Registering Georeferenced Photos to a Building Information Model to Extract Structures of Interest

Abstract

Vision-based techniques are being used to inspect structures such as buildings and infrastructure. Due to various backgrounds in the acquired images, conventional vision-based techniques rely heavily on manual processing to extract relevant structures of interest for subsequent analysis in many applications, such as distress detection. This practice is laborious, time-consuming, and error-prone. To address the challenge, this study proposes a new method that automatically matches a georeferenced real-life photo with a building information model-rendered synthetic image to allow the extraction of relevant structures of interest. Field experiments were conducted to validate and evaluate the proposed method. The average accuracy of this method is 79.21% and the processing speed is 140 seconds per image. The proposed method has the potential to reduce the workload of image processing for vision-based structure inspection.

Keywords: Vision-based inspection; Condition assessment; Region of interest; Building information model (BIM); Image processing; image-to-BIM registration.

1 Introduction

Vision-based structural inspection (VBSI) has been used to detect defects such as cracks, fractures, and spalling for building and infrastructure condition assessments. Over the past decades, many algorithms have been proposed for VBSI facilitated by the advancement of sensing and deep learning techniques. Existing studies achieved good performance on structured and ordered images that only contain the targeted structures to be inspected. These images are typically captured by a customized inspection device from certain designated view angles and distances set to control the influence of irrelevant background. However, the emerging inspection platforms provided by unmanned aerial vehicles (UAVs) [1–3] and unmanned ground vehicles (UGVs) [4] have provided a massive amount of unordered visual assets that are taken from various viewpoints and comprise both the structure of interest (SOI) and the surrounding background, i.e., sky, vegetation, and pedestrians. Directly identifying defects from such unordered images is a challenging task [5, 6] because the background information in an image undermines detection performance [7] from two aspects. First, it might increase the probability of false positives. For example, cracks are typically identified as continual-distributed pixels with a strip shape in a binary image. Similar patterns detected in the background can lead to undesirable false positives, as shown in Fig. 1. Second, processing irrelevant areas in the photos will bring in extra computation consumption. As a result, preprocessing is performed to extract the relevant SOI to overcome the influence of an irrelevant background before an unordered image can be used for further defect detection [8, 9].

Traditional SOI extraction techniques rely on prior knowledge [10-12], which either extracts the SOI or removes the background based on specific patterns, such as "a building generally has a straight contour" or "vegetation has a green color." However, different structures may have very different shapes and appearances, and their surrounding environments vary as the seasons alternate and the geographical location changes. Hence, it is very difficult to find a determined pattern for accurately extracting the SOI from the space-time varying background. This variation in the image features means that extra effort is needed to manually determine the pattern for extraction. The inefficient process causes an unnecessary waste of labor and may delay the detection of safety issues, as well as subsequent restoration work. Research efforts that aim to automate the extraction of machinery or workers from jobsite images are difficult to be generalized to SOI extraction, because these methods require the visual assets to be consecutively captured from a fixed position [13, 14]. The studies on highway asset segmentations rely on manually labeling datasets for training [15, 16]; hence they are not fully automated. Current labor-intensive practices call for an automatic and robust SOI extraction method.

This study proposes a structure of interest (SOI) extraction algorithm to automate the image preprocessing process for defect detection from unordered photos. This method extracts the structure of interest from a georeferenced photo by registering it to the corresponding building information model (BIM) [17, 18]. The georeferenced photos can be provided by a data collecting device such as an unmanned aerial vehicle (UAV) and smart phone, which is equipped with global positioning system (GPS) and inertial measurement unit (IMU). Since a

63 BIM model is a simulated virtual scene of its real-world counterpart and has a single-color
64 background, it is straightforward to segment a BIM-rendered image into a region of interest
65 that contains the target structure and a background region. If an image is rendered in BIM using
66 the same position and posture information provided by the real photo, the segmented BIM-
67 rendered image is a useful reference for extracting the SOI from its counterpart.
68
69 The contribution is threefold. First, the process of SOI extraction is automated with the
70 proposed method, which has the potential to reduce the workload of image preprocessing and
71 shorten the data analysis cycle for defect detection based on unordered visual assets. Second,
72 the proposed method provides a special solution for extracting different structures of interest
73 from different backgrounds. Using a segmented BIM-rendered image to guide the SOI
74 extraction from the corresponding photo is robust to the influence of the varied background.
75 This strength implies that the proposed method is able to extract SOI from georeferenced
76 photos composed of various types of civil structures with space-time varying backgrounds.
77 Third, a location-based image-to-BIM registration method is proposed, which uses
78 georeferenced information for coarse alignment and realizes precise alignment by image
79 registration. The method does not require a pre-aligned camera [20] at a fixed location and
80 improves the automation level by avoiding human intervention for initial registration [19].
81

82 2 Literature review

83 2.1 Vision-based structural inspection

Stimulated by the emerging techniques in robotics, innovative devices and equipment for vision-based structural inspection have been devised. Many studies have focused on utilizing UAV to perform exterior inspections for detecting structural distress, such as Kim et al. [21], Choi and Kim [2], Morgenthal and Hallermann [3], Eschmann et al. [22], and Kang and Cha [1]. Maeda et al. [23] integrated the smartphone and automobile for road damage detection. Torok et al. [4] presented a robotic platform to collect post-disaster images for damage assessment. These newly-developed platforms are characterized by high mobility and usually have a flexible inspection route. Because of the variations in camera viewpoints and the accompanying uncertainty of illumination status, such platforms generate a massive amount of unordered and unstructured inspection photos that are taken from different view angles and contain both the structure of interest and the irrelevant background.

With the explosion of these unordered inspection photos, processing such visual assets for efficient defect detection has become a demand issue. In conventional practice, engineers are asked to manually identify the structural defects from the captured photos [2, 3, 22]. Such practice is considered time-consuming and labor-intensive, since the amount of data is huge. Therefore, researchers seek to automate the defect detection process by using computer vision and machine learning techniques. One line of work tries to detect damage by analyzing the appearance feature or the image pattern of the defects. Subirats et al. [24] used wavelet transforms for damage detection, while Gavilán et al. [25] used Hough transform to find the damage. Abdel-Qader et al. [8] found the fast Haar transform method to be the most reliable of the four investigated crack-detection techniques. The other line of work leverages deep

learning techniques to directly detect structural defects without manual features selection , and these techniques have been well documented by Kang and Cha [1], Maeda et al. [23], and Cha and Choi [26]. Despite the advancement made in these studies, the irrelevant background pixels in unordered visual assets significantly undermine the algorithm performance. As pointed out by [7], the irrelevant image regions increase the computational complexity and induce extra workload in training the network model. The probability of false positives may also increase, since similar features, which can be mistaken for structural defects (e.g. cracks) can be found in background pixels, as was reported in [23, 26]. As a result, structures of interest need to be extracted from the unordered images to enable a more efficient and accurate detection.

2.2 Image segmentation for ROI extraction

Traditional methods for region of interest (ROI) extraction rely on human prior knowledge. Based on the fact that most artificial landscape, e.g., streets and houses, has straight regions and edges, Mueller et al. [11] developed a segmentation technique for man-made object extraction. Sidike et al. [12] employed a combination of convex hull and morphological operations to yield an accurate building segmentation. These methods take advantage of the explicit appearance features of the objects of interest. However, a certain pattern used for ROI extraction in a specific case might not fit another situation where the target object has a different shape or the background environment changes. These variations in image patterns can cause extra labor requiring manually selecting the extraction features. To automate the workflow, some research efforts have sought to directly segment an image into blocks based on color and texture. As one of the most classical algorithms, JSEG was proposed by Deng and Manjunath

[27] in 2001, which includes two steps, i.e. color quantization and spatial segmentation. Jing et al. [28] and Wang et al. [29], respectively, improved the JSEG algorithm by applying homogeneity analysis and combining directional operators. These methods have avoided human intervention for feature selection, but they often lead to over-segmentation, and fail to provide semantic information to the extracted ROI. As a result, these color and texture-based methods cannot be directly applied to SOI extraction task, which requires explicitly segmenting an image into the background and region of interest.

In the area of civil engineering, image segmentation has been used to extract ROI from the visual assets for assisting construction management and facility maintenance. Chi and Caldas [13] presented a pipeline for extracting heavy equipment from the video captured by jobsite cameras. Azar and McCabe [14] investigated the automatic segmentation and identification of dump trucks from a surveillance video. These studies improved the efficiency of construction management by automating the ROI extraction process. As for facility maintenance, efforts have been made to facilitate the efficient and smart management of highway assets [15, 16, 30]. Golparvar-Fard et al. [15] trained a semantic segmentation model based on semantic texton forests to categorize image pixels into different types of highway assets. Balali and Golparvar-Fard [16] improved the time performance and reduced the labeling efforts required for the segmentation and recognition of highway assets by leveraging a lazy scheme for model training. The aforementioned studies mainly focused on some specific areas in civil engineering, such as construction site or highway management. They either manually relied on labeled datasets for training [15, 16] or consecutive video frames for segmenting moving objects from a static

background [13, 14]; thus, they are difficult to generalize when extracting civil structures with various shapes and appearances from unordered static images that have been captured from different viewpoints.

