

Marquette University

e-Publications@Marquette

Civil and Environmental Engineering Faculty
Research and Publications

Civil, Construction, and Environmental
Engineering, Department of

9-2021

Low–High Orthoimage Pairs-Based 3D Reconstruction for Elevation Determination Using Drone

Yuhan Jiang
Marquette University

Yong Bai
Marquette University, yong.bai@marquette.edu

Follow this and additional works at: https://epublications.marquette.edu/civengin_fac



Part of the [Civil Engineering Commons](#)

Recommended Citation

Jiang, Yuhan and Bai, Yong, "Low–High Orthoimage Pairs-Based 3D Reconstruction for Elevation Determination Using Drone" (2021). *Civil and Environmental Engineering Faculty Research and Publications*. 305.

https://epublications.marquette.edu/civengin_fac/305

Marquette University

e-Publications@Marquette

Civil, Construction and Environmental Engineering Faculty Research and Publications/College of Engineering

This paper is NOT THE PUBLISHED VERSION.

Access the published version via the link in the citation below.

Journal of Construction Engineering and Management, Vol. 147, No. 9 (September 2021): 04021097. [DOI](#). This article is © American Society of Civil Engineers and permission has been granted for this version to appear in [e-Publications@Marquette](#). American Society of Civil Engineers does not grant permission for this article to be further copied/distributed or hosted elsewhere without express permission from American Society of Civil Engineers.

Low–High Orthoimage Pairs-Based 3D Reconstruction for Elevation Determination Using Drone

Yuhan Jiang

Research Assistant, Department of Civil, Construction and Environmental Engineering, Marquette University, Milwaukee, WI

Yong Bai

McShane Chair and Professor, Department of Civil, Construction and Environmental Engineering, Marquette University, Milwaukee, WI

Abstract

This paper presents a 3D reconstruction method for fast elevation determination on construction sites. The proposed method is intended to automatically and accurately determine construction site elevations using drone-based, low–high orthoimage pairs. This method requires fewer images than other methods for covering a large target area of a construction site. An up–forward–down path was

designed to capture approximately 2: 1-scale images at different altitudes over target stations. A pixel grid matching and elevation determination algorithm was developed to automatically match images in dense pixel grid-style via self-adaptive patch feature descriptors, and simultaneously determine elevations based on a virtual elevation model. The 3D reconstruction results were an elevation map and an orthoimage at each station. Then, the large-scale results of the entire site were easily stitched from adjacent results with narrow overlaps. Moreover, results alignment was automatically performed via the U-net detected ground control point. Experiments validated that in 10–20 and 20–40 orthoimage pairs, 92% of 2,500- and 4,761-pixels were matched in the strongest and strong levels, which was better than sparse reconstructions via structure from motion; moreover, the elevation measurements were as accurate as photogrammetry using multiscale overlapping images.

Keywords

Drone; Construction site; 3D reconstruction; Elevation algorithm; Pixel matching.

Introduction

In earthwork operations, elevation determinations help construction professionals in optimizing earth-moving paths (Seo et al. 2011; Gwak et al. 2018), designing temporary hauling roads (Yi and Lu 2016), estimating cost and time (Hola and Schabowicz 2010), designing construction site safety facilities (Wang et al. 2015), and conducting existing facility classification (Shirowzhan and Sepasgozar 2019). State-of-the-practice construction surveying techniques include total station, GPS, measuring tape, level, and theodolite (Nichols and Day 2010). Surveying on construction sites has gradually shifted to remote sensor methods using 3D laser scanning (LiDAR) (Du and Teng 2007; Takahashi et al. 2017; Kwon et al. 2017; Maghiar and Mesta 2018; Shirowzhan and Sepasgozar 2019) and photogrammetry (Nassar and Jung 2012; Siebert and Teizer 2014; Sung and Kim 2016). Areas of photogrammetry application are not limited to 3D mapping of construction sites (Siebert and Teizer 2014), 3D modeling of construction equipment (Kim and Kim 2018), 3D reconstruction of buildings (Aguilar et al. 2019), and 3D modeling of excavation objects (Sung and Kim 2016). However, using 3D mapping of construction sites using drone photogrammetry requires many highly overlapped images, using either aerial triangulation with ordered orthoimage series or structure from motion (SfM) with unordered image collection. The stereo-vision method is much faster because the 3D geometrical information is recovered from same-scale two-frame image pairs (Sung and Kim 2016), while its measurable range (distances from objects' surfaces to cameras) is limited in close range. This work presents a novel approach for automatically and accurately determining construction site elevations using drone-based multiscale orthoimage pairs. The proposed method requires fewer images than other methods for covering a large target area of a construction site. As a result, the proposed method is faster than existing methods and techniques in image collection and image processing. Compared with contact surveying, it simplifies time-consuming outdoor procedures, avoids interfering with other construction operations, and reduces the time for raw data processing compared to remote surveying.

Background

Drone applications in construction engineering and management were initially developed for visual inspection, such as using remote-controlled helicopters for bridge inspection (Metni and Hamel 2007). Recently, the most popular drones were the DJI Phantom series (Wang et al. 2016; Seo et al.

2018; Yang et al. 2018; Moon et al. 2019) and the DJI Inspire series (Omar and Nehdi 2017; Li and Lu 2018; Aguilar et al. 2019), which are small-sized systems of a quadcopter, gimbal, and digital camera. These drone models led to the increase of drone applications in 2015, and photogrammetric 3D point clouds started to surpass 2D images in the referenced studies. Table 1 summarizes drone applications in surveying and modeling, and shows that most of the studies utilized photogrammetry to generate point clouds. The other categories of drone applications include structure and material visual inspection (Metni and Hamel 2007; Zhong et al. 2018; Freimuth and König 2018; Morgenthal et al. 2019), progress monitoring (Han et al. 2018; Zhang et al. 2015; Han and Golparvar-Fard 2017), and safety-related work (Kim et al. 2017, 2019).

Table 1. Surveying and modeling application with drone

References	Surveying and modeling	Drone and camera
Aguilar et al. (2019)	Building/hybrid point cloud: drone and terrestrial photogrammetry	DJI Inspire 1 (SZ DJI Technology, Shenzhen, China)
Chen et al. (2018)	Building/hybrid point cloud: drone photogrammetry and laser scanning	—
Inzerillo et al. (2018)	Road pavement/point cloud: drone and close-range photogrammetry	Quadcopter/GoPro Hero 3 (GoPro, San Mateo, California)
Kim and Kim (2018)	Concrete mixer truck/point cloud: drone photogrammetry	—
Li and Lu (2018)	Construction site/hybrid point cloud: drone photogrammetry and laser scanning	DJI Inspire 1 Pro (SZ DJI Technology, Shenzhen, China)
Moon et al. (2019)	Construction site/hybrid point cloud: drone photogrammetry and laser scanning	DJI Phantom 3 (SZ DJI Technology, Shenzhen, China)
Park et al. (2019)	Construction site/hybrid point cloud: drone and unmanned ground vehicle (UGV) photogrammetry	DJI Mavic (SZ DJI Technology, Shenzhen, China)
Siebert and Teizer (2014)	Construction site/digital elevation model (DTM)	MikroKopter Qual XL (MikroKopter, Moormerland, Leer, Lower Saxony, Germany)/Sony NEX5N (Sony Group Corporation, Sony City, Minato, Tokyo, Japan)
Yang et al. (2018)	Construction site/DTM	DJI Phantom 3 (SZ DJI Technology, Shenzhen, China)

Models and Workflows of Drone and Image-Based 3D Reconstruction

Determining 3D geometrical data via a single image without other useful reference information is an ill-posed problem (Eigen et al. 2014). The reference information of depth can be obtained via LiDAR (Intel 2020a) and Active IR Stereo technology (Intel 2020b). Shang and Shen (2018) used visual simultaneous localization and mapping (SLAM) technology for real-time construction site 3D reconstruction, and achieved an accuracy of average unit error of 3.3 cm compared with photogrammetry. Due to the limited sensing range of the used depth camera, Intel RealSense R200 (operating range 0.5– 3.5 m) (Intel 2016), the drone altitude was set at 4– 5 m above the ground (Shang and Shen 2018). Moreover, elevation information can be obtained from similar objects via a well-trained convolutional encoder-

decoder network model (Jiang and Bai 2020b). This deep-learning model links each pixel of an object's top-view images to its elevation data with an elevation accuracy of 52.43% pixels within a 5.0-cm error range (Jiang and Bai 2020b).

Otherwise, two images and known camera positions are the minimum requirements. Sung and Kim (2016) utilized the stereo-vision method for 3D modeling of a construction site. Dual cameras were faced in front of target objects to capture two same-scale images; then, within a close range, objects' geometrical data were determined from the horizontal disparity of the same-scale left-right image pair via the triangulation model. Adapting this method and extending the coverage with multiple same-scale left-right top-view image pairs is no different from the aerial triangulation method of photogrammetry. Additionally, Matthies et al. (1997, 2007), Li et al. (2002), Xiong et al. (2005), and Meng et al. (2013) applied descent images continuously to determine the topography of landing terrain for space aircraft, where images were captured at different altitudes over the target landing area. Meng et al. (2013) proved that the vertical distance between two adjacent images should be equal to the lower image's altitude; otherwise, the disparity of neighbor pixels around the image's center are not distinguished. Based on that, Jiang and Bai (2020a) developed a vertical baseline stereo-vision model with a small-sized quadrotor drone system for determining vertical distances from ground surfaces to a drone's camera.

Furthermore, photogrammetry, or SfM, is an inexpensive and efficient multi-image-based 3D mapping method (Nassar and Jung 2012; Siebert and Teizer 2014; Haur et al. 2018). Fig. 1 shows the general workflow of photogrammetry and SfM, which starts with acquiring overlapping images. The second step is keypoints matching via scale-invariant feature transform (SIFT) (Lowe 2004) or speeded up robust features (SURF) (Bay et al. 2008), which can extract numbers of feature keypoints at first, and return a matched sparse and irregular keypoints result. This is because most candidate feature keypoints are excluded by the criteria of low contrast and points on edges, and the weakly matched keypoints are excluded as well (Lowe 2004; Solem 2012). After that, the sparse keypoints can be transformed into a 3D sparse point cloud via the aerial triangulation method in case of overlapping top-view image series, or via SfM in case of unordered image collections (Snavely 2010). The other processes are required to produce a dense point cloud via patch-based multiview stereo (PMVS) (Furukawa and Ponce 2010), and generate a triangular irregular networks (TINs) mesh model for volume estimation (Shewchuk 2002; Remondino 2003). Moreover, the generated model is a scale model, which needs at least three ground control points (GCPs) to align to the real-world coordinate (Westoby et al. 2012) or calibrate the model using the accurate GPS or GPS-RTK (real time kinematic) information.

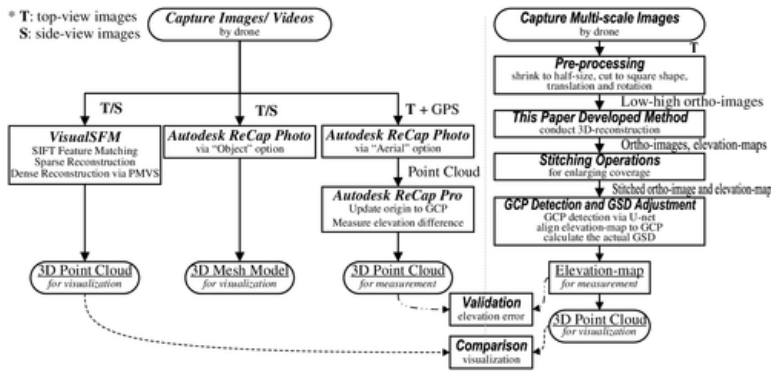


Fig. 1. Comparison of 3D reconstruction workflows.

Requirements and Limitations of Drone and Image-Based 3D Reconstruction

Photogrammetry requires a minimum of 70% and 40% image overlap in longitudinal and traversal coverage (Siebert and Teizer 2014); increasing the longitudinal overlap from 80% to 90% has no significant improvement in measurement accuracy (Takahashi et al. 2017), but results in using more images to cover the target area. Moreover, VisualSFM requires a powerful workstation (with powerful CPU and GPU) to conduct and speed up image matching, sparse reconstruction, and dense reconstruction (Wu 2007, 2013). Autodesk ReCap (Autodesk, Mill Valley, California) requires uploading the collected overlapping images to cloud servers, waiting, and downloading the results to a local computer (Fig. 1). However, photogrammetry has limitations in poorly textured surfaces and reflective materials, such as dense vegetation, paint finish, water, and snow, because these uniform surfaces are unable to generate sufficient keypoints due to low contrast and variation (Lowe 2004; Solem 2012; Westoby et al. 2012). Thus, in practice, when applying photogrammetry in 3D modeling uniform texture objects, attaching marks to target surfaces, such as vehicles and facades, is required (Erickson et al. 2013; Daftry et al. 2015).

Additionally, photogrammetry with top views has poor performance in steep or near-vertical topography, where its error rate has a positive correlation to the slope in the range of 55°–90° (Zhao and Lin 2016; Haur et al. 2018). Adding additional side views by facing the camera lens toward the target objects could help the photogrammetry work for vertical surfaces, such as facade 3D reconstruction (Roca et al. 2013; Daftry et al. 2015); however, this is not a safe option for 3D reconstruction of the slopes of pits and ditches on an active excavation site, because flying a drone close to objects on construction sites can result in collisions and falling hazards, and could invade workers' personal space, diverting their attention and thereby endangering their safety (Susini 2015; Moud et al. 2019). With the latest announced Intel RealSense Depth Camera D455 (recommended range 0.4–6 m) (Intel 2020b), the drone altitude (Shang and Shen 2018) would increase by just a small extent. Thus, for safety, it is better to keep drones away from and above construction sites, and use top views to avoid interruption with other construction operations.

Furthermore, image corresponding matching is an essential process of multi-images based 3D reconstruction. The SIFT method has advantages in image scaling and rotation transformations (Lowe 2004), while the patch-based correlation method has advantages in customization and dense matching, which has been verified in PMVS (Furukawa and Ponce 2010). Normalized cross correlation

(NCC) (Lewis 1995) is suitable for matching same-scale images. NCC is robust in image translations and image brightness changes (Solem 2012; Kaehler and Bradski 2016), but in the case of matching multiscale images, the prescaled patch feature descriptors are needed to make patches to the same dimension for NCC score calculation. Then, pixels can be manually selected from an image in the customized styles, such as a densely packed and uniformly spaced pixel grid used in Jiang and Bai (2020a), and the best-matched corresponding pixels will be returned with high NCC scores. Moreover, compared with the irregular and sparse SIFT matches, the dense pixel grid matches are more convenient for creating and updating as-built 2D site plans.

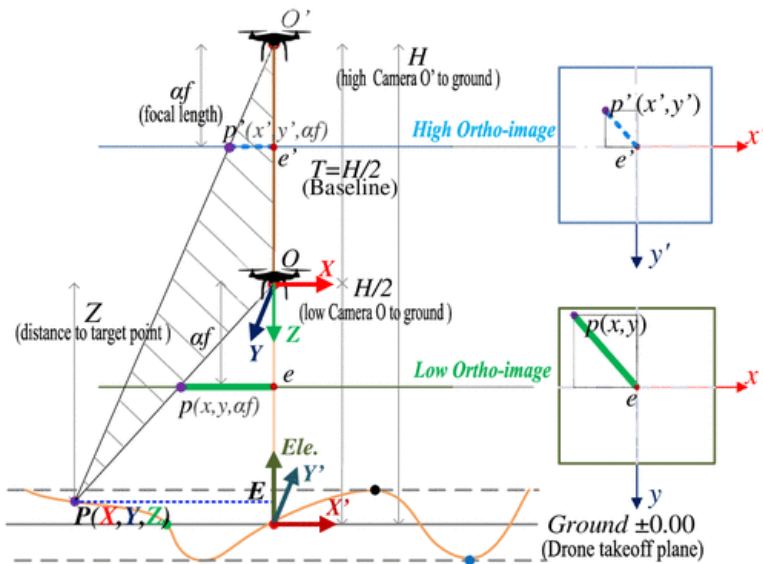


Fig. 2. Low–high orthoimage pair triangulation model.

