Automation in Construction 99 (2019) 168–182

Contents lists available at ScienceDirect

Automation in Construction

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

Remote proximity monitoring between mobile construction resources using camera-mounted UAVs

Daeho Kim, Meiyin Liu, SangHyun Lee*, Vineet R. Kamat

Department of Civil and Environmental Engineering, Univ. of Michigan, MI 48109, United States of America

ARTICLE INFO

Keywords:
Struck-by accident
Proximity monitoring
Unmanned aerial vehicle
Deep neural network
Computer vision

ABSTRACT

Struck-by accidents have resulted in a significant number of fatal and nonfatal injuries in the construction industry. As a proactive safety measure against struck-by hazards, the authors present an Unmanned Aerial Vehicle (UAV)-assisted visual monitoring method that can automatically measure proximities among construction entities. To attain this end, this research conducts two research thrusts: (i) object localization using a deep neural network, YOLO-V3; and (ii) development of an image rectification method that allows for the measurement of actual distance from a 2D image collected from a UAV. Tests on real-site aerial videos show the promising accuracy of the proposed method; the mean absolute distance errors for estimated proximity were less than 0.9 m and the mean absolute percentage errors were around 4%. The proposed method enables the advanced detection of struck-by hazards around workers, which in turn can make timely intervention possible. This proactive intervention can ultimately promote a safer working environment for construction workers.

1. Introduction

Struck-by accidents involving a mobile vehicle or heavy equipment has been one of the leading causes of fatal and nonfatal injuries of construction workers. From 2011 to 2015, this forcible impact contributed to 925 construction-related fatalities, which accounted for more than 18% of the overall occupational deaths in the U.S. construction industry [1,2]. Notably, the number of struck-by fatalities rose 34% from 2010 (N = 121) to 2015 (N = 162). [1,2]. During this period, struck-by fatalities in construction were unmatched by other industries, and the incident rate for non-fatal struck-by injuries (i.e., 27.4 injuries per 10,000 full-time equivalent workers) runs up to nearly twice the rate of all other industries combined [1,2].

A major research area for the prevention of the struck-by accident is attuned to the development of automation technology for onsite proximity monitoring among construction entities. Monitoring proximity between workers and equipment (or vehicles) enables the advanced detection of potential hazards, which allows for prompt feedback (e.g., visible, acoustic, and vibration alarm) to involved workers [3–9]. This proactive intervention can lead the workers to prepare for evasive actions, thereby reducing the chance of an impending collision [3,6–9].

Previous research on proximity monitoring has been dominated by a wide range of wireless sensors, including Radio Frequency Identification (RFID) [3,4], Magnetic Field (MF) [5], Global Positioning System (GPS) [7], and Bluetooth Low Energy (BLE) [8,9]. However, the implementation of this sensor-based application may be challenged in practice. For example, the prerequisite that all entities should be attached with sensors could be burdensome for both contractors and workers. It could be costly in the project where tremendous volumes of personnel, equipment, and materials are involved [10–13], and further be intrusive to workers who do not want to be proposedly tagged [11,14]. Besides, the sensing range could be affected by various factors—such as ambient condition or approach angle and speed—as the sensors operate based on wave signal propagation [5,8,9].

Against this backdrop, this research aims to develop an Unmanned Aerial Vehicle (UAV)-assisted visual monitoring method as an alternative technology for the onsite proximity monitoring. An ordinary camera mounted-UAV can capture moving entities continuously while accessing hard-to-reach areas [15]. This mobility enables the monitoring of wide areas, which is not viable with conventional imaging devices such as surveillance or portable cameras [16–27]. Further, computer vision can recognize multiple entities without installing any sensors [11–14,28,30]. Accordingly, this visual monitoring has the potential to lead cost-effective and non-invasive proximity monitoring while complementing existing sensing technologies.

To achieve this aim, this research focuses on addressing technical challenges facing computer vision techniques, i.e., object localization

* Corresponding author.
E-mail address: shdpm@umich.edu (S. Lee).

https://doi.org/10.1016/j.autcon.2018.12.014
Received 28 April 2018; Received in revised form 6 November 2018; Accepted 13 December 2018
Available online 21 December 2018

and distance measurement, which are fundamental for visual proximity monitoring. Specifically, real-site videos entail uncertain variations. For example, each frame involves different viewpoints, scene scales, and illumination. Also, each entity (i.e., workers, equipment, or vehicle) has an individually distinctive appearance. This variation would impose restrictions on the localization capability of hand-crafted algorithms since they operate as designed and thus could not be adaptive to such variations [11–14]. In addition, measuring distance on a 2D image is extremely challenging due to the lack of depth information (i.e., the 3rd coordinate of a point). It therefore needs the post-processing for 3D reconstruction, which requires a significant amount of computational cost.

To overcome these challenges, this research conducts two research thrusts: (i) the application of a deep neural netowork (i.e., YOLO-V3 [44]) for robust localization; and (ii) the development of an image rectification method that enables the measurement of actual distance on a 2D image without 3D reconstruction. This research tests the proposed method on real-site aerial videos so as to evaluate its monitoring performance in real scene settings. Finally, the result and its implication are discussed, and future research directions toward real-world application are provided.

2. Reviews on existing computer vision techniques

The visual proximity monitoring requires two vision processings: (i) object localization; and (ii) distance measurement. Object localization finds target-centered locations on a pixel-coordinate system (e.g., x-y coordinates of a worker). Based on this spatial information, the proximity (i.e., Euclidean distance) between workers and equipment (or vehicles) can be measured. This section reviews existing computer vision techniques for object localization and distance measurement and discusses their pros and cons.

2.1. Object localization

A number of studies have investigated the computer vision techniques to localize construction entities (e.g., workers, equipment, and materials) on video frames that can be classified into one of two categories: (i) object tracking; or (ii) object detection. The object tracking operates based on a tracker that interprets features (e.g., motion, color, and shape) of a given target and tracks the encoded information on successive frames. Yang et al. [28] proposed a framework using kernel principal component analysis and kernel covariance tracking to track multiple construction workers. On the other hand, several studies conducted comparative analyses with existing tracking algorithms to investigate which one is appropriate for complex construction environment. For instance, Teizer and Vela [29] investigates four distinctive algorithms (i.e., mean-shift, Bayesian segmentation, active contours, and graph-cut) and revealed that Bayesian segmentation performed the best in tracking a construction worker. Park et al. [30] divided existing tracking algorithms into point, kernel and contour-based methods and compared their performances under different conditions of illumination and occlusion. This study concluded that kernel-based tracking is the most appropriate for construction entities, but the test results also indicated that there is no such a method that always outperforms the others [30]. The tracking methods allow us to exploit temporal information in successive frames, which can accelerate the overall localization process. However, the tracking methods cannot independently perform object localization as the location of a target should be marked on the first frame so that a tracker can be initialized [31,32].

On the other hand, detection methods localize targets frame-by-frame independently. Accordingly, it doesn't require the location initialization. Park et al. [11] proposed a detection framework to recognize construction workers by leveraging various features such as motion (by background subtraction), shape (i.e., Histogram of Gradient (HOG)), and color (i.e., color histogram). Whereas Memarzadeh et al. [12] developed a novel descriptor by combining HOG and the histogram of Hue-Saturation-Value (HSV), and trained support vector machines (SVMs) to detect multi-class objects. These studies contributed to the automation of the localization process. However, the detection algorithm would require a larger amount of computational cost than the tracking due to the exhaustive searching mechanism that investigates every possible location in an image [31,32].

The previous localization methods were mainly based on the hand-crafted features—such as HOG, HSV, Scale Invariant Feature Transform (SIFT), and Speeded-Up Robust Features (SURF). Although the hand-crafted features could work well in a customized imaging condition (e.g., controlled viewpoint, scale, and illumination), they could lose their representative power for a target in unconformable conditions—such as viewpoint variation, scale variation, illumination variation, background clutter, or intra-class variation [12,14,30–32]. For example, the object detector that uses HOG could fail to detect same object if a huge illumination difference occurs, while the one that uses SIFT could fail to detect equipment with a distinctive appearance. Therefore, the higher-level of representation of a target is required in localizing construction entities on a UAV-captured video where the dynamic viewpoint gets to amplify such variations.

2.2. Distance measurement

Proximity monitoring is completed by measuring the straight-line distances among targets, which can be straightforward given 3D spatial information. However, using a 3D sensing device (e.g., stereo-vision camera, RGB-D sensor, or Flash LADAR) may not be much viable for onsite operation due to its limited sensing range and the vulnerability to outdoor conditions [33,34]. For example, stereo-vision camera (e.g., Bumblebee XB3, Point Grey Research, Inc.) is restricted to low resolutions and requires a significant amount of computational cost [33]. Also, RGB-D sensor (e.g., MS Kinect™) and Flash LADAR are susceptible to sunlight as well as have restricted measuring range (i.e., 5 m and 10 m, respectively) [33,34].

In construction, there have been few studies attempting to monitor the proximity among construction entities using 2D computer vision [13,35]. These studies estimated proximity by measuring pixel distances among detected objects and used the value in evaluating workers' safety level. Although the pixel distance can be useful in determining relative safety level, it would not be able to represent the real scale of distance due to the lack of depth (i.e., scene scale) and the projective distortion. To be more specific, an ordinary camera maps 3D real space onto a 2D image plane through its monocular lens by perspective projection. During this compressive process, the depth information is lost, and projective distortion occurs, making the original properties of a scene (e.g., length, area, length ratio and area ratio, angle, and parallelism) and proximity distorted.

On this problem, several studies presented post-processing as a solution to recover the depth information (i.e., the lost 3rd coordinate). Brilakis et al. [14] proposed a triangulation framework using multiple 2D cameras to determine the 3D coordinates of construction resources whereas Yang et al. [36] attempted another triangulation algorithm, i.e., Structure from Motion (SFM), for 3D reconstruction.

Although the recovered depth information enables distance measurement, such epipolar geometry-based post-processing requires a significant amount of computational cost for extracting features, calculating fundamental matrix, and lastly triangulation [33]. Moreover, this triangulation is viable only if the camera's extrinsic parameters (i.e., location and orientation) and feature matching are given at a very precise level. Hence, this 3D reconstruction technique may not be the best choice for onsite proximity monitoring, specifically in the context of a mobile UAV.

