Automation in Construction 133 (2022) 103996

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Implementation experiments on convolutional neural network training using synthetic images for 3D pose estimation of an excavator on real images

Bilawal Mahmood, SangUk Han [*], Jongwon Seo

Department of Civil and Environmental Engineering, Hanyang University, Seoul, South Korea

ARTICLE INFO

Keywords:
Excavator
3D pose estimation
Synthetic dataset
Kinematic constraints
Perspective camera projection

ABSTRACT

Remote and descriptive visualization of spatio-temporal information of excavator activities may increase awareness about jobsite hazards and operational performance in earthwork operations. One of the emerging approaches to collect this information is to extract the 3D pose of an excavator from the video frames using a convolutional neural network (CNN). However, this method requires labeling the training datasets, which are difficult to prepare because of conditions unsuitable for installing the motion capture sensors. This study investigates the performance of a CNN for estimating the 3D pose when trained on a synthetic dataset. In particular, a kinematic constraint is proposed to update the model parameters efficiently during training. The results show that the proposed method estimated the 3D poses of a real excavator with an average pose error of 9.63°. Hence, the proposed data augmentation method could help address the training data issues and improves the learning of real data complexity.

1. Introduction

Earthmoving operations are an essential part of construction projects, such as dam construction, foundation work, road construction, and airport construction. In these operations, earth materials, such as soil, are removed from the ground and transferred to another location. Most of the activities in these operations are driven by heavy equipment, such as a loader or an excavator. Because of the use of heavy equipment, these activities are among the most critical in construction from the safety and economic perspectives. From a safety perspective, the fatality rate in earthmoving operations is 112% higher than that in general construction [1]. Struck-by fatalities are one of the four most deadly hazards (i.e., fall, electrocution, caught-in, and struck-by fatalities) in the field of construction [2], and approximately 75% of struck-by fatalities are caused by inappropriate interactions between construction workers and heavy earthwork equipment [1,2]. Among the heavy earthwork equipment, the most significant proportion of fatalities is caused by excavators on the jobsite [3]. From an economic perspective, the earthmoving operation cost accounts for more than 20% of the total project cost [4]. The cost of earthwork significantly depends on the productivity and efficiency of the excavator [3]. The productivity of an excavator is the measure of the output, i.e., the amount of soil excavated and transported per unit time (i.e., an hour), while the efficiency is the percentage of total project time in which the excavator is productive. For the former, the productivity of the excavator is affected by the number of trucks assigned to transport the soil. For instance, a poorly assigned number of dump trucks to the excavator can increase the idle time of the excavator; in return, this time can decrease the overall project productivity. Meanwhile, the efficiency of the excavator is affected by the sequence of the boom, arm, and bucket motions, and the trajectory length of the bucket motion cycle [5]. For instance, a short bucket trajectory cycle to dig the soil from the ground, dumping it in a truck, and then returning to dig again, can efficiently reduce the operational time and fuel consumption [5]. Thus, on-site monitoring of excavator activities is crucial for ensuring the safety and productivity of earthwork. Effective means of monitoring the excavator activities can facilitate the collection of spatial and temporal information, such as the working pose and trajectory required for productivity and safety analysis. Using this information, a manager can analyze the reasons for poor safety performance and low productivity.

Generally, excavator-related activities are manually monitored to evaluate the worker's safety [6,7] and excavator productivity [8–10]. For safety evaluation, an inspector manually observes the jobsite to identify the hazards and violations [11]. For instance, the location of

[*] Corresponding author.
E-mail address: sanguk@hanyang.ac.kr (S. Han).

https://doi.org/10.1016/j.autcon.2021.103996
Received 15 June 2021; Received in revised form 18 September 2021; Accepted 5 October 2021
Available online 14 October 2021
0926-5805/© 2021 Elsevier B.V. All rights reserved.

workers near the excavator and the spatial extent of excavator parts during earthwork operations are manually observed and evaluated to identify the hazards. In addition, the working speed of the excavator, action smoothness, and pose angles are manually observed and then compared with the standards provided by the manufacturer to identify the violations. For productivity estimation, the activities of the excavator are recognized, and then their time duration is calculated (e.g., the time required to excavate and dump a soil load into a truck) using a stopwatch. The productivity of the excavator is the ratio of the amount of excavated soil to the time duration for this excavation (e.g., m^3/h). To calculate this time duration for productivity estimation, random work sampling is conducted [12]. In this work sampling, the inspector visits the jobsite randomly and calculates the activity duration for unbiased monitoring. Usually, work sampling is performed once or twice a week [7] depending on the project size, and then a report on the site conditions is prepared based on the inspector's perception; the site inspector spends 30–50% of his/her time on this process [10]. Consequently, the traditional method of monitoring may not be sufficient for a manager to identify the hazards proactively or inefficient operations on an earthwork site where productivity and safety conditions change continuously. In addition, this report is subjective and may not provide a clear mental model of the job site's complexity and dynamics [6,7] to analyze the incident.

To facilitate the monitoring of excavator activities in a continuous and automated manner, this paper proposes a vision-based excavator pose estimation method. This pose estimation method calculates the relative position of the excavator elements, i.e., the boom, arm, and bucket, from an image. This pose can be linked to a virtual excavator model to visualize the spatial information (i.e., the pose) of a real excavator in a virtual environment. Then, an animation is created from the sequences of these poses to visualize the temporal information (i.e., trajectories) of excavator activities. By visualizing this animation of equipment movements in such a virtual environment of a construction jobsite, the safety and productivity performance of the excavator can be monitored and analyzed remotely. For example, the 3d pose information representing articulated motions of an excavator may allow for recognizing pre-defined situational scenarios of unsafe incidents (e.g., a bucket moving closer to a worker) in real-time [13], classifying the motions into specific activities (e.g., idling, loading, dumping) [14] and measuring their cycle times for productivity [15], providing an operator with visualized information that helps perceive the ongoing motions of a bucket arm relative to the as-planned place for productivity improvement [16], and observing the trajectory smoothness of a bucket and an arm for the assessment of the operator's skill [17]. Specifically, for this pose estimation, a CNN model is trained on synthetic excavator images with corresponding poses instead of a real image dataset because it is challenging to measure the 3D poses of an excavator on a job site directly, whose dataset is required to train the pose estimation model. Thus, this study evaluates the performance of synthetic image-based CNN model training for the 3D pose estimation of a real excavator. By continuously recovering the excavator poses and then visualizing these pose trajectories in a virtual environment, a manager can proactively evaluate the excavator performance in a dynamic construction site.

2. Literature review

The spatial orientation and configuration of an articulated subject (e.g., humans or excavators), whose body parts are connected to each other through rotating joints, can be described by the 3D pose. The 3D pose can be represented in two ways: 1) the relative position of the joints in a Cartesian coordinate system, and 2) the relative angle of the parts around its joint. This 3D pose has been studied for sensor-based and vision-based methods. In sensor-based methods, a sensor network is attached to the subject, and these sensors measure the change in their angle or position to calculate the 3D pose. On the contrary, in vision-based methods, images of the subject are captured using a specific visual device, and the 3D pose is estimated by calculating the visual features in these images. These visual features can either be designed manually to detect the specific shapes of the subject in images (e.g., the histogram of oriented gradients) or can be automatically extracted and trained to locate the subject parts in the image (e.g., a convolutional neural network (CNN)). The details of the 3D pose estimation methods are further described in the following sections, and the potential issues in machine learning approaches are discussed from a technical perspective.

2.1. 3D pose estimation methods

3D pose estimation methods can be categorized as sensor-based and vision-based methods. In sensor-based methods, inertial measurement units (IMUs) [18], global positioning systems (GPS), wireless local area networks (WLANs), radio frequency identification (RFID), and ultra-wideband (UWB)-based methods are used. An IMU sensor is attached to the excavator elements [18] and the change in IMU angular orientation is measured. IMU-based pose estimation does not depend on the visibility of the excavator, and the pose can be estimated even when the excavator parts are occluded by other equipment. However, the IMU-based method requires initial manual calibration before recovering the pose data. In addition, measuring the change in angle continuously suffers from magnetic interference and drift issues over time [19]. In the GPS-based pose estimation method, GPS sensors are attached to the excavator elements, and the 3D pose is estimated using the GPS position and kinematics of the excavator. Similar to the IMU-based method, the GPS-based method does not have visibility issues. In addition, each pose estimated from the GPS is independent of other estimated poses, which resolves the drift issues. However, GPS receivers may detect signal blockages in urban populated areas [20]. In WLAN-, RFID-, and UWB-based 3D pose estimation, a signal source is placed at a fixed position, and signal receiver tags are attached to each excavator element. To estimate the 3D pose of the excavator, the location of these tags is calculated based on the time of arrival, angle of arrival, and received signal strength of signals received back on the source. These methods are more accurate than IMU-based and GPS-based methods [21] and are predominantly used on actual jobsites [21]. However, these systems require the sensors to be precalibrated [22]. In addition, it is not possible to place a signal source at a fixed position in dynamic jobsite conditions.

In the vision-based method, two methods are used: marker-based and marker-less. For marker-based pose estimation, a network of markers is attached to the excavator, and the detected location of these markers in the image is used to calculate the 3D pose of an excavator [16,23,24]. These marker-based methods are being used extensively [25]. However, these methods require precise pre-installation of the markers at different locations of excavators, which can easily be broken and lost during site operations. For marker-less pose estimation, first, the features are extracted from visual data, and then the 3D pose is calculated from the features. These visual data can be depth images [14,26–28], multi-view images [29], and monocular images [30]. Depth images already have depth information in each pixel of the RGB image, which makes 3D pose estimation a keypoint detection problem. For instance, human body parts are detected using depth-based clustering of pixels [31] and then recognized as Randomized Trees [32]. However, the sensors for depth images may be difficult to handle in a dynamic construction environment because they have to be installed at a fixed location and have a limited scanning range [29]. In multi-view images, parts of excavators are detected in RGB images using color-based clustering of pixels, and the parts are then recognized by fitting the geometric shapes. Further depth information is calculated via triangulation of the detected parts in paired images [29] however, this method may be slow and requires the calibration of multiple cameras focusing on the same view [33].

With the advancement in CNNs, 3D pose estimation using monocular images has been studied for humans. Monocular image-based 3D pose estimation can be categorized into model-free 3D pose estimation and model-based 3D pose estimation [34]. Model-free methods estimate the 3D pose of each body joint independently and do not consider their mutual relationship. In contrast, the model-based method estimates the pose of each joint based on the mutual connectivity relationship. The model-free method can be further divided into two methods: direct location estimation and 2D with depth estimation. For direct location estimation, a CNN model is directly trained for images against the joint location [35–37]. On the contrary, in 2D with depth-based 3D pose estimation, 2D keypoint locations are calculated, and then the depth of the individual joint is estimated to convert these 2D keypoints into 3D keypoints [38–40]. Compared to direct pose estimation, 2D with depth-based pose estimation shows better performance [34]. In model-based 3D pose estimation, a mutual relationship among the body parts is included in the training of the CNN model. These relationships restrict the CNN model from estimating unrealistic poses. For instance, joint connectivity [41] and bone-length ratios [30] are calculated for the estimated pose, and CNN model parameters are corrected to fulfill these constraints. Thus, in monocular image-based pose estimation methods, the model-based CNN method is more suitable for excavator 3D pose estimation than the model-free method because, in the training of model-based CNNs, constraint parameters are used to refine the estimated pose that will conserve the joint configuration and mutual relationship of different parts of the excavator. However, existing model-based CNN methods are mostly designed for human pose estimation, and technical issues may arise for the implementation of these methods for excavator pose estimation.

2.2. Technical issues in CNN-based pose estimation

To train a CNN for 3D pose estimation, a labeled dataset and an appropriate loss function are required to compute the errors and update the model parameters through an iterative learning processes. For instance, the loss function estimates the training error by comparing the estimated poses from the CNN and the corresponding ground truth poses in the labeled dataset. Then, this training error is adjusted in the CNN weight parameters. For an excavator, these loss functions are specifically formulated in a model-based method to calculate the deviations of the estimated pose from natural pose constraints (i.e., avoiding awkward poses not performed in practice). In previous studies [30,41], additional loss functions have been used to preserve the spatial constraints of the human body (e.g., bone length [30] and mutual relationship of joints [41]). However, these loss functions are unable to preserve the natural kinematic constraints [42], such as joint rotation along an axis that is naturally constrained or a joint rotation beyond the natural limits. This kinematic error occurs because previous methods assume a single scaling factor for image generation [42] and they scale down the depths of all the subject parts with a single scaling factor. However, there is non-linear scaling of subject parts owing to the perspective projection, and this effect is relatively large in excavators because their size is larger than the average size of humans. In this regard, the intrinsic parameters of the camera are calculated to correct the pose for kinematic constraints [42]. However, this method is not generalized for all the image-capturing scenarios, such as cameras with different focal lengths at different positions.

The training dataset comprises images labeled with corresponding poses. In this pose labeling, pixels of joint positions are marked on the image, and the camera depth of those pixels is calculated in the image space. Motion capture sensors have been used to prepare a training dataset [43] in the field of human 3D pose estimation. However, this method may not be suitable for an excavator because it is difficult to install such sensors onto heavy equipment operating in harsh construction environments, which makes it difficult to obtain 3D poses for labeling the actual excavator images. A virtual excavator model has been used as an alternative to calculate the poses of an excavator in a real image [44]. For instance, the virtual model is rotated to obtain an approximate pose of the excavator in an image. However, it is time-consuming to rotate the virtual model manually to make it look similar to the pose in the image. For automated labeling, a robotic arm in an indoor environment [25] and images of a virtual excavator model [45] have also been proposed to obtain the 3D pose labels of training images. This robotic arm or virtual model is simulated for realistic movement of the excavator such that the pose data are automatically calculated after applying a motion and capturing an image. For automated generation of the training dataset, compared to a robotic arm, virtual model-based image capturing and 3D pose labeling methods provide a cheap and easy-to-implement solution. However, synthetic images captured from the virtual model do not have natural noise, occlusions, backgrounds, and lighting conditions similar to the site images, which makes the visual features of these images different from the real images. Recently, the performance of these images in stereo vision-based excavator pose estimation has been studied using traditional image processing techniques (i.e., color-based clustering of pixels) [29]. However, to estimate the 3D pose of the excavator from a monocular image, the performance of synthetic image-based CNN model training remains unknown.