2.3 Registering 2D images to a 3D digital model

Researchers have been exploring the registration of 2D images (static or dynamic) to a 3D model (e.g. BIM models, CAD) for many years. Using the information retrieved from a 3D model to augment the real-life image innovates the traditional way of progress monitoring and quality assurance. Golparvar-Fard et al. [31] registered time-lapsed photographs collected by a fixed camera to a 4D CAD; then, they superimposed the as-planned model images onto as-built photos to visualize the construction progress. The registration was realized by geometric camera calibration, which calculated the camera intrinsic and extrinsic parameters based on selected feature correspondence between a 2D image and a 3D model. Since the proposed method requires the photo-captured device to be installed at a fixed point with a fixed posture, it falls short of handling the unordered photos collected from different viewpoints and view angles. Karsch et al. [32]and Forsyth et al. [33] investigated the unordered photo registration problem by implementing a user-assisted structure-from-motion (SfM) operation. The method utilized the correspondence points from the 3D mesh model and the initial image (denoted by an anchor image) designated by the user to calculate the camera extrinsic parameters. With the help of the anchor image, the rest of the images that contain common scale-invariant feature points can be aligned with the 3D model. However, this method still requires the unordered photos to have common matched feature points. Based on this content-based image retrieval,

Park et al. [34] proposed a photo registration method that has no limitation on the camera viewpoints; instead, their method relies on a pre-generated dataset of BIM images. However, this image retrieval process is time-consuming.

In general, current practice in 2D-to-3D registration mainly focuses on progress monitoring of construction site, where the collected photos are typically object/building-centric and captured from certain specific points of view. This is not the case for structural inspection using UAV/UGV, since the inspection photos are taken from uncertain locations with various postures. Such inspection practice determines that existing methods are not applicable and calls for a new 2D-to-3D image registration method that can automatically and effectively align the unordered inspection photos with 3D models.

3 Methodology

Fig. 2 illustrates the overall procedure of the proposed SOI extraction algorithm. A real-world photo, along with its georeferenced information (e.g. position, posture, focal length, aspect), is input for the registration operation. Thereafter, a BIM-rendered image aligned with the input photo is obtained. This registered BIM image is then further processed to generate a binary mask. As the final step, the generated mask is used to extract the region of interest from the background.

The reasons for using a BIM model, instead of a plain 3D model, are as follows. First, due to the prevalence of BIM, it is easier to integrate our method with the existing facility management

194 workflow by using BIM as a reference. Second, the visibility of constituent elements can be
195 controlled in a BIM model, which allows only rendering a part of the scenario (i.e., SOI) by
196 hiding irrelevant elements. By contrast, a plain 3D model can only render the entire scene as a
197 whole. Since the registration relies on detecting the feature correspondence from the two types
198 of images (i.e., real-life and BIM-rendered), the BIM model geometry should be as similar as
199 possible to its real-world counterpart; hence, a 350-level of development (LOD) is required.
200

201 **3.1 Location-based image registration to BIM**

202 Fig. 3 shows our proposed method for aligning real photos and BIM-rendered images. This
203 method consists of two main steps: (1) generating a virtual counterpart based on the photo-
204 captured position, posture and optical parameters and (2) image registration with a real-world
205 photo for precise alignment.
206

207 *3.1.1 Rendering BIM correspondence for coarse alignment*

208 A BIM image similar to the real one is rendered and generated by using georeferenced
209 parameters provided by a real-world photo. The parameters include two aspects: (1) physical
210 parameters that describe the position and posture information of the real camera (coordinates,
211 yaw, pitch, and roll) and (2) optical parameters that describe the camera lens and the projection
212 system (field of view, geometry of imaging plane, and resolution). Since the virtual camera in
213 the BIM engine uses different parameters (as shown in Fig. 4), a matching algorithm is devised
214 to translate the real camera parameters to its counterpart.
215

A BIM project usually uses a local coordinate system, while the real camera position is usually recorded in an 84-format WGS (World Geodetic System) [35]. Therefore, the coordinates of the real one is transformed before being used as the virtual camera position, as Eq. (1).

$$\boldsymbol{p}_{BIM} = f_{trans}(\boldsymbol{p}_{WGS84}) \tag{1}$$

where $\boldsymbol{p}_{WGS84}$ and $\boldsymbol{p}_{BIM}$ are respectively camera coordinates in theWGS-84 system and BIM system, i.e. $[lon \quad lat \quad alt]^{\mathrm{T}}$ and $[x \quad y \quad z]^{\mathrm{T}}$. Then, $f_{trans}(\boldsymbol{x})$ is the transformation function. The transformation process typically involves four steps.

The first step is to transform the WGS-84 coordinates to country/region coordinates, which is actually a geometric transformation between two 3D Cartesian coordinate systems. Eq. (2) is the transformation formula.

$$\boldsymbol{p}_{cou} = (1+m_1)\begin{bmatrix} 1 & \varepsilon_Z & -\varepsilon_Y \\ -\varepsilon_Z & 1 & \varepsilon_X \\ \varepsilon_Y & -\varepsilon_X & 1 \end{bmatrix}\boldsymbol{p}_{WGS84} + \begin{bmatrix} \Delta X_1 \\ \Delta Y_1 \\ \Delta Z_1 \end{bmatrix} \tag{2}$$

where $\boldsymbol{p}_{WGS84}$ and $\boldsymbol{p}_{cou}$ are the coordinates in the WGS-84 system and country/region system, respectively. $[\Delta X_1 \quad \Delta Y_1 \quad \Delta Z_1]^{\mathrm{T}}$ is the translation vector; ε_X, ε_Y, and ε_Z are the rotation angle around X axis, Y axis, and Z axis, and m_1 is a scale factor. The value of these parameters can be directly obtained from survey departments.

The next step is to project the 3D country/region coordinates to 2D plane coordinates (as shown in Eq. (3)), which is usually performed with a GIS (geographic information system) software.

$$\boldsymbol{p}_{proj} = f_{proj}(\boldsymbol{p}_{cou}) \tag{3}$$

Here, $\boldsymbol{p}_{proj}$ is the coordinates after projection, and $f_{proj}(x)$ represents the projection function.

In the third step, the projection coordinates are converted to a local coordinate system, which can be expressed as:

$$\boldsymbol{p}_{loc} = (1+m_3)\begin{bmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & \dfrac{1}{1+m_3} \end{bmatrix}\boldsymbol{p}_{proj} + \begin{bmatrix} \Delta X_3 \\ \Delta Y_3 \\ -\zeta \end{bmatrix} \tag{4}$$

where $\boldsymbol{p}_{loc}$ represent the coordinates under the local coordinate system; m_3 is the scale factor; ω is the rotation angle; ΔX_3 and ΔY_3 are the translation values, and ζ is the height anomaly between the quasigeoid and the reference ellipsoid. These parameters can be obtained from local survey departments.

A BIM project often sets one of the control points in the local coordinate system as its project survey point. Eq. (5) shows how to convert the local coordinates to the BIM coordinates.

$$\boldsymbol{p}_{BIM} = \boldsymbol{p}_{loc} + \begin{bmatrix} \Delta X_4 \\ \Delta Y_4 \\ \Delta H_4 \end{bmatrix} \tag{5}$$

where, ΔX_4, ΔY_4, and ΔH_4 are the translation values, which are the opposite of the coordinates of the selected control point.

The camera orientation in BIM is represented by a vector that describes the observing direction and a vector that describes the camera up direction, which can be obtained by Eq. (6) and Eq. (7), respectively.

$$\boldsymbol{v}_{eye} = (\cos\beta\cos(\frac{\pi}{2}\text{-}\alpha), \cos\beta\sin(\frac{\pi}{2}\text{-}\alpha), \sin\beta)^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{v}_{up} = \begin{bmatrix} \sin(\frac{\pi}{2}-\alpha)\sin\varphi - \cos(\frac{\pi}{2}-\alpha)\sin\beta\cos\varphi \\ -\cos(\frac{\pi}{2}-\alpha)\sin\varphi - \sin(\frac{\pi}{2}-\alpha)\sin\beta\cos\varphi \\ \cos\beta\cos\varphi \end{bmatrix} \tag{7}$$

where $\boldsymbol{v}_{eye}$ is a normalized vector of the observing direction; $\boldsymbol{v}_{up}$ is a normalized vector orthogonal to the camera rigid body, which reflects the rotation of the camera around the observing direction; finally, α, β, and φ are yaw, pitch, and roll angle, respectively.

As illustrated by Fig. 4, the virtual camera uses a perspective projection system, which is defined by four parameters, i.e., *fovV*, *aspect*, *near*, and *far*. These parameters are matched to the real camera according to Eq. (8).

$$\begin{bmatrix} fov_V \\ aspect \\ near \\ far \end{bmatrix} = \begin{bmatrix} fov_R \\ w_R / h_R \\ m \\ +\infty \end{bmatrix} \tag{8}$$

where *fovV* stipulates the virtual camera field of view, while fov_R is the correspondence of the real camera; *aspect* is a width-to-height ratio of the projection plane; w_R and h_R are respectively the width and the height of the imaging plane of the real camera; *near* and *far* represent the distance from the origin to the near clipping plane and the far clipping plane, which is equal to a minimal constant *m* and infinity, respectively. Using the above calculated

physical and optical parameters, a BIM image that is coarsely aligned with its real-world counterpart can be generated.

3.1.2 Image registration for precise alignment

The BIM-rendered image needs to be registered for a precise alignment with its real-world counterpart, because the image pairs are usually not consistent with each other due to inaccurate georeferenced information, imaging distortion, and data noise. It should be noted that although the BIM image is rendered with an aspect determined by the resolution of the photo (i.e., w_R / h_R), it usually has a different size than its counterpart, e.g., the virtual one is 800*600 while the real one is 4032*3016. Therefore, the BIM-rendered image is scaled to the same resolution as its counterpart before precise alignment is performed. An affine transform is adopted for image registration, which is illustrated as:

$$\begin{bmatrix} x_{BIMt} \\ y_{BIMt} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & h_x & 0 \\ h_y & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{BIM} \\ y_{BIM} \\ 1 \end{bmatrix} \tag{9}$$

where, $\begin{bmatrix} x_{BIM} & y_{BIM} & 1 \end{bmatrix}^T$ and $\begin{bmatrix} x_{BIMt} & y_{BIMt} & 1 \end{bmatrix}^T$ are respectively the homogeneous coordinates of image pixels before and after transformation. Moreover, $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$, $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 0 & h_x & 0 \\ h_y & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ are transformation matrices, i.e., translation matrix, rotation matrix, scale matrix, and shear matrix, respectively.

