sensors

Article

Indoor Scene Point Cloud Registration Algorithm Based on RGB-D Camera Calibration

Chi-Yi Tsai * and Chih-Hung Huang

Department of Electrical and Computer Engineering, TamKang University, 151 Yingzhuan Road,
Tamsui District, New Taipei City 251, Taiwan; kuku6118@gmail.com
* Correspondence: chiyi_tsai@mail.tku.edu.tw

Received: 13 July 2017; Accepted: 10 August 2017; Published: 15 August 2017

Abstract: With the increasing popularity of RGB-depth (RGB-D) sensor, research on the use of RGB-D sensors to reconstruct three-dimensional (3D) indoor scenes has gained more and more attention. In this paper, an automatic point cloud registration algorithm is proposed to efficiently handle the task of 3D indoor scene reconstruction using pan-tilt platforms on a fixed position. The proposed algorithm aims to align multiple point clouds using extrinsic parameters of the RGB-D camera obtained from every preset pan-tilt control point. A computationally efficient global registration method is proposed based on transformation matrices formed by the offline calibrated extrinsic parameters. Then, a local registration method, which is an optional operation in the proposed algorithm, is employed to refine the preliminary alignment result. Experimental results validate the quality and computational efficiency of the proposed point cloud alignment algorithm by comparing it with two state-of-the-art methods.

Keywords: point cloud registration; point cloud alignment; indoor scene reconstruction; multi-view calibration; RGB-D mapping

1. Introduction

Three-dimensional (3D) scene reconstruction is an important issue for several applications of robotic vision such as map construction [1], environment recognition [2], augmented reality [3,4], and simultaneous localization and mapping (SLAM) [5,6]. Furthermore, 3D scene reconstruction usually requires numerous types of sensors such as stereo cameras, RGB-depth cameras, time-of-flight (TOF) cameras, lasers, and LIDAR. In this paper, we discuss how to employ RGB-D camera data for 3D scene reconstruction applications. RGB-D cameras use the structured light technique [7] to re-project the dispersed 3D data point set. The 3D colored point cloud information is obtained by using the RGB-D feature of combining two-dimensional (2D) RGB and depth images. Each frame of the RGB-D point cloud information has an independent coordinate system, which can be defined as the camera coordinate system C_i at time i. When reconstructing a system, we expect to map each coordinate system C_i to the same world coordinate system W, which serves as the same mapping target for each coordinate system C_i. Therefore, in addition to defining a world coordinate system W, transformation matrices that transform each coordinate system C_i to the world coordinate W are required. In our method, extrinsic parameters obtained from camera calibration are used to derive the transformation matrices which are then used to achieve a coarse reconstruction. Finally, some of the existing fine registration methods are employed to conduct the optimal 3D scene reconstruction.

Until now, several multi-view camera calibration studies have been performed with the aim of obtaining the transformation matrices for the transformation between different views. Transformation matrices have been used in numerous applications, for example, in a previous report [8], the bundle adjustment method was proposed to be employed to calibrate multi-view stereo cameras. In another report [9], the use of landmarks in a given scene as calibration references was proposed and the existing feature-matching methods were employed in the study to conduct multi-view landmark matching. Multi-view camera calibration was achieved in this manner, based on which a novel camera sequence method was suggested for integrating the information in an entire system and promoting the robustness of future applications. Li et al. [10] recommended using a feature description-based method, instead of the traditional checkerboard methods for camera calibration, by employing a special reference diagram as the camera calibration reference point in the feature description matching. In this study, we use the most common checkerboard multi-view camera calibration method to obtain the transformation relationships for different rotation angles, and apply them for point cloud registration. Holz et al. [11] documented the basic implementation method of using a point cloud library (PCL) to conduct registration. The method primarily involved capturing and aligning the feature descriptors to conduct an initial registration of the point clouds, and using the classic iterative closest point (ICP) [12] method to achieve fine registration. For all the operations, relevant application programming interfaces were provided by the PCL.

Local registration methods have proven to be able to generate an accurate reconstruction model if multiple point clouds are close enough to each other. Recently, several local registration methods with a higher accuracy have been proposed. Some studies [13,14] provide an overview of the fine and coarse registration methods. According to Reference [14], there are four types of local registration methods: the ICP method, Chen's method [15,16], signed distance fields [17], and genetic algorithms [18]. Here, we chose the ICP method for our research because it is the most commonly used method in practice. The ICP method was proposed by Besl et al. [12] for the efficient registration of 3D shapes including free-form curves and surfaces. Subsequently, various extensions of the ICP method have emerged. Rusinkiewicz et al. [19] derived an ICP variant based on the uniform sampling of the space of normals which could provide comparatively good convergence results for relatively small scenes and sparse features. Low [20] suggested approximating the nonlinear least-squares optimization problem with a linear least-squares problem to solve the point-to-plane ICP algorithm. The author used an approximation method to increase the efficiency for determining the optimal solution. Serafin and Grisetti [21] formulated a new optimized cost function that included not only the point-to-point distance, but also surface normals or tangents; the new function was shown to not only increase the convergence speed, but also be advantageous in regard to the function robustness. For example, in a report [22] a system called KinectFusion was proposed; the system was based on a graphics processing unit (GPU)-accelerated ICP method and used a handheld moving Kinect to reconstruct indoor 3D scenes.

Feature descriptors are a technique that has been developed over a long time, evolving from the original 2D image feature descriptors to the current several 3D feature descriptors. Compared with 2D descriptors, 3D descriptors are richer in geometric information. Regardless of the computational efficiency decline owing to the increase in information, 3D descriptors can provide more accurate information regarding 3D objects or scenes. Rusu et al. [23,24] recommended the point feature histogram (PFH) descriptor, and further improved it to yield the fast point feature histogram (FPFH) [25] that was applied to point registration problems. FPFH significantly reduced the computational load of PFH by using some cache methods and modifying a few formulas; consequently, PFH was able to immediately achieve 3D point registration. Tombari et al. [26] recommended the signature of histograms of orientations (SHOT) descriptor, and reviewed some of the existing methods, classifying them as signatures and histograms methods. Then, the two types of methods were combined to generate a new 3D descriptor method, with the SHOT information lying between the signatures and histograms, and exhibiting both their features. Nascimento et al. [27] suggested a binary appearance and shape elements (BASE) descriptor that integrated both the intensity and shape information of a point cloud, and it was established in binary, thereby significantly enhancing the matching speed. Furthermore, the BASE descriptor could be applied to the scenes that were under poor lighting conditions and had sparse textures. Schmiedel et al. [28] proposed the interest-robust normal distribution transforms (NDT)-map (IRON) descriptor and applied it to robot localization; this descriptor was a feature descriptor used to reflect surface curvatures and object shapes, and it was established via NDT mapping.

On the other hand, a different way to reconstruct 3D indoor scenes using multiple RGB-D cameras can be interpreted as a fusion of depth maps obtained from different views, which is known as RGB-D image stitching [29,30]. In Reference [29], Song et al. proposed a rotated top-bottom dual-Kinect system to extend the field of view (FOV) of depth scanning based on the synchronized alignment of two RGB-D sensors. In other words, they employed two Kinects to maximize the FOV by stitching two depth images together to form a depth panorama; however, their system requires at least two RGB-D cameras to accomplish the RGB-D image stitching task. Recently, Li et al. [30] proposed an image-based RGB-D image stitching method that uses the registration data from color images to register depth maps. In other words, a calibration process is firstly conducted to find the relationship between the depth map and the color image, which is then used to align the depth map with the color image. By doing so, the problem of registering depth maps can be transformed into the problem of color image stitching, which usually requires significant computation and is difficult to run in real time.

From the above literature review, it can be seen that most of the modern point cloud alignment algorithms require a lot of computing power. With a greater amount of valid data in the point cloud, the required amount of computing will grow in a non-linear fashion. To improve the processing speed of point cloud alignment, this paper presents a novel calibration-based method which employs camera calibration techniques to find out the transformation matrices between some prefixed motor control points in the offline state, and then uses these transformation matrices directly in the online state. However, current camera calibration approaches usually induce calibration errors in the extrinsic transformation matrices, which may cause large errors in the registration result. To overcome this issue, an alignment calibration method is proposed to refine the calibrated transformation matrices. Unlike the FOV extension approaches in [29,30], the proposed method uses a single RGB-D sensor combined with a sequential panning-tilting capturing process to produce an accurate global registration result. Moreover, the computational load of the proposed method is very low because it only needs a simple coordinate transformation computation for each input point cloud data.

The following content describes the indoor scene reconstruction system that is proposed in this study. First, we introduce the related work in Section 2, including a simple description and classification of the current registration methods. Second, Section 3 systematically presents the processing steps of the proposed offline and online operations. Next, Section 4 provides the implementation details, including the offline mathematical models for system calibration and the coordinate system transformation methods for online operations. Last, we compare the details of several different point cloud alignment methods and present the results in Section 5. The conclusion is given in Section 6.