3. Method

To visualize the excavator activities, we experimentally evaluated the performance of vision-based 3D pose estimation when synthetic excavator images were used for CNN model training. The real excavator video frames were then used for testing the trained CNN model. To prepare a synthetic dataset of images and the corresponding 3D pose data for CNN model training, a method that can capture images from any camera view angle is required; these training images should represent an actual construction jobsite and excavator dimensions, and the method should be fast enough to label the 3D poses in images. To fulfill these requirements, synthetic images of the excavator were used to train a CNN model for excavator 3D pose estimation. To generate a synthetic training dataset, a virtual excavator model was generated with a shape and dimensions similar to that of a real excavator. Then, the images of the virtual excavator were captured using a virtual camera from all possible viewpoints. A 3D pose corresponding to the captured image was obtained by projecting the excavator joint positions to the image space. However, these images did not have natural visual features such as lightening, occlusions, image distortion, and poor visibility owing to issues like dust in the air and natural background scenes. Data augmentation techniques were thus applied to make these synthetic images look similar to real jobsites.

Fig. 1 provides an overview of the experimental stages in the vision-based excavator pose estimation method. The methodology includes the training, testing, and post-processing stages. In the training stage, a virtual model of actual excavator dimensions was created by joining simple geometric shapes (e.g., planes, spheres, and surfaces). Then, the kinematic relationship between the excavator elements (e.g., the boom, arm, and bucket) was defined, and random rotations were applied to these excavator elements. After rotating these elements, a virtual camera was placed randomly around the excavator, whose line of sight passed through the excavator center. This virtual camera rendered an image of the excavator using ray tracing and rasterization. To obtain the 3D pose of the excavator corresponding to this image, the pixel coordinates of the joints of the excavator were considered as a 2D pose. The distance of these joints in the virtual model to the image plane was scaled down, called the joint depth, to the image scale as the third dimension of the pose. This data augmentation technique was applied to make these synthetic images realistic. To train a CNN model that estimates the 3D pose of the given excavator images, three loss functions were defined to correct the 2D pose estimation error, the error in the estimation of the joint depth, and the kinematic constraint error to place all the excavator joints in a plane, respectively. In the testing stage, the CNN model trained with virtual images was tested with video frames of a real excavator, and the estimated 3D poses were converted to angular poses

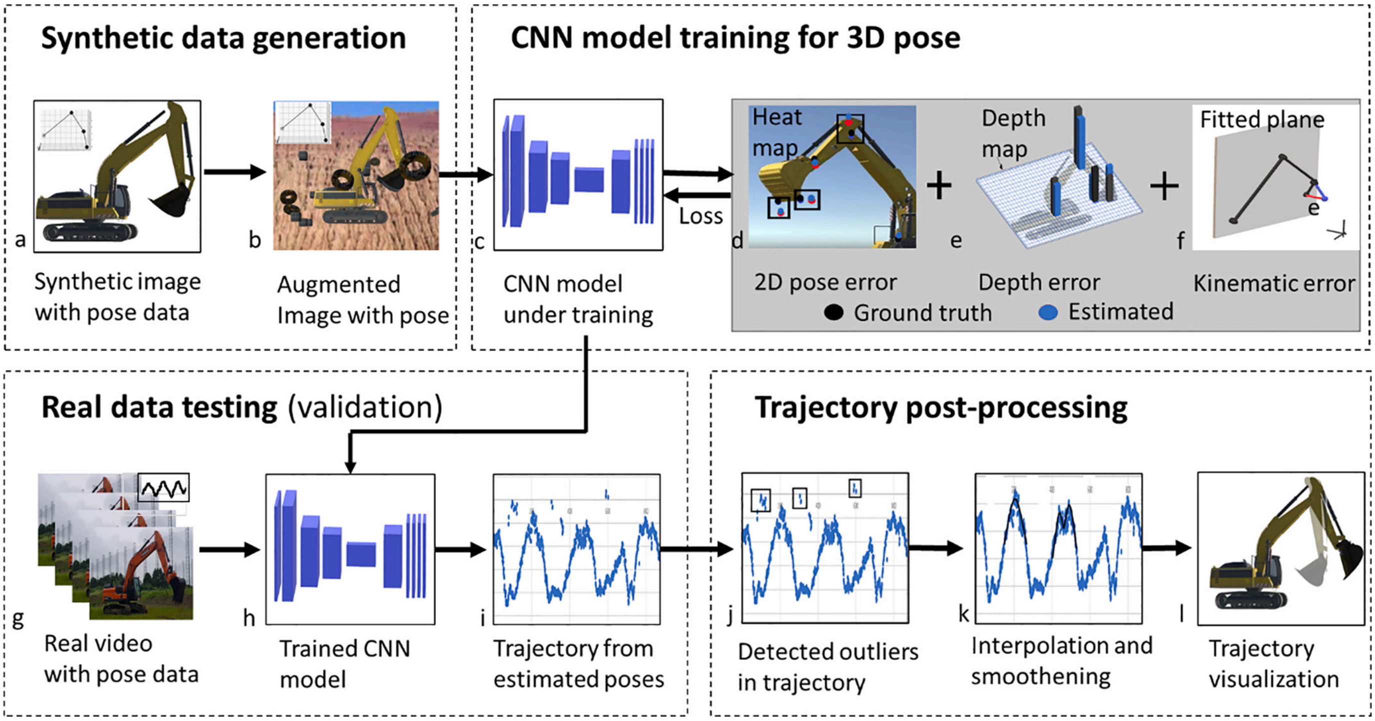
Synthetic data generation
a
Synthetic image with pose data
b
Augmented Image with pose
CNN model training for 3D pose
c
CNN model under training
Loss
Heat map
d
2D pose error
Depth map
e
Depth error
Fitted plane
f
Kinematic error
● Ground truth
● Estimated
Real data testing (validation)
g
Real video with pose data
h
Trained CNN model
i
Trajectory from estimated poses
Trajectory post-processing
j
Detected outliers in trajectory
k
Interpolation and smoothening
l
Trajectory visualization

Fig. 1. Research overview for evaluation of the CNN-based excavator 3D pose estimation

to compare the result with the ground truth (i.e., poses measured using sensors). The video frames of a real excavator whose elements rotate randomly served as the input to the CNN model trained on synthetic images. To evaluate the 3D pose estimation performance, these CNN-based estimated poses were compared with ground truth poses. In this study, the ground truth was the angular pose data of the boom, arm, and bucket collected with IMU sensors that were pre-calibrated and attached to the excavator elements. The estimated poses collectively form a trajectory of the element's activity. This vision-based pose estimation can be further improved if the consecutive poses are considered as a trajectory. By correcting this trajectory, incorrectly estimated poses can be excluded. In the post-processing stage, the error in the CNN-based excavator pose estimation was further reduced by utilizing a temporal sequence of poses called the trajectory. In the trajectory, incorrect

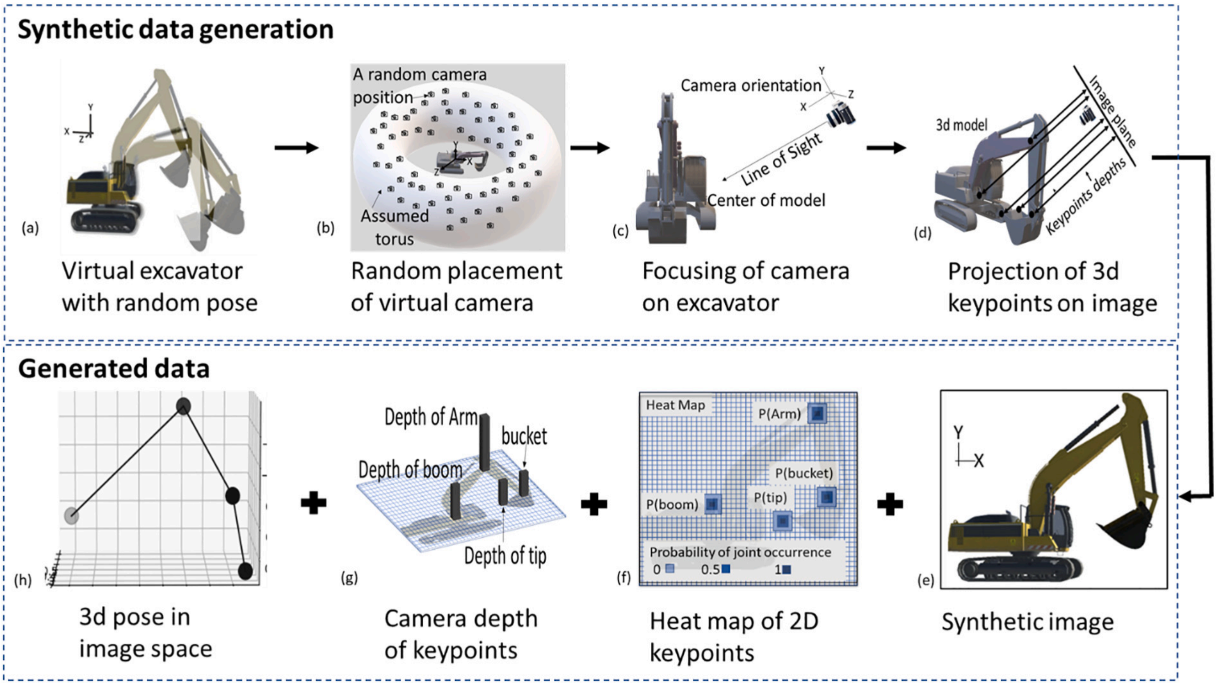

Fig. 2. Description of the processes involved in synthetic data generation and generated data with these processes. In synthetic data generation, (a) virtual excavator poses are generated randomly, (b) virtual camera positions around the excavator are selected, on these positions, (c) camera is focused on the excavator, and then (d) excavator joints position are projected on the image plane. As a result of these processes, the following data is generated: (e) generated synthetic image, (f) projected key-points represented as heatmap, (g) camera depth represented as a depth map, and (h) 3d pose of excavator scaled-down in image space.

estimations may occur when the excavator's joints are self-occluded. These errors can be detected using the camera viewpoints or traditional outlier detection algorithms. These incorrect estimations were excluded from the trajectory, and then the correct pose was estimated by interpolating and averaging the surrounding poses in the trajectory.

3.1. Synthetic data generation

To train a CNN model for estimating the 3D pose of an excavator with an excavator image provided, a dataset of synthetic images (e.g., Unity images) and the corresponding excavator poses were generated. For this dataset generation, a virtual excavator model was simulated to mimic the actual excavator poses, and then virtual cameras were placed around this model to capture the images and the camera relative poses of such poses. Furthermore, data augmentation techniques were applied to these images to generate realism in them. For this synthetic dataset generation, four steps were followed: (1) generating poses in the virtual excavator model for mimicking the real poses of the excavator (Fig. 2a), (2) placing a virtual camera around the excavator (Fig. 2b), (3) orienting the camera towards the virtual model (Fig. 2c), and (4) projecting the model with its keypoints on a virtual image (Fig. 2d). The dataset generated from this process includes the synthetic images (Fig. 2e), the keypoint locations in the image represented as a heat map of excavator joint positions (Fig. 2f), and the camera depth of excavator joints (Fig. 2g). These keypoint locations in the image and camera depths of the keypoints were combined to create the 3D pose of the excavator (Fig. 2h).

To generate the real excavator poses in a virtual environment, the poses of the virtual model were characterized based on the local rotation angle of the excavator's joints. These angles were randomly generated within the kinematic constraints of the excavator. To implement these pose angles in the virtual model, the joints of the virtual excavator model were first simulated using forward kinematics, and the pose angles were then applied to these simulated joints using rotation matrices. Forward kinematics are considered to calculate the position of the end effector of each joint in the chain of joints after applying a local rotation angle. For the simulation of the forward kinematics of the excavator in the virtual model, the joint type, rotation axis, and rotation matrices of the joints were defined to calculate the position of the end effector of every joint in the chain for relative rotation angles. An excavator has revolute joints that rotate along a single axis. The boom, arm, and bucket joints in the excavator rotate along the horizontal axis. We consider θ_1, θ_2, and θ_3 as the angles of the boom, the arm, and the bucket with the horizontal, respectively. Thus, considering the corresponding joint as the origin, the locations of the boom, arm, and bucket ends can be calculated using Eqs. (1–6) [46]. Using these equations and choosing the realistic angles for the boom, arm, and bucket, the excavator poses can be mimicked in the virtual model.

$$xb = l_b cos\theta_1 \tag{1}$$

$$yb = l_b sin\theta_1 \tag{2}$$

$$xa = l_b cos\theta_1 + l_a cos(\theta_1 + \theta_2) \tag{3}$$

$$ya = l_b sin\theta_1 + l_a sin(\theta_1 + \theta_2) \tag{4}$$

$$xbck = l_b cos\theta_1 + l_a cos(\theta_1 + \theta_2) + l_{bck} cos(\theta_1 + \theta_2 + \theta_3) \tag{5}$$

$$ybck = l_b sin\theta_1 + l_a sin(\theta_1 + \theta_2) + l_{bck} sin(\theta_1 + \theta_2 + \theta_3) \tag{6}$$

where (xb, yb), (xa, ya), and (xbck, ybck) are the coordinates of the boom, arm, and bucket ends, respectively; θ_1, θ_2, and θ_3 are the angles of the boom, arm, and bucket, respectively.