The purpose of image registration is to find the optimal transformation matrices for a BIM image to maximize the cost function illustrated by Eq. (10), which, as denoted by mutual information [36], measures the similarity between a BIM image and a real photo.

$$I(R;B) = \sum_{r,b} P_{RB}(r,b) \log \frac{P_{RB}(r,b)}{P_R(r)P_B(b)} \tag{10}$$

where $I(R;B)$ is the mutual information between a real image with intensity r and a BIM image with intensity b; $P_R(r)$ and $P_B(b)$ are the marginal distributions of the real and BIM image intensity. $P_{RB}(r,b)$ is the joint distribution.

The evolutionary algorithm is used to optimize the mutual information with respect to the transformation matrices. The optimization process is inspired by the notion of "survival of the fittest" from Darwinian evolution, and comprises four typical steps, i.e., initialization, evaluation, selection, and variation. In the initialization phase, the initial solutions (denoted by the initial population of individuals) for the transformation matrices are randomly generated. The fitness scores of the individuals in maximizing the mutual information $I(R;B)$ are then evaluated, and the most suitable ones are selected for reproduction of the next generation. At the variation stage, new individuals are bred through crossover and mutation operations for evaluation in the next cycle . The circle of "evaluation-selection-variation" continues until certain termination criteria (e.g., maximum iteration number, or converge) are satisfied. More information on image registration based on mutual information can be found in [36].

3.2 SOI extraction with BIM image mask

311 The registered BIM image aligned with its counterpart is further processed to generate a mask
312 (as shown in Fig. 5(a)). First, the RGB image is converted to a grayscale format. Since a BIM
313 image background is single colored (e.g., plain white), it is straightforward to turn the grayscale
314 image to a binary image by setting the grayscale of the background pixels at zero, while the
315 others are set at one. The morphology-based dilation method is used to fill in the holes in the
316 region of interest.
317
318 The generated mask is leveraged to extract the structure of interest. As illustrated by Fig. 5(b),
319 the extraction is realized by the operation of two image matrices. After image registration, the
320 pixels with the value of one (white color) in the binary mask image constitute an estimated
321 region of interest (denoted by Ω_{roi} in Fig. 5(b)). Conversely, the pixels with the value of zero
322 (black color) represent the estimated background. The matrix of the mask image is denoted by
323 M_{ij}, which represents the value of the pixel at the *i* row and *j* column. The matrix of the original
324 real-world photo is denoted by O_{ij}, which represents the intensity of the pixel at the *i* row and
325 *j* column. The extracted image E_{ij} can be obtained by multiplying the corresponding elements
326 in M_{ij} and O_{ij}. This operation maintains the estimated ROI as the original intensity while
327 turning the background into a universal black color.
328

329 ## 4 Experiment validation

330 ### 4.1 Experiment scheme

331 Two experiments were performed to testify the effectiveness and precision of the proposed
332 algorithm. The target structures of interest are, respectively, the John D. Tickle (JDT) building,

and the Student Union at the University of Tennessee, Knoxville. A smartphone, Xiaomi MI 6, was used as the photo-capture device, which has an equivalent focal length of 27 mm, and an image resolution (width*height) of 4032*3016 pixels. The proposed algorithm was run on a laptop, ASUS VivoBook S15, with an Intel Core i7-8550U processor, and Nvidia GeForce MX150 GPU. The BIM image was rendered by a web-based BIM model viewer — the Autodesk Forge Viewer, which provides a flexible programming interface to customize the rendering view angle, viewpoint, aspect, field of view, etc. The image registration was performed by the MATLAB image processing toolbox.

4.2 Experiment results

4.2.1 Assessment metric

To quantitatively evaluate the experiment results, an index called Intersection over Union (*IoU*) is used to determine the alignment precision between the extracted SOI and the ground truth SOI. The ground truth SOI is denoted by S_{gro}, while the SOI extracted by the proposed method is denoted by S_{ext}. The *IoU* is defined as a ratio of the area of $S_{gro} \cap S_{ext}$ to the area of $S_{gro} \cup S_{ext}$ (as shown in Eq. (11)).

$$IoU = \frac{A(S_{gro} \cap S_{ext})}{A(S_{gro} \cup S_{ext})} \tag{11}$$

where, $A(x)$ is the area of region x, which can be reflected by the quantity of pixels in the region. The larger the *IoU* is, the better the extraction result is in accordance with the ground truth. When *IoU* equals to one, a complete overlap is achieved, which indicates a 100% precision.

4.2.2 Case one — JDT building

Fig. 6 shows the layout of the experiment site at the JDT building and the corresponding BIM model. Six locations were specified to take photographs containing both the structure of interest and the background, i.e., from Loc #1-1 to Loc #1-6 in Fig. 6 (a). At each location, multiple photos were captured at different camera angles. Twenty-three photos were collected in this experiment.

Fig. 7 shows the results for registering the captured real-world photos to the BIM model, where a BIM-rendered image is overlaid onto its counterpart. The difference between the image pairs is represented by different false colors. The region where the superimposed image is bright and the underlying one is dark will look green, while the region with the opposite pattern will look magenta. If both images are dark, the region will be dark. if both images are bright, the region will be bright. The code number at the top of each group describes the photo-captured location and the sequence number. For example, code number "#1-1-1" represents the 1st photo captured at Location #1-1. The row "coarse alignment" presents the alignment level of the raw BIM images that are generated based on georeferenced information, while the row "precise alignment" shows the results of further image registration operation. The *IoU* value is labeled at each image to indicate its quantitative alignment level. As can be seen from the figure, the BIM images at the coarse alignment stage align well in general with the corresponding real-world photos (with an average *IoU* of 78.6%). After the precise alignment (image registration) operation, the alignment level is further improved, wherein significant improvement is observed at image #1-2-1, #1-2-5, and #1-4-1. The average *IoU* of precise alignment is 82.2%.

The SOIs are extracted based on the precise alignment results, as shown in Fig. 8–Fig. 11. The first row of these figures show the captured photos with ground-truth SOIs traced by red lines. The second row and the third row respectively show the segmentation and SOI extraction results. The results exhibit a good performance in general, with the exceptions of # 1-4-3 and # 1-4-4, which show significant deviations from the ground truth.

4.2.3 Case two — Student Union

Fig. 12 show four locations (#2-1, #2-2, #2-3, and #2-4) designated for capturing photos of the Student Union from different view angles. Twenty-one photos were collected.

Similar to case one, Fig. 13 shows the results for registering real-world photos to the BIM model. As can be seen from the figure, most of the BIM images at the coarse alignment stage align well with their corresponding real-world counterparts, except for images #2-2-4, #2-2-5, and #2-2-6. The average *IoU* at this stage is 74.8%. After the precise alignment (image registration) operation, the alignment level is improved, and the average *IoU* increased to 75.9%. Images #2-4-1, #2-3-3, and #2-3-6 witnessed significant improvement in their alignment level, while no obvious change was observed in #2-1-1–#2-1-3, and #2-2-1–#2-2-3. The ground truth (1st row) and the extracted SOI (2nd and 3rd row) based on the results of precise alignments are presented from Fig. 14 to Fig. 17. These alignments exhibit a good performance in general, with the exceptions of images #2-2-4, #2-2-5, and #2-2-6.

4.3 Performance assessment of the proposed method

399 The *IoU* value is used as a metric to evaluate SOI extraction accuracy. The frequency
400 distribution histogram of the *IoU* values of all 44 groups of images collected from the two
401 experiments is shown in Fig. 18. The average *IoU* value is 79.21%, and a total of 36 images
402 attained an *IoU* value of over 70%, accounting for 81.8%. By comparison, the OASGR [7], a
403 state-of-the-art ROI extraction algorithm, achieved an average *IoU* value of 68.9% on the
404 Pascal VOC Challenge 2007 dataset [37]. In [38], an extraction with an *IoU* value larger than
405 50% is regarded as a correct result. The average *IoU* value of our method is higher than the
406 OASGR *IoU* value and above the criteria set by [38], which demonstrates a quite promising
407 performance. In terms of efficiency, the average running time of our method for processing
408 each image was about 140 s, which can be further improved by using parallel computation or
409 a high-performance workstation.
410
411 **4.4 Discussion**
412 The proposed structure of interest extraction algorithm is validated by the experiment results.
413 Among all 44 testing photos, the average *IoU* value is 79.2%, and those with an attained *IoU*
414 value of over 70% account for 81.8%. The proposed method can achieve an accuracy that is
415 better than the state of the art, and does not require model training or human intervention.
416
417 The efficacy of the proposed method in automating the SOI extraction process is verified. Both
418 experiments achieved an *IoU* value of over 75%, which demonstrates the proposed method can
419 work properly with no dependence on the appearance and style of the target structure.
420 Equivalent high performance has been attained on images with different illumination (e.g.,