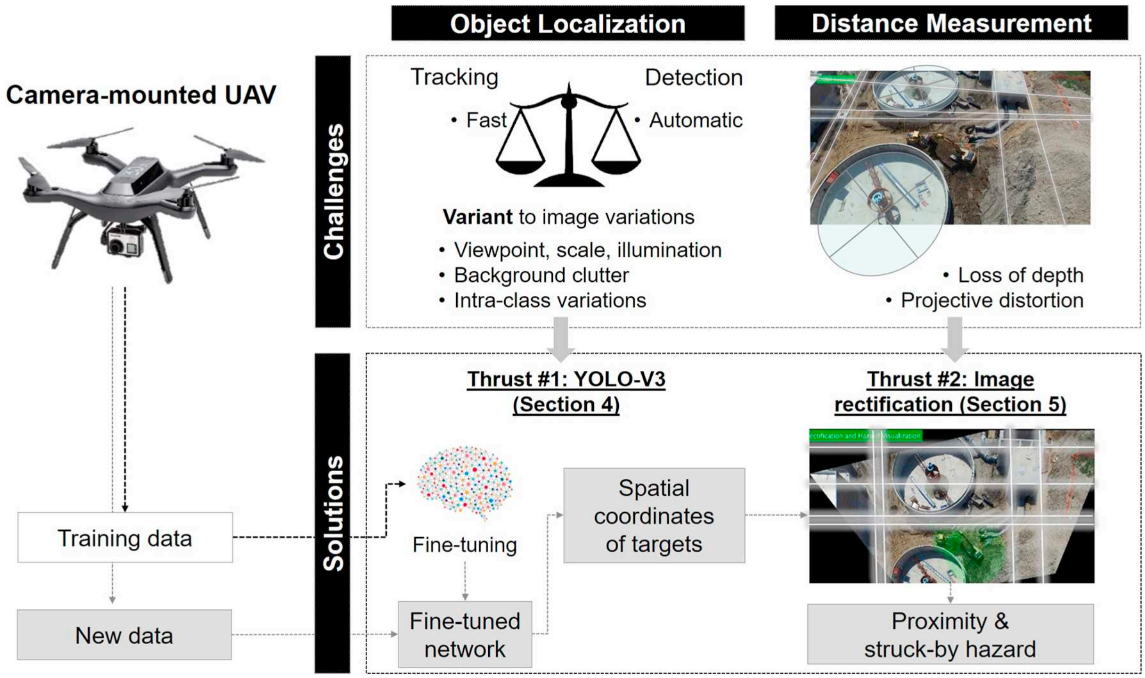

Fig. 1. Two research thrusts for UAV-assisted visual proximity monitoring.

3. Research objectives

The methods to date have shown a potential of visual localization and distance measurement on an image, but have not yet reached the capability to be used for onsite proximity monitoring. The localization techniques may not be sufficiently robust against casual variations of real scenes. In addition, the 3D reconstruction would not be a viable option for proximity monitoring on account of its massive computations and sensitivity to given parameters (e.g., camera's location and orientation).

With these challenges, the objective of this research is to achieve (i) automated, fast, and robust localization of construction entities, and (ii) cost-effective but reliable distance measurement directly from a 2D image. Toward these ends, this research conducts two research thrusts: (i) the application of a deep neural network, i.e., YOLO-V3 [44] to object localization; and (ii) the development of an image rectification method that allows of measuring actual distance on a 2D image without the 3D inference (Fig. 1). In the following two sections, the details on the proposed methods are explained with the test result. In succession, tests on aerial construction site videos and discussions on the test result will follow.

4. Thrust #1: YOLO-V3 for object localization

Recently, deep neural networks (DNNs) have demonstrated superior performance in object detection, overcoming the detection challenges across the computer vision community—such as COCO detection challenges (Table 1) [44]. The deep networks enable the extraction of fine-grained features, which have demonstrated a more robust operation in the object detection [37–44]. At the same time, the DNNs have substantially reduced their computational costs as well with the advancement of computing mechanism (e.g., parallel computing) and hardware [e.g., graphical processing unit (GPU)] [40,44]. Table 1 shows state of the art DNNs for object detection and their performances [i.e., mean average precision (mAP) and frame per second (FPS)] on the COCO benchmark dataset [44].

In construction, there have been several efforts to use the DNNs for the localization of construction entities. For example, Fang et al. [41] attempted to detect non-hardhat-use using Faster R-CNN; Kim et al. [42] applied region-based fully convolutional networks (R-FCN) for detecting equipment in tunnel construction; on the other hand, Kolar et al. [43] designed a customized DNN by combining a VGG-16 (i.e., feature extractor used in Faster R-CNN) and a multi-layers perception (MLP) network for safety guardrail detection. Evidently, these studies showed the successful introduction of the DNNs to construction research, validating its detection performances (e.g., mAP) on construction data. For this study, however, the region proposal network (RPN)-based DNNs—such as Faster R-CNN or R-FCN—would not be the best option due to their high computational cost. As shown in Table 1, the FPS for the Faster R-CNN (i.e., 17 FPS) and R-FCN (i.e., 12 FPS) are insufficient for real-time operation (i.e., 30 FPS).

Table 1
State of the art DNNs for object detection: performance on COCO dataset (provided by [44])

Model	Train dataset	Test dataset	mAP	FPS
Faster R-CNN	COCO train-val	COCO test-dev	42.70%	17
SSD321	COCO train-val	COCO test-dev	45.40%	16
DSSD321	COCO train-val	COCO test-dev	46.10%	12
R-FCN	COCO train-val	COCO test-dev	51.90%	12
Retinanet-50-500	COCO train-val	COCO test-dev	50.90%	14
YOLO-V2	COCO train-val	COCO test-dev	48.10%	40
YOLO-V3	COCO train-val	COCO test-dev	55.30%	35

In this sense, this study applies YOLO-V3 [44] that allows for real-time operation (i.e., 35 FPS) as well as state of the art detection performance (i.e., 55.3 mAP on COCO dataset, Table 1). The YOLO-V3 doesn't require an additional step for region proposal. Instead, it realizes the convolutional implementation of sliding window during its operation, thereby making one-stage detection and real-time operation possible [44]. With this advantage, YOLO-V3 could afford to have a deeper convolutional network and thus achieve the state of the art performance on object detection.

The published YOLO-V3 network, pre-trained with ImageNet, is not learned from the construction contexts such as construction equipment, workers, and backgrounds. Furthermore, this network will not be compatible with UAV-captured images because they are not experienced with aerial viewpoints. For example, a human in a UAV-captured

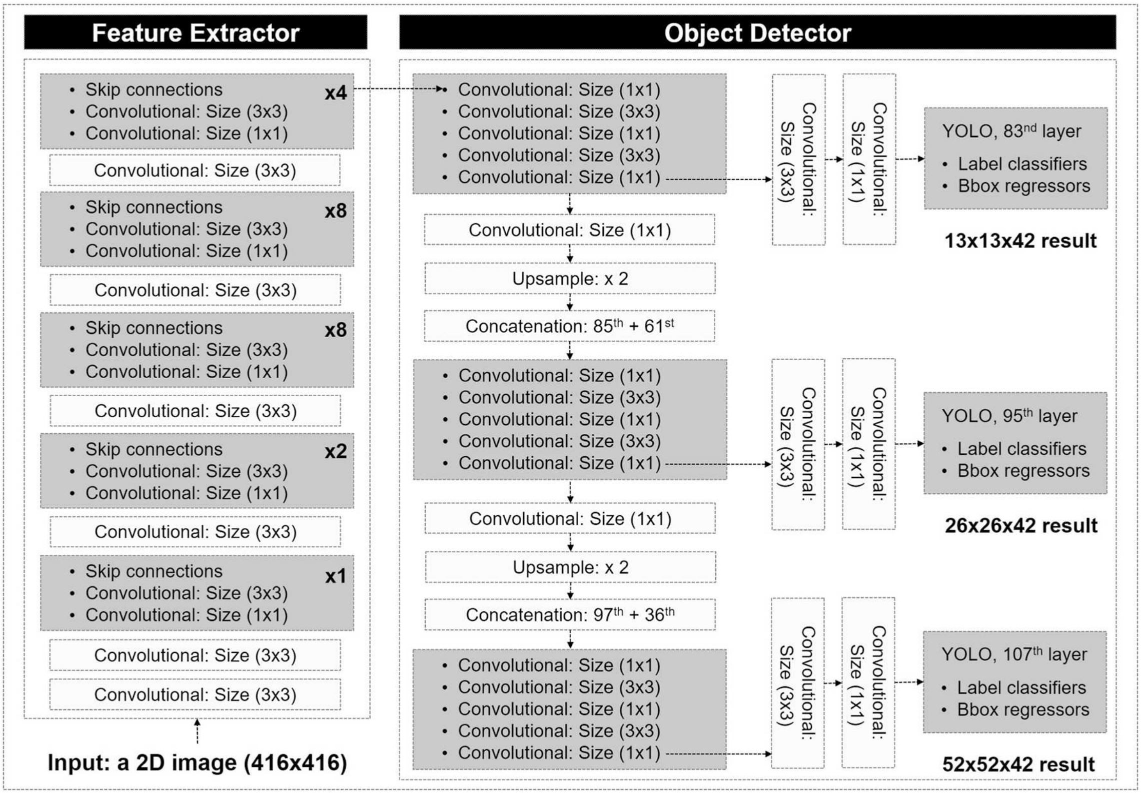
Feature Extractor
Skip connections
Convolutional: Size (3x3)
Convolutional: Size (1x1)
x4
Convolutional: Size (3x3)
Skip connections
Convolutional: Size (3x3)
Convolutional: Size (1x1)
x8
Convolutional: Size (3x3)
Skip connections
Convolutional: Size (3x3)
Convolutional: Size (1x1)
x8
Convolutional: Size (3x3)
Skip connections
Convolutional: Size (3x3)
Convolutional: Size (1x1)
x2
Convolutional: Size (3x3)
Skip connections
Convolutional: Size (3x3)
Convolutional: Size (1x1)
x1
Convolutional: Size (3x3)
Convolutional: Size (3x3)
Input: a 2D image (416x416)
Object Detector
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
YOLO, 83nd layer
Label classifiers
Bbox regressors
13x13x42 result
Convolutional: Size (1x1)
Upsample: x 2
Concatenation: 85th + 61st
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
YOLO, 95th layer
Label classifiers
Bbox regressors
26x26x42 result
Convolutional: Size (1x1)
Upsample: x 2
Concatenation: 97th + 36th
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
Convolutional: Size (3x3)
Convolutional: Size (1x1)
YOLO, 107th layer
Label classifiers
Bbox regressors
52x52x42 result

Fig. 2. Architecture of YOLO-V3.

image has a completely different appearance and scale than one in ImageNet, which must puzzle the convolutional layers and deteriorate the localization performance in the end. On the other hand, to train the network with construction data from scratch must involve a significant risk of overfitting due to the imbalance between the network capacity and the amount of training data. Therefore, this research elects transfer learning to avoid the potential of overfitting as well as to fine-tune the published network to construction settings successfully.

4.1. Network description

The YOLO-V3 consists of two main networks: (i) feature extractor and (ii) object detector (Fig. 2).

- Feature extractor (from 1st to 75th layer): the first network, called darknet-53, takes a resized image (416 × 416 × 3) as an input and outputs a 3D feature tensor (13 × 13 × 1024). The darknet-53 has a deep architecture with successive 52 convolutional layers (i.e., 1 × 1 or 3 × 3), which can extract fine-grained features from a coarse data. In particular, this network incorporates residual skip connections in the intervals of two convolutional layers (i.e., total 23 shortcut layers). The connection initially devised for a residual network helps the darknet-53 to deals with the vanishing gradient problem occurring while training by residually propagating previous features into forward.
- Object detector (from 76th to 107th layer): the second network takes the 3D feature tensor (13 × 13 × 1024) and makes detection. The uniqueness of this network resides in its ability to achieve detection at three different scales, thereby improving scale invariance. This network gradually widens the feature tensor from 13 × 13 to 26 × 26, and 52 × 52 through upsampling and concatenation layers. Meanwhile, three branches come out and each makes a final feature tensor at the different scale (i.e., 13 × 13 × 42, 26 × 26 × 42 and 52 × 52 × 42 at 82nd, 94th, and 106th layer, respectively). Each final feature tensor is then fed into YOLO layer that classifies object label with class-wise logistic regressions and localizes objects with bounding box regressors.