To place a virtual camera around the excavator, a zone around the excavator needs to be defined such that the following conditions are met: a camera placed anywhere inside that zone should be able to capture the entire excavator, and the zone should not be at the exact top or bottom of the excavator . This zone can be defined as the horizontal torus around the excavator model (Fig. 2b). Inside this zone, the camera can be placed either at a uniform distance interval [21] or at a random position. In this study, the random placement of virtual cameras inside specified torus zone was employed because this method is comparatively easy to implement and automatically generates a balanced dataset for all the possible camera poses. To define this zone, the center of the virtual excavator (Eq. 8) is considered as the center of the torus. A major radius (R) and a minor radius (r) of the torus are selected based on the field of view of the camera. To generate the points where the camera will be placed, a random point is first selected at the centerline inside this torus using Eqs. (9 and 10). Then, the camera position is selected by randomly generating a camera position in an assumed sphere around this selected point (Eqs. (12–15)).

$$\alpha = \mathrm{Rand}(0, 2\pi) \tag{7}$$

$$\mathrm{CExcavator} = \mathrm{Xc}, \mathrm{Yc}, \mathrm{Zc} \tag{8}$$

$$\mathrm{Xt} = \mathrm{Xc} + \mathrm{RCos}\alpha \tag{9}$$

$$\mathrm{Zt} = \mathrm{Zc} + \mathrm{RSin}\alpha \tag{10}$$

$$\Phi, \Theta = \mathrm{Rand}(0, 2\pi) \tag{11}$$

$$\mathrm{x} = \mathrm{rSin}\ \Phi \mathrm{Cos}\ \Theta \tag{12}$$

$$\mathrm{y} = \mathrm{r\ Sin}\ \Phi\ \mathrm{Sin}\ \Theta \tag{13}$$

$$\mathrm{z} = \mathrm{r\ Cos}\ \Phi \tag{14}$$

$$\mathrm{Camera} = \mathrm{Xt} + \mathrm{x}, \mathrm{Yc} + \mathrm{y}, \mathrm{Zt} + \mathrm{z} \tag{15}$$

To orient the camera towards the virtual model, it is rotated around its own axes such that the line of sight of the camera passes through the center of the excavator model (Fig. 2c). This rotation of the camera ensures that the entire excavator is visible in the image. To calculate the rotation angles of the camera, a direction vector is calculated from the camera to the center of the model (Eq. 16). Then, from this direction vector, the Euler angle along the horizontal axis for orientation in the up-down direction is called the pitch (Eq. 17); the Euler angle along the vertical axis for orientation in the left-right direction is called the yaw (Eq. 18). The excavator model in the plane of the camera is realized by rotating the camera along the pitch and yaw.

$$\mathrm{d.X}, \mathrm{d.Y}, \mathrm{d.Z} = \mathrm{C.x} - \mathrm{E.x}, \mathrm{C.y} - \mathrm{E.y}, \mathrm{C.z} - \mathrm{E.z} \tag{16}$$

$$pitch = sin^{-1}(-d.Y) \tag{17}$$

$$\mathrm{yaw} = tan^{-1}\ (\mathrm{d.X}, \mathrm{d.Z}) \tag{18}$$

where C.x, C.y, and C.z are the global coordinates of the camera, and E.x, E.y, and E.z are the global coordinates of the center of the excavator.

To project a model with its keypoints on a virtual image, the virtual camera generates a virtual image of the visible excavator surface using a ray-tracing algorithm [47] and calculates the image coordinates of keypoints using a camera projection matrix. Ray-tracing algorithms project a ray of light from each pixel of the image towards the model and store the reflected visual information. This reflected visual information usually provides information of the color and illumination in the reflected light, which is stored in each pixel of the virtual image. To calculate the image coordinates of the keypoints, the world coordinates of the keypoints are transformed to a camera coordinate system using a camera transformation matrix, and the image coordinates are calculated using a perspective transformation matrix [48]. In perspective projection, a frustum-shaped camera field of view is converted into a cube to create a square image (Fig. 3). Owing to the perspective projection,

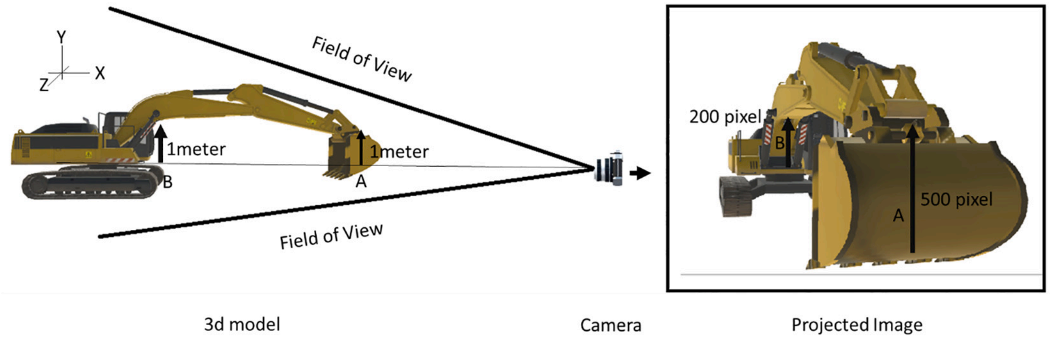

Fig. 3. Illustration of camera perspective projection. Heights A and B appear similar in 3D after camera projection in image space.

distant objects look smaller than close objects. For example, in Fig. 3, the body of the excavator (B) appears smaller than bucket (A) because the bucket is near the camera.

The dataset generated from the above process includes (1) a virtual image of the excavator, (2) image coordinates of the keypoints, (3) camera depth of keypoints, and (4) 3D pose of the excavator. The image coordinates and camera depth of the keypoints are obtained as the output of the pose-estimating CNN model and collectively represent the 3D pose of the excavator. These keypoints are single pixels in an image with a 100% probability of occurrence. However, it is difficult to train a CNN model for a single-pixel target; therefore, a heat map is generated for each keypoint of the joint. In a heat map, the target pixel of the keypoint is assigned a 100% probability of occurrence, and the surrounding pixels to keypoints have a lower probability of occurrence. To calculate the depth of each keypoint, the shortest distance between the excavator joint and image plane is measured. These depths correspond to real-world distances that need to be scaled down to the image scale to preserve the length ratios of the excavator elements in the image space. Usually, a weak perspective projection is used to calculate the scaling factor and thus generate the depths in an image space [30]. The weak perspective projection model reduces computational complexity by assuming that the size of the object is very small compared to the camera depth [49] and that a single scaling factor is sufficient to estimate the entire object in the image. This scaling factor is calculated by calculating the distance between the two farthest keypoints (i.e., the joints) in the image and in the real world and then calculating the ratio of these distances [30]. By multiplying this scaling factor with the real depth of the key point, the depths are made compatible with the image.

Image augmentation on a synthetic dataset is applied to increase keypoint detection in the real image. Image augmentation increases the diversity of the dataset to avoid overfitting in the CNN model training process and to train a model for noise in a real-world environment, which may be caused by poor lightning/weather conditions, occlusion, or quantization error in image capturing. It helps the CNN model to learn the difference between the target object shape and surroundings. To ensure that the image is realistic, this generated image is augmented on a real-world scene. These scenes are construction sites, mountains, fields, and river banks (Fig. 4b). In addition, flying distractors (e.g., cubes, pyramids, and gears) are placed around the excavator model to train the model to differentiate between similar shapes of the surrounding objects (Fig. 4c). During the training of the CNN model, Gaussian noise is added to the RGB values of the images; colors are inverted; the contrast is changed, and the image is compressed. An example of such an augmentation is shown in Fig. 5.

3.2. CNN model training

A CNN model has learnable weights and biases that are multiplied by input image features to classify an image into defined classes or to perform other tasks such as detection and regression. These weights are learned iteratively by back-propagating the error of classification and then adjusting the model weights for this error until the CNN model can be classified correctly. In this study, the CNN structure and input/output data type used were the same as those used in human pose estimation [30]. This structure had the shape of an hourglass; in the initial layers, the height and width of the features were reduced using convolution; in the subsequent layers, the height and weight were increased again using deconvolution (Fig. 6b). The input of this structure was a three-channel (red, green, and blue) image (Fig. 6a). The output of this structure was the classification of each pixel for each keypoint, represented as the heat map of each keypoint and the depth of each joint. Two losses have been conventionally considered in human pose estimation: 2D loss and depth loss. For 2D loss, the ground truth heat map is subtracted from the CNN-generated heat map (Eq. (19)). Ground truth heat maps were generated for each joint position in the image. The CNN model classifies each pixel of the image for a specific joint class and then generates the output heat maps. For depth loss, the difference in the estimated depth of each joint in the image space with the ground truth depth was calculated. In

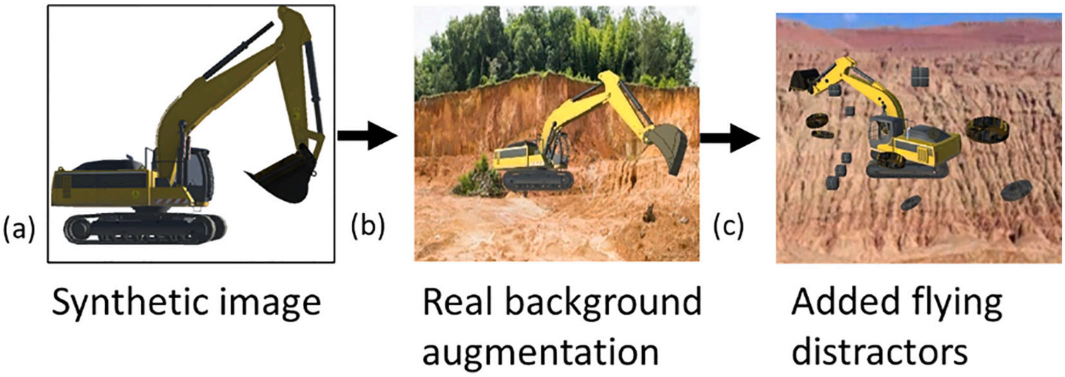

Fig. 4. Training dataset preparation: (a) synthetic image, (b) rendered background, (c) rendered background and flying distractors.

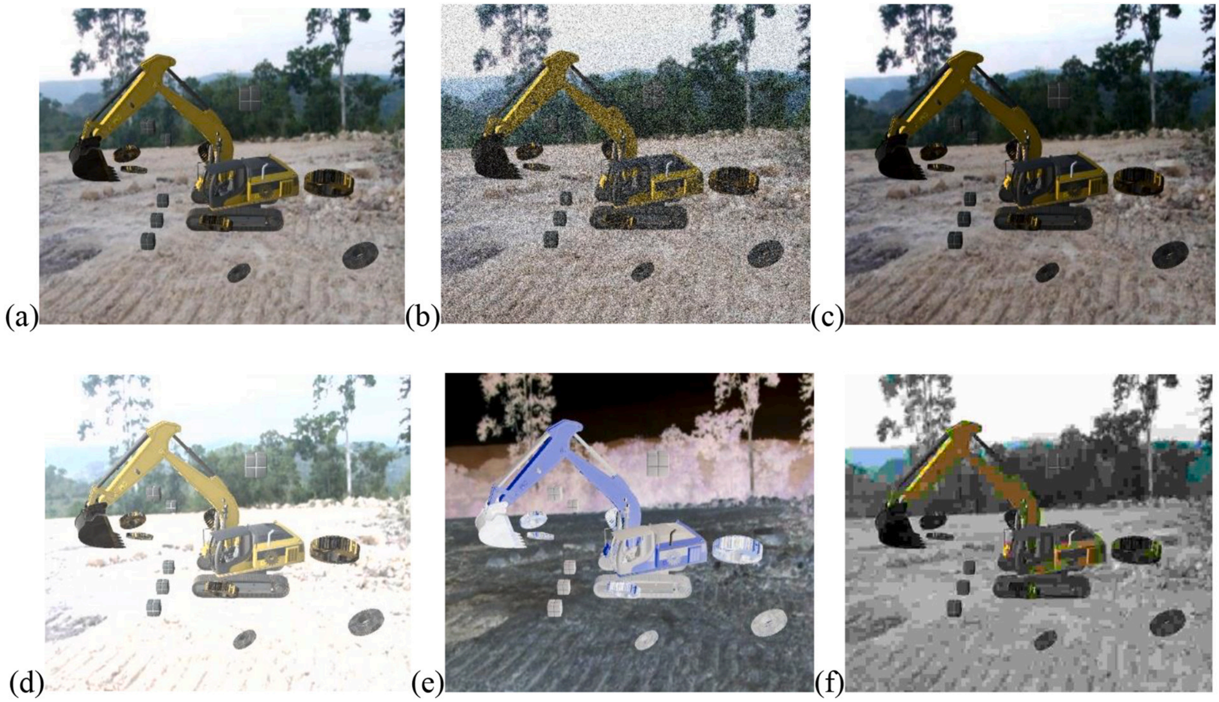

Fig. 5. Application of image augmentation: (a) original image, (b) Gaussian noise, (c) gamma contrast, (d) brightness, (e) invert colors, (f) compress image

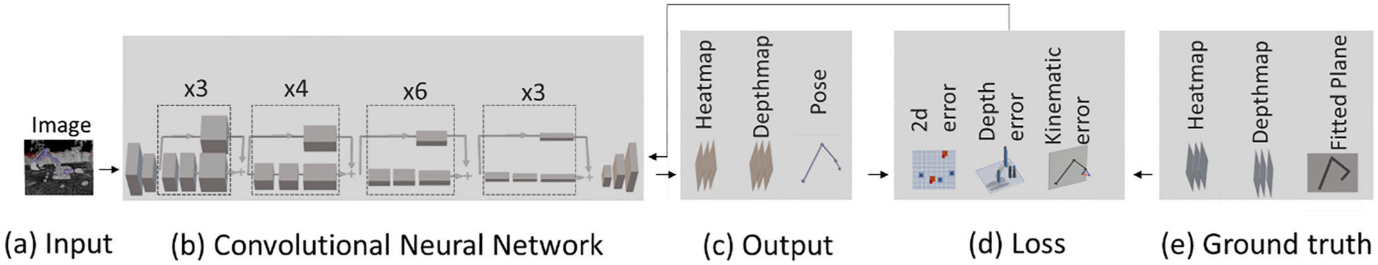

Fig. 6. CNN model training for excavator 3D pose estimation

$$\text{Mean square error} = \left(\sum |y - \hat{y}|^2\right) \Big/ n \tag{19}$$

addition to these two losses, an additional kinematic loss was introduced in this study to compensate for the error in the scaling of depth owing to nonlinear perspective projection. A single factor for scaling the joint depth in the image scale may cause errors in the 3D pose [50] when the camera is near the excavator. To correct this error in the pose, the kinematic constraint of the excavator's joints was used. According to this constraint, the excavator's joints reside on a geometric plane; when the estimated excavator's joints do not follow this constraint, a pose error occurs. Therefore, a third loss, referred to as the kinematic loss, is also added in the CNN model training to ensure that the joints of the excavator are in a plane. Kinematic loss refers to the difference between the corrected bucket end depth in the ground truth and the estimated bucket depth end (Fig. 7). The bucket end is chosen for this loss when a large pose estimation error is observed in the bucket end [25]. To correct the bucket's end depth in the ground truth, the boom joint is considered as the origin of the coordinate system, and then a plane is fitted on the joint coordinates. A linear regression model is used to fit the plane (Eq. 20). Image coordinates in the form of row and column indices, are considered as the independent variables of the regression model, and the corresponding depths are considered as the dependent variables. Subsequently, the coefficients of this plane equation are used to calculate the depth of a bucket end using the average bucket end coordinates. This bucket depth is the ground truth, which is compared with the CNN-estimated depth to correct the kinematic loss.where y is the actual pixel value, $\hat{Y}$ is the predicted pixel value, and n is the number of pixels.

$$z = w_0 \, x + w_1 \, y \tag{20}$$

where z is the scaled depth; x, y are image pixel indices, and w_0, w_1 are coefficients.

