IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2022, accepted 17 July 2022, date of publication 21 July 2022, date of current version 27 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192869

RESEARCH ARTICLE

Multimodal Fusion of Deeply Inferred Point Clouds for 3D Scene Reconstruction Using Cross-Entropy ICP

WATCHARAPHONG YOOKWAN[1], KRISANA CHINNASARN[2], (Senior Member, IEEE), CHAKCHAI SO-IN[3], (Senior Member, IEEE), AND PARAMATE HORKAEW[1]

[1]School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand
[2]Faculty of Informatics, Burapha University, Chon Buri 20131, Thailand
[3]Applied Network Technology (ANT) Laboratory, College of Computing, Khon Kaen University, Khon Kaen 40002, Thailand

Corresponding authors: Paramate Horkaew (phorkaew@sut.ac.th) and Krisana Chinnasarn (krisana@buu.ac.th)

This work was supported in part by the National Research Council of Thailand (NRCT) through the International Research Network Program under Grant IRN61W0006.

ABSTRACT Depth estimation is a crucial step toward 3D scene understanding. Most traditional systems rely on direct sensing of this information by means of photogrammetry or on stereo imaging. As the scenes getting more complex, these modalities were impeded by, for instances, occlusion and imperfect lighting condition, etc. As a consequence, reconstructed surfaces are normally left with voids, due to missing data. Therefore, surface regularization is often required as post-processing. With the recent advances in deep learning, depth inference from a monocular image has attracted considerable interests. Many convolutional architectures have been proposed to infer depth information from a monocular image, with promising results. Thus far, visual cues learned and generalized by these networks may be ambiguous, resulting in inaccurate estimation. To address these issues, this paper presents an effective method for fusing point clouds extracted from depth values, directly measured by an infrared camera and estimated by a modified ResNet-50 from an RGB image, of the same scene. To ensure robustness and efficiency of finding the correspondence between and aligning these point clouds, an information theoretic alignment strategy, called CEICP, was proposed. The experimental results on a public dataset demonstrated that the proposed method outperformed its counterparts, while producing good quality surface renditions of the underlying scene.

INDEX TERMS Depth estimation, entropy, ICP, photogrammetry, RestNet-50, scene reconstruction.

I. INTRODUCTION

Reconstruction of a three-dimensional (3D) scene is an important element in various scientific and engineering applications, such as computer-aided geometric design (CAGD), graphics, computer vision, medical image analysis, computational modeling, augmented reality (AR), and digital multimedia, etc. [1] For instance, in computer aided diagnosis (CAD), 3D information on an anatomical shape and its peripherals, possibly with associated lesions, reconstructed from the tomographic scan of a patient [2], are of great clinical values. Presented with this information, a physician can make accurate diagnosis and prognosis of the disease, as well as can perform therapeutic intervention [3]. In remote sensing (RS), a digital elevation model (DEM) is used to assess topographic properties of an underlying terrain. A DEM can be created by means of photogrammetry, such as airborne laser altimetry, or synthetic aperture radar [4], etc. With these methods, an object or shape is described by a set of 3D points, called point cloud, sampled on its surface. Generally, the position of each point is uniquely defined by Cartesian coordinate, (x, y, z) or $\mathbf{p} = [\mathrm{x\ y\ z}]^{\mathrm{T}}$. Typical photogrammetry collects these 3D points on an object surface by taking either passive or active approach, such as optical laser-based range, structured light scanners, and Light Detection and Ranging (LiDAR) scanners. With the most recent

The associate editor coordinating the review of this manuscript and approving it for publication was Miaohui Wang.

developments in optical sciences, these scanners have significantly enhanced the way in which we tackle reverse engineering and rapid prototyping. The fidelity of their reconstruction has so far evolved in lockstep with those produced by current CAD application software. This have enabled accelerated convergence of these key technologies, and hence their extensive adoption in computer graphics and vision, namely, modeling, recognition, and analysis of real environments. Accordingly, 3D scanners have been utilized in a range of applications across all domains of data-driven science, and at different scales. Even just before the emergence of Metaverse, the vast proliferation of computerized point cloud analyses had already been triggered and attained by competing invention and commercialization of low-cost real-time scanners, e.g., Microsoft™ Kinect [5]. These innovations have made profound impacts on various research and development areas, including automotive, machinery and artificial organ designs, archaeology, military and defense, urban planning, and digital laboratory, etc.

Three-dimensional estimation using stereo vision, has attracted considerable interests, mainly due to its relatively low implementation cost, and resource consumption. Nonetheless, its estimation quality highly relies upon accurate dense correspondence between cameras, which, in turns, is scene dependent. Mismatched pixels, due to image texture properties, may result in holes or incorrect topology, being reconstructed. Alternatively, monocular depth estimation (MDE) infers depth from single image, based on motion or visual cue. Its main advantage is dismissing the need to calibrate the alignment between cameras, and hence its induced errors. Thus far, it lacks certain judgments and mental perception, normally learnt by human experience. These factors have rendered it unsuitable for shape critical applications. These pros and cons, therefore, motivated this research in fusing depths inferred from multiple modalities.

The remainder of this paper is organized as follow: Section II surveys prominent algorithms, in 3D scene analysis. Section III describes the proposed method, which consists of point cloud acquisition, its estimation from a natural light (RGB) image, their information theoretic fusion, and mesh triangulation. Our main contributions are:

- Incorporating depth information, estimated from a natural light image by deep learning, and
- Introducing a novel ICP, based on entropy, for multimodal alignment.

Subsequently, experimental results on a public dataset are reported and discussed in sections IV. Finally, section V makes the concluding remarks and suggests future works that are worth investigated.

II. RELATED WORKS

This section provides detailed review on state-of-the-arts in 3D scene analyses, which consists of point cloud extraction and their fusion, and surface reconstruction.

A. POINT CLOUD EXTRACTION

In the most recent literature, the problem has been widely investigated. Li *et al.* [6] extracted point clouds from a low cost RGB-D sensor. That work emphasized on enhancing the depth quality, by incorporating semantic labeling of the environment by means of deep learning. Then, heuristic regularization was applied to noisy data by planar assertion. However, its application was limited to only known classes, trained by the neural network. An MDE approach was taken by another work [7]. To increase the resolution of a point cloud, extracted from an image, using encode-decoder network, it was enhanced by attention-based reconstruction. Although it could generalize to unseen shapes, that to their collections in a complex scene was not demonstrated. Similarly, Mandikal *et al.* [8] proposed 3D-LMNet, or a latent embedding matching method. In their work, a correct representation of an ambiguous point cloud was predicted by multiple plausible reconstructions. Rather than directly extracting a point cloud, Choi *et al.* [9], embedded randomly generated points onto shape information, extracted from an image. With that method, random cloud points were deformed in a coarse-to-fine manner to match both point-specific and shape features. A similar generative approach was also taken in [10], where 2D convolution operations were used to predict a 3D structure from multiple viewpoints. To determine optimal projections, pseudo-rendered depth maps were considered for geometric reasoning. Merits of these methods [7]–[10] were only demonstrated by experiments on the same public dataset of individual shapes.

B. POINT CLOUDS FUSION

Despite reasonably good rendition for most applications, several studies have shown that reconstructed surface quality and its coverage could be significantly improved by correlating multiple point sets, acquired at different aspects, lighting conditions, and timeframes, or perhaps with other modalities. In the literature, fusing two sets of points have been posed as transforming one set to another (or a group representation thereof) so that they are aligned. It has been done using various techniques, e.g., image analysis [11], [12], pose estimation [13]–[15], model reconstruction [16]–[18], and simultaneous localization and mapping (SLAM) [19]–[21]. Our survey on decades of research revealed some highly influent works, proposed for the task. They include fast global registration (FGR) [22], fast point feature histogram (FPFH) [23], robust point matching (RPM) [24], thin-plate spline robust point matching (TPS-RPM) [25], kernel correlation (KC) [26], Gaussian mixture models (GMMs) [27], coherent point drift (CPD) [28], and certainly, classic iterative closest point (ICP) [29]. These techniques may be characterized and examined by their point matching and optimization strategies, as follow. The FGR was proposed to overcome the limitation of conventional ICP. With this method, provided pre-computed FPFH [23] between two surfaces, they were densely aligned without initialization

nor correspondence query at each iteration, while optimizing the point registration. Gold *et al.* [24] proposed a robust method, called RPM, which optimized an objective function, derived from the mean field theory using soft-assigned (EM-like) and deterministic annealing. However, the RPM was limited to only affine and piecewise-affine transformation. To admit higher degrees of freedom (DoF), a feature-based non-rigid registration, also called TPS-RPM was suggested by Chui and Rangarajan [25]. Nevertheless, its extension to higher spaces beyond three-dimensions were not trivial. More recently, similar kernel-based matching was proposed by Tsin and Kanade [26]. In their work, registration between two sets of point cloud was posed as maximizing kernel correlation (KC) that was not only affinity measure but also a function of their quadratic entropy. Due to the weighted contributions of multiply linked dynamic points, KC objective was smooth and had unique convergence. However, the method was susceptible to noise and outliers within the object. It is evident that these methods implied Gaussian mixture model (GMM) on the point set distributions. On one hand, if these point sets were acquired from the same object, GMM can efficiently describe the underlying geometry and hence its solution (e.g., gradient and convergence) is trivial. On the other hand, when the distributions are not homogenous or initial correspondence is too far from the solution, these ICP variants may fail.

Right after the acquisition, an object is merely implied by cloud distribution, without any topological structure. Therefore, the next step is to impose local connectivity on the registered point sets, by surface interpolation. Normally in digital modeling and for efficient representation, a surface is interpolated piecewise and thus approximated over the point cloud by a mesh of simple geometry, such as triangle. To this end, many existing works employed Ball Pivoting Algorithm (BPA) [30]–[32]. Regardless of cloud acquisitions nor approximation methods used, most reconstructed scenes are often left with voids, due to occlusion and/ or inadequate sampling. As a consequence, hole filling is often required in post processing [33], as illustrated in **Figure 1**.

