Single-Shot Extrinsic Calibration of a Generically Configured RGB-D Camera Rig from Scene Constraints

Jiaolong Yang†
School of Computer Science
Beijing Institute of Technology

Yuchao Dai‡
Research School of Computer Science
CECS, The Australian National University

Hongdong Li‡
Research School of Engineering
CECS, The Australian National University

Henry Gardner†
Research School of Computer Science
CECS, The Australian National University

Yunde Jia†
School of Computer Science
Beijing Institute of Technology

ABSTRACT

With the increasing use of commodity RGB-D cameras for computer vision, robotics, mixed and augmented reality and other areas, it is of significant practical interest to calibrate the relative pose between a depth (D) camera and an RGB camera in these types of setups. In this paper, we propose a new single-shot, correspondence-free method to extrinsically calibrate a generically configured RGB-D camera rig. We formulate the extrinsic calibration problem as one of geometric 2D-3D registration which exploits scene constraints to achieve single-shot extrinsic calibration. Our method first reconstructs sparse point clouds from a single-view 2D image. These sparse point clouds are then registered with dense point clouds from the depth camera. Finally, we directly optimize the warping quality by evaluating scene constraints in 3D point clouds. Our single-shot extrinsic calibration method does not require correspondences across multiple color images or across different modalities and it is more flexible than existing methods. The scene constraints can be very simple and we demonstrate that a scene containing three sheets of paper is sufficient to obtain reliable calibration and with a lower geometric error than existing methods.

Index Terms: I.4.1 [IMAGE PROCESSING AND COMPUTER VISION]: Digitization and Image Capture—Camera calibration; H.5.1 [INFORMATION INTERFACES AND PRESENTATION]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Range data; I.4.3 [IMAGE PROCESSING AND COMPUTER VISION]: Enhancement—Registration

1 INTRODUCTION

Augmented Reality (AR) and Mixed Reality (MR) systems enhance a user’s interaction with the real world through additional information generated by computer models of the world. With the popularity of commodity RGB-Depth cameras such as the Microsoft Kinect [16], the development of certain 3D-perception-based AR/MR applications has become easier than in the past. In most AR/MR systems these applications require the insertion of 3D graphics models of real and virtual objects. In order to align these graphical models with the scene itself, it is necessary to know the relative pose between the RGB camera and the depth camera. More generally, whenever one employs cameras of different modalities to observe a target from different viewpoints, it is often desirable to calibrate (or register) the obtained images in order to properly combine the information captured by different cameras.

In this paper, we study the extrinsic calibration (or geometric-registration) problem for an RGB-D camera rig—i.e. a system made up of one regular, visual camera and one depth-sensing camera as shown in Fig. 1. The two cameras can be displaced relatively far away from each other. The only constraint is that both cameras must share a common area for both extrinsic calibration and for AR/MR applications. We make the assumption that each camera has already been intrinsically calibrated, thus we deal with extrinsic calibration only. This assumption is not too restrictive as the intrinsic parameters can be readily obtained using a separate intrinsic-calibration procedure or from manufacturers’ specifications. Additionally, in practical AR/MR applications, the intrinsic parameters of the cameras are generally fixed while the extrinsic parameters can be subject to change (for example, the rig might include a roving, handheld camera).

Up until now, the color-and-depth extrinsic calibration (registration) task has been mostly achieved in a way that is very similar to the conventional procedure of calibrating a regular visual camera. Typically, this involves waving a calibration checkerboard in front of the cameras, and processing images across several shots and correspondences across the different cameras (which can have different modalities). In this paper, we propose a new method to extrinsically calibrate an RGB-D camera rig in a single-shot and correspondence-free style, where minimal human interaction is required. Rather than using a specified calibration pattern, our method makes use of a small set of known scene constraints (known distances and angles) in a single-shot. Being a single-shot method implies that the extrinsic calibration can be recomputed if the camera poses change between shots (as with a hand-held camera). Our method could even be applied to post-process single shots provid-

Figure 1: A customized RGB-D camera rig consisting of a high-resolution color camera, and a Kinect-for-Windows depth sensor. This rig was used in the experimental work described in this paper.
ing some geometric constraints within those images are known. The set of constraints can be quite simple as we demonstrate below in Section 4.

