

Article

Towards Automated Measurement of As-Built Components Using Computer Vision

Husein Perez *  and Joseph H. M. Tah 

Oxford Institute for Sustainable Development, School of the Built Environment, Oxford Brookes University, Oxford OX3 0BP, UK; jtah@brookes.ac.uk

* Correspondence: hperez@brookes.ac.uk

Abstract: Regular inspections during construction work ensure that the completed work aligns with the plans and specifications and that it is within the planned time and budget. This requires frequent physical site observations to independently measure and verify the completion percentage of the construction progress performed over periods of time. The current computer vision techniques for measuring as-built elements predominantly employ three-dimensional laser scanning or three-dimensional photogrammetry modeling to ascertain the geometric properties of as-built elements on construction sites. Both techniques require data acquisition from several positions and angles to generate sufficient information about the element's coordinates, making the deployment of these techniques on dynamic construction project sites challenging. This paper proposes a pipeline for automating the measurement of as-built components using artificial intelligence and computer vision techniques. The pipeline requires a single image obtained with a stereo camera system to measure the sizes of selected objects or as-built components. The results in this work were demonstrated by measuring the sizes of concrete walls and columns. The novelty of this work is attributed to the use of a single image and a single target for developing a fully automated computer vision-based method for measuring any given object. The proposed solution is suitable for use in measuring the sizes of as-built components in built assets. It has the potential to be further developed and integrated with building information modelling applications for use on construction projects for progress monitoring.



Citation: Perez, H.; Tah, J.H.M.

Towards Automated Measurement of As-Built Components Using Computer Vision. *Sensors* **2023**, *23*, 7110. <https://doi.org/10.3390/s23167110>

Academic Editors: Xin Ning and Wenfa Li

Received: 28 July 2023

Revised: 7 August 2023

Accepted: 9 August 2023

Published: 11 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: machine learning; computer vision; automated measurement

1. Introduction

The accurate and up-to-date measurement of as-built components is essential for the design, construction, operation, and maintenance of as-built assets. Additionally, the measurement of as-built components is an essential part of construction project management functions, such as cost and schedule controls, financial reporting, claims, and productivity measurement. It is, therefore, considered to be one of the most crucial, yet challenging tasks facing site managers. Most current approaches are still predominately manual, time-consuming, and error-prone. Site managers normally spend a significant amount of time measuring, recording, and analysing as-built information [1–3]. The lack of accurate and up-to-date as-built information due to laborious and manual data collection practices could lead to increased costs, delays, and poor project performance, which in turn, could reduce the ability to detect or manage the variability and uncertainty inherent in the project's activities [4–6].

In recent years, however, the construction industry has been exploring various emerging technologies to support the visual inspection and progress monitoring of construction work [7]. The on-site application of these technologies has indeed demonstrated significant potential for digitising and automating the capturing, measuring, and reporting updates of the as-built components and project information [6,8].

One notable example of these technological tools for automating the measurement of as-built components is the use of computer vision (CV). CV is a digitisation process used for determining project progress that combines computer science, architecture, construction engineering, and management disciplines. It takes visual media, such as photos, videos, or scans as inputs and produces decisions or other forms of representation as outputs [9].

The two most popular CV-based techniques for measuring the as-built components are laser-based scanning, and imaging-based photogrammetry [4,8,10,11]. Three-dimensional (3D) laser scanning is used to generate 3D point clouds that are processed to enable the estimation of sizes and quantities of as-built components [12]. The imaging-based approach, on the other hand, emulates human visualisation to extract three-dimensional (3D) geometrical information of objects from two-dimensional (2D) inputs [13–15].

A fully automated CV-based method for measuring as-built components consists of four main sub-processes; data acquisition, information retrieval and processing, measurement estimation, and producing valuable output [16–19]. The sub-processes involve different techniques to achieve the desired outputs with their own benefits and limitations [8,20].

Despite some studies having made significant strides in automating CV-based methods for as-built component measurements, at present, there are still no applications that are fully automated. This is due to two main reasons. Firstly, the technologies involved are still emerging and undergoing experimentation with only a few functional demonstrations available [21]. Secondly, existing studies do not typically address the four stages together but focus on the individual stages, such as 3D point cloud generation in data acquisition [22,23], and feature recognition in information retrieval and processing [24–26].

This study aims to address this research gap by developing a pipeline for a fully automated as-built component measurement approach using CV-based methods. The proposed pipeline can run in real-time and is intended to estimate the size of as-built components of built assets. The pipeline employs stereo camera techniques for data acquisition, machine learning, object detection, instance segmentation for information retrieval and the processing of as-built elements, Green's theorem [27] for the measurement estimation of the size of the object(s) under consideration, and visualisation of the output as labelled images. To demonstrate the work, a neural network model was trained on concrete walls and columns, but the same principles can be extended to cover other types of as-built components.

The rest of this paper is structured as follows: Section 2 provides a literature review of related previous studies, highlighting the existing research and theories relevant to the subject matter. Section 3 focuses on the instrumentation and materials used in this study, outlining the experimental setup and tools employed for data collection. We delve into the methodology of geometric estimation, explaining the mathematical models and algorithms utilised to estimate the geometric properties of the as-built components of interest. Section 4 presents the results obtained from the experiments, including a detailed analysis and discussion. Finally, Section 5 summarises the key findings and conclusions drawn from this study, highlighting the implications, significance, and limitations of this research.

2. Literature Review

The measurement of as-built components is crucial in the design, construction, operation, and maintenance of built assets. It plays a vital role in continuously monitoring and periodically updating the actual work conducted on a construction site, comparing it with the as-planned or anticipated progress [28–30]. Identifying variations between the planned and actual progress is essential for schedule updating [31]. The most common CV-based method for undertaking spatial measurements of the actual work on construction sites is 3D laser scanning. During the process, the construction site is scanned from different angles and locations at different times to generate spatial data, which can then be used to estimate the quantities of work performed within the time interval considered between two successive scans. A 3D laser scanning yields data in the form of 3D points, known

as “point clouds”, which are later displayed as images that can be viewed from different perspectives using specialised software systems [32,33]. Many researchers have proposed and demonstrated that the technology can be used to obtain 3D data on the actual progress of a project efficiently [31,34–37].

However, this method has certain limitations as 3D data can be obtained only on the as-built components that are located within the laser scanner’s range and field of view. Secondly, even components that are physically within the range of the scanner may still be blocked from view by various pieces of equipment and other obstacles located around the construction site, resulting in an incomplete 3D data set obtained on a construction site. To overcome this problem, researchers have proposed UAV-based 3D laser scanning methods [38–46]. The authors argued that this approach can provide visual and detailed progress information with good area coverage and views from human-inaccessible angles. UAV-based data acquisition, however, requires careful operation handling as it can pose potential safety hazards and cause distractions to workers on-site. UAVs also require accurate path planning to avoid obstruction, which in the case of any sudden rotational motion or sharp angular movements can result in motion blur. They can also be affected by wind speeds and other environmental anomalies.

In addition to issues related to the acquisition approach, there are other limitations associated with 3D laser scanning. This includes the time required to perform a single scan, and the number of scan positions necessary to acquire accurate information. The technique is also costly, technically intricate, and requires skilled experts to capture and model the whole project. Moreover, the collected 3D point cloud also requires extensive time and computational resources to process data and produce meaningful interpretations, which may not be adequate for use in complex project sites to generate real-time updates [20]. The incomplete or partially occluded patches in a 3D point cloud will also incur technical challenges during the registration of multiple point clouds [23,47–49].

On the other hand, the availability of high-quality and precise still image cameras has advanced 3D modelling from photo images [50]. As a result, an image-based scanning method called photogrammetry has been proposed as an alternative to 3D laser scanning [51]. With photogrammetry, the geometrical properties of an object on site are generated from its photo image. The technique, however, requires strategically placing many targets on the object(s) being photographed to identify the object’s coordinates, and several photos of the object are then taken from different positions and angles to generate sufficient information on object coordinates [52]. The use of image-based scanning may also incur other practical limitations, particularly when extracting geometrical properties of surfaces with little texture or poor definition [53]. Additionally, a recent study has also shown that the accuracy of the model generated from the image-based reconstruction is less than the laser scanner and becomes even less accurate as the length of the element increases. According to the study, the process of reconstructing a 3D model from an image dataset remains reliant on human intervention at various steps to improve the output quality [54].

Compared to existing photogrammetry techniques that require the placement of many targets on the object and several photos taken from different positions and angles to generate sufficient geometrical information, the proposed method requires a single image obtained from a stereo camera system and a single target to extract the information about the object’s coordinates. Additionally, unlike 3D laser-based scanning, which also requires the construction site to be scanned from different locations and generates computationally extensive 3D point clouds, the proposed pipeline is capable of generating real-time updates of as-built components on construction sites.

3. Instrumentation

In this section, we discuss the pipeline for the full CV-based method, which was developed for measuring as-built components. To demonstrate the results, images containing concrete walls and columns captured from buildings at Oxford Brookes University, Headington campus, was used to apply the pipeline to estimate the sizes and areas of concrete

elements. The pipeline, which is depicted in Figure 1, can run in real-time, and it comprises seven steps: camera calibration, scene capturing, calculating the distance to a point in the scene, instance segmentation, depth map generation, estimating world coordinates, and finally calculating the area of the object of interest. This section will be divided into four subsections: data acquisition; information retrieval and processing; as-built component measurements; and finally visualisation of the output.