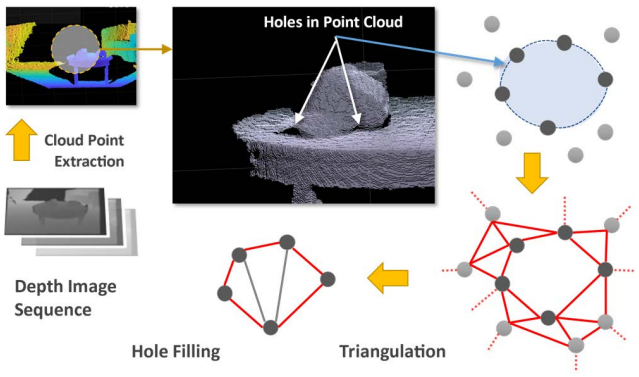

FIGURE 1. A conventional 3D scene reconstruction pipeline, consisting of point cloud extraction from a sequence of depth images, and triangulation, which often followed by hole filling.

To address these issues, this paper approaches the reconstruction of 3D scene through information theoretic fusion between point clouds, acquired from different modalities. Specifically, to lessen the need for hole filling, cloud points extracted from a depth scan were aligned and fused with those learned by a convolutional neural network (CNN), called ResNet-50 [34], from an RGB image of the same scene. Since pixels in a color image were continuously distributed, the 3D points extracted from their estimated depth could fulfill any void present in the first point set, before surface reconstruction. Motivated by the work of Tsin and Kanade [26], we modified ICP by incorporating cross entropy (CE) function, later referred to as CEICP, for robustly aligning these point sets. Subsequently, a BPA was used for surface approximation. Any inaccuracy due to image-to-depth estimation would be remedied by rolling α-shape. An overview of the proposed strategy is outlined in **Figure 2**.

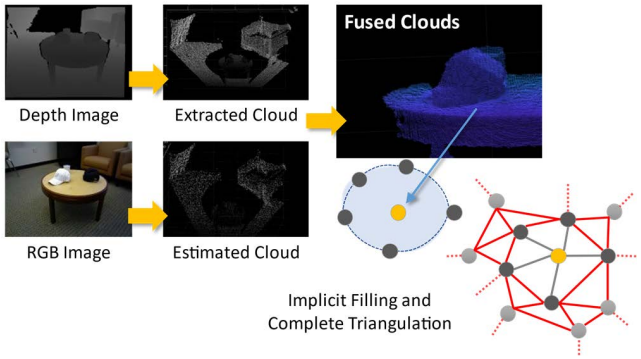

FIGURE 2. General concept of the proposed cloud fusion strategy. It consists of point cloud estimation from both depth and RGB images, their fusion to minimize void in the dataset, and surface triangulation.

C. SURFACE RECONSTRUCTION

Reconstructing an underlying surface from point cloud is ill-posed. It is one of the most investigated research areas in geometry processing. Like many inversed problems in engineering, unless the point cloud is precisely and systematically sampled on a given surface, the reverse does not necessarily hold, nor is it unique. For a simple surface of known class, e.g., quadratic, estimating its governing parameters could be as trivial as least squares fitting to the points. However, for those with much complex geometry, estimation is often done locally and then joined piecewise, e.g., by spline polynomial. On one hand, imposing such local structure on the final geometry is preferred for natural objects, as it is not only robust but also able to reduce acquisition error due to noise. On the other hand, doing so may adversely smooth out otherwise its salient features. Therefore, many applications make only an assumption on its local topology and reconstruct the surface by using a homeomorphic mesh of the original data [2], [35]. One of the most widely used techniques taking the latter approach and hence was also employed here is Ball-Pivot algorithm (BPA). Formally, BPA takes the input of N points,

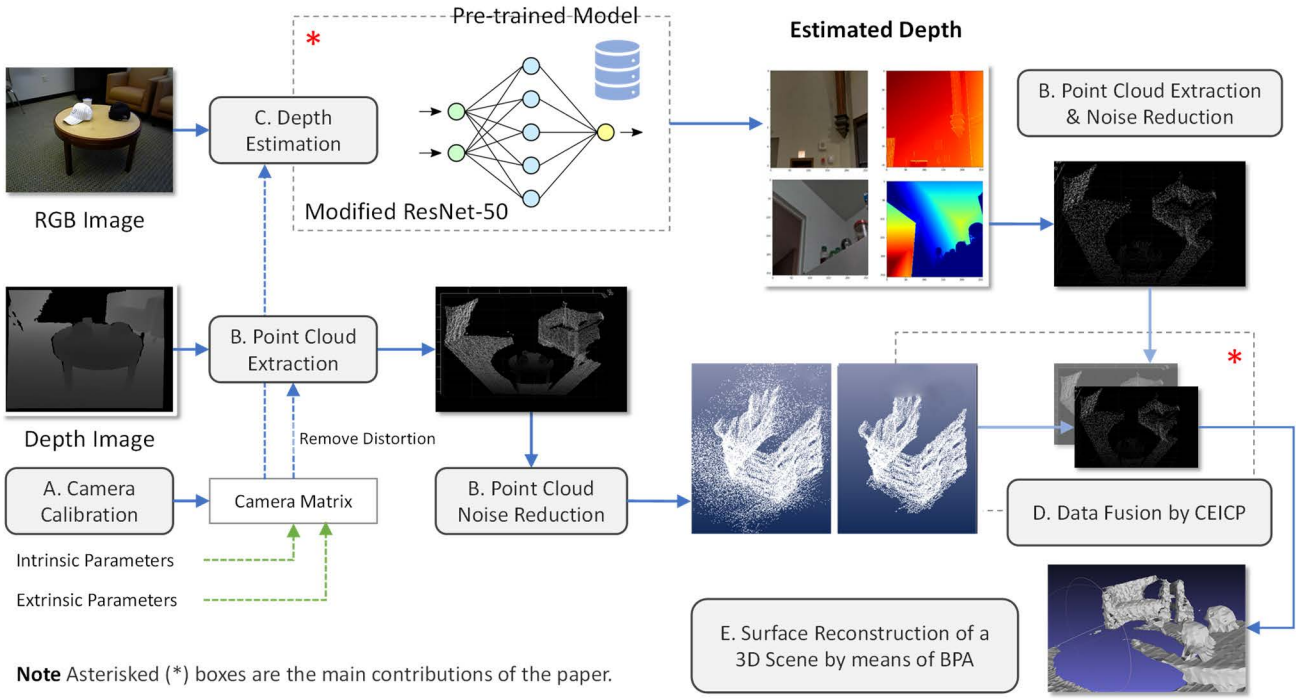

FIGURE 3. **Overview of the proposed multimodal fusion for 3D scene reconstruction. Its main contributions are highlighted by the asterisks, i.e., depth estimation by deep learning and data fusion by CEICP. Description of each step is elaborated in the following sub-sections, as annotated in each associated block (i.e., A to E).**

in $\mathbf{R}^{\mathrm{n}}$, sampled from an unknown surface U, as per Eq. (1).

$$\mathbf{P} = \left\{\mathbf{p}_i \in U | 1 \leq i \leq N, U \subset \mathbf{R}^n\right\} \tag{1}$$

BPA aims at finding a homeomorphic mesh, M, that best approximates this unknown surface. But the resultant mesh, typically of triangular type, may contain vertices that are not present, or do not connect to those in the original point set. Given a densely distributed point cloud, Bernardini *et al.* [30] created an approximating mesh as follow. Firstly, a seed triangle, whose vertices were in the initial point set, was chosen. After that, a ball (α-shape) with prespecified radius was pivoted on one of the edges in the seed triangle and rolled through the point cloud, until it collided with another point. A new triangle was formed by joining the newly discovered point to the pivoting edge. This process was repeated as the ball rolled through the cloud, joining triangles already created, until all points in the cloud were visited, and hence connected to the mesh. BPA process is data-driven but insensitive to noise, as it is tapered off by a pivoting ball. Additional constraint was imposed in another work by H. Seo et. al [31], where mesh-aware BPA (MABPA), was introduced. Given one or more oriented manifolds, each embedded on a triangular mesh, the MABPA generated a concave hull mesh, which comprised of the outmost triangles in the input meshes, and whose all vertices included some of those in the same. The method may be generalized to meshing of virtual membrane around a human internal organ. In RS application, in order to remove holes, found in flat ground regions of LiDAR scans, typically caused by a single-layer point cloud being unable to form tetrahedra in 3D Triangulated Irregular Network (TIN), Ma *et al.* [36] proposed an improved BPA, based on spatial sorting of points, by their distance and angles. As a result, bottom boundaries could be directly identified without having to first construct and process TIN. Existing works based on BPA similarly created a surface mesh by expanding the radius of the α-shape, i.e., ball radius (BR). However, as this radius increased, the quality of a produced mesh degraded progressively, as more holes emerged. This is due to larger ball discarding points within closer proximity of its pivot triangle, and hence connecting further candidates, rather than its apparent neighbor. This inevitably left unmeshed points in the final TIN model.

III. PROPOSED METHOD

This paper proposes a novel 3D mesh reconstruction method by fusing multi-modal point clouds, i.e., those acquired by depth scan and inferred from the corresponding RGB image. Its description is outlined by the diagram in **Figure 3**.

The multi-modal input consists of two data pipelines. Firstly, a set of cloud points were extracted from the depth image of a scene. Physical coordinate of a given point was computed from its depth, taken into the account camera parameters, calibrated per an acquisition. In the second pipeline, another set of point cloud were inferred by a pre-trained deep learning (DL) network from the natural light (RGB) image of the same scene. The first point cloud

then had its outliers reduced by spatial noise filtering. Subsequently, the proposed CEICP was used to merge both point clouds, from which 3D TIN was finally reconstructed by using BPA. The following subsections have detailed description of this process. More specifically, they are A) camera calibration, B) point cloud acquisition, C) depth estimation by deep learning from an RGB image, D) entropy-based point cloud fusion, and E) surface reconstruction by using BPA.

