = 230	Threshold = 240	Threshold = 250
			
			

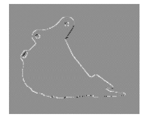
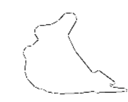
C. Edge Line Detection

The final goal of this paper is not only automating the generation of a 3D edge bounding model but also proposing a method for accumulating equipment data to create an equipment library. There are two types of data that can be collected through the whole process: a vectorized file of an edge line and a 3D model. In the edge line detection stage, we draw an edge line and save it as a vectorized file. This stage follows three steps which are (i) Contour Extraction, (ii) Contour Selection, and (iii) Vectorization.

Contour Extraction. To detect the edge line from preprocessed images, 4 different methods from the OpenCV library were tested : (i) Canny, (ii) Sobel, (iii) findContours, (iv) findContours+approxPolyDP.

Table Ⅲ shows the edge detection results of the four tested methods on the bucket image. Edge line detected by Canny and two findConoturs-based methods showed precise and clean lines. However, the lines made without the approxPolyDP had up to 14.2 times more points than the one made with it. This is because the more precise the line is, the more it divides into short vertical segments along the pixel shape. The lower the image resolution is, the stronger this phenomenon appears. For that reason, this paper used the findContours+approxPolyDP method for edge detection. approxPolyDP() is an implementation of the Ramer-Douglas-Peucker algorithm in OpenCV. The epsilon value (the maximum distance between a point and a curve) used on the bucket image was 0.002 times the total line length.

TABLE III. THE COMPARISON OF EDGE LINE ALGORITHMS

Canny	Sobel	Find Contours	Contour-Approximation
			
Number of Points That Make Up Each Edge Line			
568	Not Available	545	40

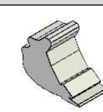
Contour Selection. The 2.2 Preprocessing stage may fail to separate the entire object from the background as a whole when the imported image has extremely low resolution. In this case, the pixels on the outer parts may be split into small individual pieces. Comparing the areas of the contoured lines and selecting the line with the largest area helps in choosing the targeted outline.

Vectorization. The output of the "findContours+approxPolyDP" method is a two-dimensional array composed of the coordinates that represent the vertices of the edge line. The vectorization of this line was done by creating an empty DXF file and adding lines that connect neighboring vertices using the ezdxf interface on it. The ezdxf interface is a Python package developed by Manfred Moitzi which creates various versions of "dxf" files.

D. 3D Model Generation

To create a 3D model, a vectorized "dxf" file is imported into Autodesk Revit. The vector edge line is placed on the XY plane, and the depth information of the Z axis was collected from the manufacturer's specification information. A 3D model is produced by creating a surface with the edge line and applying the depth value to the surface. Various sizes of models can be generated by adjusting the scale of the created model for several purposes. Table Ⅳ presents the output of each stage from image selection to 3D model generation.

TABLE IV. THE OUTPUT OF EACH STAGE

(Step 1) Image Selection	(Step 2) Preprocessing	(Step 3) Edge Line Detection	(Step 4) 3D Model Generation
			

IV. DISCUSSION

A. Result Analysis

Figure 2 shows an entire 3D model of Cat® 313 GC excavator generated by the method presented [36]. The 3D models of each part of the equipment, such as arm, boom, and superstructure, were also produced in the way described above. If compared side by side, the real photograph and the generated model seem similar enough in terms of the overall shape, the location of the connections, and the size proportion of each part. All algorithms, steps, and values described in the above methodology are tested in the process of generating the

3D model shown in Figure 2 to check their validity and necessity.

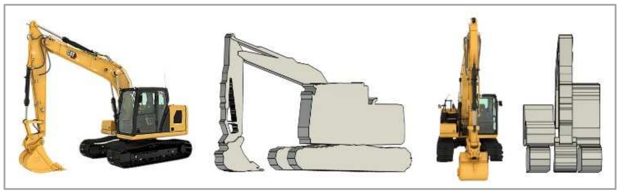


Fig. 2: The excavator image and the entire 3D model

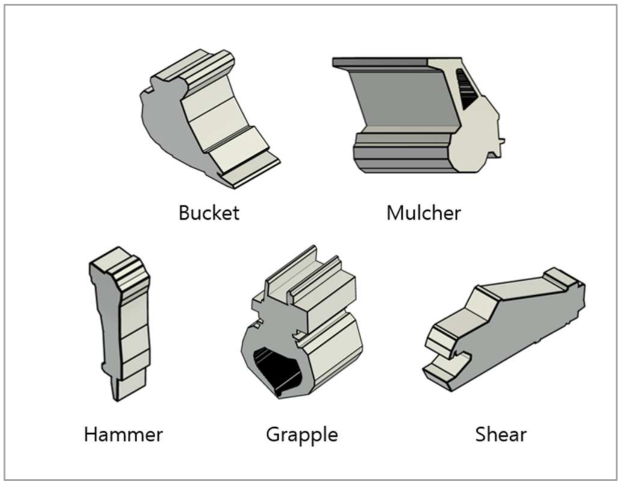


Fig. 3: The 3D models of various attachments

Moreover, the possibility of creating a construction equipment library using the suggested method was also assessed. The 3D model of the attachments including mulchers, hammers, grapples, and thumbs is shown in Figure 3. It indicates that the proposed method can generate a 3D model regardless of which type of equipment is if there is a side image. Users can easily generate a 3D model with their desired attachments, check its working range, and conduct an interference or collision review. In addition, the produced model and its vectorized outline will be accumulated in the library and can be utilized again in the future.