Organization: Following a discussion of related work, in the next subsection, we present the formulation of our method in Section 2. In our method, we directly minimize the geometric registration error between the RGB and depth cameras. This not only frees us from the burden of finding accurate cross-modality correspondences, but also directly optimizes the quality of registration. We formulate the problem as one of geometric 2D-3D registration that utilizes metric information from the scene directly. As an additional benefit, our method also leads to a way of estimating 3D structure for a 2D color image. The process of nonlinear optimization requires a good initial guess, such as the one that we subsequently present in our Section 3. In Section 4, we present results of our method for a synthetic scene, for a real-world scene and for an augmented reality application. These experiments show that our simple set of scene constraints can be used to accurately register the RGB and depth cameras.

1.1 Related Work
Extrinsic calibration of a generically configured RGB and depth camera pair has attracted considerable attention from computer vision [27], mixed and augmented reality [17][8] and robotics communities [28]. In this section we briefly review the most relevant prior work to our method.

Closest to our method for extrinsic calibration of a color camera with a depth camera, is the work by Herrera et al. [9] and Zhang and Zhang [27]. Herrera et al. [9] presented a method to jointly calibrate two color cameras and a depth camera by using a planar pattern surface which was imaged from various poses. Zhang and Zhang [27] proposed calibrating the color camera and depth camera (Kinect), where correspondences between the color image and the depth image are used to improve accuracy. Smisek et al. [21] calibrated Kinect cameras using correspondences between the RGB image and the infrared image.

In other related work, Zhang and Pless [28] proposed a practical procedure to extrinsically calibrate an RGB camera with a 2D Laser-Rangeﬁnder (LRF), where a checkerboard pattern was moved freely in front of both sensors. Extrinsic calibration was achieved by solving a set of linear constraints which were subsequently refined by iterative minimization of the reprojection error. Scaramuzza et al. [20] claimed to have achieved extrinsic self-calibration of an RGB camera with a 3D LRF using a method that can be done “on the ﬂy”, but their method actually requires manual intervention to select point correspondences between the RGB image and the transformed LRF image. Alismail et al. [1] used a simple calibration target consisting of a single circle and solved the extrinsic parameters by point-to-plane Iterative Closest Point (ICP) with nonlinear optimization via the Levenberg-Marquardt algorithm. Like Zhang and Pless [28], Vasconcelos et al. [25] also studied the calibration of an RGB camera with a 2D LRF and they showed that a set of three pairs of planes and lines provides a minimal conﬁguration to solve the calibration problem linearly.

More recent work by Geiger et al. [7] obtained extrinsic calibration in a single shot but depended on the usage of a multiple-checkerboard-pattern conﬁguration and also involved an explicit segmentation of the planar regions corresponding to the checkerboard. In contrast, our single-shot method is much more general because it utilizes various kinds of scene constraints and evaluates the calibration with registration error, thus optimizing warping quality directly.

Extrinsic calibration of a camera rig also has a close relationship with hand-eye calibration which has been intensively studied in computer vision and robotics [24, 10, 4]. However, these methods work in scenarios where relative motion can be easily measured and motion of the camera rig is required. In addition, they have commonly used an algebraic error to evaluate the performance. Neither of these restrictions is necessarily satisfied for the problem of extrinsic calibration of a generically configured RGB-D camera rig.

2 Single-shot extrinsic calibration of an RGB and depth camera rig from scene constraints

2.1 Problem statement
The ultimate goal of extrinsic calibration of an RGB-D camera rig is to bring the obtained 2D color image and 3D depth image into perfect geometric alignment (registration). This process can be intuitively understood as either,

- to color each pixel in the depth image with the correct color, or, conversely,
- to assign each pixel in the color image a correct depth value.

Given a perfect registration, these two statements are equivalent. Mathematically, the underlying task estimates the relative geometric transformation between the two cameras. The transformation involves a rotation matrix, \( R \), and a translation vector \( t \), forming a 6-dof (degrees-of-freedom) rigid transformation. Finding this set of extrinsic parameters, \( \Theta = [R, t] \), is precisely the goal of extrinsic calibration.