3.3. Real data testing

To test this trained CNN model, video images of a real excavator were used as the input to the trained CNN model and were compared with the ground truth pose. To capture real video images, a video camera was set on the ground near the excavator, and a video was recorded while the boom, arm, and bucket were rotated randomly. These captured images were individually fed to the trained CNN model for estimating the 3D poses of the excavator. The parameters of these estimated poses were the 3D coordinates of the keypoints of the excavator in the camera's frame of reference (Fig. 8). These pose parameters need to be homogenized for

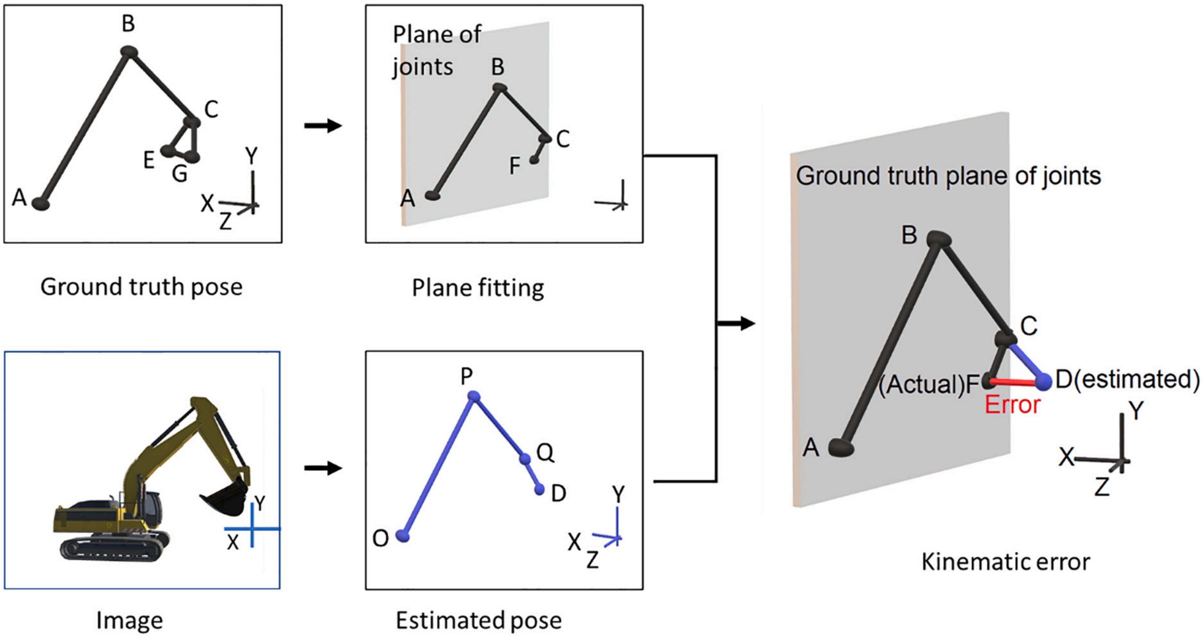

Fig. 7. Description of kinematic constraint-based error in pose estimation

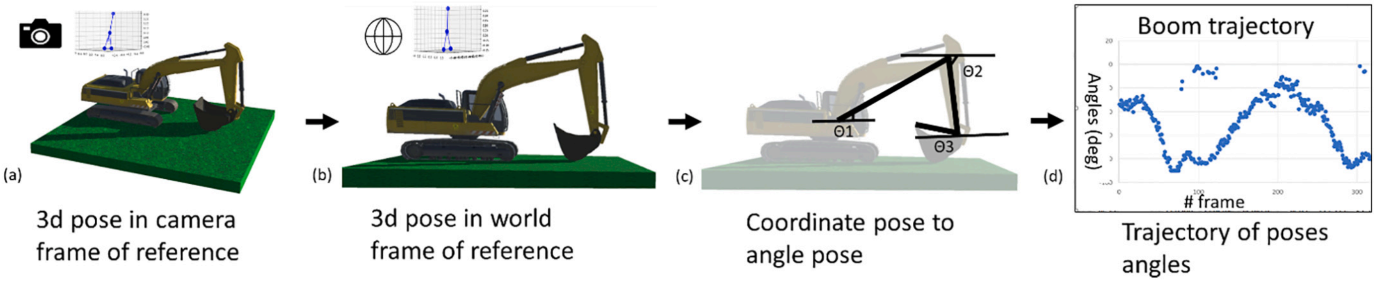
(a)
3d pose in camera frame of reference
(b)
3d pose in world frame of reference
(c)
Θ1
Θ2
Θ3
Coordinate pose to angle pose
(d)
Boom trajectory
Angles (deg)
# frame
Trajectory of poses angles

Fig. 8. Compatibility of vision-based and sensor-based estimated pose: (a) 3D model and its joints in the camera's frame of reference, (b) 3D model and its joints converted into the world frame of reference, (c) pose angles compatible with sensor-based angle, (d) trajectory of poses

concatenating poses and simplified for further motion analysis [51]. To homogenize the estimated poses, these poses were transformed to a world frame of reference (Fig. 8b). Subsequently, to simplify these homogenized poses, rotation angles around the excavator's joint were calculated (Fig. 8c). For the homogenization of poses, the excavator was assumed to be on a horizontal ground such that the fitted plane on the excavator's joints was perpendicular to the ground. To transform the estimated pose in the world frame of reference, a transformation matrix was calculated by considering the boom joint as the origin and then making a normal vector of the fitting plane horizontal. To simplify the three coordinate parameters to one rotation angle around the joint, the angles of the boom, arm, and bucket were calculated with respect to the horizontal. These simplified poses were then concatenated to create a trajectory for the motion analysis (Fig. 8d). The ground truth pose calculation is described in the experimental section.

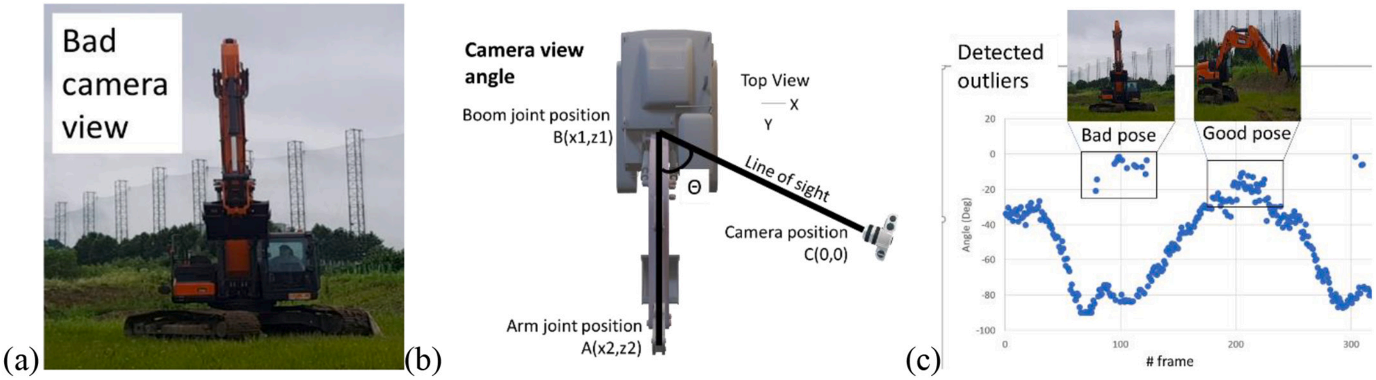

Fig. 9. Detection of bad view pose: (a) bad camera views that hide the joints behind the bucket or arm, (b) description of camera's view angle, (c) detected outliers using bad camera view angle.

3.4. Trajectory post processing

To improve the CNN-based pose estimation, the trajectory formed by consecutive pose angles (Fig. 8d) is post-processed by detecting the incorrect estimation (Fig. 9c) using an outlier detection algorithm and then adjusting the trajectory using interpolation and smoothing (Figs. 10 and 11). These incorrect estimations can be detected either by applying the traditional outlier detection method or by detecting bad camera views, which are the views where self-occlusion occurs. In the traditional method, a statistical model is fitted (e.g., ARIMA model and the slope of trajectory), and then a threshold is set to screen out the anomaly in the time series of the trajectory. For bad camera view detection, a pose of the excavator is detected with respect to the camera such that self-occlusion occurs at this pose; for example, the front view of the excavator hides its backside in the image (Fig. 9a). This view can be detected by calculating the angle between the boom vector (AB in Fig. 9b) and the camera's line of sight (CB in Fig. 9b). The traditional method of outlier detection can be used when the trajectory follows a specific trend. However, the trajectory in our experiments did not follow any trend because it was generated by random rotation of the excavator elements. Because of the stochastic nature of the trajectory, a camera view filter was used for outlier detection. To adjust the trajectory after removing the bad estimations, missing values were interpolated using spline curve fitting (Fig. 10), and then this interpolated trajectory was smoothened using a moving average (Fig. 11). Spline curve fitting breaks down the trajectory into pieces, and a polynomial function is fitted to each piece. However, this corrected trajectory has irregularities at the overlapping points of the trajectory pieces, which render the trajectory unrealistic and may cause an error in the estimated pose trajectory. Irregularities in the trajectory can be removed using a moving average (Fig. 11). The moving average slides a window of fixed length with fixed intervals over the trajectory values, and the average value of this window is used to correct the irregular points.

4. Implementation experiment

Experiments were conducted on images of different excavator types for evaluating the generalization of vision-based pose estimation when a CNN model was trained on a synthetic dataset and tested on a real dataset. Additionally, post-processing techniques were tested to improve pose estimation. The virtual excavator model (Fig. 2a) was used to generate a training dataset. This virtual excavator model had dimensions and shapes similar to those of the real excavator used in our experiments. This virtual model was captured using a virtual camera with a focal length of 142 mm and a sensor size of 200 mm × 200 mm. This camera was placed 15 to 30 m away and at a height of −15 to 25 m from the center of the virtual excavator. These parameters were selected to ensure that the full excavator was visible in the image. Synthetic images generated from these camera parameters do not initially have occlusions, lightning and site conditions, or dirt on the excavator. To introduce these natural features of construction into synthetic images, data augmentation techniques are applied. Fig. 13a shows the real excavator that was used to generate a testing dataset. For the testing dataset, 300 images of wheeled, dragline and crawler excavators (100 of each category) were labeled manually for the 2D pose. These images had various models, sizes, colors and scales of excavators. Also, 100 images of the real excavator, which was under experimentation, were also labeled for 2D pose manually. Scale variability of the testing dataset in terms of image resolutions is shown in Fig. 12. Here, an image with a scale of 1 has a resolution of 256 pixels (i.e., the input size of the CNN model). The testing dataset for 3D pose estimation of a specific excavator model was captured using a Samsung Galaxy S8 Plus camera. The excavator and camera were operated on horizontal ground, and the camera was placed at a distance between 20 m to 30 m at a height of 1.5 m. For training data generation, 3000 synthetic images and corresponding excavator poses were generated from the virtual excavator model, while the camera was placed randomly at every shot and the boom, arm, and bucket of the real excavator were rotated randomly. For testing the data generation, 2000 video frames were captured, while the boom, arm, and bucket of the real excavator were rotated randomly. To calculate the poses of the excavator while its movements were being recorded, sensors were attached to the boom, arm, and bucket [18] as shown in Fig. 13a. These sensors had an inertial measurement unit for measuring the rotation angle of the excavator element with respect to the initial position (Fig. 13a). Video frames and sensor-based poses were calibrated manually for an initial pose and for an interval of 1/10 s afterward to determine the correspondence between them. The recorded sensor data were calibrated with the initial frame by finding a factor whose addition made the first recorded angle with respect to the horizontal plane. This calibration factor was added to all the sensor data. In contrast, the CNN-based pose was estimated as the positional pose that was made compatible with the sensor-based pose by converting the coordinates of the key-point from the image space (Fig. 8a) to the world space (Fig. 8b). Subsequently, the global angle was calculated using the boom, arm, and bucket vectors (Fig. 8c). Eventually, the CNN-based pose estimation error was calculated by comparing the estimated pose with the sensor-based pose in units of degrees.

For training the data generation, data augmentation was performed by augmenting 45 real scenes and putting flying distractors around the excavator, such as pyramids, cubes, and gears. Other data augmentation techniques such as brightness addition, Gaussian noise addition, color inversion, and image compression were applied to synthetic images before being applied to the CNN model. To increase the brightness, RGB values of the image were added using a randomly selected value between −30 and 30. To add the Gaussian noise, RGB values of the image were added by a randomly selected value between 0 and 25.5. Color inversion was applied at a probability of 15%. In addition, 50–90% of the image compression was applied to synthetic images. These parameters were selected after experimenting with a different range of augmentation and then visualizing the estimated 2D poses from real excavator images.