2. Related Work

Point registration has been a very actively researched topic. It is the process of performing a transformation to determine the mapping relationship between two point clouds in the 3D space. A transformation relationship can be expressed in terms of rotation and translation, using camera calibration models to describe how cameras project 3D scenes to 2D images. In the following expression for a camera calibration model

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\begin{bmatrix} \mathbf{R}_w^c & \mathbf{t}_w^c \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

we see that 2D-3D projection is determined by an extrinsic parameter matrix $\begin{bmatrix} \mathbf{R}_w^c & \mathbf{t}_w^c \end{bmatrix} \in \Re^{3\times 4}$ and intrinsic parameter matrix $\mathbf{K}$. Here, $\mathbf{R}_w^c$ is a rotation matrix and $\mathbf{t}_w^c$ is a translation vector from the world frame to the camera frame. Then, $\begin{bmatrix} u & v & 1 \end{bmatrix}^T$ is a homogenous coordinate of a 2D pixel on the image plane, whereas $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ is a homogenous coordinate of a 3D point in the world frame. s is a size ratio related to the depth of field, and it can be obtained from the depth images of a RGB-D camera.

Assume that there are two point data $\mathbf{p}_i \in \mathbf{P}$ and $\mathbf{q}_j \in \mathbf{Q}$, where $\mathbf{p}_i = [x_i, y_i, z_i]^T$ is the ith point of the point cloud $\mathbf{P}$ and $\mathbf{q}_j = [x_j, y_j, z_j]^T$ is the jth point of the point cloud $\mathbf{Q}$. If there is a mapping relationship between $\mathbf{p}_i$ and $\mathbf{q}_j$, the Euclidean distance can be expressed as $\|\mathbf{p}_i - \mathbf{q}_j\|$. To reduce the distance, we must define a cost function in terms of the rotation matrix $\mathbf{R}_p^q$ and translation vector $\mathbf{t}_p^q$ between two point clouds $\mathbf{P}$ and $\mathbf{Q}$. According to Reference [31], there are three different types of point cloud registrations—namely global, local, and local descriptors registration—to provide the 3D coordinate rotation matrix $\mathbf{R}_p^q$ and translation vector $\mathbf{t}_p^q$. In the following section, we describe the difference between these three types of registrations and introduce their representative methods.

2.1. Local Registration

In this method, the distance between two point clouds should be short enough. Given a distance that is sufficiently short, a more accurate and rigid transformation may be obtained with the local registration of point clouds. Recently, the most dominant method has been the ICP method which determines the adjacent points in the two close point clouds P and Q, and defines a cost function E_p^q as follows:

$$E_p^q(\mathbf{R}_p^q, \mathbf{t}_p^q)\Big|_{\mathbf{p}_k^l \in \mathbf{P}, \mathbf{q}_k^l \in \mathbf{Q}} = \sum_{k=1}^{N} \|(\mathbf{R}_p^q \mathbf{p}_k^l + \mathbf{t}_p^q) - \mathbf{q}_k^l\|^2 \tag{2}$$

where $\mathbf{p}_k^l$ and $\mathbf{q}_k^l$ are the point correspondences that are sufficiently close to each other. E_{local} is the sum of the errors of all the N point correspondences and k is the corresponding index. Then, an optimization method can be employed to determine the optimal E_{local}. The advantage of this process is that it can evaluate the corresponding data points that are sufficiently close enough to each other, and thereby achieve a more accurate registration; however, this method has a prerequisite that the initial point clouds should be close enough to each other.

To minimize the cost function (Equation (2)), the random sample consensus (RANSAC) method can be implemented to iteratively obtain the optimal rotation matrix $\mathbf{R}_p^q$ and translation vector $\mathbf{t}_p^q$. This process is advantageous in that it is able to determine the corresponding point clouds, which are sufficiently close to each other and generate a registration that is more accurate. In addition to the requirement of a short enough distance between the two initial point clouds, this process also involves significant computation. Segal et al. [32] proposed a new plane-to-plane ICP method incorporating the original and point-to-plane ICP methods, and demonstrated that the new method could provide better experimental results than the previous two methods.

2.2. Global Registration

Global registration can be initiated with two point clouds in any status, without requiring them to be close enough to each other, and is able to obtain a result similar to local registration. Several methods belong to the category of methods globally searching the point correspondences between two sets of point clouds or image features. Alternatively, a few methods employ some geometric approaches to determine the local point correspondences, i.e., the four-point congruent sets (4PCS) method [33]. When the point correspondences in the point clouds $\mathbf{P}$ and $\mathbf{Q}$ are obtained, the cost function can be defined as $E_p^q(\mathbf{R}_p^q, \mathbf{t}_p^q)\Big|_{\mathbf{p}_k^g \in \mathbf{P}, \mathbf{q}_k^g \in \mathbf{Q}}$, where $\mathbf{p}_k^g$ and $\mathbf{q}_k^g$ are the detected point correspondences. Next, the cost function E_p^q can be obtained using an optimization method. The advantage of this process is that the point cloud registration can be conducted at a random position, but this process usually has a higher degree of computation complexity as compared with local registration. In Reference [33], the 4PCS method is proposed for global registration. Figure 1 illustrates a schematic of the four-point algorithm in the 4PCS method. This method randomly three points in a point cloud, referred to as $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, and then uses a three-point plane to determine a fourth point $\mathbf{p}_4$. Subsequently, this method employs the cross-connected lines to obtain a cross point e, and calculates the ratio of the cross point distance $r_1 = \|\mathbf{p}_1 - \mathbf{e}\| / \|\mathbf{p}_1 - \mathbf{p}_2\|$ and $r_2 = \|\mathbf{p}_3 - \mathbf{e}\| / \|\mathbf{p}_3 - \mathbf{p}_4\|$. Given r_1 and r_2, all the points in the point cloud $\mathbf{Q}$ can now be connected, and the online candidates $\mathbf{e}_1 = \mathbf{q}_i + r_1(\mathbf{q}_j - \mathbf{q}_i)$ and $\mathbf{e}_2 = \mathbf{q}_i + r_2(\mathbf{q}_j - \mathbf{q}_i)$ can be calculated according to r_1 and r_2. When the candidates $\mathbf{e}_1$ and $\mathbf{e}_2$ overlap with each other, the four points that correspond to $\mathbf{P}$ can be evaluated. With the four points, it is possible to obtain the rotation matrix $\mathbf{R}_p^q$ and translation vector $\mathbf{t}_p^q$ in the 3D space. The advantage of global registration is that this method can conduct point cloud registration at a random position, but this method usually leads to larger registration errors as compared with local registration.

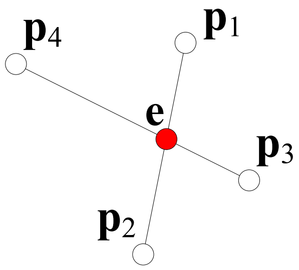

Figure 1. Schematic of the four-point algorithm in the four-point congruent sets (4PCS) method.

2.3. Local Descriptors Registration

Local descriptor registration requires a feature descriptors matching process to search for point correspondences in two point clouds, with the data of the two point clouds not required to be close to each other initially. By contrast, local and global registration methods search for point correspondences locally or globally under certain conditions without using feature descriptors, leading to an inefficient point correspondence determination process.

The feature descriptors usually contain information about point features in a local region such as surface, normal, signature, and texture. In this method, the feature descriptors may be extracted by different approaches to have the search capability for the point correspondences. By using feature descriptors to search for similar points that are the most close to one another in terms of the Euclidean distance, we can determine the point correspondences $\mathbf{p}_k^d \in \mathbf{P}$ and $\mathbf{q}_k^d \in \mathbf{Q}$ in two point clouds, with k as the corresponding index. Furthermore, we can use the point correspondences to define the cost function as $E_p^q(\mathbf{R}_p^q, \mathbf{t}_p^q)\Big|_{\mathbf{p}_k^d \in \mathbf{P}, \mathbf{q}_k^d \in \mathbf{Q}}$.

A relatively classic descriptor is the PFH descriptor which extracts the translation invariance feature from a key point. This feature descriptor is a 3D data that is related to the angles between the key point and an adjacent point. In a study [26], further developments have been made to this feature descriptor to obtain the FPFH descriptor that was used in point registration problems. It was shown that the computation could be significantly reduced by using some cache methods and modifying formulas, and, therefore, the 3D point registration could be conducted instantly. Zhou et al. [34] recommended a fast global registration method based on FPFH which allowed rapid registration convergence; compared with some of the existing methods, this method was a more real-time computation while having a certain registration accuracy.