4.2. Test result

The total of 4512 frames capturing construction workers and equipment were extracted from construction site videos and labeled as shown in Fig. 3. Of these, 4114 images were used for the fine-tuning and the other data, 398 consecutive images (i.e., a section of a UAV video), were used for testing. This test considered the three types of object classes: (i) construction worker; (ii) wheel loader; and (iii) excavator (Fig. 3).

The first role of the YOLO-V3 in proximity monitoring is to make correct object detections. To test the detection performance of the fine-tuned network, this test uses mean average precision (mAP, Eq. (1)) and average intersection over union (IoU, Eq. (2)), which are the typical evaluation metrics used for detection challenges—such as PASCAL VOC and COCO. As shown in Table 2, the tuned network could reach to acceptable mAP and average IoU: (i) mAP = 90.82% and (ii) average IoU = 80.97% in this test.

$$\text{mAP} = \frac{1}{n} * \sum_{1}^{n} \left(\frac{1}{11} * \sum_{r=0.0,0.1,\ldots,1.0} AP_r \right) \tag{1}$$

Note: n stands for the total number of object classes; AP_r stands for maximum precision at a certain recall value r (i.e., 0, 0.1, 0.2, …, 1.0).

$$\text{Average IoU} = \frac{1}{k} * \sum_{1}^{k} \left(\frac{AoO}{AoU} \right) \tag{2}$$

Note: k stands for the total number of detected objects; AoO stands for area of overlap; AoU stands for area of union.

In proximity monitoring, it is also critical to find the correct location for detected objects (i.e., object-centered coordinates). Hence, this test further evaluates the fine-tuned network by the average localization

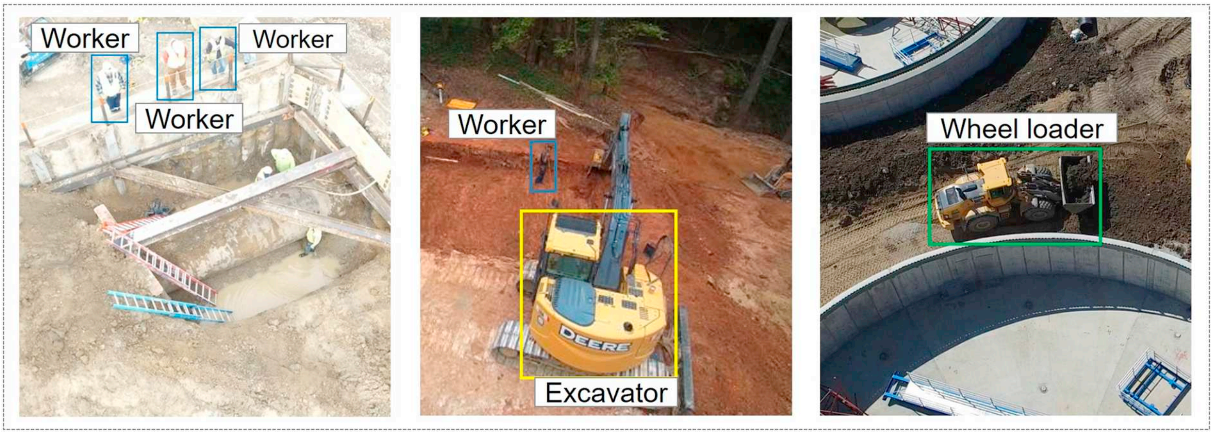

Fig. 3. Examples of training dataset and labels.

Table 2
Result of object detection by YOLO-V3: mAP and average IoU.

# Iter.	Average precision			mAP	Average IoU	Average precision
	Excavator	Worker	Wheel loader			Reference object
500	14.01%	0.00%	17.79%	10.60%	0.00%	0.17%
600	27.04%	11.86%	42.62%	27.17%	38.83%	67.98%
700	56.97%	63.86%	62.87%	61.23%	38.77%	57.37%
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1000	83.48%	80.48%	85.77%	83.24%	60.99%	89.43%
1100	90.71%	90.57%	82.65%	87.98%	69.00%	90.36%
1200	89.05%	86.98%	79.05%	85.03%	63.72%	87.04%
.	.	.	.	.	.	.
.	.	.	.	.	.	.
10000	90.84%	90.63%	90.79%	90.75%	77.16%	90.91%
10100	90.77%	90.73%	90.82%	90.77%	78.18%	90.86%
10200	90.79%	90.73%	90.79%	90.77%	78.65%	90.84%
.						
	.	.	.	.	.	.
19800	**90.77%**	**90.86%**	**90.84%**	**90.82%**	**80.97%**	**90.86%**
19900	90.77%	90.76%	90.84%	90.79%	80.36%	90.86%
20000	90.75%	90.84%	90.82%	90.80%	78.82%	90.84%

Note: mAP and average IoU are for excavator, worker, and wheel loader; reference object is the material to be used for image rectification whose role and function will be detailed in the next section.

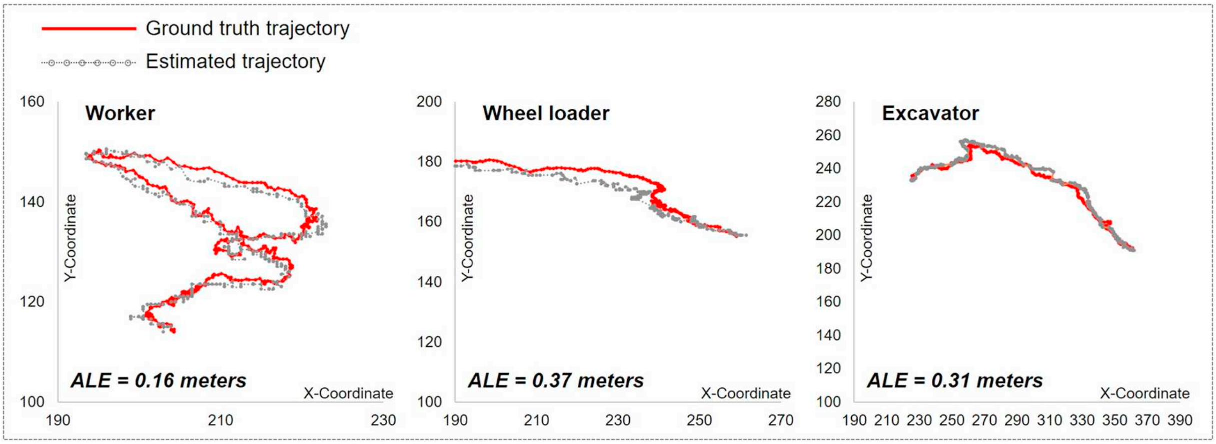

Fig. 4. Result of object localization by YOLO-V3: trajectories.

error (ALE) (i.e., the average of the Euclidean distance between ground truth position and estimated position, Eq. (3)). As shown in Fig. 4, the fine-tuned network showed promising localization performance, tracking ground truth consistently with the acceptable ALE: (i) worker = 0.16 m; (ii) wheel loader = 0.37 m; and (iii) excavator = 0.31 m (Table 3).

Table 3
Result of object localization by YOLO-V3: ALE.

Frame #	Estimated coordinates						Localization error (unit: meters)		
	Worker		Wheel loader		Excavator		Worker	wheel Loader	Excavator
	X	Y	X	Y	X	Y			
1	203	114	262	156	362	191	0.14	0.32	0.37
2	203	115	260	156	361	191	0.16	0.15	0.27
3	203	115	261	156	361	191	0.21	0.24	0.33
4	204	115	260	156	361	191	0.17	0.17	0.17
5	204	115	260	156	361	191	0.12	0.27	0.18
6	203	115	260	156	361	192	0.14	0.35	0.06
7	204	115	259	156	360	192	0.04	0.22	0.14
8	203	115	259	156	360	192	0.09	0.21	0.18
9	203	116	259	156	359	192	0.03	0.18	0.13
10	203	116	258	156	359	192	0.08	0.17	0.17
					⋮				
389	211	130	92	126	228	234	0.19	0.45	0.26
390	211	130	92	126	227	234	0.15	0.43	0.24
391	211	131	91	126	228	234	0.16	0.44	0.25
392	212	130	90	126	227	233	0.19	0.34	0.27
393	212	130	90	124	226	234	0.11	0.29	0.13
394	211	130	90	124	226	233	0.04	0.25	0.21
395	212	130	90	123	226	233	0.09	0.26	0.09
396	212	130	89	123	226	233	0.08	0.23	0.02
397	212	129	89	123	226	233	0.18	0.21	0.08
398	213	129	88	122	225	233	0.14	0.11	0.13
Average localization error (ALE, unit: meters)							0.16	0.37	0.31

$$\mathrm{ALE} = \frac{1}{n} * \sum_{i=1}^{n} SF\sqrt{(x_{gt} - x_e)^2 + (y_{gt} - y_e)^2} \tag{3}$$

Note: n stands for the total number of frame; *SF* stands for the scale coefficient that converts pixel distance to the metric unit (i.e., meter); x_{gt} and y_{gt} stand for coordinates of ground truth; and x_e and y_e stands for the estimated coordinates.

5. Thrust #2: image rectification for distance measurement

While a camera maps 3D space onto a 2D image plane, projective distortion emerges, distorting original properties of a scene. Fig. 5 provides a detailed example of the projective distortion. In the left-side image, the two ellipses are actually circles having same properties (i.e., diameter = 27.4 m), and also the tetragonal object is a square (i.e., width = height = 2.89 m). As such, measured proximity on a 2D image must be distorted and unreliable. While previous studies have focused on recovering depth information on a 2D image, this research approaches this problem by focusing on the removal of this projective distortion. The key insight is that the 3D distance between two objects placed on the same plane can be measured even with a 2D image if the projective distortion can be successfully rectified (Fig. 5). That is, instead of measuring the depth of points, this research homogenizes the 3rd coordinates of points, thereby making distance measuring possible on a 2D image with a minimum computation. Along this way, this method leverages a reference object whose dimension is already known (e.g., a column foundation). This reference provides a geometric cue to estimate the homography between a distorted and rectified image as well as allows measuring the unique scene scale. After the rectification, the proximity can be measured in a metric unit, and the struck-by hazard can be visualized considering the unique scene scale.

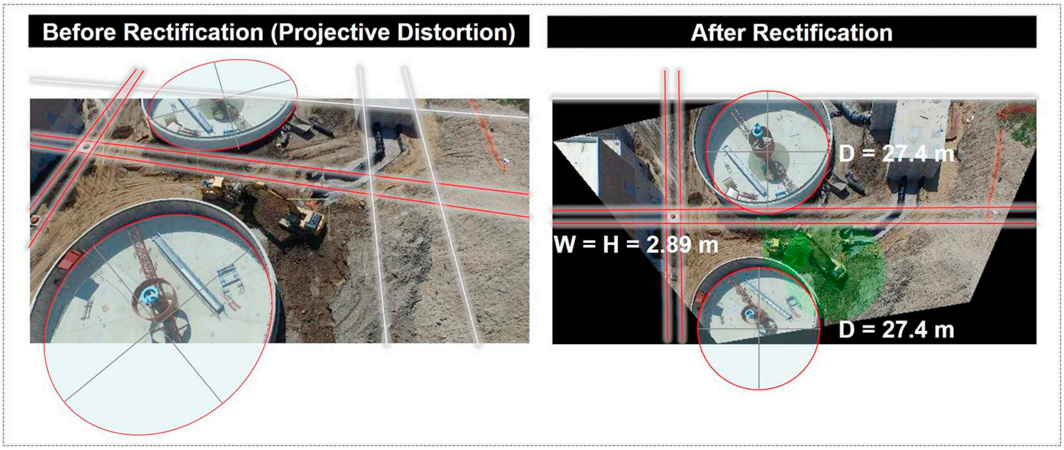

Fig. 5. Projective distortion: before and after rectification.

