A Phase-Congruency-Based Scene Abstraction Approach for 2D-3D Registration of Aerial Optical and LiDAR Images

Yasmine Megahed, Ahmed Shaker, and Wai Yeung Yan, *Senior Member, IEEE*

***Abstract*—Registration of aerial images to enrich 3-D light detection and ranging (LiDAR) points with radiometric information can enhance the capability of object detection, scene classification, and semantic segmentation. However, airborne LiDAR data may not always come with on-board optical images collected during the same flight mission. Indirect georeferencing can be adopted, if ancillary imagery data are found available. Nevertheless, automatic recognition of control primitives in LiDAR and imagery datasets becomes challenging, especially when they are collected on different dates. This article proposes a generic registration mechanism based on using the phase congruency (PC) model and scene abstraction to overcome the stated challenges. The approach relies on the use of a PC measure to compute the image moments that determine the study scene's edges. Potential candidate points can be identified based on thresholding the image moments' values. A shape context descriptor is adopted to automatically pair symmetric candidate points to produce a final set of control points. Coordinate transformation parameters between the two datasets were estimated using a least squares adjustment for four registration models: first- (affine), second-, third-order polynomials, and direct linear transform models. Datasets covering different urban landscapes were used to examine the proposed workflow. The root-mean-square error of the registration is between one and two pixels. The proposed workflow is found to be computationally efficient especially with small-sized datasets, and generic enough to be applied in registering various imagery data and LiDAR point clouds.**

***Index Terms*—Aerial imagery, airborne light detection and ranging (LiDAR), Canny edge detector, image registration, phase congruency (PC), scene abstraction, shape context.**

I. Introduction

RECENT studies indicate the fact that the world is undergoing the largest wave of urban growth in history [1]. North America is one of the most urbanized regions with 82% of its population living in urban areas as reported in 2018 [2]. This tangible urban sprawl consumes available resources and leads to a shortage of public services. As a result, it requires definitive administrative plans to precisely assess the quality of current urban areas and to develop new strategies to cope with the estimated urbanization in the future. Besides, this urban expansion highlights the necessity of resource-efficient and technology-driven cities known as smart cities [3]. From a geomatics perspective, a digital city is the main component of a smart city, and relies fundamentally on urban spatial data integration from spaceborne, airborne, and terrestrial sensors, along with GIS management [4]. The complementary properties of LiDAR and optical data are vital for detecting and analyzing urban settlements, since these techniques are capable of revealing the study scene with the underlying spatial, temporal, and topological information and patterns [5], [6]. Therefore, accurate registration of multisensor data becomes essential to fully maximize the potential of integrating different technologies. Ideally, an on-board camera together within airborne LiDAR systems can simultaneously collect both 2-D optical imagery and 3-D point cloud data, where direct georeferencing can be applied to register both datasets on a projected coordinate system. Indirect georeferencing can be adopted if the optical images and the LiDAR point clouds are, respectively, collected on different dates or during different missions.

Remote sensing image registration relies on the use of common control points being identified between the target and reference datasets to construct a coordinate transformation model [7], [8]. If there exists a lack of sharp and permanent control points due to the nature of the study scene, other registration primitives, including lines, curves, and polygons, can also be utilized [9]– [11]. Researchers abandoned the direct identification of common points when dealing with LiDAR data since they represent laser footprints rather than identifiable points in corresponding imagery data.

Scale-invariant feature transform (SIFT), line segment detector (LSD), attraction field map (AFM), and Canny operator are common feature extraction algorithms that have been used for the determination of control primitives. SIFT is a point extraction method that is applied to match different views of a scene [12]. It accommodates noise in data, and the extracted point features are invariant to translation, rotation, scale, and partially invariant to

Manuscript received March 26, 2020; revised June 4, 2020, August 26, 2020, and September 23, 2020; accepted October 8, 2020. Date of publication October 26, 2020; date of current version January 6, 2021. This work was supported in part by the Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (RGPIN-2015-03960), in part by the FCE Start-Up Fund of The Hong Kong Polytechnic University (BE2U), and in part by the Early Career Scheme by the Research Grants Council of the Hong Kong Special Administrative Region under Project 25213320 . *(Corresponding author: Yasmine Megahed.)*

Yasmine Megahed and Ahmed Shaker are with the Department of Civil Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: yasmine.megahed@ryerson.ca; ahmed.shaker@ryerson.ca).

Wai Yeung Yan is with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University,, Hong KongDepartment of Civil Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: waiyeung.yan@polyu.edu.hk).

Digital Object Identifier 10.1109/JSTARS.2020.3033770

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

change in illumination [13]. It detects points of interest and then accumulates the calculated statistics of local intensity gradient directions in a local neighborhood that surrounds each point of interest [13]. Once extracted from reference images, point features are stored in a database for the subsequent comparison with the corresponding features of a new image. A probability matching uses minimum Euclidean distance for a point on the new image to locate its nearest neighbor in the database and targets it as a counterpart [12]. Li *et al.* [14] used the SIFT-based approach to register multispectral, multidate, and multisensor satellite images. It caused significant mismatched points due to pixels of the same area having different intensities. Hence, the authors proposed scale-orientation joint restriction criteria for robust point matching by excluding incorrect paired-up points. They were able to obtain a registration root-mean-square error (RMSE) less than two pixels. Nasir *et al.* [15] registered real-world terrestrial images of low SR, as an essential step in super resolution, where a high-SR image is generated from a sequence of low-SR images. The author applied SIFT to extract candidate point features; however, they used belief propagation for point matching instead of the minimum Euclidean distance method, which was previously found to ignore the descriptors' geometric characteristics. Additionally, they used the random sample consensus (RANSAC) to eliminate mismatched points. No quantitative evaluation was performed for the registration results; however, the authors reported satisfactory results based on visual inspection.

The LSD method detects line segments defined as zones' linear boundaries on an image, where the grey level sharply changes from white to black, or vice versa. It constructs a level-line field, where the level-line angle of each pixel is calculated. Pixels with the same level-line angle are clustered as a line-support region, whose direction is considered as its minimum bounding rectangle. Pixels within the rectangle with a level-line angle value close to that of the rectangle are considered aligned points. For each rectangle, the total number of pixels along with the number of aligned points determines whether it is a potential line segment or not. The model can be applied to any digital image without manipulating its parameters [16]. Akinlar and Topal [17] introduced the edge lines (EDLines) algorithm, where enhancements to the LSD method were developed to overcome long execution times, discontinuous line segments, and poor performance when processing images with white noise. Lyu and Jiang [18] applied the EDLines approach to register multitemporal remote sensing images, using line segments as control parameters. The authors merged segment fragments that belong to the same line segment, and a line descriptor with gradually changing bands for segment matching. They reached a registration accuracy of around one pixel.

The most recent development to the LSD method is introduced by Xue *et al.* [19], namely the AFM method. It accommodates for incorrect edge pixel identification, incomplete line segments detection, excessive postdetection steps to isolate line edges from edge pixels, and approximate representation of a line segment by a set of connected pixels causing the zig-zag effect, commonly known in digital image processing. The single-stage algorithm relies on computer vision concepts and deep learning techniques to include the entire image pixels in the production of line segments and relates a line segment map to its spatial regions. In this way, the AFM method deals with the LSD approach's drawbacks as region coloring problems, allowing for the application of semantic segmentation technologies to further enhance the LSD method. The AFM algorithm achieved advances in accuracy and execution time; however, it has not been tested in image registration yet.

Canny is a gradient-based edge detector. It applies Gaussian low-pass filter to reduce noise prior to gradient estimation for edge sharpening. The Sobel kernel computes the gradient of image pixels in both directions. Gradient magnitude determines the edge's strength, whereas gradient direction determines the direction of maximum intensity change for each pixel. Edges are then sharpened by eliminating pixels except for those of large strength. Finally, a double threshold is used to separate strong, weak, and suppressed edges [20]. It is widely used in practice due to its good signal-to-noise ratio, accurate edge extraction [21], and ease of application by adjusting only the Gaussian filter variance and the threshold parameters [22]. However, the determination of optimal thresholds is a key problem, and the results of using multiple values cannot be quantitatively assessed [23]. This is problematic knowing that the algorithm fundamentally relies on the thresholds' values, and the double threshold method itself has poor self-adaptability, resulting in incomplete edge detection information [22], [24].

Habib *et al.* [25] used 3-D straight-line features as registration primitives for the coregistration between photogrammetric and LiDAR data. Linear features in the image space were represented by a set of intermediate points, with an additional constraint provided. One endpoint of the photogrammetric line segment must lie along the vector connecting the endpoints defining the LiDAR line segment, whereas in the LiDAR space, the authors manually identified planar patches, checked their planarity and removed outliers, and determined endpoints by intersecting neighboring planar patches with different orientations. The identified features were matched manually, and the approach was tested on terrestrial and aerial datasets for a limited number of features, all of them have only linear characteristics. Kwak *et al.* [26] registered an aerial image to LiDAR data for a study zone that is known for its flat roof surfaces. Hence, the authors used plane roof centroids as control primitives to register both data. In imagery space, they extracted building boundaries, derived straight line equations from their endpoints to detect corner points, identified closed polygons, and finally calculated the 2-D coordinates of each polygon's centroid. In LiDAR space, they constructed a triangulated irregular network (TIN) from nonground points, eliminated TINs belonging to trees, and computed the planar coordinates of roof centroids. The z-coordinate was considered as the median value of LiDAR points inside each roof polygon. The identified centroids in both datasets were paired up manually.

Mitishita *et al.* [27] registered photogrammetric and LiDAR data using centroids of isolated and thin-wall-free rectangular roof planes. For LiDAR data, the authors manually extracted points located in the proximity of the building in question, roughly calculated the coordinates of the rectangle, interpolated roof points onto a grid, and selected on-roof interpolated LiDAR points. They considered x and y coordinates of a roof centroid as the mean x and mean y coordinates, respectively, of the interpolated roof points, whereas z-coordinate was the mean z of the interpolated points along with the profiles closest to the roof borders. For image data, the authors determined rectangular roof centroids by calculating the coordinates of the intersection of the two diagonal straight lines defined by the four roof corners, which were also identified manually. The bundle adjustment of the centroids did not show better results than the experiments, which used presignalized control points collected using a GPS survey. Zhang *et al.* [28] constructed an approach based on the fact that correct registration parameters result in a back-projection of a LiDAR point cloud object being within its boundary in the image. The authors used control objects as registration primitives. They manually extracted objects from LiDAR data and their corresponding boundaries from optical data represented by polygons, back-projected the 3-D LiDAR points' coordinates to the 2-D optical image, and computed the registration parameters to compare them to the provided parameters to assess the registration quality. The study accommodates for man-made features and natural scenes; nevertheless, it highlighted concerns regarding the accuracy of object extraction.

Yang and Chen [29] attempted to automate the registration process. The authors registered sequent images and LiDAR data captured by mini-UAVs, based on control points as primitives. They extracted building outlines from LiDAR data, converted them into rectangular polygons, and back-projected those extracted outlines on the images to locate their corresponding features. Feature detection on images was not straightforward though. A building area was determined, building patches were separated, and contours were extracted, in addition to some regularization processes. The authors matched extracted edges and corners, then used endpoints in course registration. Despite the automated approach, its scope is limited to urban buildings with rectangular or L-shape roofs. Additionally, misalignment problems were reported in areas of occlusion or inaccurate outline extraction.

Zheng *et al.* [30] registered close-range photogrammetric and terrestrial LiDAR data based on two photogrammetric principles: forward intersection and the closest point. The first states that conjugate rays from multiple images' points intersect at the same 3-D point, which theoretically should be located on the LiDAR data surface, according to the second concept. The authors matched corresponding image points using the SIFT operator and solved the 3-D coordinates of points of intersection using collinearity equations. Finally, they applied bundle adjustment to calculate the error in these coordinates as well as the error in the external orientation and lens distortion parameters, coordinates of the principal points, and principal distances. Each of the latter resembles the normal vector of the best fitting plane of the closest LiDAR points to a 3-D point, determined by the K-D tree algorithm. The methodology needs no feature extraction from LiDAR data, decreases nonrigid deformations caused by lens distortion, and increases the accuracy by eliminating image points with large distances from the LiDAR data surface from the registration process. However, it was tested on a small stone carving and cave walls. In addition, it failed to function on flat LiDAR data surfaces.

Huang *et al.* [31] extended the method to register large-scale aerial optical photos and airborne LiDAR data. It was tested on four datasets covering a mix of urban and rural areas. Three of them had both data acquired at the same time while the aerial photos and LiDAR data were collected on different dates in the fourth dataset. To accommodate the large size of airborne LiDAR data and maintain effective processing time, 3-D LiDAR points were partitioned into tiles, and the search for the closest LiDAR point was carried out tile by tile and in order. Additionally, the authors used the principal component analysis algorithm to select the most significant gross points resulted from discontinuity and different data ranges and targeted them for elimination. Checkpoints' coordinates on images were determined by the DPGrid software, whereas their corresponding coordinates on LiDAR data were calculated by intersecting two lines or three planes. The study could reach a subpixel registration accuracy within an efficient workflow; however, still performs poorly in the case of flat LiDAR data.