3. System Architecture

In this section, we introduce the system architecture proposed in this study. Figure 2 illustrates the system architecture of the 3D scene reconstruction. First, the point cloud information for the scene reconstruction has to be obtained via device sensing. In an indoor scene, we can use a pan-tilt-driven RGB-D camera to record parts of the scene. It can be seen from Figure 3 that a camera driven by the pan-tilt records continuous frames F_1, F_2, and F_3. Here, $\mathbf{T}_1^2$ is the view transformation matrix to transform F_1 to F_2, whereas $\mathbf{T}_2^3$ is the view transformation matrix to transform F_2 to F_3. F_i contains the information of the RGB and depth images, which are transformed—with tools—to yield the color point cloud information $\mathbf{P}_i = \{\{\mathbf{p}_1, \mathbf{c}_1\}, \{\mathbf{p}_2, \mathbf{c}_2\}, ...\}$. Here, $\mathbf{p}_j = [x_j, y_j, z_j]^T$ is the position of point j in a point cloud (in the F_i coordinate) and $\mathbf{c}_j = [r_j, g_j, b_j]^T$ is the color of point j in the point cloud. Given that each F_i has its own coordinate system, it is necessary to use the transformation matrix $\mathbf{T}_i^0$ to combine those coordinates into one. Note that the proposed system does not include processes of feature matching, outlier removal, and motion estimation, because the required transformation matrices are obtained from the offline calibration. This design greatly reduces the processing time of 3D scene reconstruction and is also the main difference between the proposed method and the existing ICP-based methods.

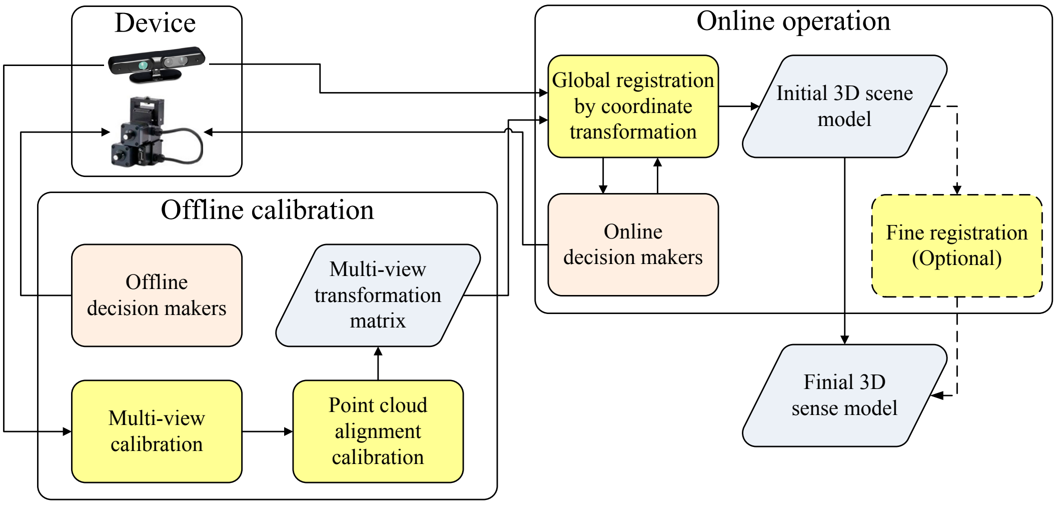

Figure 2. System architecture showing the 3D scene reconstruction method, consisting of three parts, namely device, offline calibration, and online operation.

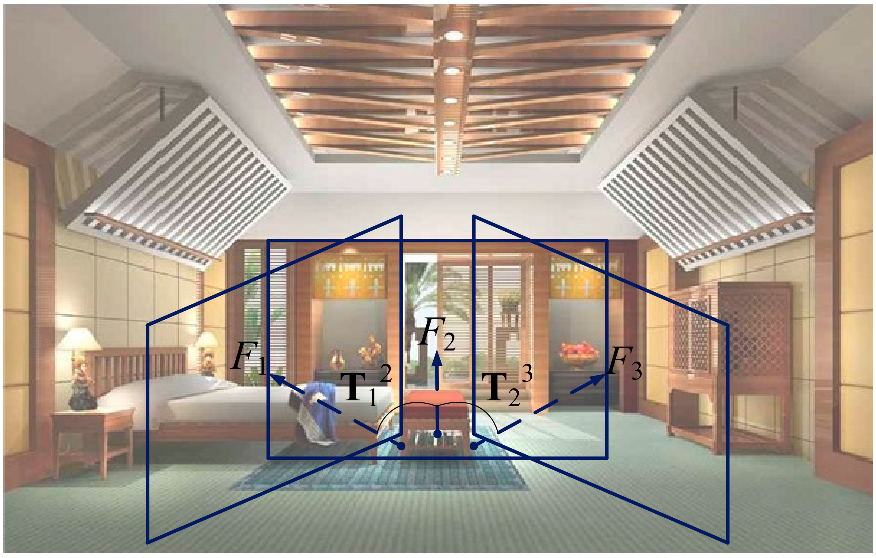

Figure 3. Schematic of the recording images obtained by rotating a device, with the blue rectangular areas representing the image planes. $\mathbf{T}_1^2$ and $\mathbf{T}_2^3$ are the two transformation matrices for the transformation between the two image planes.

3.1. Forward Kinematics Approach

Because the RGB-D camera is driven by a pan-tilt device, the transformation matrix can be directly obtained from a forward kinematics analysis of the motion platform [35]. Figure 4 illustrates the motion model of a pan-tilt device considered in the proposed system. Because the coordinate transformation of the camera frame is determined by connection links of the pan-tilt unit, it is mandatory that the transformation between each connection should be taken into consideration. Given that the pan, tilt, RGB-D camera, and camera frames are all on different coordinate axes, the corresponding forward kinematics Equation can be derived from the conventional Denavit-Hartenberg (D-H) link parameter method [36], which defines four D-H link parameters: the link length a_k, the link twist α_k, the link offset d_k, and the joint angle θ_k. Table 1 shows the D-H link parameters of the pan-tilt device used in the experiment (see Section 5). According to Table 1, the transformation matrix between $k-1$ and k joint frames can be computed by

$$\begin{aligned}
\mathbf{A}_k^{k-1} &= \mathbf{R}_z(\theta_k)\mathbf{T}_z(d_k)\mathbf{T}_x(a_k)\mathbf{R}_x(\alpha_k\Big) \\
&= \begin{bmatrix} c_{\theta_k} & -s_{\theta_k} & 0 & 0 \\ s_{\theta_k} & c_{\theta_k} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & a_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_k} & -s_{\alpha_k} & 0 \\ 0 & s_{\alpha_k} & c_{\alpha_k} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_k} & -s_{\theta_k}c_{\alpha_k} & s_{\theta_k}s_{\alpha_k} & a_k c_{\theta_k} \\ s_{\theta_k} & c_{\theta_k}c_{\alpha_k} & -c_{\theta_k}s_{\alpha_k} & a_k s_{\theta_k} \\ 0 & s_{\alpha_k} & c_{\alpha_k} & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3}$$

where $c_\theta \equiv \cos\theta$ and $s_\theta \equiv \sin\theta$. Based on the joint coordinate transformation in Equation (3), the forward kinematics Equation of the pan-tilt device can be obtained by chain multiplying four link transformation matrices such that

$$\mathbf{T}_i^0 = \prod_{k=1}^{4} \mathbf{A}_k^{k-1} = \begin{bmatrix} s_{\theta_p} & -c_{\theta_p}c_{\theta_t+90} & c_{\theta_p}s_{\theta_t+90} & a_3 s_{\theta_p} + a_2 c_{\theta_p}c_{\theta_t+90} \\ -c_{\theta_p} & -s_{\theta_p}c_{\theta_t+90} & s_{\theta_p}s_{\theta_t+90} & a_2 s_{\theta_p}c_{\theta_t+90} - a_3 c_{\theta_p} \\ 0 & -s_{\theta_t+90} & -c_{\theta_t+90} & d_1 + a_2 s_{\theta_t+90} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where θ_p and θ_t denote the pan angle and the tilt angle of the device, respectively. Therefore, the transformation matrix in Equation (4) can be used to align the point cloud at the frame F_i to F_0 directly.

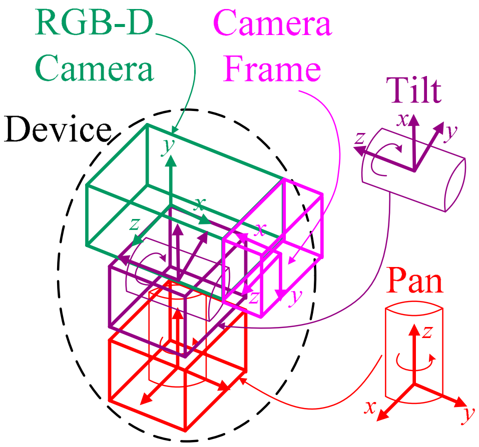

Figure 4. Motion model of the pan-tilt device used in the proposed 3D scene reconstruction system.

Table 1. D-H link parameters of the pan-tilt device used in the experiment.

k	α_k (deg)	a_k (mm)	d_k (mm)	θ_k (deg)
1	90	0	45	θ_p
2	90	74.466	0	$\theta_t + 90$
3	0	23	0	−90
4	0	0	0	180

3.2. Camera Calibration Approach

Another way to obtain the transformation matrix information of each F_i coordinate frame pertains to the offline calibration of the device. In this approach, the motor must be rotated to generate different camera views, and the rotation control is achieved via offline decision-makers. The following are the steps for the offline calibration operations:

1. Start to capture the initial frame F_0, and define its coordinate as a world coordinate W.
2. Conduct multi-view calibration to obtain the extrinsic parameter matrix for the initial world coordinate.
3. Control the motor and allow it to rotate to the next point.
4. Capture the frame F_i, and conduct multi-view calibration to obtain the extrinsic parameter matrix.
5. Generate point cloud $\mathbf{P}_i$, and conduct point cloud alignment calibration to obtain the transformation matrix.
6. If completed, store all the transformation matrices; otherwise, go back to Step 3 to make a decision.