421 strong light in #1-4-1 and overcast in #1-3-5), and different types of elements in the background
422 (e.g., trees in #1-2-5, irrelevant buildings in #1-1-1, and a complex environment in #2-1-2).
423 The results indicate the robustness of the proposed method for dealing with complex and varied
424 backgrounds. In other words, the proposed method is not designated for a specific type of target
425 structure with a specific surrounding environment but provides a generic algorithm suitable for
426 georeferenced photos once the corresponding BIM model is accessible. As a result, the
427 execution of the algorithm is automated without involving any human intervention or prior
428 knowledge for feature selection.
429
430 The image registration can compensate for the deviation between the real-world photo and
431 BIM-rendered image caused by inaccurate georeferenced information and imaging distortion,
432 thereby improving the alignment accuracy (with an average 3.6% and 1.1% of improvement
433 for case one and case two, respectively). The increase of *IoU* value after image registration can
434 go up 10% to 20%, as shown in images #1-2-1, #1-4-1, #2-3-2, and #2-4-1. However, one
435 observation in the experiments is that under certain circumstances when the angle between the
436 line-of-sight and the structure of interest is small, the alignment accuracy did not increase
437 significantly, as shown in images #1-5-1, #2-1-1, and #2-2-1 in Fig. 7 and 13. Some photos
438 witnessed a decrease of *IoU* value after registration, e.g., Images #1-3-3, and #1-3-4 in Fig. 7,
439 and images #2-3-3 and #2-4-3 in Fig.13. In the case of images #1-3-3 and #1-3-4, the deviation
440 between the real building and the BIM model (see Fig. 19(a)) induced a registration failure,
441 which then reduced the *IoU* value. In terms of images #2-3-3 and #2-4-3, the image registration
442 actually improves the alignment level of the exterior contour of the building, as can be seen

443 from Fig. 19(b). However, the transformation of the BIM image for achieving this alignment
444 induced a deviation of other parts in the images, which reduced the intersection between the
445 ground truth and extracted result (shaded part in Fig. 19(b)), and then led to the decrease of the
446 *IoU* value.
447
448 The experiments show several undesirable extraction results (as shown in images #1-4-3, #1-
449 4-4 and #2-2-4 to #2-2-6), which have *IoU* values of less than 40%. These extraction failures
450 are due to the imprecise georeferenced information provided by the real-world photos. For
451 example, with interferences from magnetic disturbances, the detected yaw value deviated
452 considerably when photo # 2-2-4 was taken. As a result, the generated BIM image with the
453 inaccurate yaw value shows great deviation from the real-world photo (see Fig. 19(c)), which
454 is difficult to compensate by subsequent image registration.
455
456 **5. Conclusions**
457 Structure of interest (SOI) extraction is a critical preprocessing step for improving the
458 performance of computer vision-based structural inspection. As an attempt to automate the
459 process, this study proposes to extract SOI by registering a georeferenced photo to a
460 corresponding building information model. The method for aligning real photos and BIM-
461 rendered images is explored based on georeferenced information and image registration. The
462 SOI in a real-world photo is subsequently extracted by converting the registered BIM image
463 into a binary mask. The experiments carried out at the John D. Tickle building and the Student
464 Union at the University of Tennessee, demonstrated the potential performance of the proposed

method in extracting SOI from images with a complex and varied background. Since no manual efforts are needed for finding suitable patterns, the SOI extraction process is automated with the proposed method.

Further research efforts are needed to address the following limitations. First, the received GPS signal and IMU data can yield unreliable georeferenced information, due to occlusion or electromagnetic interference. In this case, the deviation would be too huge to be compensated by image registration. As a result, measures should be taken to guarantee the robustness and performance of the GPS localization and IMU measurement. Second, although the proposed method exhibits high performance in removing the irrelevant background, it falls short of processing a foreground. In fact, the vision-based defect detection result would also be affected by the foreground pixels overlaid on the region of interest. One possible solution is to combine the proposed method with color-and-texture-based segmentation. After the background is subtracted using our proposed method, the foreground pixels can be removed based on texture or color heterogenicity between the foreground and target structure.

References

[1] D. Kang, Y.-J. Cha, Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging, Computer-Aided Civil and Infrastructure Engineering, 33 (2018) 885-902.

[2] S.-s. Choi, E.-k. Kim, Ieee, Building Crack Inspection using Small UAV, 2015 17th International Conference on Advanced Communication Technology2015, pp. 235-238.

[3] G. Morgenthal, N. Hallermann, Quality Assessment of Unmanned Aerial Vehicle (UAV) Based Visual Inspection of Structures, Advances in Structural Engineering, 17 (2014) 289-302.

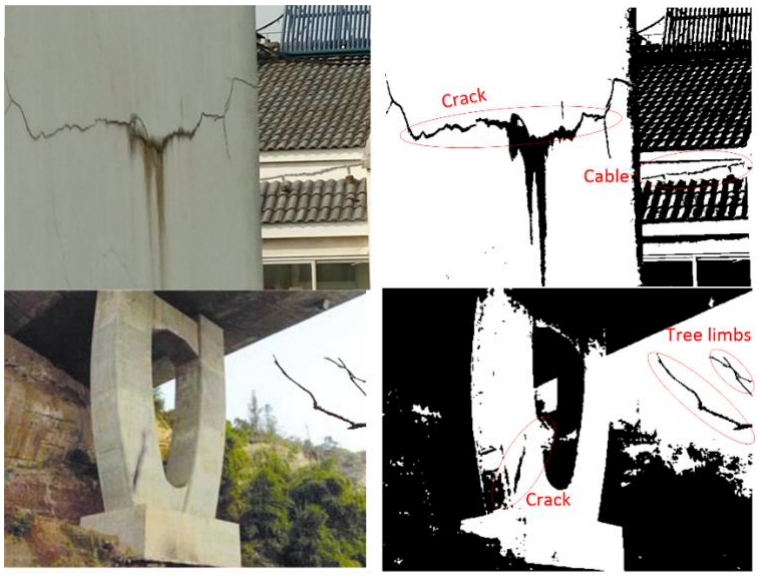
[4] M.M. Torok, M. Golparvar-Fard, K.B. Kochersberger, Image-Based Automated 3D Crack Detection for Post-disaster Building Assessment, J Comput Civil Eng, 28 (2014).

[5] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, P.J.A.E.I. Fieguth, A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure, 29 (2015) 196-210.

[6] C.M. Yeum, Computer vision-based structural assessment exploiting large volumes of images, (2016).
[7] Q. Li, Z. Wei, C. Zhao, Optimized Automatic Seeded Region Growing Algorithm with Application to ROI Extraction, International Journal of Image and Graphics, 17 (2017).
[8] L. Abdel-Qader, O. Abudayyeh, M.E. Kelly, Analysis of edge-detection techniques for crack identification in bridges, J Comput Civil Eng, 17 (2003) 255-263.
[9] C.M. Yeum, J. Choi, S.J. Dyke, Autonomous image localization for visual inspection of civil infrastructure, Smart Mater Struct, 26 (2017).
[10] R. Huang, B. Yang, F. Liang, W. Dai, J. Li, M. Tian, W. Xu, A top-down strategy for buildings extraction from complex urban scenes using airborne LiDAR point clouds, Infrared Phys Techn, 92 (2018) 203-218.
[11] M. Mueller, K. Segl, H. Kaufmann, Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery, Pattern Recogn, 37 (2004) 1619-1628.
[12] P. Sidike, D. Prince, A. Essa, V.K. Asari, Ieee, AUTOMATIC BUILDING CHANGE DETECTION THROUGH ADAPTIVE LOCAL TEXTURAL FEATURES AND SEQUENTIAL BACKGROUND REMOVAL, 2016 Ieee International Geoscience and Remote Sensing Symposium2016, pp. 2857-2860.
[13] S. Chi, C.H. Caldas, Automated Object Identification Using Optical Video Cameras on Construction Sites, Computer-Aided Civil and Infrastructure Engineering, 26 (2011) 368-380.
[14] E.R. Azar, B. McCabe, Automated Visual Recognition of Dump Trucks in Construction Videos, J Comput Civil Eng, 26 (2012) 769-781.
[15] M. Golparvar-Fard, V. Balali, J.M. de la Garza, Segmentation and Recognition of Highway Assets Using Image-Based 3D Point Clouds and Semantic Texton Forests, J Comput Civil Eng, 29 (2015).
[16] V. Balali, M. Golparvar-Fard, Segmentation and recognition of roadway assets from car-mounted camera video streams using a scalable non-parametric image parsing method, Automat Constr, 49 (2015) 27-39.
[17] Q. Lu, S. Lee, Development of a Semi-Automatic Image-based Object Recognition System for Reconstructing As-is BIM Objects based on Fuzzy Multi-Attribute Utility Theory, CIB W78 2016 Conference, 2016.
[18] Q. Lu, S.J.J.o.C.i.C.E. Lee, Image-based technologies for constructing as-is building information models for existing buildings, 31 (2017) 04017005.
[19] C. Kropp, C. Koch, M.J.A.i.C. König, Interior construction state recognition with 4D BIM registered image sequences, 86 (2018) 11-32.
[20] Y. Ibrahim, T.C. Lukins, X. Zhang, E. Trucco, A.J.A.E.I. Kaka, Towards automated progress assessment of workpackage components in construction projects using computer vision, 23 (2009) 93-103.
[21] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, H.-J. Jung, Application of Crack Identification Techniques for an Aging Concrete Bridge Inspection Using an Unmanned Aerial Vehicle, Sensors-Basel, 18 (2018).
[22] C. Eschmann, C.M. Kuo, C.H. Kuo, C. Boller, Unmanned aircraft systems for remote building inspection and monitoring, 6th European Workshop on Structural Health Monitoring 2012, EWSHM 2012, July 3, 2012 - July 6, 2012, German Society for Non-Destructive Testing eV, DGZfP e.V., Dresden, Germany, 2012, pp. 1179-1186.
[23] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata, Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images, Computer-Aided Civil and Infrastructure Engineering, 33 (2018) 1127-1141.
[24] P. Subirats, J. Dumoulin, V. Legeay, D. Barba, Ieee, Automation of pavement surface crack detection using the continuous wavelet transform, 2006 Ieee International Conference on Image Processing, Icip 2006, Proceedings2006, pp. 3037-+.
[25] M. Gavilán, D. Balcones, O. Marcos, D.F. Llorca, M.A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, A.J.S. Amírola, Adaptive road crack detection system by pavement classification, Sensors-Basel, 11 (2011) 9628-9657.
[26] Y.-J. Cha, W. Choi, O. Buyukozturk, Deep Learning-Based Crack Damage Detection Using Convolutional Neural