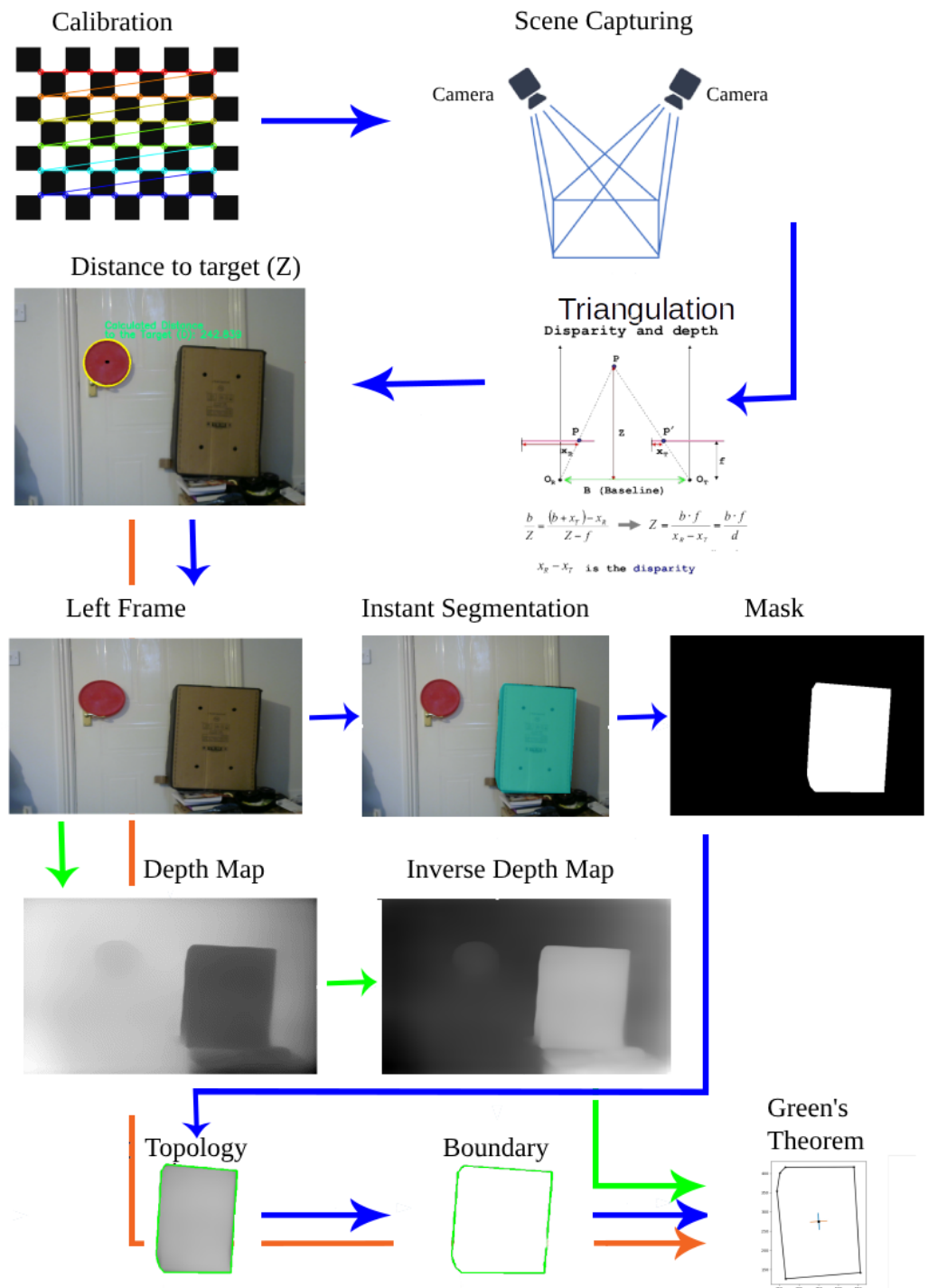


Figure 1. The proposed pipeline showing all sub-stages: stereo camera calibration, scene capturing, object segmentation, generation of the absolute depth map, extraction of the boundary of the object, and finally area estimation.

3.1. Initialisation

Camera calibration: a stereo camera system was used to capture the scene containing the object of interest. It is important for the camera system to be accurately calibrated. The calibration is a one-time process used for determining the intrinsic (principle point, distortion parameters, and focal length) and the extrinsic parameters (rotation and translation) of each camera and the relative poses between them. These parameter sets are essential for attaining 3D information given a set of 2D coordinates of corresponding image points [55]. The process of recovering the third missing dimension is an ill-posed problem and is known in image geometry applications as depth estimation [56,57].

There are many different techniques used for approximating the intrinsic and extrinsic parameters for a specific camera model. The most common one is Zhang's method [58], (the one adopted in this work) and the direct linear transformation (DLT) [59].

Zhang's method uses multiple views of a 3D pattern of a known structure but an unknown position and orientation in space. It is a flexible technique for camera calibration and well-suited for use without specialised knowledge of 3D geometry or computer vision. The technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. During the calibration process, both the camera and the planar pattern can be freely moved, but motion does not need to be known [58,60].

DLT, on the other hand, is a mathematical approach that aims to solve the problem of determining the pinhole camera parameters from at least six correspondences between 2D image points and 3D world points. A camera model maps each point of the 3D world to a point of the 2D image through a projection operation. The pinhole camera model makes the assumption that the aperture size of the camera is small so that it can be considered a point. Thus, the ray of light has to pass across a single point and the camera centre; there are no lenses, no distortion, and there is an infinite depth of field [59,61].

In their simplest form, the intrinsic parameters can be represented by a 3×3 matrix called the camera matrix, denoted by the letter K , as presented below:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where f_x, f_y are the lengths of the focal point (in pixels), which is the distance from the centre of the lens to the principal points of the lens. c_x, c_y are the principal points which are the points on the image where a ray of light travelling perpendicular to the image plane passes through the focal point of the lens and intersects with the camera's sensor.

3.2. Data Acquisition

Scene capturing: A stereo vision system, also known as binocular stereo vision, is a machine vision technique that uses exactly two cameras to capture a scene from two viewpoints. The two cameras are separated by a short distance known as the baseline b and are mounted almost parallel to one another. The principle of stereo vision is similar to that of the 3D perception of the human eyes. It can provide a 3D perception with real-time depth measurements based on the triangulation of rays from the two viewpoints (see Figure 2).

b is the baseline, f is the focal length of the camera, and u_L and u_R are the projections of the real-world point P in an image acquired by the left and right cameras. X_A and Z_A are the X-axis and the optical axis of the left camera, respectively, whereas X_B and Z_B are the X-axis and the optical axis of the right camera, respectively. P is a real-world point defined by the coordinates X, Y , and Z [62].

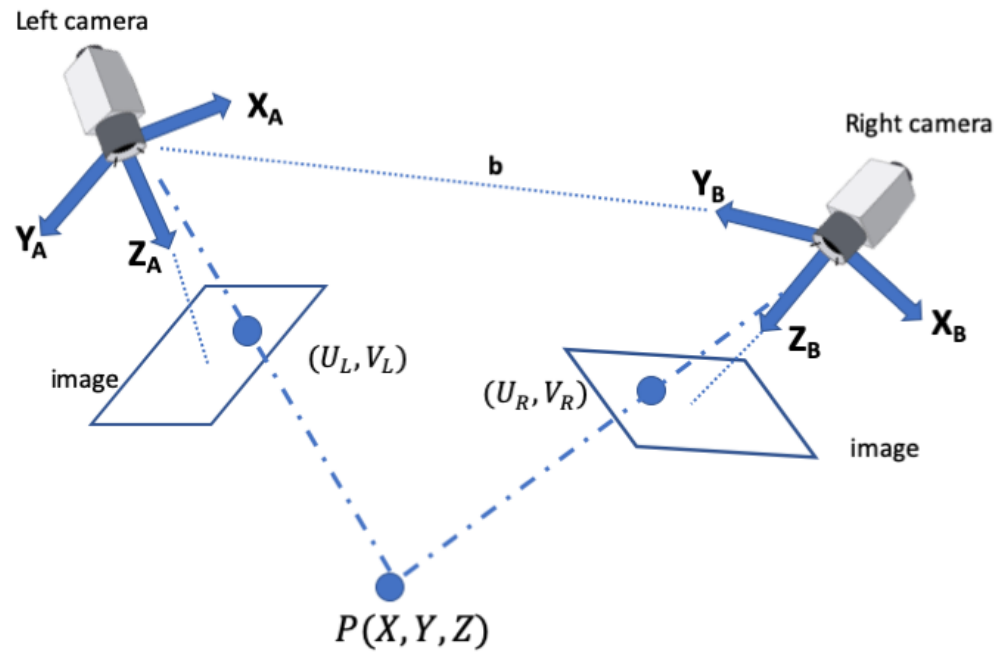


Figure 2. Typical stereo vision system: b represents the distance between the principal points, (U_L, V_L) , (U_R, V_R) are the 2D projections of the real-world point $P(X, Y, Z)$.

Calculating the distance to the target (Z): In order to calculate the depth of information Z , which is the distance to the real-world point P , we first calculate the disparity D , which is the horizontal shift in position between two corresponding points projected on the image plane in the stereo vision system. In this approach, a red circular target was used to calculate the horizontal disparity between the centres of the circles appearing in the left and right frames. It is important that both the target and the object of interest are visible in the two frames, and that both the left and the right frames have at least a 30% overlap. The calculations of the depth and disparity values are shown in (2) and (3).

$$D = u_L - u_R \quad (2)$$

$$Z = f * b / D \quad (3)$$

where b is the baseline, f is the focal length of the camera obtained from (1), u_L and u_R are the projections of the real-world point P in an image acquired by the left and right cameras [63].

When capturing a scene from two distinct viewpoints using a stereo camera system, the left and right frames are not lined up perfectly, and when the cameras rotate or move forward or backward, the pixels will also move accordingly. This makes matching the corresponding pixels in each frame a very challenging task. To simplify the subsequent stereo correspondence problem, a process called rectification is applied first (see Figure 3). Stereo rectification is the determination of two image transformations (or *homographies*) that map corresponding points on the two images and projections of the same point in the 3D space onto the same horizontal line in the transformed images [64,65].