Therefore, prior to a scene reconstruction, we require conducting offline calibration to obtain multi-view transformation matrices. The implementation of offline calibration will be discussed in detail in Section 3.

With respect to online operation, we first develop a model for the 3D scene reconstruction. Because this system consists of a motor-controlled and point cloud-registered part, the online decision-makers are entirely responsible for the allocation and processing. Subsequently, 3D scene reconstruction is conducted according to the following steps:

1. Start to capture the initial frame F_0 and generate point cloud $\mathbf{P}_0$, and define the coordinate of the point cloud $\mathbf{P}_0$ as the world coordinate W.
2. Control the motor and allow it to rotate to the next point, wait until the motor rotates to the fixed point, and then haul back the motor to complete the command.
3. Capture the frame F_i and generate a point cloud $\mathbf{P}_i$.
4. Transform the point cloud $\mathbf{P}_i$ to $\mathbf{P}_i{}^0$ using the corresponding transformation matrix $\mathbf{T}_i{}^0$ estimated in the offline calibration.
5. If completed, generate the initial 3D scene reconstruction model; otherwise, go back to Step 3 to make a decision of aligning the next point cloud.
6. If the fine registration step is enabled, then use one of the existing ICP methods to refine the initial 3D scene reconstruction model; otherwise, use the initial 3D scene reconstruction model as the final result.
7. Generate the final 3D scene reconstruction model.

When the 3D scene reconstruction is complete, the outlier points can be removed using some of the existing outlier removal methods [37,38]. The detailed operations of the point cloud registration will be presented in Section 4.

4. The Proposed Method

This study primarily proposes a 3D scene construction method that integrates motor control and camera calibration. The camera calibration is conducted via a camera to first detect and identify a key point in the checkerboard. With offline calibration, a relative transformation matrix for the camera in a fixed position can be obtained and used for global registration. In the following content, we present the offline calibration and global registration methods proposed by us, and also introduce how to integrate them with local registration.

4.1. Offline Calibration

This subsection primarily introduces the proposed offline calibration method of generating multi-view transformation matrices based on multi-view camera calibration. Suppose that the intrinsic parameter matrix $\mathbf{K}$ can be obtained by fixing the extrinsic parameter matrix as $\begin{bmatrix} \mathbf{I} & 0_{3\times1} \end{bmatrix}$ (do not move the view) and using camera calibration [10]. Our method aims to obtain the extrinsic parameter matrix between the initial frame F_0 and the ith frame F_i. Figure 5 illustrates the different images in which the object (checkerboard) exists in each image captured from a different view. In this approach, the intrinsic parameters are fixed, whereas the extrinsic parameter matrices are obtained via camera calibration. Based on the extrinsic parameters, the following formula is obtained:

$$\begin{bmatrix} \mathbf{P}_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_r^i & \mathbf{t}_r^i \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_r \\ 1 \end{bmatrix} = \mathbf{T}_r^i \begin{bmatrix} \mathbf{P}_r \\ 1 \end{bmatrix} \tag{5}$$

where $\mathbf{P}_r$ is the coordinate of the key point on the checkerboard that is used as the coordinate system. $\mathbf{T}_r^i$ is the transformation matrix to map $\mathbf{P}_r$ to $\mathbf{P}_i$. With camera calibration, we can obtain the transformation matrices at different views. Next, we want to obtain the relative transformation matrix between two views. Provided that F_0 is defined as the world coordinate W, we can assume that all $\mathbf{P}_i$ should be transformed to F_0, and we thereby obtain the following formula:

$$(\mathbf{R}_r^i)^T \mathbf{P}_i - (\mathbf{R}_r^i)^T \mathbf{t}_r^i = (\mathbf{R}_r^0)^T \mathbf{P}_0 - (\mathbf{R}_r^0)^T \mathbf{t}_r^0 \tag{6}$$

Then we expand and reorganize Equation (6) to form a complete transformation formula of $\mathbf{P}_i$ to F_0 such that

$$\mathbf{R}_i^0 \mathbf{P}_i + \mathbf{t}_i^0 = \mathbf{R}_r^0 (\mathbf{R}_r^i)^T \mathbf{P}_i - \mathbf{R}_r^0 (\mathbf{R}_r^i)^T \mathbf{t}_r^i + \mathbf{t}_r^0 \tag{7}$$

where we define the rotation matrix $\mathbf{R}_i^0 = \mathbf{R}_r^0 (\mathbf{R}_r^i)^T$ for the transformation of $\mathbf{P}_i$ to the F_0 coordinate system and the translation vector $\mathbf{t}_i^0 = -\mathbf{R}_r^0 (\mathbf{R}_r^i)^T \mathbf{t}_r^i + \mathbf{t}_r^0$. Finally, we calculate a transformation matrix of each view according to Equation (7), and thereby obtain all the transformation matrices as follows:

$$\mathbf{T}_i^0 = \begin{bmatrix} \mathbf{R}_i^0 & \mathbf{t}_i^0 \\ 0_{1\times3} & 1 \end{bmatrix} \tag{8}$$

These are the initial transformation matrices that are obtained by applying linear methods to the 2D point correspondences. However, these transformation matrices encounter calibration errors, which may deteriorate the registration result. To deal with this issue, we include fine registration in the offline calibration, a step that needs the identifiable geometric features of a scene. We can place some simple geometric objects in the scene as shown in Figure 6, in which each scene contains some geometric objects in addition to the board. The next step is to conduct 3D point alignment calibration, and this step requires local registration for conducting calibration. Here, we choose to use ICP. Because we can choose the way that the objects are placed in a scene to obtain good results, we do not need to select a complex method as long as the geometric objects are separated enough from one another and are representative. This part must be conducted using two adjacent point clouds. Suppose that point cloud $\mathbf{P}_i^0$ which consists of each point $\mathbf{P}_i$ and $\mathbf{P}_0$ can be obtained after the transformation matrix in Equation (8), and, therefore, two adjacent point clouds can be denoted as $\mathbf{P}_{i-1}^{0}$ and $\mathbf{P}_{i}^{0}$. Next, with alignment calibration, we can obtain a refinement transformation between $\mathbf{P}_{i-1}^{0}$ and $\mathbf{P}_{i}^{0}$ as follows:

$$\mathbf{P}_{i-1}^{0} = \mathbf{R}_{0}^{(i)}\mathbf{P}_{i}^{0} + \mathbf{t}_{0}^{(i)} \tag{9}$$

where $\mathbf{R}_{0}^{(i)}$ and $\mathbf{t}_{0}^{(i)}$ are the ith refinement rotation matrix aligning $\mathbf{P}_{i}^{0}$ to $\mathbf{P}_{i-1}^{0}$ in the F_0 coordinate system and the ith refinement translation vector, respectively. Given that $\mathbf{P}_{i}^{0}$ and $\mathbf{P}_{i-1}^{0}$ are all in the F_0 coordinate system, we can substitute Equation (7) into Equation (9) to form a new transformation relationship as follows:

$$\mathbf{P}_{i-1}^{0} = \mathbf{R}_{0}^{(i)}\mathbf{R}_{i}^{0}\mathbf{P}_{i} + \mathbf{R}_{0}^{(i)}\mathbf{t}_{i}^{0} + \mathbf{t}_{0}^{(i)} \tag{10}$$

According to Equation (10), we can obtain the aligned transformation matrix against the calibration errors such that

$$\hat{\mathbf{T}}_{i}^{0} = \begin{bmatrix} \mathbf{R}_{0}^{(i)}\mathbf{R}_{i}^{0} & \mathbf{R}_{0}^{(i)}\mathbf{t}_{i}^{0} + \mathbf{t}_{0}^{(i)} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{11}$$

Finally, the information of these transformation matrices is stored for use in the online operation.

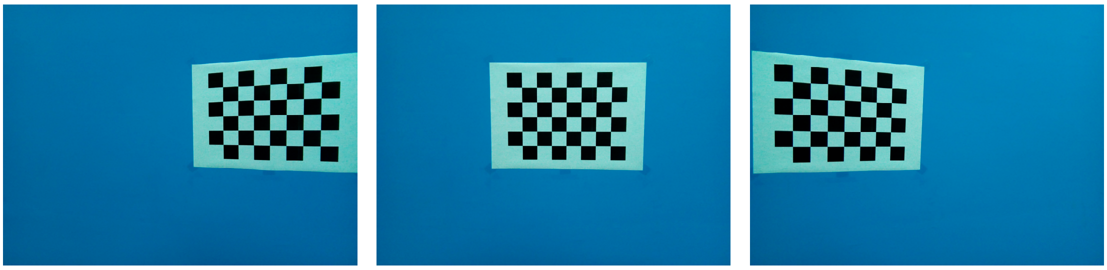

Figure 5. Illustration of recording images of a fixed reference diagram from different views. The images from left to right indicate the camera views that are rotated from left to right.