Cui *et al.* [32] applied a linear-based transformation model to register panoramic images and 3-D LiDAR points acquired from a mobile mapping system. The authors used the EDISON edge detector to extract edge pixels from the images, whereas they extracted linear features from labeled LiDAR points out of buildings, pole-like, and curbs objects. A region growing segmentation was applied on building points to generate planar segments, and the points were projected on the 3-D plane model of each segment, whose boundary points were detected and fitted into 3-D lines by conventional least squares (LS) with constraints: lines must pass by outmost points, and only vertical and horizontal lines with a sufficient number of points are considered. Pole-like objects were extracted using a percentile-based pole recognition algorithm. Each object was divided into subsegments, each with an enclosing rectangle and a 2-D centroid. Neighboring subsegments with maximum diagonal length were kept, and the points were fitted into 3-D lines knowing the centroids and the maximum and minimum Z-values. Vertical curbs and the ground surface were considered to intersect at curb lines. A RANSAC method fitted curb points into a plane parallel to the Z-axis, after the removal of outliers. The points were fitted into 2-D lines, and the ground heights were considered their Z-values. A RANSAC paradigm removed the outliers in the interrelated lines extracted from camera images and LiDAR points. The 3-D lines extracted from LiDAR data were projected to the images to determine their corresponding 2-D lines, given that a ray from a camera's perspective center to an image edge pixel intersects a line in the world coordinate system at a point. Euclidean distance in the image space was used as the similarity metric in the nonlinear LS, was used to estimate the corresponding image coordinates of the 3-D lines' endpoints. The authors reported a satisfactory visual evaluation; however, disturbing elements, calibration errors in laser scanners, and finding correlated features in both datasets affected the RMSE value.

It is distinct that most of the research work related to registration of LiDAR and optical datasets for urban regions mainly relies on the matching of linear features, such as flat rectangular or L-shaped building roofs, in addition to other sharp-identified elements, such as corners, road intersections, and landmarks. In this way, the registration process is not only limited to residential urban areas, where these objects exist in abundance but also it is constrained by acquiring both datasets at the same time to ensure matching congruent features. Moreover, some of the presented approaches demand the application of several subprocesses to precisely pair the registration primitives. To illustrate, some of the aforementioned studies implemented algorithms for feature extraction, outlier removal, edge detection, and shape fitting, in addition to other mathematical and statistical models to identify the primitives. Beside how complex and time consuming these methods are, the bundle adjustment of matched primitives does not always guarantee satisfying registration results compared to conventional registration using control points collected by traditional means of GPS surveys, for example. The reason as reported by many studies is usually related to the inaccuracies in the extraction of building roofs, as the so-far applied registration methodologies are sensitive to the position of the building relative to its neighboring blocks. They were found to yield proper results when the entire building is isolated. Furthermore, some studies extract and/or pair registration primitives manually.

This article proposes a more generic point-based registration workflow to register 3-D airborne LiDAR point clouds and 2-D aerial photos, acquired for different urban morphologies: industrial, residential, and coastal shore, and at different dates. The accommodation for a wider variety of urban land uses comes from the abstraction of urban scenes to their main elements, which are commonly found in both datasets. These elements could be any complete or partial objects in the urban view being investigated (e.g., streets, canals, corridors, walls, lakes, sidewalks, curbs, tree crowns, vehicles, etc.). The adopted workflow is straightforward; first, 3-D LiDAR data are converted to 2-D imagery, then the phase congruency (PC) model is implemented on both LiDAR and imagery datasets as cell points for feature perception. Second, candidate control points (CCPs) can be determined by adjusting a predefined threshold range. Depending on the nature of investigated urban scenes and height and intensity values of LiDAR data, this step may be subjected to a manual intrusion that makes the workflow more interactive, where scenes are abstracted to their primary elements. In this case, points located on the border of these features are targeted as a set of CCPs. The shape context descriptor (SCD) method is implemented to match potential points. Finally, a bundle adjustment based on LS is applied to the matched pairs of points to estimate the registration parameters. This study reflects great contributions to 2D-3D image registration due to multiple reasons: the wide spectrum of imagery sources and urban layouts, which the presented work can deal with, the simplicity of the proposed approach, finally and most importantly, the noncompulsion for the data to be acquired on the same date, and the primitives not have to be derived using only traditional urban features, such as rectangular building roofs.

II. Methods

A. Overall Workflow

The methodology implemented in this study is illustrated in Fig. 1. The airborne LiDAR dataset is first converted to a 2-D image based on either intensity (I) or height (H) records, whichever is more representative of the urban morphology being investigated. The PC model runs on the LiDAR and aerial 2-D images' cell centroids to calculate a PC measure for each point, as a feature significance indicator [7].

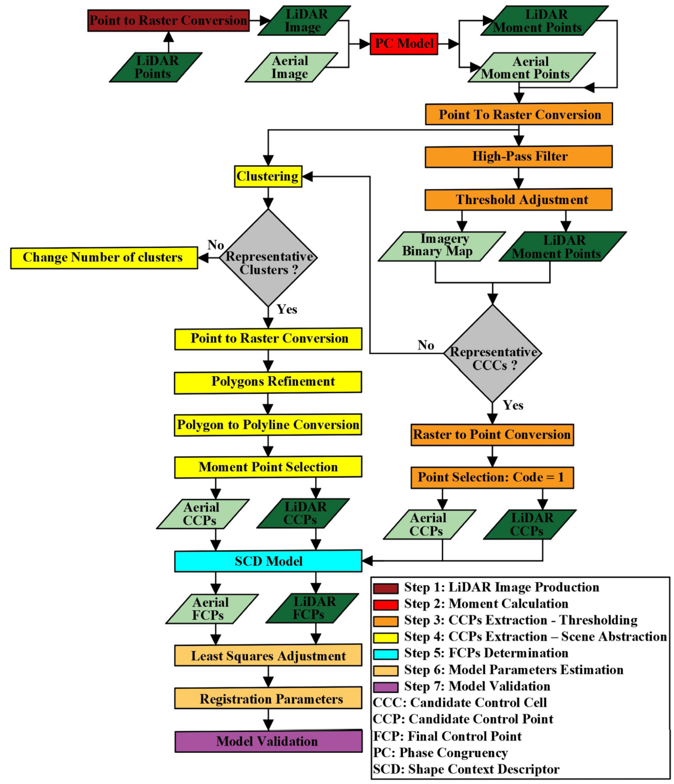

Fig. 1. Proposed workflow for a 2D-3D registration.

Moments of each point are calculated knowing the PC measure in different orientations. These moments determine the inclination of a point to be an edge or a corner point. The visualization of the output moment points shows patterns of different elements located in the two images. After the conversion of these moment points to raster, a high-pass filter is run to ease the isolation of CCPs. A threshold tolerance is adjusted on the filter output raster, where pixels within the range are assigned a value of 1 (candidate control cells—CCCs), or 0 otherwise. CCCs are then checked for representativeness. The results of this visual inspection determine whether the rest of the registration workflow continues automatically or semiautomatically.

If the CCCs are found to be consistently distributed around congruent features within the LiDAR and aerial images, the approach remains automatic and CCCs are targeted as CCPs after being converted to points, whereas should CCCs be heterogeneous in both data, the segment of the registration approach in regards to CCPs determination encounters some interactivity to downgrade the scene's details to basic polylines commonly found in both images. In this case, clustering is implemented on the moment raster to have it partitioned to its main elements. The predefined number of clusters is the key factor to be manipulated until achieving satisfying segments that are representative enough when visually comparing both images. The clustered raster is then converted to polygons, which are further refined for a better representative scene abstraction by merging, splitting, and deleting. Afterward, final polygons are converted to polylines. Moment points within a buffer around these detected edges are considered the CCPs. The buffer value is the minimal that allows the detection of moment points solely on the polylines. Later, CCPs are input to the SCD model to be matched in pairs as final control points (FCPs). Subsequently, an LS adjustment solves the transformation parameters of different registration models. The difference between original and calculated FCPs' coordinates determines the model development accuracy. Finally, registration models are validated by comparing original and calculated coordinates of several checkpoints obtained separately from the FCPs.

B. Conversion of 3-D LiDAR Point Clouds to 2-D Images

In this study, a binning interpolation is run on the LiDAR data points' intensity or height attribute to determine the cell values of the output image. It has been addressed in different research works, such as in [33]–[35]. A binning interpolation identifies the 2-D geographic extent of LiDAR points and then constructs a grid, where the cell size is the output image's SR identified by the analyst. This interpolation algorithm requires a cell assignment method to determine the output cell value based on the points that fall within its extent, in addition to a void filling method to determine the value of cells that do not contain any LiDAR points. The "Average" cell assignment and "Linear" void fill approaches are applied. The first assigns the mean value of all points that are located within the cell, whereas the second calculates the value of empty cells by triangulating across void areas and applying linear interpolation on the triangulated point to determine the cell value. This process is particularly important when laser dropouts are found in water bodies and river streams.

C. Features Detection Using the PC Model

The local energy model [36] was developed to detect image features in 1-D based on the postulate that edges and corners are located on points, where the Fourier components of a signal are maximized in phase. The PC function normalizes the local energy function to provide a dimensionless measure of PC [37]. The advantages of applying the PC model over other gradient-based approaches in the detection of edges and corners are summarized as follows [38].

(1) Simplicity of concept, as the model looks for points in the image where a high degree of order in the frequency domain occurs, with no assumptions about the waveform shape.
(2) The model measures the significance of a point by a dimensionless quantity that ranges from 0 to 1 indicating out-of-phase and in-phase points, respectively. This absolute measure eases the process of setting threshold values for the selection of CCPs. In this way, the predefined threshold range can be applied universally over various classes of images.
(3) The PC measure is invariant to changes in image illumination and spatial magnification, unlike gradient-based edge detectors that target points of maximal intensity gradient as significant points. Consequently, gradient-based algorithms determine threshold values empirically, which makes these methods case subjective.
(4) The model abandons the Fourier transform in obtaining local frequency information, as it does not consider the spread of frequencies at points of congruency. For instance, assuming analyzing a single-tone signal like a sine wave, it always has a PC value of one everywhere, since it will be in perfect congruence with itself. To overcome this shortcoming, the model uses wavelets for local frequency analysis, where a scalable window moves along the signal being analyzed, and the spectrum for each position is calculated. The process is repeated with a change in the window size for every new cycle [39]. This multiresolution spatial-scale representation is achieved by using a bank of filters, each filter results from rescaling a specific wave shape, and aims to select particular frequencies of the signal for analysis. The wavelet bank ensures a controllable frequency range through the multiple scales, at which PC is calculated effectively with high spatial localization. Hence, features are not separated from their surroundings, but rather, they are related. For example, a feature is deemed to be more significant if it has a high PC value over a wide range of spatial scales than a feature with similar congruency that is obtained over a limited scope of spatial scales. Optimal wavelet bank design is the one that maintains a smooth sum of spectra with the minimum number of filters, for efficient computations.
(5) The model accommodates filters (wavelet bank) in the 2-D by applying a spreading function(s) across the filter in a perpendicular direction to its orientation. To depict, a 1-D analysis is applied over several orientations, and results are combined to obtain a PC measure for the entire 2-D image.
(6) Noise, as a major drawback of the PC model, is reduced by introducing a compensation term that devalues the PC of a feature based on its noise's local energy magnitude, independently in each orientation.

The PC model is constructed according to the following steps, and equations are as described in [37] and [38]:

$$\mathrm{PC}(x) = \sum_n W(x) \cdot \lfloor A_n(x)(\cos(\phi_n(x) - \bar{\phi}(x)) - |\sin(\phi_n(x) - \bar{\phi}(x)| - T) \rfloor \Big/ \sum_n A_n(x) + \epsilon \quad (1)$$

where $\mathrm{PC}(x)$ is the PC measure at point x, which varies between 0 and 1 indicating no and full congruency, respectively, at a certain orientation θ. It is a matrix that has the same dimension of the image being analyzed. The value between $\lfloor\rfloor$ is considered if positive, otherwise, it is replaced by 0. $A_n(x)$ is the point amplitude at scale n, given by

$$A_n(x) = \sqrt{(I(x) * M_n^e)^2 + (I(x) * M_n^o)^2} \quad (2)$$

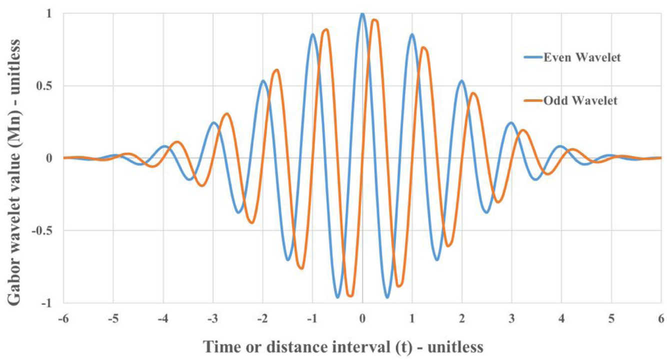

Fig. 2. Sample of a Gabor wavelet.

where $I(x)$ is the pixel value at the point x. Symmetric/asymmetric wavelets are required to preserve local frequency information at a particular point, which is obtained using logarithmic Gabor filters over multiple scales and orientations. Gabor wavelet consists of two filters in quadrature: sine and cosine waves, each modulated by a Gaussian function (see Fig. 2).