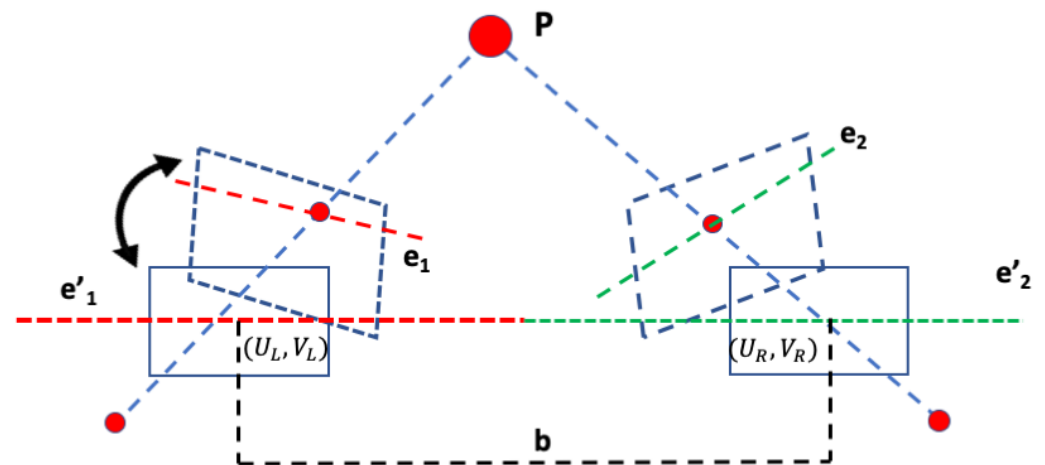


Figure 3. Illustration of image rectification. Epipolar lines e_1 , e_2 are projected on a common line (dashed red–green). The distance between principle points (U_L, V_L) and (U_R, V_R) along the common line is the baseline b [65].

The target can be placed anywhere in the scene and only needs to be visible by both cameras. Placing the target closer or on the object of interest would, however, improve accuracy. A target is any artefact object that can be distinguished from the surroundings, either by shape or colour. A red circular object was used as a target to facilitate image processing techniques to detect the (red) colour and determine the circumference of the target in both left and right frames. The circular shape allows us to easily obtain the centre of the disk in both frames. Therefore, the horizontal displacement (the disparity D) between the two centres in the left and right frame is the difference between the x components of the target's centre point ($x_L - x_R$). To calculate the depth (i.e., the distance to the target (Z)) the triangulation method was applied to estimate the absolute distance to the target using (2) and (3).

Depth map generation: The depth estimation to all corresponding points projected on the image plane in the stereo vision system using the triangulation method will, inevitably, generate depth maps that are, in most cases, rough and sparse wherever matching between corresponding pixels fails [66]. Meanwhile, with the rapid development of deep/convolutional neural networks (CNNs), monocular depth estimation based on deep learning has been widely studied recently, showing promising accuracy. These CNN-based methods are able to generate dense depth maps from single images where the depth at every pixel in the image is estimated by the neural network in an end-to-end manner [67–69] (see Figure 4). With CNN-based methods, the estimation of the absolute depth (i.e., depth from the object to the camera) directly from a single image can be ambiguous in scale; for example, an object may appear to be the same as another identically shaped but smaller object at a nearer distance [70]. The relative depth, on the other hand, which is the ratio between the depths of two points in an image, is scale-invariant. This principle also applies to humans since it is easier to choose the nearer between two points than to estimate the absolute depth of each point; therefore, relative depths are easier to estimate than ordinary (absolute) depths [70]. The adopted monocular depth estimation in this work [71] generates a relatively dense depth map of each pixel in a single image with values between 0 and 1.0, where pixels with higher values are closer to the camera, and pixels with small values are further from the camera. By inverting this dense map, it is possible to assign small relative distances to closer pixels to the camera and higher relative distances to the furthest pixels from the camera.

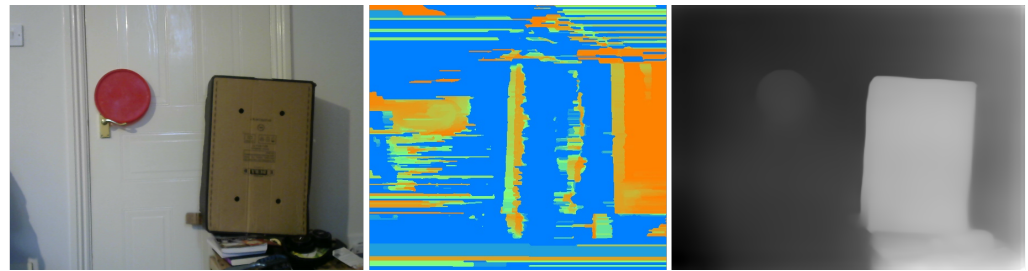


Figure 4. Depth maps: original image (left), deteriorated sparse depth map (middle), CNN-based dense depth map (right).

Next, the (absolute) distance of the target obtained from (2) and (3) is used to compute a scalar S , such that:

$$S_{cm/px} = \frac{Z_{xt,yt}(cm)}{R_{xt,yt}(px)} \quad (4)$$

where $Z_{xt,yt}$ is the (absolute) distance to the centre of the target in cm, and $R_{xt,yt}$ is the (relative) distance to the centre of the target in pixels (see Figure 5).

In this scenario, the scale $S = 242.839/0.782 = 342.857$. By multiplying the scalar S with every entry in the relatively dense depth map, it is possible to generate the absolute depth map. To illustrate this, suppose a point p_i is a pixel anywhere in the image, with relative distance equal to the one at the centre of the target $R_{xi,yj} = 0.782$, i.e., both the point p_i and the centre of the target are the same distance from the camera, thus, the absolute distance at $p_i = 0.782 \times 342.857 = 242.389$ cm, which is the same absolute distance of the target from the camera. Similarly, if p_j is a pixel with relative distance $R_{xi,yj} = 0.384$, which is closer to the camera than the target, then the real distance of that point is $p_j = 0.384 \times 342.857 = 130$ cm.

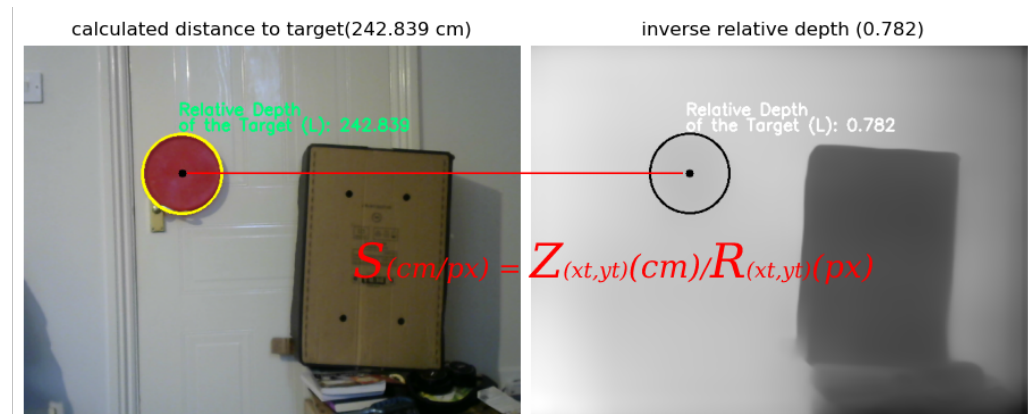


Figure 5. S is the ratio of the absolute distance Z in the left image to the relative distance R from the depth map in the right image.

3.3. Information Retrieval and Processing

Instance segmentation: Now, the attention is turned to information retrieval and processing using object detection. Since the current implementation is devoted to measuring concrete structures (concrete columns and walls) only, a neural network model [72] was trained to extract (segment) those objects from a given image. A sample of the dataset used for training the model is presented in Figure 6).