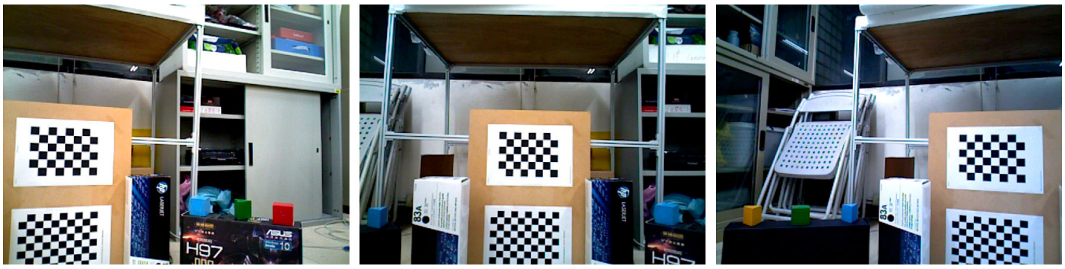

Figure 6. Illustration of the scene of point alignment calibration that is used in offline calibration. These scenes contain the same geometric objects.

4.2. Online Operation

This subsection primarily introduces how to use the multi-view transformation matrices to achieve initial and fine registration. Given that we have stored the transformation matrices in offline calibration, it will be easy for us to rapidly achieve initial registration. In Section 3, we introduced the initial world coordinate W and defined it as the coordinate of F_0, while all the transformation matrices that we have obtained map to F_0. First, we take F_1 as an example. Provided that we want to let point cloud $\mathbf{P}_1$ of F_1 be transformed to F_0, we can use the following formula:

$$\mathbf{P}_{1}^{0} = \hat{\mathbf{T}}_{1}^{0}\mathbf{P}_{1} \tag{12}$$

Here, $\mathbf{P}_1^0$ is the transformed point cloud $\mathbf{P}_1$ in the F_0 coordinate system. When the point clouds at all the views are processed by Equation (12), we can obtain an initial reconstructed 3D scene model $\mathbf{P}_w = \{\mathbf{P}_0, \mathbf{P}_1^0, \mathbf{P}_2^0, \ldots, \mathbf{P}_N^0\}$, where N represents the number of multiple views.

Furthermore, an optional fine registration can be used to reduce the convergence error between the cloud point of $\mathbf{P}_0$ and that of $\mathbf{P}_1$. In other words, we conduct fine registration by using some existing local registration methods, such as ICP. After refining the transformation matrix, the new point cloud will be generated as shown below,

$$\hat{\mathbf{P}}_1^0 = \widetilde{\mathbf{T}}_0^{(1)} \mathbf{P}_1^0 \tag{13}$$

where $\hat{\mathbf{P}}_1^0$ is the point cloud $\mathbf{P}_1$ transformed to the F_0 coordinate after local registration. $\widetilde{\mathbf{T}}_0^{(1)}$ is a refined transformation matrix that is generated with the fine registration of point cloud $\mathbf{P}_1^0$ which has already been subjected to the initial registration. If the point clouds at all the views are treated according to Equations (12) and (13) and then combined into the same coordinate of F_0, we can finally obtain a better reconstructed 3D scene model $\hat{\mathbf{P}}_w = \{\mathbf{P}_0, \hat{\mathbf{P}}_1^0, \hat{\mathbf{P}}_2^0, \ldots, \hat{\mathbf{P}}_N^0\}$.

5. Experimental Results

Because the proposed algorithm is based on the combination of pan-tilt camera control and coordinate transformation to perform point cloud alignment, we did not use the existing public database in the experiment due to the requirement of offline calibration. A RGB-D camera and a pan-tilt platform were used in the experiment. We used the ASUS Xtion Pro RGB-D camera (ASUSTek Computer Inc., Taipei, Taiwan) [39] and mounted the RGB-D camera on the FLIR E46-17 pan-tilt platform (FLIR Motion Control System Inc., Goleta, CA, USA) [40], as shown in Figure 7. Table 2 shows the specifications of the FLIR E46-17 pan-tilt platform, which was used to drive the RGB-D camera for capturing indoor scene point clouds. Two state-of-the-art methods were employed in the experiment to compare with the proposed method. The first one is the fast global registration (FGR) method proposed in Reference [34], and the second one is the Super4PCS method proposed in Reference [31]. These two methods do not require offline calibration and usually can provide robust and accurate registration results. Moreover, we also employed the ICP algorithm as the refinement process for all methods used in the experiment. All experiments were carried out on an Ubuntu 14.04 notebook computer equipped with Intel Core i7-3530M CPU and 8 GB memory.

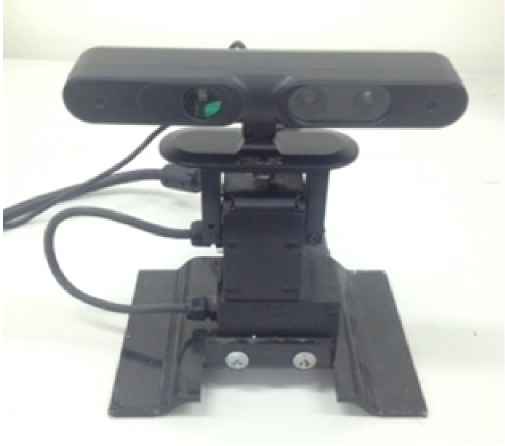

Figure 7. The physical pan-tilt device used in the experiments, which is an ASUS Xtion Pro RGB-D camera [39] mounted on an E46-17 pan-tilt platform [40].

Table 2. Specifications of the E46-17 pan-tilt platform [40].

Rated Payload	Maximum Speed	Position Resolution	Tilt Range	Pan Range
2.72 Kg	300°/s	0.013°	−47° to +31°	−159° to +159°

5.1. Point Cloud Registration Results

Figure 8 presents the comparison results between the proposed method (Equation (11)) and the forward kinematics method (Equation (4)). Visually observing Figure 8 finds that both methods provide similar point cloud registration results. Therefore, the correctness of the forward kinematics Equation (4) is verified. Figure 9 shows the experimental results of point cloud registration obtained from the proposed and the state-of-the-art methods. In this experiment, the minimum distance between the scene and the camera is about 1 m. In Figure 8, the pictures listed in the left column are two original point clouds captured from different viewers in the same scene. We can see from Figure 9 that all methods succeed in aligning the two point clouds. The pictures listed in the second column are the results of the proposed method, which illustrate that an initial registration result (the upper second column) can be obtained efficiently. The third column of the pictures presents the registration results of the Super4PCS method. Only using Super4PCS also can obtain an initial alignment result (the upper third column), which can be further coupled with the ICP method to make a refinement result (the lower third column). The right column of the pictures presents the registration results of the FGR method. The initial alignment result of the FGR method is shown in the upper right column, and it also can be improved by the ICP method to make the registration result better (the lower right column). Figures 10–12 show the experimental results of three different rooms obtained from the proposed and the compared methods. One can see that the proposed method still provides a competitive point cloud registration performance in comparison to the Super4PCS and FGR methods. Figure 13 shows the registration results of the Room 1, Room 2, and Room 3 datasets obtained from the proposed method; each dataset contains 15 point clouds captured in different camera views that need to be aligned. One can see that the initial registration results of the proposed method are very close to the ICP refined ones. Therefore, the above experimental results validate the efficiency and robustness of the proposed method.

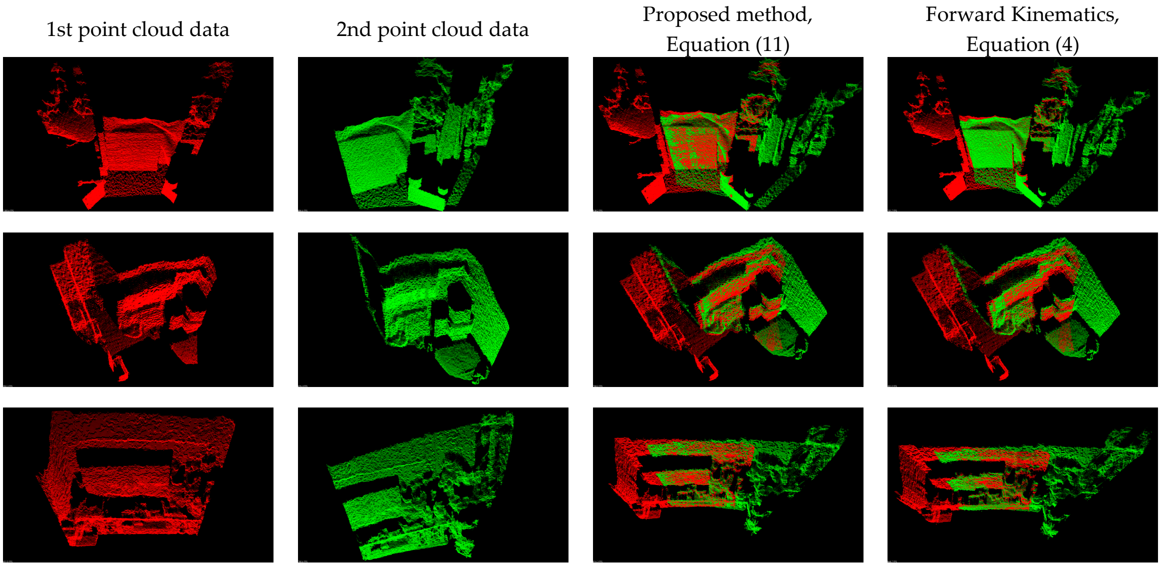
1st point cloud data
2nd point cloud data
Proposed method, Equation (11)
Forward Kinematics, Equation (4)

Figure 8. Comparison results between the proposed method and the forward kinematics method.