Research Scope and Contributions

This paper presents a 3D reconstruction method for construction site elevation determination and earthwork estimation. The new method uses few multiscale top-view images (Fig. 1). For each station, the 3D geometrical information of the scanned ground surfaces is determined via a low–high orthoimage pair triangulation model. In Fig. 2, the low orthoimage is captured at the low altitude $H/2$ and the high orthoimage is captured at the high altitude H , which have the same principal point e/e' with the approximate 2: 1 scaling relation, 1: 2 for ground sampling distance (GSD) (Jiang and Bai 2020a). In Fig. 2, the origin of *World Coordinate* (X, Y, Z) is set at *Camera 0*, and the Z -axis is along the camera's principal ray. If the image point pair $p(x, y, af)$ and $p'(x', y', af)$ on the low and high orthoimages are matched, then the construction site point $P(X, Y, Z)$ can be determined by Eq. (1a), with elevation $Ele. = H/2 - Z$ relative to $Ground \pm 0.00$ (Jiang and Bai 2020a)

(1a)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{H}{2f_x} \frac{xx'}{x-x'} \\ \frac{H}{2f_y} \frac{yy'}{y-y'} \\ \frac{H}{2} \frac{x'}{x-x'} \text{ or } \frac{H}{2} \frac{y'}{y-y'} \end{bmatrix}$$

In Eq. (1a), f_x and f_y are focal length for the image in x (width) and y (height) direction; ideally, it has $f_x = f_y = \alpha f$ is the camera's focal length; α is the factor to convert sensor size (mm) to image size (pixel). In Fig. 2, from $\Delta Ope \cong \Delta OPE$, has

(1b)

$$\frac{X}{x} = \frac{Z}{f_x}$$

From $\Delta Op'e' \cong \Delta O'PE$ has

(1c)

$$\frac{X}{x'} = \frac{Z + \frac{H}{2}}{f_x}$$

Subtracting Eq. (1b) from Eq. (1c) gives

(1d)

$$X = \frac{H}{2f_x} \frac{xx'}{x-x'}$$

Similarly, we get

(1e)

$$Y = \frac{H}{2f_y} \frac{yy'}{y-y'}$$

From Eq. (1b) gives

(1f)

$$Z = \frac{X}{x} f_x$$

Similarly, we get

(1g)

$$Z = \frac{Y}{y} f_y$$

Put Eq. (1d) into Eq. (1f) which obtains

(1h)

$$Z = \frac{H}{2} \frac{x'}{x - x'}$$

Similarly, we get

(1i)

$$Z = \frac{H}{2} \frac{y'}{y - y'}$$

When $x' = (x/2)$, $y' = (y/2)$, it has $Z = (H/2)$, which means the point $P(X, Y, Z)$ on *Ground* ± 0.00 (Jiang and Bai 2020a). In addition, for an arbitrary baseline T (vertical distance between the drone's two positions), Eq. (1a) can be rewritten as

(1j)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{T}{f_x} \frac{xx'}{x - x'} \\ \frac{T}{f_y} \frac{yy'}{y - y'} \\ T \frac{x'}{x - x'} \text{ or } T \frac{y'}{y - y'} \end{bmatrix}$$

For each station, due to the 1: 2 GSD relation, four pixels in a low orthoimage cover the same area as a target pixel in a high orthoimage. In reverse, a reference pixel $p(u, v)$ may assemble four subsets with its neighbors via four scaling directions and generate four pixel feature descriptors $g_{u^*, v^*}(u, v)$ via average pooling (Jiang and Bai 2020a). Fig. 3 shows an example of $g_{u-1, v-1}(u, v) = 60$ matches with the target pixel $p'(u', v') = 60$, meaning that the reference pixel $p(u, v)$ is the bottom-right subpixel $p'(u' + 0.5, v' + 0.5)$ of the matched target pixel. Eq. (2c) shows the pixel-to-subpixel matching scheme in pixel coordinate and image coordinate. This scheme follows the rules of *Pixel Coordinate* (u, v) to *Image Coordinate* (x, y) by

(2a)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u - w/2 + 0.5 \\ v - h/2 + 0.5 \end{bmatrix}$$

Image Coordinate (x, y) to *Pixel Coordinate* (u, v) by

(2b)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \text{int}(x + w/2) \\ \text{int}(y + h/2) \end{bmatrix}$$

In addition, in Eq. (2a) *Pixel Coordinate* adjusts 0.5 to get its *Image Coordinate*; similarly, the 0.5 subpixel needs to adjust 0.25 to get its *Image Coordinate*.

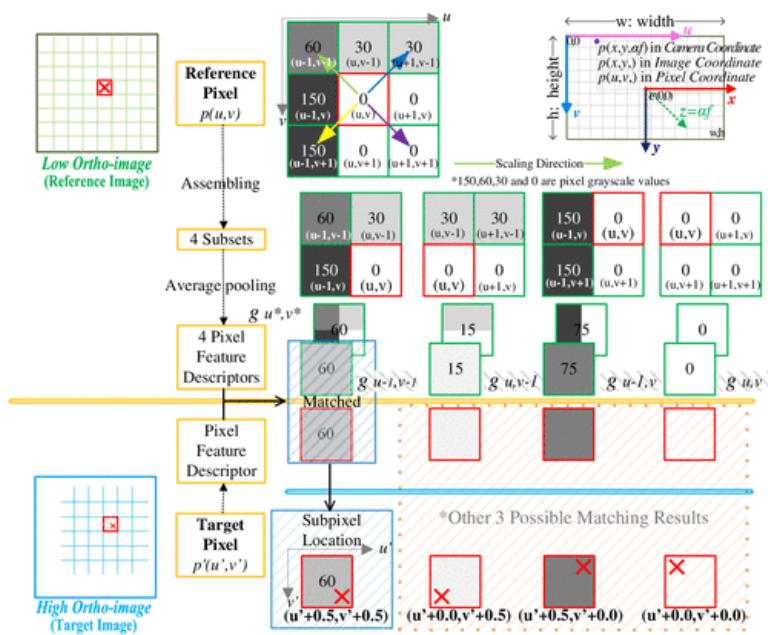


Fig. 3. Pixel-to-subpixel matching scheme.

If a target pixel $p'(u', v')$ matches with a scaling created feature descriptors in

(2c)

$$g_{u^*,v^*}(u, v) = \begin{bmatrix} g_{u-1,v-1}(u, v) \\ g_{u,v-1}(u, v) \\ g_{u-1,v}(u, v) \\ g_{u,v}(u, v) \end{bmatrix} \text{ then return the target subpixel in } \begin{bmatrix} p'(u' + 0.5, v' + 0.5) \\ p'(u' + 0.0, v' + 0.5) \\ p'(u' + 0.5, v' + 0.0) \\ p'(u' + 0.0, v' + 0.0) \end{bmatrix}$$

$$\begin{aligned} \text{target point } p'(x', y') \text{ in } & \begin{bmatrix} \left(u' - \frac{w}{2} + 0.75, v' - \frac{h}{2} + 0.75\right) \\ \left(u' - \frac{w}{2} + 0.25, v' - \frac{h}{2} + 0.75\right) \\ \left(u' - \frac{w}{2} + 0.75, v' - \frac{h}{2} + 0.25\right) \\ \left(u' - \frac{w}{2} + 0.25, v' - \frac{h}{2} + 0.25\right) \end{bmatrix} \text{ and reference point } p(x, y) \\ & = [u - (w/2) + 0.5, v - (h/2) + 0.5] \text{ as pixel-to-subpixel matching results} \end{aligned}$$

For automatically employing this proposed 3D reconstruction method, the following specific objectives and tasks are addressed in this paper: (1) self-adaptive patch feature descriptors for matching 2: 1-scale low–high orthoimages via NCC score; (2) low–high orthoimage pair-based 3D reconstruction that matches pixel pairs in a dense pixel grid-style, while simultaneously determining the elevation data at a quick rate; (3) drone flight path and image acquisition strategies; (4) orthoimage and elevation map results 2D stitching; and (5) GCP detection and GSD adjustment to ensure the accuracy of earthwork estimation. Furthermore, experiments were conducted and comprehensively compared with photogrammetry software Autodesk ReCap and SfM application VisualSfM, followed by the workflows in Fig. 1, to validate the efficiency of the proposed method on an experimental site.

3D Reconstruction Method Development

This section presents and discusses the development of the following *PixelGrid Matching and Elevation Determination Algorithm (PGM&ED)* to realize the low–high orthoimage pair-based 3D reconstruction. It contains two sampling strategies: (1) using a virtual elevation model to discretize a construction site and (2) using a pixel grid format to simplify image matching. Additionally, the four-scaling patch feature descriptors were developed for matching dense pixel-to-subpixel pairs in low–high orthoimage pairs.

Table

	Pixel Grid Matching and Elevation Determination Algorithm ($Img_{H/2}, Img_H, H/2$) # Use low-high orthoimage pair and the altitude difference as inputs
1	initial $Img_1, Img_2, Img_3, Img_4 = ROTATE(Img_{H/2}, [0,90,180,270]); Img'_1, Img'_2, Img'_3, Img'_4 = ROTATE(Img_H, [0,90,180,270])$ # Image Rotation for i in $[1,2,3,4]$ # Repeat the pixel grid matching and elevation-map algorithm 4 times
2	$PixelGrid_i = Pixel\ Grid\ Matching\ Algorithm(Img_i, Img'_i, Grid_{size}, H/2)$ # Multiprocessing with 4 CPU Cores: Each CPU Core computing one orthoimage pair
3	$R_{PixelGrid_i} = f(PixelGrid_i)$ # Transform the matched pixel grids to the same orientation and the same coordinate
4	$x_i[], y_i[], D_i[], W_i[] = R_{PixelGrid_i}[1:4]$ # Assign matched results to separate lists, $x[]$ and $y[]$ are image point coordinate, $D[]$ is the elevations, and $W[]$ is NCC scores
5	$Q_i = MAX(W_i \cdot LowFence, 0.001)$ # Define the threshold classifier to determine match quality, $W[] \geq Q$ strongly matched pixel, $W[] < Q$ weakly matched pixel for j in $[1, \dots, N]$ # N is the num. of total matched pixels
6	$x_{list} \cdot APPEND(x_1[j]); y_{list} \cdot APPEND(y_1[j])$ # Transformed Image Coordinates $p(x, y)_{i=2,3,4}$ are the same as $p(x, y)_{i=1}$
7	$Ele_{list} \cdot APPEND(MEDIAN(D_{i W_i[j] \geq Q_i}[j])); C_{list} \cdot APPEND(1_{W_1[j] \geq Q_1} 2_{W_2[j] \geq Q_2} 3_{W_3[j] \geq Q_3} 4_{W_4[j] \geq Q_4})$ # Combining four results to enhanced elevation result, see Table 2
8	$X' = x_{list}[j] \times GSD; Y' = -y_{list}[j] \times GSD; Ele. = Ele_{list}[j]$ # see Eq. (3f)
9	$PointCloud_{list} \cdot APPEND(P(X', Y', Ele.))$ # Create Point Cloud
10	$Ele_{map\ Matrix}[u - Grid_{size}/2 : u + Grid_{size}/2, v - Grid_{size}/2 : v + Grid_{size}/2] = gray(Ele_{u,v})$ # Update Elevation-map by converting elevations to 8bit-grayscale values
11	$Ortho_{image}, Elevation_{map} = REMOVE([Img_{H/2}, Ele_{map\ Matrix}], Margin)$ # Create the same-sized Orthoimage and Elevation-map
12	Save $Ortho_{image}, Elevation_{map}, PointCloud_{list}$

Table 2. Matching quality mark and enhanced elevation

Matching quality level		Compare with Q_i $= \text{MAX}(W_i$ $\cdot \text{LowerFence}, 0.001)$				Matching quality label $C_{list}(u, v) =$ $1_{W_1 \geq Q_1} 2_{W_2 \geq Q_2} 3_{W_3 \geq Q_3} 4_{W_4 \geq Q_4}$	Enhanced elevation $Ele_{.list}(u, v) =$ $MEDIAN(D_{i W_i \geq Q_i})$
		W_1	W_2	W_3	W_4		
Strongest	–	\geq	\geq	\geq	\geq	1234	$MEDIAN(D_1, D_2, D_3, D_4)$
Strong	┐	\geq	\geq	\geq	$<$	123	$MEDIAN(D_1, D_2, D_3)$
	┌	\geq	\geq	$<$	\geq	124	$MEDIAN(D_1, D_2, D_4)$
	└	\geq	$<$	\geq	\geq	134	$MEDIAN(D_1, D_3, D_4)$
	┘	$<$	\geq	\geq	\geq	234	$MEDIAN(D_2, D_3, D_4)$
Weak	┐	\geq	\geq	$<$	$<$	12	$MEDIAN(D_1, D_2)$
	┌	\geq	$<$	\geq	$<$	13	$MEDIAN(D_1, D_3)$
	└	\geq	$<$	$<$	\geq	14	$MEDIAN(D_1, D_4)$
	┘	$<$	\geq	\geq	$<$	23	$MEDIAN(D_2, D_3)$
	┐	$<$	\geq	$<$	\geq	24	$MEDIAN(D_2, D_4)$
	┘	$<$	$<$	\geq	\geq	34	$MEDIAN(D_3, D_4)$
Weaker	┐	\geq	$<$	$<$	$<$	1	D_1
	┌	$<$	\geq	$<$	$<$	2	D_2
	└	$<$	$<$	\geq	$<$	3	D_3
	┘	$<$	$<$	$<$	\geq	4	D_4
Weakest	–	$<$	$<$	$<$	$<$	0	$MEDIAN(D_1, D_2, D_3, D_4)$

Note: W_i are the NCC scores of all matched pixels in $PixelGrid_i$; $W_i \cdot LowerFence_i = Q1 - 1.5 \times (Q3 - Q1)$, which is the lower fence of the NCC scores list in the matched $PixelGrid_i$. Any NCC score less than this lower fence is considered as an outlier; $1_{W_1 \geq Q_1}$ means the $W_1(u, v)$ is a strongly matched result in the original orthoimage pair, and 1 will be assigned to matching quality label $C_{list}(u, v)$. Similarly, 2 is for 90° rotation, 3 for 180° rotation, and 4 for 270° rotation of the orthoimage pair. $D_{1|W_1 \geq Q_1}(u, v)$ means pixel $p(u, v)$ is strongly matched in the original orthoimage pair; if not, $D_{1|W_1 < Q_1}(u, v)$ will not be used to enhance the $Ele_{.list}(u, v)$.



Fig. 4. Example of patch feature descriptors with $R = 1$. Patches in $Pixel Coordinate (u, v)$; $u5v5$, $u0v5$, $u5v0$, $u0v0$ are reference patch feature descriptors of size 3×3 pixel, which are generated from 6×6 – pixel patches by average pooling from the reference image; and $u'v'$ is target patch feature descriptor of size 3×3 pixel.

Self-Adaptive Patch Feature Descriptors

Based on the pixel-to-subpixel matching scheme (Fig. 3), the researchers developed the target patch feature descriptor $u'v' = I'_{u',v' \in (2R+1) \times (2R+1)}(u', v')$ to represent the target pixel/point in the target image (high orthoimage). The window $(2R + 1) \times (2R + 1)$ is self-adapting and depends on the previous matching result, because RR will be increased during the matching until a minimum threshold is satisfied. Similarly, elements in $g_{u^*,v^*}(u, v)$ are extended to $[u5v5, u0v5, u5v0, u0v0]$, which are reference patch feature descriptors of size $(2R + 1) \times (2R + 1)$ used to represent the reference pixel/point in its four scaling directions. Each reference patch feature descriptor is generated from a patch $I_{u,v \in [2 \times (2R+1)] \times [2 \times (2R+1)]}(u, v)$ in the reference image with the 2×2 average pooling.