M_n^e and M_n^o are the even and odd wavelet values at scale n, respectively. Each is the cosine and the sine component, respectively, of a Gabor wavelet equation

$$M_n = \exp(-\pi(t - J\Delta t)^2/\alpha^2)\cos(2\pi k\Delta f(t - J\Delta t)) + \exp(-\pi(t - J\Delta t)^2/\alpha^2)\sin(2\pi k\Delta f(t - J\Delta t)) \quad (3)$$

where t is an element of a vector, which has an odd dimension equal to $2n + 1$. The vector's central value is 0, which increases by $\Delta t = 1$ in $+x$ direction and decreases by $\Delta t = 1$ in $-x$ direction. This plots the function from $-n$ to $+n$. Since the analysis is carried out in the spatial domain, it is worth to mention that Δt is a distance interval. a^2, k, and Δf are constants: 20, 1, and 1, respectively. j is the wavelet displacement from the origin, and it is equal to 0 in this case [40]. $[M_n^e]$ and $[M_n^o]$ have the same dimensions of $[t]$. Each element of both matrices is calculated by substituting in the even and odd wavelet equation, respectively, with its corresponding t value. Both $[M_n^e]$ and $[M_n^o]$ represent the filters by which images are convoluted to calculate $A_n(x)$, which is a matrix with the same dimensions of the examined image. $\phi_n(x)$ is the phase of point x at scale n. It is a matrix with the same dimensions of the input image, and is denoted by

$$\phi_n(x) = \arctan 2(I(x) * M_n^e, I(x) * M_n^o) \quad (4)$$

$\bar{\phi}(x)$ is the mean phase at point x, given by

$$\bar{\phi}(x) = \frac{1}{N}\sum_{n=1}^{N}\phi_n(x) \quad (5)$$

where n is a certain scale within the total number of applied scales N. T is the noise compensation term, given by

$$T = K\bar{A}_0''\frac{1 - (\frac{1}{m})^N}{1 - \frac{1}{m}} \quad (6)$$

where K is a scaling factor ($\sim$ 2.5), m is a scale factor between successive filters ($\sim$ 1.5), and $\bar{A}_0''$ is the mean noise at the smallest scale filter ($n = 0$). $\bar{A}_0''$ is calculated via

$$\bar{A}_0'' = \exp(\overline{\log A_0(x)}). \quad (7)$$

Since the PC is considered significant only over a wide range of frequencies, $W(x)$, as listed in (8), is a weighting function is a weighting function that reduces the PC value at locations where the spread of filters is limited, by implementing a measure of the spread of filter response $S(x)$. Both $S(x)$ and $W(x)$ are matrices of dimensions equal to the analyzed image. $W(x)$ is thus given by

$$W(x) = \frac{1}{1 + \exp(g(c - S(x)))} \quad (8)$$

where c is the cutoff filter value that is equal to 0.4. Meaning that a frequency zone in a filter is considered narrow if it is $\leq$ 0.4, and thus, the PC values are penalized when they occur in this slim range of frequencies. g is an amplitude gain factor, equals 10, that is used to accommodate for the sharpness of the filter at the cutoff value. $S(x)$ is calculated from

$$S(x) = \frac{1}{N}\left(\frac{\sum_n A_n(x)}{\epsilon + A_{\max}(x)}\right) \quad (9)$$

where $A_{\max}(x)$ is the amplitude of filter pair having maximum response at x. ϵ is a constant larger than 0 ($\sim$ 0.01) to avoid the division by 0.

D. Determination of CCPs

Maximum and minimum moments M and m account for obtaining local frequency information in different orientations, PC(θ). M indicates significant PC in one orientation, whereas m implies significant PC in more than one direction. Therefore, large values of M mark edge features, whereas large values of m mark corner features. M and m are given by

$$M = \frac{1}{2}\left[c + a + \sqrt{b^2 + (a - c)^2}\right] \quad (10)$$

$$m = \frac{1}{2}\left[c + a - \sqrt{b^2 - (a - c)^2}\right] \quad (11)$$

where a, b, and c are calculated from

$$a = \sum[\text{PC}(\theta)\cos\theta]^2 \quad (12)$$

$$b = 2\sum[\text{PC}(\theta)\cos\theta]\cdot[\text{PC}(\theta)\sin\theta] \quad (13)$$

$$c = \sum[\text{PC}(\theta)\sin\theta]^2. \quad (14)$$

With proper threshold values for M and m, feature points of edges and corners can be defined as CCPs.

E. Matching CCPs Using SCD

The SCD model [41] is applied to automatically match point pairs out of CCPs as FCPs. It measures the geometric similarity to find corresponding matches in two sets of points in a graph-matching framework, which is invariant to changes in translation and scale. These two point sets are the CCPs determined in both aerial and LiDAR images. For each CCPs set, distance and azimuth angle from each point to the rest of the points within the same set are calculated. Distances are normalized by dividing by the median distance to account for different scales between point sets if exist.

A log-polar coordinate system is established for each point to the remaining ones in the same set. The horizontal axis ranges from zero distance to the maximum normalized distance, and the vertical axis starts from zero azimuth angle to the maximum angel value (2π). Both axes are divided into bins to form a grid/matrix, where each cell hosts the number of points that have normalized distances and azimuth angles to the point under investigation, within the cell's horizontal and vertical boundaries, respectively.

Finally, the correlation between each matrix in the first set of CCPs and the matrices in the other CCPs set is calculated after flattening the matrices. The correlation value ranges from 0 to 1, indicating no and a full correlation between a pair of points, respectively. Pairs of a correlation higher than a predetermined threshold value (e.g., 0.95) are targeted as FCPs, which are subsequently used for registration.

F. Registration of Aerial and LiDAR Data

Four common transformation models are examined to perform the data registration: first- (affine), second-, and third-order polynomials, in addition to the direct linear transform (DLT) model. These four empirical models were applied for two reasons. First, the downloaded aerial photos lack metadata files; hence, no information about the data acquisition is provided, which eliminates the application of a physical model, such as collinearity equations. Second, using empirical registration models makes the proposed methodology generic enough to fit different imagery data regardless of their acquisition technique.

Linear parametric LS method is applied, if a high correlation cutoff value is found during the aforementioned matching process. The coordinate transformation parameters using polynomials are estimated from

$$X = (A^T A)^{-1}(A^T b) \tag{15}$$

where X is the vector of the unknown coordinate transformation parameters, b is the vector of observations constructed by the FCPs' coordinates on aerial data, and A is the design matrix that contains the FCP on LiDAR data. To account for outliers, iteratively reweighted LS can be adopted to provide a robust estimation of the transformation parameters [42]. The parameters of the DLT model are determined by using nonlinear parametric LS in

$$X = (J^T J)^{-1}(J^T k) \tag{16}$$

where X is the vector of unknowns, J is the Jacobean matrix constructed by the partial derivatives of the DLT equations with respect to each unknown, and k is the misclosure vector that represents the difference between the function when substituted by initial and estimated parameter values.

G. Accuracy Assessment

The model development accuracy is determined based on (17), where n is the number of FCP pairs, r_x and r_y are the residuals of x and y coordinates, respectively. They result from the difference between adjusted and observed coordinates on aerial datasets, respectively. Observed coordinates are those of the FCPs, whereas the adjusted ones are the FCPs' calculated coordinates knowing their correspondents on the LiDAR dataset, along with the transformation parameters. On the other hand, a registration model is validated using a set of manually collected and well-distributed checkpoints identified on both the aerial and LiDAR datasets. The validation RMSE is calculated using the same equation, where n, in this case, is the number of checkpoints' pairs, and r_x and r_y are the residuals in x and y coordinates of the checkpoints, respectively

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(r_{xi}^2 + r_{yi}^2\right)}. \tag{17}$$

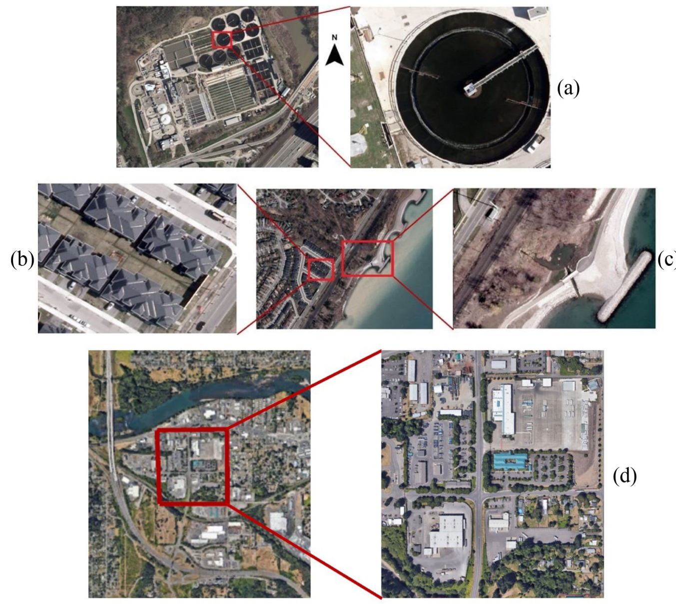

Fig. 3. Study areas (Google Earth, 2020).

III. Experimental Work

A. Study Areas and Datasets

Four urban morphologies were tested to check the validity of the proposed method: (a) industrial, (b) residential, (c) coastal shore, and (d) industrial-residential, as illustrated in Fig. 3. The first three regions are located in the Greater Toronto Area, Ontario, Canada, whereas the fourth one is located in Springfield city in Lane County, OR, USA.

Area (a) includes parts of the Humber wastewater treatment plant, which is located in the south-west of Toronto. The selected zone (43°38′5.15″N and 79°28′44.70″W) covers an approximate area of 3085 m^2. It is bound by Lake Ontario from the south and the Humber River from the east. Areas (b) and (c) are residential houses and coastal shore, respectively, which are located in Scarborough. Area (b) is within Adam's Creek neighborhood

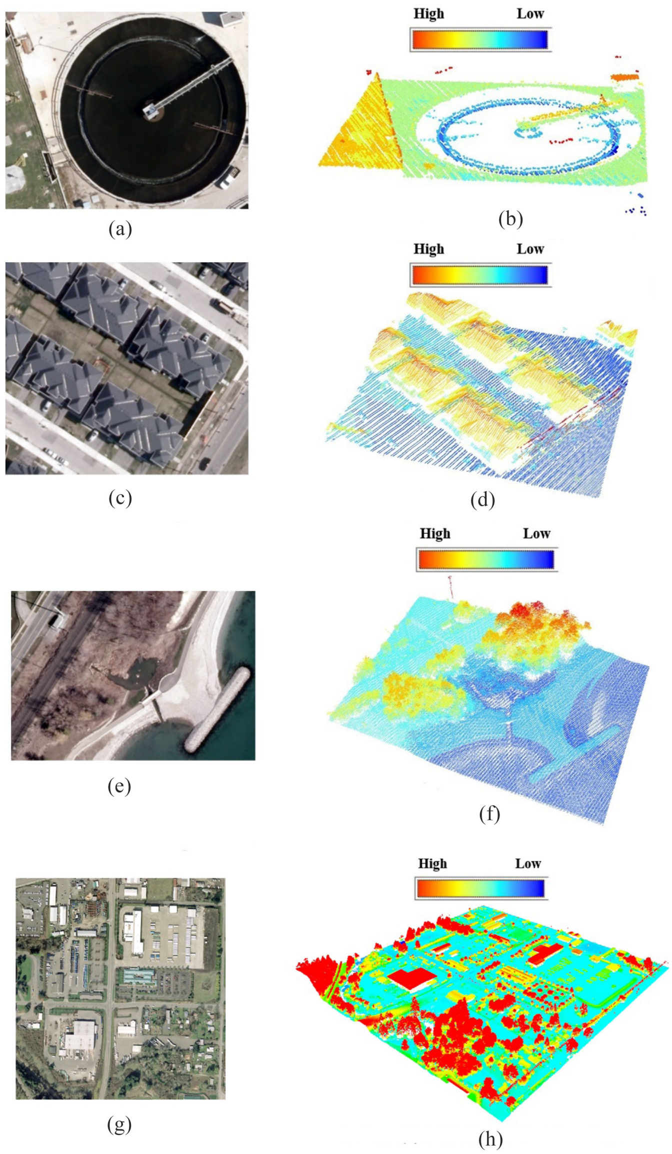

Fig. 4. Datasets. (a) Zone (a): Aerial image—RGB. (b) Zone (a): LiDAR data—height values. (c) Zone (b): Aerial image—RGB. (d) Zone (b): LiDAR data—height values. (e) Zone (c): Aerial image—RGB. (f) Zone (c): LiDAR data—height values. (g) Zone (d): Aerial image—RGB. (h) Zone (d): LiDAR data—height values.