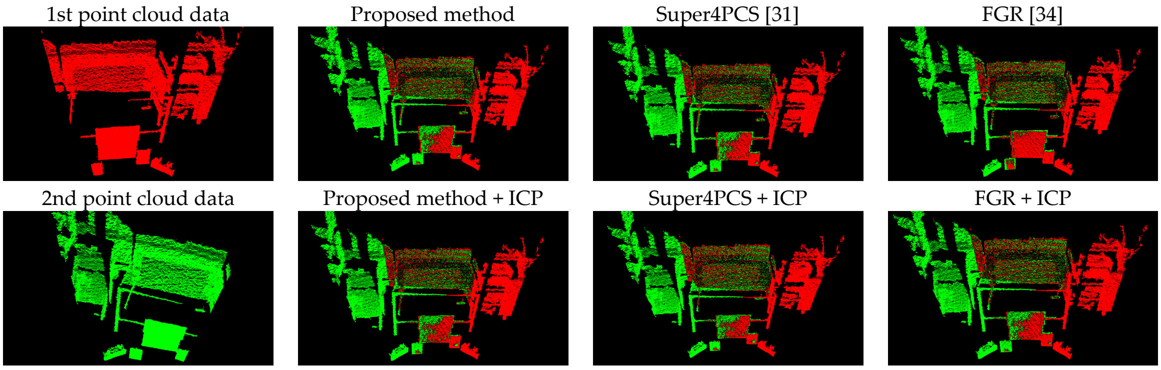

Figure 9. Experimental results of the 1 m dataset (a minimum of 1 m distance between the scene and the camera). The left column is the original point cloud data, the upper second column is the experimental result of the proposed method, and the lower second column is the result of the proposed method + ICP. The upper third column is the experimental result of Super4PCS, and the lower third column is the result of Super4PCS + ICP. The upper fourth column is the experimental result of FGR, and the lower fourth column is the result of FGR + ICP.

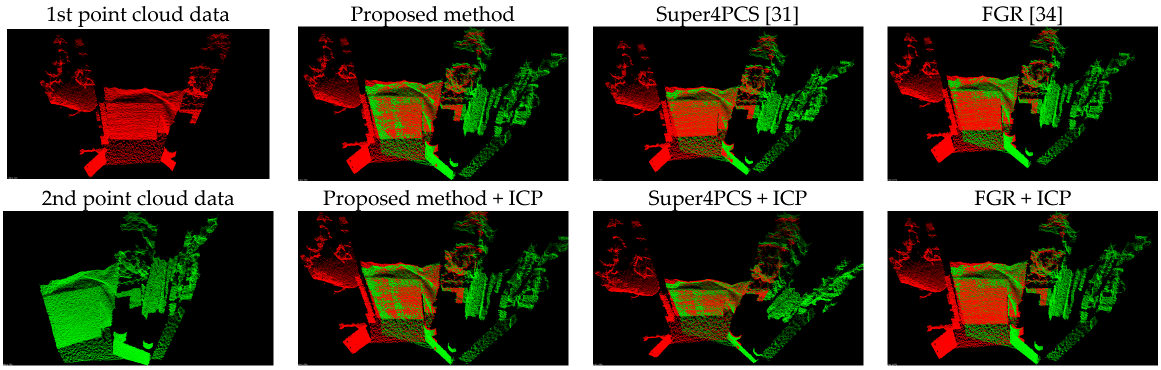

Figure 10. Experimental results of the Room 1 dataset.

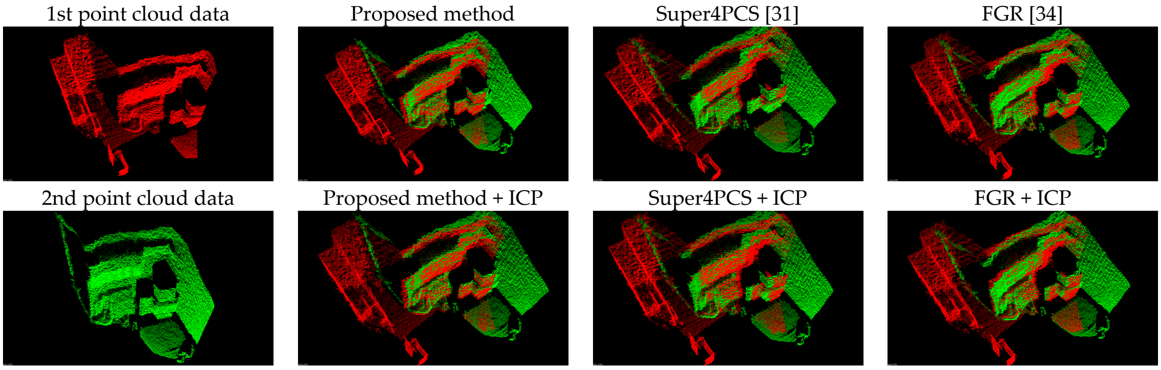

Figure 11. Experimental results of the Room 2 dataset.

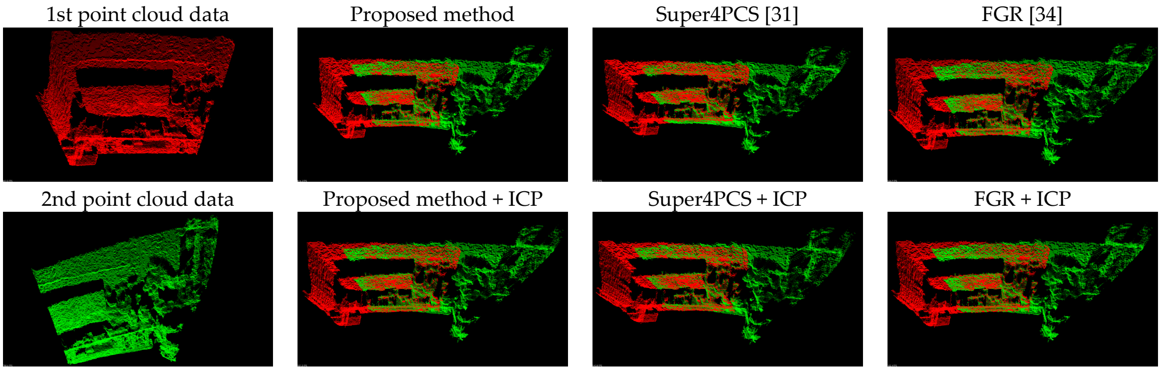

Figure 12. Experimental results of the Room 3 dataset.

