Extrinsic Calibration of 3D Sensors Using a Spherical Target

Minghao Ruan and Daniel Huber
The Robotics Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
mhr|dhuber@cmu.edu

Abstract

With the emergence of relatively low-cost real-time 3D imaging sensors, new applications for suites of 3D sensors are becoming practical. For example, 3D sensors in an industrial robotic workcell can monitor workers' positions to ensure their safety. This paper introduces a simple-to-use method for extrinsic calibration of multiple 3D sensors observing a common workspace. Traditional planar target camera calibration techniques are not well-suited for such situations, because multiple cameras may not observe the same target. Our method uses a hand-held spherical target, which is imaged from various points within the workspace. The algorithm automatically detects the sphere in a sequence of views and simultaneously estimates the sphere centers and extrinsic parameters to align an arbitrary network of 3D sensors. We demonstrate the approach with examples of calibrating heterogeneous collections of 3D cameras and achieve better results than traditional, image-based calibration.

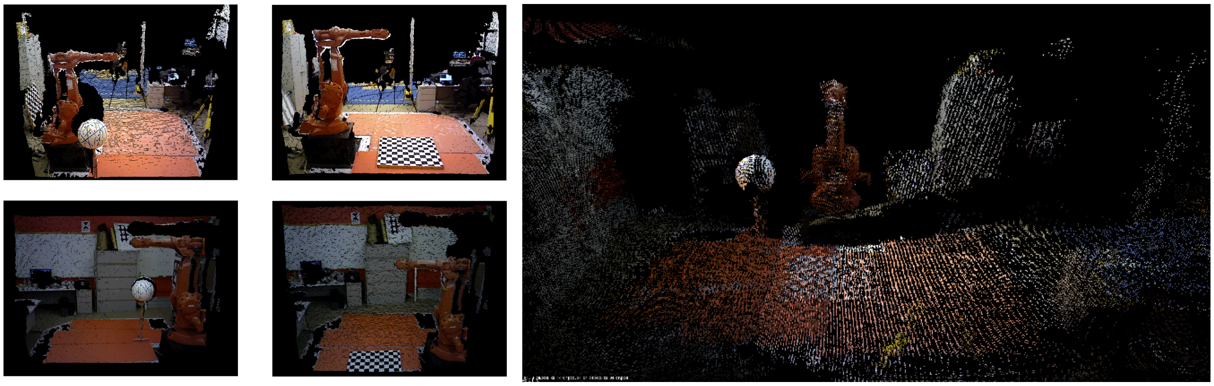
1. Introduction

Recent developments in 3D imaging have led to relatively low cost, real-time 3D cameras. Examples include time-of-flight flash lidars (e.g., Mesa's Swissranger), active stereo cameras (e.g., Microsoft's Kinect), and passive stereo cameras (e.g., Point Grey's Bumblebee and Tyzx's Deepsea2). The emergence of affordable 3D cameras has enabled new applications that require a suite of two or more 3D sensors working together. One example of such an application is the Hybrid Safety System (HSS), developed as part of the Intelligent Monitoring of Assembly Operations (IMAO) project [12]. The HSS uses a heterogeneous suite of 3D sensors to monitor an industrial robotic workcell, enabling workers to operate safely within the workspace of an operating robot [2]. In order to fuse data from multiple 3D sensors, their relative positions (i.e., extrinsic parameters) must be carefully calibrated. A similar calibration requirement arises when trying to combine data from multiple Kinect sensors [3].

In order for applications like the HSS to be useful in the field, the calibration process must be extremely simple – a procedure that can be performed by a factory worker with little or no camera calibration experience or training. Current camera calibration algorithms, such as Zhang's method [14], popularized by Bouget's Matlab Camera Calibration Toolbox [4], use a checkerboard or similar target, which is observed from several distances and orientations throughout the cameras' shared field of view. Such an approach requires a significant amount of knowledge by the person performing the calibration. They must be ensure that the target is oriented not too obliquely to the cameras, that it is close enough that the grid corners can be accurately detected and estimated, and that a sufficient number of views with different orientations, distances, and positions within the shared viewing region is collected.