The photographs of the system setup are shown in **Figure 4**. The RGB-D camera used in this study was Kinect^{TM} for Xbox One^{TM}. It consisted of an infrared (IR) depth and RGB cameras, whose spatial resolution and frame rate were 640 × 480 pixels and 30 frame per second (FPS), respectively.

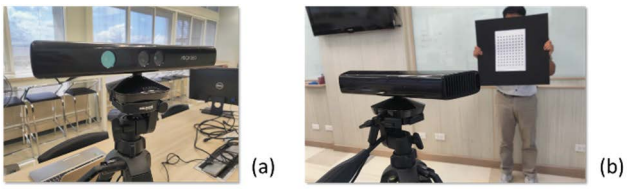

FIGURE 4. Pictures of system configuration, illustrating a Kinect^{TM} for Xbox One^{TM}(a) and a person holding a planar object (b).

Although it was anticipated that the proposed system could be generalized to fusing a static scene with a moving camera, so as to capture it at different aspects for better 3D coverage, in the subsequent experiments, the results are reported only for a pair of depth and RGB images, taken at the same time.

A. CAMERA CALIBRATION

This is a key step toward scene reconstruction from an image. It tried to ascertain the geometric parameters governing image acquisition process [37]. The parameters considered in this study were those of the camera (i.e., focal length, principal point, and skew of axis) and its geometry (i.e., rotation and translation), later referred to as intrinsic and extrinsic parameters, respectively. Given a calibration pattern of well-defined geometry, these parameters were then estimated from known physical points (in the real-world) and their projection on the image plane. To accurately estimate these parameters, optical distortion had to be corrected by Eq. (2).

$$\begin{aligned} u &= x \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \\ v &= y \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \end{aligned} \tag{2}$$

where (x, y) and (u, v) were the coordinates in a depth image before and after correction, respectively. The variables, k_1, k_2, and k_3, were radial distortion coefficients, while r was the distance from (x, y) to lens center, defined by $r^2 = x^2 + y^2$.

1) INTRINSIC PARAMETERS

In this study, the intrinsic parameters of the camera consisted of focal length $\mathbf{f} = (f_x, f_y)$ and optical center $\mathbf{c} = (c_x, c_y)$ of its lens. The former refers to the point, at which light rays converge to produce a sharp and detailed image, on the sensor, while the latter does to the point, where light rays are undeflected, while passing through the lens. Passing through any other point, light rays would deflect toward to or away from the optical center, **c**. In the calibration, these parameters were defined by a 3 × 3 camera matrix in the homogeneous coordinates, as expressed in Eq. (3).

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2) EXTRINSIC PARAMETERS

The extrinsic parameters described the relative position of the camera in 3D space. They were defined by a rotation matrix, **R**, and a translation vector, **t**, which sent the camera from the origin to its current position. Suppose that the projection was perspective. Once calibrated, a point seen by this camera and appeared at (u, v) on the sensor (i.e., corrected depth image), would correspond to that at (x, y, z) in the real-world coordinate systems, as expressed in Eq. (4):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}^{-1} \left(\mathbf{K}^{-1} \begin{bmatrix} u_h \\ v_h \\ h \end{bmatrix} - \mathbf{t} \right) \tag{4}$$

where h is homogeneous coordinate and $u_h = u \cdot h$ and $v_h = v \cdot h$. Note that h depended on the distance from a point to the sensor in the camera coordinate system. However, since u_h and v_h also varied as h, so during the calibration, h could be eliminated, this reducing the rank of Eq. (4) to 2. Note that these parameters were applied only to the depth image, to demonstrate that it may be taken independently of the RGB one.

3) DUAL CAMERA SETTING

In the experimental setting, the dual-camera system consisted of depth and RGB cameras, equipped with infrared and natural light sensors, respectively. These cameras were mounted in a canonical configuration. The picture used for calibration, depicted in **Figure 5a**, contained 99 black circles, regularly distributed at 9 × 11, and spaced at 35mm intervals, on a white background. It was printed and attached to a black cardboard, serving as the calibration plane, as shown in **Figure 5b**.

The photographs of this plane were taken by the cameras aiming at two different viewpoints. Each viewpoint captured, eight different plane orientations, four of which were used for determining the camera parameters, while the others were for testing. Their dimensions were 1030 × 1380 pixels.

Figure 6 illustrates the 3D points, representing the centers of the reconstructed circles.

In the first viewpoint, shown in **Figure 6(a)**, green, blue, pink, and black dots represent the points associated with 1st, 2nd, 3rd, and 4th plane orientations, respectively. Likewise, those displayed in **Figure 6(b)** were reconstructed from the pictures taken at the second viewpoint. It is thus safe to conclude that the calibration model, as expressed by the Eq. (2)-(4) was sufficient for the current problem.

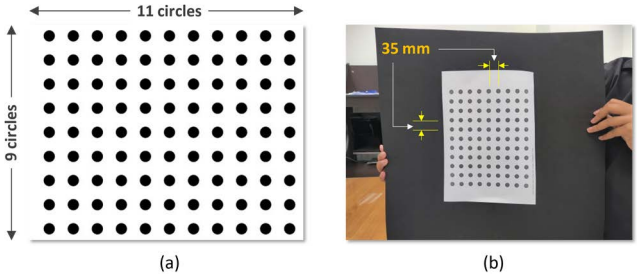

FIGURE 5. The calibration pattern consisting of 9 × 11 black circles on a white background (a) and its printed version on a cardboard (b).

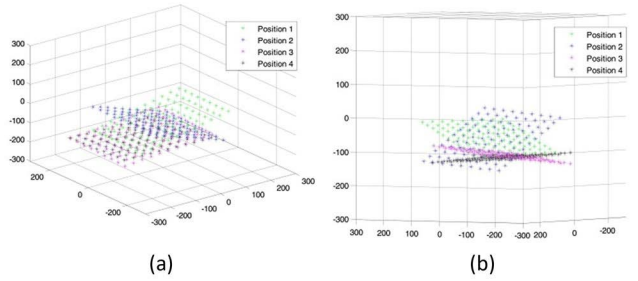

FIGURE 6. Extracted patterns at 4 different orientations, taken in viewpoints 1 (a) and 2 (b).

In our setting [5], since depths were presented on the same matrix as the RGB image, both depth and RGB images were then corrected for distortion, using Eq. (2) and (3), with the same intrinsic camera parameters (shown as dotted arrows in **Figure 3**). Lastly, the corresponding images were resampled, by using bilinear interpolation model.

B. POINT CLOUD ACQUISITION

Conversion of the depth map from a binary calibration pattern into point cloud was trivial, due to clear distinction between objects (circles) and background. Doing so from a real scene was, however, much challenging. **Figure 7** illustrates a depth map and the corresponding point cloud, extracted from an actual scene. It is evident that, due to imperfect lighting condition, unaccountable lens distortion, and noise, matching dual camera images could be inaccurate, causing ambiguities in the depth map and hence contaminating the corresponding point cloud with outliers, voids, and spatial noise [37].

Consequently, cloud point meshing might fail to produce acceptable result. However, if excessive regularization was used, the reconstructed mesh might lack some crucial details, or had incorrect geometry. To address this issue, i.e., preserving their feature while denoising, a statistical-based structural adaptive filter [38] was applied to the extracted points. Supposed that local distribution of this point cloud was Gaussian, an outlier would be removed, if it lied outside this distribution. Let $\mathbf{p}$ be a point in the cloud. Then, an average distance to its N nearest neighbors was computed by Eq. (5).

$$d(\mathbf{p}) = \left(1/N\right) \sum\nolimits_{\mathbf{q} \in \Omega_N(\mathbf{p})} \|\mathbf{p} - \mathbf{q}\| \tag{5}$$

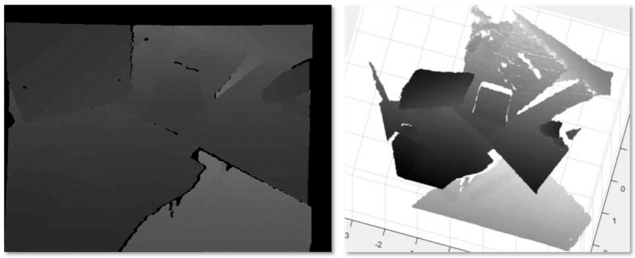

FIGURE 7. An example of a depth image (a), from which the point cloud was extracted (b). It can be seen that lens distortion and noise are mostly regulated, but there remain outliers and some regions with undefined depth.

Let μ and σ be mean and standard deviation, respectively, of d over all points. Then, a point would be considered outlier, unless an average distance to its N nearest neighbors was within a specified threshold. In other words, given an original cloud, $\mathbf{P}_o$, then its smoothed version, $\mathbf{P}$, is defined in Eq. (6).

$$\mathbf{P} = \{\mathbf{p} \in \mathbf{P}_o | d(\mathbf{p}) \leq \mu + t \cdot \sigma\} \tag{6}$$

where t was an empirical threshold. In this study, N and t were set to 4 and 1.0, respectively. Since the threshold, t, was specified, based on the statistics of distances, this filter was thus robust and adaptive to sparse points and gross outliers.

C. DEPTH INFERENCE FROM AN RGB IMAGE

Depth estimation from images is a fundamental task in computer vision and many other applications, e.g., SLAM, navigation, object identification, and semantic segmentation, etc. Particularly, the task is a vital step toward inferring geometrical properties of an underlying scene.

However, since depth acquired from a stereo camera relies on establishing the correspondence between images, by using Epipolar geometry, the quality of the extracted cloud points thus greatly depends on the surface properties, its continuity, its texture pattern and repetitions, and lighting environment. These factors are known to contribute to erroneous 3D reconstruction, such as voids, ambiguity, and impaired or missing features, etc. we anticipated that, these errors could be alleviated, by coalescing with much regularized depths, by other interpretation, e.g., visual cue. One of the main contributions of this study is, therefore, reconstruction by fusion. To this end, the second set of point cloud being fused was inferred from an RGB image by using CNN. Unlike those previously extracted from an infrared camera, in this section, depth values were estimated from a natural light image.