During the training and testing of the CNN model, the input image was scaled down to 256 × 256 pixels. The output of the CNN model was in the form of 64 × 64 pixel heat maps. The CNN model had 50 convolutional layers of the residual network structure [30], followed by three deconvolutional layers, as used in the implementation of 3D human pose estimation [30]. To train the model faster, pre-trained ResNet 50 layers were used for CNN model training initialization and then fine-tuned in three phases. In the first phase, the model was trained for a heat map of keypoints using 2D loss. In the second phase, after training the model for the heat map, this model was further fine-tuned for additional depth maps using depth loss. In the third phase, the trained model has trained again for depth correction using kinematic loss. In the first training phase, the model was trained with a learning

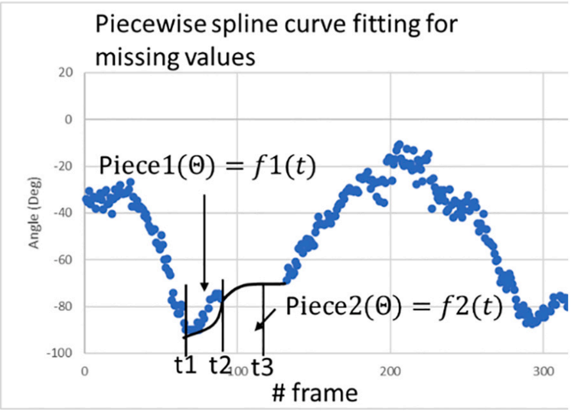

Fig. 10. Description of spline curve fitting for estimating missing values.

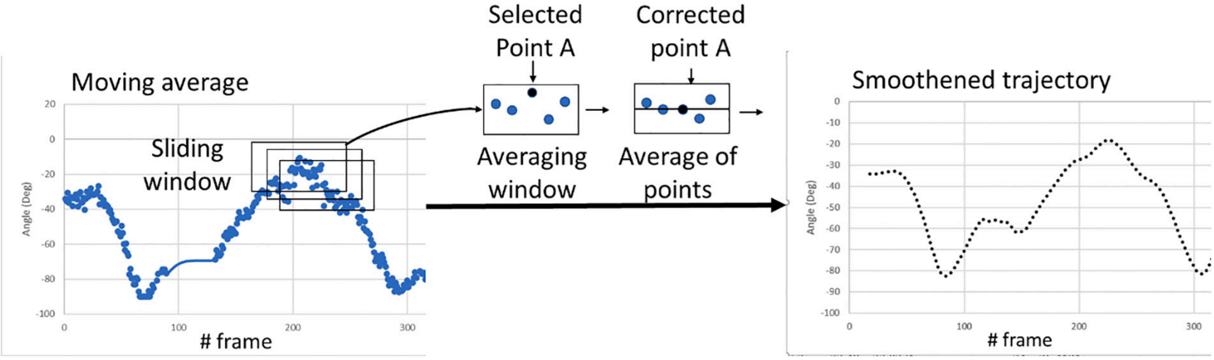

Fig. 11. Smoothing of trajectory using moving average

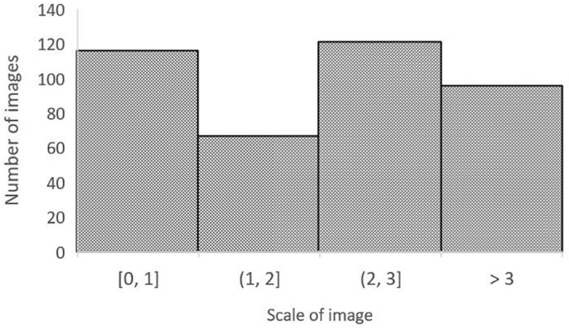

Fig. 12. Scale variability of testing image dataset where scale 1 is of resolution 256 pixels.

rate of 0.0001, which decreased exponentially at a rate of 10^n, where n was 1 for the 3rd epoch and 2 for the 4th epoch. In the second training phase, the learning rate was set to 0.0001, which decreased exponentially at a rate of 10^n, where n was 1 for the 10th epoch and 2 for the 50th epoch, and the depth loss was multiplied by 0.1 to train the model specifically for depth. In the third training phase, the learning rate was set to 0.00005, which decreased exponentially at a rate of 10^n, where n was 1 for the 10th epoch and 2 for the 50th epoch, and the kinematic loss was multiplied by 0.0005 to train the model specifically for kinematic loss. These parameters were initially selected based on [30] and then fine tuned to minimize the training loss. The training of all the phases was continued until the loss became constant.

Post-processing includes three steps: 1) detecting bad pose estimations, 2) adjusting the trajectory, and 3) smoothing the trajectory. To detect bad pose estimations, two methods were tested: the traditional method of outlier detection and bad camera view detection. In the traditional method, the slope of the trajectory was calculated using the difference of two consecutive angles in the trajectory, and in this slope trajectory, the outlier was the point larger than the average slope of the trajectory. For the camera's view filter, poses where the boom had a horizontal angle of 90° with the camera line of view were excluded (Fig. 9c). To adjust the trajectory, spline curve fitting of order one was applied to interpolate the missing data points. The order of the spline curve was selected as one because the spline curve of order one does not consider the overall trend of the complete trajectory; instead, it fits a linear curve locally [52]. This poses a trajectory from random movements of excavator parts, and this trajectory does not have any systematic trends. To smooth the trajectory, a moving average was applied to the smoothing trajectory. The moving average slides a window of specific length over the trajectory values and adjusts the abnormal value in that window. An experiment was conducted to determine the optimal window size, where the pose estimation error was calculated after applying a moving average to the trajectory, and the window length was varied from 1 to 10 at an interval of 1 frame.

For evaluating the generalization of the CNN model, 2d pose estimation performance was calculated for the different types of excavators using the percentage of correctly localized key-points. For considering a key point as localized, the same localization definition is used as used in human 2D pose estimation; a key point is localized when the distance between the estimated position and the ground truth position is less than 50% of the normalized head segment length—this distance is called as the threshold [53].

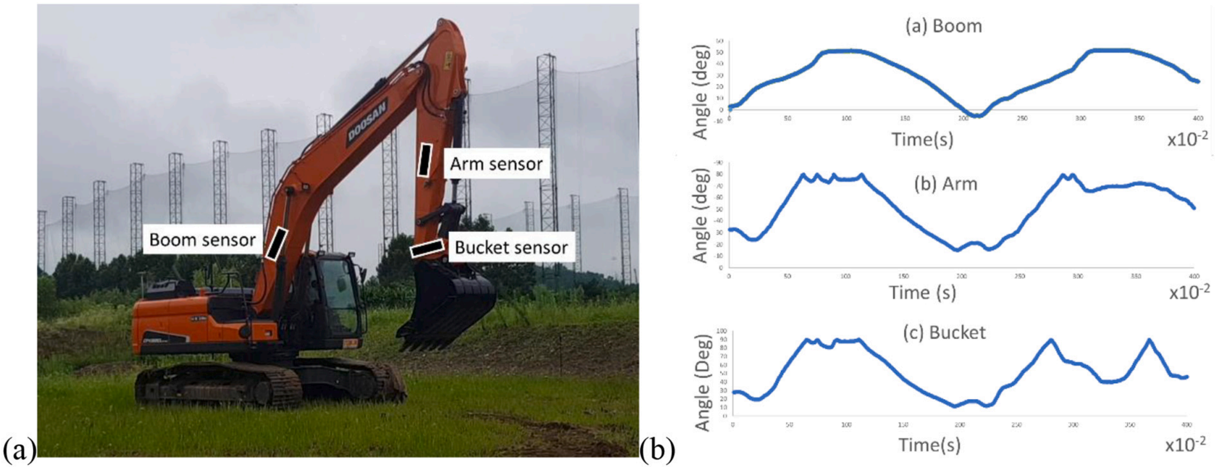

Fig. 13. Description ground truth data: (a) sensor locations to measure the pose angles of the boom, arm, and bucket; (b) trajectory formed by a sequence of measured poses angles.

To evaluate the performance of the proposed kinematic loss function and the proposed post-processing, called the treatments, two matrices were used: 1) absolute pose estimation error, which was computed by comparing the estimated pose with the sensor-based estimated pose, called the ground truth, and 2) a *t*-test comparing the mean absolute error before and after the application of the treatments. The t-test compares the difference in the mean error of the two sets of values. If the mean error in both the results is different, the applied treatment has a significant effect on the performance of the pose estimation. To validate the significance of the tested treatments, for example, the use of kinematic loss in training and then the use of the post-processing stage, a t-test was conducted. In the t-test, we hypothesized that the mean error of pose estimation from the treatment of kinematic loss and post-processing of trajectory was the same as that of the CNN-based pose estimation. This hypothesis was rejected if the probability of occurrence was less than 0.05.

5. Results

Experiments were conducted to evaluate the proposed treatment and identify the optimized parameters, through which the overall performance of the excavator 3D pose estimation was assessed. The results are presented and discussed from the following aspects: 1) 2D pose estimation results to verify the effectiveness of data augmentation on synthetic data, 2) window length for moving average, and 3) performance of 3D pose estimation in terms of error in estimated angle and the significance of applied treatment compared to pose estimation error without treatments.

First, data augmentation was applied to the synthetic images, and the level of augmentation was determined based on 2D pose estimation in real images. The appropriate level of augmentation was selected by visually inspecting the pose estimation accuracy, as it was not possible to label the 2D poses in images of the jobsite. The pose estimation results for the real images, after applying the augmentation described in the experimental section, are shown in Fig. 14. The results showed that the augmentation techniques applied to the synthetic images assisted in the development of visual features present in the real images, and by training a CNN model with such augmented images, the detection capability of the model to detect the excavator parts in real images increases.

Second, after removing incorrect pose estimations from the trajectory and filling the gaps with the spline curve, the trajectory is smoothened using a moving average to further improve the pose estimation accuracy. The moving average takes a set of poses, called a window, from the estimated poses trajectory, and then corrects the abnormal value in that window using the average of that window. However, this smoothening could worsen the results. For instance, increasing the moving average window length can disturb the trend, and lowering that length will make no difference in the results. Fig. 15 shows the effect of the window length on the pose estimation error in the bucket. The results showed that increasing the window length to four poses (e.g., the four frames) reduced the pose estimation error; however, further increase in the window length did not change the overall performance. This might disturb the trends in trajectory; hence, in this experiment, a moving average with a window length of four poses was considered suitable for improving the trajectory without disturbing the trends.

To evaluate the performance of the pose estimation methods and the proposed treatment used in the method, the error in the estimated pose was calculated. To calculate this error, positional poses (i.e., coordinates

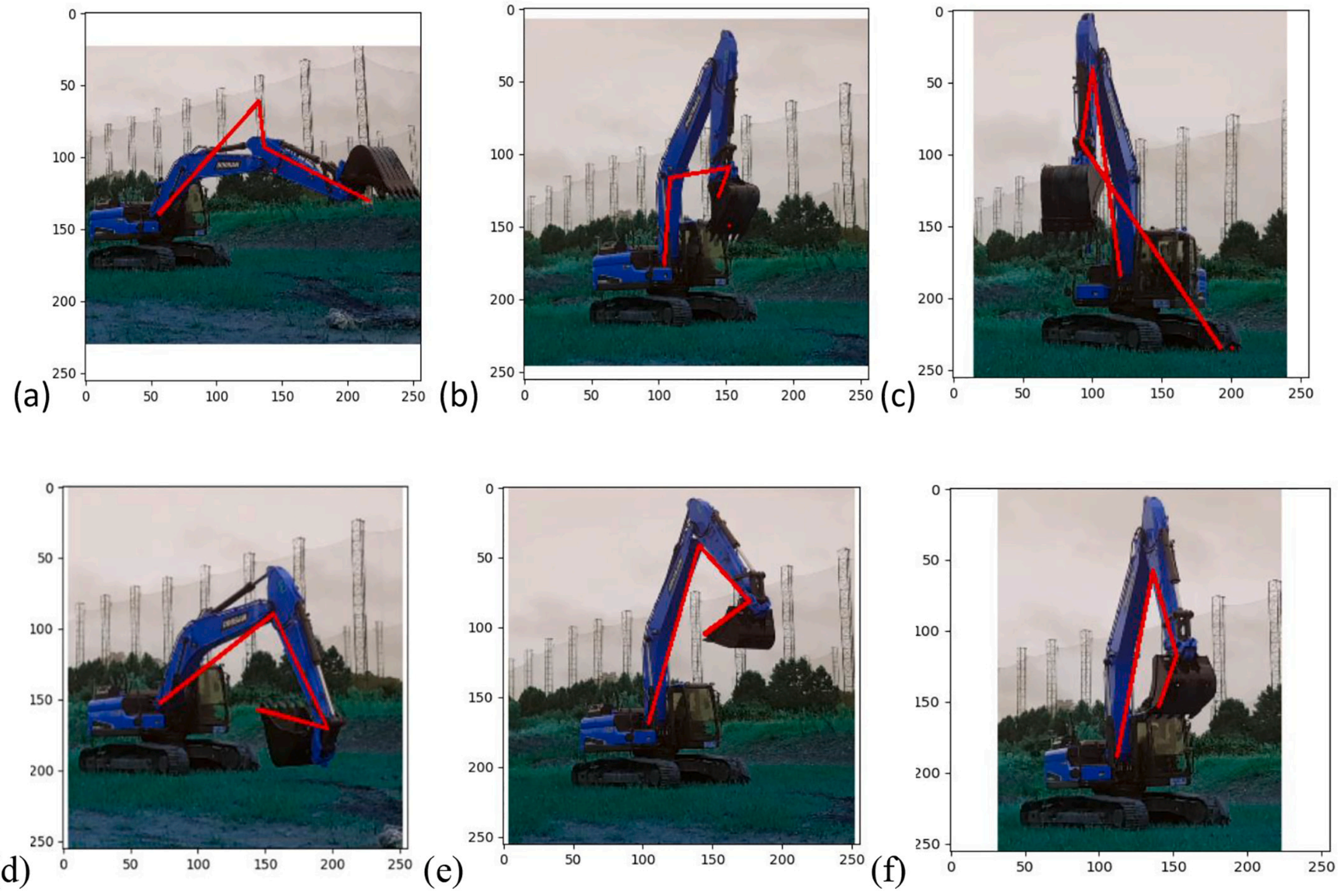

Fig. 14. 2D pose estimation after training a CNN model with synthetic images (a–c); and augmented synthetic images (d–f).