As the reference patch feature descriptors have the same size as the target patch feature descriptor, the NCC score can be used for the pixel-to-subpixel matching with the following procedures:

- (1) calculate the four NCC scores between reference patch feature descriptors $u5v5$, $u0v5$, $u5v0$, $u0v0$, and target patch feature descriptor $u'v'$; (2) choose the largest NCC score as the scaling direction; and (3) calculate the matched subpixel coordinate for target pixel/point by Eq. (2c). The example in Fig. 4 has the patch size $2R + 1 = 3$ -pixel for the target patch feature descriptor and four-scaling reference patch feature descriptors, where the 3×3 – pixel target patch feature descriptor $u'v'$ is matched with the reference patch feature descriptor $u5v5$ [pooled from a 6×6 – pixel patch around the reference pixel $p(u, v)$]. Therefore, with the predefined image scaling, the patch-based NCC method is useful for matching the low–high orthoimage pair.

Virtual Elevation Model and Algorithm

The researchers developed the virtual elevation model in Fig. 5 to avoid brute-force matching all pixels in the target image for a given reference pixel. In detail, a construction site is being divided into discrete virtual elevation planes with the positive *Ele.* values above the origin plane (*Ele.* = 0, drone takeoff plane). The origin plane has distance $H/2$ to the drone's low altitude position, so a construction site point P on a virtual plane *Ele.* has distance

(3a)

$$Z = H/2 - Ele.$$

to the drone. Taking this Z expression into Eqs. (1h) and (1i) has the new expression

(3b)

$$x' = f(x, Ele., H/2) = x \frac{1 - Ele. \cdot \frac{2}{H}}{2 - Ele. \cdot \frac{2}{H}}$$

and

(3c)

$$y' = f(y, Ele., H/2) = y \frac{1 - Ele. \cdot \frac{2}{H}}{2 - Ele. \cdot \frac{2}{H}}$$

In addition, for an arbitrary baseline T , Eqs. (3b) and (3c) can be rewritten as

(3d)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \frac{\alpha}{1 + \alpha} \\ y \frac{\alpha}{1 + \alpha} \end{bmatrix} \text{ where } \alpha = \frac{(H - T) - Ele.}{T}$$

Thus, for a given reference point $p(x, y)$ and a known $H/2$, each virtual plane $Ele._i$ can generate a candidate target point

(3e)

$$p'_i(x'_i, y'_i) = f(p(x, y), Ele._i, H/2)$$

If the reference point matches with the candidate target point $p'_i(x'_i, y'_i)$ on virtual plane $Ele._i$, then the point P has the *ElevationCoordinate*($X', Y', Ele._i$) as in

(3f)

$$\begin{bmatrix} X' \\ Y' \\ Ele. \end{bmatrix} = \begin{bmatrix} X \\ -Y \\ Ele. \end{bmatrix} = \begin{bmatrix} x \cdot GSD \\ -y \cdot GSD \\ Ele. \end{bmatrix}$$

where $Ele. \in (-H/2, H/2)$.

Based on the virtual elevation model, the following virtual elevation algorithm is proposed, which inputs a reference point $p(x, y)$, and outputs a matched target point $p'_{matched}(x', y')$ and an elevation value $Ele_{matched}$ simultaneously.

Table

Virtual Elevation Algorithm ($p(x, y)$)	
1	initial $Ele_{\text{guess}}; NCC_{\text{max}} = 0; R_{\text{adj}\cdot\text{ratio}+} = 1; R_{\text{adj}\cdot\text{ratio}-} = 1; Ele_{\text{current}-} = Ele_{\text{guess}}; Ele_{\text{current}+} = Ele_{\text{guess}} + Ele_{\text{Step}}$
2	while $Ele_{\text{current}-} > -H/2$ or $DeptEle_{\text{current}+} < H/2$
3	if $Ele_{\text{current}-} > -H/2$ # Existing Lower elevation planes are not processed $p'(x', y') = f(p(x, y), Ele_{\text{current}-}, H/2)$ $NCC_{\text{current}-}, p'_{\text{current}-}(x', y') = \mathbf{NCC}(p, p', R \times R_{\text{adj}\cdot\text{ratio}-})$ # Return the best NCC score and the subpixel from Eq. (2c)
4	if $NCC_{\text{current}-} > NCC_{\text{max}}$ # Update the best match results $NCC_{\text{max}} = NCC_{\text{current}-}; Ele_{\text{matched}} = Ele_{\text{current}-}; p'_{\text{matched}}(x', y') = p'_{\text{current}-}(x', y')$ # Option: If $NCC_{\text{current}-} < \alpha NCC_{\text{max}}$ then stop matching the Lower elevation planes
5	if $Ele_{\text{current}+} < H/2$ # Existing Upper elevation planes are not processed $p'(x', y') = f(p(x, y), Ele_{\text{current}+}, H/2)$ $NCC_{\text{current}+}, p'_{\text{current}+}(x', y') = \mathbf{NCC}(p, p', R \times R_{\text{adj}\cdot\text{ratio}+})$ # Return the best NCC score and the subpixel from Eq. (2c)
6	if $NCC_{\text{current}+} > NCC_{\text{max}}$ # Update the best match results $NCC_{\text{max}} = NCC_{\text{current}+}; Ele_{\text{matched}} = Ele_{\text{current}+}; p'_{\text{matched}}(x', y') = p'_{\text{current}+}(x', y')$ # Option: If $NCC_{\text{current}+} < \alpha NCC_{\text{max}}$ then stop matching the Upper elevation planes
7	if $NCC_{\text{max}} < Threshold_{\text{low}}$ # Increase the patch size for poorly textured regions $R_{\text{adj}\cdot\text{ratio}-} += 0.2; R_{\text{adj}\cdot\text{ratio}+} += 0.2$
8	else $Ele_{\text{current}-} -= Ele_{\text{Step}}$ # Move next Lower elevation plane $Ele_{\text{current}+} += Ele_{\text{Step}}$ # Move next Upper elevation plane
9	return $p(x, y), p'_{\text{matched}}(x', y'), Ele_{\text{matched}}, NCC_{\text{max}}$

For matching a series of points in the low–high orthoimage pair, setting $Ele_{guess} = Ele_{i-1}$ or using the most-similar-texture neighboring point's Ele . would speed up the pixel-matching operation. Moreover, the while-loop starts at Ele_{guess} , and parallel computing the upper elevation plane group ($\geq Ele_{guess} + Ele_{step}$) and the lower elevation plane group ($\leq Ele_{guess}$). In addition, during the matching operation, the patch sizes are self-adjusted by parameters $R_{adj-ratio+}$ and $R_{adj-ratio-}$ in the upper group and lower group, respectively.

Pixel Grid Format and Matching Algorithm

In this work, the pixel grid format [Fig. 6(a)] was designed for sampling the image matching. In detail, the proposed pixel grid matching algorithm [Fig. 6(b)] starts at the first pixel $p(Margin, Margin)$ and goes through the low orthoimage row by row with the step $GridSize$. Then, the pixels are selected with an interval of $GridSize$, and are designed to share their elevations to neighboring pixels within patch $Ele_{map_{Matrix}}[u - GridSize/2: u + GridSize/2, v - GridSize/2: v + GridSize/2]$ for forming and updating a patchwise elevation map. The $Margin$ in each low orthoimage's edge is designed to guarantee that all selected pixels have their patch feature descriptors. For each pixel $p(u, v)$, the virtual elevation algorithm is proposed to process it and its neighbors [$p(u - s, v), p(u + s, v), p(u, v - s), p(u, v + s)$]; then, the median value of these five results is set as the final result for $p(u, v)$ [Fig. 6(b)]. The distance (s) to neighboring pixels can be adjusted from 1 to $GridSize$, where using a large s will smooth the elevation changes in the 3D reconstruction results.

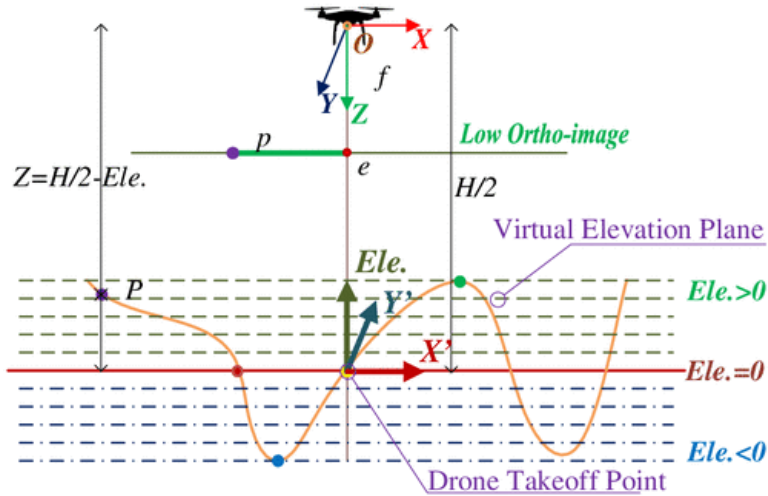


Fig. 5. Virtual elevation model.

The matched pixels and elevations are updated in $PixelGrid_{List}$ during the pixel grid matching operation, which provides Ele_{guess} for the remaining pixels. This configuration could speed up the virtual elevation algorithm to return the best matching result earlier than set $Ele_{guess} = 0$. Under this condition, the pixel matching results may be impacted by the first pixel in the beginning rows. Thus, *Lines 1* and *2* of *PGM&ED* rotate the low–high orthoimages with 90° , 180° , and 270° in a counterclockwise fashion and multiprocessing the pixel grid matching algorithm (in 4 CPU cores/threads) to produce four $PixelGrid_{List}$ results, which start from each corner of the orthoimage pair. Also, in *Line 3*, the four matching results are transformed back to the original coordinate as the four $R_PixelGrid_{List}$. In each $R_PixelGrid_i$, if a matched pixel has NCC score $W_i(u, v)$ less than 0.001

and $W_i \cdot LowerFence$ (Table 2), it is considered a weakly matched pixel; otherwise, it is a strongly matched pixel. Combining the four $R_PixelGrid_{i=1,2,3,4}$ matching results, pixels have five level matching qualities that are listed in Table 2. For example, the “strong”-level with the “123”-label means that a pixel has strongly matched results in the original, 90° rotation, and 180° rotation orthoimage pairs, and weakly matched result for 270°; then, median value $MEDIAN(D_1, D_2, D_3)$ will be assigned to its enhanced elevation Ele_{list} . Furthermore, Lines 10 and 11 generate the orthoimage and elevation map (stores elevation data as 8-bit greyscale value [0,255] in this work) via removing the *Margin* (elevation data blank) region from the low orthoimage and Ele_map_{Matrix} . Then, RGB pixels in the orthoimage and the grey pixels in the elevation map are aligned to the same coordinate [Fig. 6(a)]. Moreover, saving Ele_map_{Matrix} in a spreadsheet and customizing each cell with a float elevation value will work for an extreme accuracy demand in modeling; in addition, the 2D elevation maps are convenient to show elevation changes via the color changes and the x/y-profiles.

Parameter Configurations and Discussion

The *PGM&ED* has the recommended parameter configurations listed in Table 3, which assumes that the square-shaped orthoimages have the size of $1,824 \times 1,824$ -pixel via the image processing of resizing and cut. After running the *PGM&ED*, the output orthoimages and elevation maps have $8.6 \times 8.6 \text{ m}^2$ coverage for 10–20 orthoimage pairs and $17.9 \times 17.9 \text{ m}^2$ coverage for 20–40 orthoimage pairs.

Table 3. Algorithm parameters and configurations

Parameters	Value ($H/2 - H$)		
	10–20 m	20–40 m	Comments
Orthoimage size	1,824 × 1,824 pixels		See Fig. 8(b)
Ground sampling distance	0.5482 cm/pixel	1.0965 cm/pixel	$GSD = (H/2 \cdot h_{sensor}) / (f \cdot h_{image})$, focal length = 8.8 mm, Sensor height = 8.8 mm
Grid size	32 pixels	24 pixels	Pixel grid format [see Fig. 6(a)]
Initial patch size	$R = 19$ pixels		$R \times R_{adj\cdot ratio^*}$ is self-adaptive in Virtual elevation algorithm
	$Patch\ Size = 39 \times 39$		$PatchSize = (2R + 1) \times (2R + 1)$ (see Fig. 4)
Maximum patch size	$R \times R_{adj\cdot ratio^*} = 76$ pixels		$R_{adj\cdot ratio^*} \in [1,4]$
Margin size	128 pixels	96 pixels	$4 \times Maximum(GridSize, R)$ [see Fig. 6(a)]
Expected output size	1,568 × 1,568 pixels	1,632 × 1,632 pixels	$OrthoimageSize - 2 \times MarginSize$
Pixel number in grid	2,500	4,761	$(Expected\ Output\ Size / GridSize + 1)^2$
Covered area on site	8.596 × 8.596 m ²	17.895 × 17.895 m ²	$GSD \times Output\ Image\ Size$
Elevation range	[−5 m, 5 m]	[−10 m, 10 m]	[− $H/4$, $H/4$]
Virtual plane number	200 major/1,000 minor		Virtual elevation model format (see Fig. 5)
Major elevation step	0.05 m	0.1 m	$Ele_{\cdot step} = H/2/200$
Minor elevation step	0.01 m	0.02 m	$Ele_{\cdot step}/5$
Elevation map	$Ele_{map_{u,v}} = 255 \times \frac{Ele_{\cdot u,v} + H/4}{H/2}$		An 8-bit greyscale image, $Ele_{\cdot u,v} = H/2 \times Ele_{map_{u,v}} / 255 - H/4$
Distance to neighbor pixels	$s = GridSize/2$		$s \in [1, GridSize]$ [see Fig. 6(b)]

Image central region	$Radis = 192$ pixels		Only matching between $Ele_{guess} - Ele_{step}$ and $Ele_{guess} + Ele_{step}$
Strong-matching threshold	$Maximum(LowerFence_i, 0.001)$		see Table 2
Matched pixel quality level/mark	Mark the strongest matched pixels with green dots, strong with cyan dots, weak with blue dots, weaker with pink dots, and weakest with red dots (see Table 2)		

In the virtual elevation algorithm, $Ele_{.step}$ is a key parameter. This work set the elevation range as $[-H/4, H/4]$, which is $[-5,5]$ m for 10–20 orthoimage pairs and $[-10,10]$ m for 20–40 orthoimage pairs; then, the 8-bit greyscale elevation maps have the resolution of 0.0392 m (10 m/255) and 0.0784 m (20 m/255), respectively. For balancing the accuracy and processing time, 200 major and 1,000 minor planes are proposed in the range of $[-H/4, H/4]$; the virtual elevation algorithm searches major planes at first; if two adjacent major planes return the same NCC score, then the algorithm adjusts the $Ele_{.step}$ to $Ele_{.step}/5$ to search the five minor planes between those two major planes and return the best matching result. Moreover, this work set $Threshold_{low} = 0.4$ (Line 7), which has the recommended range of $[0.3,0.7]$; raising it can improve the matching accuracy in poorly textured orthoimage pairs but may result in errors as well. In addition, the “early stop” scheme is recommended, which stops matching the next lower or upper elevation plane if the corresponding direction occurs $NCC_{current*} < \alpha NCC_{max}$; and, the recommended setting is $\alpha = 0.7$.