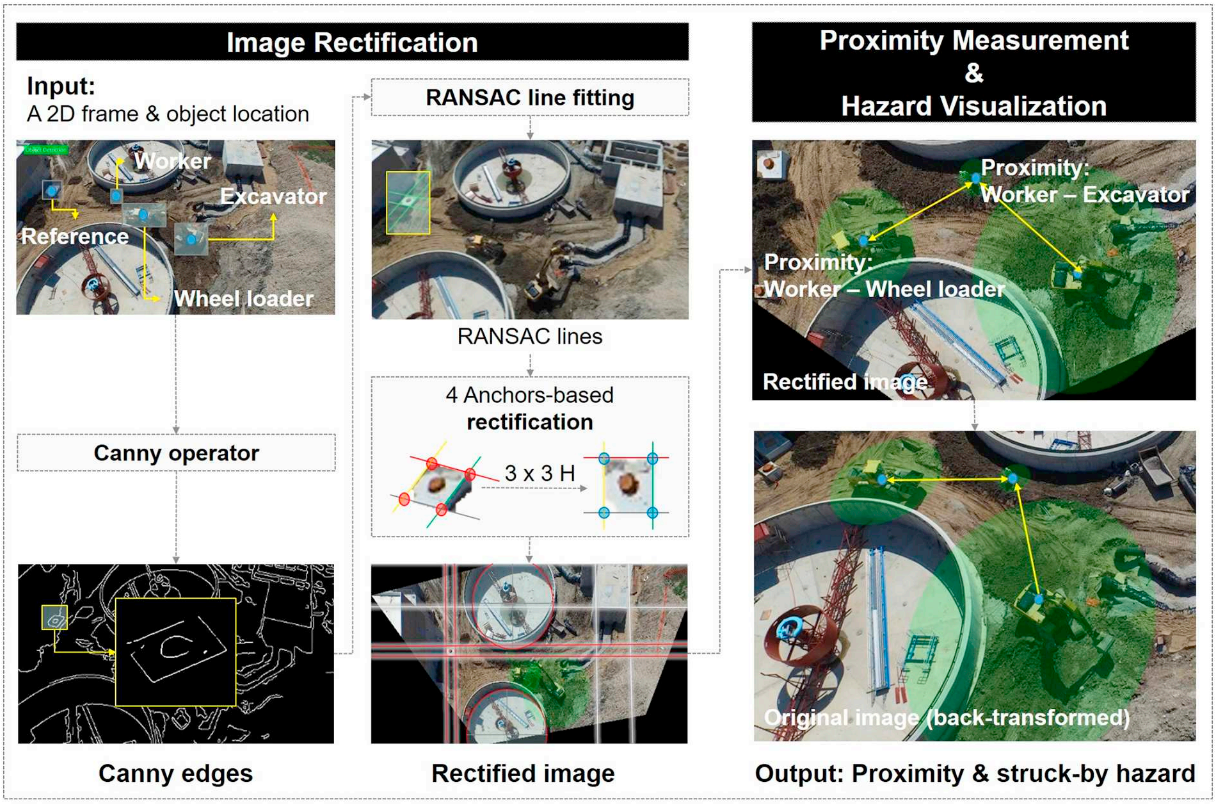

Fig. 6. Overall process of automated image rectification.

5.1. Method description

The proposed method consists of the following six steps: (i) edge detection; (ii) line fitting; (iii) rectification; (iv) proximity measurement; (v) outlier filtering; and (iv) hazard visualization (Fig. 6). The detail explanations are stated as follows.

- Edge detection: the Canny operator is used to detect the edges of the reference object. Because the bounding box of the reference can be given from the fine-tuned object detector (Table 2), it can be applied only to the inside of the box so that the unnecessary edges irrelevant to the reference object can be filtered out. Firstly, the Gaussian filter (size = 7 × 7, sigma = 1) is applied to remove noises on the input image. Then, the Sobel operator generates the edge map with its magnitude and direction. Subsequently, the non-maximum suppression refines candidate edges to have the minimum thickness. Lastly, the hysteresis thresholding (i.e., high threshold = 0.6 and low threshold = 0.24) filters out the false positive edges. Accordingly, delicate (i.e., one-pixel thickness) and accurate edges can be detected, which are used as samples for fitting the reference object's contours in the next step.
- Line fitting: using the detected edges, the contours of the reference object can be inferred. Among several line fitting methods (e.g., HOUGH transform), this method adopts the RANdom Sample Consensus (RANSAC) that discounts outliers for robust operation. Through the RANSAC line fitting, the best lines passing through detected edges are inferred as contours of the reference object. Firstly, two points are sampled at random. Then the line passing through them is drawn with its inline zone. In sequence, the number of inliers is counted. By iterating this, the best line having the largest number of inliers is saved as a contour. In this method, the threshold value of the one-pixel distance is used for determining inlier boundary and the model is iterated 2000 times for fitting one contour. Through the RANSAC, the four contours of the reference object are inferred, and in turn, the four anchor points (i.e., the crossing point of two contours) can be detected.
- Rectification: the way to rectify a distorted image starts from finding the geometric transformation matrix (i.e., homography) that links the distorted dimension to the corresponding ground truth. By matching the estimated location of the four anchor points and that of the ground truth, the linear equation, i.e., Eq. (4), is established, which can be solved by direct linear transformation (DLT) algorithm using singular value decomposition (SVD). Once the 3 × 3 homography is found, the whole frame can be rectified by applying this homography to every single pixel over a frame (Eq. (5)).

$$\begin{bmatrix} x_1 x_1 1\ 0\ 0\ 0 - x'_1 x_1 - x'_1 y_1 \\ 0\ 0\ 0 x_1 y_1 1 - y'_1 x_1 - y'_1 y_1 \\ \vdots \\ x_4 x_4 1\ 0\ 0\ 0 - x'_4 x_4 - x'_4 y_4 \\ 0\ 0\ 0 x_4 y_4 1 - y'_4 x_4 - y'_4 y_4 \end{bmatrix} * \begin{bmatrix} h_{11} \\ h_{12} \\ \vdots \\ \vdots \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

Note: ($x_{(1–4)}$, $y_{(1–4)}$) stand for the estimated locations of anchor points; ($x'_{(1–4)}$, $y'_{(1–4)}$) stand for the ground truth location of anchor points; and $h_{(11–32)}$ stand for the elements of the 3 × 3 homography (h_{33} is always 1).

$$W \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

Note: (X,Y) stands for the rectified coordinates of an original pixel; (x,y) stands for the coordinates of an original pixel; and W stands for a scale factor.

- Proximity measurement: after removing the projective distortion, the proximity between a worker and equipment can be estimated by calculating the Euclidean distance between them. In doing so, the

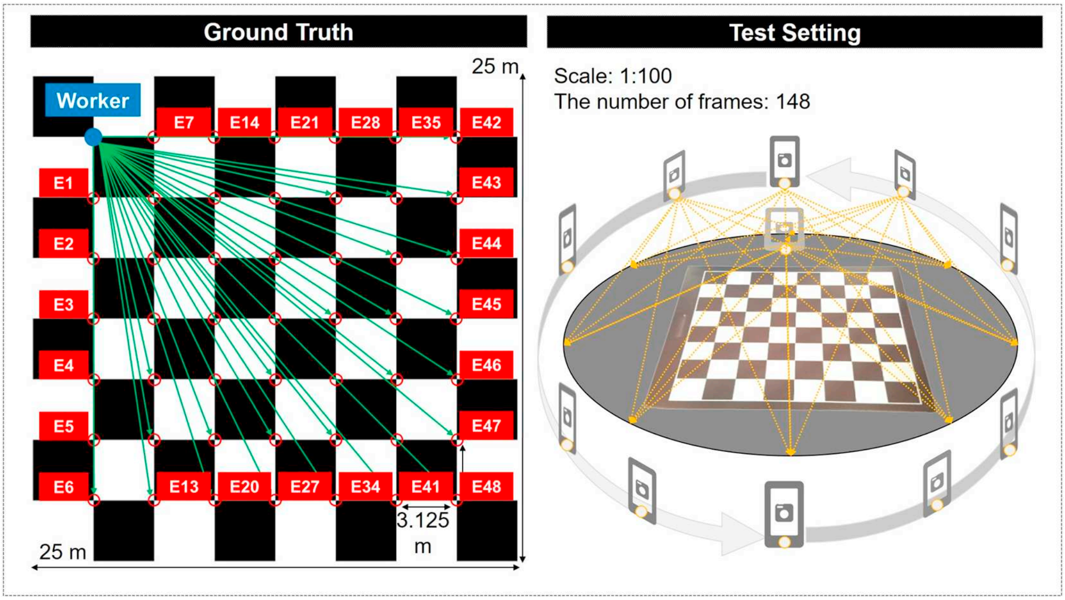

Fig. 7. Rectification test: ground truth vs. test setting.

pixel distance is converted to the metric unit, considering the scene scale known from the reference object's dimension (Eq. (6)).

$$Proximity_{meter} = \frac{Reference_{meter}}{Reference_{pixel}} * Proximity_{pixel} \tag{6}$$

Note: $Reference_{meter}$ stands for the ground truth width of the reference (unit: meter); and $Reference_{pixel}$ stands for the estimated width of the reference on the rectified image (unit: pixel).

- Outlier filtering: the misdetection of an anchor point will deteriorate the overall process, resulting in irregular outliers of proximity. An outlier filter is therefore embedded to automatically detect and offset potential outliers. This filter tracks the mean value of previous two estimations and determines whether the current one is an outlier or not by inlier thresholding (i.e., inlier buffer = 50 pixel distances). Once an outlier is detected, the filter replaces it with the mean value of the previous two estimations.
- Struck-by hazard visualization: additionally, the struck-by hazard around equipment is visualized with a user-adjustable diameter. This research uses the action radius of equipment as a default value for the diameter of a struck-by hazard.

5.2. Test result

A lab-scale test was conducted to evaluate the effect of rectification in measuring distance (i.e., proximity). Fig. 7 illustrates the test settings. The 8 × 8 square checkerboard (width = height = 25 cm) was used to describe a real ground plane (width = height = 25 m) with 1:100 scale. The left top corner was selected as a worker's location and the others as possible locations of equipment, from which the ground truths for the 48 proximities were established (Table 4). An aerial video was filmed using a mobile cell phone, by taking UAV-like motion (i.e., varying location and orientation), as if the video was recorded by a camera-mounted UAV (Fig. 7).

This test measured the proximities both on original and rectified images, and compared them with pre-defined ground truth proximities. The overall accuracy (i.e., for before and after rectification) was determined by the mean absolute percentage error (MAPE) (Eq. (7)).

$$\text{Accuracy} = 100\% - \frac{1}{n} * \sum_{i=1}^{n} {}^{|P_g - P_e|}\!\big/_{P_g} * 100 \tag{7}$$

Note: n stands for the total number of targets (i.e., 48); P_g stands for ground truth proximity; and P_e stands for estimated proximity.

Table 4
Proximity accuracy (before rectification).