Aside from being difficult for an untrained user to perform, planar target calibration algorithms suffer from the additional problem that for many camera arrangements, it is difficult or impossible to see the same planar target from all viewpoints. Consider a case where four cameras are placed at the upper corners of a room facing inwards. In this case, the only location potentially visible from all sensors is a horizontally oriented target near the floor. Furthermore, the comparatively low resolution of many 3D sensors makes it difficult to estimate checkerboard corners or other fiducial points accurately, and some 3D imagers don't measure appearance at all, making it impossible to use image-based techniques. These limitations point to a need for a simple, easy-to-use calibration method that can estimate the extrinsics for arbitrarily-placed 3D cameras observing a common workspace.

Inspired by the sphere-based registration algorithms used with terrestrial laser scanners, we propose an algorithm for extrinsic calibration of multiple 3D cameras that uses only a single, hand-held sphere target. To calibrate the cameras, a user simply carries the target around the workspace, moving it to different positions and heights within the shared viewing area. The algorithm automatically segments out



(a) Original Views (b) Fused point cloud

Figure 1. An extreme example of calibration task of two facing Asus Xtion Pros in which [11] produces suboptimal results. Furthermore, traditional image based algorithm will not work well as the checkerboard is hardly simultaneously visible in both views.

the person, detects the sphere in the data from each camera, and then optimizes the cameras' relative positions based on the fact that the sphere center in each view corresponds to the same underlying physical location. We derive a novel algorithm that simultaneously optimizes the sphere center estimates and the relative camera positions, providing a maximum likelihood estimate of the camera extrinsics and sphere positions.

In this paper, we propose a calibration method that is based on a simple spherical target. We present detailed steps to reliably segment out points on the surface of the sphere in each image followed by an unbiased estimator to compute the center of the sphere. The relative poses of all the sensors are first calculated with linear solver followed by a non-linear bundle adjustment. The final pose of the sensors together with the detected sphere in each frame are then simultaneously optimized. The primary contributions of this paper are 1) a novel MLE algorithm for simultaneously estimating 3D camera extrinsics and sphere positions and 2) an extremely simple, end-to-end extrinsic calibration algorithm.

The advantage and drawbacks of some related methods are discussed in Section 2. Details of our calibration pipeline are presented in Section 3. In Section 4, our method is demonstrated on a common double RGB-D camera setup as well as a particular multi-sensor configuration which consists three different depth sensors.

2. Related Work

Calibration of multiple perspective cameras is a well-studied problem in computer vision. The checkerboard method based on Zhang [14] and popularized by Bouget's Matlab toolbox [4] and OpenCV library [5] is the *de facto* algorithm for general camera calibration problems. Since many 3D sensors also produce intensity images, extrinsic calibration of a multiple 3D sensor system is often solved in the image space using well-established the checkerboard-based techniques. For example Kim *et al.* [10] calibrate their Swiss Rangers using infrared images of a checkerboard. Though such indirect methods work reasonably well for some problems, they are less effective to direct approaches because they essentially employ the weakness of a range sensor (i.e., the Swiss Ranger's low resolution imager) to calibrate the sensor's strength, which is the range measurement.

The checkerboard pattern can also be used as a hybrid approach to jointly calibrate perspective cameras, which can see corners, and range sensors, which can measure planes. The single-shot calibration system developed by Geiger *et al.* [8] is reliable and easy to use. However their system implicitly requires large overlaps in the field of view so the targets can be seen by all sensors simultaneously. This critical assumption, however, no longer holds when the application is designed to maximize coverage from distinct angles such as the IMAO system [12].