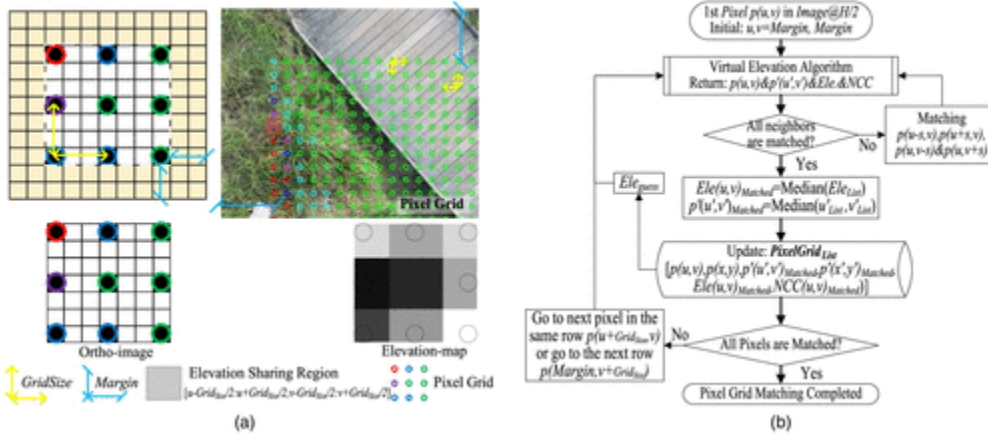


Fig. 6. (a) Pixel grid and elevation map format (image by authors); and (b) pixel grid matching algorithm.

In the pixel grid matching algorithm, $GridSize$ is a key parameter; reducing it can generate more detailed results while also raising the processing time. Testing results showed that the processing time varies for different site shapes. A relatively flat site takes less time than a site with many elevation changes via the activation of the “early stop” scheme in the virtual elevation algorithm. The recommended pixel grid settings are 2,500-pixel grid ($GridSize = 32$ pixels) for 10–20 orthoimage pairs and 4,761-pixel grid ($GridSize = 24$ pixels) for 20–40 orthoimage pairs, which need around 2–5 min for 3D reconstruction. Moreover, the researchers recommend setting the distance $s = GridSize/2$ [in Fig. 6(b)] to balance the smoothing of the elevation map and retain details of elevation changes.

Furthermore, as the drone’s shift and rotation may exist, the image center region $Radius = 192$ -pixel is an important parameter. Pixels in this region are limited to matching within $[Ele_{.guess} - Ele_{.step}, Ele_{.guess} + Ele_{.step}]$, which makes their elevation results close to their surroundings to avoid the incorrect elevation results; this is because, in Eq. (3b), $x' = x[1 - Ele \cdot (2/H)]/[2 - Ele \cdot (2/H)]$ and $y' = y[1 - Ele \cdot (2/H)]/[2 - Ele \cdot (2/H)]$ in Eq. (3c), when the reference point $p(x, y)$ is close to image center (0,0), the target point $p'(x', y')$ is close to (0,0) and becomes insensitive to Ele . changes, which may result in the virtual elevation algorithm generating the same NCC score for the different Ele . planes. This work also applied this setting for pixels near the Y -axis ($x \leq GridSize$). Since

the *PGM&ED (Line 2)* repeats the pixel grid matching algorithm four times, the noise points on both the *X – axis* and *Y – axis* are reduced. Moreover, in Table 2, the “weakest” means a pixel is weakly matched in the four $R_PixelGrid_{List}$ results; then, the Ele. of a weakest pixel will be inherited from an adjacent pixel with matching quality label ($C \geq 1$) and the most similar texture, and the updated pixel will be assigned a new matching quality label $C = 5$ to participate in enhancing the remaining pixels.

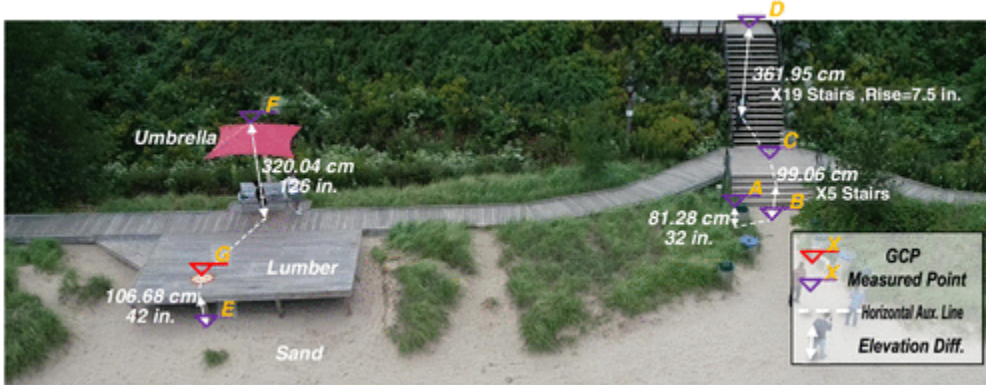


Fig. 7. Experimental site. (Image by authors.)

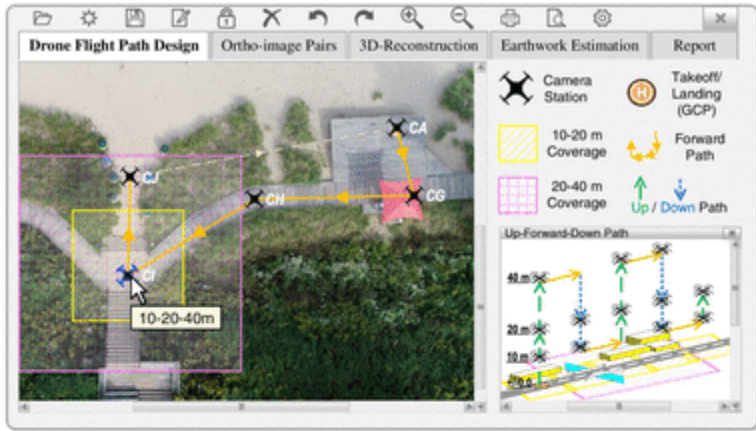
Experiment Results and Discussions

The comparative experiments were conducted and followed the workflows in Fig. 1. In detail, the developed low–high orthoimage pair-based 3D reconstruction method was programmed in Python 3.6.8 and ran on a Linux workstation (2×Intel Xeon Gold 5122 CPU@3.6 GHz with 8 cores/16 threads); and the VisualSFM ran on a Windows workstation (Intel Core i7-7800X CPU@3.5 GHz with 6 cores/12 threads). Moreover, the experimental site (Fig. 7) was selected in a park, which has all the features of natural sites, such as the ground surfaces and vegetation, before construction. The site also has most features found at excavation sites, such as the edge of the lumber platform that simulated the vertical slope, the stairs that simulated other slopes, the flat lumber surfaces that simulated the excavated flat surfaces, and the drone takeoff/landing pad that simulated the GCP. Moreover, the ground truth (GT) elevation differences between several surfaces and points were measured onsite for evaluating the developed method and Autodesk ReCap.

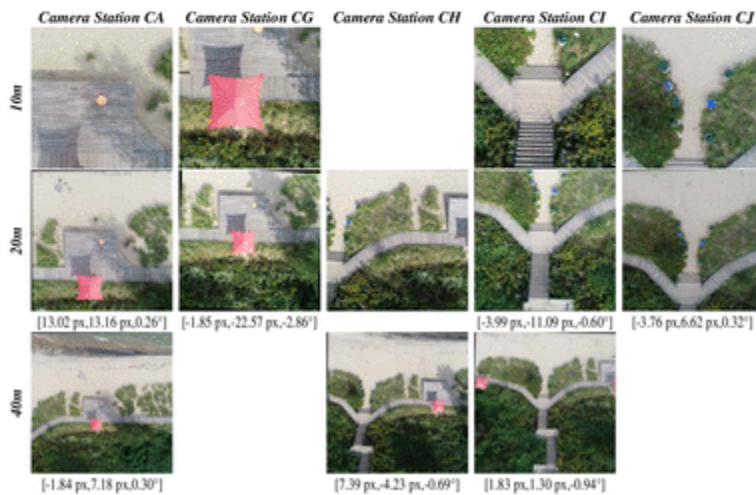
Drone Flight Path and Low–High Orthoimage Pairs Acquisition

The top-view images of target areas can be easily obtained via yielding the pitch-axis of the drone’s camera gimbal to negative 90°. The low orthoimage and its GSD determine the coverage area of each station (a low–high orthoimage pair). In the case of target areas being larger than a station’s coverage, such as a roadway project, an up–forward–down path [Fig. 8(a)] was proposed to capture a series of low–high image pairs. In detail, the drone is planning to launch and reach the desired altitudes $H/2$ and H to capture the low–high orthoimage pair at the takeoff station. After that, it moves forward to the next station, where the distance between the two stations should produce the adjacent low orthoimages having a narrow-overlapped strip for image stitching. The drone takes a high image at altitude H at the second station first, then moves downward to capture the low image at the second station when it reaches altitude $H/2$. Moreover, the drone will continue moving forward and repeat the up–forward–down steps until it acquires enough low–high orthoimage pairs to cover the

target areas or the entire construction site. This designed path will guarantee that, at each station, the drone moves vertically along its camera's principal ray, which can be implemented by a drone pilot in manual mode or by the waypoint mode in the drone controller application (such as Waypoints 2.0 in DJI GO). Besides, it is convenient to add and modify some orthoimage pairs later. Additionally, low-high orthoimage pairs can be captured with multiple altitude configurations, such as using the lower altitude image pairs for details and the higher altitude image pairs for modeling the overall shape of a construction site.



(a)



(b)

Fig. 8. (a) Drone flight path; and (b) orthoimages. (Images by authors.) Three preprocessing steps were implemented to generate these orthoimages: (1) shrink original images ($4,864 \times 3,648$ – pixel) to half size; (2) cut half-size images to a square shape ($1,824 \times 1,824$ – pixel); and (3) align high images to low images by translation and rotation (width-translation, height-translation, rotation) based on SIFT keypoints.

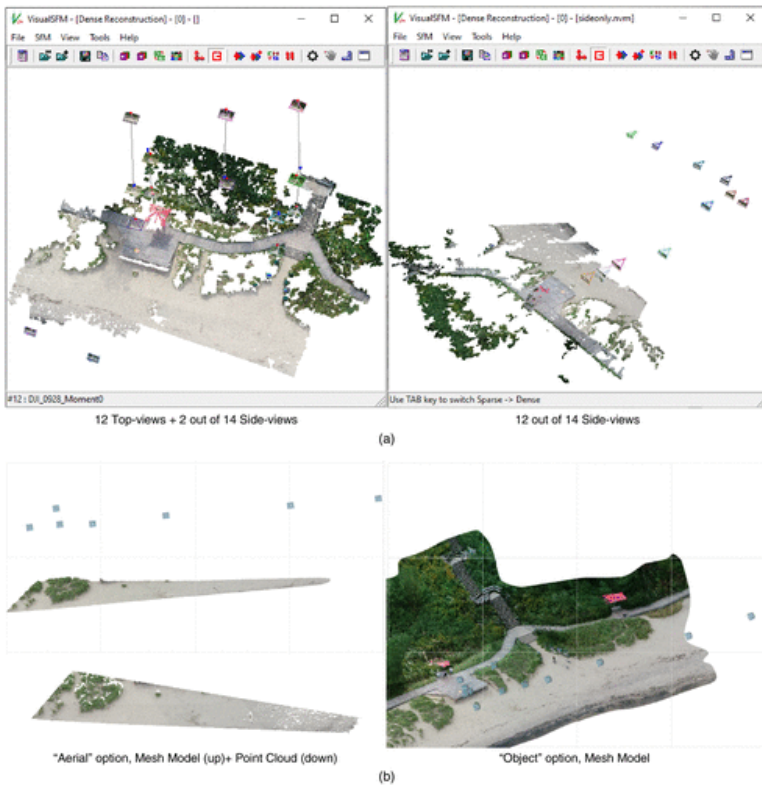


Fig. 9. (a) 3D point clouds by VisualSFM; and (b) 3D mesh models by Autodesk ReCap. (Images by authors.)

This work used a drone, the DJI Phantom 4 Pro V2.0 (SZ DJI Technology, Shenzhen, China), to capture the top-view images on the experimental site. Because the site is small, the researchers manually launched the drone at point G and flew to point C with the designed path in Fig. 8(a), and captured images at the five selected camera stations. At stations CA and CI, the drone captured images at 10-, 20-, and 40-m heights, which have flat central regions. At stations CG and CJ, images at 10- and 20-m heights were captured, which have concavo-convex central regions. At station CH, images at 20- and 40-m heights were captured, which were prepared to stitch with CA and CI. These 12 top-view images (referred to as 10-CA, captured at altitude of 10 m at the CA station) had the original size of $4,864 \times 3,648$ pixels. In the experiment, it was difficult to fly the drone along a strict vertical path; slightly horizontal shift and rotation occurred, and resulted in the low–high images having slightly different orientations and principal points e and e' . Thus, it was necessary to align the high-altitude image to the low-altitude image's center via slight image rotation and translation before assembling a low–high orthoimage pair. Also, these transformations may be required in the automated “waypoint” mode due to the impacts of wind and GPS signal interference. After the preprocessing, listed in Fig. 8(b), all the high orthoimages were rotated in the range of $[-2.86, 0.32]$ degrees and translated in the range of $[-3.99, 13.02]$ pixels in image-width-direction and $[-22.57, 13.16]$ pixels in image-height-direction; then, four 10–20 orthoimage pairs and three 20–40 orthoimage pairs were assembled, as shown in Fig. 8(b).

Photogrammetry Results and Evaluation

The VisualSFM and Autodesk ReCap were fully tested with top-view and side-view images of the experimental site. In addition to the 12 top-view images, 14 side-view images with $4,096 \times$

2,160-pixel size were extracted from a video (captured by the same drone) of the same experimental site. At first, 12 top views and 14 side views were imported into VisualSFM, then SIFT keypoints matching, sparse reconstruction, bundle adjustment, and dense reconstruction were conducted in sequence (Wu 2007, 2013; Wu et al. 2011). The 3D dense point cloud result is shown in the left of Fig. 9(a), while only two out of 14 side views assisted this 3D reconstruction. Moreover, the right of Fig. 9(a) shows a side-view-only point cloud, which was created with 12 out of 14 side views. Although the PMVS (Furukawa and Ponce 2010) was successfully applied in dense reconstructions, most vegetation and the umbrella were missing in these dense point clouds. This aligns with the conclusion presented in the literature—that photogrammetry (SfM) has limitations for poorly textured surfaces.

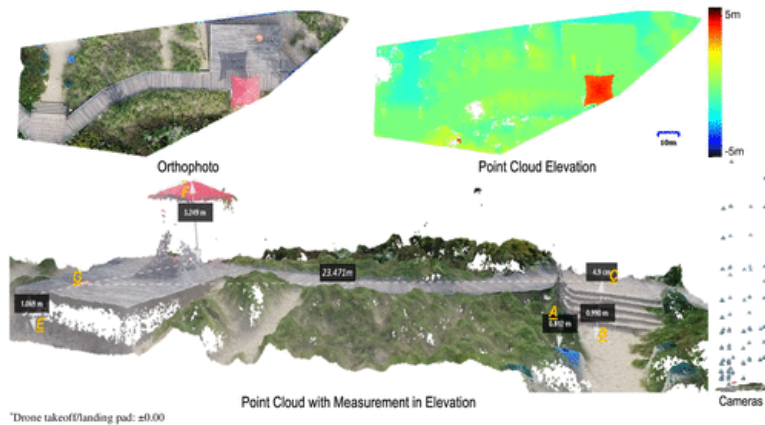


Fig. 10. Drone photogrammetry by Autodesk ReCap. (Images by authors.)