Such an extrinsic calibration for an RGB-D camera rig can be formulated as the “pose estimation problem”: when the two cameras are observing the same scene, for each scene point we can simply obtain its 3D coordinates from the depth image. If we are able to identify the corresponding pairs of 2D image points and depth points, then we can obtain the mapping relationship between the 3D scene and its 2D image coordinates, which is deﬁned in terms of the (as yet unknown) extrinsic parameters \( \Theta \). Solving this camera pose problem gives the desired RGB-D calibration. But this approach, while conceptually simple and straightforward, is not an easy task in practice. The main difﬁculty comes from the necessary requirement of knowing cross-modality feature correspondences between the RGB image and the depth image. Moreover, most existing methods require multiple shots of a calibration pattern to achieve extrinsic calibration.

2.2 Our Approach
In this paper, we solve the extrinsic calibration problem in a “single-shot” and “correspondence-free” style. Our method aims to optimize the ﬁnal “warping” quality directly by minimizing the geometric registration error between a color camera and a depth camera. We evaluate the warping quality from scene knowledge. Under perfect extrinsic calibration, scene knowledge observed from the color image should have identical measurements in the corresponding 3D point clouds from the depth image. Thus we can potentially achieve single-shot calibration by minimizing the discrepancy between this prior scene knowledge and its estimation. In addition, we can build a 3D estimation for the RGB image under assumptions about the smoothness and continuity of the scene. We call this process “inverse projection”, which estimates 3D positions from 2D image coordinates.

Our method works by capturing a single shot of a scene, provided that certain constraints about the scene are easy to access. The extrinsic calibration task is achieved by solving a 2D-3D registration problem. Of course, in the absence of scene constraints, doing such a 2D-3D registration is generally impossible due to the information loss in the projection from the 3D to 2D. Nevertheless, our knowledge (including qualitative assessments) of the scene can help provide feedback on the quality of the registration. For example, we invite the reader to look at the schematic example in Fig. 2(b) where it is not difﬁcult to suspect (or to guess) that this situation is
very likely to be not registered well while Fig. 2(c) gives a much better registration. In this example, scene constraints including known distances, distance equivalency and known angles are involved (as discussed in the next subsection).

In the following sub-sections, we first introduce our inverse projection method to estimate a 3D position for a 2D image point. We then illustrate how to incorporate different scene constraints in evaluating the registration performance. Finally, we present our geometric-registration-error-minimization based extrinsic-calibration method that directly optimizes the warping quality between the two images.

2.3 Inverse projection estimation

Given a rigid body transformation between the RGB and depth cameras, \( \Theta = (R, t) \), we can transform the point clouds \( \mathcal{P}_D = \{ (x_i^d, y_i^d, z_i^d) \} \) from the depth camera to the coordinates of the color camera, and project it onto the image plane using the intrinsic matrix \( K \). This procedure can be expressed as

\[
\lambda_i^d = \begin{bmatrix} x_i^d \\ y_i^d \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} x_i^c \\ y_i^c \\ 1 \end{bmatrix},
\]

where \( \lambda_i^d \) gives the color image point corresponding to the 3D point \( (x_i^d, y_i^d, z_i^d) \) and \( \lambda_i^c \) is the unknown projective depth. This mapping relationship can be compactly expressed as:

\[
(u_i^d, v_i^d) = g(\Theta) \circ (u_i^c, v_i^c),
\]

where \( g \) denotes the transformation from a 3D point in the depth camera to the color image coordinate.

Now that we have obtained 3D positions for some image pixels on the color image, the question is: can we obtain 3D positions for all the pixels in the color image? This is generally impossible as the projection from 3D to 2D is an information-loss procedure. Nonetheless, we can make a local (piecewise) smoothness assumption, under which the inverse projection \( g^{-1}(\Theta) \) may be well defined, and we can recover \( (x_j^d, y_j^d, z_j^d) \) through \( g^{-1}(\Theta) \circ (u_j^d, v_j^d) \) based on the local structure around particular \( (u_j^d, v_j^d) \).