($43°47'2.75''$N and $79°07'41.62''$W), and it covers an approximate area of 6501 m^2. Area (c) covers parts of Adam's Creek provincial park, which is surrounded by Lake Ontario from the east. This coastal-shore zone is located at $43°47'3.31''$N and $79°07'32.36''$W, and covers an approximate area of 23 299 m^2. Area (d) ($44°02'33.37''$N and $123°02'30.17''$W) is mainly an industrial area that contains lots of stores and corporates, beside some residential settlements. It has an approximate area of 352 420 m^2.

The aerial images and corresponding LiDAR point clouds of the four study zones are displayed in Fig. 4. Orthophotographies were downloaded from the Geospatial Map and Data Centre [43] for zones (a)–(c). Zone (d)'s LiDAR and aerial datasets were downloaded from the State of Oregon Department of Geology and Mineral Industries [44]. The SR of the tiles covering zone (a) is 8 cm, 20 cm for zones (b) and (c), and 15 cm[1] for zone (d). Zone (a)'s image consists of RGB bands, whereas images of zones (b)–(d) have RGB and NIR spectral bands. Characteristics of LiDAR data are shown in Table I. All datasets are projected; however, the georeference file of the aerial photos was deleted to ruin the pixels' geolocation, to test the model registration capabilities.

TABLE I
SUMMARY OF THE EXPERIMENTAL LIDAR DATASETS

Zone	Sensor	Point count	Point spacing (m)	Max. returns	Laser wavelength	Software	Year
(a)	Galaxy	14,262	0.453	5	1064 nm	TerraScan	2016
(b)	Titan	273,238	0.154	4	C_1: 1550 nm, C_2: 1064 nm, C_3:532 nm	OptechLMS	2015
(c)	Titan	232,000	0.101	4	C_1: 1550 nm, C_2: 1064 nm, C_3:532 nm	OptechLMS	2015
(d)	Galaxy	2,420,118	1.203	4	1064 nm	TerraScan	2012

All calculations in this study were developed using Python programming language on the integrated development environment (IDE) Enthought Canopy—version 2.1.9. Numpy and GDAL libraries were used for matrices and raster data manipulation, respectively. In addition, some functions embedded in ArcMap 10.5 and ArcScene 10.5 were implemented in the analysis. LiDAR 3-D points were provided in LAS files and were converted to text files via the "las2txt" tool in LAStools software, to be analyzed by Python. Information about LAS files was generated by the "lasinfo" tool in LAStools.

B. Image Generation From LiDAR Data

LiDAR LAS files were converted to 2-D images. The generated images were set to have the same pixel size as their corresponding aerial dataset. The choice between points' intensity and height to derive images was done visually based on the scene nature. Whichever yields the most discriminative image against its urban elements should be used.

Since the PC model runs on a single image layer, RGB bands in the aerial image of zone (a) were combined according to (18) to provide an individual greyscale layer [45], whereas NIR band layers in aerial data of zones (b)–(d) were directly input to the model, due to the nature of both scenes that makes NIR of better discrimination

$$\text{Grey-scale} = 0.299\text{R} + 0.587\tilde{\ }\text{G} + 0.114\text{B}. \qquad (18)$$

C. PC Model for Edge Points Detection

In designing the bank of filters as moving windows, N was set to 5, and t ranged from $-n$ to $+n$ with $\Delta t = 1$. These values are as what was applied in [37]–[40], where they were found optimal to give significant PC measures with the least number of filters and computation time. PC in this study was

[1]0.5 foot—units for this zone are converted from imperial to metric system for consistency purposes.

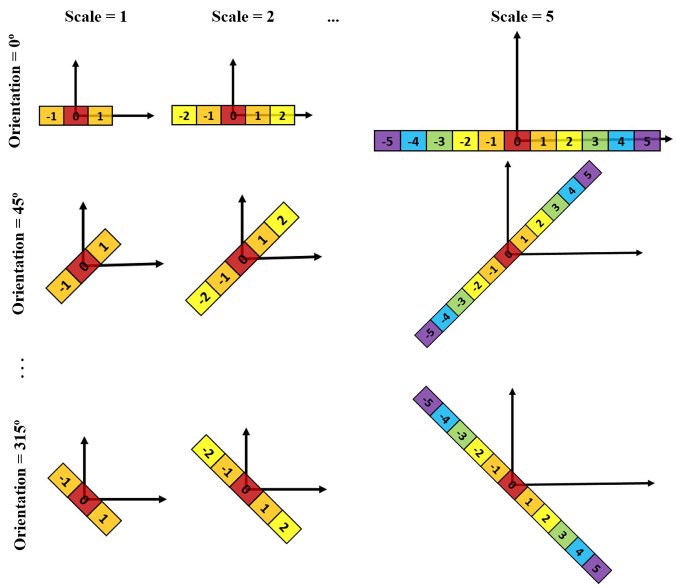

Fig. 5. Filters with multiple scales and orientations.

calculated over eight orientations, starting from $0°$ to $360°$ with an increment of $45°$, to acquire more robust local frequency information. Designed filters with t values over different scales and orientations are given in Fig. 5. Odd and even filter values in each scale were calculated from (3), by substituting in the corresponding part with the aforementioned t values. Aerial and LiDAR images when visualized based on moment values should show a pattern that reflects a separation among the scene's elements, as a logical consequence of running an edge/corner detection approach. However, there are two worthy points in regards to running the PC model on the datasets of this study, which are as follows.

(1) K in (6) was adjusted to a value lower than 2.5. The latter was too large as it increased T to an extent that made the subtraction result between $\lfloor\rfloor$ in (1) a negative value, and hence was replaced by zero, ending up with a zero PC measure. The problem was highlighted when $\mathrm{PC}(\theta)_{\max}$, the maximum calculated PC measure over an image at an orientation θ, was also zero. This means that there was no congruency at all, thus neither edges nor corners could be detected.

(2) The high SR of the images was an obstacle for the model to perform in some cases. It forced the model to search for a change in frequency in a too narrow range, where a point was considered significant with respect to its surroundings in a slim range while it is not, in a general sense. Consequently, the visualization of moments gave no pattern; therefore, the input images to the model were downsampled to larger cell size. This eliminated the detection of unnecessary edge and corner points and reduced the computational time as well. "Nearest" resampling technique was applied, which uses the center of the cell on the output raster to locate it on the input raster, to assign it its value. It is a simple approach that also preserves cells' values in the resampled output [46].

D. Selection of CCPs

M results were found to have a higher potential to yield CCPs than m, as the data are richer in edges than corner points. For better threshold setting to select CCPs, M points were converted to raster, then the resulted raster was subjected to a high-pass filter for better edge detection. Multiple threshold ranges were tried until visually reaching interrelated CCPs sets on both high-pass filter raster datasets from aerial and LiDAR data. To segregate CCPs, binary raster was generated by assigning a value of 1 to pixels within the threshold range, whereas the remaining pixels were assigned 0. These binary raster data were converted to points, then a selection query was performed to isolate the points with an attribute value of 1, to be matched in pairs.

This workflow failed to adjust threshold ranges that can identify relevant sets of CCPs in the data of zones (b)–(d). To overcome this drawback, the scenes were abstracted to their primary features that are common in both data types. M points on the perimeter of these features were targeted as CCPs. First, M images of zones (b) and (c) were clustered using the Iso cluster unsupervised classification. It is an iterative algorithm that requires the number of clusters to be identified at the beginning. Arbitrary means are assigned to each cluster in the first iteration, and each pixel is assigned to the cluster of the closest mean. New means are calculated and the process keeps running until either a maximum number of iterations or a convergence value (migration of cells from one cluster to another is minimal) is achieved. After reaching representative clusters, raster data were converted to polygons. Small polygons were merged with neighboring larger polygons. Afterward, polygons were converted to polylines that represent the polygons' boundaries. Finally, M points within a buffer around these polylines were targeted as CCPs. The number of clusters was set to two, and the buffer distances were 1 and 2 m for zones (b) and (c), respectively. These values were decided based on the visual inspection of both scenes in regards to the minimum common abstractions and the resampling size.

Zone (d) was processed differently. To obtain mutual sets of CCPs on both images, a binary map indicating edge and nonedge cells was needed. It could have been generated by the direct application of the Canny edge detector on the images. However, the operator was applied to the M images instead, to eliminate the noise effect usually reported when using the Canny algorithm. Common edge features on Canny maps of aerial and LiDAR images were manually isolated, and the points on these pixels were targeted as CCPs.

E. SCD for Matching CCPs as FCPs

Fig. 6 explains the main elements of the model. Assuming several m-CCPs on the aerial data, azimuth angles (Az) and normalized distances (d) between the members of m-CCPs were calculated and stored in d/Az-matrix, which has a length of m. Each element of the d/Az-matrix is a matrix with a length of m-1, since neither Az nor d are calculated between a point and

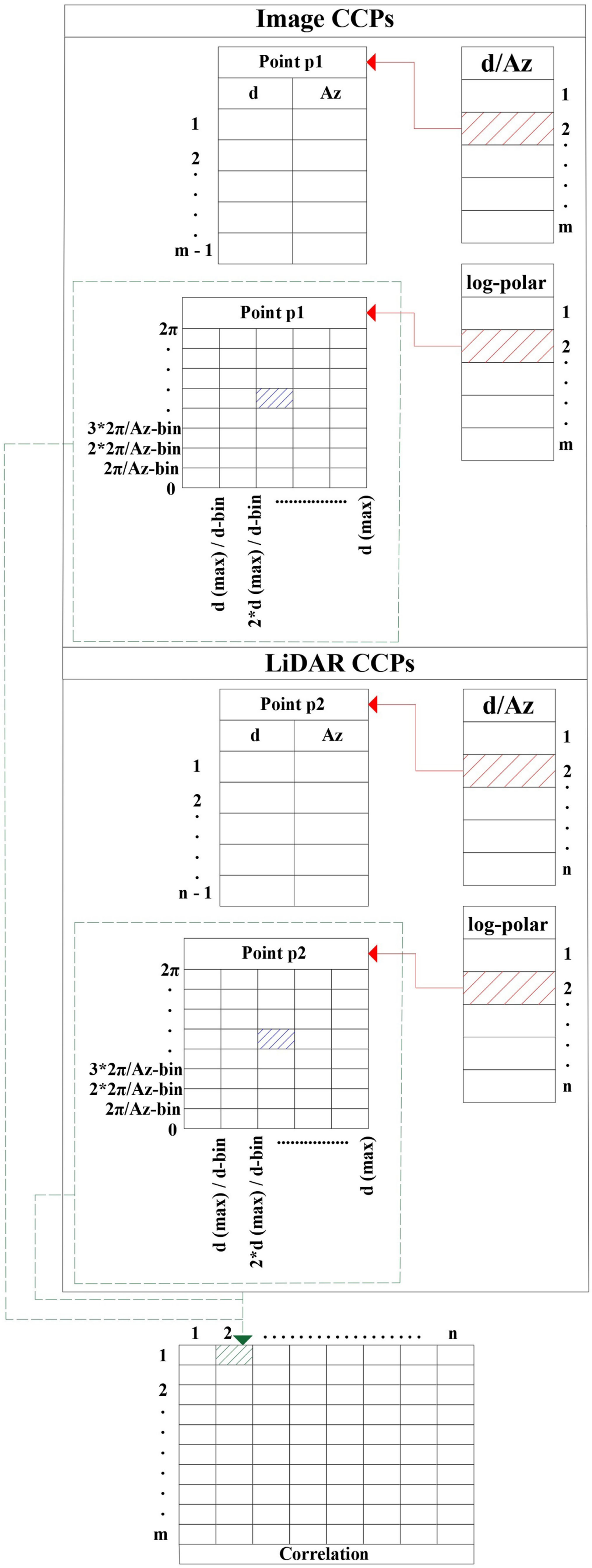

Fig. 6. SCD model.

itself. On the other hand, the log-polar matrix has a length of m, where each element represents the log-polar diagram of each of m-CCPs. A single diagram has a horizontal bin on the distance axis (d-bin) of five, and a vertical bin on the angle axis (Az-bin) of eight.

To illustrate, the cell hatched in blue to the left includes the number of points in m-CCPs that have a d and an Az from point p_1 in the ranges ($2d_{\max}/d$-bin, $3d_{\max}/d$-bin) and ($4*2\pi/Az$-bin, $5*2\pi/Az$-bin), respectively, where $d_{\max}$ is the maximum normalized distance calculated from p_1 in m-CCPs. The same logic applies to n-CCPs on LiDAR data. The correlation between the log-polar diagram of each m-CCPs and every log-polar diagram of n-CCPs was calculated and stored in the ($m \times n$) correlation matrix. For example, the green hatched cell saves the correlation value between the log-polar of the first point in m-CCPs and the second point in n-CCPs. The correlation relation is one-to-one, meaning that an individual point from m-CCPs must be matched with only a single point from n-CCPs. Therefore, a point in m-CCPs was paired up with the point of the maximum correlation value in n-CCPs. Correlated pairs with a value equal to or higher than a predefined correlation threshold were targeted as FCPs. A minimum correlation of 0.95 was applied for the three study areas.