Other approaches that directly work with 3D shapes also exist. Auvinet *et al.* [3] shows that a plane can be used to calibrate multiple Kinects. Their method finds corresponding 3D points by intersecting multiple orthogonal plane-triplets from different frames. This method only works in environments with numerous orthogonal planes, or the user is required to hold a single plane in many different orientations for each sensor. Such an approach would require user training and would be more prone to data collection errors. Furthermore, the method requires at least triple the number of input images for calibration compared to our proposed method. Miller *et al.* [11] propose a targetless online calibration algorithm that directly infers the extrinsic parameters from the motion of objects in the scene. Their key contribution is an occlusion-aware energy function that improves Iterative Closest Point (ICP) alignment when the overlap between two views is small. Though the algorithm can deal with a reasonable amount of occlusion, its performance eventually degrades as the sensors get close to opposite viewing angles, in which case the algorithm will start producing "squishing" artifacts. Guan and Pollefeys [9] use a spherical target for their unified framework to jointly calibrate cameras and ToF sensors. The key problem they try to solve is to reliably estimate the center of the sphere in the depth sensor. Unfortunately, they rely on the sphere having a glossy surface so the center of the sphere is projected as a highlighted spot on the infrared image. As a result, they reduce the 3D calibration to a 2D bundle adjustment problem. Furthermore, due to the explicit use of infrared images, their framework is difficult to generalize beyond the flash LIDAR family of sensors.

Our approach is similar to that of [9] in that we also take advantage of the spatial invariance of a spherical target. Instead of solving the 2D sub-problem, we deal with 3D shapes directly and utilize all the available information to constrain the optimization. The result is an accurate and easy-to-use algorithm that functions well even when the sensors are placed in opposite views as shown in Figure 1.

3. The Calibration Algorithm

We assume each sensor has known intrinsic parameters and an overlapping view frustum with at least one other sensor. Most flash LIDARs and active stereos are calibrated by the manufacturer. It is possible to retrieve the intrinsic parameters (in particular the focal length and image center offsets) by solving the PnP problem using OpenCV [5]. Passive stereos can be easily calibrated with Bouget's Matlab toolbox [4]. We also assume that the 3D data from each sensor is, or can be, organized in a regular 2D grid – commonly known as an organized point cloud.

The calibration target for our algorithm is a sphere mounted on a stick or stand. Point clouds are collected from each of the 3D sensors with the sphere held or placed in various places within the shared viewing area. The sphere can be tracked over time, but in our implementation, it is detected independently in each of a small number of selected frames (3 or more are required). Our algorithm consists of four steps, with the first two performed for each selected frame and each sensor: 1) detect and segment the sphere from each point cloud; 2) estimate the center of each detected sphere; 3) solve for the extrinsic parameters; and 4) simultaneously optimize the extrinsic parameters and sphere center estimates. Each step is explained in detail in the rest of this section.

3.1. Detect and Segment the Sphere

In order to extract the center of the sphere, points on the sphere have to be grouped and segmented from the raw point cloud. As this work hinges on accurate estimation of the sphere, outliers from either sensor noise or other foreground objects must be removed.

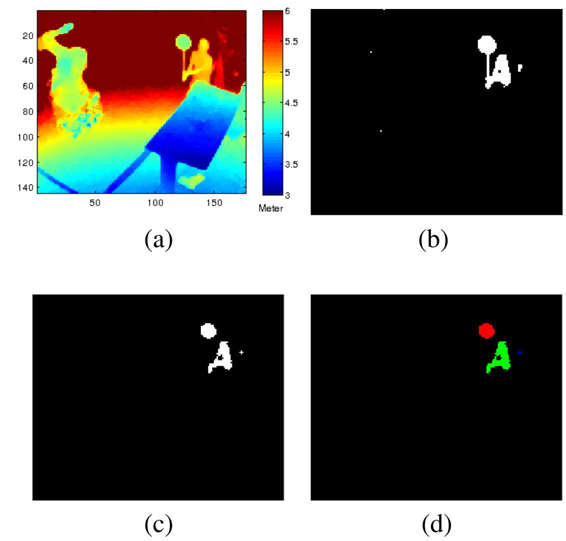


Figure 2. An example of segmenting the target from the supporting stick held by a human operator. (a) The original range image. (b) The binary image from background subtraction. (c) Output of the 2.5D opening operator. (d) Color labeled clusters.