We use triangulation to estimate this inverse projection \( g^{-1}(\Theta) \), assuming a locally smooth surface. Specifically, we obtain surface triangles for the dense 3D point clouds \( \{ (x_k^d, y_k^d, z_k^d) \} \) from the depth camera. Given a current estimation of \( \Theta \), the triangles \( \Delta_k \) are first rigidly transformed and then projected onto the image plane of the RGB camera. Note that the projected triangles do not necessarily correspond to the 2D Delaunay triangulation of the projected 3D points because, due to self-occlusion of the scene, there are possibly triangles overlaid on top of each other. For an image point \( (u_i^d, v_i^d) \), we first find the projected triangles containing this point, then back-project the image point onto the 3D space. We then obtain the estimated 3D positions from triangles containing the image point. If there are multiple 3D points corresponding to the image point, we choose the one with the smallest depth, thus effectively handling the occlusion. The procedure of inverse projection is illustrated in Fig. 3.

Finally, the inverse projection can be written as

\[
(u_i^d, v_i^d) = g^{-1}(\Theta) \circ (u_i^c, v_i^c),
\]

where \( (u_i^c, v_i^c) \) is any point on the image plane of the RGB camera.

2.4 Scene constraints

Once we have the inverse projection \( g^{-1}(\Theta) \), we can evaluate metric information, and compare it with prior knowledge about the scene. This prior knowledge can include (but is not limited to) three broad types of constraints which we discuss here.

Known-distance constraints: Suppose we have two feature points from the color image denoted as \( (u_i^c, v_i^c) \) and \( (u_j^c, v_j^c) \), and we know their Euclidean distance, \( l_{ij} \). By applying the inverse projection, we obtain their world-be distance, which is \( \|g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)\| \). Then, the discrepancy from the known distance, given by

\[
e_k(\Theta) = \|g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)\| - l_{ij},
\]

measures how good the tentative registration is. This known-distance constraint fixes the distance between points on two lines. Given enough known distance constraints, we are able to recover the 3D coordinates of the points. In general there needs to be at least one known distance constraint to fix the global scale of the 2D-3D registration.

Distance-equivalency constraints: If, for instance, we know that the distance between one pair of feature points, \( (u_i^c, v_i^c) \) and \( (u_j^c, v_j^c) \) should be the same as the distance between another pair of points, \( (u_k^c, v_k^c) \) and \( (u_l^c, v_l^c) \), then the observed discrepancy is expressed as \( e_u(\Theta) = \|g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)\| - \|g^{-1}(\Theta) \circ (u_k^c, v_k^c) - g^{-1}(\Theta) \circ (u_l^c, v_l^c)\| \). This is another useful constraint but it cannot be applied alone as solely using distance-equivalency constraints could result in the trivial solution of all distances being zero.
Angular constraints: Besides the scene constraints from distance measurements, we can also evaluate angular constraints about the scene such as the preservation of orthogonal and parallel lines in the images. Discrepancy from an orthogonal constraint can be expressed as $e_{o}(\Theta) = (g^{-1}(\Theta) \circ (u_{i}, v_{i}')) - g^{-1}(\Theta) \circ (u_{i}', v_{i}')$ for parallel lines and $e_{o}(\Theta) = (g^{-1}(\Theta) \circ (u_{i}, v_{i})') - g^{-1}(\Theta) \circ (u_{i}', v_{i})'$ for parallel lines. Discrepancy from a parallel constraint can be expressed as $e_{p}(\Theta) = (g^{-1}(\Theta) \circ (u_{i}, v_{i})') - g^{-1}(\Theta) \circ (u_{i}', v_{i})'$. 

2.5 Geometric error minimization

Given an RGB color image, assume that certain metric information about the scene is available (e.g., inter-point distances between some pairs of image features, parallel or orthogonal constraints between lines, distance equivalency and so on – a set of rather mild and general conditions on the images), then for any tentative 2D-3D registration (parameterized by $\Theta$), we can always quantitatively measure registration quality using the discrepancy between the estimation and the a priori knowledge. Minimizing this discrepancy directly leads to the optimal extrinsic calibration, as well as a direct optimization of the warping quality. This is the main idea behind our proposed method.

Mathematically, our method formulates the problem of extrinsic calibration as searching for an optimal rigid transformation $\Theta^* = \{R^*, t^*\}$ that minimizes geometric error:

$$\Theta^* = \arg \min_{\Theta \in SE(3)} \sum_{i} e_i(\Theta)^2, \quad (5)$$

where $e_i(\Theta)$ is the discrepancy between a measurement and its corresponding prior knowledge under the transformation $\Theta$.