IV. Results and Discussion

A. Image Generation From LiDAR Data

Fig. 7 shows the generated aerial and LiDAR raster data. Zone (a)'s LiDAR image was derived based on its height values. The area represents a waste-water treatment tank with outer and inner rings, in addition to a moving arm, a building at the upper right corner, and a yard to the left. All these features share covers that are similar in intensity, but different in elevation values relative to each other. In contrast, zone (b) includes low-rising houses with a slim range of height difference with respect to their surroundings; asphalt roads, land markings, and grassy landscape. On the other hand, these elements widely vary in their intensity values. This is why the image was produced based on C_2 intensity records. Likewise, the image from zone (c)'s LiDAR data was obtained out of C_2 intensity records, because the scene contains features of different materials; waterbodies, sand shore, dense forest, concrete sidewalks, asphalt roads, and steel railway rails. The same applies to zone (d)'s LiDAR image as well.

The scene elements of zone (a) appear on both images, and thus, a straightforward automatic registration was anticipated. On the contrary, the roof details appear clearly in the aerial data of zone (b), whereas they are not captured in the corresponding LiDAR image. The same applies to zone (c), where the forestry part in the middle is obviously planted in the aerial data, whereas those tree canopies do not appear clearly in the corresponding LiDAR image. Also, the tree coverage appears dissimilar in zone (d)'s both images. These differences suggested that the use of direct thresholding would capture unmutual edge points. Hence, the isolation of CCPs would need interactivity via a semiautomatic registration process.

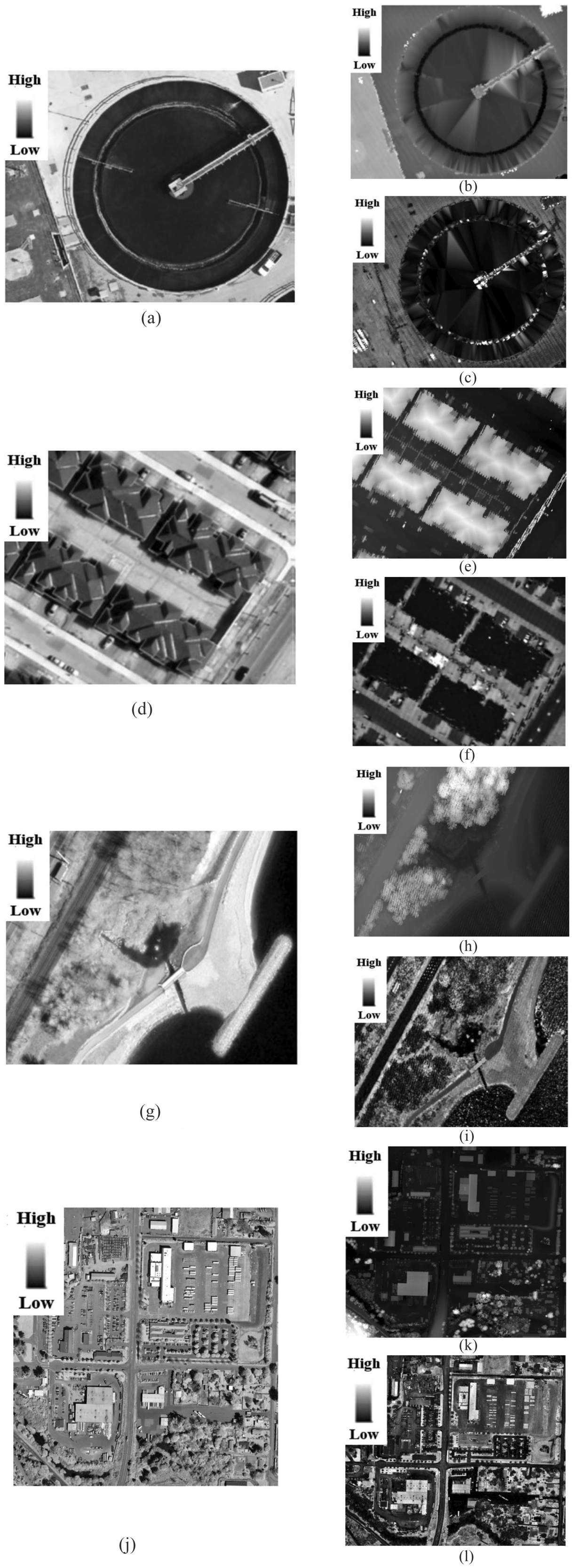

Fig. 7. Generated images—SRs: zone (a): 8 cm and zones (b) and (c): 20 cm. (a) Zone (a): Aerial grey-scale image. (b) Zone (a): LiDAR elevation image. (c) Zone (a): LiDAR intensity image. (d) Zone (b): Aerial NIR image. (e) Zone (b): LiDAR elevation image. (f) Zone (b): LiDAR intensity image (C_2). (g) Zone (c): Aerial NIR image. (h) Zone (c): LiDAR elevation image. (i) Zone (c): LiDAR intensity image (C_2). (j) Zone (d): Aerial NIR image. (k) Zone (d): LiDAR elevation image. (l) Zone (d): LiDAR intensity image.

TABLE II
OPTIMAL PARAMETER VALUES FOR REPRESENTATIVE *M* AND *m*—IMAGE (1): AERIAL PHOTO, IMAGE (2): LIDAR IMAGE

Parameter	Zone (a)		Zone (b)		Zone (c)		Zone (d)	
	Image (1)	Image (2)	Image (1)	Image (2)	Image (1)	Image (2)	Image (1)	Image (2)
K	0.30	0.10	0.16	0.35	0.17	0.65	0.00	0.01
SR (cm)	32	40	70	70	150	160	15	15
a^2	20							
k	1							
Δf	1							
m	1.5							
c	0.4							
g	10							
ϵ	0.01							

B. Extraction of Edge Points From PC Model

Table II gives the resampled pixel size and *K* value at which the visualization of the data based on *M* or *m* gave a representative pattern. It also summarizes the rest of the constant values that were substituted in the PC model equations. The downscaled image size was determined by trial and error, after viewing the pattern resulted from different pixel sizes, *K* was also determined empirically by trial and error. It was found to result in a $\mathrm{PC}(\theta)_{\max}$ close to 1. As a best practice, *K* should be adjusted first and kept constant in all orientations for the pixel size under investigation, where the $\mathrm{PC}(\theta)_{\max}$ values from all orientations are close to each other and close to 1. Then, different pixel sizes are tried. *K* may need to be readjusted when altering the pixel size, if it yields $\mathrm{PC}(\theta)_{\max}$ not close to 1.

Obviously, the *M* points resulting from aerial and LiDAR images did not show identical patterns (see Fig. 8), and this is justified by many reasons. First, both datasets were obtained at different times, which causes some features to appear on one image without showing up on the other. For example, the vehicle at the lower right corner of the aerial photo covering zone (a) did not exist at the acquisition time of the corresponding LiDAR data [see Fig. 7(a) and (b)]. Likewise, the forestry extent between the railway and the lakeshore fully appears on the aerial image of zone (c), but was subjected to seasonal changes on the corresponding LiDAR image, where it appears partially as a consequence [see Fig. 7(e) and (f)]. Besides, aerial and LiDAR data are originally different in nature; sensor type, acquisition mechanism, and data dimension. For instance, since the LiDAR data of zone (a) has significant water dropouts on the water surface [see Fig. 4(b)], the interpolation worked poorly and misrepresentation of this area occurred in the LiDAR image [see Fig. 7(b)]. Furthermore, shadow plays a glaring role magnifying this difference in patterns. As a common shortcoming when processing aerial photos, shadows are misanalyzed as a scene feature, and hence, its edges were displayed in the *M* pattern, which is not the case in LiDAR data. As examples, the building in the upper right corner on the aerial photo of zone (a) [see Fig. 7(a)] as well as the building roofs in zone (b) [see Fig. 7(c)] have shadows that were detected as edges. Nevertheless, the boundaries of common segments in both images were still enough to identify CCPs.

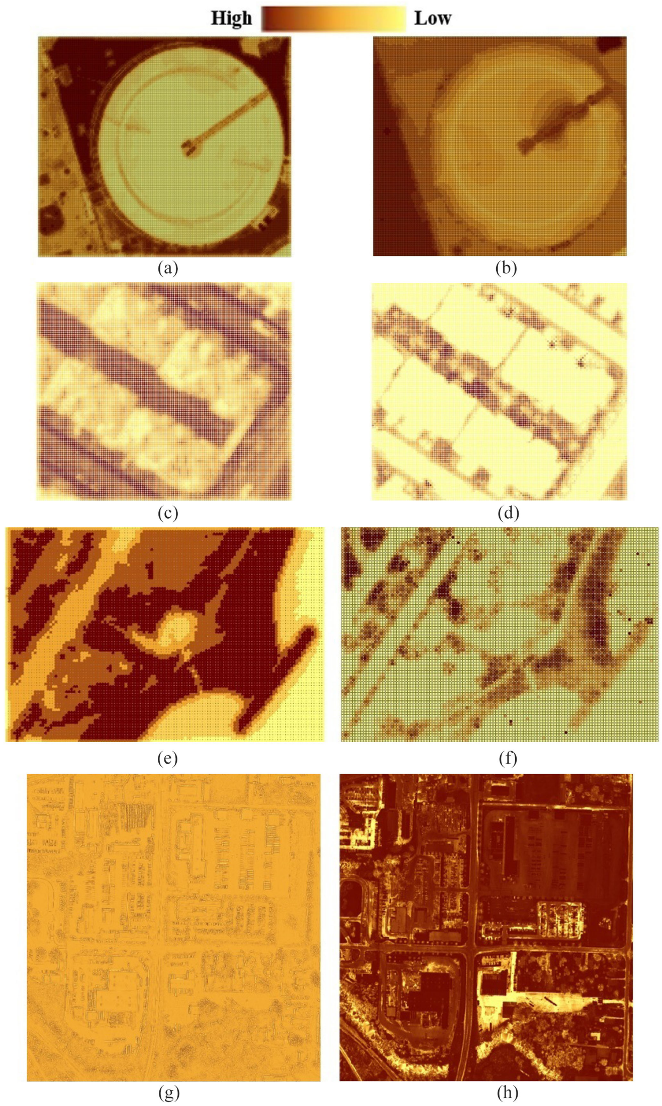

Fig. 8. *M* points derived from resampled aerial and LiDAR images. (a) Zone (a): Aerial photo. (b) Zone (a): LiDAR image. (c) Zone (b): Aerial photo. (d) Zone (b): LiDAR image. (e) Zone (c): Aerial photo. (f) Zone (c): LiDAR image. (g) Zone (d): Aerial photo. (h) Zone (d): LiDAR image.

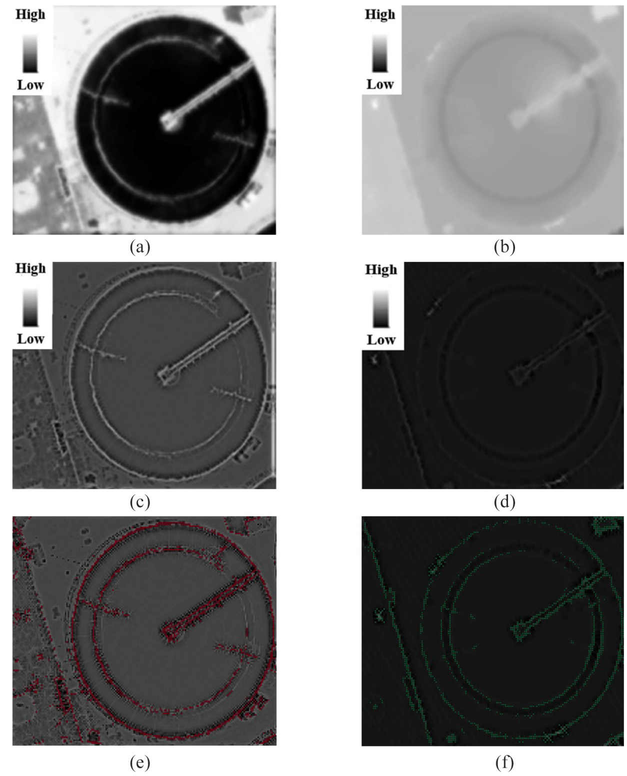

Fig. 9. CCPs by threshold setting—Zone (a). (a) M raster: Aerial photo (SR = 32 cm). (b) M raster: LiDAR image (SR = 40 cm). (c) High-pass filter: Aerial photo. (d) High-pass filter: LiDAR image. (e) CCPs: Aerial photo.