Target #	Ground truth (unit: meter)	Proximity after rectification (below row = frame #)						
		1	2	3	······	146	147	148
1	3.13	3.13	3.13	3.13	······	3.13	3.13	3.13
2	6.25	6.24	6.24	6.24	······	9.16	9.16	9.12
3	9.38	9.36	9.36	9.36	······	12.07	12.05	11.99
4	12.50	12.49	12.48	12.48	······	14.95	14.91	14.80
5	15.63	15.67	15.66	15.65	······	14.95	14.91	14.80
44	19.76	19.78	19.76	19.73	······	16.47	16.40	16.28
45	20.96	20.99	20.97	20.94	······	17.61	17.52	17.37
46	22.53	22.61	22.60	22.56	······	19.01	18.90	18.73
47	24.41	24.59	24.57	24.53	······	20.63	20.50	20.29
48	26.52	26.84	26.82	26.78	······	20.63	20.50	20.29
100% - MAPE		99.71	99.72	99.70	······	86.27	86.00	85.61
Overall accuracy (%)								93.51%

Table 5
Proximity accuracy (after rectification).

Target #	Ground truth (unit: meter)	Proximity after rectification (below row = frame #)						
		1	2	3	......	146	147	148
1	3.13	3.13	3.13	3.13	......	3.13	3.13	3.13
2	6.25	6.24	6.24	6.23	......	6.24	6.24	6.24
3	9.38	9.36	9.34	9.31	......	9.35	9.33	9.34
4	12.50	12.50	12.46	12.40	......	12.47	12.43	12.46
5	15.63	15.69	15.63	15.53	......	15.63	15.55	15.60
				⋮				
44	19.76	19.91	19.93	19.84	......	20.43	20.15	20.23
45	20.96	21.11	21.12	21.00	......	21.65	21.33	21.43
46	22.53	22.74	22.73	22.56	......	23.29	22.91	23.05
47	24.41	24.73	24.70	24.47	......	25.31	24.86	25.03
48	26.52	27.01	26.96	26.66	......	27.61	27.07	27.28
100% - MAPE		99.58	99.00	99.55	......	92.87	91.60	92.87
Overall accuracy (%)								97.43%

As the result, it was shown that the average accuracy of proximity after the rectification was more than 97% (Table 5), which outperforms the original accuracy by 3.93 points (i.e., before = 93.51%, Table 4). Furthermore, it was revealed that the effect of rectification is to be greater when a higher extent of projective distortion exists on an image. For example, in the case of the 110th frame of a diagonal viewpoint, the rectification could improve the accuracy by 25 points (i.e., before = 68.32% and after = 93.33%) (Fig. 8). Given the fact that the extent of the distortion is far more serious in usual UAV-captured videos (Fig. 5), the effect of rectification is expected to be greater than this lab scale test.

Table 6
Overview of the test for mobile entities.

Categories		Description
The # of total frames		10,614
The # of frames analyzed		398
Resolution		3840 × 2140
Target's action radius	A worker	2 m
	A wheel loader	5.8 m [46]
	An excavator	12.1 m [45]
Reference object	A quadrate concrete footing	Dimension: width = height = 2.89 m
Evaluation metrics		ADE and MAPE

6. Tests on real-site aerial videos

To evaluate the proposed method's accuracy in real-world application, this research conducts two tests on real-site aerial videos. The first tests the ability for mobile construction entities to work a normal operation whereas the second test targets stationary entities in a controlled environment.

6.1. Test on mobile construction entities

Table 6 provides the overview of the first test. The video was filmed at a real construction site by a camera mounted-UAV. It is comprised of 10,614 consecutive frames. The 398 frames capturing worker-equipment interactions were sampled for this test. In this work, the proximity between a worker and two pieces of equipment (i.e., wheel loader and excavator) were analyzed (Fig. 9).

The primary challenge of this test was to secure a comparison benchmark. While it would have been ideal to directly measure ground truth proximity on the site while filming the video, it was a challenge to measure the proximity on the field without interrupting the site operations, while also facing additional barriers to implementation (e.g.,

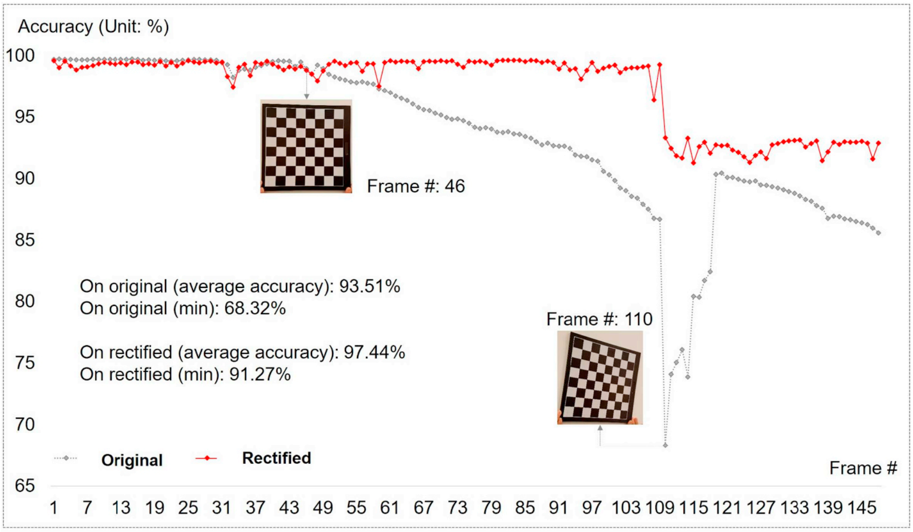

Fig. 8. Proximity accuracy: before vs. after rectification.

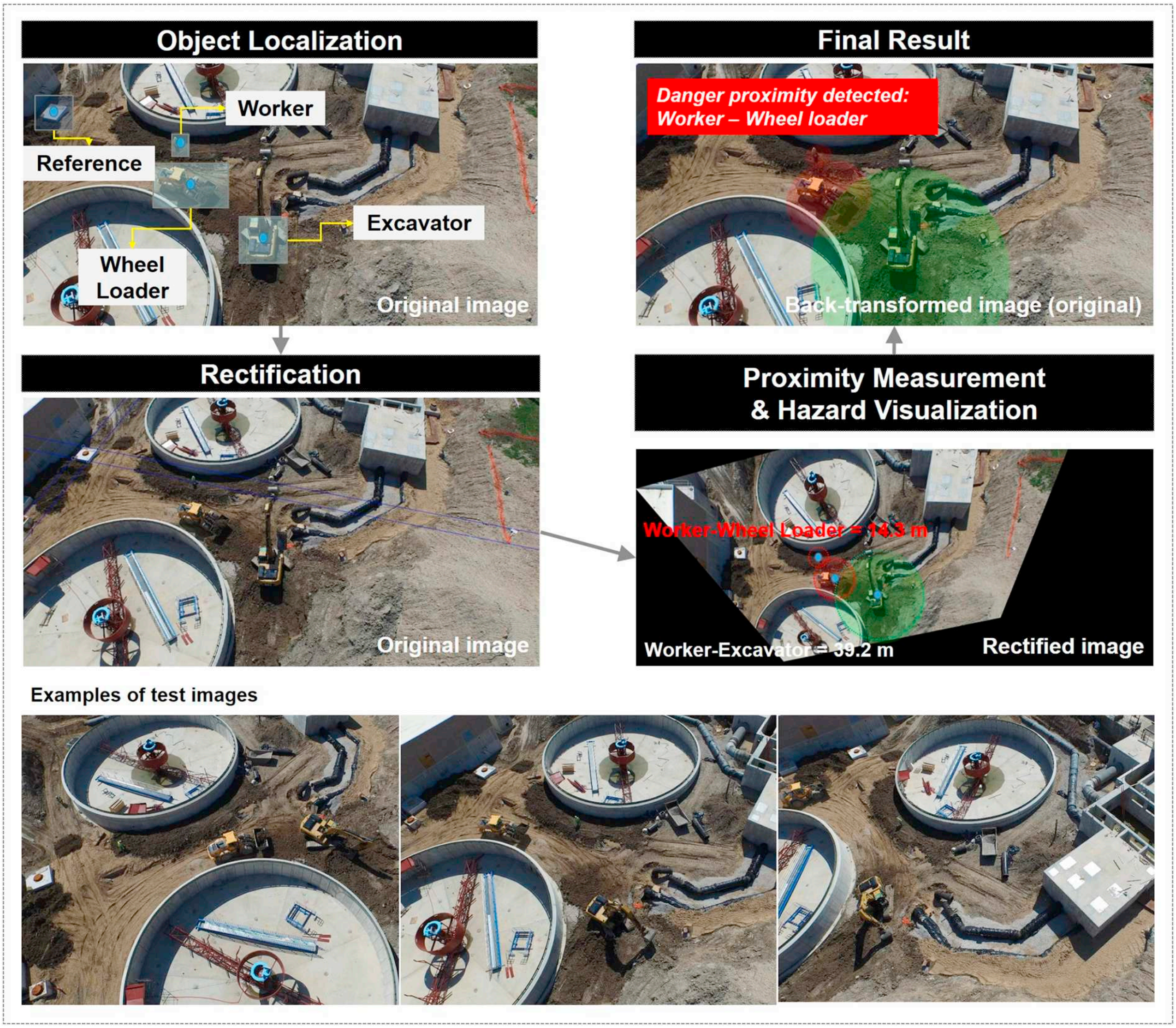
Object Localization
Worker
Reference
Excavator
Wheel Loader
Original image
Final Result
Danger proximity detected: Worker – Wheel loader
Back-transformed image (original)
Rectification
Original image
Proximity Measurement & Hazard Visualization
Worker-Wheel Loader = 14.3 m
Worker-Excavator = 39.2 m
Rectified image
Examples of test images

Fig. 9. Test on mobile construction entities: operational procedure.

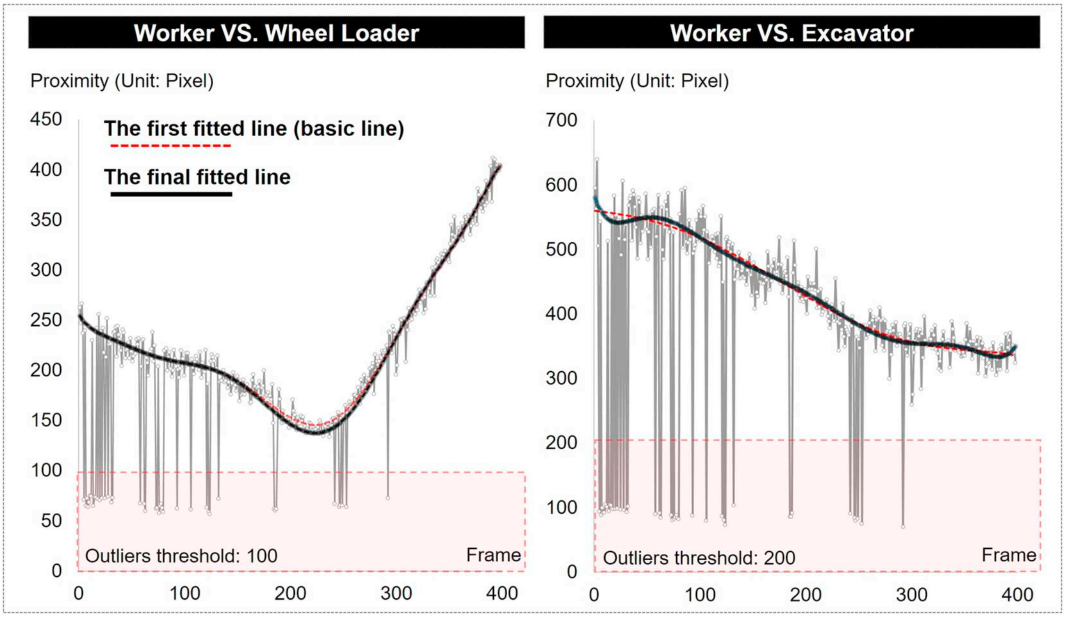

Fig. 10. Inference on comparison benchmark.