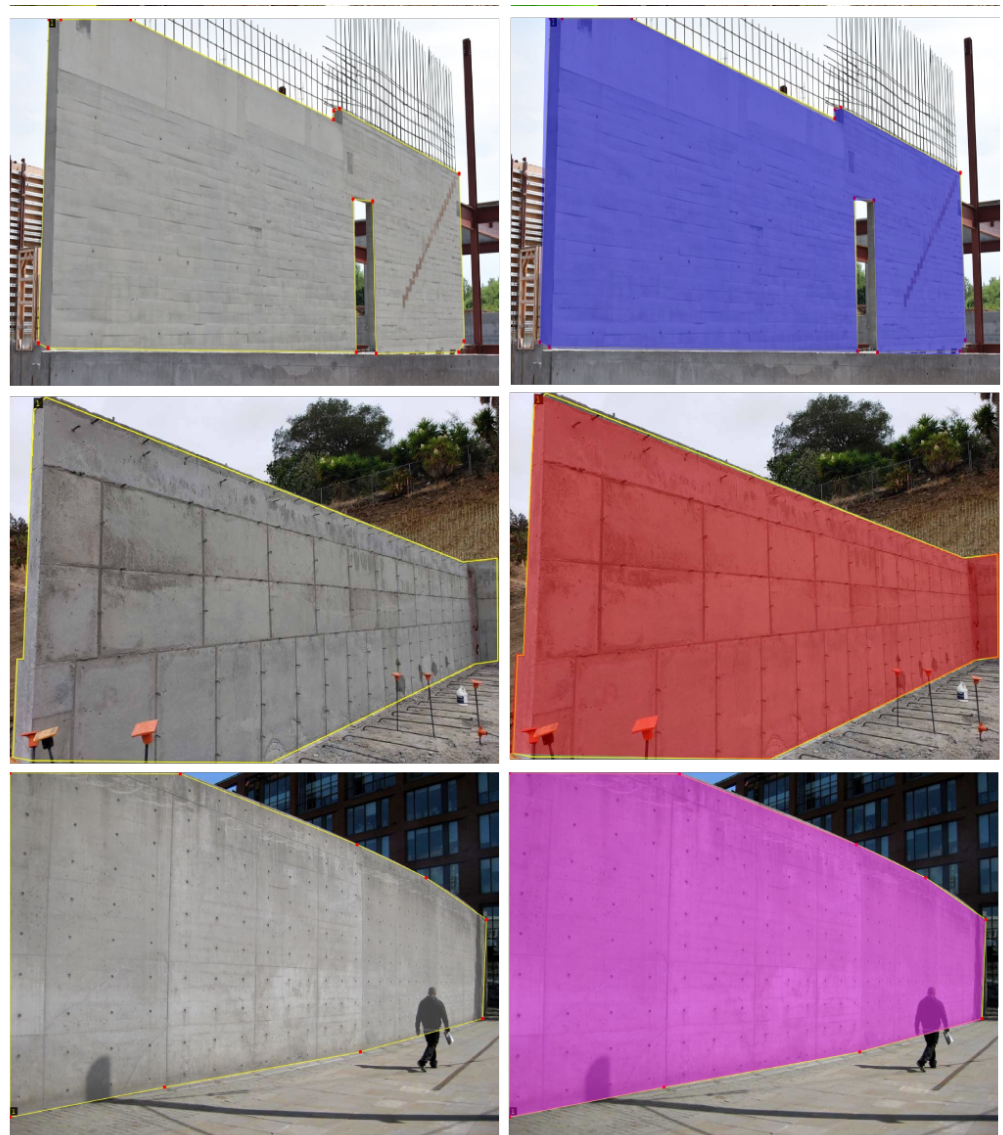


Figure 6. Sample of images with concrete structures used to train the neural network model.

Object detection and segmentation is the process of identifying the presence of an object in the image. It associates every pixel of that object with a single class label, e.g., a person, box, car, and so on [73]. For every class, the neural network applies a unique colour mask over all the pixels of that object. There are two types of object segmentation: (1) semantic, where the neural network treats multiple objects of the same class as a single entity, and (2) instance segmentation, which, in contrast to semantic, treats multiple objects of the same class as distinct individual instances [73].

Figure 7 demonstrates the process of using object segmentation in an image to extract the corresponding pixels with absolute depth values of that object from the dense depth map.

First, a single image (left frame) is passed through the neural network for instance segmentation (Figure 7, top left). The model generates a colour mask over the object of interest and assigns all pixels related to the object with a single label (Figure 7, top middle). The corresponding masks of each object, which are saved separately, are used to extract the boundary of that object (Figure 7, top right).

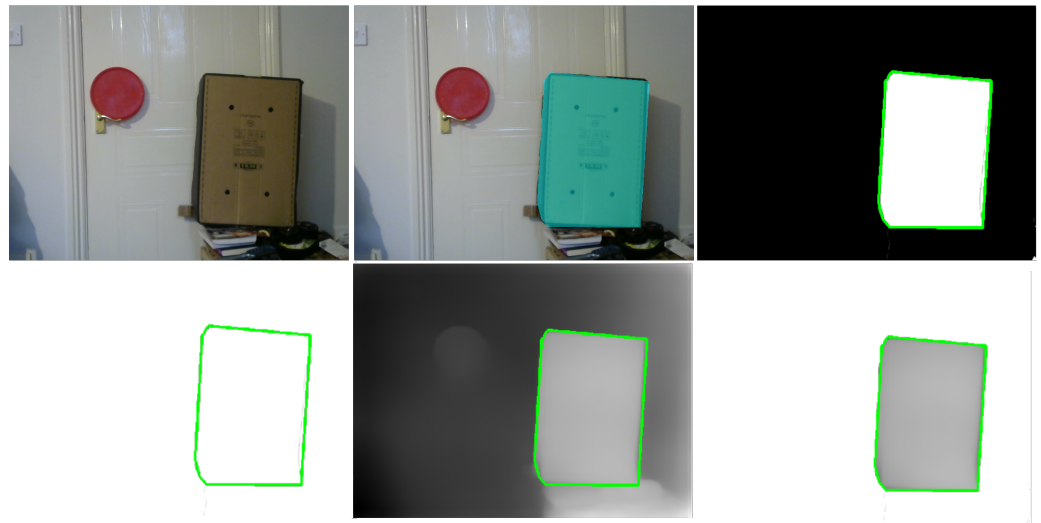


Figure 7. Extraction of the object boundary using the left frame: object detection, mask generation, separation of boundary points from the depth map.

The extracted boundary of the object (Figure 7, bottom left) is projected on the dense depth map containing the computed absolute depths (Figure 7, bottom middle) to separate only those pixels of the object of interest (Figure 7, bottom right). The mathematical formulation of this process is presented in the next section.

3.4. As-Built Component Measurement

Estimate world coordinates. The next step of the proposed pipeline is to compute the real-world coordinates of the object and estimate the area of the object. The conversion to a real-world coordinate system (in cm) from the image-coordinate system (in pixels) is governed by the following equations:

$$X_w = (x - c_x) * Z_{xi,yj} / f_x \quad (5)$$

$$Y_w = (y - c_y) * Z_{xi,yj} / f_y \quad (6)$$

X_w, Y_w are the computed two-dimensional real-world coordinates of each pixel in the object. x, y are the coordinates of each pixel in the object, c_x, c_y are the principal coordinates of the camera, which are estimated during the calibration, f_x, f_y are the lengths of the focal point (in pixels) also found from the camera matrix (K), and $Z_{xi,yj}$ is the absolute distance at that pixel. The code in Algorithm 1 illustrates the procedure of obtaining the real-world coordinates of the object of interest.

Calculating the area of the object of interest: Lastly, Green's theorem [27] was applied to calculate the area of the two-dimensional irregular region, i.e., the closure $\bar{D}$, which is enclosed by the boundary ∂D and denoted in Algorithm 1 as BV .

3.5. Visualisation of the Output

The proposed pipeline, which was entirely developed in Python, can produce an output of any text or graphical format. This makes the integration of such an output with any BIM model an easy task. Full demonstrations of the output samples are presented in Figures 10, 11 and 13.

Algorithm 1: Real-world coordinates.**input** : Camera Matrix (**K**)

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Focal Length **Alpha** = $\{f_x, f_y\}$,
 Boundary $\partial D := \{p \in X | p\} D$,
 Principal Point **O** = $\{c_x, c_y\}$,
 Image **M**, Mask **D**, $S \in M : \{M := (X, d)\}$,
 Boundary $\partial D := \{p \in X | p\} D$,

output: Set of boundary points

$$BV\{X_w, Y_w\} = \{(X_{w1}, Y_{w1}), \dots, (X_{wn}, Y_{wn})\}$$

```

1 //Initialisation
2   OBJ ← Do
3   C ← ∂D
4   BV ← {ϕ}                                     ▷ Boundary vertices
5 for each xi ∈ OBJ do
6   for each yj ∈ OBJ do
7     // Calculate world coordinates,
8     Xw ← (x − cx) * OBJ{xi, yj} / fx
9     Yw ← (y − cy) * OBJ{xi, yj} / fy
10    if {x, y} ∈ ∂D then
11      BV{ } ← BV{ } ∪ {Xw, Yw}                 ▷ add point to the boundary
12    else
13      i ← i + 1
14      j ← j + 1
15    end
16  end
17 end
18 return (BV{Xw, Yw})

```

4. Results and Discussion

A discussion of the results through two examples follows: one for measuring a concrete column and the other for measuring a concrete wall at an Oxford Brookes University building. The building is a new addition to Oxford Brookes University and is mainly built from concrete components.

In the first experiment, a stereo camera system with a baseline ($b = 25$ cm) was used to capture the left and right frames of the scene. The stereo camera system was placed at a distance of 5.2 m away from the column. The distance and the angle of the acquisition were chosen randomly, allowing the whole object (a concrete column) to appear completely in both frames. A target (red circular disk) was placed on the object at the same level of the stereo system, where it was also visible in both cameras (Figure 8a,b).

Once the scene was captured, a (red) colour filter was applied to allocate the target in both frames and calculate the coordinates of the centre of the circle in the left frame (x_l, y_l) (Figure 8d-top row) and in the right frame (Figure 8d-bottom row) (x_r, y_r), respectively. Next, using the principles of triangulation demonstrated in Figure 9, the depth $Z_{xt,yt}$ (e.g., distance to the centre of the target) was calculated using the following formula:

$$Z_{xt,yt} = \frac{f \cdot B}{X_r - X_l} \quad (7)$$

where f is the horizontal focal length (in pixels), B is the baseline, and $x_r - x_l$ is the horizontal disparity. In this experiment, the calculated depth Z_{x_t, y_t} to the centre of the target was 5.17 m and the result is displayed in the left frame (Figure 8e-top row) and the right frame (Figure 8e-bottom row), respectively.