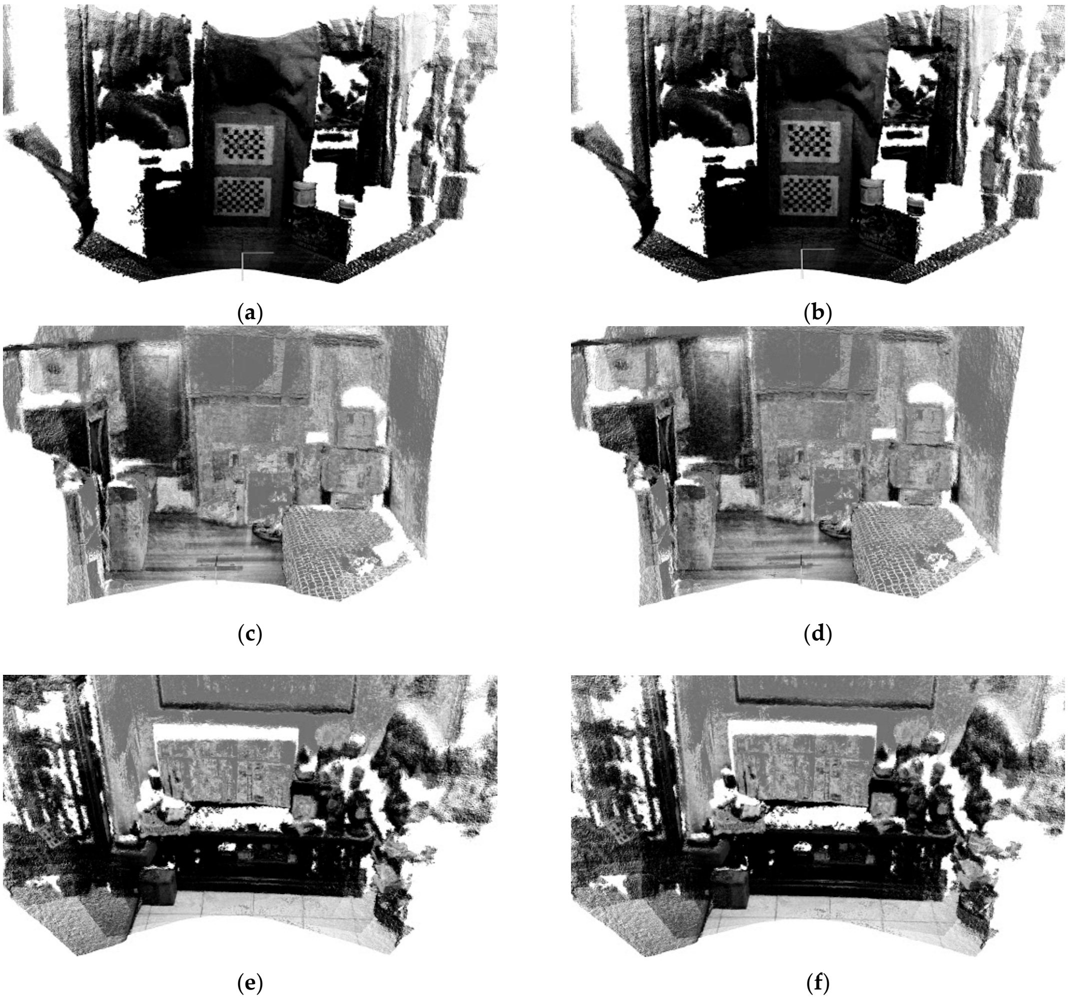

Figure 13. Registration results of the proposed method. (**a**) The initial registration result and (**b**) the refined registration result of the Room 1 dataset. (**c**) The initial registration result and (**d**) the refined registration result of the Room 2 dataset. (**e**) The initial registration result and (**f**) the refined registration result of the Room 3 dataset.

5.2. Quantitative Evaluation

To quantitatively evaluate the registration performance of the proposed and the compared methods, we employed the root mean square (RMS) metric [41] in the experiment as follows:

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} \mathbf{d}_{\min}^{2}(i)}{N}} \tag{14}$$

where $\mathbf{d}_{\min} \in \Re^{N}$ is a vector containing the minimum distance (in millimeters) of N closest points between two aligned data scenes, and $\mathbf{d}_{\min}(i)$ denotes the ith element of the vector $\mathbf{d}_{\min}$. To evaluate the registration performance of the proposed aligned transformation matrix $\hat{\mathbf{T}}_{i}^{0}$, Table 3 records three online registration comparison results between the Equations (4), (8) and (11) solutions. It is clear that the solutions of Equations (4) and (8) produce registration results with higher RMS errors due to modeling uncertainties and camera calibration errors, respectively. In contrast, the solution of Equation (11) produces better registration results against the calibration errors. Therefore, the registration performance of the proposed formula in Equation (11) is validated.

Table 3. Average RMS comparison results between forward kinematics, camera calibration, and the proposed methods.

Average RMS	**Online Registration Methods**		
Test Dataset	**Forward Kinematics Equation (4)**	**Camera Calibration Equation (8)**	**Proposed Method Equation (11)**
Room 1	18.7566	21.1770	18.4981
Room 2	26.6134	30.5964	26.3199
Room 3	19.5648	25.5299	18.3148

To investigate the effect of the minimum distance between the scene and the camera to the registration performance, we considered three cases, 1 m, 3 m and 5 m, in the experiment. Table 4 shows the average RMS errors of the registration results in these three cases. One can observe that the proposed method provides the best initial registration result for all three cases, followed by the FGR method and the Super4PCS method. After the fine registration process, the FGR method yields the best result, which is also very close to the refinement result of the proposed method. Furthermore, when the minimum distance between the scene and the camera increases from 1 m to 3 m, the RMS error of the proposed method just slightly increases about 0.2374 in average. In contrast, the average RMS errors of the two compared methods increase about 8.0790 and 0.4554 for the Super4PCS and the FGR methods, respectively. This implies that the proposed method is more robust to the variation of the distance between the scene and the camera when compared to the two state-of-the-art methods. However, when the minimum distance increases to 5 m, the average RMS errors of all methods increase rapidly above about 31. This is caused by the sensor limitation of the ASUS Xtion Pro RGB-D camera, whose effective sensing distance ranges from 0.8 m to 3.5 m. Therefore, in the case of the 5 m minimum distance, the captured point cloud has a large number of invalid data points, which significantly deteriorates the registration performance of the proposed and the compared methods.

Table 4 also records the RMS results of the three registration results shown in Figures 10–12. It is clear from Table 4 that the proposed method produces the best initial registration results compared to the FGR method and Super4PCS method. On the other hand, the ICP refinement process only makes a minor improvement to the proposed method. Because the proposed method can provide accurate initial registration results, the ICP refinement process is an ignorable step in the proposed multi-view point cloud registration system.

Table 4. Average RMS comparison results between the proposed and the state-of-the-art methods.

Test Dataset	Proposed Method	Super4PCS	FGR	Proposed Method + ICP	Super4PCS + ICP	FGR + ICP
1 m dataset	19.6716	22.1680	19.7152	19.5366	20.3303	19.2841
3 m dataset	19.9090	30.2470	20.1706	19.6401	29.2774	19.0025
5 m dataset	32.0353	43.1703	33.9440	31.5003	40.6470	31.2734
Room 1	18.4981	27.0479	18.8040	18.3856	24.4929	18.4195
Room 2	26.3199	29.0571	29.5579	25.7527	25.6207	25.7157
Room 3	18.3148	20.9321	18.7524	18.2912	18.3036	18.2979

5.3. Computational Efficiency

One major advantage of the proposed method is its high computational efficiency. Table 5 shows the average processing time of all the methods in the experiment. Observing Table 5 finds that the processing time of the proposed method is only about 4 ms regardless of the effect of distance and environment variations, which is much faster than the Super4PCS and the FGR methods to produce the initial registration result. However, the ICP refinement greatly increases the processing time of the proposed method. Therefore, without the ICP refinement process, the proposed method is able to accurately align multi-view point clouds in real time. Note that the average processing time of the proposed alignment calibration is about 52,914.98 ms. However, this step is only performed once in the offline calibration process.

Table 5. Average processing time (in milliseconds).

Test Dataset	Proposed Method	Super4PCS	FGR	Proposed Method + ICP	Super4PCS + ICP	FGR + ICP
1 m dataset	4.11	553,958.70	12,077.14	27,378.52	599,331.00	70,038.29
3 m dataset	3.14	528,159.80	30,508.29	17,385.10	574,146.00	119,717.10
5 m dataset	4.18	100,250.80	48,501.62	32,572.52	237,221.20	65,188.71
Room 1	3.77	636,178.79	10,902.57	40,000.84	732,519.07	60,492.71
Room 2	4.24	135,305.92	57,163.71	52,331.24	261,104.14	160,310.5
Room 3	3.90	582,703.07	25,278.57	46,932.69	671,707.43	53,883.57

6. Conclusions

In this paper, an efficient point cloud registration method is proposed to align multi-view point clouds based on a camera calibration technique. One advantage of the proposed method is that it converges quickly and only needs to perform point cloud transformations without any iterative process. The proposed method consists of an offline calibration and an online operation. In the offline calibration, the relationship between different fixed camera views is calibrated to form several specific coordinate transformation matrices. Moreover, an alignment calibration process is proposed to obtain more accurate transformation matrices against the calibration errors. In the online operation, the point clouds captured from these fixed camera views can be directly and accurately aligned through the corresponding transformation matrices. Experimental results show that the proposed method is able to produce accurate point cloud registration results, even if no fine registration process is performed. Furthermore, the proposed method can process in real time. The average processing time of the proposed method is only about 4 ms running on a commercial notebook computer to align two point clouds, which is much faster than the existing ICP-based point cloud registration approaches. These advantages can make the registration of multi-view point clouds possibly become an online process with high registration accuracy via a sequential panning-tilting capturing process of a single RGB-D camera.

In the future, the proposed method will be further extended to combine with different control platforms to facilitate a variety of 3D reconstruction applications, such as 3D dense-map building, object model reconstruction, etc.

Acknowledgments: This work was supported by the Ministry of Science and Technology of Taiwan, ROC, under grant MOST 106-2221-E-032-006 and was supported in part by NUWA Robotics Co., Ltd., Nanjing East Road, Zhongshan District, Taipei City 104, Taiwan.

Author Contributions: Chi-Yi Tsai proposed the idea and supervised the research; Chih-Hung Huang designed and implemented the algorithm; Chih-Hung Huang performed the experiments and collected the experimental data; Chi-Yi Tsai analyzed the data and wrote/revised the manuscript.