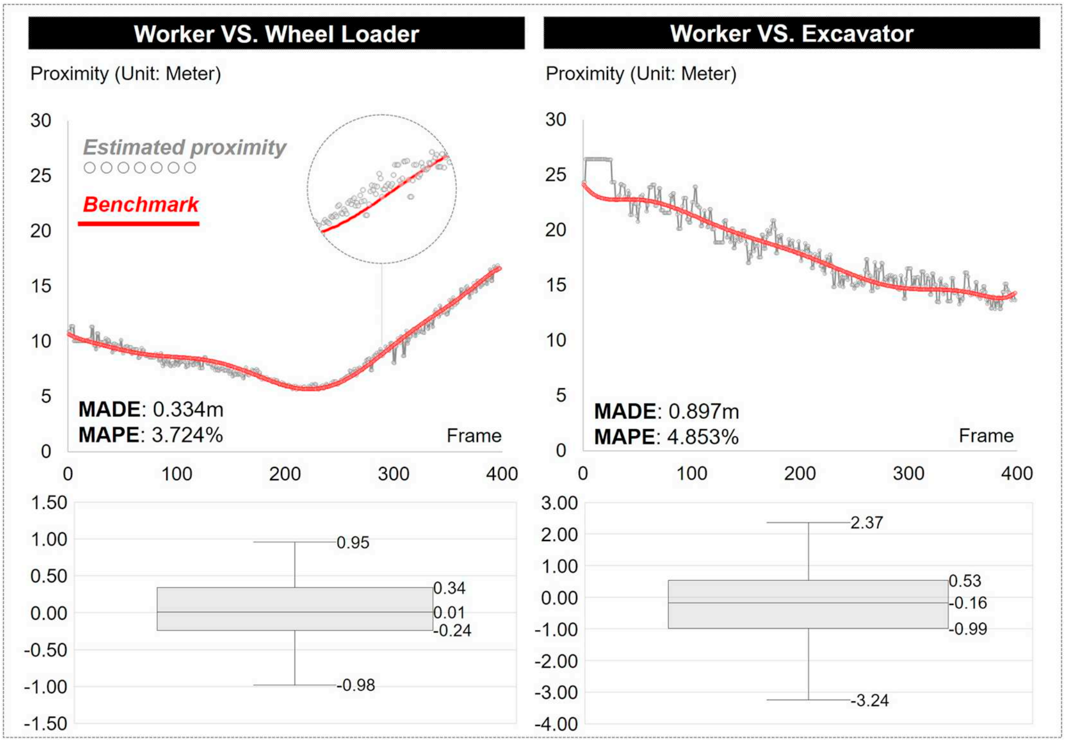

Fig. 11. Test result for mobile entities: estimation vs comparison benchmark.

safety issues). As an alternative, we used entities' location information, which we annotated manually, and applied a statistical inference process to secure a reasonable substitute for the ground truth proximity. Once correct locations for the two targets were given, errorless rectification allowed for calculating the ground truth proximity between them. In the real scene application, however, the rectification could be influenced by noises, which can result in a ground truth estimation dispersed with outliers. As shown in Fig. 10, this raw estimation (i.e., each point) itself cannot be reliable as it contains a wide scope of errors and ignores continuity of a proximity. However, the obvious trend line exists in there, which can be a valid comparison benchmark, once a reasonable inference process is given. The following steps were applied to attain this end (Fig. 10): (i) removing outliers by thresholding; (ii) fitting baseline (i.e., dotted line); (iii) removing additional outliers from the baseline with 1.0 standard deviation; and (iv) fitting the final trend line (i.e., solid line). In this test, 9th-order polynomial model was used for fitting the baseline and the final trend line, considering the proximity pattern of given test dataset.

As shown in Fig. 11, the proximity estimate was compared to the comparison benchmark. As an evaluation metric for accuracy, the mean absolute distance error (MADE) was used (Eq. (8)) along with the corresponding MAPE (Eq. (9)). It was shown that the estimation was close to the benchmark proximity in both cases (i.e., worker-wheel loader and worker-excavator) with the acceptable MAPE: 3.72% and 4.85%, respectively. The MADE for worker—wheel loader was 0.33 m and that for worker—excavator was 0.89 m. Moreover, the proposed method showed unbiased performance with having evenly spread distance errors (i.e., residuals) around median values (i.e., 0.01 m and −0.16 m, respectably).

$$\text{MADE} = \frac{1}{n} * \sum_{i=1}^{n} |P_b - P_e| \tag{8}$$

$$\text{MAPE} = \frac{1}{n} * \sum_{i=1}^{n} |P_b - P_e| / P_b * 100 \tag{9}$$

Note: n stands for the number of frames (i.e., 398); P_b stands for benchmark proximity; and P_e stands for estimated proximity.

6.2. Test on stationary construction entities

The details of the second test are summarized in Table 7 and Fig. 12. An aerial video was filmed at a real construction site using a mobile cell phone, by taking UAV-like motion. Unlike the previous work, this test fixed the locations of targets to secure the ground truth proximity. In this test, the proposed method estimated the proximity between a stationary worker and an excavator (Fig. 12). And the estimate was compared to the pre-defined ground truth proximity. Two cases of ground truth proximity were analyzed in this test: (i) case #1: 15 m and (ii) case #2: 20 m (Fig. 12).

The proximity estimate was compared to the ground truth. It turned out that the estimation was close to the ground truth in both cases (i.e., 15 m and 20 m) with the acceptable MAPE: (i) case #1: 4.214% and (ii) case #2: 4.462% (Fig. 13). The MADE for the case #1 was 0.632 m and that for the second case was 0.892 m (Fig. 13).

Table 7
Overview of the test for stationary entities.

Categories		Description
Resolution		1920 × 1080
Ground truth	Case #1	15 m
	Case #2	20 m
The # of frames analyzed	Case #1	50 frames
	Case #2	50 frames
Target	A worker	Stationary
	An excavator	Stationary
Reference object	A tetragonal concrete footing	Dimension: 2.4 m–2.6 m-0.6 m-2.6 m-0.8 m
Evaluation metrics		ADE and MAPE

7. Discussion on test results

In the first test to target mobile construction entities, the fine-tuned

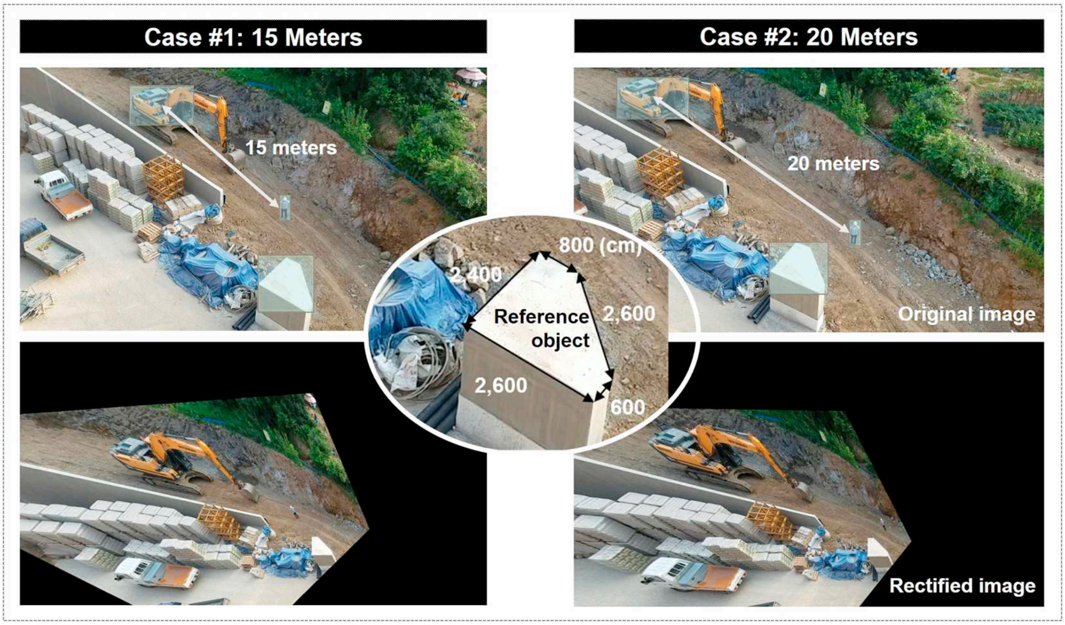

Fig. 12. Test on stationary entities: operational procedure.

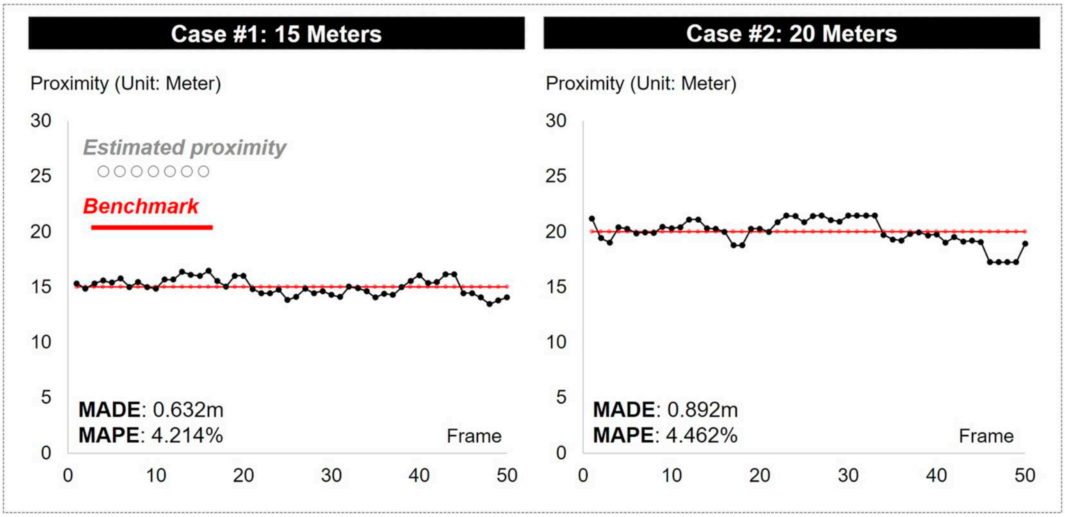

Fig. 13. Test result for stationary entities: estimation vs ground truth.

detector (i.e., YOLO-V3) showed robust localization performance; the localization error (Eq. (3)) for the three construction entities (i.e., a worker, a wheel loader, and an excavator) could be held around 0.3 m even under viewpoint, scale, and illumination variations occurring in the test videos. In achieving the invariant performance were two primary contributories: (i) transfer learning; and (ii) fine-tuning with the data having a wide range of variations. First, balancing between the model capacity and the amount of training data is critical in avoiding overfitting. However, the amount of data collected in this research (i.e., 4512 images) was not ideal for training the original YOLO-V3 architecture to have deep layers (i.e., total 106 layers) from scratch. This research, therefore, elected transfer learning. To be more specific, we took the YOLO-V3 network pre-trained with ImageNet benchmark dataset and used its weights as the starting point of fine-tuning. Naturally, network modifications were made for fitting the original architecture to our dataset (i.e., adjustment of the size of the final feature tensors). By starting from pre-validated weights, the network could achieve well-balanced training without overfitting, thereby making it possible to have an equivalently robust performance on both the training and test dataset. Second, fine-tuning with data involving a wide range of variations helped to enhance the invariant localization capability. This research primarily used images extracted from the videos captured in various construction sites, which covered a wide range of variations regarding illumination, viewpoint, and scale. Fine-tuning with the variable data helped to optimize parameters, e.g., coefficients of convolution kernels, to be invariant to such variations. The parameters could construct consistent feature tensors in successive frames, which in turn led to the robust localization results.