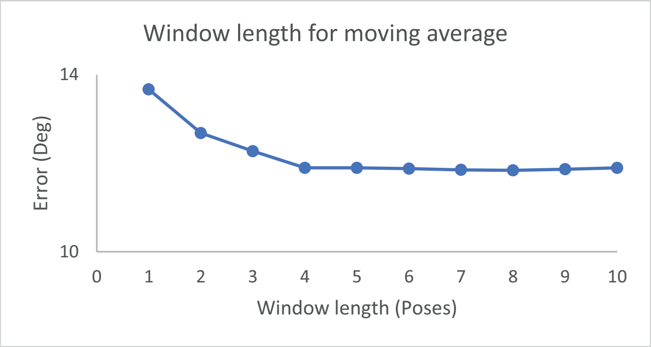

Fig. 15. Selection of window length in the moving average algorithm.

Table 1
2D pose estimation result.

Excavator type	Excavator Element	Accuracy (%)
Crawler, similar to the virtual model	Boom Joint	96.18
	Arm Joint	96.94
	Bucket Joint	94.65
	Bucket end	92.78
Dragline	Boom Joint	58.88
	Arm Joint	54.91
	Bucket Joint	21.56
	Bucket end	33.33
wheel	Boom Joint	83.87
	Arm Joint	87.09
	Bucket Joint	77.41
	Bucket end	67.74
Crawler, different from the virtual model	Boom Joint	92.94
	Arm Joint	95.29
	Bucket Joint	90.41
	Bucket end	74.11

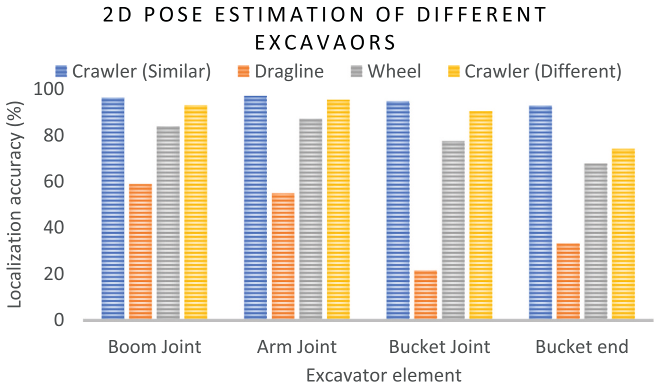

Fig. 17. 2D keypoints localization accuracy in images of different excavator types.

in the image space) estimated from the CNN-based method were converted into an angular pose (e.g., using the method in the real data testing section). Then, this pose was compared with the sensor-based pose. This error was calculated and compared for poses estimated using (1) a CNN trained with the loss function commonly used in the traditional human pose estimation, (2) a CNN trained with the proposed kinematic loss, (3) a CNN trained with the proposed kinematic loss and applying the traditional post-processing techniques (i.e., trajectory slope-based filtering), and (4) a CNN trained with the proposed kinematic loss and applying the proposed post-processing technique (i.e., camera view filter), as presented in Table 2. The significance of each method was evaluated by conducting a *t*-test. For this t-test, the mean of the results from each method was compared to that of a basic CNN-based result.

For evaluating the performances of the proposed method on the images of different types of excavators, 2D pose estimation was performed and assessed in terms of key-point localization accuracy. The results of the 2d pose estimation evaluation show that the CNN model trained using synthetic images of the excavator can detect the 2d key-points accurately when the shape of the excavator is similar. For example, in Fig. 16, key-points of a crawler and a wheeled excavator were detected accurately, while the CNN model performed poorly when the shape of elements in the excavator was different (e.g. a rope in a dragline excavator, instead of a solid arm). Results of key-points detection accuracy are summarized in Table 1, and these results are also represented as a histogram in Fig. 17. Table 1 shows that the CNN model shows relatively poor performance in detecting elements of a dragline excavator. Fig. 17 illustrates that bucket ends show relatively lower localization accuracy.

Overall, the results show that the proposed method slightly outperforms the other treatments, as summarized in Table 2 and Fig. 18. In Table 2, it can be observed that the least pose estimation error is obtained for a CNN trained with the proposed kinematic loss when the camera view filter-based post-processing technique is applied. Additionally, the *p*-value of the *t*-test was less than 0.05, which verifies the significance of the proposed method. Specifically, the highest error occurred on the bucket (i.e., 20.12°), which was reduced to 16.38° (i.e., the second row of 'Proposed Loss Function' in Table 1) after applying the kinematic loss function during the CNN model training. The t-test result of this analysis showed that there is a significant improvement in bucket

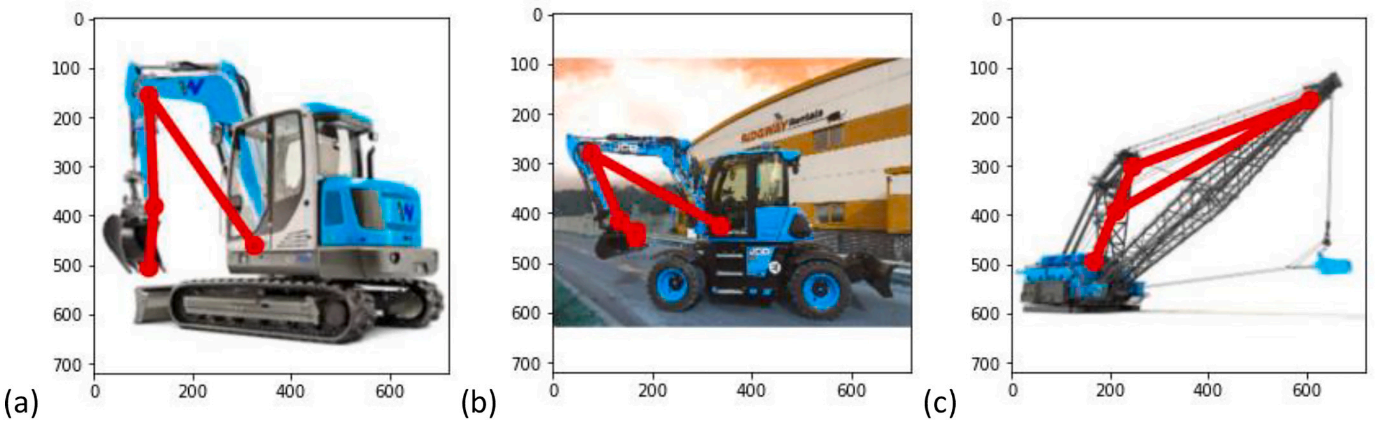

Fig. 16. Examples of 2D pose estimation results for different excavator models; (a) a crawler excavator, (b) wheeled excavator, and (c) a dragline.

Table 2
3D pose estimation results.

Treatment Applied	Excavator element	Pose Error (degree)	Standard Deviation	T value	P value
Traditional Loss Function	boom	9.05	8.53	–	–
	arm	13.79	11.11	–	–
	bucket	20.12	15.42	–	–
Proposed Loss Function	boom	7.74	4.31	6.13	0
	arm	13.16	8.63	21.07	0
	bucket	16.38	14.58	7.88	0
Proposed Loss Function + Traditional Post-Processing	boom	7.71	4.31	6.32	0
	arm	14.36	11.66	0.5	0.55
	bucket	12.51	11.95	17.4	0
Proposed Loss Function + Proposed Post-Processing	boom	7.16	3.67	9.10	0
	arm	9.9	11.21	11.02	0
	bucket	11.85	11.16	19.43	0

pose estimation after correcting the pose for kinematics violation. The significance of this treatment also confirms that the pose error generated due to the perspective camera projection can be corrected if the kinematics of the articulated object can be modeled in a geometric shape (e.g., a plane fitted on joints). This bucket error was further reduced to 11.85° after the incorrect poses were detected using a camera view filter. Additionally, the results showed that, compared to the traditional slope of trajectory-based incorrect pose detection (i.e., the third row in Table 1), camera view filter-based incorrect pose detection (e.g., the fourth row in Table 2) is slightly more effective. The effectiveness of this incorrect pose detection filter proved that incorrect poses mostly occurred in vision-based pose estimation methods when most of the body parts are occluded (e.g., self-occlusion), and these incorrect poses can be detected if such a camera relative pose of the whole body is detected when most of the body parts get self-occluded.

The estimated pose trajectories from the proposed method were plotted with the ground truth trajectory (i.e., sensor-based) to visualize the accuracy of the estimated trajectory (Fig. 19). In these plots, both the trajectories are close to each other, which validates the effectiveness of the proposed method. In addition, these plots show that the overall trajectory trend was preserved. For example, the pose angles of the boom in both trajectories (estimated and ground truth) (Fig. 19a) simultaneously increased from frame number 0 to frame number 100. This concurrency in the plots validates that the proposed method can estimate the excavator poses from the video frames of real job sites.

As a proof of concept for excavator trajectory visualization, the estimated poses were transformed into a virtual model, as shown in Fig. 20. For instance, the estimated pose angles are in the world frame of reference. To implement these pose angles on the virtual excavator model, the kinematic equations of the excavators (Eqs. 1–6) are used to convert these angles into a local frame of reference. Implementation of these angles in the virtual model showed that the estimated poses could be visualized in a virtual environment (Fig. 20).

6. Discussion

The application of machine learning techniques such as the CNN for 3D pose estimation of an excavator is challenging because the harsh jobsite conditions make it difficult to collect the training image datasets with pose labels. Synthetic data generation may help in obtaining a labeled training dataset in a controlled environment. In this context, this study evaluates the performance of a CNN model trained with synthetic images and tested with actual images. From a technical aspect, this study also proposes two modeling and post-processing methods: (1) use of a kinematic loss function in the training of a CNN model to make the estimated pose more realistic, and (2) pose trajectory adjustments to improve the overall accuracy of 3D pose estimation for visualization. For the performance evaluation, experiments were conducted to collect real excavator images with pose labels that were measured using motion capture sensors. The experimental results show that the CNN trained with synthetic data can achieve a mean error of 7.16° at the boom, 9.9° at the arm, and 11.85° at the bucket when applied to actual excavator images. This result implies that the proposed approach using synthetic images generated in a virtual environment may help overcome the difficulties in collecting training data for pose estimation. In particular, it can be observed that the use of the proposed kinematic constraint during training could help correct the errors (e.g., camera-depth) caused by the camera's perspective projection; for example, the bucket pose estimation was improved from 20.12° to 16.38°. In addition, post-processing of the trajectory of poses could adjust the bad pose estimations; for example, the bucket error was further reduced to 11.85°.

Specifically, synthetic images do not have visual features that are present in real jobsite images, such as lightening, occlusions, distortion in images, poor visibility owing to the dust in the air, and natural background scenes. Because of the unavailability of such features, the CNN model trained with these images could not estimate the excavator poses accurately, as shown in Figs. 13a–c. These natural visual features can be artificially generated in synthetic images by adding augmented natural scenes in the background, adding Gaussian noise to represent the dust, adding brightness to represent the natural lighting conditions, compressing the pictures to distort the image for poor visibility, inverting the color to represent the wornness of the excavator, and adding a flying distractor around the excavator to mimic occlusions. The results (Figs. 13d–f) show that by adding this augmentation to the

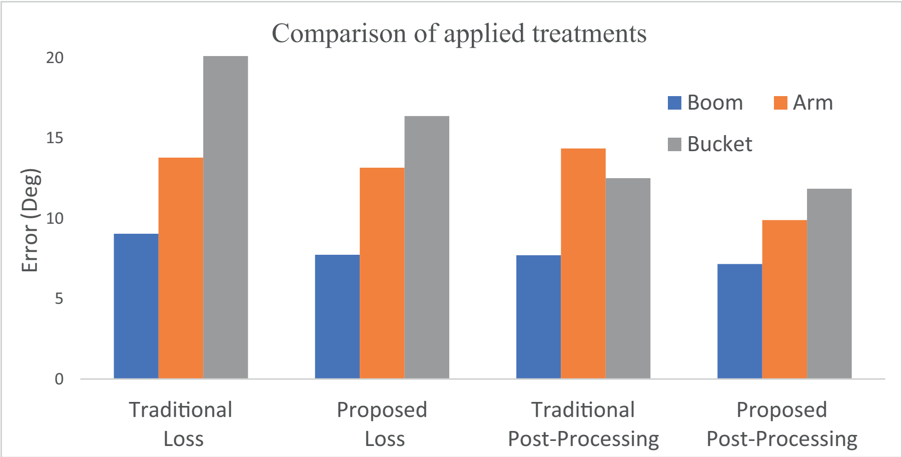

Fig. 18. Pose estimation error from tested methods.