Due to the fact that there is no explicit form of the inverse-projection function, we are not able to employ analytic gradient-based methods for solving the minimization problem. Instead, the implicit inverse-projection function means that evaluating the scene knowledge with respect to the rigid transformation results in a complex, nonlinear, optimization problem and numerical gradient-based methods such as the Levenberg-Marquardt algorithm [14] can be used. The desired rigid transformations reside on the Riemannian manifold $SE(3)$, which is homeomorphic to $SO(3) \times \mathbb{R}^3$ [23]. Thus we can deal with rotation and translation individually.

The constraints on $SO(3)$ have to be involved in parameterizing the rotation (angle-axis representation is used in our algorithm implementation). Alternatively, other gradient free algorithms such as the Nelder-Mead simplex downhill algorithm [15] can also be used by adapting a “simplex downhill on manifold” optimization algorithm from [5], that searches the 6 parameters on the manifold.

To solve the non-linear minimization problem in evaluating scene constraints, we need a good initial guess such as the one described in the next section.

3 A SIMPLE SOLUTION FOR AN INITIAL ESTIMATE OF THE EXTRINSIC CALIBRATION

In this section we present a simple solution to estimate the extrinsic calibration, which can be used to initialize the nonlinear optimization. Note that, although simple, this method is a self-contained solution and is interesting in its own right. The method takes advantage of the scene constraints to reconstruct the extracted color-image points and then to register the reconstructed 3D points with the dense 3D point clouds from the depth camera to obtain a solution for the rigid body transformation. Note that the objective function for 3D registration is different from directly evaluating scene constraints as in Eq. (5).

3.1 Single view 3D reconstruction

Generally, a single view 3D reconstruction is impossible without any scene information. But, with partial scene constraints such as known distances, distance equivalency and known angles, we are able to reconstruct the 3D scene from measurements on a single-view image.

Under the color camera coordinate, the perspective imaging process is expressed as $\lambda_i[a_i' v_i' 1]^T = K[I 0] [X_i' Y_i' Z_i' 1]^T$, where $[X_i' Y_i' Z_i' 1]^T$ is the image measurement, $[X_i Y_i Z_i]^T$ is the corresponding 3D position and $\lambda_i$ is the unknown projective depth. This equation actually gives a direction constraint on the 3D position, say, $[X_i Y_i Z_i]^T = \lambda_i[K^{-1}[a_i' v_i' 1]^T]$, i.e., the 3D point lies on the ray with direction $K^{-1}[a_i' v_i' 1]^T$ with an unknown projective depth $\lambda_i$ to be determined.

With scene constraints such as known distances, distance equivalency and known angles, we have further constraints on the projective depths. Thus it is possible to recover the scene structure. Take the known distance constraint as an example (Fig. 4), the distance between two 3D points is measured as:

$$d_{ij} = \|\lambda_i K^{-1}[a_i' v_i' 1]^T - \lambda_j K^{-1}[a_j' v_j' 1]^T\|_2, \quad (6)$$

which gives constraint on the projective depths $\lambda_i$ and $\lambda_j$. By defining $a_{ij} = (K^{-1}[a_i' v_i' 1]^T)^T (K^{-1}[a_j' v_j' 1]^T)$, Eq. (6) gives the following bilinear equation on $\lambda_i$ and $\lambda_j$:

$$d_{ij}^2 = \lambda_i^2 a_{ii} + \lambda_j^2 a_{jj} - 2 \lambda_i \lambda_j a_{ij}, \quad (7)$$

which can be equivalently expressed as:

$$[\lambda_i \lambda_j] \begin{bmatrix} a_{ii} & -a_{ij} \\ -a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \lambda_i \\ \lambda_j \end{bmatrix} = d_{ij}^2, \forall (i, j) \in \mathcal{N}, \quad (8)$$

where $\mathcal{N}$ defines the set of all measured point pairs.

We define a vector $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$, which contains all the projective depths to determine. $Y = \Lambda \Lambda^T$ is defined as the Gram matrix and $\text{rank}(Y) = 1$. Define $k_{ij} = \text{tr}(Y)$ as being element-wise zero except for $k_{ii} = a_{ii}, k_{jj} = a_{jj}$, and $k_{ij} = -a_{ij}$. Then, the bilinear constraint on the projective depth can be expressed as:

$$\text{tr}(\Lambda_i Y) = d_{ij}^2, \forall (i, j) \in \mathcal{N}. \quad (9)$$

Figure 4: Single view reconstruction with scene constraints.
Finally the problem of single view 3D reconstruction from known distance constraints is formulated as:

Find \( \Lambda \),

such that \( \text{tr}(A_{ij}Y) = d_{ij}^2, \forall (i, j) \in \mathcal{N} \),

\( Y = \Lambda \Lambda^T \),

\( \text{rank}(Y) = 1 \).