With the localization result, the image rectification method could lead to a reliable proximity measurement between the three entities, successfully removing the projective distortion. First of all, the anchor-points detection using the Canny operator and the RANSAC line fitting was hardly affected by the viewpoint, scale, and illumination variations with advantages of non-maximum suppression and hysteresis thresholding. Given the precise locations of the anchor-points, the rectification method could solve the unique solution for the geometric transformation matrix toward the undistorted original scene and thus could get reliable proximity estimates. On the other hand, the rectification could not be successful at times (i.e., 37/398, in the test for mobile construction) due to aggregates of noise pixels (e.g., sands covering the reference objects). However, all outliers of the estimated proximity resulted from the rectification failures could be successfully detected and refined by the outlier filter. As the result, the proposed method could achieve a promising accuracy of the proximity estimate (i.e., worker—wheel loader: 0.33 m MADE and 3.72% MAPE, worker—excavator: 0.89 m MADE and 4.85% MAPE). In real-world applications, specifically, when detecting struck-by hazards, this minor amount of error would be offset by adding an extra buffer (e.g., 1 m) to the action radius of equipment.

Following the first test, we conducted an additional test focusing on stationary targets (i.e., an excavator and a worker). This test is designed to compare the proximity estimates from our method with the ground truth proximity directly measured on the site in order to validate the proposed method more convincingly. Consequently, the proposed method showed promising accuracy in the second test as well. The MADEs for both cases (i.e., 15 m and 20 m proximity) was less than 1 m and corresponding MAPEs were around 4%. The results clearly show the validity of the proposed method.

8. Toward real-world applications

Despite the promising performance, there are several areas where we can improve toward real-world applications. This section discusses several improvement points, specifically as to (i) generalization capability of the fine-tuned network; (ii) rectification accuracy; (iii) computing efficiency; and (iv) the development of an integrated system.

8.1. Generalization capability of the YOLO-V3

This study collected construction images whose extent, however, may not be enough to generalize the YOLO-V3 to the usual construction environment. This is because a network trained with the limited source of data would involve high variance, whose capability would be restricted into small contexts of the limited training data. The one absolute solution to improving the generalization capability should be training with a vast volume of data capturing various construction scenes. As an axiom of deep learning, the more data a model consumes, the better generalized capability the model will achieve. With the vast dataset secured, a comprehensive comparative analysis of various DNNs can also follow. The performance of a DNN depends on its model architecture as well as adjustable hyper-parameters (e.g., the number and size of convolutional layers or learning rate) [36–39]. Therefore, examining alternative networks with various scenarios of hyper-parameters will lead to finding the best network for the localization of construction entities.

8.2. Rectification accuracy

When a reference object appears unsoiled in an image, the proposed rectification method can output a complete rectification result. As shown in Fig. 14, the precise locations of the anchors can be detected, and an accordingly perfect rectification can be secured on a virtual image. However, it may be difficult to expect such a clear state of an object in the real construction site. In the test video, there were several aggregates of noise pixels covering the reference object such as sands or a plate. The presence of these noises could cause false-positive contour lines and anchor-points because the anchor-points detection algorithms, i.e., Canny operator and RANSAC line fitting, are operated depending on the configuration of pixel values. In the first test, the rectification failures happened 37 times among the total 398 frames all of which

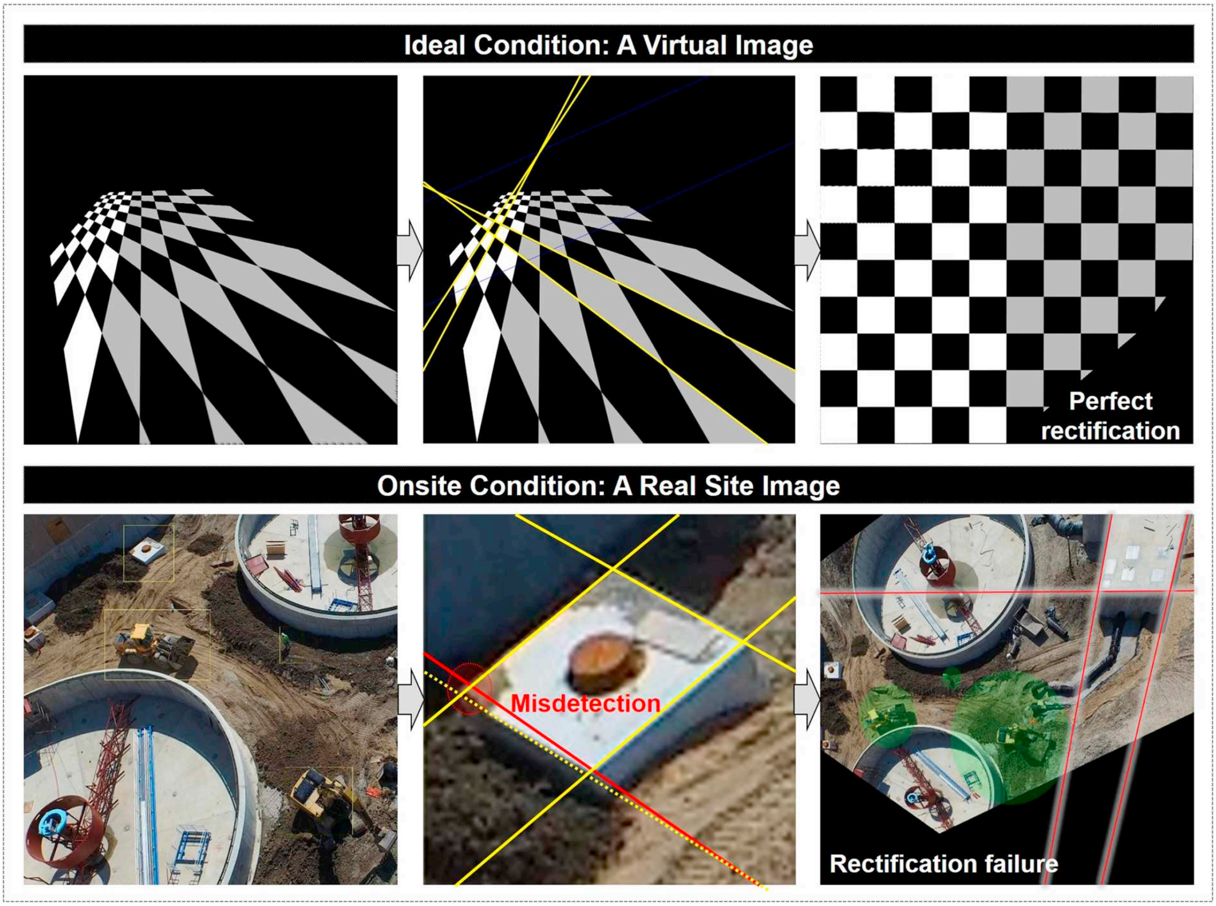
Ideal Condition: A Virtual Image
Perfect rectification
Onsite Condition: A Real Site Image
Misdetection
Rectification failure

Fig. 14. Rectification performance: virtual condition vs. onsite condition.

were attributed to the noise pixels. As in the test with a virtual image (Fig. 14), the use of unsoiled reference object that has clear contour lines and distinctive color to surroundings would be the one simple but powerful solution for this problem. Furthermore, designing a new filter that can automatically remove aggregates of noise pixels having nonlinear patterns could also be an effective solution to reduce the chance of a mis-rectification.

8.3. Computational efficiency

Computational efficiency significantly matters in proximity monitoring as it ultimately aims at timely intervention. This research used a single graphic processing unit (GPU, NVIDIA Tesla K40) server, and consequently the total computational cost of the proposed method was estimated at approximately 0.278 s per frame (i.e., 0.028 s for localization and 0.25 s for rectification). However, when considering the travel speed of a vehicle (or equipment), the sparse estimation (i.e., 3.6 times per a second) proves insufficient. For example, a wheel loader (B877, SDLG) can travel 3.4 m for the 0.305 s and also an excavator (328D LCR, CAT) can swing 3.2 m in that moment [41,42]. This momentary change could result in a struck-by accident while the proximity monitoring lagged. However, there are still opportunities to improve the computational efficiency of the proposed method. Computing cost of a DNN significantly depends on a capacity of parallel computing. Hence, the use of multiple GPU servers (i.e., cloud server) would improve the computational efficiency in object localization. Additionally, computing cost for the rectification would also be reduced by parallel programming and therefore leveraging the GPU capacity.

8.4. Development of an integrated system

It is also necessary to build a system incorporating (i) imaging devices (i.e., a camera-mounted UAVs), (ii) a cloud computing device, and (iii) feedback receivers (e.g., wrist band or smart safety glasses), for real-world applications. Leveraging Internet of Thing (IoT) cloud platform currently available on the market (e.g., Amazon Web Services (AWS) IoT Platform, Microsoft Azure IoT Hub, IBM Watson IoT Platform, and Oracle IoT Platform) can be a promising solution to achieve this end. This platform could connect multiple imaging devices (i.e., UAVs) generating massive data covering a wide range of site to cloud computing device that can process proximity monitoring in near real-time. Also, the prompt feedbacks could be delivered to workers in struck-by hazards via wearable devices—such as wrist band or smart safety glasses—connected to the cloud. This IoT cloud platform would enable rapid proximity monitoring and intervention with a huge computing capacity.

9. Conclusion

As an alternative technology for onsite proximity monitoring between construction entities, computer vision methods for UAV-assisted visual proximity monitoring were presented in this paper. A DNN for object detection, i.e., YOLO-V3, was applied to the robust and fast localization of construction entities. In addition, an image rectification method that allows for measuring actual proximity on a 2D image was developed. When operated together, the methods can consistently monitor proximity between construction entities in a fully automated way. Tests on real-site aerial videos showed a promising performance of the proposed method; the MADEs were less than 0.9 m and the corresponding MAPEs were around 4%. However, there still remains plenty of room for improvement for real-world application: (i) improving the generalization capability of the fine-tuned network; (ii) improving the computational efficiency of the rectification method; and (iii) building an IoT cloud-based integrated system. With such critical refinement, the proposed method can serve as a proactive and applicable measure for safety intervention against struck-by hazards on construction sites, and can ultimately promote a safer working environment for construction workers.

Acknowledgements

The work presented in this paper was supported financially by a National Science Foundation Award (No. IIS-1734266, '*Scene Understanding and Predictive Monitoring for Safe Human-Robot Collaboration in Unstructured and Dynamic Construction Environment*'). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation. Lastly, the authors wish to specially thank Weston Tanner and John McGlennon from WALSH Construction Co. for their considerate assistance in collecting onsite data.