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

References

1. Souza, A.A.S.; Maia, R.; Goncalves, L.M.G. 3D Probabilistic Occupancy Grid to Robotic Mapping with Stereo Vision. In *Current Advancements in Stereo*; Bhatti, A., Ed.; InTech Open Access: Rijeka, Croatia, 2012; ISBN 978-953-51-0660-9.
2. Saxena, A.; Chung, S.H.; Ng, A.Y. 3-D Depth Reconstruction from a Single Still Image. *Int. J. Comput. Vis.* **2008**, *76*, 53–69. [CrossRef]
3. Kahn, A. Reducing the Gap between Augmented Reality and 3D Modeling with Real-Time Depth Imaging. *Int. J. Comput. Vis.* **2013**, *17*, 111–123. [CrossRef]
4. Yang, S.; Maturana, D.; Scherer, S. Real-time 3D Scene Layout from a Single Image Using Convolutional Neural Networks. In Proceedings of the Robotics and Automation of the Conference, Stockholm, Sweden, 16–21 May 2016; pp. 2183–2189.
5. Choi, C.; Trevor, A.J.B.; Christensen, H.I. RGB-D Edge Detection and Edge-based Registration. In Proceedings of the Intelligent Robots and Systems of the Conference, Tokyo, Japan, 3–7 November 2013; pp. 1568–1575.
6. Liu, F.; Lv, Q.; Lin, H.; Zhang, Y.; Qi, K. An Image Registration Algorithm Based on FREAK-FAST for Visual SLAM. In Proceedings of the Chinese Control of the Conference, Chengdu, China, 27–29 July 2016.
7. Zanuttigh, P.; Marin, G.; Mutto, C.D.; Dominio, F.; Minto, L.; Cortelazzo, G.M. Operating Principles of Structured Light Depth Cameras. In *Time-of-Flight and Structured Light Depth Cameras*; Pietro, Z., Giulio, M., Carlo, D.M., Fabio, D., Ludovico, M., Guido, M.C., Eds.; Springer International Publishing: Heidelberg, Germany, 2016; pp. 43–79.
8. Furukawa, Y.; Ponce, J. Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment. *Int. J. Comput. Vis.* **2009**, *84*, 257–268. [CrossRef]
9. Puwein, J.; Ziegler, R.; Vogel, J.; Pollefeys, M. Robust Multi-View Camera Calibration for Wide-Baseline Camera Networks. In Proceedings of the Applications of Compute Vision of the Workshop, Kona, HI, USA, 5–7 January 2011.
10. Li, B.; Heng, L.; Koser, K.; Pollefeys, M. A Multiple-Camera System Calibration Toolbox Using a Feature Descriptor-Based Calibration Pattern. In Proceedings of the Intelligent Robots and Systems of the Conference, Tokyo, Japan, 3–7 November 2013.
11. Holz, D.; Ichim, A.E.; Tombari, F.; Rusu, R.B.; Behnke, S. Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D. *IEEE Robot. Autom. Mag.* **2015**, *22*, 110–124. [CrossRef]
12. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
13. Matabosch, C.; Salvi, J.; Fofi, D.; Meriaudeau, F. Range Image Registration for Industrial Inspection. In Proceedings of the SPIE of the Conference, San Jose, CA, USA, 17 January 2005; pp. 216–227.
14. Salvi, J.; Matabosch, C.; Fofi, D.; Forest, J. A Review of Recent Range Image Registration Methods with Accuracy Evaluation. *Image Vis. Comput.* **2007**, *25*, 578–596. [CrossRef]
15. Chen, Y.; Medioni, G. Object Modeling by Registration of Multiple Range Images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
16. Pulli, K. Multiview Registration for Large Data Sets. In Proceedings of the 3-D Digital Imaging and Modeling of the Conference, Ottawa, ON, Canada, 8 October 1999.
17. Masuda, T. Object Shape Modeling from Multiple Range Images by Matching Signed Distance Fields. In Proceedings of the 3D Data Processing Visualization and Transmission of the Symposium, Padova, Italy, 19–21 June 2002.

18. Silva, L.; Bellon, O.R.P.; Boyer, K.L. Enhanced, Robust Genetic Algorithms for Multiview Range Image Registration. In Proceedings of the 3-D Digital Imaging and Modeling of the Conference, Banff, AB, Canada, 6–10 October 2003.
19. Rusinkiewicz, S.; Levoy, M. Efficient Variants of the ICP Algorithm. In Proceedings of the 3-D Digital Imaging and Modeling of the Conference, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
20. Low, K.-L. *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*; Technical Report TR04-004; Department of Computer Science University, North Carolina at Chapel Hill: Orange, NC, USA, 2004.
21. Serafin, J.; Grisetti, G. NICP: Dense Normal Based Point Cloud Registration. In Proceedings of the Intelligent Robots and Systems of the Conference, Hamburg, Germany, 28 September–2 October 2015; pp. 742–749.
22. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In Proceedings of the User Interface Software and Technology of the Conference, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
23. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M. Learning Informative Point Classes for Acquisition of Object Model Maps. In Proceedings of the Control, Automation, Robotics and Vision of the Conference, Hanoi, Vietnam, 17–20 December 2008.
24. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetx, M. Aligning Point Cloud Views using Persistent Feature Histograms. In Proceedings of the Intelligent Robots and Systems of the Conference, Nice, France, 22–26 September 2008.
25. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (FPFH) for 3D Registration. In Proceedings of the Robotics and Automation of the Conference, Kobe, Japan, 12–17 May 2009.
26. Tombari, F.; Salti, S.; Stefano, L.D. Unique Signatures of Histograms for Local Surface Description. In Proceedings of the Computer Vision of the European Conference, Heraklion, Crete, Greece, 5–11 September 2010.
27. Nascimento, E.R.; Schwartz, W.R.; Oliveira, G.L.; Veira, A.W.; Campos, M.F.M.; Mesquita, D.B. Appearance Geometry Fusion for Enhanced Dense 3D Alignment. In Proceedings of the Graphics, Patterns and Images of the Conference, Ouro Preto, Brazil, 22–25 August 2012.
28. Schmiedel, T.; Einhorn, E.; Gross, H.-M. IRON: A Fast Interest Point Descriptor for Robust NDT-Map Matching and Its Application to Robot Localization. In Proceedings of the Intelligent Robots and Systems of the Conference, Hamburg, Germany, 28 September–2 October 2015.
29. Song, W.; Yun, S.; Jung, S.-W.; Won, C.S. Rotated Top-Bottom Dual-Kinect for Improved Field of View. *Multimed. Tools Appl.* **2016**, *75*, 8569–8593. [CrossRef]
30. Li, H.; Liu, H.; Cao, N.; Peng, Y.; Xie, S.; Luo, J.; Sun, Y. Real-Time RGB-D Image Stitching Using Multiple Kinects for Improved Field of View. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1–8. [CrossRef]
31. Mellado, N.; Mitra, N.; Aiger, D. SUPER 4PCS Fast Global Pointcloud Registration via Smart Indexing. In Proceedings of the Computer Graphics of the European Association Conference, Strasbourg, France, 7–11 April 2014.
32. Segal, A.V.; Haehnel, D.; Thrun, S. Generalized-ICP. In Proceedings of the Robotics Science and Systems of the Conference, Seattle, WA, USA, 29 June–1 July 2009.
33. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-Points Congruent Sets for Robust Pairwise Surface Registration. *ACM Trans. Graph.* **2008**, *27*, 1–10. [CrossRef]
34. Zhou, Q.-Y.; Park, J.; Koltun, V. Fast Global Registration. In Proceedings of the Computer Vision of the European Conference, Amsterdam, The Netherlands, 8–16 October 2016; pp. 766–782.
35. Murray, R.M.; Li, Z.; Sastry, S.S. A Mathematical Introduction to Robotic Manipulation. In *Forward Kinematics*; CRC Press: Boca Raton, FL, USA, 1994; pp. 83–95.
36. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. Forward Kinematics: the Denavit-hartenberg Convention. In *Robot Dynamics and Control*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2004; pp. 57–82. ISBN 978-0-471-61243-8.
37. Matkan, A.A.; Hajeb, M.; Mirbagheri, B.; Sadeghian, S.; Ahmadi, M. Spatial Analysis for Outlier Removal from LiDAR Data. In Proceedings of the Geospatial Information Research of the European Conference, Tehran, Iran, 15–17 November 2014.

38. Wolff, K.; Kim, C.; Zimmer, H.; Schroers, C.; Botsch, M.; Sorkine-Hornung, O.; Sorkine-Hornung, A. Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction. In Proceedings of the 3D Vision of the Conference, Stanford, CA, USA, 25–28 October 2016.
39. ASUS Xtion Pro. Available online: https://www.asus.com/3D-Sensor/Xtion_PRO/ (accessed on 12 August 2017).
40. E46-17 Pan-tilt Platform. Available online: http://www.flir.com/mcs/view/?id=63554 (accessed on 12 August 2017).
41. Automatic Registration of Ground-Based LiDAR Point Clouds of Forested Areas. Available online: http://www.cis.rit.edu/DocumentLibrary/admin/uploads/CIS000133.pdf (accessed on 12 August 2017).