Motivated by a self-supervised method proposed by Godard *et al.* [39], where depth map was estimated by a combination of network architectures. The method predicted depth using a fully connected U-Net and poses between image pair using a pose network, with ResNet-18 as encoder. Moreover, the weights were initialized by pre-trained ImageNet.

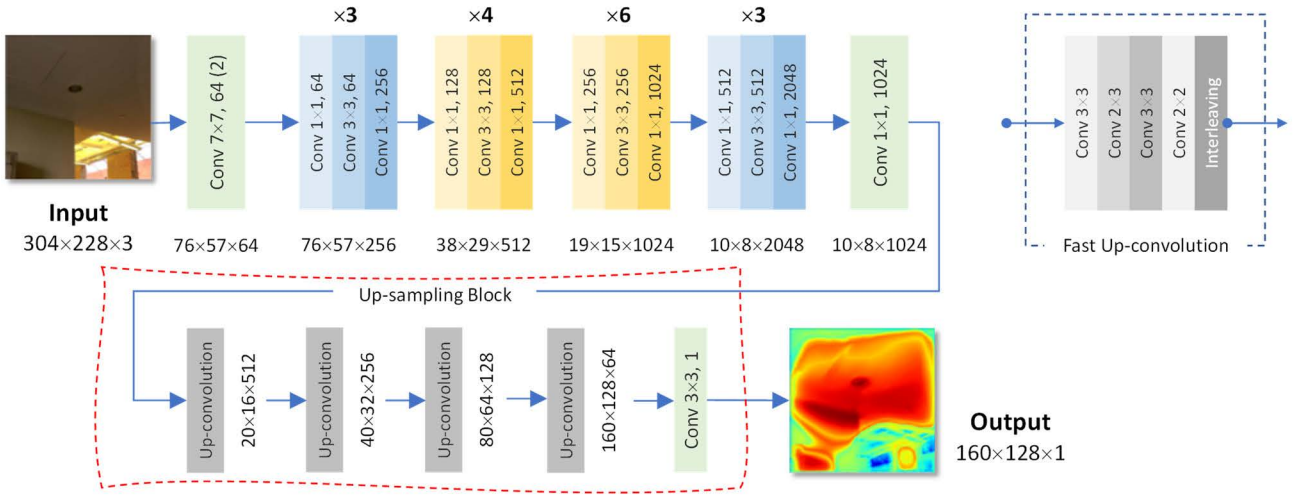

FIGURE 8. Modified ResNet-50 architecture [39], for depth estimation, where fully connected layer was replaced by up-sampling blocks.

In this study, the modified ResNet-50 [39], shown in **Figure 8**, was adopted. The network was trained with RGB and corresponding depth images, as input and target, respectively. Imaging data, all of size 304 × 228 pixels, were obtained from the KITTI dataset [40]. The network was configured with 22 layers, using 32 batch size, 0.0002 learning rate (LR), and 30 epochs. During training, appearance-based loss function was used. Furthermore, we introduced a modified minimum re-projection loss, calculated per each pixel, and eigen splits were used to estimate the final depth map. Some examples of the estimations by our method were illustrated in **Figure 9**.

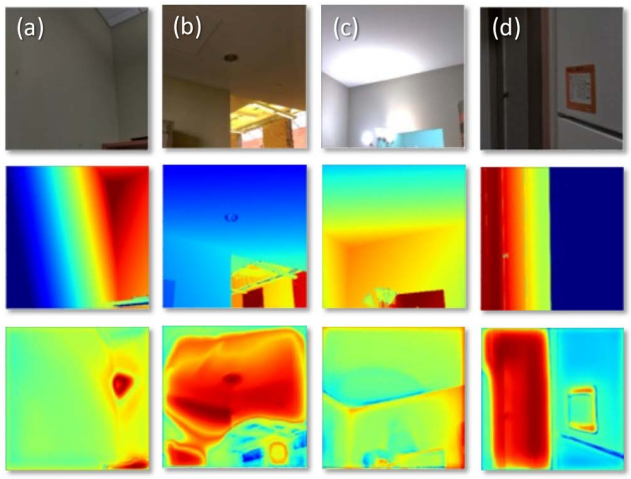

FIGURE 9. Four examples (a-d) of RGB images (top), and the respective ground truth (middle) and estimated depth images (bottom). It is notable that these estimates correspond to ground plane perception, vanishing point, and relative size of objects in the scene, etc.

Similar to the previous subsection, a resultant depth map was converted into a dense point cloud. The next subsection describes in detail, how to fuse this point cloud, as estimated by the modified ResNet-50, with that acquired by an infrared camera, by taking an information-theoretic approach.

D. POINT CLOUD FUSION

It has been well accepted that fusion of geometrical data from different sensors can help enhancing the quality of 3D model reconstruction, over that obtained by only one sensor alone. Missing information in one dataset, can be completed or implied by that present in another. Therefore, in order to ensure valid reciprocal, a dense correspondence between source and target clouds must be established. One of the most effective methods, often employed in the literature is iterative closest point (ICP), whose basic algorithm and proposed modifications are presented below.

It is worth mentioned here that, ICP was usually applied to align dynamic objects. But in the presented case, it was assumed that no movement existed between the extracted and estimated cloud points. Nevertheless, they appearance did differ, due to complementary interpretation of depths, i.e., based on Epipolar geometry and visual cue learnt by a CNN, respectively. Not only did voids occur in one set and not in the other, but their geometrical features might also not coincide. These called for a special treatment on their similarity metric, as proposed herewith.

1) ICP FUNDAMENTALS

ICP algorithm is invaluable for reconstructing an object from multiple surfaces, aligning an anatomical model to a patient specific scan, localizing a moving robot, and optimizing its path planning (particularly, when an equipped wheel odometry is unreliable due to slippery terrain), among other applications. Basically, ICP tries to find the best transformation that sends a given source point to its closest (or the most possible match) target point, so that overall difference, with respect to some measure, between these point sets is minimized. Nonetheless, depending on an initial orientation and

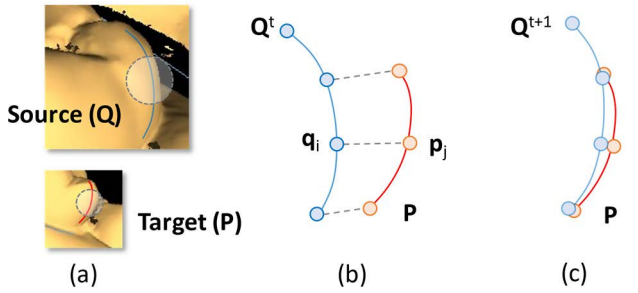

FIGURE 10. Given source (Q) and target (P) points (a), ICP tries to establish the closest pairs (dashed lines) (b) and iteratively (t → t+1) find the transformation that best matches them (c).

TABLE 1. Basic iterative closest point algorithm [29].

ALGORITHM	Iterative Closest Point (ICP)
INPUT	Target (P) and Source Points (Q)
OUTPUT	Transformation T: Q → Q′
BEGIN	
1:	COMPUTE $\mathbf{p}_{CG}$ and $\mathbf{q}_{CG}$ // centroid
2:	TRANSLATE Q → Q − ($\mathbf{p}_{CG}$ − $\mathbf{q}_{CG}$)
3:	WHILE $\|Q^{t+1} - Q^t\| > \epsilon$ AND $t < T$ DO
4:	FOREACH $\mathbf{q}_i$ in Q^t DO
5:	$\mathbf{p}_j, w_{ij}$ = closest ($\mathbf{q}_i$, P)
6:	// weight w_{ij}, normally is 1
7:	END FOREACH
8:	$\mathbf{T}^t = \min_T \sum_i w_{ij} \|\mathbf{p}_j - \mathbf{T}(\mathbf{q}_i)\|^2$
9:	$Q^{t+1} = \mathbf{T}^t(Q^t)$
10:	$t = t + 1$
11:	END WHILE
END	

capture range, these matched pairs may not all be exact correspondences, hence the transformations may not be unique at first. So, ICP repeats this process, where correspondences are iteratively updated until converges. **Figure 10** displayed a diagram of a conventional ICP procedure [29], and **Table 1**, describes its algorithm.

Conventional ICP is effective, but it remains inefficient. Its convergence rate is low, especially for a pair of cloud points, whose resolutions were higher. Therefore, as another contribution, this paper also proposed a modified ICP, based on cross-entropy and called it CEICP.

2) CROSS-ENTROPY ICP (CEICP)

In a typical registration, ICP tries to align multiple point sets, with respect to a distance measure. However, such measure is generally sensitive, resulting in ICP performing poorly, when either dataset contains a significant number of outliers or has higher noise floor [41]. These are especially the case in photogrammetry, where depths are evaluated, rather than fathomed directly on an object surface. In addition, point clouds involved in this study, although were of the same scene, but acquired from two imaging modalities, with different not only precision but also interpretation of depths, i.e., by correlating projected light patterns and by deep learning from visual cue. Therefore, their appearances differed and determining their similarity by mutual information is much pertinent than by spatial difference.

The proposed CEICP aims at finding the correct correspondence between two sets of cloud points. In other words, it was the one that gave the most appropriate probability distributions of random variables, drawn from these sets. By definition, cross-entropy (CE) between distributions P and Q of the same underlying event (i.e., 3D scene) measures the average number of data unit (or bits) required to uniquely identify a co-existing event (p, q), where $p \in P$, and $q \in Q$. Suppose that Q was an estimated probability distribution of the true distribution, P. At each iteration, an index mapping between a point, p, to its nearest one, q, based on the current rigid transformation, was determined. Then, a new transformation that sent all points in P to their closest ones in Q, with the least errors, was computed. Similar to basic ICP, the process was repeated, but while maximizing CE, until convergence. The cross-entropy of the distribution Q, relative to the distribution P, over the a given sample space, $H(P, Q)$ is expressed in Eq. (7).

$$H(P, Q) = E_{PQ}\left[-\log f(p, q)\right] \tag{7}$$

where $E_{PQ}[\cdot]$ is an expected value operator, with respect to the joint probability of both distributions.