References

[1] CPWR, The Center for Construction Research and Training, Struck-by injuries and prevention in the construction industry, www.cpwr.com, (2017) (Sep. 17, 2018).

[2] BLS, Bureau of Labor Statistics, Census of fatal occupational injuries (CFOI), https://www.bls.gov/iif/oshcfoi1.htm, (2011–2015) (Sep. 17, 2018).

[3] J. Teizer, B.S. Allread, C.E. Fullerton, J. Hinze, Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system, Autom. Constr. 19 (2010) 630–640, https://doi.org/10.1016/j.autcon.2010.02.009.

[4] E. Marks, J. Teizer, Proximity sensing and warning technology for heavy construction equipment operation, Construction Research Congress 2012, West Lafayette, IN, USA, 2012, https://doi.org/10.1061/9780784412329.099.

[5] J. Teizer, Wearable, wireless identification sensing platform: self-monitoring alert and reporting technology for hazard avoidance and training (SmartHat), Electron. J. Inf. Technol. Constr. 20 (2015) 295–312 http://www.itcon.org/2015/19 (Oct. 31, 2018).

[6] S.G. Pratt, D.E. Fosbroke, S.M. Marsh, "Building safer highway workzones: measures to prevent injuries from vehicles and equipment." Department of Health and Human Services: Center for Disease Control and Prevention, https://stacks.cdc.gov/view/cdc/6422, (2001) (Sep. 17, 2018).

[7] T.M. Ruff, Monitoring blind spots: a major concern for haul trucks, Eng. Min. J. 202 (12) (2001) 17–26 https://stacks.cdc.gov/view/cdc/9080 (Sep. 17, 2018).

[8] J.W. Park, E. Marks, Y.K. Cho, W. Suryanto, Performance test of wireless technologies for personnel and equipment proximity sensing in work zones, J. Constr. Eng. Manag. 142 (1) (2016) 04015049, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001031.

[9] J.W. Park, X. Yang, Y.K. Cho, J.W. Seo, Improving dynamic proximity sensing and processing for smart work-zone safety, Autom. Constr. 84 (2017) (2017) 111–120, https://doi.org/10.1016/j.autcon.2017.08.025.

[10] D.H. Kim, K. Yin, M. Liu, S.H. Lee, V.R. Kamat, Feasibility of a drone-based on-site proximity detection in an outdoor construction site, IWCCE 2017, Seattle, WA, USA, 2017, https://doi.org/10.1061/9780784480847.049.

[11] M.W. Park, I. Brilakis, Construction worker detection in video frames for initializing vision trackers, Autom. Constr. 28 (2012) 15–25, https://doi.org/10.1016/j.autcon.2012.06.001.

[12] M. Memarzadeh, M. Golparvar-Fard, J.C. Niebles, Automated 2D detection of construction equipment and workers from site video streams using histogram of oriented gradients and colors, Autom. Constr. 32 (2013) 24–37, https://doi.org/10.1016/j.autcon.2012.12.002.

[13] H.J. Kim, K.N. Kim, H.K. Kim, Vision-based object-centric safety assessment using fuzzy inference: monitoring struck-by accidents with moving objects, J. Comput. Civ. Eng. 30 (2016) 04015075, , https://doi.org/10.1061/(ASCE)CP.1943-5487.0000562.

[14] I. Brilakis, M.W. Park, G. Jog, Automated vision tracking of project related entities, Adv. Eng. Inform. 25 (2011) 713–724, https://doi.org/10.1016/j.aei.2011.01.003.

[15] Y.J. Ham, K.K. Han, J. Lin, M. Golparvar-Fard, Visual Monitoring of Civil Infrastructure Systems Via Camera-equipped Unmanned Aerial Vehicles (UAVs): A Review of Related Works, 4(1) Springer, Visualization in Engineering, 2016, pp. 1–8, https://doi.org/10.1186/s40327-015-0029-z.

[16] J. Lin, K. Han, Y. Fukuchi, M. Eda, M. Golparvar-Fard, Model Based Monitoring of Work in Progress Via Images Taken by Camera Equipped UAV and BIM, International Conference on Civil and Building Engineering Informatics, Tokoy, Japan, 978-4-9907371-1-5, 2015.

[17] K. Han, J. Lin, M. Golparvar-Fard, A Formalism for utilization of autonomous vision-based systems and integrated project models for construction progress monitoring, Conference on Autonomous and Robotic Construction of Infrastructure. Ames, IA, USA, 2015 https://lib.dr.iastate.edu/intrans_reports/141/ (Sep. 17, 2018).

[18] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, G. Reitmayr, Augmented reality for construction site monitoring and documentation, Poc. IEEE 102 (2) (2014) 137–154, https://doi.org/10.1109/JPROC.2013.2294314.

[19] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, S. Tadokoro, Collaborative mapping of an earthquake damaged building via ground and aerial robots, J. Field Serv. Robot.

29 (5) (2014) 832–841, https://doi.org/10.1007/978-3-642-40686-7_3.

[20] C. Wefelscheid, R. Hansch, O. Hellwich, Three-dimensional building reconstruction using images obtained by unmanned aerial vehicles, International Conference on Unmanned Aerial Vehicle in Geomatics, Zurich, Switzerland, 2011 https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-1-22/183/2011/isprsarchives-XXXVIII-1-C22-183-2011.pdf (Sep. 17, 2018).

[21] C. Eschmann, C.M. Kuo, C. Boller, Unmanned aircraft systems for remote building inspection and monitoring, 6th European Workshop on Structural Health Monitoring, Dresden, Germany, 2012 http://www.ecphm2012.com/Portals/98/BB/th2b1.pdf (Spe. 17, 2018).

[22] J. Fernandez Galarreta, N. Kerle, M. Gerke, UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning, Nat. Hazards Earth Syst. Sci. 15 (6) (2015) 1087–1101, https://doi.org/10.5194/nhess-15-1087-2015 (2015).

[23] S. Ye, S. Nourzad, A. Pradhan, I. Bartoli, A. Kontsos, Automated detection of damaged areas after hurricane sandy using aerial color images, Computing in Civil and Building Engineering (2014), Reston, VA. USA, 2014, https://doi.org/10.1061/9780784413616.223.

[24] N. Kerle, J. Fernandez Galarreta, M. Gerke, Urban structural damage assessment with oblique UAV imagery, object-based image analysis and semantic reasoning, Asian Conference on Remote Sensing 2014, At Nay Pyi, Myanmar, 2014 http://a-a-rs.org/acrs/administrator/components/com_jresearch/files/publications/OS-310Kerle_etal_ACRS_2014.pdfhttp://a-a-rs.org/acrs/administrator/components/com_jresearch/files/publications/OS-310Kerle_etal_ACRS_2014.pdf (Sep. 17, 2018).

[25] P. Oskouie, B. Becerik-Gerber, L. Soibelman, A data quality-driven framework for asset condition assessment using LiDAR and image data, Comput. Civ. Eng. 2015 (2015) 240–248, https://doi.org/10.1061/9780784479247.030.

[26] R.J. Dobson, C. Brooks, C. Roussi, T. Colling, Developing an unpaved road assessment system for practical deployment with high-resolution optical data collection using a helicopter UAV, International Conference on Unmanned Aircraft Systems, Piscataway, NJ. USA, 2013, https://doi.org/10.1109/ICUAS.2013.6564695.

[27] C. Zhang, A. Elaksher, An unmanned aerial vehicle-based imaging system for 3D measurement of unpaved road surface distresses, Comput. Aided Civ. Inf. Eng. 27 (2) (2012) 118–129, https://doi.org/10.1111/j.1467-8667.2011.00727.x.

[28] J. Yang, O. Arif, P.A. Vela, J. Teizer, Z. Shi, Tracking multiple workers on construction sites using video cameras, Adv. Eng. Inform. 24 (2010) 428–434, https://doi.org/10.1016/j.aei.2010.06.008.

[29] J. Teizer, P.A. Vela, Personnel tracking on construction sites using video cameras, Adv. Eng. Inform. 23 (2009) (2009) 452–462, https://doi.org/10.1016/j.aei.2009.06.011.

[30] M.W. Park, A. Makhmalbaf, I. Brilakis, Comparative study of vision tracking methods for tracking of construction site resources, Autom. Constr. 20 (2011) (2011) 905–915, https://doi.org/10.1016/j.autcon.2011.03.007.

[31] M.W. Park, I. Brilakis, Continuous localization of construction workers via integration of detection and tracking, Autom. Constr. 72 (2016) (2016) 129–142, https://doi.org/10.1016/j.autcon.2016.08.039.

[32] Z. Zhu, X. Ren, Zhi Chen, Integrated detection and tracking of workforce and equipment from construction jobsite videos, Autom. Constr. 81 (2017) (2017) 161–171, https://doi.org/10.1016/j.autcon.2017.05.005.

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

[34] S.H. Chi, C.H. Caldas, Image-based safety assessment: automated spatial safety risk identification of earthmoving and surface mining activities, J. Constr. Eng. Manag. 138 (3) (2012) 341–351, https://doi.org/10.1061/(ASCE)CO.1943-7862.0000438.

[35] K.N. Kim, H.J. Kim, H.K. Kim, Image-based construction hazard avoidance system using augmented reality in wearable device, Autom. Constr. 83 (2017) (2017) 390–403, https://doi.org/10.1016/j.autcon.2017.06.014.

[36] M.D. Yang, C.F. Chao, K.S. Huang, L.Y. Lu, Y.P. Chen, Image-based 3D scene reconstruction and exploration in augmented reality, Autom. Constr. 33 (2013) (2013) 48–60, https://doi.org/10.1016/j.autcon.2012.09.017.

[37] R. Girshick, J. Donahue, T. Darrell, Region-based convolutional networks for accurate object detection and segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2015) 142–158, https://doi.org/10.1109/TPAMI.2015.2437384.

[38] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 1904–1916, https://doi.org/10.1007/978-3-319-10578-9_23.

[39] R. Girshick, Fast R-CNN, International Conference on Computer Vision (ICCV), Santiago, Chille, 2015, https://doi.org/10.1109/ICCV.2015.169.

[40] S. Ren, K. He, R. Girshick, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 1137–1149, https://doi.org/10.1109/TPAMI.2016.2577031.

[41] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T.M. Rose, W. An, Detecting non-hardhat-use by a deep learning method from far-field surveillance videos, Autom. Constr. 85 (2018) (2018) 1–9, https://doi.org/10.1016/j.autcon.2017.09.018.

[42] H.J. Kim, S.D. Bang, H.Y. Jeong, Y.J. Ham, H.K. Kim, Analyzing context and productivity of tunnel earthmoving process using imaging and simulation, Autom. Constr. 92 (2018) (2018) 188–198, https://doi.org/10.1016/j.autcon.2018.04.002.

[43] Z. Kolar, H. Chen, X. Luo, Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images, Autom. Constr. 89 (2018) (2018) 58–70, https://doi.org/10.1016/j.autcon.2018.01.003.

[44] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, arXiv:1804.02767, (2018).

[45] C.A.T. Machine, 328D LCR hydraulic excavator, http://s7d2.scene7.com/is/content/Caterpillar/C775795, (2012) (Sep. 17, 2018).

[46] S.D.L.G. Machine, Reliability in action: backhoe loader B877, http://www.sdlg-africa.com/wp-content/uploads/, (2014) (Sep. 17, 2018).