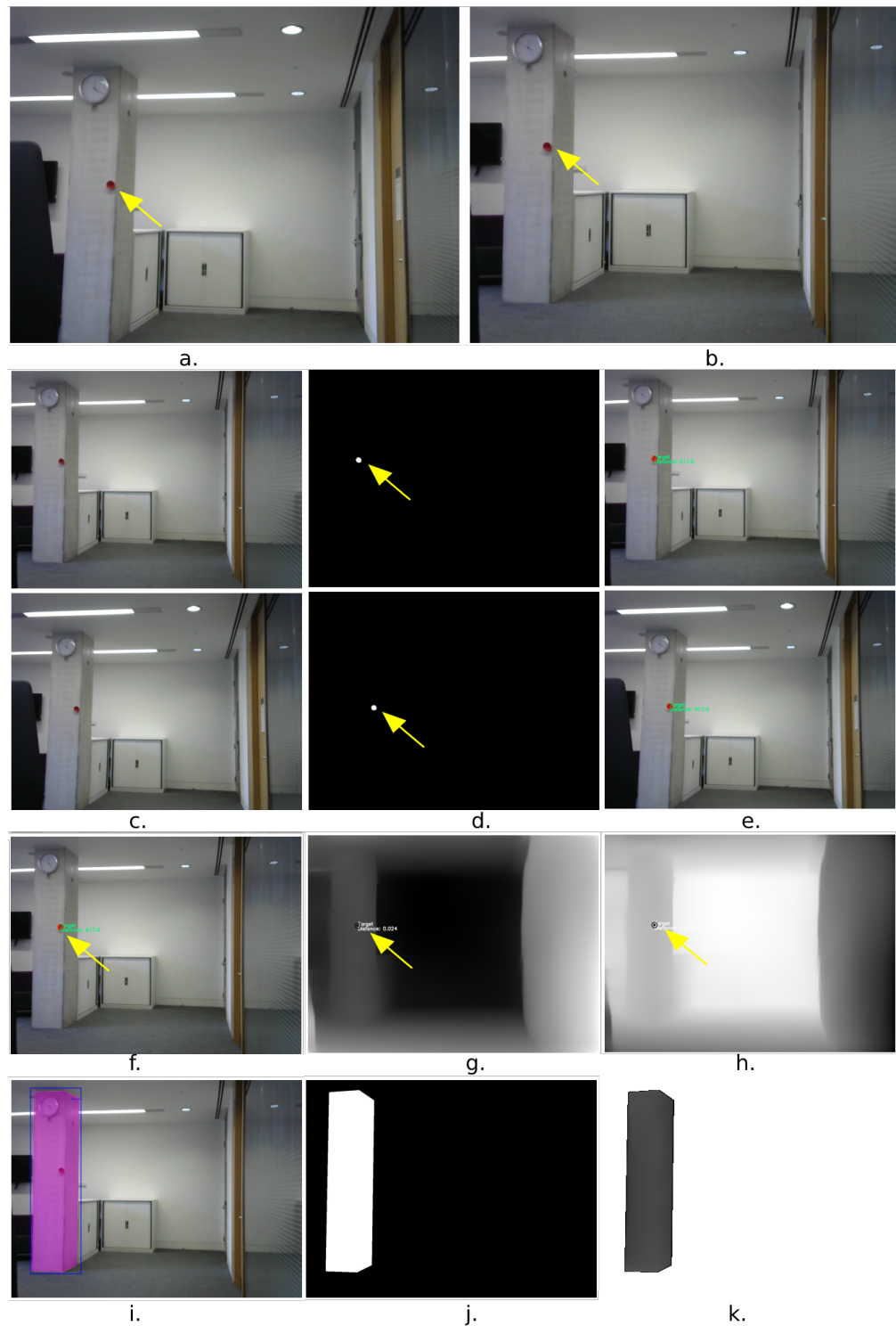


Figure 8. Experiment 1: estimating the area of a concrete column: in the first row scene capture (a,b), in the second are third row depth estimation to target in the left and right frames respectively (c–e), in the fourth row, the depth map (g), and corresponding inverse (h), using the left frame (f). In the last row, object segmentation (i), masking (j), and object extraction (k).

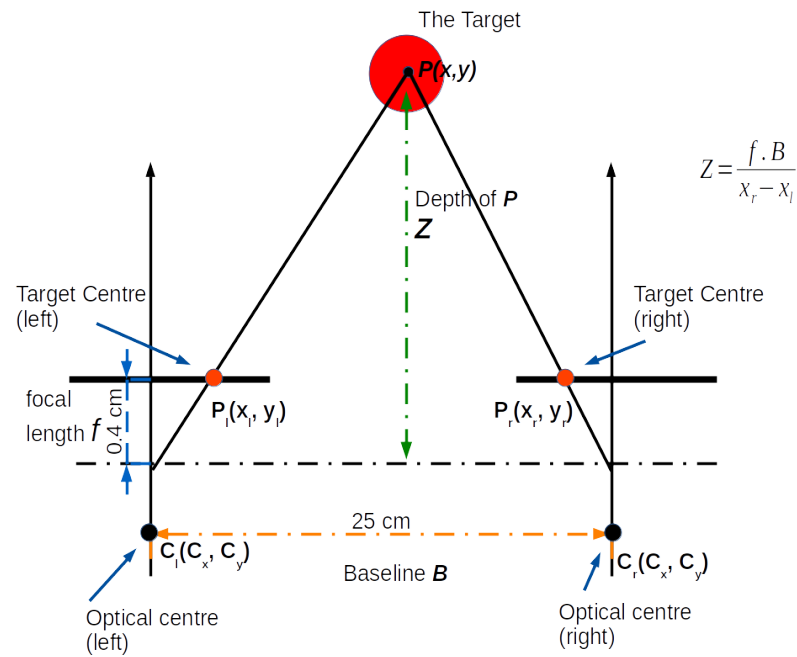


Figure 9. Depth from disparity: $P(x,y)$ is the real-world coordinates of the target centre, Z is the calculated distance (Depth) to the centre of the target. P_l and P_r represent the coordinates of the centre of the target in the left and right images, respectively. C_l and C_r represent the principal points of the left and right cameras estimated during the calibration stage.

The left frame (Figure 8f) was then passed through the CNN-based monocular depth estimation model to generate the inverse dense depth map containing the relative depths of the scene (Figure 8g). The dense depth map was then inverted again so that a small relative depth indicates the closer points and the larger values refer to further points (Figure 8h).

To calculate the scalar S for this experiment, both $Z_{x_t, y_t} = 517$ cm and $R_{x_t, y_t} = 0.976$ were used, referring to the relative depth values obtained from the inverted depth map (Figure 8h) at points x_t, y_t , i.e., the target centres.

The scalar S is case-dependant, i.e., it varies depending on the position of the target in that scene. For this case, the scalar S is calculated as follows:

$$S = \frac{D_{cm}}{R_{px}} = \frac{512}{0.976} = 529.713 \text{ cm/px.} \quad (8)$$

Now, the inverted depth map shown in Figure 8h was multiplied by the scalar S to generate a dense map with absolute depths.

To extract the object (the concrete column) from the scene, the left frame was passed through the trained module for object detection and instance segmentation as shown in Figure 8i. The output mask is shown in Figure 8j; corresponding to the detected object, i.e., the concrete column, it is projected on the dense map with the absolute depth values, which were generated in the previous step to extract only those segmented pixels related to the concrete column. Finally, the code in Algorithm 1 was applied to split the vertices belonging to the boundary (BV) and then Green's theorem was used to estimate the area of the concrete column.

The actual surface areas of the column are shown in Figure 10(left), as follows: $270 \text{ cm (h)} \times 50 \text{ cm (w)} = 1.35 \text{ m}^2 \times 2 \text{ faces} = 2.70 \text{ m}^2$. The pixel coordinates shown in Figure 10(middle) are those of the mask generated during the instance segmentation process, and are used to calculate the real-world reconstruction of column Figure 10(right). The calculated surface area of the column is $A = 2.5104 \text{ m}^2$. The area of the front face (Figure 11(middle)) is $A = 1.1898 \text{ m}^2$, and the area of the side face is $A = 1.32055 \text{ m}^2$ (Figure 11(right)).

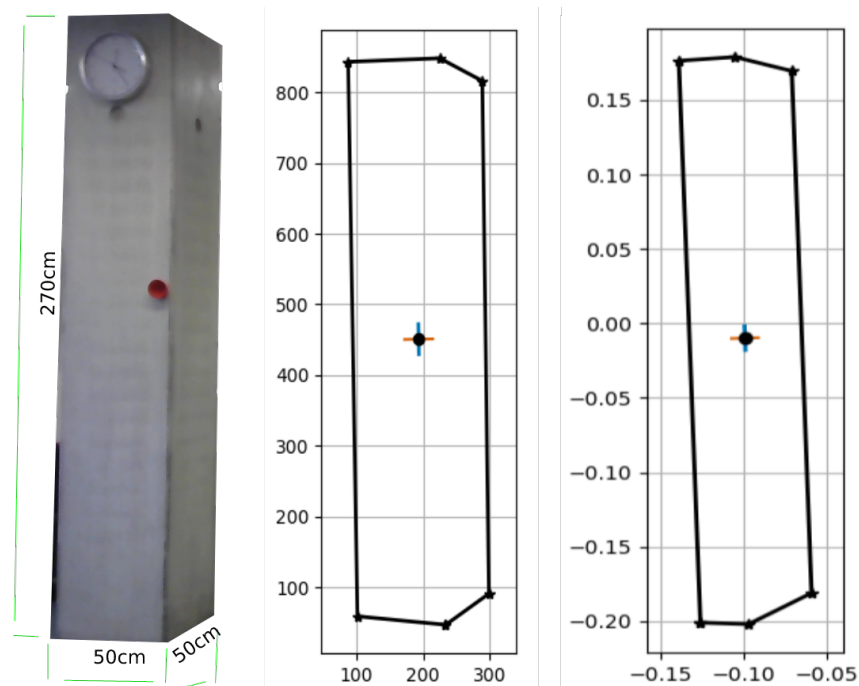


Figure 10. Real-world reconstruction of the concrete column: (left) the actual column, (middle) the pixel coordinates, and (right) real-world reconstruction.