Additionally, all top views and side views were imported into Autodesk ReCap Photo and tested with its two options. In Fig. 9(b), the Aerial option results indicate that 7 out of 14 side views and no top views were used; the Object option results indicate that 12 out of 14 side views (the same as VisualSfM) were used to produce a 3D mesh model. These results show that these two options prefer the front-view images, that the camera lens should face the target object. However, flying a drone in a close distance to record objects' vertical side surfaces can increase the risk of falling and result in indirect hazards at construction sites (Susini 2015; Moud et al. 2019); thus, merging vertical side surfaces from the point cloud results of 3D laser scanning (Chen et al. 2018; Li and Lu 2018; Aguilar et al. 2019) or UGV photogrammetry (Park et al. 2019) would be preferable.

Furthermore, as the multiscale images (facing the building at various distances) enhanced the facade 3D reconstruction in Daftry et al. (2015), this work captured an additional 88 top-view images (at altitudes of 5–150 m) on the experimental site and then imported them into Autodesk ReCap Photo (allows up to 100 images). It took 54 min and 23 s for processing the 100 images and generating the 3D-mapping results in Fig. 10, which has the area coverage of about 300 m², but missed the “× 19 Stairs.” The measured elevation differences of the point cloud are marked in Fig. 10 and compared with the GTs in Table 4, which satisfied the 5.00-cm error standard (Takahashi et al. 2017). Moreover, an experiment was conducted with a higher altitude top-view subgroup (60 images, altitudes ≥ 40 m), and the results have a smaller area of coverage as 203 m². In contrast, the experimental results of the lower-altitude top-view subgroup (52 images, altitudes 10, 20, and

40 m) have a larger area of coverage as 318 m². Therefore, the low-altitude (< 40 m) images control the area of coverage in the Aerial 3D-mapping.

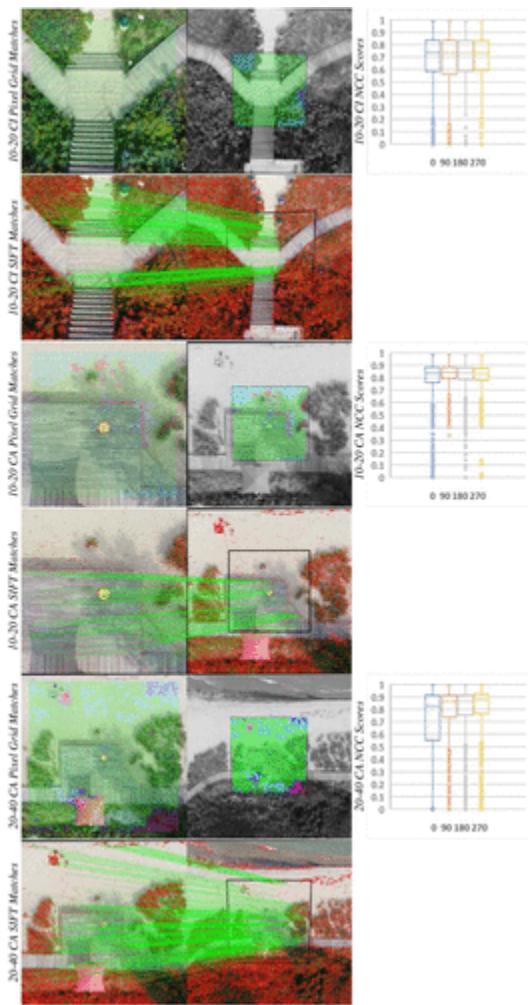


Fig. 11. Comparison of low-high orthoimage pair matches. (Images by authors.) In boxplots, “0” is the original orthoimage pair, and “90”, “180”, and “270” are the rotated orthoimage pairs.

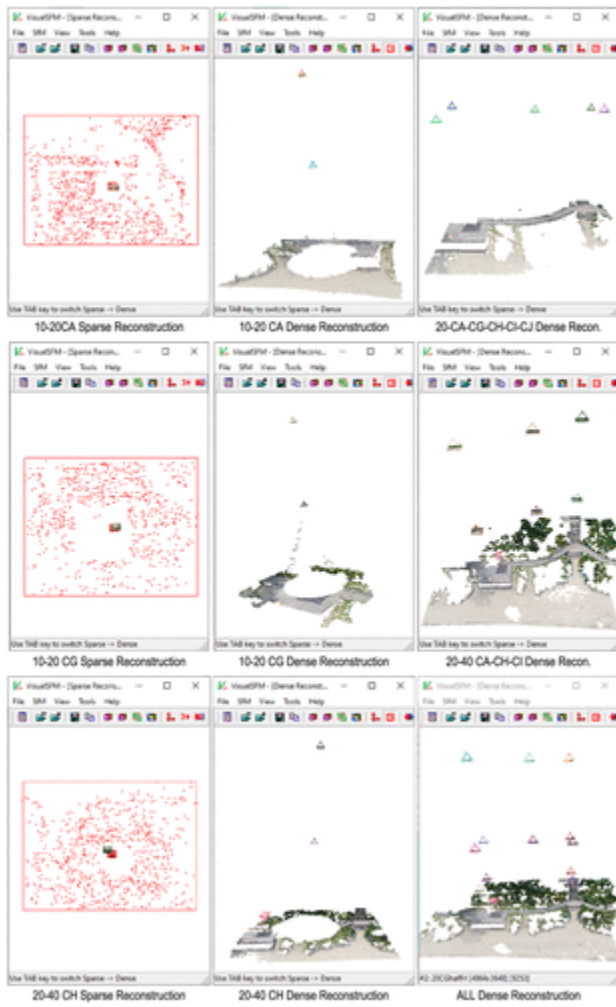


Fig. 12. Low-high orthoimage 3D reconstruction by VisualSFM.

Table 4. Elevation measurement and comparison

		Photogrammetry via Autodesk ReCap		3D reconstruction via method proposed in this paper				
Point- point	GT elevation difference (cm) (a)	Measured difference (cm) (b)	Elevation difference (cm) (b-a)	Elevation map	Elevation coordinate (m)	Measured difference (cm) (c)	Elevation difference (cm) (c-a)	Elevation error (cm)
C-A	17.78	—	—	10–20 CI	C(0.00)-A(-0.1765)	17.65	-0.13	0.13
A-B	81.28	81.20	-0.08	10–20 CJ	A(0.8039)-B(0.00)	80.39	-0.89	0.89
C-B	99.06	99.00	-0.06	10–20 CI	C(0.00)-B(-1.0000)	100.00	0.94	0.94
				20–40 CI	C(0.00)-B(-0.9804)	98.04	-1.02	1.02
D-C	361.95	—	—	20–40 CI	D(3.6471)-C(0.00)	364.71	2.76	2.76
G-E	106.68	106.90	0.22	10–20 CA	G(0.00)-E(-1.0784)	107.84	1.16	1.16
				20–40 CA	G(0.00)-E(-1.0588)	105.88	-0.80	0.8
F-G	320.04	324.90	4.86	10–20 CG	F(3.1961)-G(0.00)	319.61	-0.43	0.43
				20–40 CA	F(3.1765)-G(0.00)	317.65	-2.39	2.39

3D Reconstruction Comparison and Discussion

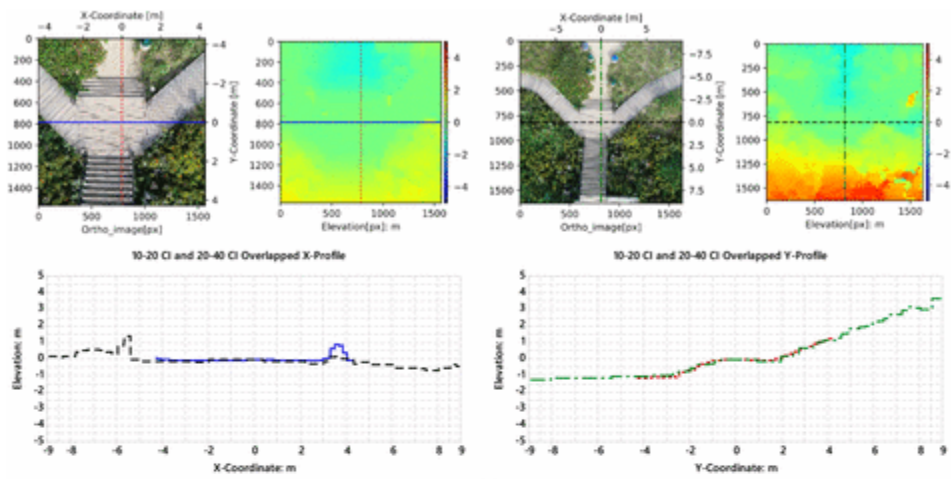
The four 10–20 orthoimage pairs and three 20–40 orthoimage pairs were all processed by the developed method within 4 min (speed range: 742.2–1,789.8 pixel/min), and the pixel grid matching results are shown in Fig. 11 and summarized in Table 5. The generated 2,500-pixel grids for 10–20 orthoimage pairs and 4,761-pixel grids for 20–40 orthoimage pairs have strongly matched pixels (which are defined as “matching qualities are strongest” and “strong-level” in Table 2, and account for [92.52%, 98.64%] of the total matched pixels in Table 5) evenly distributed throughout each low–high orthoimage pair, even in the poorly textured sand regions and dense vegetation regions. This is because the developed method can self-adjust the size of patch feature descriptors ($R \times R_{adj\cdot ratio^*}$), which uses a large patch to improve the matching results in poorly textured surfaces. Thus, the shaded regions of the umbrella on the lumber surface, and most shaded regions of the tall tree on the lumber and the sand surfaces, are well matched, which overcame the impact of environment brightness changes. In contrast, in Fig. 11, the SIFT method only matched 216, 432, and 569 sparse keypoints for 10–20 AC, 10–20 CI, and 20–40 CA orthoimages due to low contrast.

Table 5. Pixel grid matching quality

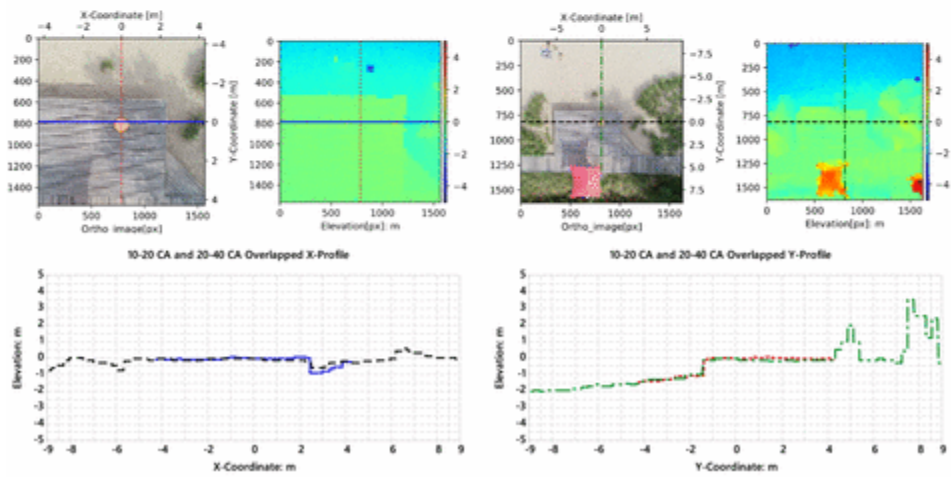
Matching quality			10– 20CA		10– 20CG		10– 20CI		10– 20CJ		20– 40CA		20– 40CH		20– 40CI	
Level	Mark	Label	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage
Strongest	Green	1234	1,960	78.4	2,037	81.48	2,141	85.64	2,254	90.16	3,855	80.97	4,288	90.07	4,108	86.28
Strong	Cyan	234	287	11.48	124	4.96	156	6.24	11	0.44	240	5.04	277	5.82	6	0.13
		134	13	0.52	28	1.12	147	5.88	87	3.48	93	1.95	51	1.07	57	1.2
		124	54	2.16	74	2.96	8	0.32	67	2.68	87	1.83	52	1.09	130	2.73
Weak	Blue	123	49	1.96	186	7.44	14	0.56	6	0.24	130	2.73	14	0.29	216	4.54
		34	2	0.08	3	0.12	28	1.12	1	0.04	67	1.41	1	0.02	29	0.61
		24	1	0.04	2	0.08	1	0.04	9	0.36	1	0.02	1	0.02	1	0.02
Weaker	Pink	23	1	0.04	16	0.64	—	—	4	0.16	1	0.02	1	0.02	7	0.15
		14	7	0.28	8	0.32	—	—	14	0.56	33	0.69	21	0.44	49	1.03
		13	4	0.16	2	0.08	—	—	1	0.04	10	0.21	2	0.04	10	0.21
		12	14	0.56	14	0.56	—	—	1	0.04	62	1.3	13	0.27	5	0.11
		4	7	0.28	1	0.04	—	—	5	0.2	—	—	10	0.21	111	2.33
Weakest	Red	3	5	0.2	2	0.08	5	0.2	20	0.8	—	—	2	0.04	5	0.11
		2	—	—	2	0.08	—	—	3	0.12	—	—	3	0.06	1	0.02
		1	19	0.76	1	0.04	—	—	—	—	151	3.17	11	0.23	6	0.13
Total matches			2,500		2,500		2,500	—	2,500	—	4,761	—	4,761	—	4,761	—
Time (min)			1.93	—	2.75	—	1.86	—	3.37	—	3.18	—	2.66	—	3.00	—
Speed (pixel/min/pixel/min)			1,295.3	—	909.1	—	1,342.9	—	742.2	—	1,496.4	—	1,789.8	—	1,588.8	—
Strongly matches (strongest and strong)			2,363	94.52	2,449	97.96	2,466	98.64	2,425	97.00	4,405	92.52	4,682	98.34	4,517	94.88
versus SIFT matches			216	—	473	—	432	—	325	—	569	—	1,324	—	1,002	—

In Fig. 11, the NCC score boxplots indicate that more than 75% of the matched pixels have NCC scores better than the $Threshold_{low} = 0.4$, and the NCC scores are different in each orthoimage pair and different in the four rotations. The reason has been discussed in the Method Development section, as the different starting corners produce different Ele_{guess} for remaining pixels [Fig. 6(b)]. In these boxplots, the outliers have a positive correlation with the weakest matched pixels, as in Fig. 11, where the 10–20 CA has the maximum weakest pixels, 77 out of 2,500 in Table 5. In addition, the correlations between the absolute value of rotation and translations [in the preprocessing; see Fig. 8(b)] with the number of weakest pixels in Table 5 were analyzed, which showed that the translation in the image-width-direction has a significant positive correlation with the number of weakest pixels. In contrast, the translation in image height and the rotation have no significant impact on the pixel grid matching quality. Because 10–20 CA had the largest shift in image width, it had worse pixel grid matching than others. Thus, minimizing the image width direction shift is important while acquiring the low–high images at each station.

Additionally, the original images ($4,864 \times 3,648$ -pixel) of the four 10–20 orthoimage pairs and three 20–40 orthoimage pairs were processed by VisualSFM with 11 trials. The 3D reconstruction results are shown in Fig. 12 and summarized in Table 6. For the dense reconstruction via PMVS (Furukawa and Ponce 2010) and 12 CPU threads (boost up to 4.00 GHz), it took 35–68 s for trials of each low–high image pair, 2.53 min for the 20–40 CA-CH-CI trial (six images), and 7.12 min for the ALL trial (12 images). For each low–high image pair trial, the processing speed was faster than the developed method (via four CPU threads; average speed 3.38 GHz). Future research would break the pixel grid into several groups and speed them up via multiprocessing in several CPU threads. However, the multiprocessing of the three 20–40 orthoimage pairs, 20–40CA, 20–40CH, and 20–40 CI, via the developed method and 12 CPU threads (3.18, 2.66, and 3.00 min) was not slower than the adjusted processing time 2.99 min ($2.53 \text{ min}/3.38 \times 4.00 \text{ GHz}$) of the 20–40 CA-CH-CI trial. Moreover, after multiprocessing the four 10–20 orthoimage pairs (the longest time is 3.37 min in Table 5), the total time is $3.37 + 3.18 = 6.55$ min, which is still faster than VisualSFM for processing the ALL trial with 12 images for 7.12 min.