Networks, Computer-Aided Civil and Infrastructure Engineering, 32 (2017) 361-378.
[27] Y.N. Deng, B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, Ieee T Pattern Anal, 23 (2001) 800-810.
[28] F. Jing, M.J. Li, H.J. Zhang, B. Zhang, I. Ieee, Unsupervised image segmentation using local homogeneity analysis, 2003.
[29] Y.-G. Wang, J. Yang, Y.-C. Chang, Color-texture image segmentation by integrating directional operators into JSEG method, Pattern Recogn Lett, 27 (2006) 1983-1990.
[30] J.P. Wu, Y. Tsai, Enhanced roadway inventory using a 2-D sign video image recognition algorithm, Computer-Aided Civil and Infrastructure Engineering, 21 (2006) 369-382.
[31] M. Golparvar-Fard, F. Pena-Mora, C.A. Arboleda, S. Lee, Visualization of Construction Progress Monitoring with 4D Simulation Model Overlaid on Time-Lapsed Photographs, J Comput Civil Eng, 23 (2009) 391-404.
[32] K. Karsch, M. Golparvar-Fard, D. Forsyth, ConstructAide: Analyzing and Visualizing Construction Sites through Photographs and Building Models, Acm T Graphic, 33 (2014).
[33] D.A.K. Forsyth, Kevin; Golparvar-Fard, Mani 4D VIZUALIZATION OF BUILDING DESIGN AND CONSTRUCTION MODELING WITH PHOTOGRAPHS, United States, 2015.
[34] J. Park, H.B. Cai, D. Perissin, Bringing Information to the Field: Automated Photo Registration and 4D BIM, J Comput Civil Eng, 32 (2018).
[35] D. Liu, J. Chen, D. Hu, Z. Zhang, Dynamic BIM-augmented UAV safety inspection for water diversion project, Computers in Industry, 108 (2019) 163-177.
[36] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, P. Suetens, Multimodality image registration by maximization of mutual information, IEEE transactions on medical imaging, 16 (1997) 187-198.
[37] T.P.V.O.C. Homepage, The PASCAL Visual Object Classes Homepage.
[38] M.J. Ferguson, Seongwoon; Law, Kincho H, Worksite Object Characterization for Automatically Updating Building Information Models, The 2019 ASCE International Conference on Computing in Civil EngineeringGeorgia Institute of Technology, Atlanta, Georgia, United States, 2019.

563 **Figures**



564

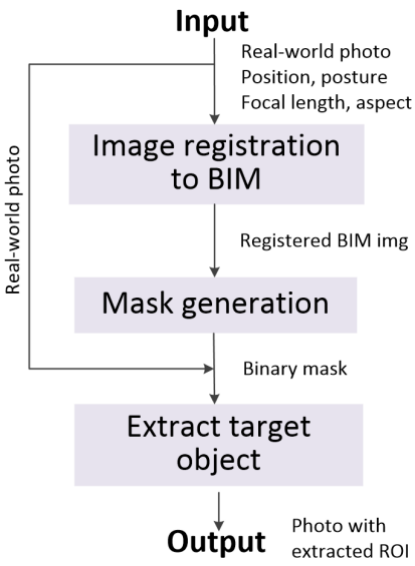
565 **Fig. 1.** Unordered images with a background that has similar patterns which can be mistook

566 for structural cracks: (a) RGB images of building exterior wall and bridge pier and (b

567 corresponding binary image with cable that could be considered as a crack as well as stains

568 caused from water dripping out of cracks.

569



570

571 **Fig. 2.** Overall procedure of the proposed method.

572

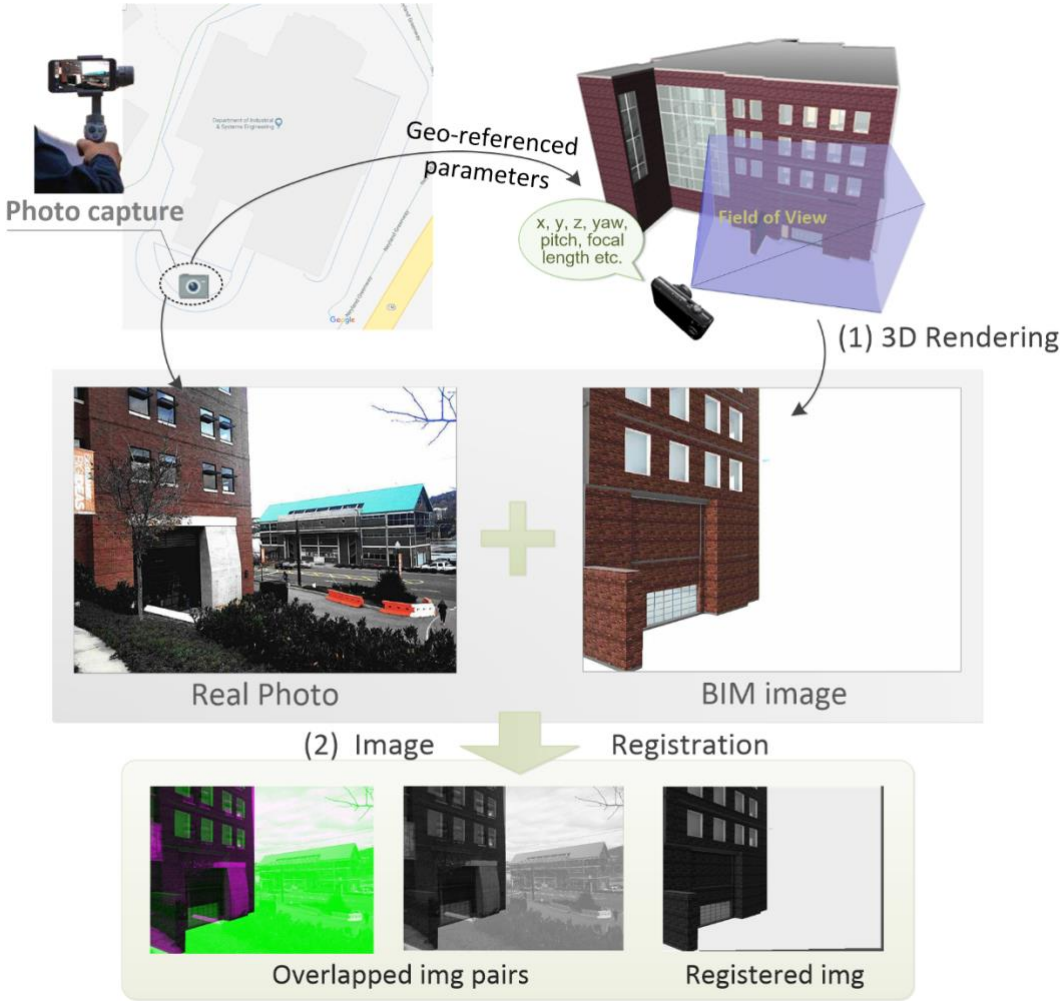


Fig. 3. Location-based image registration to BIM model.

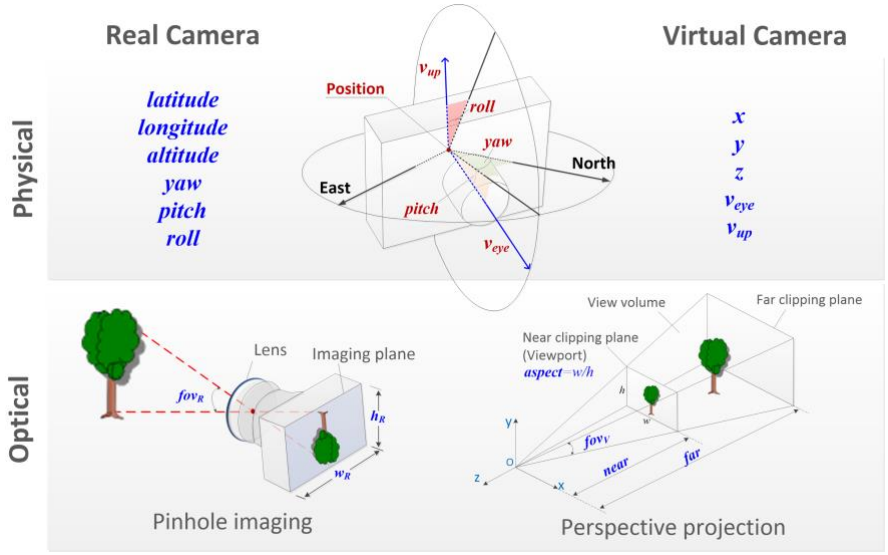


Fig. 4. Different physical and optical parameters used by real and virtual camera.

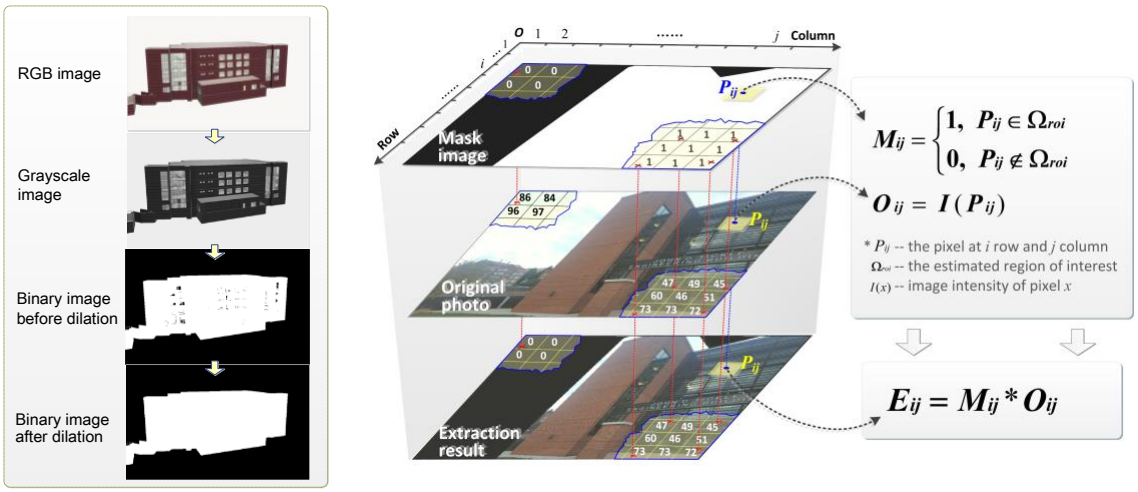


Fig. 5. (a) Turning a BIM-rendered image into a mask and (b) ROI extraction with mask operation.