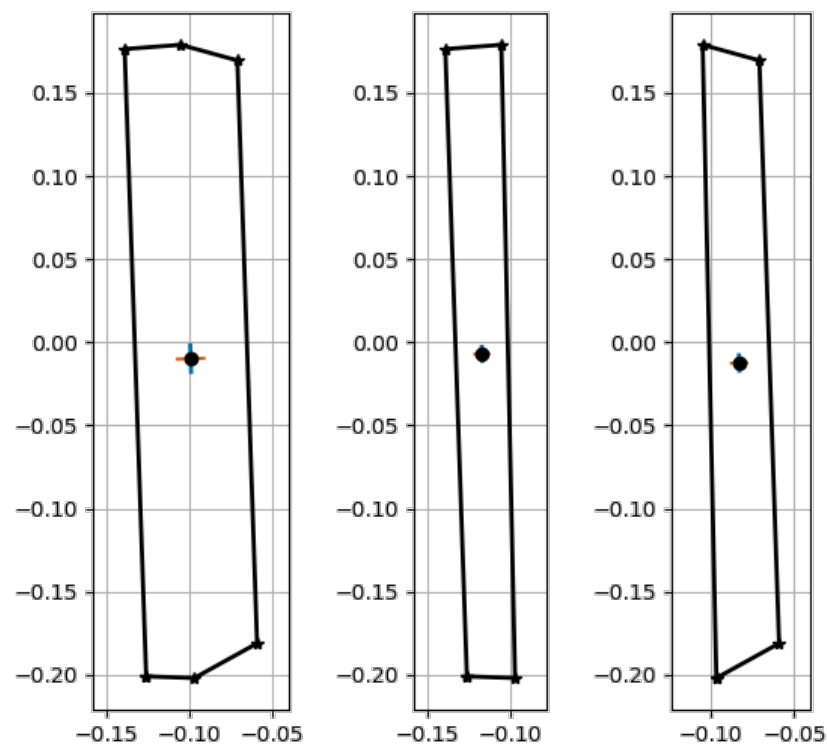


Figure 11. Reconstructed concrete column: the left is the total surface area, the middle is the front face area, and the right is the side face area.

The percentage error in this case, i.e., the ABS $((2.5104 - 2.7)/2.7) \times 100$, is 7.022%.

In the second experiment, the same stereo camera system was used with the baseline ($b = 25$ cm) to capture the left and right frames of the scene containing a section of concrete wall as depicted in Figure 12a,b. The stereo camera system was placed at a distance of 5.6 m

away from the wall section, with the target placed at the same level as the stereo system, where it is visible by both cameras (Figure 12a,b).

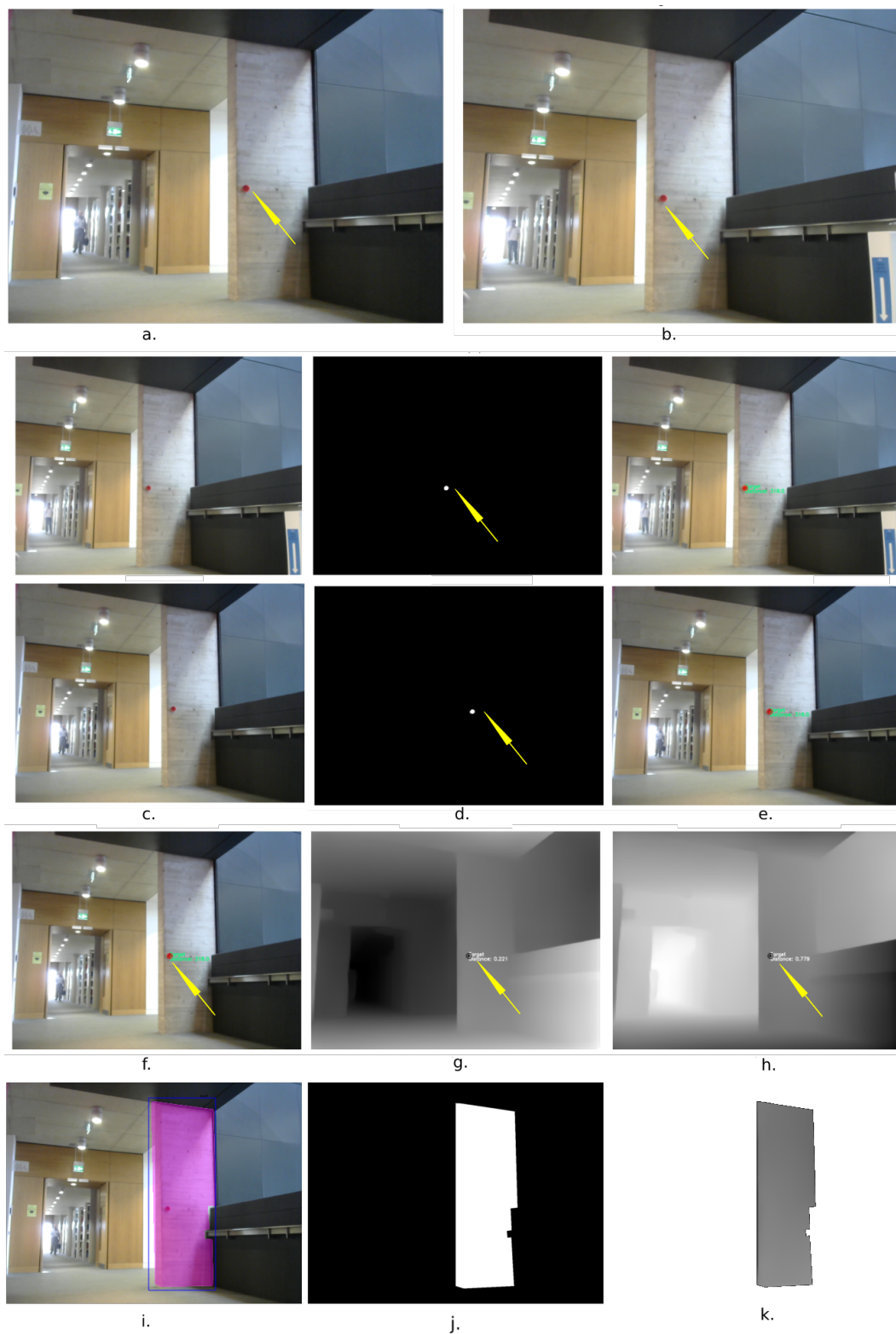


Figure 12. Experiment 2: estimation of the area of a concrete wall: in the first row scene capture (a,b), in the second are third row depth estimation to target in the left and right frames respectively (c–e), in the fourth row, the depth map (g), and corresponding inverse (h), using the left frame (f). In the last row, object segmentation (i), masking (j), and object extraction (k).

In this experiment, the calculated depth $Z_{xt,yt}$ to the centre of the target was 5.59 m, corresponding to $R_{xt,yt} = 0.779$. Therefore, the scalar S in this case is:

$$S = \frac{559}{0.779} = 717.586 \text{ cm/px.} \quad (9)$$

Similarly, the inverted depth map shown in Figure 12h was multiplied by the scalar S to generate a dense map with absolute depths; it then passed the left frame through the trained module for object detection and instance segmentation to extract the concrete wall section, as shown in Figure 12i. The output mask, which is shown in Figure 12j, corresponding to the detected object, i.e., the concrete wall, is projected on the dense map with the absolute depth values, which were generated in the respective step of the first experiment to extract only those segmented pixels related to the concrete wall. Finally, the code in Algorithm 1 was applied to split the vertices belonging to the boundary (BV); we used Green's theorem to estimate the area of the concrete wall.

The actual surface area of the wall section is slightly more complex, it is the sum of the bottom half, the upper half, and the side face, as shown in Figure 13(left): $(130 \text{ cm} \times 80 \text{ cm}) + (170 \text{ cm} \times 100 \text{ cm}) + (30 \text{ cm} \times 300 \text{ cm}) = 36400 \text{ cm}^2$ or 3.64 m^2 . The pixel coordinates shown in Figure 13(middle) refer to the mask generated during the instance segmentation process and are used to calculate the real-world reconstruction of the column Figure 13(right). The calculated surface area of the column is $A = 3.3143 \text{ m}^2$. The percentage error in this case, i.e., the ABS $((3.3143 - 3.640)/3.640) \times 100$ is 8.947%.

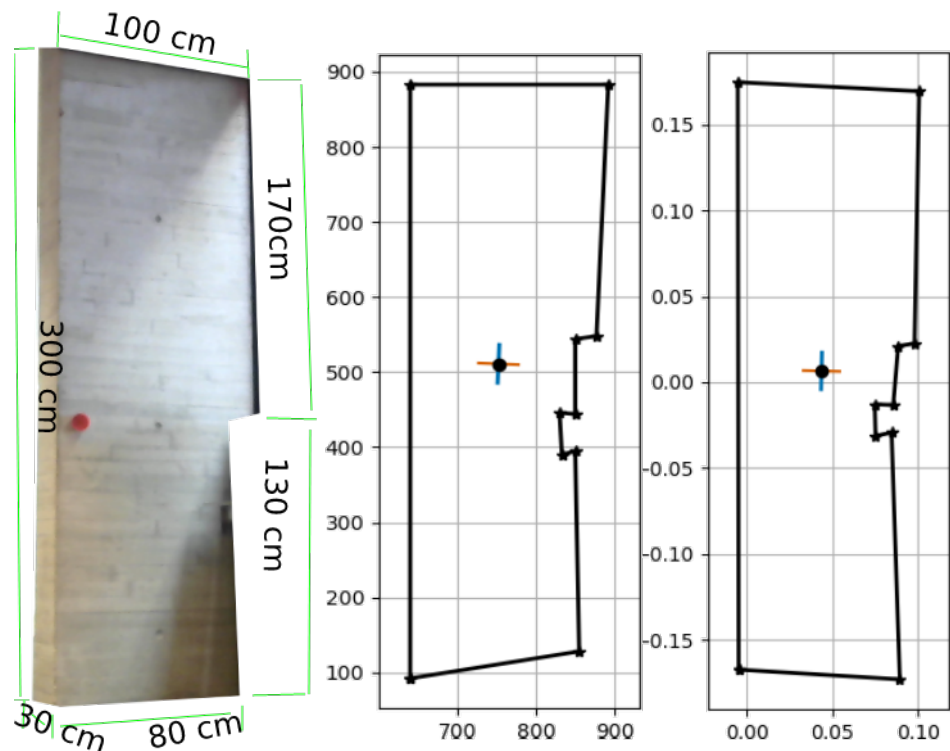


Figure 13. Real-world reconstruction of the concrete wall: the **left** is the total surface area, the **middle** is the front face area, and the **right** is the side face area.