(a)



(b)

Fig. 13. (a) CI station overlapped elevation results; and (b) CA station overlapped elevation results. (Image reprinted from Jiang and Bai 2020b, © ASCE.)

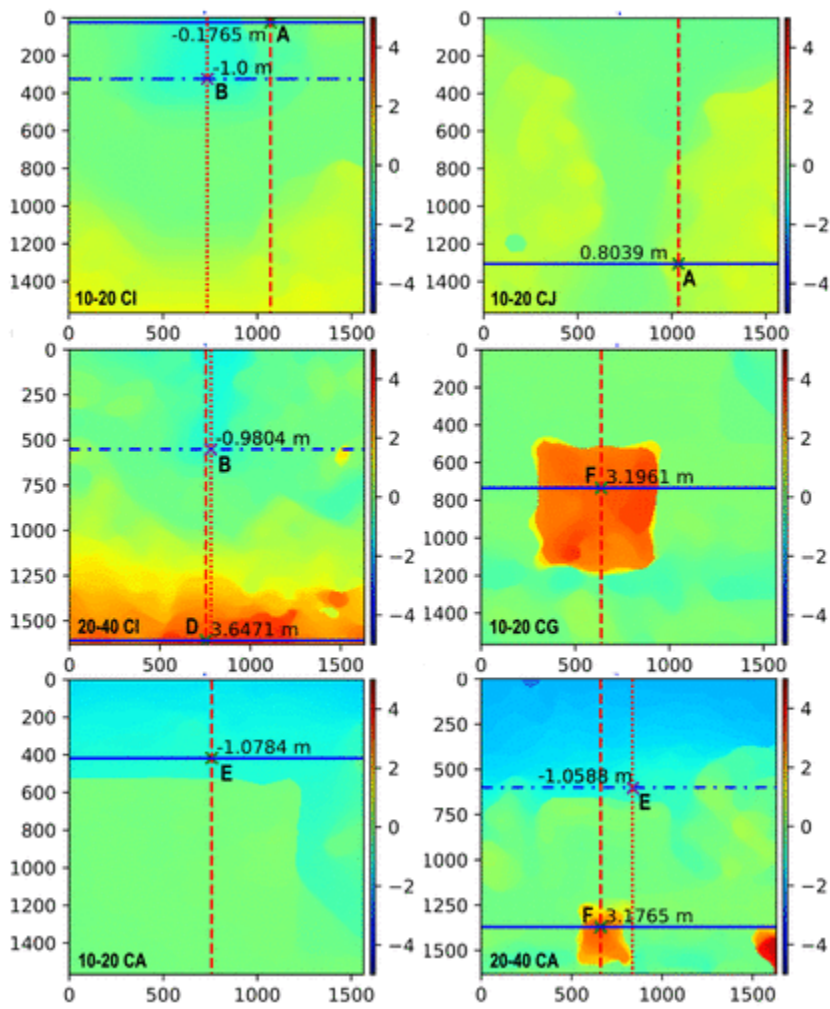


Fig. 14. Elevation measurement. 10-20 CG was aligned to the wood slab as ± 0.00 ; and others were set image center as ± 0.00 .

Table 6. SIFT matching and 3D reconstruction quality

Trial	Image		SIFT feature matching			Pixel grid matching		Sparse reconstruction		Dense reconstruction		The developed method
	Num.	Multiscala	Total matches (a)	Inlier matches (b)	Quality (b/a) (%)	Total matches	Quality (%)	Points (c)	SfM quality (c/a) (%)	Vertices	Time (min)	Time (min)
10–20 CA	2	Yes, 2: 1	1,085	1,038	95.67	2,500	94.52	1,002	92.35	151,086	1.13	1.93
10–20 CG	2	Yes, 2: 1	770	724	94.03	2,500	97.96	716	92.99	112,135	0.72	2.75
10–20 CI	2	Yes, 2: 1	668	592	88.62	2,500	98.64	538	80.54	105,279	0.77	1.86
10–20 CJ	2	Yes, 2: 1	577	538	93.24	2,500	97.00	511	88.56	87,711	0.72	3.37
20–40 CA	2	Yes, 2: 1	697	666	95.55	4,761	92.52	612	87.80	153,005	0.80	3.18
20–40 CH	2	Yes, 2: 1	922	868	94.14	4,761	98.34	800	86.77	131,462	0.77	2.66
20–40 CI	2	Yes, 2: 1	727	689	94.77	4,761	94.88	652	89.68	72,888	0.58	3.00
40-CA-CH-CI	3	No	—	—	—	—	—	1,271	—	179,625	1.05	—
20-CA-CG-CH-CI-CJ	5	No	—	—	—	—	—	1,331	—	211,147	1.43	—
20–40 CA-CH-CI	6	Yes, 2: 1	—	—	—	—	—	1,836	—	370,537	2.53	3.18^a
ALL	12	Yes, 2: 1	—	—	—	—	—	3,609	—	840,399	7.12	6.55^b

Note: Bold numbers highlight comparisons between the two “Total Matches” columns, the two “Quality” columns, and the two “Time, min” columns.

^a Maximum of 3.18, 2.66 and 3.00.

^b 3.18 plus maximum of 1.93, 2.75, 1.86, and 3.37.

In Table 6, [88.62%, 95.67%] of the SIFT matches are the inlier matches (good matches) (OpenCV 2020), and only [80.54%, 92.99%] of the SIFT matches were used in the sparse reconstruction of 511–1,002 points. These good matching quality ratios and numbers are worse than the developed method, and the 3D reconstructed sparse points are much less than the pixels of each low–high orthoimage pair, which required the further process of dense reconstruction via PMVS. However, the dense reconstruction produced 72,888–153,005 vertices for the seven trials of low–high image pairs, but they are missing the central part, vegetation, and the umbrella (Fig. 12). For the same-scale trials of 40-CA-CH-CI and 20-CA-CG-CH-CI-CJ, the issue of the missing central part was fixed, but the vegetation and umbrella were still missing. The trials of 20–40 CA-CH-CI and ALL fixed the issue to some extent, which verified that multiscale images can enhance 3D reconstruction quality (Daftry et al. 2015).

Furthermore, for the developed method, the weakest matched pixels account for only [0.00%, 3.08%], which primarily occurred on the regions of singular trees. These singular trees were also missed in the photogrammetric point cloud [Figs. 9(a), 10, and 12]. This is the same finding as that in the literature because, as singular trees on top-view images are their side surfaces, the photogrammetry performs poorly there as the trees' heights are suddenly raised from their surroundings and create vertical surfaces. Other weakest matched pixel pairs occur on the ground next to the upper-right corner of the lumber platform in 10–20 CA (Fig. 11), because the lumber surface and the sand surface have low contrast texture caused by the shade of the nearby tall tree. The sparse reconstruction results in Fig. 12 indicated that the state-of-the-art SIFT matches were worse at the upper-right corner of the lumber platform as well, and the subsequent dense reconstruction via PMVS did not fix it. Therefore, repeating the developed pixel grid matching algorithm from four starting corners of the square-shaped orthoimage (*Line 2 of PGM&ED*) has better image matching and 3D reconstruction performance for low–high orthoimages.

Elevation Comparison and Discussion

The determined elevations (converted from elevation maps, and set center as elevation ± 0.00) are shown in Figs. 13 and 14, which show that the developed method is valid in flat central regions such as the CI station shown in Fig. 13(a) and CA station shown in Fig. 13(b); it works perfectly in the concavo-convex central regions, such as CG and CJ stations shown in Fig. 14; moreover, the new method can handle steep and near-vertical topography such as the vertical side of the garbage can in CI and CJ station, the edge of the lumber platform in CA station, the umbrella in CA and CG stations, and the stairways in CI stations. In addition, the overlapped X/Y-Profile of 10–20 orthoimage pairs and 20–40 orthoimage pairs at stations CI and CA are matched at most parts in Figs. 13(a and b). Given that the 20–40 orthoimage pairs' GSD and Ele_{step} are twice that of the 10–20 orthoimage pairs (Table 3), it is reasonable to have less detailed elevation variations such as edges, salient pole, and concave pole in the higher-altitude orthoimage pairs.

Most of the weakest matched pixels were assigned with the correct elevations from the strongly matched neighboring pixels' elevations, such as in Fig. 13(b), where the 10–20 CA elevations were determined well, but in Fig. 11 pixels were weakest matched on the ground (next to the upper-right corner of the lumber platform). In contrast, with only two images, the SfM approach leaves a big hole there (Fig. 12). There are a few noise pixels in the generated elevation maps, such as the dark small patch in 10–20 CA [Fig. 13(b)]. Due to the setting $Threshold_{low} = 0.4$, two selected pixels in this

shaded and poorly textured region were matched on the wrong elevation planes with a “weak” matching quality level, and because of their matching quality label $C \geq 1$, they were not being replaced, but shared their wrong elevations to the adjacent two “weakest” matched pixels. To address this issue, the researchers proposed a median filter with size $(4GridSize + 1) \times (4GridSize + 1)$ to remove these noise points, while keeping the edges of elevation changes (Fig. 14).

The measured elevation coordinates in Fig. 14 are compared with the GTs in Table 4, which have elevation differences $[-0.89, 1.16]$ cm for the 10–20 orthoimage pairs, and $[-2.39, 2.76]$ cm for the 20–40 orthoimage pairs. Both satisfy the 5.00-cm error standard (Takahashi et al. 2017) and have the same accuracy as the photogrammetric dense point cloud in Fig. 10, which has the measurement differences in $[-0.08, 4.86]$ cm. Moreover, *Comparison 1* in Table 7 shows two out of nine measured virtual planes fall in the lower interval $[-Ele_{step}, 0]$, and seven out of nine fall in the upper interval $[0, Ele_{step}]$. Also, they were all matched within the expected discrete virtual elevation plane range based on the GTs. In Fig. 14, the measured elevations via the 8-bit elevation maps are multiples of 0.0392 m for 10–20 orthoimage pairs and multiples of 0.0784 m for 20–40 orthoimage pairs, because each determined elevation is rounded to an integral number in $[0, 255]$ for generating the 8-bit elevation maps; as a result, the 8-bit elevation maps have systematic errors of 0.0196 m for 10–20 orthoimage pairs and 0.0392 m for 20–40 orthoimage pairs. Then, in Table 7, for any *Measured Virtual Plane* that falls in the lower interval, the *Elevation Coordinate* minus the *Error* obtains the *Virtual Plane with Error*; otherwise, obtain it via *Elevation Coordinate* plus the *Error*. After amplifying the errors, *Comparison 2* still has the same results as *Comparison 1*, which all fall in the designed range of virtual elevation planes. Thus, the developed virtual elevation model and algorithm are robust at different camera stations with different altitudes, demonstrating better performance than photogrammetry (SfM) in the poorly textured region with few images.

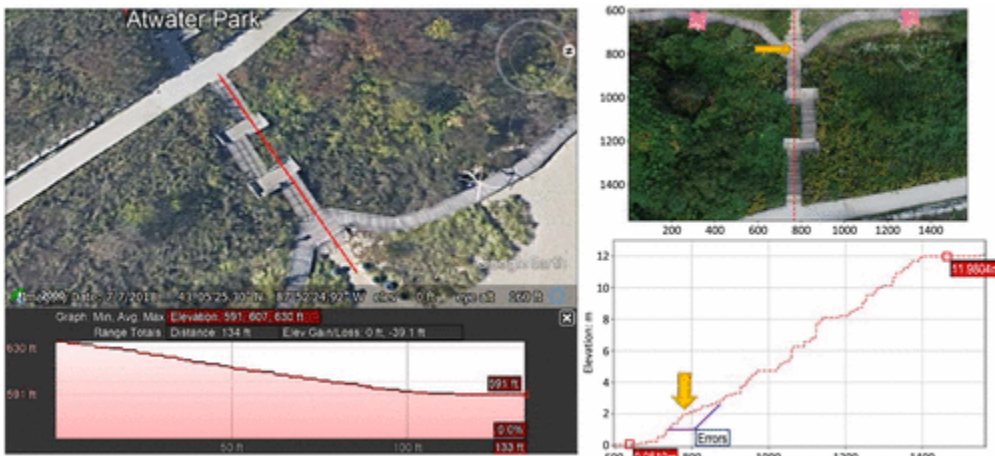


Fig. 15. Elevation comparison with Google Earth Pro. (Map data © 2020 Google.)

Table 7. Virtual elevation analysis

Point	Elevation map	Experimental results			Discrete virtual plane based on GTs					Comparison 1		Systematic error adjustment			Comparison 2	
		Elevation step (cm)	Elevation coordinate (m)	Measured virtual plane	GT Elevation coordinate (m)	Elevation step	Lower plane	GT virtual plane	Upper plane	Fall in the lower interval?	Fall in the upper interval?	Elevation map error (m)	Elevation \pm error (m)	Adj. virtual plane	Fall in the lower interval?	Fall in the upper interval?
		(a)	(b)	(c = b/a)	(d)	(e = d/a)	(f)	(g)	(h)	(f < c < g)	(g \leq c < h)	(i)	(j = b \pm i)	(k = j / a)	(f < k < g)	(g \leq k < h)
A	10–20 CI	5	-0.1765	-3.5	-0.1778	-3.6	-5	-4	-3	—	Yes	0.0196	-0.1569	-3.1	—	Yes
	10–20 CJ	5	0.8039	16.1	0.8128	16.3	15	16	17	—	Yes	0.0196	0.8235	16.5	—	Yes
B	10–20 CI	5	-1.0000	-20.0	-0.9906	-19.8	-21	-20	-19	—	Yes	0.0196	-0.9804	-19.6	—	Yes
	20–40 CI	10	-0.9804	-9.8	-0.9906	-9.9	-11	-10	-9	—	Yes	0.0392	-0.9412	-9.4	—	Yes
D	20–40 CI	10	3.6471	36.5	3.6195	36.2	35	36	37	—	Yes	0.0392	3.6863	36.9	—	Yes
E	10–20 CA	5	-1.0784	-21.6	-1.0668	-21.3	-23	-22	-21	—	Yes	0.0196	-1.0588	-21.2	—	Yes
	20–40 CA	10	-1.0588	-10.6	-1.0668	-10.7	-12	-11	-10	—	Yes	0.0392	-1.0196	-10.2	—	Yes
F	10–20 CG	5	3.1961	63.9	3.2004	64.0	63	64	65	Yes	—	0.0196	3.1765	63.5	Yes	—
	20–40 CA	10	3.1765	31.8	3.2004	32.0	31	32	33	Yes	—	0.0392	3.1373	31.4	Yes	—

Furthermore, the setting of *PGM&ED* can be customized to fit the shape of the site with a narrower elevation range than $[-H/4, H/4]$, and set a constant Ele_{step} to conduct rapid 3D reconstruction. Fig. 15 shows the elevation results of 70–140 orthoimage pairs at station CI, which used 3.23 min with the setting of $Ele_{step} = 0.05$ m, elevation range $[-13, 13]$ m, $GridSize = 32$ -pixel, and the $Threshold_{low} = 0.2$. Although results showed that errors in the image center region (marked with arrows in Fig. 15) were not successfully fixed, the measured elevation difference via the developed method (11.9294 m) is very close to the 11.8872 m (39 ft) via Google Earth Pro. Therefore, coarse elevation results determined via high-altitude orthoimage pairs are useful in modeling the overall shapes of a target site, and its detailed elevations would be obtained via stitching multiple low-altitude results.