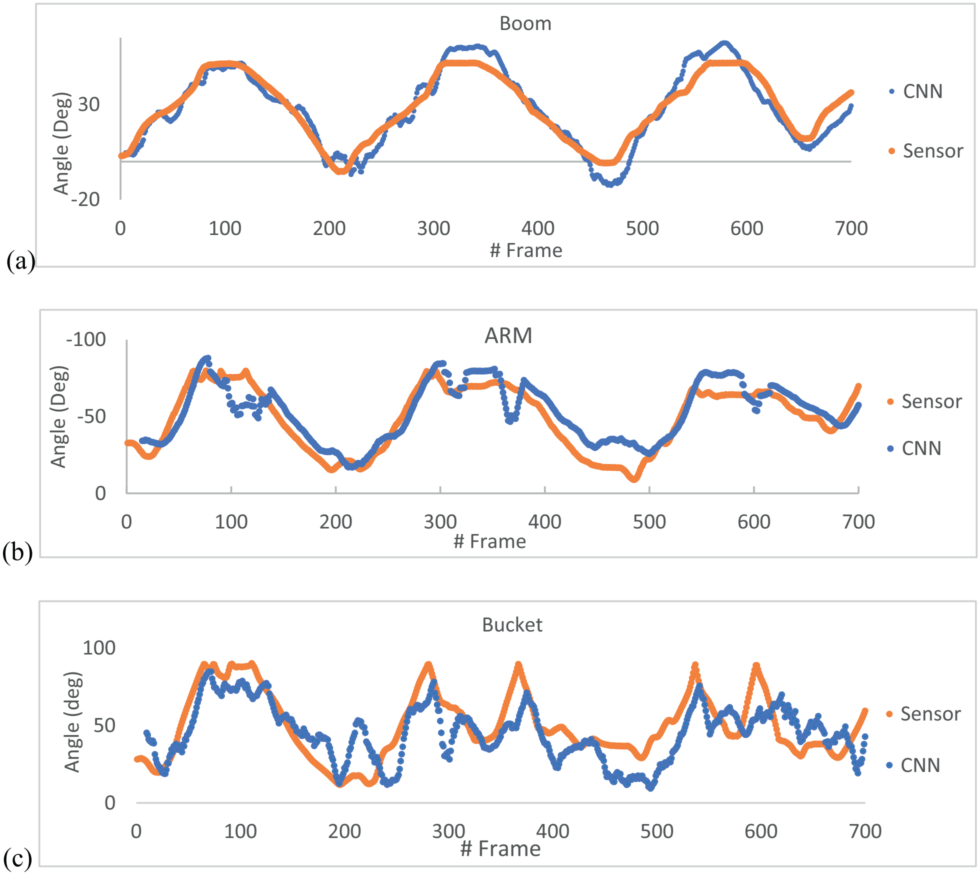

Fig. 19. Actual and estimated trajectory of pose angles for (a) boom (b) arm (c) bucket.

synthetic images, the CNN model could learn about the natural noise present in real images.

Kinematic constraints may be violated in excavator pose estimation owing to the presence of the camera depth error. This depth error occurs because of perspective projection in image generation, and the position of the excavator's joint in the 3D pose is misplaced owing to this depth error. This misplacement of joint position in training data causes errors in CNN-based pose estimation. To correct this estimation error, a kinematic loss was used during the CNN model training, which fits a plane on the excavator's joint to estimate the deviation of the bucket end from this plane. This loss function helps the CNN model to learn the kinematic constraints of the excavator; in return, the CNN model automatically corrects the depth error by fitting a plane on the excavator's joints and restores the kinematics of the excavator. The results showed that the plane fitting on the excavator's joints might better provide a solution for recovering the kinematic constraint of the estimated pose of the excavator to estimate realistic poses.

A motion trajectory contains temporal information that can be used to adjust the incorrect poses. In this context, two methods are evaluated for detecting the inaccurate poses; (1) using the traditional method of detecting abnormal pose relative to neighbor poses in trajectory, and (2) using camera view filters detecting inaccurate poses where maximum self-occlusion occurs. The traditional method basically uses the slope of trajectory to find an outlier, while the proposed method is designed to detect incorrect poses by assuming that inaccurate poses occur when a boom and image plane make 90° angle and all the excavator elements are self-occluded. This state of the boom relative to the camera was detected by measuring the horizontal angle between the camera plane and the boom vector. After removing these incorrect poses from the trajectory, the adjusted poses were inserted using spline curve fitting, and finally, the curve was further refined using a moving average. The results showed that applying the use of temporal characteristics of a trajectory improves the overall pose estimation performance. The *t*-test result also validates that the effect of these applied methods is significant (Table 1). However, using the slope of the trajectory can make results worse. For example, in Table 1 (row 3), the pose error of the arm increased after applying the traditional post-processing technique. This issue may happen due to false positive or false negative classification of poses when only the slope of motion trajectories was used. For example, false classification of poses may happen because there are consistent incorrect poses in the neighbor of a correct pose that can be detected falsely. This false classification may also happen because the arm was visible in the camera from the front side for a considerable time, and the estimated pose was wrong consistently having a uniform slope in trajectory. This phenomenon may falsely classify incorrect poses such as a larger error of “proposed loss function + traditional post-processing” than “traditional loss function”.

Furthermore, overall, the CNN model learns to detect key elements of an excavator for a specific excavator model in the natural environment. Meanwhile, this CNN model can particularly work well for similarly shaped excavators. For instance, this model can detect excavator elements of a crawler and a wheel-based excavator with more than 90% accuracy, while for dissimilar excavator models, the CNN model fails to detect a rope of a dragline excavator and its bucket of different shapes with 33.33% accuracy. These results indicate that the accuracy can further be improved by training a CNN model with a dataset generated from various virtual excavator models. In addition, it can be inferred that this approach of estimating 3D pose from 2D key points can be challenging for excavators of the different boom, arm, and bucket ratios because two different arm lengths can look similar depending upon the perspective of the camera view. Additional information, such as boom to arm length ratio can thus be given to the CNN model during training to

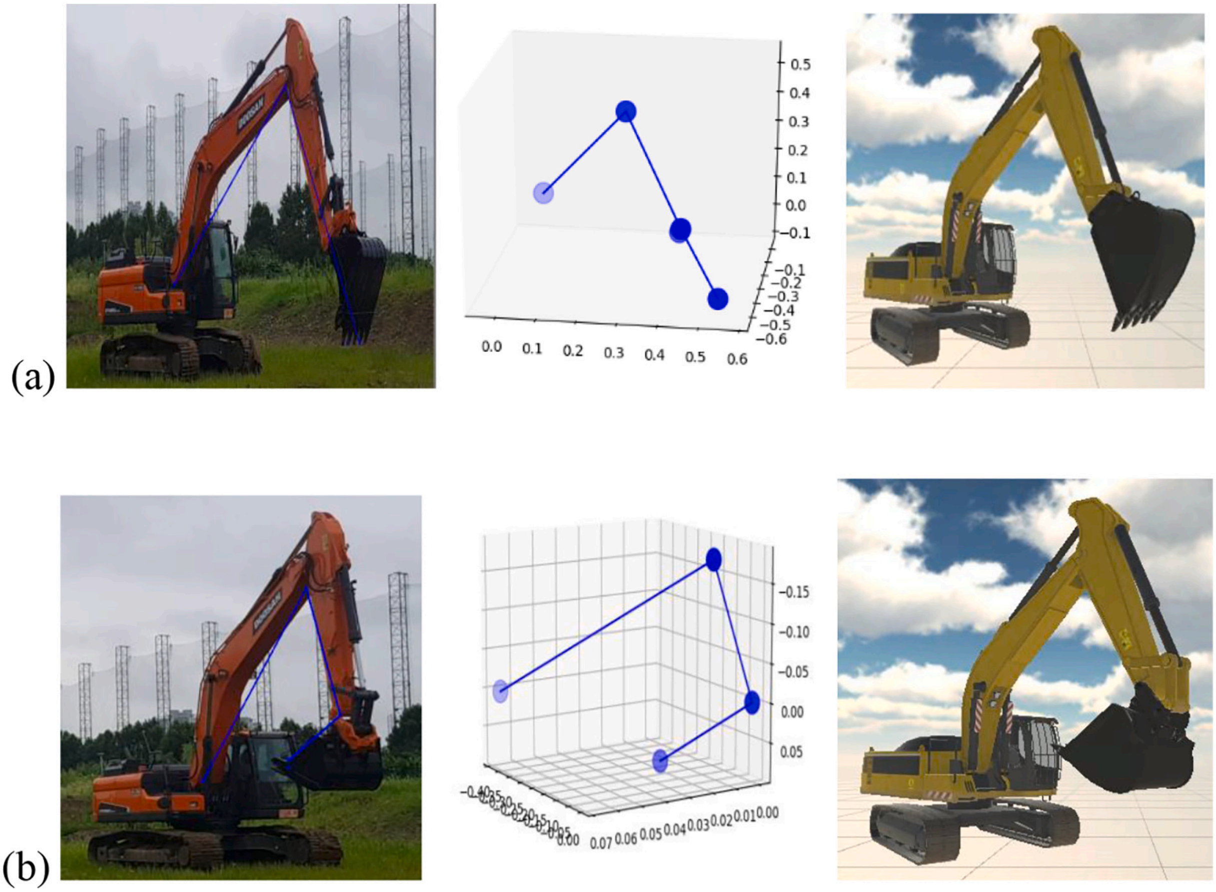

Fig. 20. Visualization of the estimated pose.

deal with different types of machines.

To evaluate the resulting pose estimation errors for the applicability of 3d pose estimation in construction, the angular error is converted to a positional error using the equation of arc length (arc = radius × angle). As a result, the angular pose estimation error of the proposed method is equivalent to 0.5142 m (Table 3). This positional error may be regarded as acceptable to monitor safe interactions based on proximity [54] and activity analysis of an excavator [55]. For example, Wang and Razavi [54] proposed a sensing approach to measuring safe interaction between an excavator and other onsite resources, resulting in a positional error of 0.7 m for detecting unsafe proximity. For activity analysis, ultra-wideband technology has been proposed to track onsite resources with a positional error of less than 1 m in Maalek and Sadeghpour's study [55]. In this regard, the proposed method might potentially be applied for safety and activity analysis. However, in Lundeen et al. [16], 0.25 m positional error in pose estimation was required for avoiding underground utility strikes, which is hard to achieve with the vision-based pose estimation in this study.

Further research efforts are required to address the following potential limitations. First, in the experiment, only a single excavator was used for data collection. As the shape, dimension, and color of an excavator may have an impact on the performance of vision-based methods, the proposed method should be further validated with additional datasets representing the various conditions of an actual site. Second, in this study, the estimated poses resulting from the proposed method were compared with those measured using the motion sensors. However, it can be visually observed that the ground truth data may also involve some degree of error. Nonetheless, this phenomenon demonstrates the challenges in obtaining the accurate pose labels in a field setting, which this study aims to address. Finally, to correct the inaccurately estimated poses, smoothening of the motion trajectory was performed using a moving average with a window length of four frames, which was empirically determined in the experiment. However, this window length can be selected based on the specific working speed of the excavator on site. In addition, spline curve fitting can adjust the trajectory when an excavator part is occluded for a short time; however, if the bucket is inside the ground most of the time and not visible on the camera, trajectory prediction algorithms such as the Gaussian process [56] could be applied and tested for such cases.

Table 3
Conversion of angular pose error to positional pose error.

	Angular error (deg)	Angular error (rad)	Element radius (m)	Positional Error (m)
Boom	7.16	0.1249	5	0.6245
Arm	9.9	0.1727	3	0.5181
Bucket	11.85	0.2	2	0.4
Mean	–	–	–	0.5142

7. Conclusion

This study proposed the use of synthetic images for training a CNN-based 3D pose estimation model, which was tested and applied to actual images from jobsites. In this study, a CNN model, in which a kinematic constraint was adopted for loss computations was trained with synthetic images generated in a virtual environment (e.g., Unity) and tested with field images collected through experiments. The estimated poses, which were further adjusted through trajectory-based post-processing, were then compared with the poses measured using motion capture sensors. From the experimental results, the major contributions of this study can be summarized as follows: (1) the use of synthetic images may provide a solution to prepare a sufficient amount of training datasets for vision-based pose estimation if the real-world visual features can properly be augmented in the image, (2) the loss function of a kinematic constraint (e.g., joints of the boom, arm, and bucket laid on one plane) may allow for correcting the pose errors potentially caused by perspective projection in image generation during the training of a CNN model, and (3) the pose trajectory can be improved by detecting and correcting the incorrect excavator poses commonly resulting from occlusions, which can make the motions of an excavator more realistic. Overall, this research provides an insight into the data that can be used for training, ways to implement a CNN model for pose estimation, and the techniques to better visualize the resulting pose data for the monitoring of an excavator.

The proposed method estimates the 3d poses of an excavator from images, which can be represented as the trajectory of the pose angle. As in previous studies, such 3D poses of an excavator can potentially be used for safety monitoring [13], productivity analysis [15], automated machine guidance [16], and skill assessment of an operator [17]. First, for safety monitoring, hazardous interactions between articulated construction equipment and workers are identified using the 3d pose of the equipment and relative locations of workers [13]. For this identification, the 3d pose of the equipment and locations of the workers are recorded, and this recorded data is linked with virtual models to mimic the jobsite operations in a virtual environment. Then, probable hazardous situations (e.g., stuck-by incidents) can be identified in this virtual environment using 3d pose-based motion parameters (e.g., the heading direction and rotation speed of equipment, the relative location of workers). For example, 3d trajectories of the equipment arm can be estimated with a pose estimating sensor (e.g., inertial measurement units) and the locations of the worker can be estimated using a localization sensor (e.g., ultra-wideband). By linking these spatial parameters with the virtual excavator and virtual workers, animation of jobsite operations can be created for monitoring. In this animation, a probable stuck-by-incident can be identified and prevented by pre-defining and alarming unsafe situations (i.e., scenarios) when the arm of equipment is heading towards a nearby worker. Second, for productivity analysis [15], the 3d pose estimation can be used to measure the cycle time of an excavator, particularly when applied in conjunction with action recognition (e.g., detecting loading, swinging, dumping, and returning of a bucket) [14]. Once the cycle time is measured, the production rate (i.e., m^3/h) of the excavator can be estimated given the quantity (i.e., soil volume) of a bucket. For example, the number of cycles per hour can be computed by dividing one hour by the duration of one cycle, and then the production rate can be calculated by multiplying the number of cycles per hour by the bucket size of an excavator. The proposed method thus may serve as a foundation for continuous monitoring of the excavator's production rate. In addition, safety and productivity issues can further be monitored and analyzed in an indirect way. For machine guidance [16], the animation of trenching operations is visualized to the operator in the cabin while the excavator is working in the blind zone. For creating this animation, firstly, the excavation surface is modeled using remote sensing techniques (e.g., a drone, a laser scanner). Then, 3d poses and locations of the excavator are estimated using IMU sensors and a GPS attached to the excavator body. This spatial data is linked to a virtual excavator placed on the modeled surface, by which an animation is created in the virtual environment. This animation has been displayed to the operator on a computer screen during trenching operations [16]. With the visualized information, the operator could continuously obtain updates on bucket locations (e.g., relative height and distance from a target surface) and could move the bucket to excavate the trench at the right place with desired depth without an external helper's assistance. Also, visualizing the bucket pose relative to an underground utility can help avoid the undesirable hitting of the bucket to underground resources. This visual guidance to the operator can improve productivity from 15% to 30% and this visual guidance can also save cost from 4% to 6% [57]. As another example, the pose information can be used for the skill assessment of an operator by analyzing the trajectory of excavator poses. In Bernold [17], the pose trajectories of an excavator's bucket and sequences of a rotation in a boom, an arm, and a bucket of an excavator have been recorded to measure the skill of an operator. The smoothness of bucket trajectories and rotation sequences of excavator elements (i.e., a boom, an arm, and a bucket) were used as measures of an operator's skill [17]. For example, a smooth trajectory of an excavator bucket with simultaneous rotations of excavator elements represents a skilled worker; this smooth trajectory requires less force to excavate and thus less fuel consumption.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2018R1A5A1025137).