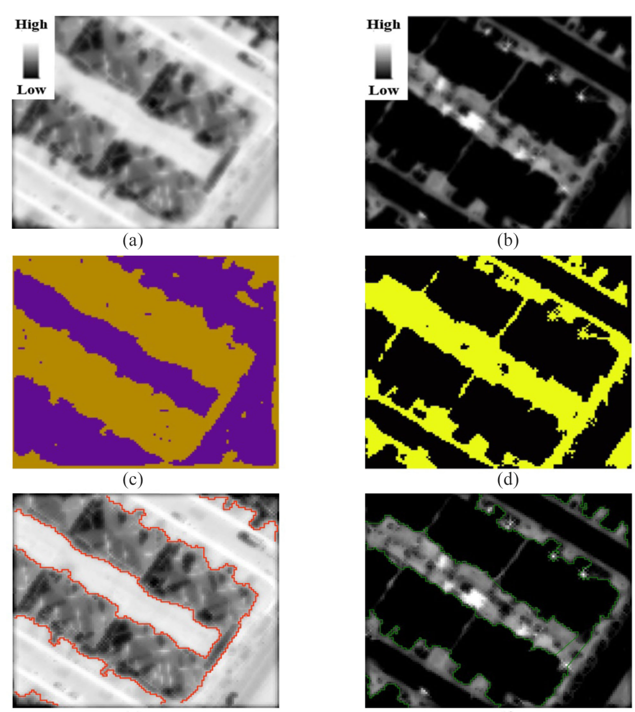

Fig. 10. CCPs by scene abstraction (clustering)—Zone (b). (a) M raster: Aerial photo (SR = 70 cm). (b) M raster: LiDAR image (SR = 70 cm). (c) Clusters: Aerial photo. (d) Clusters: LiDAR image. (e) Outline extraction: Aerial photo. (f) Outline extraction: LiDAR image.

C. Identification of CCPs

M points were converted to raster, which was input to a high-pass filter for the ease of adjusting a threshold range (see Fig. 9). Thresholding resulted in 1679 and 1580 CCPs in aerial and LiDAR data of zone (a), respectively. Thresholding did not work properly with zones (b)–(d), as the tangible differences in *M* patterns between aerial and LiDAR data, discussed in Section IV-B, led to a failure in adjusting thresholds capable of separating similar sets of CCPs in both images. Hence, raster datasets from *M* points of zone (b) were partitioned into two clusters, outlines were extracted and refined, *M* points within a buffer of 1 m were targeted as CCPs (see Fig. 10). 1674 and 1947 CCPs were detected out of the *M* points of aerial and LiDAR data, respectively.

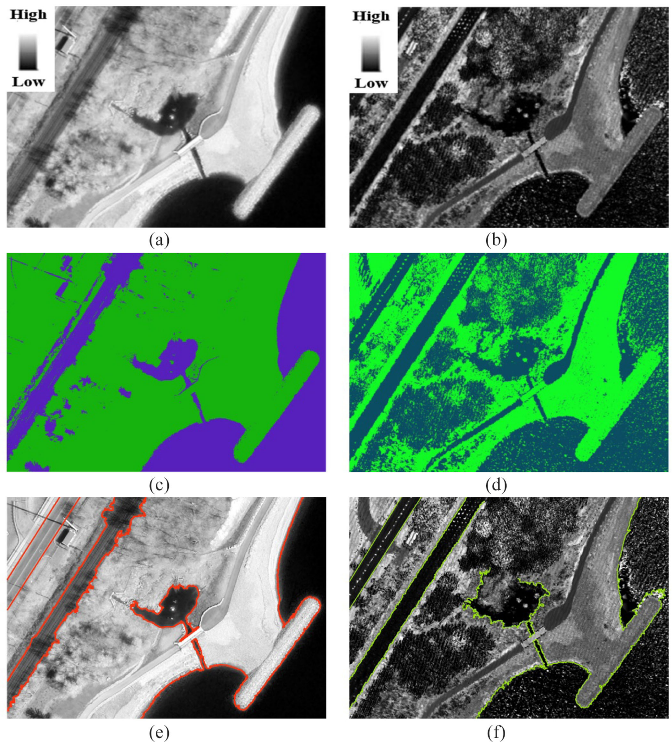

Fig. 11. CCPs by scene abstraction (clustering)—Zone (c). (a) Original aerial photo: NIR, SR = 20 cm. (b) Original LiDAR image: C_2, SR = 20 cm. (c) Clusters: Aerial photo. (d) Clusters: LiDAR image. (e) Outline extraction: Aerial photo. (f) Outline extraction: LiDAR image.

The same workflow was applied in processing zone (c); however, the clustering algorithm was run directly on the original images, as it gave a better abstraction than running it on the M raster, due to the heterogeneity of its both M patterns. CCPs were those M points that were located in a buffer distance of 2 m around the refined cluster boundaries. In this case, the PC model contributed to the identification of the optimal image downscale for efficient processing. Moreover, it highlighted a major drawback of the PC model; noise, as reported by [38], which is highlighted in this study due to feature variation in both datasets. Finally, a total of 1715 and 1564 CCPs were extracted from aerial and LiDAR datasets, respectively (see Fig. 11) .

An alternative to clustering, zone (d)'s scenes were abstracted by the inclusion of a Canny edge detector when applied to the M images to extract CCPs. The Gaussian blur (sigma parameter) was set to 5, and the double threshold values were adjusted to 0.1 and 0.3. Fig. 12(a) and (b) shows the output edge pixels on aerial and LiDAR images, with the M in the background. A zoom-in illustrates the reduction of noise (false detected edges) in comparison to running the operator directly on the raw images; NIR and C_2. Due to the massive partially related sets of edge pixels detected in both images, their sizes had to be reduced prior to matching in pairs. Hence, common edge features in both images were isolated manually, and resulted in initial CCPs of 22 876 and 23 825 points on aerial and LiDAR images, respectively, which is proportional to the zone's large area being processed at a fine SR. These sizes are still too large as inputs to

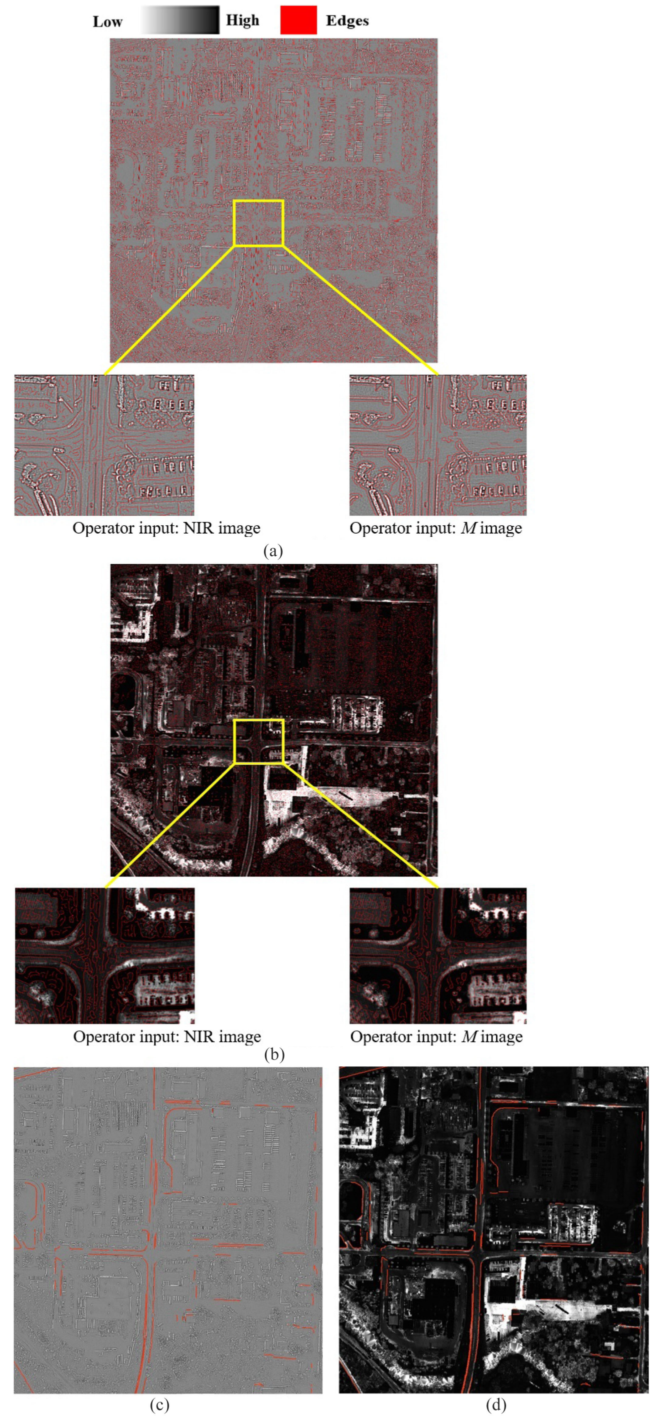

Fig. 12. CCPs by scene abstraction (Canny operator inclusion)—Zone (d). (a) Canny on aerial photo. (b) Canny on LiDAR image. (c) CCPs: Aerial photo. (d) CCPs: LiDAR image.

the SCD model, given that each point in a CCPs set is examined against the entire points on the other CCPs set. The workstation [Windows 10 Pro for Workstations OS 64-b, 3.2-GHz processor (16 CPUs), 131 072-MB RAM] on which these analyses were performed processes two sets of CCPs, each of 3000 points, in almost 14 h. Hence, only endpoints of edge features were

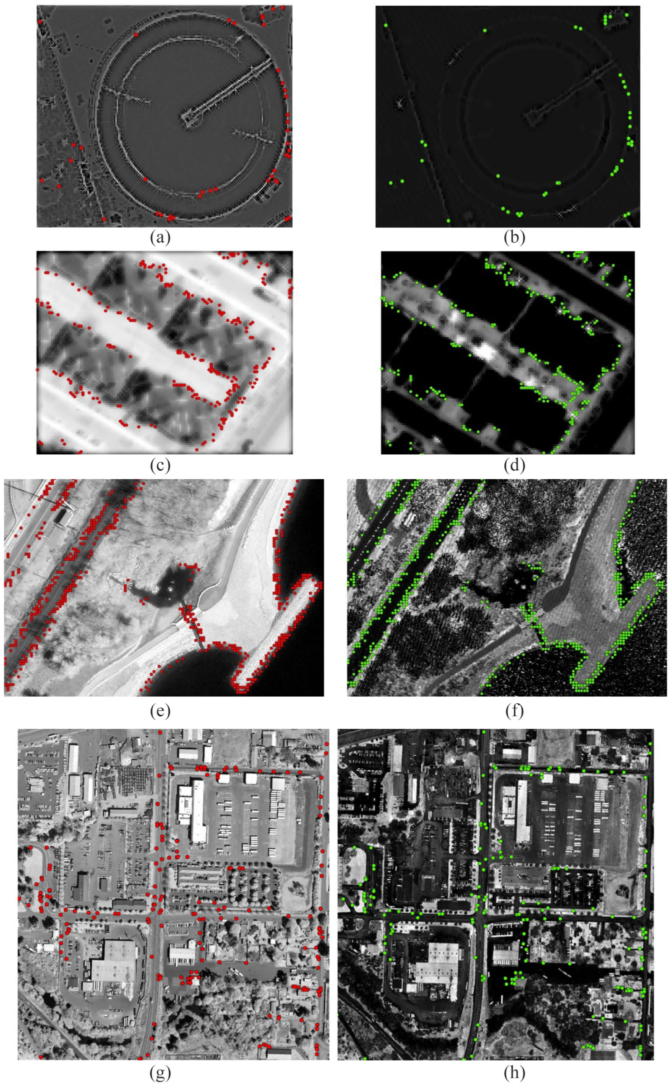

Fig. 13. FCPs determined in the four zones. (a) Zone (a): Aerial photo. (b) Zone (a): LiDAR image. (c) Zone (b): Aerial photo. (d) Zone (b): LiDAR image. (e) Zone (c): Aerial photo. (f) Zone (c): LiDAR image. (g) Zone (d): Aerial photo. (h) Zone (d): LiDAR image.

considered as CCPs. They are 188 and 186 points on aerial and LiDAR images, respectively [see Fig. 12(c) and (d)].

D. Determination of FCPs by Matching CCPs

The SCD successfully matched 43, 252, and 182 pairs of points with a correlation of ≥ 0.95 in zones (a), (b), and (d), respectively, whereas 515 points were paired up in zone (c) with a correlation of ≥ 0.97 (see Fig. 13). Since the model in general accommodates scale differences between input images, they should share the same extent. Accurate registration requires similar sets of CCPs to be obtained in both aerial and LiDAR images. These similar sets are optimally achieved when having the same edges detected in both images. This is hard to come by when processing remote sensing data that are convoluted in nature, especially when they are acquired on different dates/platforms. However, the model could successfully result in consequential FCPs in the three study zones. Finally, high correlation values are not the sole factor to consider while pairing up the FCPs, their even distribution over the study area is also critical in order to avoid distortion at locations that lack control points. Therefore, high correlation thresholds may be tolerated, if more points were to be included for better FCPs dissemination.

E. Registration Results and Model Validation

Fig. 14 illustrates the model development accuracy and validation accuracy of the four coordinate transformation models when applied in the registration of the four zones' data. The development accuracy represents the deviation of the CCPs' coordinates from their calculated ones after estimation of the parameters on the aerial data. However, it cannot be the sole indicator to describe the model precision, as higher order polynomials gave lower RMSE when implemented, but yielded distorted registration due to overfitting. Therefore, model validation was required in order to ensure registration accuracy. A total of 7, 26, 15, and 96 checkpoints were manually collected on the four study zones (a), (b), (c), and (d), respectively. They were chosen on the original images at distinguishable locations, as shown in Fig. 15.