Results Stitching, Alignment, and Measurement

The stitching of 2D images is much easier than the merging of 3D point clouds when a low-orthoimage is unable to cover the entire target area. The steps for stitching include: (1) pick up two common points/objects in the adjacent orthoimages to determine the scale relation for scaling orthoimages and elevation maps; (2) pick up another two common points as the common boundary for the adjacent orthoimages, and rotate them (and elevation maps) to make the boundary in horizontal (up–down stitching) or vertical (left–right stitching); and (3) stitch orthoimages and align elevation maps on this common boundary. Fig. 16 shows three stitched results of the seven collected low–high orthoimage pairs, and the 3D dense point clouds [generated using each center pixel of the 8×8 -pixel patch from the stitched orthoimages and elevation maps by Eq. (3f)]. Because only top-view images were used in 3D reconstruction, these point clouds missed vertical side surfaces similar to the photogrammetric point cloud in Fig. 10, but the missed side points have no impact on elevation determinations, as the overall shape of the experimental site and the top surface of the small objects (e.g., garbage cans) were well 3D reconstructed. The point clouds show errors on the boundary of the umbrella, where pixels on 20–40 CA were weakly matched, and there are no SIFT matches in this poorly textured and shaded region (Fig. 11). In addition, the dense point clouds of VisualSFM in Figs. 9(a) and 12, and Autodesk ReCap in Fig. 10, also have errors here, showing either as holes or determined incorrect elevations. However, the developed method can reduce this issue via the following approaches. First, use 10–20 orthoimage pair for more detailed 3D reconstruction, e.g., the 10–20 CG in Fig. 16. Second, decrease $GridSize$ and Ele_{step} for dense matching more pixels and smoothing virtual elevation plane changes. The captured low-altitude images may have some side surfaces of small objects [e.g., garbage can in Fig. 8(b)] due to the reflected rays being converged through the camera lens instead of passing in parallel into the lens. Enlarging the altitude or flying the drone over these objects can eliminate this issue, and the horizontal position of a point on the vertical side surfaces can be calibrated by Eq. (1a) if necessary, or by removing them from elevation maps for ground elevation determination via the method in Jiang et al. (2020).

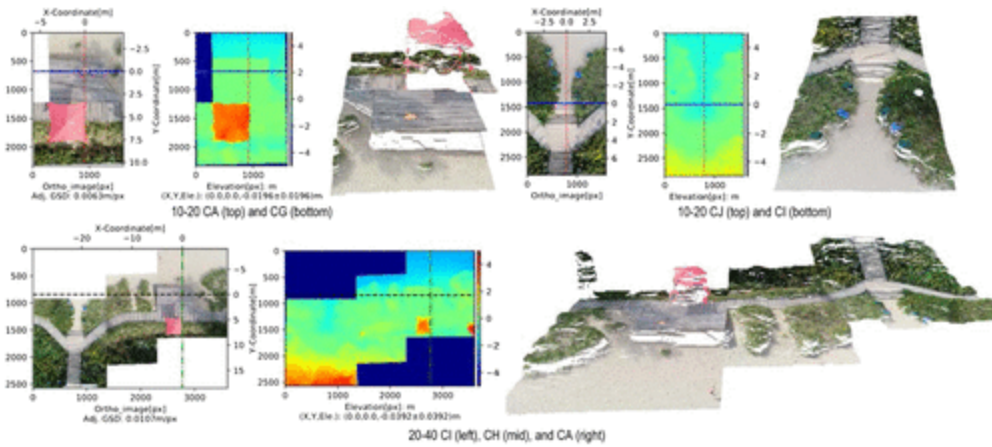


Fig. 16. Stitched orthoimages, elevation maps, and point clouds. (Images by authors.) Point clouds were extracted from stitched orthoimages (texture) and elevation maps using the center of each 8×8 – pixel patch.

In Fig. 16, the stitched orthoimages of 10–20 CA-CG and CJ-CI show that they have slightly different scales because camera positions were not always at the exact altitudes of 10 and 20 m to the drone takeoff/landing pad (Fig. 12). This indicates that the vertical distance between the drone’s low and high positions and the distance of low position to the ground are not strict to the designed $H/2$; thus, the determined Elevation needs an elevation alignment based on a GCP. This work identified the drone takeoff/landing pad as the GCP, and aligned the elevation map by setting ± 0.00 at the GCP via pixelwise binary segmentation by a well-trained U-net (Ronneberger et al. 2015). For the stitched 10–20 CA-CG and 20–40 CI-CH-CA orthoimage inputs, the U-net predictions have 11,032 and 3,838 pixels belonging to GCP with intersections over union (IoUs) of 0.9557 and 0.9553, respectively. Because the landing pad’s diameter is 75 cm, the accurate GSD can calculate via $GSD_{adj.} = 75 / (2\sqrt{Pixel_{GCP}/PI})$, where $Pixel_{GCP}$ is pixel number of the GCP (pad) in U-net predictions. Then, the stitched 10–20 CA-CG and 20–40 CI-CH-CA have the $GSD_{adj.}$ of 0.6328 cm/pixel and 1.0729 cm/pixel, which are slightly different from the GSDs in Table 3. Moreover, *ElevationCoordinate* was converted via $GSD_{adj.}$ and set the origin at the pad’s center. Since systematic errors exist in 8-bit elevation maps, the measured GCPs in Fig. 16 are slightly different from ± 0.00 , but after eliminating systematic errors, they are exactly ± 0.00 .

Furthermore, Fig. 17 shows the point C has the *ElevationCoordinate* ($-21.2395, 9.9938, -0.0392 \pm 0.0392$) m, the horizontal distance of 23.4732 m and elevation difference of 0.0392 ± 0.0392 m between C and G are both similar to the 23.471 m and 4.9 cm that were measured from the photogrammetric point cloud in Fig. 10. Thus, the developed method has a high accuracy in 3D reconstruction using only three pairs of 20–40 orthoimages. Moreover, for earthwork estimation, this work set each pixel as the unit base, which has the unit area of $GSD_{adj.} \times GSD_{adj.}$, and then multiplied the unit area by the elevation differences of each pixel between the design elevation and the current elevation, to sum the volume estimations. In Fig. 17, earthwork plans “f1” and “c1” were drawn in the orthoimage by pointing out objects via cursor and typing designed elevations via keyboard; then, the cut/fill volume was automatically computed based on $GSD_{adj.} \times GSD_{adj.}$, and elevation differences between Current Elevation and Cut_Fill Design in pixelwise were multiplied.

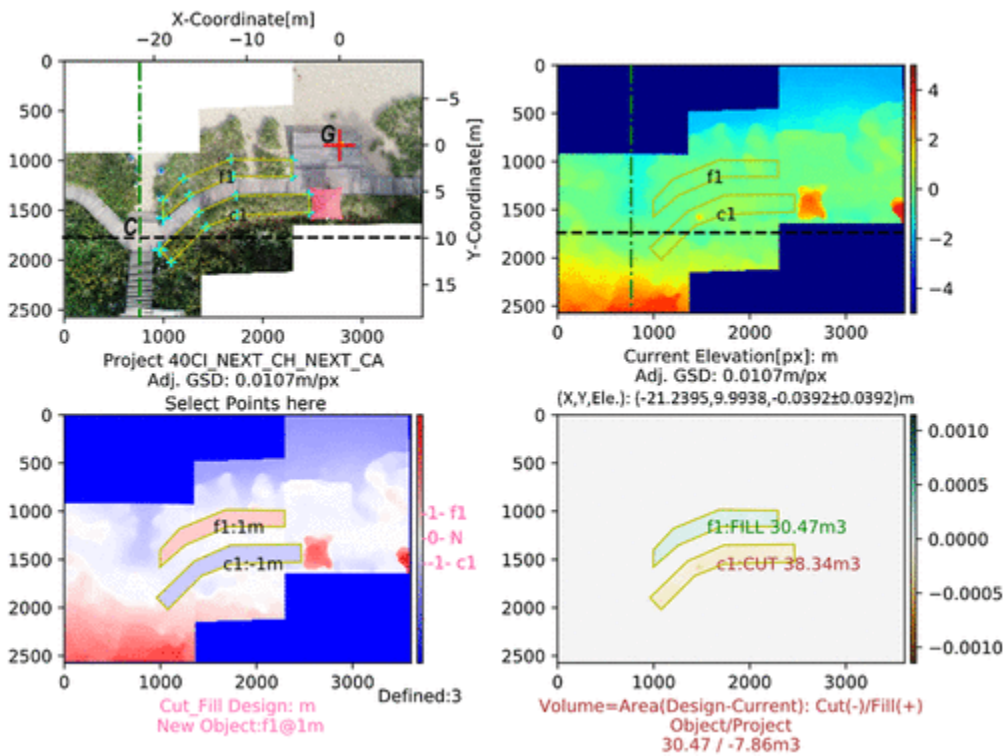


Fig. 17. Earthwork estimation. (Image by author.)

Conclusions and Recommendations

This paper presents an effective, rapid, and easily implementable image-based 3D reconstruction method for automatic determination of construction site elevations using drone technology, summarized in Fig. 1, and the achievements of the new method are listed in Table 8. The input images are small-sized quadcopter drone-based low–high orthoimage pairs instead of same-scale overlapping images. For each low–high orthoimage pair, the 3D reconstruction is based on the developed pixel grid matching and elevation determination algorithm. An up–forward–down flight path [Fig. 8(a)] was designed to capture adjacent low–high orthoimage pairs for enlarging the coverage of the developed 3D reconstruction method. The stitching operation requires a very narrow overlapping strip compared with the high overlapping ratio needed in photogrammetry; thus, the experimental site only needs three 20–40 orthoimage pairs to cover (Fig. 16). The processing time of these 20–40 orthoimage pairs is about 3 min via Python 3.6.8 and multiprocessing with 12 CPU threads, which is as fast as processing the same six images via VisualSFM + PMVS and 12 CPU threads. Speed can be further increased by programming the developed algorithms in C/C++, and breaking a pixel grid into several groups and multiprocessing them in several CPU threads. Moreover, the 3D reconstruction quality is as accurate as photogrammetry via Autodesk ReCap, which took 55 min to process the uploaded 100 images on its cloud server. In addition, the alignment of stitched elevation maps was efficiently conducted via U-net detected GCP (drone landing pad, IoU of 0.955). Then the earthwork estimation was easily done, as demonstrated in Fig. 17.

Table 8. Achievement in this work

Performance	Limitations in existing methods	Fulfilled in this work	Detailed achievement and comparison
Image numbers and coverage	Photogrammetry requests high ratio overlapping images	Two images for one target station, and results stitching with narrow overlaps	The stitched results of three pairs of 20–40 orthoimages have the same coverage as 100 multiscale images via Autodesk ReCap
Image matching quality	SIFT matches are irregular and missed in regions of low contrast and variation, missed in edges	Self-adaptive patch feature descriptors for customized matching	The pixel grid matches are densely and uniformly distributed in images; see the detailed comparison of low–high image pairs matching results in Fig. 11
3D reconstruction efficiency	Separate sparse reconstruction and dense reconstruction, which require a powerful workstation	Combined image matching, sparse and dense reconstruction in one operation	The pixel grid matching and elevation determination algorithm matches the pixel grid and determines pixels' elevations simultaneously; this work conducted experiments to compare the 3D reconstruction processing time and quality of the developed method versus VisualSFM (Tables 5 and 6)
Image type and 3D reconstruction quality	Same-scale top-view images (contain top surfaces) and side-view images (contain top and side surface)	2: 1-scale top-view images only	The point clouds cover the entire surface of the experimental site (Fig. 16), while the same-scale top-view trial in Fig. 12 leaves blanks in the vegetation and umbrella regions. Moreover, this work conducted experiments to prove that using multiscale images can improve photogrammetry in the poorly textured region (Fig. 12)
Flight altitude	Close range (about 5 m) in Visual SLAM	Low altitude at 10 or 20 m for detailed 3D reconstruction, and can be extended to 70 m for coarse elevations	The developed method obtains depth information based on a low–high orthoimage pair triangulation, which has an adjustable baseline (distance between cameras); while the Visual SLAM obtains depth information via the classical stereo-vision model, which has a fixed baseline and results in limited sensing range
Alignment	Requires precise GPS, or at least three GCPs	Automatically detects the drone takeoff and landing pad as GCP	A U-net model was used to detect a drone landing pad, calculate GSD for orthoimages and elevation map, and update the elevation coordinate origin to the pad's center

Accuracy	Elevation within 5 cm5 cm error (photogrammetry)	Elevation within 5 cm5 cm error	Elevation measurement is conducted in elevation maps (Fig. 14) and compared in Table 4
Modeling and measurement	3D point cloud and mesh model	Elevation map and 3D point cloud	Volume estimation based on elevation difference in each pixel and each pixel's coverage ($GSD \times GSD$)
Safety and efficiency	Over construction sites via drone, move toward target objects using state-of-the-practice surveying techniques	Always over and away from construction sites	The developed method has the advantage of data (image) acquisition requiring less time, without interrupting other construction operations, which is suitable for obtaining as-built elevations and monitoring construction progress

In this work, the patch feature descriptors and NCC scores were used for automatically matching the approximately 2: 1-scale low–high orthoimage pairs with the pixel grid matching and elevation determination algorithm. The developed method was robust for poorly textured surfaces and large sloped surfaces. It provided an accurate pixel grid match for the low–high orthoimage pair at least 92% of the time (Table 5). It also gave an accurate elevation result for the strongly matched pixel grid within the acceptable elevation error of less than 5.00 cm (Table 4). Furthermore, Fig. 2 illustrates the proposed method’s recovery 3D information via the low–high ($H/2$ - H) orthoimage pair triangulation model with the special baseline $H/2$ (vertical distance between the drone’s two positions). For an arbitrary baseline T , the model can be rewritten as Eq. (1j); then, the developed virtual elevation model in Fig. 5 can be rewritten as Eq. (3d) as well. The developed four-scaling patch feature descriptors may have poor performance in matching low–high orthoimage pairs that do not have the approximately 2: 1-scale relation (not at $H/2$ and H). However, making two same-dimension patch feature descriptors and matching them via NCC score still works. Future work can consider using a fully convolutional network (FCN) to generate the patch feature descriptor uv^{FCN} for the selected pixel $p(u, v)$ in the low orthoimage. For example, if the low (reference) image is at 60 m, and the high (target) image is at 80 m, which is a 4 : 3 scale relation (3 : 4 in GSD), the proposed FCN model should be able to output a 30×30 -pixel patch as uv^{FCN} for an input 40×40 -pixel patch; then, the uv^{FCN} can be used to match $u'v'$ [30×30 -pixel target patch feature descriptor for a candidate target pixel $p'(u', v')$] by Eq. (3d)].

Data Availability Statement

The images are available from the corresponding author upon reasonable request. The Python codes of the pixel grid matching and elevation determination algorithm are available from the corresponding author upon reasonable request.

Acknowledgments

This work was financially supported by the McShane Endowment fund at Marquette University. The authors are thankful for the reviewers’ valuable comments.

References

- Aguilar, R., M. F. Noel, and L. F. Ramos. 2019. “Integration of reverse engineering and non-linear numerical analysis for the seismic assessment of historical adobe buildings.” *Autom. Constr.* 98 (Feb): 1–15. <https://doi.org/10.1016/j.autcon.2018.11.010>.
- Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. 2008. “Speeded-up robust features (SURF).” *Comput. Vis. Image Underst.* 110 (3): 346–359. <https://doi.org/10.1016/j.cviu.2007.09.014>.
- Chen, K., W. Lu, F. Xue, P. Tang, and L. H. Li. 2018. “Automatic building information model reconstruction in high-density urban areas: Augmenting multi-source data with architectural knowledge.” *Autom. Constr.* 93 (Sep): 22–34. <https://doi.org/10.1016/j.autcon.2018.05.009>.
- Daftry, S., C. Hoppe, and H. Bischof. 2015. “Building with drones: Accurate 3D facade reconstruction using MAVs.” In *Proc., 2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 3487–3494. New York: IEEE.
- Du, J. C., and H. C. Teng. 2007. “3D laser scanning and GPS technology for landslide earthwork volume estimation.” *Autom. Constr.* 16 (5): 657–663. <https://doi.org/10.1016/j.autcon.2006.11.002>.