The above definition may be formulated using Kullback-Leibler divergence [42], $D_{KL}(Q|P)$, of Q from P, which is also known as the relative entropy of Q with respect to P, i.e.,

$$H(P, Q) = H(Q) + D_{KL}(Q|P) \tag{8}$$

In Eq. (8), $H(Q)$ is the entropy of Q.

For probability distributions, P and Q, defined on the same support, X, relative entropy D_{KL} can be constructed by measuring extra information, required to encode samples from P using a code optimized for Q. Since X was a surface embedded on $\mathbf{R}^3$, the information was projected onto its L^2 norm. In addition, since $H(Q)$ was constant during ICP, then maximizing Eq. (8) is equivalent to maximizing the information, I, as defined in Eq. (9).

$$I = -\sum\nolimits_i f\left(\|p_j - q_i\|\right) \log f\left(\|p_j - q_i\|\right) \tag{9}$$

It is clear that Eq. (9) was maximized when both P and Q were perfectly aligned. In addition, **Figure 11** demonstrates example evaluations of **Eq. (9)**, as CEICP proceeded.

However, finding for q_j, the best correspondence (i, j) that maximizes I was computationally expensive. Each respective information was thus implied via w_{ij} in **TABLE 1**, instead. Let $\mathbf{P}$ and $\mathbf{Q}$ be the input point clouds, obtained from an infrared depth image, and inferred by ResNet-50 from a photograph of the same scene, respectively. We then sought for a transformation, $\boldsymbol{T} = \mathbf{R}|\,\mathbf{t}$, that satisfied the best correspondence between these point sets. Firstly, for every point $\mathbf{q}_i$ in $\mathbf{Q}$, find the nearest point $\mathbf{p}_j$ in $\mathbf{P}$, with respect to their Euclidean distance, d_{ij}, given in Eq. (10).

$$d_{ij} = \|\mathbf{p}_j - \mathbf{q}_i\| \tag{10}$$

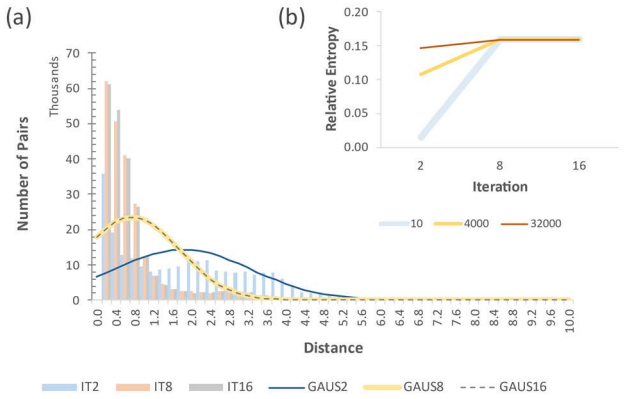

FIGURE 11. As probability distributions of L^2 norm between pairs (i, j) (bars) and their Gaussian fitted curves (lines) got sharper (a), during the course of CEICP, i.e., at iterations 2, 8 and 16, the relative entropy delegate I also increased, as illustrated by a few selected point pairs, i.e., 10, 4 000, and 32 000 (b), hence driving weights w_{ij} to the correct alignment.

For a pair $x_{ij} \in X$ compute its CE, $H(p, q)$, by using Eq. (8) and (9). Then, if the distance d_{ij} was more than a predefined threshold, T_d, the pair may be an outlier, and hence was discarded from the support, X. Subsequently, for each of remaining pairs (i, j), its contribution to the alignment was given by the information weight, w_{ij}, expressed in Eq. (11).

$$w_{ij} = -f\left(d_{ij}\right) \log f\left(d_{ij}\right) \tag{11}$$

where f was the Gaussian distribution of d, computed at each iteration. Given this correspondence, X, the rotation ($\mathbf{R}$) and translation ($\mathbf{t}$) matrixes that transformed $\mathbf{P}$ to $\mathbf{Q}$, was determined by using singular value decomposition (SVD). To this end, the centroids of $\mathbf{P}$ and $\mathbf{Q}$ were computed, taking into account only points supported by X, and expressed as $\mathbf{c}_p$ and $\mathbf{c}_q$, respectively. Then, both point clouds were translated following Eq. (12), so that their respective centroids coincide with the origin, and denoted as $\mathbf{P}'$ and $\mathbf{Q}'$, respectively.

$$\mathbf{c}_p = \frac{1}{N_j}\sum_j \mathbf{p}_j \mathbf{c}_q = \frac{1}{N_i}\sum_i \mathbf{q}_i \tag{12}$$

Next, the covariance matrix, weighted by $\mathbf{W}$, was evaluated, and decomposed by using SVD, as per Eq. (13) and (14).

$$\mathbf{W} = \mathbf{diag}[w_{ij}] \tag{13}$$

$$\mathbf{P}'\mathbf{W}\mathbf{Q}'^{\mathrm{T}} = \mathbf{U}\Lambda\mathbf{V}^{\mathrm{T}} \tag{14}$$

Finally, the matrixes $\mathbf{R}$ and $\mathbf{t}$, which best moved $\mathbf{Q}$ to $\mathbf{Q}*$, closest to $\mathbf{P}$, were given by Eq. (15) and (16), respectively.

$$\mathbf{R} = \mathbf{V}\mathbf{U}^{\mathrm{T}} \tag{15}$$

$$\mathbf{t} = \mathbf{c}_p - \mathbf{R}\mathbf{c}_q \tag{16}$$

At each iteration, the error of a resultant transformation, $\mathbf{T}$, i.e., $E(\mathbf{R}, \mathbf{t})$, was given by Eq. (17).

$$E(\mathbf{R}, \mathbf{t}) = \sum_{(i,j)\in X} w_{ij} \left\| \mathbf{p}_j - (\mathbf{R}\mathbf{q}_i + \mathbf{t}) \right\| \tag{17}$$

It was expressed as the weighted distances between $\mathbf{P}$ and $\mathbf{Q}*$. CEICP repeated these steps, until the required accuracy was met, convergence, or the iterations reached a maximum number (t_{MAX}). Once completed, both $\mathbf{Q}*$ and $\mathbf{P}$, were then merged into a single point cloud. An example of CEICP result and the corresponding fusion is illustrated in **Figure 12**.

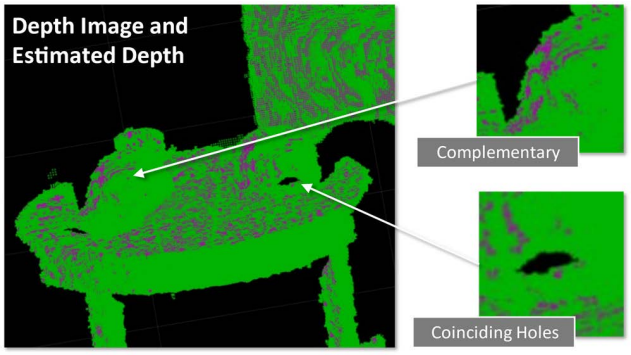

FIGURE 12. Fusion between point clouds, extracted from a depth image and estimated by modified ResNet-50, demonstrating cases when they were complementary and when their holes coincided.

E. SURFACE RECONSTRUCTION

This section describes the detailed implementation of BPA based surface reconstruction. Given fused points cloud from the previous step, a 3D TIN of a scene was created.

1) BALL PIVOTING ALGORITHM

The technique, proposed by Bernardini *et al.* [30], modeled an α-shape as a rolling ball. It was based on an assumption that if a sampled dataset $\mathbf{P}$ is sufficiently dense, then a sphere of a specified radius cannot traverse through it without colliding with one or more points within. Therefore, BPA begins with mounting initial three points with a ball. This ball then pivots while maintaining its contacts with two of these points until it contacts another. This step is shown in **Figure 13**.

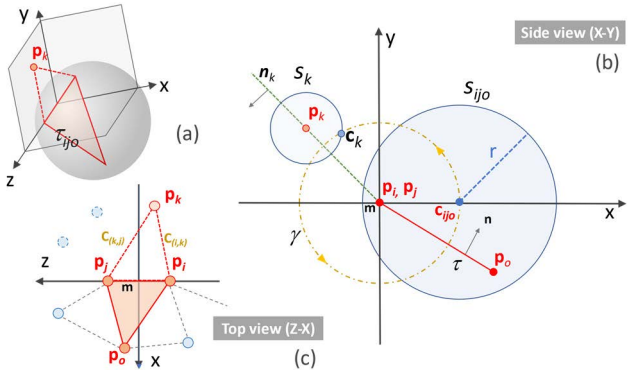

FIGURE 13. The diagram of BPA, adapted from [25], illustrating a sphere pivoting from a triangle τ_{ijo} as it hits a point p_k (a) and the corresponding side view (X-Y) (b) and top view (Z-X) (c).

BPA was initialized by a seed triangle, chosen within the point cloud. Specifically, it is defined by a selected point and its two nearest neighbors. Provided the triangle τ_{ijo} consisting of vertices, $\mathbf{p}_i$, $\mathbf{p}_j$, and $\mathbf{p}_o$, a ball of radius r was pivoting on

the edge ($\mathbf{p}_i$, $\mathbf{p}_j$), which lied on the z-axis. Before proceeding, the sphere was checked, whether it enclosed another point in the cloud, otherwise a new triangle was chosen. The local coordinate, depicted in **Figure 13**, is specified such that the midpoint of this edge (m) is at the origin. Initially, this r-ball intersected the (x-y) plane at the circle s_{ijo}, centered at $\mathbf{c}_{ijo}$.