Table Ⅴ shows the time taken to build a DXF file of each unit part by using the method. The testing environment for this measurement is Apple M1 Pro with 16GB RAM in MacBook Pro 14 with Python 3 on Jupyter Lab. The result demonstrates that the time required will be between 10 to 100 milliseconds when an input image satisfies the two conditions described in III-A Image Selection. To sum up, the method we suggest has satisfied the research objectives which are creating a vectorized edge line data from a 2D image and generating a 3D model based on the line data.

TABLE V. THE TIME RECORD TAKEN TO BUILD EACH MODEL

Bucket	Arm	Boom
15.79 ms	55.40 ms	39.16 ms
Superstructure	Shoe	Mulcher
34.40 ms	20.84 ms	21.52 ms
Hammer	Grapple	Shear
14.93 ms	32.75 ms	37.07 ms

B. *Application Plan*

This paper presented a framework to rapidly create BIM libraries for construction equipment utilizing images and information of the equipment. Through construction equipment libraries from this framework, we can easily utilize BIM libraries for simulation in practical applications, resulting in time and cost savings. This is because the libraries can be generated using minimal resources of equipment images and data. In addition, it is possible to automatically simulate equipment interference reviews, equipment movement planning, and material movement checks in a virtual space. By simulating construction sites, we can share information about construction procedures and manage the construction schedule more efficiently.

V. CONCLUSION AND FUTURE WORK

In this paper, we suggested a novel framework to automatically generate 3D BIM libraries with side-view images of construction equipment. By generating a 3D model of an excavator and its attachments, we showed the practicality of our method. Also, the records of the lead time to build the 3D model proved the work efficiency of utilizing it in the field. This study holds significant implications as it offers a method for easily and swiftly generating a low-cost equipment BIM library solely from equipment images, enabling assembly with completed models. This technology applies not only to equipment but also to various fields that require libraries, enabling the effortless automatic generation of BIM models from images depicting construction structures and other physical forms. In the future, upon the development of equipment simulation systems, the rapid creation of virtual equipment models could facilitate the development of a digital twin environment, seamlessly integrating with real-world scenarios. However, a comparison between the time taken when using the method and the time when manually produced was not addressed in this study and could be examined more in further research. Based on this paper, we suggest some possible topics of future studies as follows: detecting equipment in images with no background removal (i.e., taken from a construction site, captured from a video) using AI object detection, developing a dynamic model for an interference and work range review, and comparing the performance numerically between various edge detection algorithms.

REFERENCES

[1] Aiming to Become the Leader in Smart Construction"… DL E&C, Accelerates Implementation of 'Digital Twin Environment', https://www.wikileaks-kr.org/news/articleView.html?idxno=138352

[2] Unmanned and Automated Construction Equipment Leads the Innovation of Smart Construction, http://www.engjournal.co.kr/news/articleView.html?idxno=2436

[3] Lv, F., Wang, J., Cui, B., Yu, J., Sun, J., and Zhang, J., “An improved extreme gradient boosting approach to vehicle speed prediction for construction simulation of earthwork,” *Automation in construction.*, vol.119, 103351, 2020.

[4] Amrollahibuki, R., “Modeling Construction Equipment in 4D Simulation and Application in VR Safety Training,” 2019.

[5] AbouRizk, S., Halpin, D., Mohamed, Y., and Hermann, U., “Research in modeling and simulation for improving construction engineering operations,” *Journal of Construction Engineering and Management.*, vol. 137, no. 10, pp. 843-852, 2011.

[6] Rashid, K. M., and Louis, J., “Times-series data augmentation and deep learning for construction equipment activity recognition,” *Advanced Engineering Informatics.*, vol. 42, 100944, 2019.

[7] “BIM Execution Guidelines for Owners,” *Ministry of Land Infrastructure and Transport.*, 2022.

[8] NBS Source, https://source.thenbs.com/

[9] NBS National BIM Library, https://www.designingbuildings.co.uk/wiki/NBS_National_BIM_Library

[10] Fuzor, http://www.fuzor.co.kr/

[11] Smart Planner, https://www.smartapp.com/

[12] Peurifoy, R.L. and Ledbetter, W.B., "Construction planning, equipment and methods," *New York: McGraw-Hill.,* 1985.

[13] Naskoudakis, I., and Petroutsatou, K., "A thematic review of main researches on construction equipment over the recent years," *Procedia engineering.*, vol. 164, pp. 206-213, 2016.