A per-pixel background model is first built from a number of frames of the empty scene. After eliminating invalid depth values, a Gaussian distribution is fitted at each pixel. For the actual calibration frames, a pixel is marked as foreground if it is five times the standard deviation *in front of* the mean depth. This simple background subtraction will eliminate the static background as shown in Figure 2(b). Since the sphere is attached to a stick and held by a person or mounted on a stand, further processing is needed to isolate the sphere. A straightforward approach would be to use the morphological closing operation to isolate the sphere from its attached stick. However, the size of the required structuring element varies as a function of distance from the target. We therefore developed a 2.5D extension to the traditional opening operator that dynamically adjusts its scale as a function of distance. The process consists of 2.5D erosion followed by 2.5D dilation. Both of these operations can be efficiently implemented using a distance transform [6]. The procedure is summarized in Algorithm 1.

3.1.1 2.5D Erosion

Given the width of the thin part of the stick/stand $2w$ in Euclidean space, the goal of this operation is to remove all points that are within a distance w (in a plane parallel to the image plane) from a background pixel. This 2.5D operation effectively removes any cluster of points with width less than $2w$ and shrinks other clusters with width larger than $2w$. Though the fixed width w in Euclidean space is non-isotropic in the 2D pixel space, their relationship can be approximated by the property of similar triangles:

$$\Delta r = f\frac{w}{Z_i} \tag{1}$$

As opposed to thresholding the distance transformed image with a constant width that has to be found by trial and error, we can compute a per-pixel adaptive threshold as shown in Line 2-7 in Algorithm 1.

3.1.2 2.5D Dilation

Since dilation is the complement of erosion, the same procedure is repeated except on the inverted binary image. Caution has to be exercised when using the approximated radius in Equation 1 because it could potentially include points that lie far in the background. Hence at a given inverted foreground pixel $\tilde{b_i}$, we find the nearest point from the inverted background and check whether they are within w of each other. Though the nearest boundary pixel found by the distance transform may not always correspond to the real nearest point in 3D, it is an accurate local approximation when w is small compared to the scale of the target.

In practice the opening operator will not restore the exact shape of the sphere. Points very close to the edge of the sphere may be lost as shown in Figure 2(c). This effect is actually beneficial, since points near the edge tend to be noisy or even mixed pixels and thus should be considered outliers when estimating the center of the sphere.

Algorithm 1: 2.5D Morphological Opening

Input: Foreground binary image: $\mathbf{B} = \{b_i\} \in \{0, 1\}$, Original point cloud: $\mathbf{P} = \{p_i\} = \{x, y, z\}_i$, Focal length: f and a fixed radius w

Output: Updated binary image $\mathbf{B}$

`/* 2.5D Non-isotropic Erosion */`

1 $\mathbf{D} = \{d_i\} \leftarrow DistanceTransform(\mathbf{B})$;

2 **foreach** $\{b_i\}$ *that is a foreground pixel* **do**

3   **if** $d_i < \Delta r_i = w \cdot f / z_i$ **then**

4   Set b_i to background pixel;

`/* 2.5D Non-isotropic Dilation */`

5 Invert the binary image: $\tilde{\mathbf{B}} \leftarrow \neg\mathbf{B}$;

6 $[\tilde{\mathbf{D}}, \tilde{\mathbf{J}}] = \{\tilde{d_i}, \tilde{j_i}\} \leftarrow DistanceTransform(\tilde{\mathbf{B}})$;