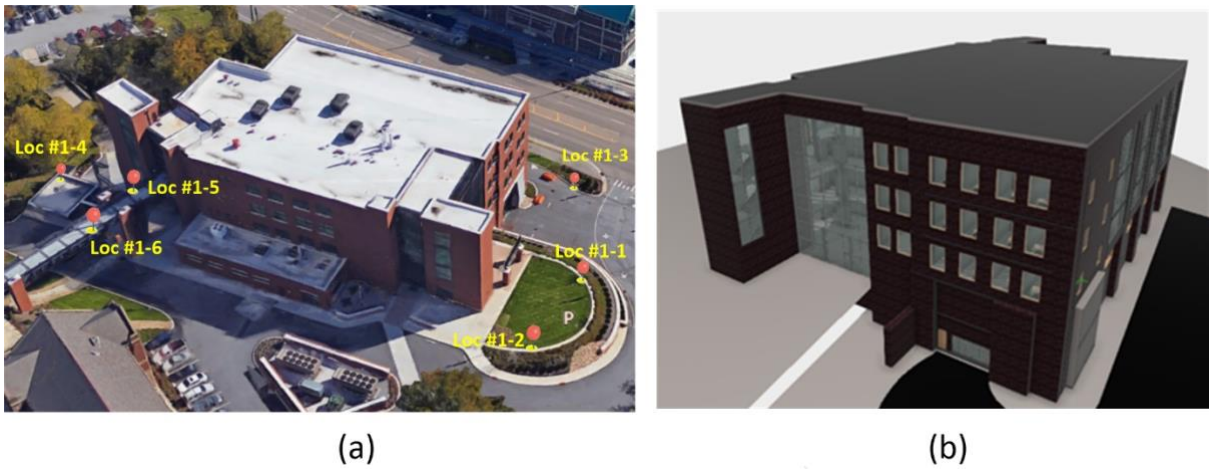


Fig. 6. Layout of experiment site at JDT building: (a) 3D simulation model from Google Earth and (b) 3D BIM model.

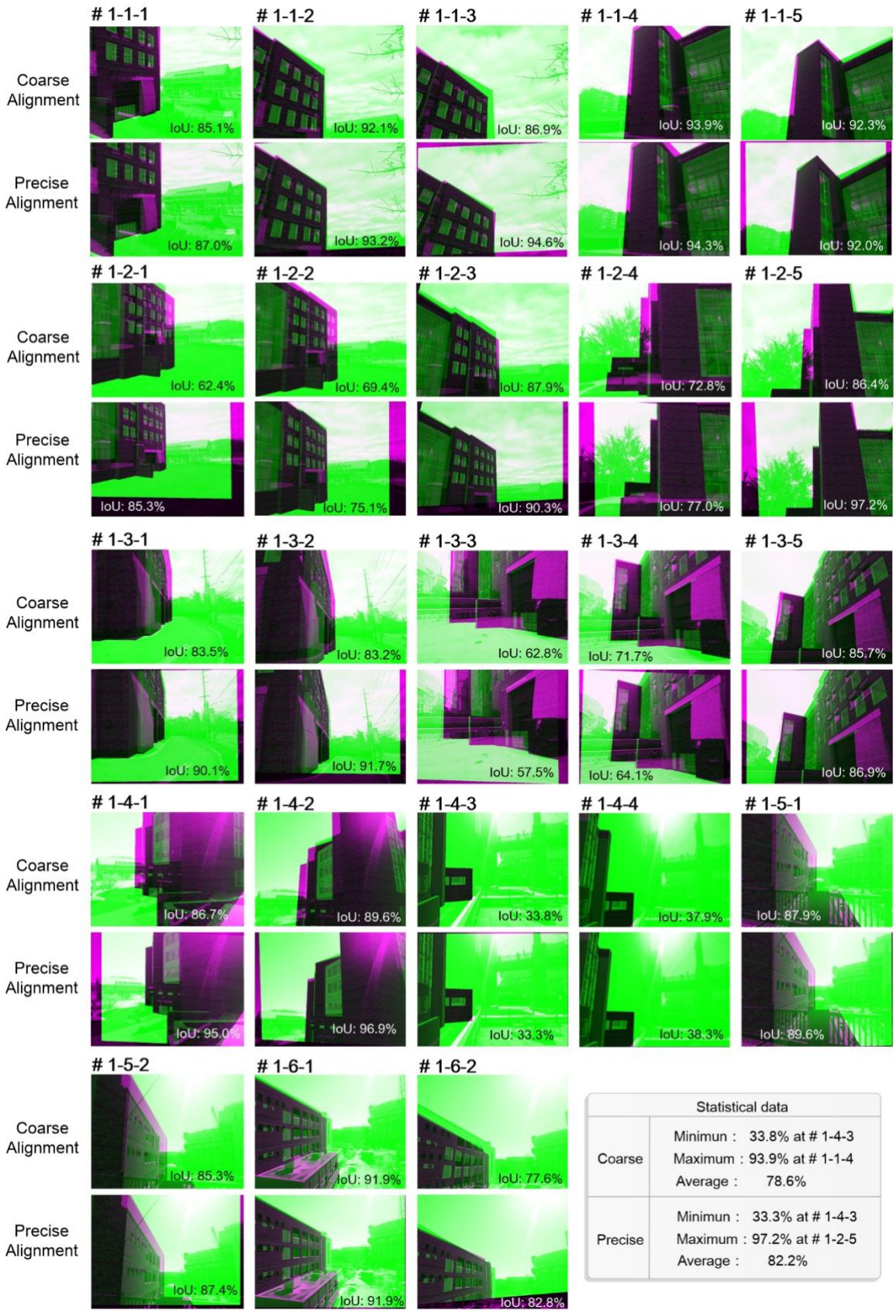


Fig. 7. Results for registering the captured photos to the BIM model of JDT building.

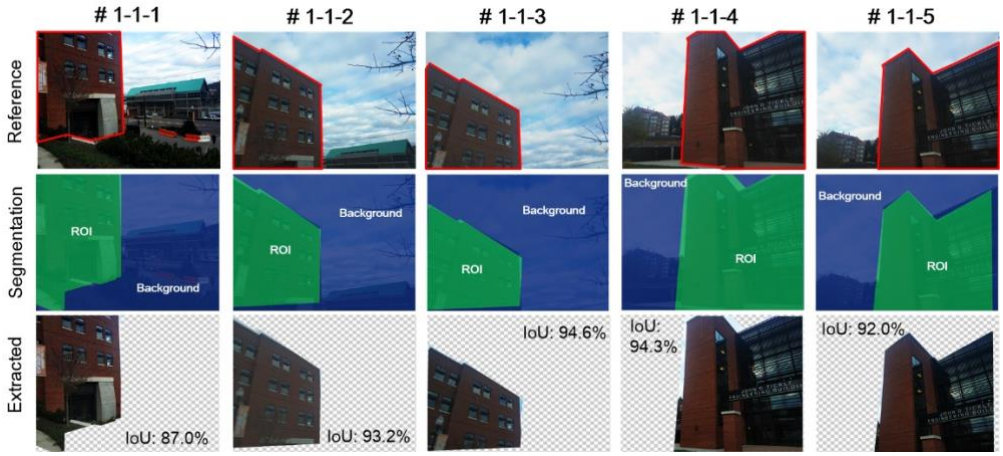


Fig. 8. SOI extraction results for location #1-1.

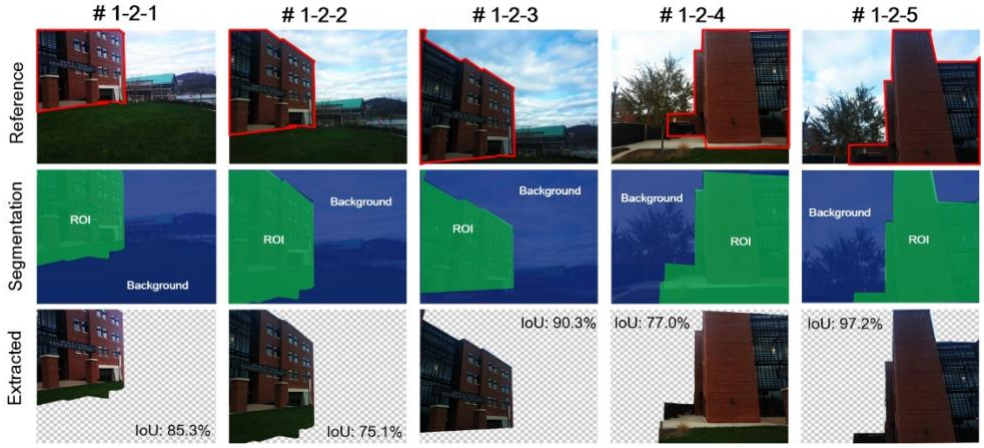


Fig. 9. SOI extraction results for location #1-2.