- Eigen, D., C. Puhrsch, and R. Fergus. 2014. "Depth map prediction from a single image using a multi-scale deep network." In Proc., 28th Conf. on Neural Information Processing Systems (NIPS 2014), NeurIPS, 2366–2374. San Diego: Neural Information Processing Systems Foundation. <https://papers.nips.cc/paper/2014/file/7bccfde7714a1ebadf06c5f4cea752c1-Paper.pdf>.
- Erickson, M. S., J. J. Bauer, and W. C. Hayes. 2013. "The accuracy of photo-based three-dimensional scanning for collision reconstruction using 123D catch." In Proc., SAE 2013 World Congress and Exhibition. Warrendale, PA: SAE International. <https://doi.org/10.4271/2013-01-0784>.
- Freimuth, H., and M. König. 2018. "Planning and executing construction inspections with unmanned aerial vehicles." *Autom. Constr.* 96 (Dec): 540–553. <https://doi.org/10.1016/j.autcon.2018.10.016>.
- Furukawa, Y., and J. Ponce. 2010. "Accurate, dense, and robust Multiview stereopsis." *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (8): 1362–1376. <https://doi.org/10.1109/TPAMI.2009.161>.
- Gwak, H. S., J. Seo, and D. E. Lee. 2018. "Optimal cut-fill pairing and sequencing method in earthwork operation." *Autom. Constr.* 87 (Mar): 60–73. <https://doi.org/10.1016/j.autcon.2017.12.010>.
- Han, K., J. Degol, and M. Golparvar-Fard. 2018. "Geometry- and appearance-based reasoning of construction progress monitoring." *J. Constr. Eng. Manage.* 144 (1): 04017110. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001428](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001428).
- Han, K. K., and M. Golparvar-Fard. 2017. "Potential of big visual data and building information modeling for construction performance analytics: An exploratory study." *Autom. Constr.* 73 (Jan): 184–198. <https://doi.org/10.1016/j.autcon.2016.11.004>.
- Haur, C. J., L. S. Kuo, C. P. Fu, Y. L. Hsu, and C. Da Heng. 2018. "Feasibility study on UAV-assisted construction surplus soil tracking control and management technique." In Proc., 5th Annual Int. Conf. on Material Science and Environmental Engineering (MSEE2017). Xiamen, China: IOP Publishing. <https://doi.org/10.1088/1757-899X/301/1/012145>.
- Hola, B., and K. Schabowicz. 2010. "Estimation of earthworks execution time cost by means of artificial neural networks." *Autom. Constr.* 19 (5): 570–579. <https://doi.org/10.1016/j.autcon.2010.02.004>.
- Intel. 2016. "Intel RealSense camera R200." Accessed August 24, 2020. <https://ark.intel.com/content/www/us/en/ark/products/92256/intel-realsense-camera-r200.html>.
- Intel. 2020a. "Intel RealSense LiDAR camera L515." Accessed August 24, 2020. <https://www.intelrealsense.com/lidar-camera-l515/>.
- Intel. 2020b. "Intel RealSense depth camera D435." Accessed August 24, 2020. <https://www.intelrealsense.com/depth-camera-d455/>.
- Inzerillo, L., G. Di Mino, and R. Roberts. 2018. "Image-based 3D reconstruction using traditional and UAV datasets for analysis of road pavement distress." *Autom. Constr.* 96 (Dec): 457–469. <https://doi.org/10.1016/j.autcon.2018.10.010>.
- Jiang, Y., and Y. Bai. 2020a. "Determination of construction site elevations using drone technology." In Proc., Construction Research Congress 2020. Reston, VA: ASCE.
- Jiang, Y., and Y. Bai. 2020b. "Estimation of construction site elevations using drone-based orthoimagery and deep learning." *J. Constr. Eng. Manage.* 146 (8): 04020086. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0001869](https://doi.org/10.1061/(ASCE)CO.1943-7862.0001869).
- Jiang, Y., Y. Bai, and S. Han. 2020. "Determining ground elevations covered by vegetation on construction sites using drone-based orthoimage and convolutional neural network." *J. Comput. Civ. Eng.* 34 (6): 04020049. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000930](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000930).

- Kaehler, A., and G. Bradski. 2016. *Learning OpenCV 3: Computer vision in C++ with the OpenCV library*. Sebastopol, CA: O'Reilly Media.
- Kim, D., M. Liu, S. Lee, and V. R. Kamat. 2019. "Remote proximity monitoring between mobile construction resources using camera-mounted UAVs." *Autom. Constr.* 99 (Mar): 168–182. <https://doi.org/10.1016/j.autcon.2018.12.014>.
- Kim, H., and H. Kim. 2018. "3D reconstruction of a concrete mixer truck for training object detectors." *Autom. Constr.* 88 (Apr): 23–30. <https://doi.org/10.1016/j.autcon.2017.12.034>.
- Kim, K., H. Kim, and H. Kim. 2017. "Image-based construction hazard avoidance system using augmented reality in wearable device." *Autom. Constr.* 83 (Nov): 390–403. <https://doi.org/10.1016/j.autcon.2017.06.014>.
- Kwon, S., J. W. Park, D. Moon, S. Jung, and H. Park. 2017. "Smart merging method for hybrid point cloud data using UAV and LIDAR in earthwork construction." *Procedia Eng.* 196: 21–28. <https://doi.org/10.1016/j.proeng.2017.07.168>.
- Lewis, J. P. 1995. "Fast template matching." In *Proc., Vision Interface 95*, 120–123. Quebec City, Canada: Canadian Image Processing and Pattern Recognition Society.
- Li, D., and M. Lu. 2018. "Integrating geometric models, site images and GIS based on Google Earth and Keyhole Markup Language." *Autom. Constr.* 89 (May): 317–331. <https://doi.org/10.1016/j.autcon.2018.02.002>.
- Li, R., F. Ma, F. Xu, L. H. Matthies, C. F. Olson, and R. E. Arvidson. 2002. "Localization of Mars rovers using descent and surface-based image data." *J. Geophys. Res. Planets* 107 (E11): FIDO-4. <https://doi.org/10.1029/2000JE001443>.
- Lowe, D. G. 2004. "Distinctive image features from scale-invariant keypoints." *Int. J. Comput. Vis.* 60 (2): 91–110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>.
- Maghiar, M., and D. Mesta. 2018. "Measurement comparison of city roadway intersection models obtained via laser-scanning and photogrammetry." In *Proc., 54th ASC Annual Int. Conf.*, 568–576. Minneapolis, MN: Application Support Center.
- Matthies, L., M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova. 2007. "Computer vision on Mars." *Int. J. Comput. Vis.* 75 (1): 67–92. <https://doi.org/10.1007/s11263-007-0046-z>.
- Matthies, L. H., C. F. Olson, G. Tharp, and S. Laubach. 1997. "Visual localization methods for Mars rovers using lander, rover, and descent imagery." Accessed May 8, 2018. <https://trs.jpl.nasa.gov/bitstream/handle/2014/22227/97-0695.pdf?sequence=1>.
- Meng, C., N. Zhou, X. Xue, and Y. Jia. 2013. "Homography-based depth recovery with descent images." *Mach. Vis. Appl.* 24 (5): 1093–1106. <https://doi.org/10.1007/s00138-013-0498-9>.
- Metni, N., and T. Hamel. 2007. "A UAV for bridge inspection: Visual servoing control law with orientation limits." *Autom. Constr.* 17 (1): 3–10. <https://doi.org/10.1016/j.autcon.2006.12.010>.
- Moon, D., S. Chung, S. Kwon, J. Seo, and J. Shin. 2019. "Comparison and utilization of point cloud generated from photogrammetry and laser scanning: 3D world model for smart heavy equipment planning." *Autom. Constr.* 98 (Feb): 322–331. <https://doi.org/10.1016/j.autcon.2018.07.020>.
- Morgenthal, G., N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, and V. Rodehorst. 2019. "Framework for automated UAS-based structural condition assessment of bridges." *Autom. Constr.* 97 (Jan): 77–95. <https://doi.org/10.1016/j.autcon.2018.10.006>.
- Moud, H. I., X. Zhang, I. Flood, A. Shojaei, Y. Zhang, and C. Capano. 2019. "Qualitative and quantitative risk analysis of unmanned aerial vehicle flights on construction job sites: A case study." *Int. J. Adv. Intell. Syst.* 12 (3 and 4): 135–146.

- Nassar, K., and Y. Jung. 2012. "Structure-from-motion approach to the reconstruction of surfaces for earthwork planning." *J. Constr. Eng. Project Manage.* 2 (3): 1–7. <https://doi.org/10.6106/JCEPM.201.2.3.001>.
- Nichols, H., and D. Day. 2010. *Moving the earth: The workbook of excavation*. 6th ed. New York: McGraw-Hill.
- Omar, T., and M. L. Nehdi. 2017. "Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography." *Autom. Constr.* 83 (Nov): 360–371. <https://doi.org/10.1016/j.autcon.2017.06.024>.
- OpenCV. 2020. "Feature matching + homography to find objects." Accessed October 31, 2020. https://docs.opencv.org/3.4.12/d1/de0/tutorial_py_feature_homography.html.
- Park, J., P. Kim, Y. K. Cho, and J. Kang. 2019. "Framework for automated registration of UAV and UGV point clouds using local features in images." *Autom. Constr.* 98 (Feb): 175–182. <https://doi.org/10.1016/j.autcon.2018.11.024>.
- Remondino, F. 2003. "From point cloud to surface: The modeling and visualization problem." In *Proc., Int. Workshop on Visualization and Animation of Reality-Based 3D Models*. Engadin, Switzerland: International Society for Photogrammetry and Remote Sensing.
- Roca, D., S. Lagüela, L. Díaz-Vilariño, J. Armesto, and P. Arias. 2013. "Low-cost aerial unit for outdoor inspection of building façades." *Autom. Constr.* 36 (Dec): 128–135. <https://doi.org/10.1016/j.autcon.2013.08.020>.
- Ronneberger, O., P. Fischer, and T. Brox. 2015. "U-Net: Convolutional networks for biomedical image segmentation." In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, 234–241. Berlin: Springer.
- Seo, J., L. Duque, and J. Wacker. 2018. "Drone-enabled bridge inspection methodology and application." *Autom. Constr.* 94 (Oct): 112–126. <https://doi.org/10.1016/j.autcon.2018.06.006>.
- Seo, J., S. Lee, J. Kim, and S. K. Kim. 2011. "Task planner design for an automated excavation system." *Autom. Constr.* 20 (7): 954–966. <https://doi.org/10.1016/j.autcon.2011.03.013>.
- Shang, Z., and Z. Shen. 2018. "Real-time 3D reconstruction on construction site using visual SLAM and UAV." In *Proc., Construction Research Congress 2018*. Reston, VA: ASCE.
- Shewchuk, J. R. 2002. "Delaunay refinement algorithms for triangular mesh generation." *Comput. Geom.* 22 (1–3): 21–74. [https://doi.org/10.1016/S0925-7721\(01\)00047-5](https://doi.org/10.1016/S0925-7721(01)00047-5).
- Shirowzhan, S., and S. M. E. Sepasgozar. 2019. "Spatial analysis using temporal point clouds in advanced GIS: Methods for ground elevation extraction in slant areas and building classifications." *ISPRS Int. J. Geo-Inf.* 8 (3): 120. <https://doi.org/10.3390/ijgi8030120>.
- Siebert, S., and J. Teizer. 2014. "Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system." *Autom. Constr.* 41 (May): 1–14. <https://doi.org/10.1016/j.autcon.2014.01.004>.
- Snavely, N. 2010. "Bundler: Structure from motion (SfM) for unordered image collections." Accessed May 8, 2018. <http://www.cs.cornell.edu/~snavely/bundler/#S1>.
- Solem, J. E. 2012. *Programming computer vision with Python: Tools and algorithms for analyzing images*. Sebastopol, CA: O'Reilly Media.
- Sung, C., and P. Y. Kim. 2016. "3D terrain reconstruction of construction sites using a stereo camera." *Autom. Constr.* 64 (Apr): 65–77. <https://doi.org/10.1016/j.autcon.2015.12.022>.
- Susini, A. 2015. "A technocritical review of drones crash risk probabilistic consequences and its societal acceptance." In *Vol. 7 of Proc., RIMMA Risk Information Management, Risk Models, and Applications*, 27–38. Berlin: RISK Information Management, Risk Models and Applications.

- Takahashi, N., R. Wakutsu, T. Kato, T. Wakaizumi, T. Ooishi, and R. Matsuoka. 2017. "Experiment on UAV photogrammetry and terrestrial laser scanning for ICT-integrated construction." In Proc., 2017 Int. Conf. on Unmanned Aerial Vehicles in Geomatics, 371–377. Bonn, Germany: International Society for Photogrammetry and Remote Sensing.
- Wang, J., S. Zhang, and J. Teizer. 2015. "Geotechnical and safety protective equipment planning using range point cloud data and rule checking in building information modeling." *Autom. Constr.* 49 (Jan): 250–261. <https://doi.org/10.1016/j.autcon.2014.09.002>.
- Wang, L., F. Chen, and H. Yin. 2016. "Detecting and tracking vehicles in traffic by unmanned aerial vehicles." *Autom. Constr.* 72 (Dec): 294–308. <https://doi.org/10.1016/j.autcon.2016.05.008>.
- Westoby, M. J., J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 2012. "'Structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications." *Geomorphology* 179 (Dec): 300–314. <https://doi.org/10.1016/j.geomorph.2012.08.021>.
- Wu, C. 2007. "SiftGPU: A GPU implementation of scale invariant feature transform (SIFT)." Accessed Apr. 4, 2020. <http://cs.unc.edu/~ccwu/siftgpu>.
- Wu, C. 2013. "VisualSFM: A visual structure from motion system." Accessed April 4, 2020. <http://ccwu.me/vsfm/index.html>.
- Wu, C., S. Agarwal, B. Curless, and S. M. Seitz. 2011. "Multicore bundle adjustment." In Proc., CVPR 2011, 3057–3064. New York: IEEE.
- Xiong, Y., C. F. Olson, and L. H. Matthies. 2005. "Computing depth maps from descent images." *Mach. Vis. Appl.* 16 (3): 139–147. <https://doi.org/10.1007/s00138-004-0160-7>.
- Yang, C., M. Tsai, S. Kang, and C. Hung. 2018. "UAV path planning method for digital terrain model reconstruction—A debris fan example." *Autom. Constr.* 93 (Sep): 214–230. <https://doi.org/10.1016/j.autcon.2018.05.024>.
- Yi, C., and M. Lu. 2016. "A mixed-integer linear programming approach for temporary haul road design in rough-grading projects." *Autom. Constr.* 71 (Nov): 314–324. <https://doi.org/10.1016/j.autcon.2016.08.022>.
- Zhang, S., J. Teizer, N. Pradhananga, and C. M. Eastman. 2015. "Workforce location tracking to model, visualize and analyze workspace requirements in building information models for construction safety planning." *Autom. Constr.* 60 (Dec): 74–86. <https://doi.org/10.1016/j.autcon.2015.09.009>.
- Zhao, W. Q., and Z. Lin. 2016. "SfM precise surface measurement: Evaluation of resolution and accuracy and error analysis." [In Chinese.] *Geogr. Geo-Inf. Sci.* 32 (6): 25–31.
- Zhong, X., X. Peng, S. Yan, M. Shen, and Y. Zhai. 2018. "Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles." *Autom. Constr.* 89 (May): 49–57. <https://doi.org/10.1016/j.autcon.2018.01.005>.