While pivoting on the edge ($\mathbf{p}_i$, $\mathbf{p}_j$), the ball center moved along the trajectory γ, i.e., around $\mathbf{m}$ with radius $\|\mathbf{c}_{ijo} - \mathbf{m}\|$. When the ball discovered a new point, $\mathbf{p}_k$, it intersected the (x-y) plane at the new circle, s_k, and its center now moved to $\mathbf{c}_k$. Orientation of the new intersection line from $\mathbf{m}$ to $\mathbf{p}_k$, and hence the newly found triangle ($\mathbf{p}_i$, $\mathbf{p}_j$, $\mathbf{p}_k$) were defined by $\mathbf{n}_k$. This process was repeated, by pivoting on an unvisited edge, until all points were exhaustively traversed. The resultant mesh then approximated the underlying 3D surface.

This algorithm was very efficient because it required linear time and storage [43]. However, its challenges were determining appropriately the ball radius or handling point clouds, more sparsely sampled than one, and also those containing excessive voids. Nonetheless, the latter issues were already remedied by the proposed multimodal fusion strategy.

IV. RESULTS AND DISCUSSION

This section presents and discusses the experimental surface reconstructions of selected scenes by using the proposed method. It is divided by key processes into four subsections, namely, camera calibration, depth estimation by deep learning, cloud point fusion, and finally surface reconstruction. In particular, since fusion based on cross-entropy is the contribution of this paper, it was benchmarked against not only the baseline algorithm, but also various state-of-the-art methods.

In these experiments, MATLAB^{TM} v.2020a was used to implement camera calibration, depth estimation, point clouds extraction, their noise reduction, and fusion of the same. The surface reconstruction was written in Python v3.10. All codes were executed on a Windows® PC equipped with an Intel® Core^{TM} i7-7700HQ CPU 2.81GHz, and 8 GB RAM.

A. CAMERA CALIBRATION

To evaluate the performance of this process, two experiments, assessing its robustness against noise when recovering camera parameters, were carried out. Gaussian noises of zero mean and 0–1.0 standard deviations were added to the calibration image at different orientations and views, in turn. The recovered intrinsic and extrinsic parameters were reported in **TABLES 2** and **3**, respectively. In the former, f_x and f_y were the focal length in respective axes, (u_0, v_0) was lens center, and k_1 and k_2 were lens radial distortion in mm^{-2}, mm^{-4}, respectively. In the later, t_x, t_y, and t_z were the translations, while r_x, r_y, and r_z, were the rotations, in respective axes.

For most recovered intrinsic parameters, the values were well within ±0.5% errors, except for radial distortions, which were quite sensitive, and exhibited large errors (bold red), at higher noise levels. However, errors of the extrinsic parameters were bound by ±5%, throughout the range of noise levels. These suggest that noise level should be kept below 0.5.

TABLE 2. Recovered camera intrinsic parameters.

Noise	f_x	f_y	u_0	v_0	k_1	k_2
0.0	500.000	500.000	300.000	250.000	0.050	0.100
0.2	500.013	500.009	300.132	249.855	0.050	0.102
0.4	499.738	500.343	300.273	249.762	0.044	0.105
0.6	498.836	501.538	299.317	249.346	0.063	0.094
0.8	501.329	500.938	299.021	250.829	0.085	**0.127**
1.0	502.474	497.638	301.327	251.043	**0.138**	**0.063**
avg.	**500.398**	**500.078**	**300.012**	**250.139**	**0.072**	**0.099**
std.	*1.2930*	*1.3346*	*0.8096*	*0.6579*	*0.0356*	*0.0208*

TABLE 3. Recovered camera extrinsic parameters.

Noise	t_x	t_y	t_z	r_x	r_y	r_z
0.0	60	-10	18	30	20	18
0.2	60.185	-10.072	18.132	30.0012	20.0008	18.0023
0.4	60.173	-10.193	17.757	30.0027	19.9969	18.0053
0.6	60.237	-9.661	18.329	29.9965	20.0053	18.0072
0.8	59.538	-10.472	18.517	30.0051	20.0073	17.9929
1.0	60.621	-10.613	17.104	30.0062	20.0089	17.9892
avg.	**60.126**	**-10.169**	**17.973**	**30.002**	**21.670**	**17.999**
std.	*0.3535*	*0.3422*	*0.5001*	*0.0035*	*4.0794*	*0.0071*

B. DEPTH ESTIMATION

Compared to the actual measures reported in KITTI indoor dataset [40], the errors of depth estimated by the proposed CNN, based on the modified ResNet-50 [39], are reported for each scene in **Figure 14**. Root-mean-square error (RMSE) and mean absolute relative error (REL) in percent, for the scenes 1, 2 and 3, were 6.328, 7.944, and 5.647, and 0.206, 0.361, and 0.161, respectively. On average, RMSE of the estimates were 6.640 ± 0.963. However, RLE of 0.243 ± 0.086 indicated that, despite favorable homogeneity, its accuracy was inferior to that acquired directly from depth image.

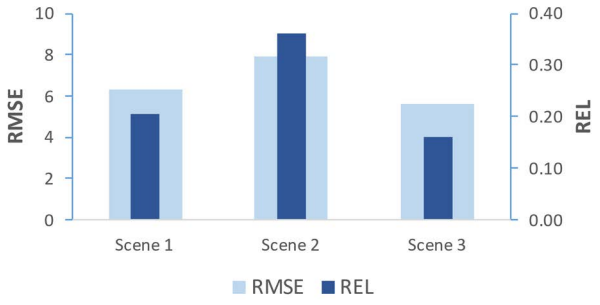

FIGURE 14. Errors of the estimated depth by ResNet-50 from 3 scenes. Despite consistently low errors, they were inferior to direct extraction.

C. CROSS-ENTROPY POINT CLOUD FUSION

It is evident from the previous result that the depth information learned and estimated by ResNet-50 was reliable but not highly accurate. Therefore, in this study, it was complementary to that extracted from the infrared scan, via fusion.

To evaluate the proposed CEICP fusion, it was compared against existing works, i.e., basic ICP [29], picky ICP (PICP) [44], RICP [45], multi-resolution ICP (MRICP) [46], fractional ICP (FICP) [47], hue ICP (HICP) [48]. These methods were benchmarked on RGB-D dataset, obtained from KITTI. This dataset consisted of three scenes, each containing 100 frames, taken by Velodyne HDL-64E depth and FL2-14S3C-C color cameras. The error metric presented in **TABLE 4** is average Hausdorff distance, yielded by each method for the examined scenes. It is evident that, the proposed CEICP outperformed its counterparts and mostly on par with MRICP.

TABLE 4. Average hausdorff distances given by different methods.

Method	Scene 1	Scene 2	Scene 3
ICP [29]	8.21	10.12	23.84
PICP [44]	97.25	19.33	85.51
RICP [45]	19.91	21.13	32.39
MRICP [46]	4.91	6.64	12.51
FICP [47]	6.51	15.78	39.7
HICP [48]	9.54	12.09	18.97
Proposed	**4.361**	**5.5484**	**7.6874**

Furthermore, the convergence of the proposed fusion was compared with those of its counterparts in **Figure 15**. Plotted in this graph are PICP and MRICP which, respectively, exploited hierarchical point selection and KD-tree search to accelerate registration over a conventional ICP. Visual examples of the missing data being completed by the proposed fusion are illustrated in **Figure 16**.

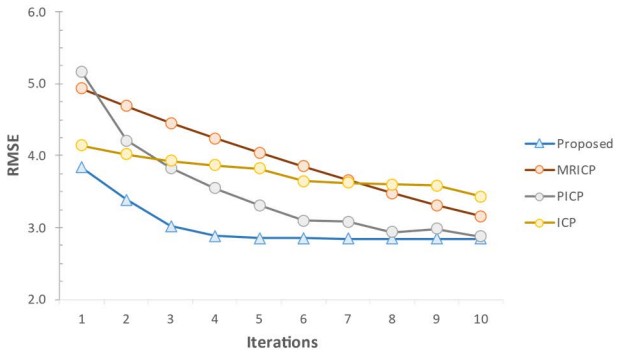

FIGURE 15. Convergence behavior of the proposed CEICP of a selected scene, compared to some existing methods, i.e., MRICP, PICP and ICP. It is notable that the proposed method outperformed these techniques.

D. SURFACE RECONSTRUCTION

This subsection reports the result of surface reconstruction by BPA. in **Figure 17** illustrates surfaces, reconstructed from a point cloud prior to fusion, at varying ball radii, i.e., relative to its average distance to a nearest neighbor. It is evident that there remained several holes in these surfaces, even with the radius as large as $r_D+2\sigma_D$. However, after multimodal fusion, they were significantly reduced, as shown in **Figure 18**.

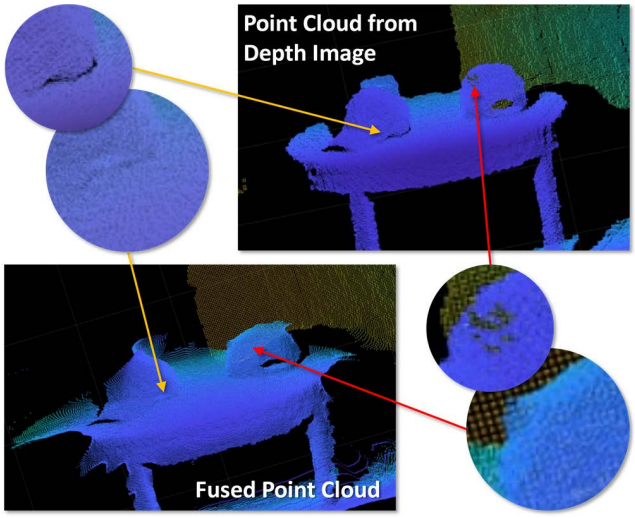

FIGURE 16. Two examples of missing data being successfully completed. Also note some remaining holes, even after the fusion.