(10)

The quadratic constraint and the rank constraint are both non-convex, thus the entire optimization problem is non-convex. We take a similar strategy to [13] to “convexify” the constraints, proposing to minimize the trace norm of \( Y \) rather than enforcing the rank-constraint implicitly. Finally we reach a trace norm minimization problem as:

\[
\min \text{trace}(Y) \\
\text{such that } \text{tr}(A_{ij}Y) = d_{ij}^2, \forall (i, j) \in \mathcal{N}.
\]

(11)

This is a standard Semi-Definite Programming (SDP) problem and we can use off-the-shelf solvers such as SDPT3 [22] to solve it efficiently. Once we obtain \( Y \), we can solve \( \Lambda \) by the singular value decomposition (SVD). Finally the 3D structure is recovered as \( X_n = \lambda_i K^{-1} [u^T, v^T, 1]^T \). As an example, a single view 3D reconstruction for Fig. 8(a) is illustrated in Fig. 8(b).

Minimal Configuration: For each ray, there is an unknown projective depth \( \lambda_i \). For \( n \) points in a complete connected graph with known distances, we have \( n(n - 1)/2 \) constraints, therefore when \( n(n - 1)/2 > n \), we will have enough constraints to solve \( \lambda_i \). Thus \( n = 3 \) gives the minimal configuration. However, under this configuration, multiple solutions exist. To retrieve a unique solution, at least 4 points with known distances should be involved.

Other Scene Constraints: In the previous paragraph, we have taken the known distance constraint as an example to demonstrate how to recover 3D points from single-view 2D image measurements. In principle, other constraints can also be incorporated into the above framework as follows:

- Distance equivalency constraint: \( d_{ij} = d_{kl} \), which gives a linear equation of \( Y \) as \( \text{tr}(A_{ij}Y) = \text{tr}(A_{kl}Y) \);
- Orthogonal constraint of lines \( L_{ij} \) and \( L_{kl} \) is expressed as \( (\lambda_i K^{-1} [u^T, v^T, 1]^T - \lambda_j K^{-1} [u^T, v^T, 1]^T) (\lambda_k K^{-1} [u^T, v^T, 1]^T) = 0 \), which gives a linear equation of \( Y \) as \( a_{ik}Y_{ik} + a_{jk}Y_{kj} - a_{ij}Y_{ij} - a_{ik}Y_{ik} = 0 \);
- Parallel constraint of lines \( L_{ij} \) and \( L_{kl} \) is expressed as \( \lambda_i K^{-1} [u^T, v^T, 1]^T - \lambda_j K^{-1} [u^T, v^T, 1]^T = 0 \), which gives three linear equations of \( Y \).

All these constraints can be incorporated into the above trace norm minimization formulation naturally. Different types of scene knowledge constrain the 3D reconstruction to different extents. For example, using only the distance equivalency constraint, we can only achieve reconstruction up to a global scale, where a trivial solution as all depths being zero is included. Angle-based constraints, such as orthogonal or parallel constraints, result in 3D reconstruction up to a global scale and rotation. For single-view reconstruction from scene constraints, at least one known distance constraint is required to obtain a global scale.

Related Work: Note that our single view 3D reconstruction has connections with the Perspective-n-Point (PnP) problem [12], where the camera motion is the main focus to solve. Meanwhile Zhang et al. [29] used domain knowledge such as distances and angles to upgrade the affine structure into a Euclidean space by minimizing the sum of Mahalanobis distances, which is solved as a general nonlinear least-squares problem. Wilczkowiak et al. [26] exploited geometric constraints through parallelepipeds for calibration and 3D modeling. Nevertheless, our method is based on recent progress in compressive sensing theory [18] and provides much more efficient implementation.