The high similarity between the two images of zone (a) enabled the PC model to detect interrelated CCPs sets on circular edges, which are usually avoided in image registration studies due to their tedious matching. However, thresholding was incapable of fully detecting CCPs on both edges of the inner tank as well as the grass boundary on the aerial photo, in contrary to the CCPs' coverage of the same elements on the corresponding LiDAR image. As a consequence, the relative relationship among one CCPs set was different from the other. Therefore, this mismatch reduced the maximum correlation value on one hand, and maintaining an even distribution of the FCPs was a driving force to lower the correlation threshold in order to include more FCPs, on the other hand, which eventually increased the RMSE value. Zone (b) has likewise relatively high RMSE values ($<$ 2 pixels), despite that its both images share the same boundary polylines after the clustering was accomplished. Nevertheless, these outlines are irregular and dissimilar in both images, and yet the two CCPs sets buffered around them are not as congruent as they should have been. This caused FCPs mismatch, lowered the correlation threshold value, and increased the RMSE value in consequence. Zone (c) yielded the best registration result ($<$ 1 pixel) since the high level of congruency between the edges' boundaries in the zone's both images enhanced the quality of the CCPs, which accordingly raised the correlation between the FCPs. A third-order polynomial model was applied in the registration of these three datasets, as it resulted in the most consistent minimum development and validation accuracies.

On the other hand, the high values of development RMSE ($<$ 5 pixels) along with its significant drop when compared to the validation RMSE values ($<$ 1 pixel), for zone (d)'s results, indicate the insufficiency of the CCPs sets in terms of quantity and quality. Obviously, using the features' endpoints instead of the entire points located on them misses out on a noticeable amount of high potential CCPs on both images. This points

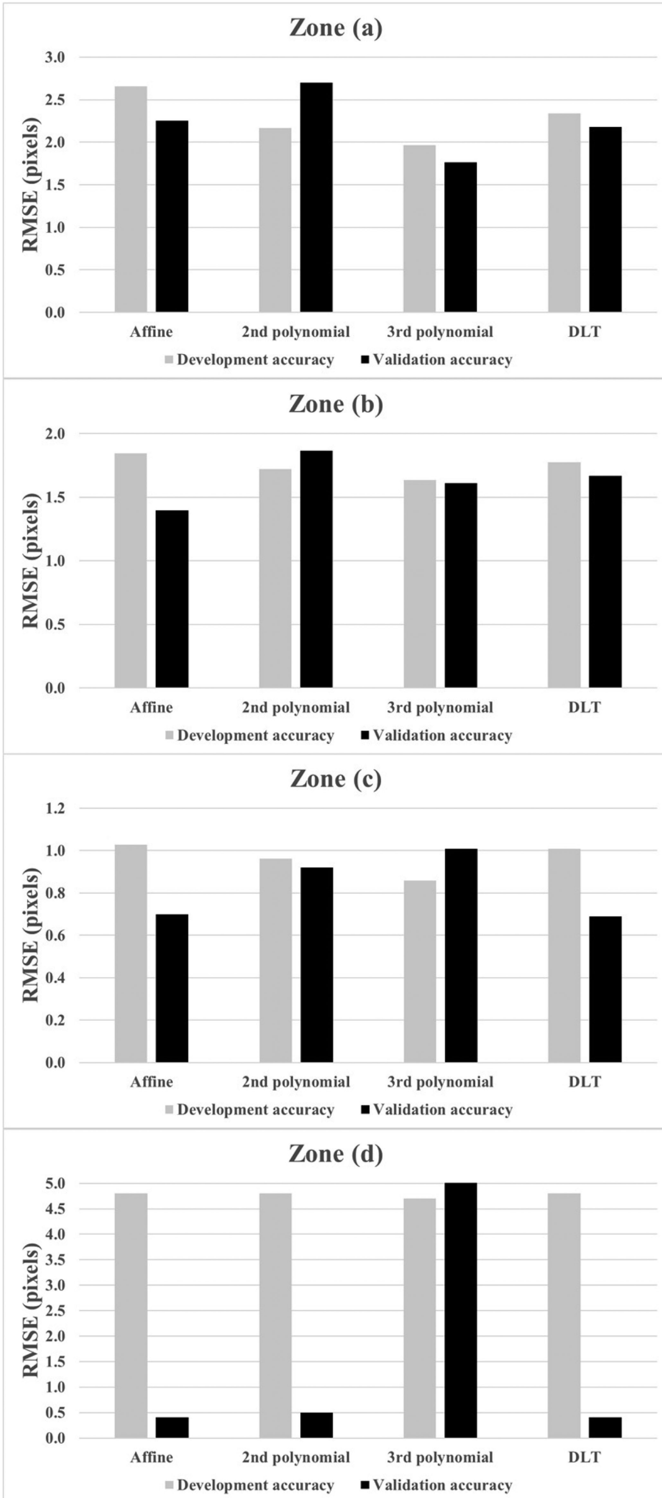

Fig. 14. Registration accuracy.

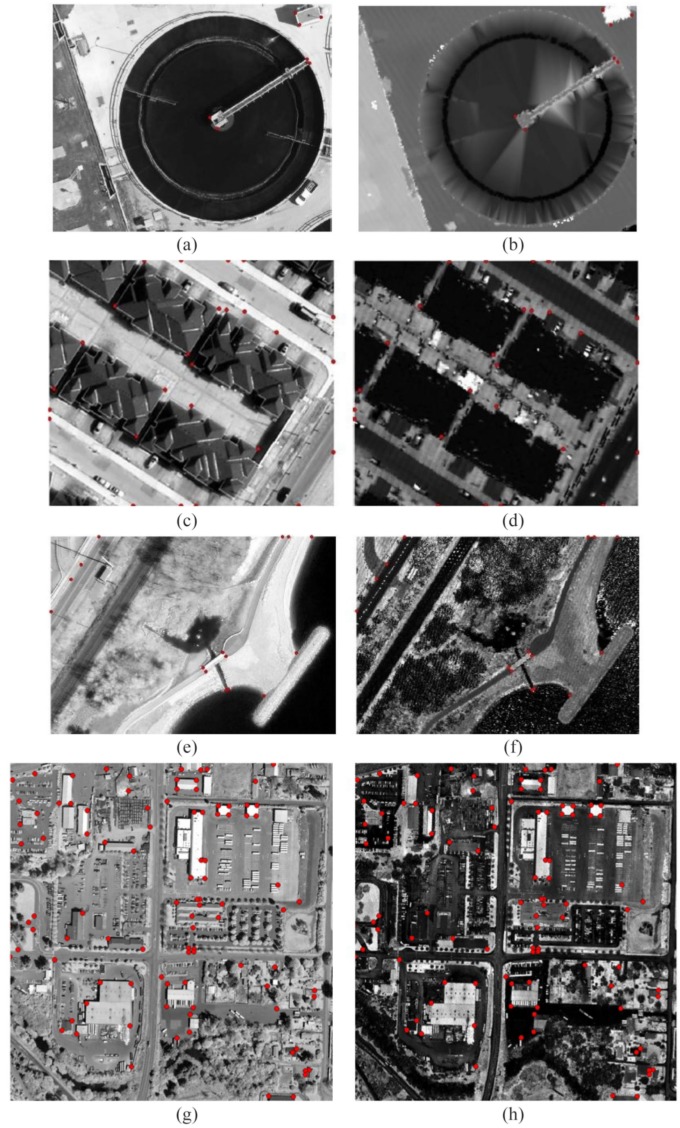

Fig. 15. Distribution of checkpoints in the four zones. (a) Zone (a): Aerial photo. (b) Zone (a): LiDAR image. (c) Zone (b): Aerial photo. (d) Zone (b): LiDAR image. (e) Zone (c): Aerial photo. (f) Zone (c): LiDAR image. (g) Zone (d): Aerial photo. (h) Zone (d): LiDAR image.

out to a drawback when applying the proposed algorithm on large study areas with fine SR, which highlights the necessity of investigating a proper downsampling algorithm that reduces the size of CCPs while maintaining interrelated sets at the same time. Alternatively, large images could be processed in tiles and/or at lower SRs. The affine model was used to register zone (d)'s datasets. Since the registration models applied in this study are empirical, qualitative evaluation of the results by navigating through the points to ensure they properly inherit their corresponding radiometric characteristics is a vital part of assessing the results.

The PC model for edges and corners detection from images is obviously sensitive to feature variation in a general sense. This drawback is significantly addressed when processing remote sensing data that are complicated by nature, and their sophistication is even highlighted in urban land-uses captured by high-SR sensors. This deficiency hardens setting a proper threshold range for CCPs selection in aerial and LiDAR raster data. Although the moment points that the PC model results in can be further processed by scene abstraction to output common boundaries as edges, these edges are not always guaranteed to have an adequate

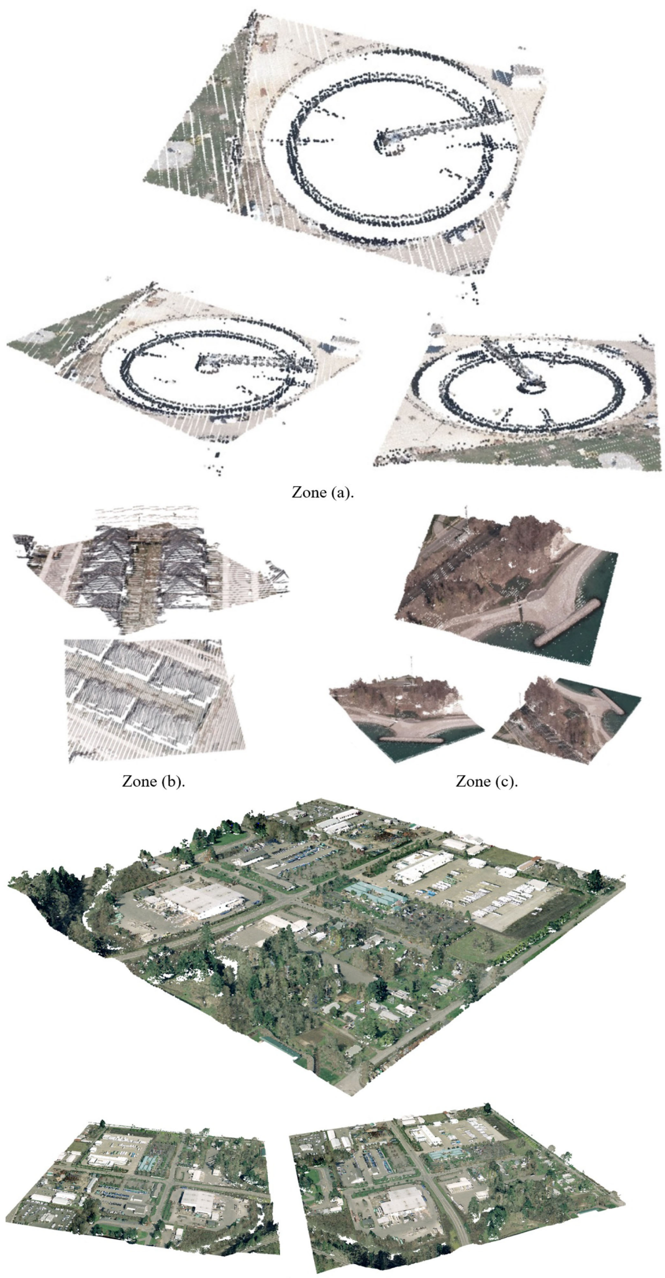

Fig. 16. Colored 3-D LiDAR points after registration.

level of similarity in both datasets for registration purposes. This is because of the core challenge that is encountered by default in this study; registering two various types of remote sensing data in terms of acquisition technique and date, beside the type of ground objects' properties recorded in each. Nevertheless, the qualitative evaluation of the results (see Fig. 16) is satisfactory considering the dissimilarities in sensor type, data acquisition technique, and time, as major issues usually confronted in the registration of two different remote sensing datasets, such as aerial and LiDAR data. The proposed method is a good choice in this case for many reasons. First, the PC model results in edge points (moment points *M*) at a SR lower than that of the original images, which eliminates feature variation existing by nature, highlighted especially in remote sensing data of high SR, and tangibly accelerates the processing time. In addition, the scene abstraction by identifying the edges' boundaries out of the *M* points accommodates for the feature variation sensitivity of the model, by targeting points on these shared polylines as CCPs. Moreover, the systematic, ease of application, and pace of processing when dealing with small study regions are all motives to further enhance the results to better suit larger datasets.