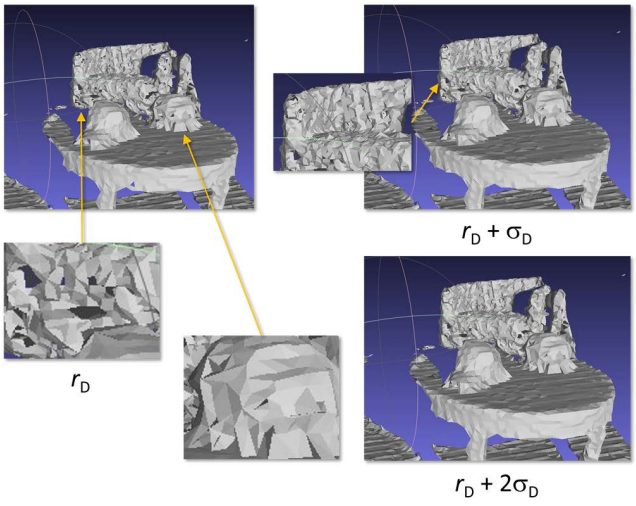

FIGURE 17. 3D TIN reconstructed with varying radii, where r_D and σ_D are mean and standard deviation of distance to a nearest neighbor.

E. DISCUSSION

The abovementioned results may be discussed as follow. As the first step, more realistic calibration could resolve visual distortion, normally presented in a reconstruction based on a simple pinhole camera model. Otherwise, straight lines in an acquired image (and subsequent reconstructed objects) may appear distorted, and even more so as they were further away from the lens center. Furthermore, missing data due to imperfect lighting, noise, and outliers in the point cloud, extracted from an infrared depth map were remedied, by statistical filter and by fusing it with that estimated by deep learning from the RGB image, captured from the same scene.

Because they were obtained from different modalities, to ensure correct correspondence, CEICP was proposed to register both point clouds, prior to fusion. The respective

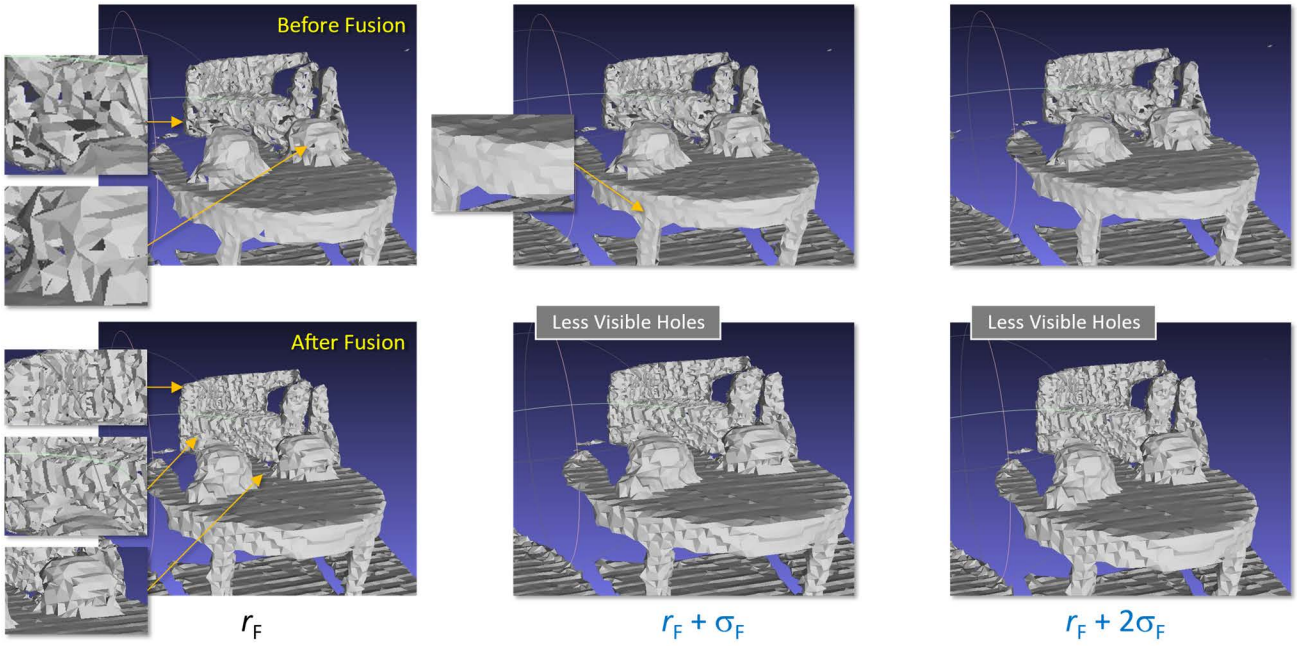

FIGURE 18. Comparison between 3D TINs reconstructed from the point cloud, before (top) and after (bottom) CEICP fusion. The ball radii were varied from r_F to $r_F + \sigma_D$, where r_F and σ_D are mean and standard deviation of distance to a nearest neighbor in the fused point cloud.

evaluations demonstrated that it outperformed state-of-the-arts. In particular, the proposed CEICP gave much accurate correspondence, and with faster convergence. This is due to that in this case, the information similarity was much pertinent than the distance one. Nonetheless, due to local minima of the entropy function, both datasets had to not much differ and be mostly aligned, to start with. Fortunately, this was the case in our setting, thanks to the camera configuration, and DL from the corresponding image. However, it is worth noted that there remained some missing data even after fusion due to coinciding voids in both datasets.

It has been well accepted that reconstruction from either of the modalities alone is insufficient. Another approach similar to ours was recently taken in [49]. They proposed an architecture, which combined monocular and stereo depth images, to refine the disparity map. The former was reconstructed by a modified VGG-16, as an autoencoder. The resultant bilinear unification was then refined by a minimum spanning tree (MST). Unlike our work, fusion was performed, based on semantic prior of the scene. Therefore, empirical weights for known object classes and their distance are required, which make it less generalizable. Alternatively, fusion directly between point clouds with different proposals, was also considered [50]. Firstly, the pointwise and voxel-wise features were extracted from a given point cloud. The corresponding voxel dense (VDS) and point spare (PSS) proposals, were merged. Secondly, semantic features were extracted from both VDS and PSS proposals and joined by proposal-aware fusion. Regression refinement was finally performed by proposal classification and regression. Although fusion was done directly on the point cloud, it was derived from the information available only in a single modality. Moreover, deep region of interest (ROI) fusion implied labeling of objects in the scene, which, unlike our work, makes it unsuitable for scenes with inseparable surfaces.

Lastly, provided the fused point cloud, the BPA was used to reconstruct the final surface of the scene. It is evident that after CEICP fusion, there were much less visible holes, when the ball radii were no less than $r_F + \sigma_D$, whilst those from a single cloud still exhibited strong evidence of missing data. It is also worth noted that, although a larger ball could eventually remove those defects, it may as well induce detail loss.

Albeit the experiments being carried out on public dataset of an indoor scene, for benchmarking purpose, they were without loss of generalization ability. That being said, analytical insights could very well benefit from that on much complex scenes, or with a moving camera, had their ground-truth measurements and peers' results been available.

V. CONCLUSION AND FUTURE WORKS

This paper presents a novel method for surface reconstruction of a scene by means of entropy-based cloud point fusion. Its main contributions were two folds. Firstly, a complementary depth map was synthesized from a 2D RGB image by a supervised CNN. The model exploited the visual cue learned from the training data, e.g., ground plane perception, vanishing point, and relative size of objects in the scene, etc. Accordingly, the point cloud, inferred from this map, although was less precise, smoothly distributed. Therefore, it was able to complete most voids, while trouncing outliers, present in the other. Secondly, to align these point clouds, rapidly and

accurately, the CEICP method has been proposed. Both numerical and visual assessments reported herein demonstrated that the proposed scheme outperformed other related works, based on a public benchmarking dataset.

It is anticipated that it can be as well applied to reconstruction of remotely sensed scenes, such as those by satellite imagery and aerial photography, and anatomical objects, acquired by medical tomographic imaging. Other prospective research directions worth considered are data fusion with and deep learning of other entities than point, such as voxel intensities and fiducial markers. In addition, treatments of their geometrical properties, e.g., feature preserving mesh filtering techniques, inhomogeneous distribution, and sparse data acquisition, etc., remain to be thoroughly investigated in future.

REFERENCES

[1] R. J. Wilson and S. Chiang, "Image processing techniques for obtaining registration information with scanning tunneling microscopy," *J. Vac. Sci. Technol. A, Vac., Surf., Films*, vol. 6, no. 2, pp. 398–400, Mar. 1988.

[2] S.-L. Lee, P. Horkaew, W. Caspersz, A. Darzi, and G.-Z. Yang, "Assessment of shape variation of the levator ani with optimal scan planning and statistical shape modeling," *J. Comput. Assist. Tomogr.*, vol. 29, no. 2, pp. 154–162, 2005.

[3] D. C. Le, J. Chansangrat, N. Keeratibharat, and P. Horkaew, "Symmetric reconstruction of functional liver segments and cross-individual correspondence of hepatectomy," *Diagnostics*, vol. 11, no. 5, p. 852, May 2021.

[4] I. Florinsky, *Digital Terrain Analysis in Soil Science and Geology*, 2nd ed. New York, NY, USA: Academic, 2016, pp. 31–41.

[5] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling Kinect sensor noise for improved 3D reconstruction and tracking," in *Proc. 2nd Int. Conf. 3D Imag., Modeling, Process., Vis. Transmiss.*, Oct. 2012, pp. 524–530.

[6] Y. Li, W. Li, S. Tang, W. Darwish, Y. Hu, and W. Chen, "Automatic indoor as-built building information models generation by using low-cost RGB-D sensors," *Sensors*, vol. 20, no. 1, p. 293, Jan. 2020.