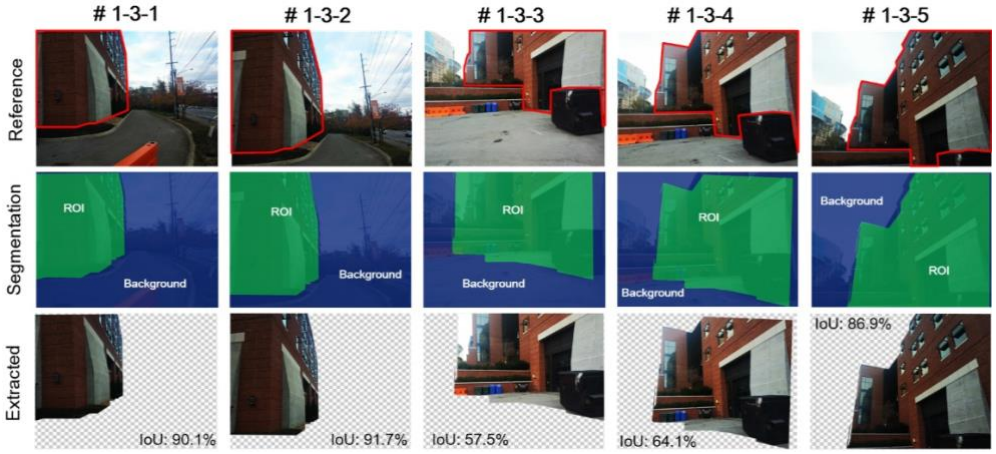


Fig. 10. SOI extraction results for location #1-3.

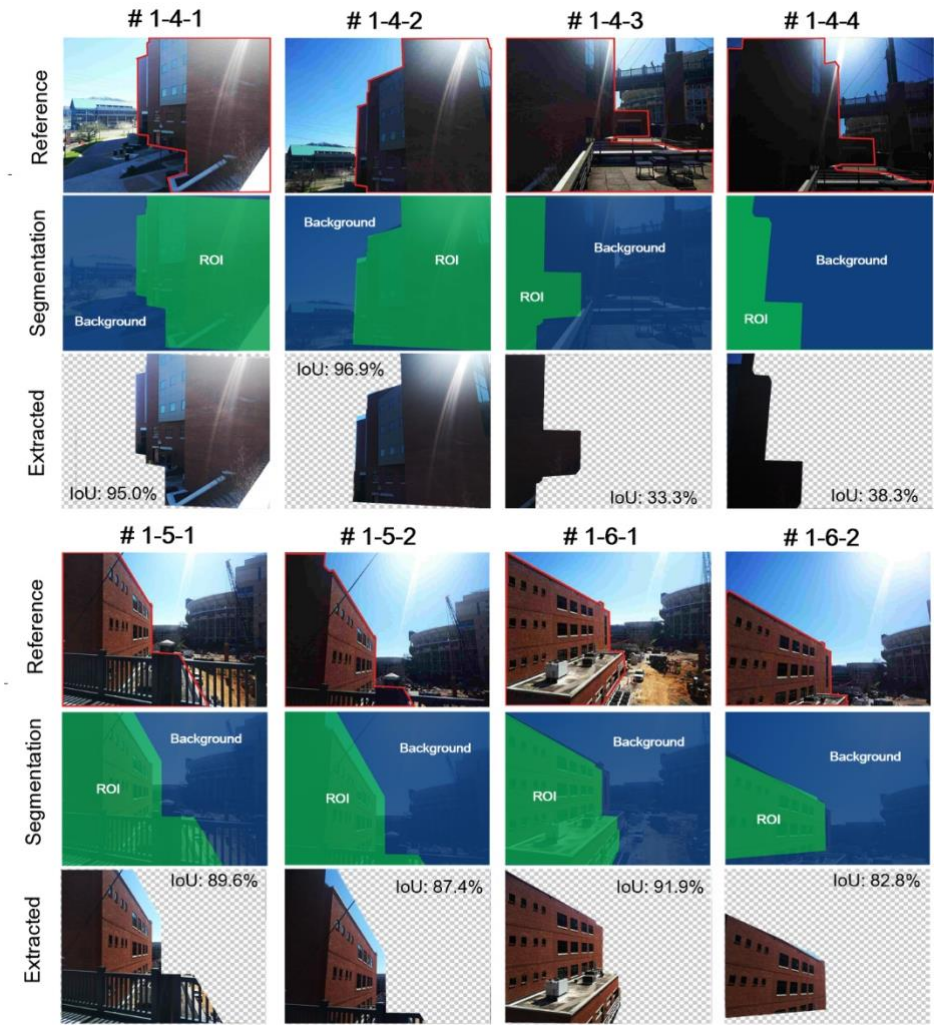
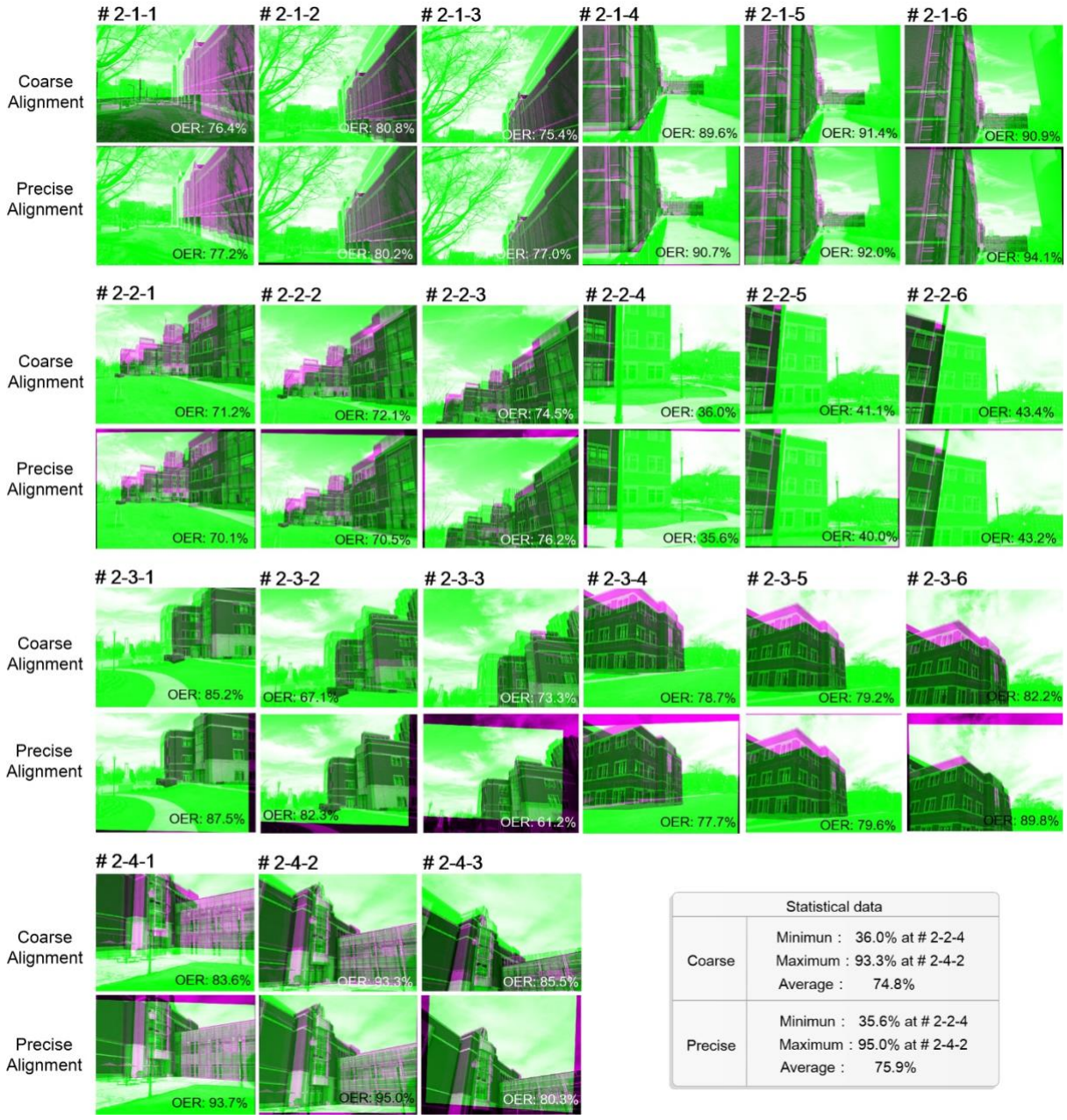


Fig. 11. SOI extraction results for location #1-4– #1-6.

Fig. 12. Layout of experiment site at Student Union: (a) 3D BIM model and (b) street map of plan view.

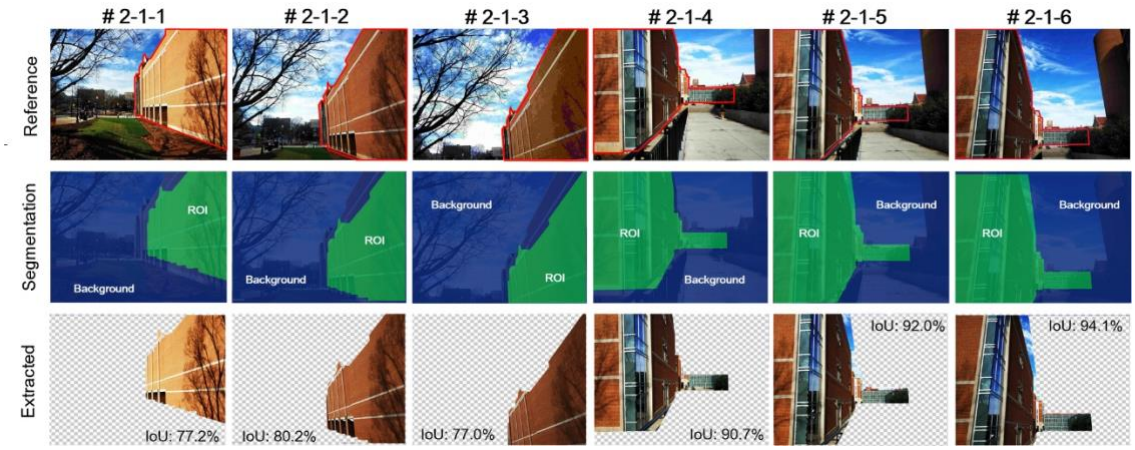
605



606

607 **Fig. 13.** Registration results of the captured photos to the BIM model of the student union.