3.2 Point set registration

Now that we have sparse point clouds \( \{ X_n \} \) from a single view 3D reconstruction, the initial transformation \( \Theta_0 \) can be obtained by registering \( \{ X_n \} \) to the dense point clouds \( \{ X_d \} \) from the depth camera. The well-known Iterative Closest Point (ICP) algorithm [2] can be used to get the solution as

\[
\Theta_0 = \arg\min_{\Theta \in SE(3)} \sum_j \min_i \| (RX_i^j + t) - X_d^j \|^2.
\]

(12)

To handle large displacements in registration, we can globalize the registration with some stochastic framework such as [19], or even obtain the globally optimal solution by using branch-and-bound methods such as [6] since \( \{ X_n \} \) is rather sparse and registration error is known a priori to be small (which provides extremely tight bounds). To further handle outliers in 3D point clouds, robust versions of ICP such as [3] and [11] can be extended. This is out of the focus of the current paper especially considering the fact that scene constraints are extracted in 2D images through human interaction.

When registering sparse point clouds with dense point clouds, uniqueness of the solution necessarily depends on the scene structure. For example, if multiple similar or even repetitive structures exist in the scene and all the sparse point clouds sample from these structures, then we may obtain different rigid transformations giving exact registrations. In real world AR/MR applications, the scene is generally complex enough to avoid such degenerate cases.

4 Experiments

In this section, we present experimental results on generically-configured RGB-D camera rigs. We first give a synthetic scenario with two cylinders to illustrate the generality and performance of our method on minimizing geometric error. Then three sheets of A4 paper in the real world are used to extrinsically calibrate a generically configured RGB-D camera rig.

Performances of extrinsic calibration and alignment were evaluated both qualitatively (by warping the depth image onto the color image) and quantitatively (by measuring the geometric error from scene constraints). All the experiments were run on a computer with 2.4GHz Intel Core i5 CPU.

4.1 Tests on synthetic data

In the first experiment, a scene containing two non-parallel cylinders was synthesized as shown in Fig. 5(a). We then synthesized a single-shot of an RGB-D camera. The color image was computed via a simple pinhole model, and the depth map was generated by the Z-buffer technique. The synthesized RGB-D image pair is shown in Fig. 5(b) (with square grids overlaid) and Fig. 5(c). We took the true side-length of the grid as constraints about the scene. The Levenberg-Marquardt algorithm was used to minimize the geometric error.

Quantitative evaluation. We used the objective function minimizing the sum of squared errors, while the root mean square (RMS) errors were recorded during each iteration as the performance measure. Convergence curve of our method is shown in Fig. 6. We compared our estimated parameters with the ground truth, and the results are given in Table 1, which shows that our method recovers the rigid transformation accurately.

Qualitative evaluation. The obtained registration results are shown in Fig. 5(d) and Fig. 5(e), which are the alignments before, and after applying our method, respectively. Visually inspected, our method yields satisfactory alignment.
Figure 5: Experiments on a synthetic scene. A scene containing two cylinders shown in (a) was synthesized. The color image shown in (b) was created by projecting the points onto the image plane using a pinhole model. The side-length of the labeled grids are known and used in our method. The depth image shown in (c) was computed with the Z-buffer technique. Initial alignment of the color and depth images are shown in (d). The alignment result with our method is shown in (e). (Best viewed on screen)

Table 1: Extrinsic calibration comparison between our method and Herrera et al. [9] are compared in Fig. 9. Our method achieves comparable or superior performance.

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>Angle (°)</th>
<th>Axis</th>
<th>Translation (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.067</td>
<td>-0.106,-0.128,-0.987</td>
<td>-0.113,-0.086,0.300</td>
</tr>
<tr>
<td>Our method</td>
<td>5.106</td>
<td>-0.096,-0.128,-0.986</td>
<td>-0.112,-0.078,0.503</td>
</tr>
</tbody>
</table>

4.2 Tests on a real-world scene

In the real-world extrinsic calibration task, we used the depth sensor on a Kinect device as our depth camera, and attached it to a high-resolution color camera (Fig. 1). Of course, our method can be adapted to other type of depth imaging sensors (e.g. 3D LIDAR, ToF camera, etc).