V. Conclusion

This study was conducted to register airborne LiDAR point clouds to aerial images. A semiautomatic 2D-3D point-based registration was applied using the PC model as CCPs identifier, in addition to the SCD model to pair up CCPs as FCPs. LiDAR points and their corresponding aerial photos were used to test the proposed methodology covering four different urban land-uses: industrial, residential, coastal shore, and industrial-residential. LiDAR data were converted to 2-D images, then the PC model was run on the resampled raster of aerial and LiDAR datasets. The PC measure of each raster point was used to calculate two moments, which served as indicators to assess the points' potentiality of being an edge or a corner. A threshold range was identified to select CCPs based on the point's moment values. Thresholding worked well with the industrial data in isolating CCPs relevant in both raster datasets. However, the dissimilarity between the aerial image and its corresponding LiDAR raster in the residential, coastal-shore areas, and industrial-residential obstructed adjusting a proper threshold range. Alternatively, a scene abstraction algorithm to select CCPs was proposed as a semiautomatic registration approach, where moment raster data were clustered to their main elements that are common in both data. Moment points within a buffer around them were targeted as CCPs. The SCD model was run to match CCPs as pairs of FCPs that are at least 95% correlated. The transformation registration parameters were solved by LS for three polynomial models of the first-, second-, and third-order, respectively, in addition to the DLT model. Checkpoints picked up manually were used for validation. The accuracy of the registration resulted in an RMSE of around two pixels for the industrial and residential areas, around one pixel for the coastal-shore zone, and around five pixels for the industrial-residential zone, which has a large area. This suggests a future work that should focus on the automatic elimination of features that do not exist in both images, in addition to a decent mechanism that downsamples sets of CCPs and keeps mutual points without being erased.

The proposed registration approach is advantageous for many reasons. First, the systematic computations and pace of processing (especially when processing small datasets at higher SRs) are boosting factors toward imagery-LiDAR data integration, since the PC model runs on LiDAR and aerial images resampled at lower SRs. Second, the semiautomatic approach overcomes the obstacle of identifying interrelated CCPs by setting a threshold range, due to the dissimilarities found between LiDAR and its corresponding aerial image, by handling it from a broader perspective. The scene abstraction algorithm clusters raster data based on their edge potentiality, which is a robust detection of feature outlines. Moreover, the simplification of these outlines, by abandoning minor boundaries in spots of feature variation and keeping only generic ones as long as they exist in both raster datasets in an even distribution, provides a wider range for locating CCPs that are no longer limited to points on conventional linear urban elements and intersections. More inclusively, CCPs in this way could be found on nonlinear shorelines, lake boundaries, sidewalks, or even on irregular tree crowns. Furthermore, the proposed approach does not require either LiDAR or aerial data to be georeferenced and applies empirical registration models without the need of knowing the physical model characteristics, which are unavailable in some cases where metadata files do not exist. This generalization encourages applying the proposed workflow on data covering larger study regions with different urban morphologies, as well as it increases the opportunities of registering LiDAR point clouds to any imagery data. On the other hand, automating both raster data to have the same extent in order for the SCD to perform, and the manual intrusion that turns the proposed registration into a semiautomatic approach, as well as enhancing the registration accuracy especially when processing large study regions are highlights to be investigated in the future along with obtaining control points by shape matching of lines and polygons.

References

[1] Y. Megahed, P. Cabral, J. Silva, and M. Caetano, "Land cover mapping analysis and urban growth modelling using remote sensing techniques in Greater Cairo Region—Egypt," *ISPRS Int. J. Geo-Inf.* 4, no. 3, pp. 1750–1769, 2015.

[2] UN, DESA, "World urbanization prospects: The 2014 revision," United Nations Dept. Econ. Social Affairs, Population Division: New York, NY, USA, 41, 2015.

[3] G. P. Hancke, B. De Carvalho e Silva, and G. P. Hancke, Jr, "The role of advanced sensing in smart cities," *Sensors*, vol. 13, no. 1, pp. 393–425, 2013.

[4] D. Li, J. Shan, Z. Shao, X. Zhou, and Y. Yao, "Geomatics for smart cities-concept, key techniques, and applications," *Geo-Spatial Inf. Sci.*, vol. 16, no. 1, pp. 13–24, 2013.

[5] Y. Zheng, Q. Weng, and Y. Zheng, "A hybrid approach for three-dimensional building reconstruction in Indianapolis from LiDAR data," *Remote Sens.*, vol. 9, no. 4, 2017, Art. no. 310.

[6] W. Y. Yan, A. Shaker, and N. El-Ashmawy, "Urban land cover classification using airborne LiDAR data: A review," *Remote Sens. Environ.*, vol. 158, pp. 295–310, 2015.

[7] A. Wong and D. A. Clausi, "ARRSI: Automatic registration of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.* vol. 45, no. 5, pp. 1483–1493, May 2007.

[8] A. Shaker, W. Y. Yan, and S. Easa, "Using stereo satellite imagery for topographic and transportation applications: An accuracy assessment," *GIScі. Remote Sens.*, vol. 47, no. 3, pp. 321–337, 2010.

[9] T. Long, W. Jiao, G. He, Z. Zhang, B. Cheng, and W. Wang, "A generic framework for image rectification using multiple types of feature," *ISPRS J. Photogrammetry Remote Sens.*, vol. 102, pp. 161–171, 2015.

[10] A. Shaker, S. M. Easa, and W. Y. Yan, "Improved parameter estimation of the line-based transformation model for remote sensing image registration," *J. Imag.*, vol. 3, no. 3, 2017, Art. no. 32.

[11] W. Y. Yan, S. M. Easa, and A. Shaker, "Polygon-based image registration: A new approach for geo-referencing historical maps," *Remote Sens. Lett.* 8, no. 7, pp. 703–712, 2017.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[13] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, 2012, Art. no. 10491.

[14] Q. Li, G. Wang, J. Liu, and S. Chen, "Robust scale-invariant feature matching for remote sensing image registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 287–291, Apr. 2009.

[15] H. Nasir, V. Stankovic, and S. Marshall, "Image registration for super resolution using scale invariant feature transform, belief propagation and random sampling consensus," in *Proc. 18th Eur. Signal Process. Conf.*, 2010, pp. 299–303.

[16] V. Gioi, R. Grompone, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *Image Process. Line*, vol. 2, pp. 35–55, 2012.

[17] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, 2011.

[18] C. Lyu and J. Jiang, "Remote sensing image registration with line segments and their intersections," *Remote Sens.*, vol. 9, no. 5, 2017, Art. no. 439.

[19] N. Xue *et al.*, "Learning regional attraction for line segment detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[20] M. Nikolic, E. Tuba, and M. Tuba, "Edge detection in medical ultrasound images using adjusted Canny edge detection algorithm," in *Proc. 24th Telecommun. Forum*, 2016, pp. 1–4.

[21] L. Xuan and Z. Hong, "An improved Canny edge detection algorithm," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Serv. Sci.*, 2017, pp. 275–278.

[22] K. Gaurav and U. Ghanekar, "Image steganography based on Canny edge detection, dilation operator and hybrid coding," *J. Inf. Secur. Appl.*, vol. 41, pp. 41–51, 2018.

[23] Z. Xu, X. Baojie, and W. Guoxin, "Canny edge detection based on Open CV," in *Proc. 13th IEEE Int. Conf. Electron. Meas. Instrum.*, 2017, pp. 53–56.

[24] R. Song, Z. Zhang, and H. Liu, "Edge connection based Canny edge detection algorithm," *Pattern Recognit. Image Anal.*, vol. 27, no. 4, pp. 740–747, 2017.

[25] A. Habib, M. Ghanma, M. Morgan, and R. Al-Ruzouq, "Photogrammetric and LiDAR data registration using linear features," *Photogrammetric Eng. Remote Sens.*, vol. 71, no. 6, pp. 699–707, 2005.

[26] T.-S. Kwak, Y.-I. Kim, K.-Y. Yu, and B.-K. Lee, "Registration of aerial imagery and aerial LiDAR data using centroids of plane roof surfaces as control information," *KSCE J. Civil Eng.*, vol. 10, no. 5, pp. 365–370, 2006.

[27] E. Mitishita, A. Habib, J. Centeno, A. Machado, J. Lay, and C. Wong, "Photogrammetric and lidar data integration using the centroid of a rectangular roof as a control point," *Photogrammetric Rec.*, vol. 23, no. 121, pp. 19–35, 2008.

[28] W. Zhang *et al.*, "Registration of optical imagery and LiDAR data using an inherent geometrical constraint," *Opt. Express*, vol. 23, no. 6, pp. 7694–7702, 2015.

[29] B. Yang and C. Chen, "Automatic registration of UAV-borne sequent images and LiDAR data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 101, pp. 262–274, 2015.

[30] S. Zheng, R. Huang, and Y. Zhou, "Registration of optical images with LiDAR data and its accuracy assessment," *Photogrammetric Eng. Remote Sens.*, vol. 79, no. 8, pp. 731–741, 2013.

[31] R. Huang, S. Zheng, and K. Hu, "Registration of aerial optical images with LiDAR data using the closest point principle and collinearity equations," *Sensors*, vol. 18, no. 6, 2018, Art. no. 1770.

[32] T. Cui, S. Ji, J. Shan, J. Gong, and K. Liu, "Line-based registration of panoramic images and LiDAR point clouds for mobile mapping," *Sensors*, vol. 17, no. 1, 2017, Art. no. 70.

[33] H. I. Reuter, A. Nelson, and A. Jarvis, "An evaluation of void-filling interpolation methods for SRTM data," *Int. J. Geographical Inf. Sci.*, vol. 21, no. 9, pp. 983–1008, 2007.

[34] N. T. H. El-Ashmawy, "Innovative approach for automatic land cover information extraction from LiDAR data," Ph.D. dissertation, Dept. Civil Eng., Faculty Eng. Articular Sci. (FEAS). Ryerson Univ., Toronto, ON, Canada, 2015.

[35] H. Kim *et al.*, "An efficient implementation of a local binning algorithm for digital elevation model generation of LiDAR/ALSM dataset," in *Proc. Amer. Geophys. Union, Fall Meeting*, 2006, Art. no. G53C-0921.

[36] M. C. Morrone and R. A. Owens, "Feature detection from local energy," *Pattern Recognit. Lett.*, vol. 6, no. 5, pp. 303–313, 1987.

[37] P. Kovesi, "Phase congruency detects corners and edges," in *Proc. Aust. Pattern Recognit. Soc. Conf.: Digit. Image Comput.: Techn. Appl.*, 2003, pp. 309–318.

[38] P. Kovesi, "Image features from phase congruency," *Videre: J. Comput. Vis. Res.*, vol. 1, no. 3, pp. 1–26, 1999.

[39] C. Valens, "A really friendly guide to wavelets." Accessed: Jul. 1999. [Online]. Available: http://mathforum.org/library/view/16425.html

[40] B. MacLennan, *Gabor Representations of Spatiotemporal Visual Images*. Knoxville, TN, USA: Dept. Comput. Sci., Univ. Tennessee, 1991.

[41] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[42] P. J. Huber, "Robust regression: Asymptotics, conjectures and Monte Carlo," *Ann. Statist.*, vol. 1, no. 5, pp. 799–821, 1973.

[43] Ontario Ministry of Natural Resources—Peterborough Region, Greater Toronto Area (GTA) Orthophotography Project 2013, May 2014. [Online]. Available: http://geo2.scholarsportal.info.ezproxy.lib.ryerson.ca

[44] State of Oregon Department of Geology and Mineral Industries, Oregon Lidar Consortium (OLC), Courtesy City of Springfield, Oregon, 2012. [Online]. Available: https://www.oregongeology.org/lidar/

[45] Z.-u. Rahman, D. J. Jobson, G. A. Woodell, and G. D. Hines, "Multisensor fusion and enhancement using the Retinex image enhancement algorithm," in *Proc. Vis. Inf. Process.* XI, 2002, vol. 4736, pp. 36–44.

[46] R. Bouchiha and K. Besbes, "Automatic remote-sensing image registration using SURF," *Int. J. Comput. Theory Eng.*, vol. 5, no. 1, pp. 88–92, 2013.

Yasmine Megahed received the M.Sc. degree in geospatial technologies with a major in remote sensing from the NOVA IMS Information Management School, Lisbon, Portugal, in 2015. She is currently working toward the Ph.D. degree with the Department of Civil Engineering, Ryerson University, Toronto, ON, Canada.

Her research focuses on remote sensing applications, especially digital urban mapping that integrates LiDAR and imagery data in the point cloud classification of urban morphologies.

Ahmed Shaker received the Ph.D. degree in satellite sensor modeling from the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong, in 2004.

He is currently a Professor with the Department of Civil Engineering and an Associate Dean of the Faculty of Engineering and Architecture Science, Ryerson University, Toronto, ON, Canada. He holds two patents and has more than 130 publications in international journals and conferences. His research interests include LiDAR data processing, satellite sensor modeling, image segmentation and classification, and 3-D modeling.

Dr. Shaker was the recipient of a number of national and international awards. He is currently serving as the President of the Canadian Remote Sensing Society (2020–2022).

Wai Yeung Yan (Senior Member, IEEE) received the Ph.D. degree in civil engineering from Ryerson University, Toronto, ON, Canada, in 2012.

He is currently an Assistant Professor with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University and an Adjunct Professor with the Department of Civil Engineering, Ryerson University. His research interests include point cloud processing, laser scanning, and remote sensing.