References

[1] C. Yuan, S. Li, H. Cai, Vision-based excavator detection and tracking using hybrid kinematic shapes and key nodes, J. Comput. Civ. Eng. 31 (1) (2017), 04016038, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000602.

[2] L. Bobadilla, A. Mostafavi, T. Carmenate, S. Bista, Predictive assessment and proactive monitoring of struck-by safety hazards in construction sites: an information space approach, Comput. in Civil and Building Eng. 2014 (2014) 989–996, https://doi.org/10.1061/9780784413616.123.

[3] F. Vahdatikhaki, A. Hammad, H. Siddiqui, Optimization-based excavator pose estimation using real-time location systems, Autom. Constr. 56 (2015) 76–92, https://doi.org/10.1016/j.autcon.2015.03.006.

[4] S.N. Naghshbandi, L. Varga, Y. Hu, Technologies for safe and resilient earthmoving operations: a systematic literature review, Autom. Constr. 125 (2021) 103632, https://doi.org/10.1016/j.autcon.2021.103632.

[5] L.E. Bernold, Quantitative assessment of backhoe operator skill, J. Constr. Eng. Manag. 133 (11) (2007) 889–899, https://doi.org/10.1061/(ASCE)0733-9364(2007)133:11(889).

[6] K.-Y. Lin, M.-H. Tsai, U.C. Gatti, J. Je-Chian Lin, C.-H. Lee, S.-C. Kang, A user-centered information and communication technology (ICT) tool to improve safety inspections, Autom. Constr. 48 (2014) 53–63, https://doi.org/10.1016/j.autcon.2014.08.012.

[7] J. Seo, S. Han, S. Lee, H. Kim, Computer vision techniques for construction safety and health monitoring, Adv. Eng. Inform. 29 (2) (2015) 239–251, https://doi.org/10.1016/j.aei.2015.02.001.

[8] M.C. Gouett, C.T. Haas, P.M. Goodrum, C.H. Caldas, Activity analysis for direct-work rate improvement in construction, J. Constr. Eng. Manag. 137 (12) (2011) 1117–1124, https://doi.org/10.1061/(ASCE)CO.1943-7862.0000375.

[9] J. Yang, M.-W. Park, P.A. Vela, M. Golparvar-Fard, Construction performance monitoring via still images, time-lapse photos, and video streams: now, tomorrow, and the future, Adv. Eng. Inform. 29 (2) (2015) 211–224, https://doi.org/10.1016/j.aei.2015.01.011.

[10] R. Navon, R. Sacks, Assessing research issues in automated project performance control (APPC), Autom. Constr. 16 (4) (2007) 474–484, https://doi.org/10.1016/j.autcon.2006.08.001.

[11] H. Guo, Y. Yu, M. Skitmore, Visualization technology-based construction safety management: a review, Autom. Constr. 73 (2017) 135–144, https://doi.org/10.1016/j.autcon.2016.10.004.

[12] T. Cheng, M. Venugopal, J. Teizer, P.A. Vela, Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments, Autom. Constr. 20 (8) (2011) 1173–1184, https://doi.org/10.1016/j.autcon.2011.05.001.

[13] L. Messi, B. Naticchia, A. Carbonari, L. Ridolfi, G.M. Di Giuda, Development of a Digital Twin Model for Real-Time Assessment of Collision Hazards, Creative Construction e-Conference 2020, Budapest University of Technology and Economics, 2020, pp. 14–19, https://doi.org/10.3311/CCC2020-003.

[14] S. Han, S. Lee, A vision-based motion capture and recognition framework for behavior-based safety management, Autom. Constr. 35 (2013) 131–141, https://doi.org/10.1016/j.autcon.2013.05.001.

[15] C. Chen, Z. Zhu, A. Hammad, Automated excavators activity recognition and productivity analysis from construction site surveillance videos, Autom. Constr. 110 (2020) 103045, https://doi.org/10.1016/j.autcon.2019.103045.

[16] K.M. Lundeen, S. Dong, N. Fredricks, M. Akula, J. Seo, V.R. Kamat, Optical marker-based end effector pose estimation for articulated excavators, Autom. Constr. 65 (2016) 51–64, https://doi.org/10.1016/j.autcon.2016.02.003.

[17] L.E. Bernold, Quantitative Assessment of Backhoe Operator Skill 133 (11), 2007, pp. 889–899, https://doi.org/10.1061/(ASCE)0733-9364(2007)133:11(889).

[18] W.A. Tanoli, A. Sharafat, J. Park, J.W. Seo, Damage prevention for underground utilities using machine guidance, Autom. Constr. 107 (2019) 102893, https://doi.org/10.1016/j.autcon.2019.102893.

[19] J. Park, J. Chen, Y.K. Cho, Self-corrective knowledge-based hybrid tracking system using BIM and multimodal sensors, Adv. Eng. Inform. 32 (2017) 126–138, https://doi.org/10.1016/j.aei.2017.02.001.

[20] L. Wang, P.D. Groves, M.K. Ziebart, Shadow matching: a new GNSS positioning technique for urban canyons, NAVIGATION, J. Institute of Navigation 64 (3) (2011) 417–430, https://doi.org/10.1002/navi.38.

[21] M.M. Soltani, Excavator Pose Estimation for Safety Monitoring by Fusing Computer Vision and RTLS Data, Building Engineering, Concordia University, PhD, 2017. https://spectrum.library.concordia.ca/983390/.

[22] Z. Aziz, C. Anumba, D. Ruikar, P. Carrillo, D. Bouchlaghem, Context aware information delivery for on-site construction operations, proceedings of the 22nd CIB-W78 conference on information Technology in Construction, Institute for Construction Informatics, Technische Universitat Dresden, Germany, CBI Publication 304 (2005) 321–332. http://irep.ntu.ac.uk/id/eprint/8790.

[23] C.-J. Liang, V.R. Kamat, C.M. Menassa, Real-time construction site layout and equipment monitoring, Construction Research Congress 2018 (2018) 64–74, https://doi.org/10.1061/9780784481264.007.

[24] E.R. Azar, C. Feng, V.R. Kamat, Feasibility of in-plane articulation monitoring of excavator arm using planar marker tracking, J. Inform. Technol. Construction 20 (15) (2015) 213–229. https://www.itcon.org/2015/15.

[25] C.J. Liang, K.M. Lundeen, W. McGee, C.C. Menassa, S. Lee, V.R. Kamat, A vision-based marker-less pose estimation system for articulated construction robots, Autom. Constr. 104 (2019) 80–94, https://doi.org/10.1016/j.autcon.2019.04.004.

[26] J. Seo, R. Starbuck, S. Han, S. Lee, T.J. Armstrong, Motion data-driven biomechanical analysis during construction tasks on sites, J. Comput. Civ. Eng. 29 (4) (2015), B4014005, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000400.

[27] S. Han, S. Lee, F. Peña-Mora, Comparative study of motion features for similarity-based modeling and classification of unsafe actions in construction, J. Comput. Civ. Eng. 28 (5) (2014), A4014005, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000339.

[28] S. Han, S. Lee, F. Peña-Mora, Vision-based detection of unsafe actions of a construction worker: case study of ladder climbing, J. Comput. Civ. Eng. 27 (6) (2013) 635–644, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000279.

[29] M.M. Soltani, Z. Zhu, A. Hammad, Framework for location data fusion and pose estimation of excavators using stereo vision, J. Comput. Civ. Eng. 32 (6) (2018), 04018045, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000783.

[30] X. Zhou, Q. Huang, X. Sun, X. Xue, Y. Wei, Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach, 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 398–407, https://doi.org/10.1109/ICCV.2017.51.

[31] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, CVPR (2011) 1297–1304, https://doi.org/10.1007/978-3-642-28661-2_5. Ieee.

[32] V. Lepetit, P. Lagger, P. Fua, Randomized trees for real-time keypoint recognition, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) Vol. 2, 2005, pp. 775–781, https://doi.org/10.1109/CVPR.2005.288.

[33] H. Tang, Q. Wang, H. Chen, Research on 3D Human Pose Estimation Using RGBD Camera, in: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 538–541, https://doi.org/10.1109/ICEIEC.2019.8784591.

[34] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, M. Shah, Deep Learning-Based Human Pose Estimation: A Survey, Tsinghua Sci. Technol. 24 (6) (2020), https://doi.org/10.26599/TST.2018.9010100.

[35] S. Li, A.B. Chan, 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network, Springer, Asian Conference on Computer Vision, 2014, pp. 332–347, https://doi.org/10.1007/978-3-319-16808-1_23.

[36] S. Li, W. Zhang, A.B. Chan, Maximum-margin structured learning with deep networks for 3d human pose estimation, in: Int. J. Comput. Vis., Springer, 2017, pp. 149–168, https://doi.org/10.1007/s11263-016-0962-x.

[37] G. Pavlakos, X. Zhou, K. Daniilidis, Ordinal depth supervision for 3D human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7307–7316, https://doi.org/10.1109/CVPR.2018.00763.

[38] J. Martinez, R. Hossain, J. Romero, J.J. Little, A simple yet effective baseline for 3d human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2640–2649, https://doi.org/10.1109/ICCV.2017.288.

[39] B. Tekin, P. Márquez-Neila, M. Salzmann, P. Fua, Learning to fuse 2d and 3D image cues for monocular body pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3941–3950, https://doi.org/10.1109/ICCV.2017.425.

[40] F. Moreno-Noguer, 3D human pose estimation from a single image via distance matrix regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2823–2832, https://doi.org/10.1109/CVPR.2017.170.

[41] X. Zhou, X. Sun, W. Zhang, S. Liang, Y. Wei, Deep Kinematic Pose Regression, Springer, European Conference on Computer Vision, 2016, pp. 186–201, https://doi.org/10.1007/978-3-319-49409-8_17.

[42] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, W. Zhang, Deep kinematics analysis for monocular 3D human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 899–908, https://doi.org/10.1109/CVPR42600.2020.00098.

[43] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments, in: IEEE transactions on pattern analysis and machine intelligence 36 (7), 2014, pp. 1325–1339, https://doi.org/10.1109/TPAMI.2013.248.

[44] W.T. Calderon, D. Roberts, M. Golparvar-Fard, Synthesizing pose sequences from 3D assets for vision-based activity analysis, J. Comput. Civ. Eng. 35 (1) (2021), 04020052, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000937.

[45] M.M. Soltani, Z. Zhu, A. Hammad, Skeleton estimation of excavator by detecting its parts, Autom. Constr. 82 (2017) 1–15, https://doi.org/10.1109/TPAMI.2013.248.

[46] B.P. Patel, J. Prajapati, Kinematics of mini hydraulic backhoe excavator part II, Int. J. Mech. Robotic Systems 1 (4) (2013), https://doi.org/10.1504/IJMRS.2013.057301.

[47] J. Schmittler, D. Pohl, T. Dahmen, C. Vogelgesang, P. Slusallek, Realtime ray tracing for current and future games, ACM SIGGRAPH 2005 Courses, Association for Computing Machinery, 2005, pp. 23–es, https://doi.org/10.1145/1198555.1198762.

[48] T. Tan, G.D. Sullivan, K.D. Baker, On Computing The Perspective Transformation Matrix and Camera Parameters, British Machine Vision Conference, 1993, pp. 1–10, https://doi.org/10.5244/C.7.13.

[49] Z. Zhang, Weak perspective projection, in: K. Ikeuchi (Ed.), Computer Vision: A Reference Guide, Springer US, Boston, MA, 2014, pp. 877–883, https://doi.org/10.1007/978-0-387-31439-6_115.

[50] N. Klug, M. Einfalt, S. Brehm, R. Lienhart, Error Bounds of Projection Models in Weakly Supervised 3D Human Pose Estimation, International Conference on 3D Vision, 2020, pp. 898–907, https://doi.org/10.1109/3DV50981.2020.00100.

[51] G. Pons-Moll, B. Rosenhahn, Model-based pose estimation, Visual analysis of humans, Springer (2011), https://doi.org/10.1007/978-0-85729-997-0, 139-170, 978-1-4471-5914-8.

[52] T. Lyche, K.M. Mørken, Knot removal for parametric B-spline curves and surfaces, Computer Aided Geometric Design 4 (3) (1987) 217–230, https://doi.org/10.1016/0167-8396(87)90013-6.

[53] Q. Dang, J. Yin, B. Wang, W. Zheng, Deep learning based 2D human pose estimation: A survey, Tsinghua Sci. Technol. 24 (6) (2019) 663–676, https://doi.org/10.26599/TST.2018.9010100.

[54] J. Wang, S.N. Razavi, Low False Alarm Rate Model for Unsafe-Proximity Detection in Construction 30 (2), 2016, p. 04015005, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000470.

[55] R. Maalek, F. Sadeghpour, Accuracy assessment of ultra-wide band technology in tracking static resources in indoor construction scenarios, Autom. Constr. 30 (2013) 170–183, https://doi.org/10.1016/j.autcon.2012.10.005.

[56] R.J. Sandzimier, H.H. Asada, A data-driven approach to prediction and optimal bucket-filling control for autonomous excavators, IEEE Robotics Automation Letters 5 (2) (2020) 2682–2689, https://doi.org/10.1109/LRA.2020.2969944.

[57] A. Hammad, F. Vahdatikhaki, C. Zhang, A novel integrated approach to project-level automated machine control/guidance systems in construction projects, J. Inform. Technol. Construction 18 (9) (2013) 162–181. https://www.itcon.org/paper/2013/9.