We set up a scene containing three sheets of A4 paper with different orientations, as shown in Fig. 8(a). These sheets of paper could just as well have been objects from an indoor scene such as a laptop screen, a book, a table or similar rigid objects with well-defined vertices. We then extracted the four corners of each sheet of A4 paper and the metric scene constraint was available as an international standard (the height and width of A4 paper). The single view 3D reconstruction result of the corner points is illustrated in Fig. 8(b). We ran a standard ICP method to register the reconstructed points with the dense point clouds from the depth camera to obtain an initial guess. Taking the quantization noise in the depth measurements obtained from the Kinect depth sensor into consideration, we utilized further constraints of the scene that the points were on planes to accurately estimate the depths for extracted points in the geometric error minimization procedure. Specifically, for each point on the color image a local plane was fitted with some nearest neighbors of vertexes of its corresponding 3D triangle. The whole calibration procedure including the corner points extraction and running of our extrinsic calibration method finished in minutes.

For comparison, the method of Herrera et al. [9] was also applied to calibrate the same RGB-D camera rig. We used 40 color and depth image pairs of a planar calibration pattern with 10 × 8 checkerboard grids. The corner points on the color images and plane regions on the depth images were manually selected to calibrate the RGB-D camera rig. To avoid any bias, the intrinsic parameters from [9] were used in our method.

Table 2: Extrinsic calibration results from our method and Herrera et al.[9], where the rotation is expressed in angle-axis representation.

<table>
<thead>
<tr>
<th></th>
<th>Angle (°)</th>
<th>Axis</th>
<th>Translation (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Herrera</td>
<td>17.225</td>
<td>0.102,-0.986,0.131</td>
<td>0.280,0.046,0.083</td>
</tr>
<tr>
<td>Our method</td>
<td>17.619</td>
<td>0.104,-0.983,0.153</td>
<td>0.273,0.043,0.091</td>
</tr>
</tbody>
</table>

Figure 8: Real-world scene and its corresponding 3D reconstruction. (a) Three sheets of A4 paper are used to provide scene constraint. (b) Single view 3D reconstruction of the extracted points.
Figure 10: An augmented reality demonstration. The first image is the original image from the RGB camera. With the extrinsic calibration result, we can fuse the depth information and color information precisely and the virtual teapot has then been added into the scene accurately. Note that occlusions have been handled effectively due to the depth information provided from extrinsic calibration. (Best viewed on screen)

Figure 9: Warping results for the generically configured RGB-D camera rig. First column: Herrera et al. [9]. Second column: our method. Significant differences are labeled with green boxes. (Best viewed on screen)

AR/MR Application. Finally, we provide a visual demonstration of an augmented reality application for a generically configured RGB-D camera rig, where a virtual teapot is added into a complex scene. As Fig. 10 shows, after being extrinsically calibrated, the 3D information from the depth camera and color information from the RGB camera can be fused accurately and occlusions in the augmented reality image can be effectively handled.

5 Conclusion

In this paper, we have presented a novel method to achieve extrinsic calibration (or self-alignment) of a generically configured RGB and depth camera rig with partially-known metric information of an observed scene in a single-shot fashion. Overall, the required human intervention is minimal and not restrictive as users only need to manually mark some points for the method to automatically obtain the extrinsic calibration. The whole procedure can be done efficiently, and the input can be as simple as three sheets of A4 paper or by other user provided scene information. Our calibration procedure can greatly facilitate mixed and augmented reality applications, which, for example, might require the use of a specialised RGB camera in addition to a commodity depth sensor or where it is desired to have a large displacement between the two cameras to cover a large region. Our method can also be adapted to other type of depth imaging sensors such as 3D LIDAR, ToF camera and etc. Additionally, as a single-shot extrinsic calibration method, our general formulation enables postprocessing of arbitrary single images, to, for example, insert graphical objects, providing some scene constraints in those images are known.

We suggest that the approach in this paper of directly minimizing the registration error in order to derive the calibration method is conceptually novel. Using this approach could not only lead to a more efficient solution than traditional approaches, but may also achieve registration results which better conform to our visual evaluation.

Acknowledgements

This work was supported in part by an Australian Research Council (ARC) Linkage Project (LP100100588) in partnership with Microsoft Corporation and Microsoft Research. Hongdong Li’s work was funded in part by two ARC Discovery grants (DP12 and DP13). Jiaolong Yang’s work was funded in part by the Specialized Research Fund for the Doctoral Program of China
REFERENCES