The side face in this reconstruction is undetected; therefore, it is hard to estimate its contribution to the total estimated area.

Limitations

Inferring the depth from a two-dimensional image is an extremely ill-posed problem. Errors may arise from many sources, but most importantly, the key contribution is attributed

to the small scale at which the calculations are performed; i.e., the pixel scale, which will eventually be transformed to the real-world scale; i.e., meters, cm, feet, or inches.

The first error source comes from the camera calibration, which is the process of estimating intrinsic and/or extrinsic parameters of the camera. During this process, the focal length of the camera in pixels was estimated, which is the distance between the lens and the image sensor when the subject is in focus, and the principal point of the camera, which is the point on the image plane onto which the perspective centre is projected. There are other intrinsic parameters that contribute to errors, such as the skewness and the distortion of the lens, but in most cases, these factors are negligible.

The second error source comes from calculating the disparity from the horizontal displacement between the left and right frames. Regardless of the approach used to estimate the disparity, the main concept is to find the same pixel in both the left and right frames and calculate the difference between the x -components of that pixel.

There is another potential error that arises during the identification of the object of interest in a scene. With instance segmentation, a mask was used to select every pixel that is related to the (whole) object. Therefore, if the mask is poorly generated, this may lead to incorrect calculations. Finally, there is the well-established problem of estimating the absolute depths of each and every pixel in the scene. Whilst CNN-based methods are well-known to be able to generate relatively dense depth maps, they are very hard to train on a specific task. Sparse depth maps, on the other hand, are rough and are not suitable for geometric estimations that require a level of precision.

5. Conclusions

The proposed pipeline offers a fully automated computer vision-based method for measuring as-built elements of built assets.

The novelty of this work is attributed to the use of a single image and a single target to develop a fully automated computer vision-based method for measuring any given object.

Stereo camera techniques were used for data acquisition and deducing depth information. Machine learning, object detection, and instance segmentation techniques were also utilised to isolate the as-built element of interest and to process the geometric information of these elements. Finally, the principles of Green's theorem were applied to estimate the size of the object(s). To demonstrate the results, a neural network was trained to detect and segment concrete walls and columns. A red disk target was placed in the field of view and we used a calibrated stereo camera system to capture the scene. A depth map was generated for this scene and the distance to the target was also calculated using triangulation methods. This information was then used to calculate the real-world dimensions of the object, which was then used to estimate the surface area. Limitations to the approach can arise during the camera calibration process and from calculating the disparity displacement between the left and right frames. Errors may also arise due to incorrect identification and segmentation of the object of interest, which may result in a poorly generated mask, which could lead to incorrect area calculations. The proposed pipeline was applied and tested on as-built elements within a university campus. However, we intend to further extend this work and examine the feasibility, scale-up, and practicality of the proposed fully automated CV-based method on real-life construction sites.

Author Contributions: Methodology, H.P.; Software, H.P.; Validation, H.P.; Formal analysis, H.P.; Investigation, H.P.; Writing—review & editing, J.H.M.T.; Supervision, J.H.M.T.; Project administration, J.H.M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3D	three-dimensional
AI	artificial intelligence
BIM	building information modelling
CV	computer vision
CNN	convolutional neural network
DLT	direct linear transformation
UAV	unmanned aerial vehicle

References

- Davidson, I.N.; Skibniewski, M.J. Simulation of automated data collection in buildings. *J. Comput. Civ. Eng.* **1995**, *9*, 9–20. [\[CrossRef\]](#)
- Navon, R. Research in automated measurement of project performance indicators. *Autom. Constr.* **2007**, *16*, 176–188. [\[CrossRef\]](#)
- Tsai, M.-K.; Yang, J.-B.; Lin, C.-Y. Synchronization-based model for improving on-site data collection performance. *Autom. Constr.* **2007**, *16*, 323–335. [\[CrossRef\]](#)
- Saidi, K.S.; Lytle, A.M.; Stone, W.C. Report of the NIST workshop on data exchange standards at the construction job site. In Proceedings of the 20th International Symposium on Automation and Robotics in Construction (ISARC), Eindhoven, The Netherlands, 21–24 September 2003; pp. 617–622.
- De Marco, A.; Briccarello, D.; Rafele, C. Cost and Schedule Monitoring of Industrial Building Projects: Case Study. *J. Constr. Eng. Manag.* **2009**, *135*, 853–862. [\[CrossRef\]](#)
- Navon, R.; Sacks, R. Assessing research issues in automated project performance control (APPC). *Autom. Constr.* **2007**, *16*, 474–484. [\[CrossRef\]](#)
- Manfren, M.; Tagliabue, L.C.; Re Cecconi, F.; Ricci, M. Long-term techno-economic performance monitoring to promote built environment decarbonisation and digital transformation—A case study. *Sustainability* **2022**, *14*, 644. [\[CrossRef\]](#)
- Omar, T.; Nehdi, L. Data acquisition technologies for construction progress tracking. *Autom. Constr.* **2016**, *70*, 143–155. [\[CrossRef\]](#)
- Bradski, G.; Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2008.
- Bohn, J.S.; Teizer, J. Benefits and Barriers of Construction Project Monitoring Using High-Resolution Automated Cameras. *J. Constr. Eng. Manag.* **2010**, *136*, 632–640. [\[CrossRef\]](#)
- Golparvar-Fard, M.; Peña-Mora, F.; Savarese, S. Integrated Sequential As-Built and As-Planned Representation with D4AR Tools in Support of Decision-Making Tasks in the AEC/FM Industry. *J. Constr. Eng. Manag.* **2011**, *137*, 1099–1116. [\[CrossRef\]](#)
- Bosché, F.; Guillemet, A.; Turkan, Y.; Haas, C.T.; Haas, R. Tracking the built status of MEP works: Assessing the value of a Scan-vs-BIM system. *J. Comput. Civ. Eng.* **2014**, *28*, 4. [\[CrossRef\]](#)
- Zhang, X.; Bakis, N.; Lukins, T.C.; Ibrahim, Y.M.; Wu, S.; Kagioglou, M.; Aouad, G.; Kaka, A.P.; Trucco, E. Automating progress measurement of construction projects. *Autom. Constr.* **2009**, *18*, 294–301. [\[CrossRef\]](#)
- Fisher, R.B.; Breckon, T.P.; Dawson-Howe, K.; Fitzgibbon, A.; Robertson, C.; Trucco, E.; Williams, C.K.I. *Dictionary of Computer Vision and Image Processing*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
- Guinchard, M.; Angeletti, M.; Boyer, F.; Catinaccio, A.; Gargiulo, C.; Lacny, L.; Laudi, E.; Scislo, L. Experimental modal analysis of lightweight structures used in particle detectors: Optical non-contact method. In Proceedings of the 9th International Particle Accelerator Conference, IPAC18, Vancouver, BC, Canada, 29 April–4 May 2018; pp. 2565–2567.
- Elazouni, A.; Salem, O.A. Progress monitoring of construction projects using pattern recognition techniques. *Constr. Manag. Econ.* **2011**, *29*, 355–370. [\[CrossRef\]](#)
- Lukins, T.C.; Trucco, E. Towards automated visual assessment of progress in construction projects. In Proceedings of the British Machine Vision Conference, Warwick, UK, 10–13 September 2007.
- Rebolj, D.; Babič, N.; Magdič, A.; Podbreznik, P.; Pšunder, M. Automated construction activity monitoring system. *Adv. Eng. Inform.* **2008**, *22*, 493–503. [\[CrossRef\]](#)
- Kim, H.; Kano, N. Comparison of construction photograph and VR image in construction progress. *Autom. Constr.* **2008**, *17*, 137–143. [\[CrossRef\]](#)
- Kopsida, M.; Brilakis, I.; Vela, P.A. A review of automated construction progress monitoring and inspection methods. In Proceedings of the 32nd CIB W78 Conference 2015, Eindhoven, The Netherlands, 27–29 October 2015; pp. 421–431.
- Álvares, J.S.; Costa, D.B. Literature review on visual construction progress monitoring using unmanned aerial vehicles. In Proceedings of the 26th Annual Conference of the International Group for Lean Construction: Evolving Lean Construction Towards Mature Production Management Across Cultures and Frontiers, Chennai, India, 6–22 July 2018; pp. 18–22.
- Teizer, J. Status quo and open challenges in vision-based sensing and tracking of temporary resources on infrastructure construction sites. *Adv. Eng. Inform.* **2015**, *29*, 225–238. [\[CrossRef\]](#)