[7] Q. Lu, M. Xiao, Y. Lu, X. Yuan, and Y. Yu, "Attention-based dense point cloud reconstruction from a single image," *IEEE Access*, vol. 7, pp. 137420–137431, 2019.

[8] P. Mandikal, K. L. Navaneet, M. Agarwal, and R. V. Babu, "3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image," in *Proc. Brit. Mach. Vis. Conf.*, Newcastle, U.K., Sep. 2018, p. 12.

[9] S. Choi, A. D. Nguyen, J. Kim, S. Ahn, and S. Lee, "Point cloud deformation for single image 3Dreconstruction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2379–2383.

[10] C. H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3D object reconstruction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 7114–7121.

[11] J. Zhang and X. Lin, "Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing," *Int. J. Image Data Fusion*, vol. 8, no. 1, pp. 1–31, Jan. 2017.

[12] P. Liang, Z. Fang, B. Huang, H. Zhou, X. Tang, and C. Zhong, "PointFusionNet: Point feature fusion network for 3D point clouds analysis," *Int. J. Speech Technol.*, vol. 51, no. 4, pp. 2063–2076, Apr. 2021.

[13] Z. Wang, Y. Xu, Q. He, Z. Fang, G. Xu, and J. Fu, "Grasping pose estimation for SCARA robot based on deep learning of point cloud," *Int. J. Adv. Manuf. Technol.*, vol. 108, no. 4, pp. 1217–1231, 2020.

[14] J. Ying and X. Zhao, "RGB-D fusion for point-cloud-based 3D human pose estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3108–3112.

[15] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3343–3352.

[16] G. Vosselman, "Fusion of laser scanning data, maps, and aerial photographs for building reconstruction," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jun. 2002, pp. 85–88.

[17] C. H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3D object reconstruction," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 7114–7121.

[18] Z. Ouyang, Y. Liu, C. Zhang, and J. Niu, "A cGANs-based scene reconstruction model using lidar point cloud," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl., IEEE Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, Dec. 2017, pp. 1107–1114.

[19] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3849–3856.

[20] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6243–6252.

[21] B. Li, Y. Wang, Y. Zhang, W. Zhao, J. Ruan, and P. Li, "GP-SLAM: Laser-based SLAM approach based on regionalized Gaussian process map reconstruction," *Auto. Robots*, vol. 44, no. 6, pp. 947–967, Jul. 2020.

[22] Q. Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 766–782.

[23] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[24] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[25] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understand.*, vol. 89, nos. 2–3, pp. 114–141, Feb. 2003.

[26] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. Eur. Conf. Comput. Vis.*, May 2004, pp. 558–569.

[27] X. Lu, S. Wu, H. Chen, S.-K. Yeung, W. Chen, and M. Zwicker, "GPF: GMM-inspired feature-preserving point set filtering," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2315–2326, Aug. 2017.

[28] O. Hirose, "A Bayesian formulation of coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2269–2286, Jul. 2020.

[29] C. Kim, H. Son, and C. Kim, "Fully automated registration of 3D data to a 3D CAD model for project progress monitoring," *Autom. Construct.*, vol. 35, pp. 587–594, Nov. 2013.

[30] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. Vis. Comput. Graphics*, vol. 5, no. 4, pp. 349–359, Oct. 1999.

[31] H. Seo, T. Kin, and T. Igarashi, "A mesh-aware ball-pivoting algorithm for generating the virtual arachnoid mater," in *Proc. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, vol. 11768, Oct. 2019, pp. 592–600.

[32] N. Wang, Y. Zhang, Z. Li, Y. Fu, H. Yu, W. Liu, X. Xue, and Y.-G. Jiang, "Pixel2Mesh: 3D mesh model generation via image guided deformation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3600–3613, Oct. 2021.

[33] X. Guo, J. Xiao, and Y. Wang, "A survey on algorithms of hole filling in 3D surface reconstruction," *Vis. Comput.*, vol. 34, no. 1, pp. 93–103, Jan. 2018.

[34] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. Int. Conf. 3D Vis.*, 2016, pp. 239–248.

[35] P. Horkaew and G. Z. Yang, *Construction of 3D Dynamic Statistical Deformable Models for Complex Topological Shapes* (Lecture Notes in Computer Science), vol. 3216. Berlin, Germany: Springer, 2004, pp. 217–224.

[36] W. Ma and Q. Li, "An improved ball pivot algorithm-based ground filtering mechanism for LiDAR data," *Remote Sens.*, vol. 11, no. 10, p. 1179, May 2019.

[37] S. Gai, F. Da, and X. Dai, "A novel dual-camera calibration method for 3D optical measurement," *Opt. Lasers Eng.*, vol. 104, pp. 126–134, May 2018.

[38] J. Digne and C. Franchis, "The bilateral filter for point clouds," *Image Process. Line*, vol. 7, pp. 278–287, Oct. 2017.

[39] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Nov. 2019, pp. 3827–3837.

[40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[41] H. Chen, X. Zhang, S. Du, Z. Wu, and N. Zheng, "A correntropy-based affine iterative closest point algorithm for robust point set registration," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 981–991, Jul. 2019.

[42] S. Kullback, *Information Theory and Statistics*. New York, NY, USA: Dover, 1968.
[43] L. Nadir. (Accessed: Apr. 20, 2022). *Ball-Pivoting Algorithm*. [Online]. Available: https://github.com/Lotemn102/Ball-Pivoting-Algorithm
[44] T. Zinsser, J. Schmidt, and H. Niemann, "A refined ICP algorithm for robust 3-D correspondence estimation," in *Proc. Int. Conf. Image Process.*, 2003, pp. 2–695.
[45] E. Trucco, A. Fusiello, and V. Roberto, "Robust motion and correspondence of noisy 3-D point sets with missing data," *Pattern Recognit. Lett.*, vol. 20, no. 9, pp. 889–898, Sep. 1999.
[46] T. Jost and H. Hugli, "A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images," in *Proc. 4th Int. Conf. 3-D Digit. Imag. Modeling*, Oct. 2003, pp. 427–433.
[47] J. M. Phillips, R. Liu, and C. Tomasi, "Outlier robust ICP for minimizing fractional RMSD," in *Proc. 6th Int. Conf. 3-D Digit. Imag. Modeling (DIM)*, Aug. 2007, pp. 427–434.
[48] H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1511–1516.
[49] M. P. Muresan, M. Raul, S. Nedevschi, and R. Danescu, "Stereo and mono depth estimation fusion for an improved and fault tolerant 3D reconstruction," in *Proc. IEEE 17th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Oct. 2021, pp. 233–240.
[50] Z. Cui and Z. Zhang, "PVF-NET: Point & voxel fusion 3D object detection framework for point cloud," in *Proc. 17th Conf. Comput. Robot Vis. (CRV)*, May 2020, pp. 125–133.

WATCHARAPHONG YOOKWAN received the B.Sc. degree in computer science and the M.Sc. degree in informatics from Burapha University, Chon Buri, Thailand, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree in computer engineering with the Suranaree University of Technology, Nakhon Ratchasima, Thailand. His research interests include image processing and digital geometry processing.

KRISANA CHINNASARN (Senior Member, IEEE) received the B.Sc. degree in statistics from Srinakharinwirot University, Mahasarakham Campus, Thailand, in 1992, the M.Sc. degree in computer science and information technology from the King Mongkut's Institute of Technology Ladkrabang, Thailand, in 1997, and the Ph.D. degree in computer science from Chulalongkorn University, Thailand, in 2004. From 1996 to 1997, he worked as a Research Assistant with the Computer Center, King Mongkut Institute of Technology Ladkrabang. From 2002 to 2003, he visited the Oxford University Computing Laboratory, Oxford, U.K., as a Ph.D. Visiting Student. Since 1997, he has been a Lecturer with the Faculty of Informatics, Burapha University, where he is a Former of the Department of Computer Science, Faculty of Science. He has been appointed as an Assistant Professor in computer science, since 2006. His research interests include machine learning and digital image processing and their applications to other science and engineering areas.

CHAKCHAI SO-IN (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in computer engineering from Kasetsart University, Bangkok, Thailand, in 1999 and 2001, respectively, and the M.S. and Ph.D. degrees in computer engineering from Washington University, St. Louis, MO, USA, in 2006 and 2010, respectively. He was an Intern at Cisco Networking Academy, CNAP-NTU, SG; Cisco Systems, Silicon Valley, USA; WiMAX Forums, USA; and Bell Labs, Alcatel-Lucent, USA. He is currently a Professor of computer science at the Department of Computer Science, Khon Kaen University. He has authored/coauthored over 100 international (technical) publications and ten books, including some in IEEE Journal on Selected Areas in Communications (J-SAC), IEEE Transactions on Cognitive Communications and Networking (TCCN), IEEE/CAA, IEEE Communications/*Wireless Communications Magazines*, IEEE System Journal (ISJ), COMNET, MONET, and ESWA, Mobile and Wireless Nets with IoT, Computer Network Laboratory, and Network Security Laboratory. His research interests include computer networking and the internet, wireless and mobile networking, the Internet of Things, wireless sensor networks, signal processing, cyber security, cyber-physical systems, and intelligent systems. He is a Senior Member of ACM. He has served as an Associate Editor for IEEE Access, *PLOS ONE*, Wireless Networks, *WCMC*, *PeerJ (CS)*, and *ECTI-CIT* and a Committee Member/Reviewer for many journals/conferences, such as IEEE, Elsevier, Springer, Wiley, IET, IEICE, and ETRI; GLOBECOM, ICC, VTC, WCNC, ICNP, ICNC, and PIMRC.

PARAMATE HORKAEW received the B.Eng. degree (Hons.) in telecommunication engineering from the King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Thailand, in 1999, and the Ph.D. degree in computer science from the Imperial College London, U.K., in 2004. He is currently an Assistant Professor at the School of Computer Engineering, Suranaree University of Technology, Thailand. His research interests include medical image analysis, computer aided diagnosis, computational anatomy, digital geometry processing, and computer vision.