[14] Naghshbandi, S. N., Varga, L., and Hu, Y., "Technologies for safe and resilient earthmoving operations: A systematic literature review," *Automation in Construction.*, vol. 125, 103632, 2021.

[15] Azar, E. R., and Kamat, V. R., "Earthmoving equipment automation: A review of technical advances and future outlook," *Journal of Information Technology in Construction (ITcon).*, vol. 22, no. 13, pp. 247-265, 2017.

[16] Chen, C., Zhu, Z., and Hammad, A., "Critical review and road map of automated methods for earthmoving equipment productivity monitoring," *Journal of Computing in Civil Engineering.*, vol. 36, no. 3, 03122001, 2022.

[17] Langroodi, A. K., Vahdatikhaki, F., and Doree, A., "Activity recognition of construction equipment using fractional random forest," *Automation in construction.*, vol. 122, 103465, 2021.

[18] Nakanishi, Y., Kaneta, T., and Nishino, S., "A review of monitoring construction equipment in support of construction project management," *Frontiers in Built Environment.*, vol. 7, 632593, 2022.

[19] Akhavian, R., and Behzadan, A. H., "Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers," *Advanced Engineering Informatics.*, vol. 29, no. 4, pp. 867-877, 2015.

[20] Luo, H., Wang, M., Wong, P. K. Y., & Cheng, J. C., "Full body pose estimation of construction equipment using computer vision and deep learning techniques," *Automation in construction.*, vol. 110, 103016, 2020.

[21] Wu, C., Wang, X., Chen, M., and Kim, M. J., "Differential received signal strength based RFID positioning for construction equipment tracking," *Advanced Engineering Informatics.*, vol. 42, 100960, 2019.

[22] Sherafat, B., Ahn, C. R., Akhavian, R., Behzadan, A. H., Golparvar-Fard, M., Kim, H., ... and Azar, E. R., "Automated methods for activity recognition of construction workers and equipment: State-of-the-art review," *Journal of Construction Engineering and Management.*, vol. 146, no. 6, 03120002, 2020.

[23] Nakanishi, Y., Kaneta, T., & Nishino, S., "A review of monitoring construction equipment in support of construction project management," *Frontiers in Built Environment*, vol. 7, 632593, 2022.

[24] Chen, C., Zhu, Z., and Hammad, A., "Critical review and road map of automated methods for earthmoving equipment productivity monitoring," *Journal of Computing in Civil Engineering*, vol. 36, no. 3, 03122001. 2022.

[25] Yang, Y., Yu, C., and Zhong, R. Y., "Generalized linear model-based data analytic approach for construction equipment management," *Advanced Engineering Informatics*., vol. 55, 101884, 2023.

[26] Schimanski, C. P., Marcher, C., Toller, G., Pasetti Monizza, G., and Matt, D. T., "Enhancing automation in the construction equipment industry through implementation of BIM. In International Conference on Cooperative Design," *Visualization and Engineering*., pp. 64-73, Oct. 2019.

[27] Shehadeh, A., Alshboul, O., Al Mamlook, R. E., and Hamedat, O., "Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression," *Automation in Construction*., vol. 129, 103827, 2021.

[28] Marzouk, M., and Moselhi, O., "Object-oriented simulation model for earthmoving operations," *Journal of Construction Engineering and Management.*, vol. 129, no. 2, pp. 173-181, 2003.

[29] Jabri, A., and Zayed, T., "Agent-based modeling and simulation of earthmoving operations," *Automation in Construction*., vol. 81, pp. 210-223, 2017.

[30] Cheng, F. F., Wang, Y. W., Ling, X. Z., and Bai, Y., "A Petri net simulation model for virtual construction of earthmoving operations," *Automation in Construction.*, vol. 20, no. 2, pp. 181-188, 2011.

[31] Li, J., Li, H., Umer, W., Wang, H., Xing, X., Zhao, S., and Hou, J., "Identification and classification of construction equipment operators' mental fatigue using wearable eye-tracking technology," *Automation in Construction.*, vol. 109, 103000, 2020.

[32] Vahdatikhaki, F., El Ammari, K., Langroodi, A. K., Miller, S., Hammad, A., and Doree, A., "Beyond data visualization: A context-realistic construction equipment training simulators," *Automation in construction.*, vol. 106, 102853, 2019.

[33] Chandra, S. S., Sepasgozar, S. M., Kumar, V. R. P., Singh, A. K., Krishnaraj, L., & Awuzie, B. O., "Assessing factors affecting construction equipment productivity using structural equation modeling," *Buildings.*, vol. 13, no. 2, pp. 502, 2023, 2023.

[34] Ha, Q. P., Yen, L., and Balaguer, C., "Robotic autonomous systems for earthmoving in military applications," *Automation in Construction*., vol. 107, 102934, 2019.

[35] Suzuki, S. and Abe, K., "Topological Structural Analysis of Digitized Binary Images by Border Following,", Computer Vison, Graphics, and Image Processing., vol. 30, pp. 32-46, 1985.

[36] CATERPILLAR (CAT), https://www.cat.com/en_US/products/new/equipment/excavators/small-excavators/102646.html