23. Borrmann, A.; Stilla, U. Automated Progress Monitoring Based on Photogrammetric Point Clouds and Precedence Relationship Graphs. In Proceedings of the 32nd International Symposium on Automation and Robotics in Construction (ISARC), Oulu, Finland, 15–18 June 2015; pp. 1–7.
24. Dimitrov, A.; Golparvar-Fard, M. Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. *Adv. Eng. Inform.* **2014**, *28*, 37–49. [\[CrossRef\]](#)
25. Kim, Y.; Nguyen, C.H.P.; Choi, Y. Automatic pipe and elbow recognition from three-dimensional point cloud model of industrial plant piping system using convolutional neural network-based primitive classification. *Autom. Constr.* **2020**, *116*, 103236. [\[CrossRef\]](#)
26. Chen, J.; Fang, Y.; Cho, Y.K. Unsupervised Recognition of Volumetric Structural Components from Building Point Clouds. In Proceedings of the ASCE International Workshop on Computing in Civil Engineering, Seattle, DC, USA, 25–27 June 2017; pp. 177–184.
27. Riley, K.F.; Hobson, M.P.; Bence, S.J. *Mathematical Methods for Physics and Engineering*; American Association of Physics Teachers: College Park, MD, USA, 1999.
28. Kim, C.; Son, H.; Kim, C. Automated construction progress measurement using a 4D building information model and 3D data. *Autom. Constr.* **2013**, *31*, 75–82. [\[CrossRef\]](#)
29. Abdel Aziz, A.M. Minimum performance bounds for evaluating contractors' performance during construction of highway pavement projects. *Constr. Manag. Econ.* **2008**, *26*, 507–529. [\[CrossRef\]](#)
30. Hwang, B.-G.; Zhao, X.; Ng, S.Y. Identifying the critical factors affecting schedule performance of public housing projects. *Habitat Int.* **2013**, *38*, 214–221. [\[CrossRef\]](#)
31. Turkan, Y.; Bosche, F.; Haas, C.T.; Haas, R. Automated progress tracking using 4D schedule and 3D sensing technologies. *Autom. Constr.* **2012**, *22*, 414–421. [\[CrossRef\]](#)
32. Witzgall, C.J.; Bernal, J.; Cheok, G. TIN techniques for data analysis and surface construction. *Christoph J. Witzgall Javier Bernal Geraldine Cheok* **2004**. [\[CrossRef\]](#)
33. Du, J.-C.; Teng, H.-C. 3D laser scanning and GPS technology for landslide earthwork volume estimation. *Autom. Constr.* **2007**, *16*, 657–663. [\[CrossRef\]](#)
34. Shih, N.-J.; Wang, P.-H. Point-cloud-based comparison between construction schedule and as-built progress: Long-range three-dimensional laser scanner's approach. *J. Archit. Eng.* **2004**, *10*, 98–102. [\[CrossRef\]](#)
35. Bosché, F. Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Adv. Eng. Inform.* **2010**, *24*, 107–118. [\[CrossRef\]](#)
36. Son, H.; Kim, C. 3D structural component recognition and modeling method using color and 3D data for construction progress monitoring. *Autom. Constr.* **2010**, *19*, 844–854. [\[CrossRef\]](#)
37. Golparvar-Fard, M.; Pena-Mora, F.; Savarese, S. Automated progress monitoring using unordered daily construction photographs and IFC-based building information models. *J. Comput. Civ. Eng.* **2015**, *29*, 04014025. [\[CrossRef\]](#)
38. Taj, G.; Anand, S.; Haneefi, A.; Kanishka, R.P.; Mythra, D. Monitoring of Historical Structures using Drones. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *955*, 012008. [\[CrossRef\]](#)
39. Ibrahim, A.; Golparvar-Fard, M.; El-Rayes, K. Metrics and methods for evaluating model-driven reality capture plans. *Comput. Civ. Infrastruct. Eng.* **2021**, *37*, 55–72. [\[CrossRef\]](#)
40. Wu, Y.; Wang, M.; Liu, X.; Wang, Z.; Ma, T.; Xie, Y.; Li, X.; Wang, X. Construction of Stretching-Bending Sequential Pattern to Recognize Work Cycles for Earthmoving Excavator from Long Video Sequences. *Sensors* **2021**, *21*, 3427. [\[CrossRef\]](#)
41. Shang, Z.; Shen, Z. Real-Time 3D Reconstruction on Construction Site Using Visual SLAM and UAV. *arXiv* **2017**, arXiv:1712.07122.
42. Shojaei, A.; Moud, H.I.; Flood, I. Proof of Concept for the Use of Small Unmanned Surface Vehicle in Built Environment Management. In Proceedings of the Construction Research Congress 2018: Construction Information Technology—Selected Papers from the Construction Research Congress, New Orleans, LA, USA, 2–4 April 2018; pp. 116–126.
43. Mahami, H.; Nasirzadeh, F.; Ahmadabadian, A.H.; Esmaili, F.; Nahavandi, S. Imaging network design to improve the automated construction progress monitoring process. *Constr. Innov.* **2019**, *19*, 386–404. [\[CrossRef\]](#)
44. Han, K.; Golparvar-Fard, M. Crowdsourcing BIM-guided collection of construction material library from site photologs. *Vis. Eng.* **2017**, *5*, 14. [\[CrossRef\]](#)
45. Kielhauser, C.; Manzano, R.R.; Hoffman, J.J.; Adey, B.T. Automated Construction Progress and Quality Monitoring for Commercial Buildings with Unmanned Aerial Systems: An Application Study from Switzerland. *Infrastructures* **2020**, *5*, 98. [\[CrossRef\]](#)
46. Braun, A.; Borrmann, A. Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. *Autom. Constr.* **2019**, *106*, 102879. [\[CrossRef\]](#)
47. Masood, M.K.; Aikala, A.; Seppänen, O.; Singh, V. Multi-Building Extraction and Alignment for As-Built Point Clouds: A Case Study With Crane Cameras. *Front. Built Environ.* **2020**, *6*, 581295. [\[CrossRef\]](#)
48. Bosché, F. Plane-based registration of construction laser scans with 3D/4D building models. *Adv. Eng. Inform.* **2012**, *26*, 90–102. [\[CrossRef\]](#)
49. Bueno, M.; Bosché, F.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Autom. Constr.* **2018**, *89*, 120–134. [\[CrossRef\]](#)
50. Styliadis, A.D. Digital documentation of historical buildings with 3-d modeling functionality. *Autom. Constr.* **2007**, *16*, 498–510. [\[CrossRef\]](#)

51. Shashi, M.; Jain, K. Use of photogrammetry in 3D modeling and visualization of buildings. *ARPN J. Eng. Appl. Sci.* **2007**, *2*, 37–40.
52. El-Omari, S.; Moselhi, O. Integrating 3D laser scanning and photogrammetry for progress measurement of construction work. *Autom. Constr.* **2008**, *18*, 1–9. [[CrossRef](#)]
53. Baltsavias, E.P. A comparison between photogrammetry and laser scanning. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 83–94. [[CrossRef](#)]
54. Golparvar-Fard, M.; Bohn, J.; Teizer, J.; Savarese, S.; Peña-Mora, F. Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques. *Autom. Constr.* **2011**, *20*, 1143–1155. [[CrossRef](#)]
55. Genovese, K.; Chi, Y.; Pan, B. Stereo-camera calibration for large-scale DIC measurements with active phase targets and planar mirrors. *Opt. Express* **2019**, *27*, 9040–9053. [[CrossRef](#)] [[PubMed](#)]
56. Bian, J.-W.; Wu, Y.-H.; Zhao, J.; Liu, Y.; Zhang, L.; Cheng, M.-M.; Reid, I. An evaluation of feature matchers for fundamental matrix estimation. *arXiv* **2019**, arXiv:1908.09474.
57. Sun, H.; Du, H.; Li, M.; Sun, H.; Zhang, X. Underwater image matching with efficient refractive-geometry estimation for measurement in glass-flume experiments. *Measurement* **2020**, *152*, 107391. [[CrossRef](#)]
58. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
59. Abdel-Aziz, Y.I.; Karara, H.M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogramm. Eng. Remote Sens.* **2015**, *81*, 103–107. [[CrossRef](#)]
60. Burger, W. Zhang's camera calibration algorithm: In-depth tutorial and implementation. *HGB16-05* **2016**, 1–6.. 10.13140/RG.2.1.1166.1688/1. [[CrossRef](#)]
61. Barone, F.; Marrazzo, M.; Oton, C.J. Camera Calibration with Weighted Direct Linear Transformation and Anisotropic Uncertainties of Image Control Points. *Sensors* **2020**, *20*, 1175. [[CrossRef](#)]
62. Abedin-Nasab, M.H. *Handbook of Robotic And image-Guided Surgery*; Elsevier: Amsterdam, The Netherlands, 2020; ISBN: 978-0-12-814245-5.
63. Kang, S.B.; Webb, J.; Zitnick, C. *An Active Multibaseline Stereo System With Real-Time Image Acquisition*; Carnegie-Mellon University. Department of Computer Science: Pittsburgh, PA, USA, 1999.
64. Hartley, R.I. Theory and Practice of Projective Rectification. *Int. J. Comput. Vis.* **1999**, *35*, 115–127. [[CrossRef](#)]
65. Lafiosca, P.; Ceccaroni, M. Rectifying homographies for stereo vision: Analytical solution for minimal distortion. *arXiv* **2022**, arXiv:2203.00123.
66. Zhao, C.; Sun, Q.; Zhang, C.; Tang, Y.; Qian, F. Monocular depth estimation based on deep learning: An overview. *Sci. China Technol. Sci.* **2020**, *63*, 1612–1627. [[CrossRef](#)]
67. Alhashim, I.; Wonka, P. High quality monocular depth estimation via transfer learning. *arXiv* **2018**, arXiv:1812.11941.
68. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp.270–279.
69. Casser, V.; Pirk, S.; Mahjourian, R.; Angelova, A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8001–8008. [[CrossRef](#)]
70. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inf. Process. Syst.* **2014**, arXiv:1406.2283. <https://doi.org/10.48550/arXiv.1406.2283>.
71. Lee, J.-H.; Kim, C.-S. Monocular depth estimation using relative depth maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9729–9738.
72. Girshick, R.; Radosavovic, I.; Gkioxari, G.; Dollár, P.; He, K. Detectron. 2018. Available online: <https://github.com/facebookresearch/detectron> (accessed on 27 July 2023).
73. Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Inf. Retr.* **2018**, *7*, 87–93. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.