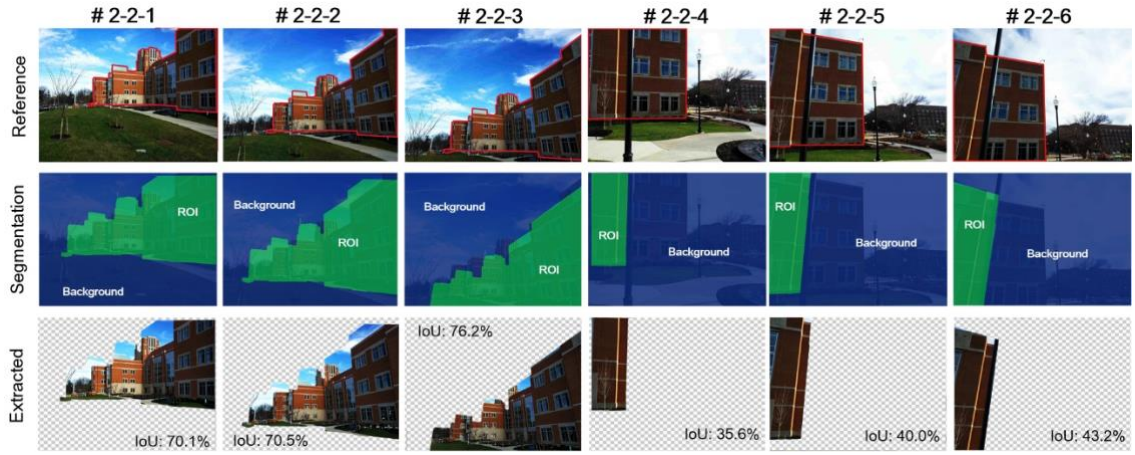
608



609

610 **Fig. 14.** SOI extraction results for Location #2-1.

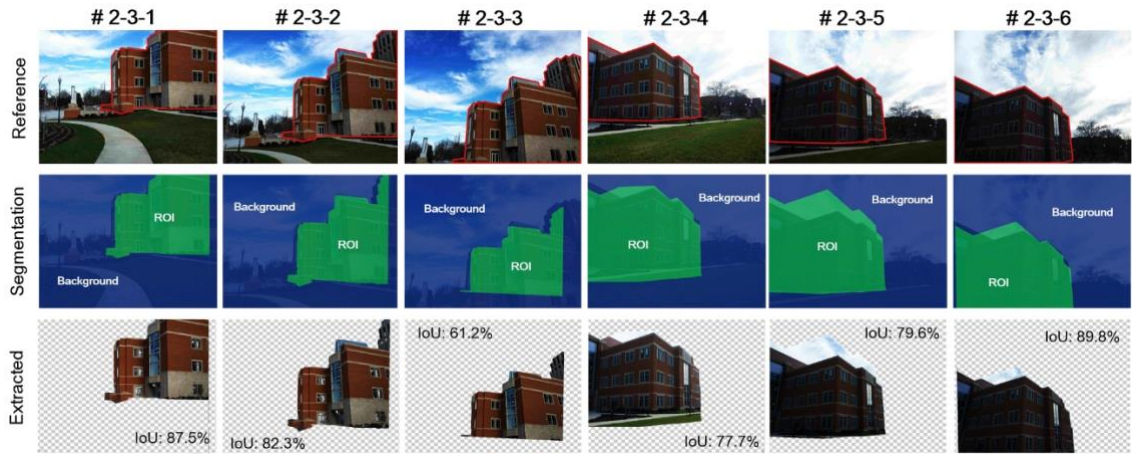
611



612

613 **Fig. 15.** SOI extraction results for Location #2-2.

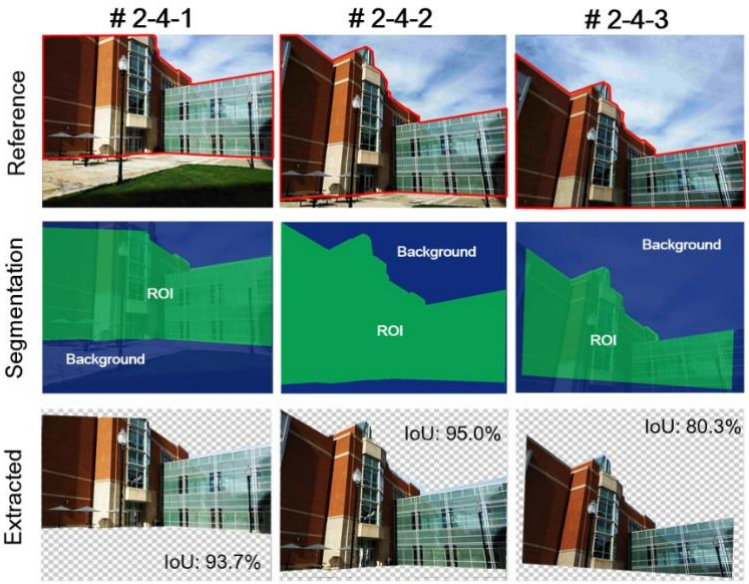
614



615

616 **Fig. 16.** SOI extraction results for Location #2-3.

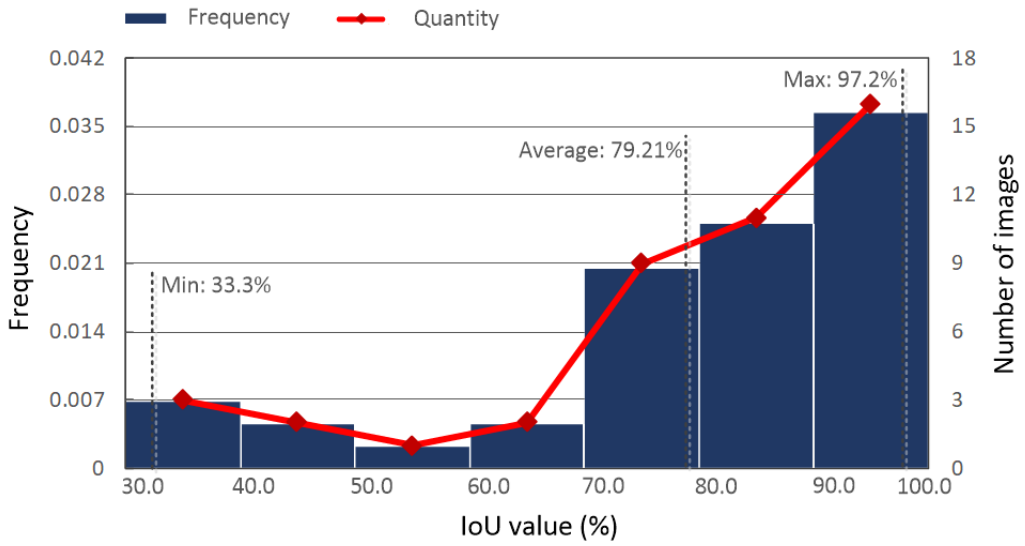
617



618

619 **Fig. 17.** SOI extraction results for Location #2-4.

620



621

622 **Fig. 18.** Frequency distribution histogram of the experiment results.

623

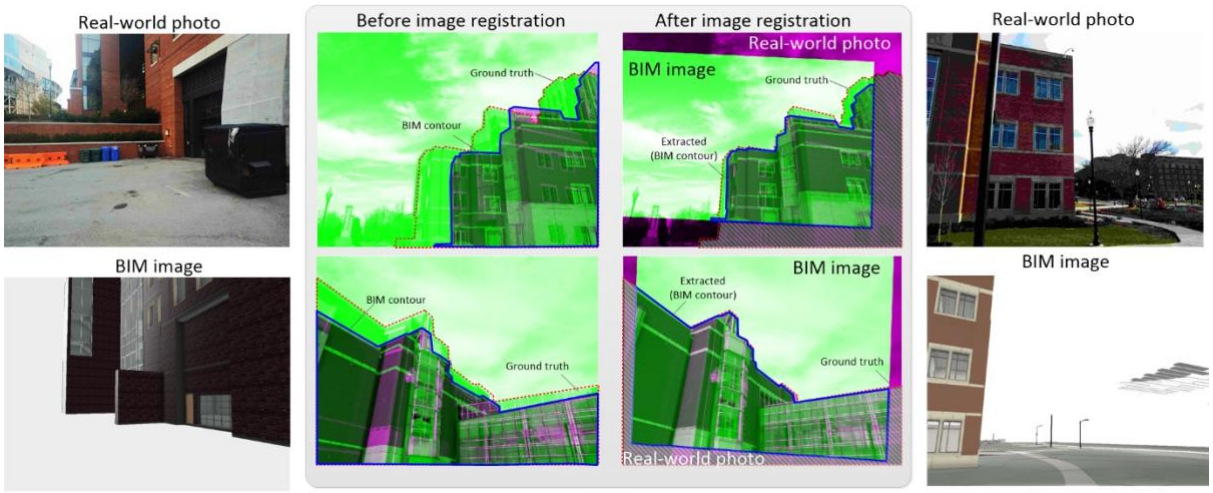


Fig. 19. (a) Real-world photo and BIM image at Location #1-3, where the real building has a terrace connecting the exit of the second floor, while the BIM model does not; (b) partial alignment reduces the intersection between the ground truth and the extracted result, and (c) real-world photo at Location #2-2 and its counterpart generated with inaccurate yaw value.