7 **foreach** $\tilde{b_i}$ *indicates a foreground pixel* **do**

8   Its closest background point is $q_i = (x_{\tilde{j_i}}, y_{\tilde{j_i}}, z_{\tilde{j_i}})$;

9   **if** $\|p_i - q_i\|_2 < w$ **then**

10   Set $\tilde{b_i}$ to background pixel;

11 Restore the binary image: $\mathbf{B} \leftarrow \neg\tilde{\mathbf{B}}$;

3.1.3 Detecting the Sphere among the Remaining Clusters

The remaining foreground clusters are grouped using connected components as labeled in different colors in Figure 2(d). Due to noise and other foreground objects, such as a human operator or a stand, multiple clusters may exist in the foreground. Each cluster is passed into a linear sphere fitter by minimizing the following cost function:

$$(A, B, C, D) = \underset{A,B,C,D}{\arg\min} \sum_i (Ax_i + By_i + Cz_i + D + x_i^2 + y_i^2 + z_i^2) \tag{2}$$

which is equivalent to minimizing the algebraic error:

$$min \sum_i (x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2 - R^2 \tag{3}$$

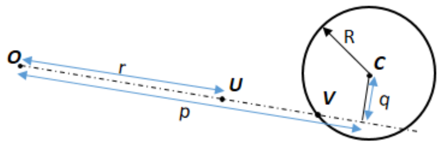
where $a = -A/2$, $b = -B/2$, $c = -C/2$ and $R = \sqrt{(A^2 + B^2 + C^2)/4 - D}$. Though this error metric is known to introduce significant bias towards smaller radii in sphere-fitting problems [1], it is useful in ranking all remaining clusters by how close they are to the target sphere. By forcing $R = R_{target}$, the cluster with minimum cost according to Equation 3 is selected. Moreover, the estimated $(a,\ b,\ c)$ also serves as the initial guess for the nonlinear refinement in the next step.

3.2. Estimating the Center of the Sphere from Noisy Points

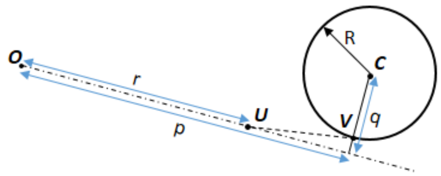
Since the radius of the spherical target is known, the natural extension is to refine the result with nonlinear geometric cost. Conventionally, the orthogonal distance of a point to the center of the sphere is optimized. This error metric, which assumes Gaussian noise along the radial direction, is shown in [13] to contradict with the fact the dominant error in range sensor measurements is along the ray direction. Franaszek *et al.* [7] further prove orthogonal distance results in two local minima when the sphere is only partially observed. A more natural error metric is the directional distance error proposed in [13] and [7]. It is piecewise differentiable and has a single minimum if started from within one radius of the true center. Interested readers are referred to [7] for more details on the derivation and proof. The important results are summarized as follows:

Given a sphere center at $\mathbf{C} = (X, Y, Z)$ and a point on the sphere $\mathbf{U} = (x, y, z)$, the cost of this relationship is defined as:

$$Err(\mathbf{C}, \mathbf{U}) = \begin{cases} (p - r) - \sqrt{R^2 - q^2}, & \text{if } q \leq R \\ \sqrt{(p - r)^2 + (q - R)^2}, & \text{if } q > R \end{cases} \tag{4}$$



(a)



(b)

Figure 3. Illustration of the directional cost. (a) When the ray OU intersects with the sphere ($q \leq R$), the cost is the distance between $\mathbf{U}$ and the intersection point $\mathbf{V}$. (b) When the ray does not intersect with the sphere ($q > R$), the cost is the distance between $\mathbf{U}$ and $\mathbf{V}$, the closest point on the sphere from the ray.

where r is the Euclidean distance from the sensor origin to the point $\mathbf{U}$, p is the length of the projection of the sphere center onto the ray OU, and q is the orthogonal distance from the sphere center to the ray OU. Written in equations:

$$r = \|\mathbf{U}\| = \sqrt{(}x^2 + y^2 + z^2) \tag{5}$$

$$p = \frac{\mathbf{U}}{\|\mathbf{U}\|} \cdot \mathbf{C} = \frac{1}{r}(Xx + Yy + Zz) \tag{6}$$

$$q = \frac{\mathbf{U}}{\|\mathbf{U}\|} \times \mathbf{C} \tag{7}$$

Therefore, when $q < R$ (which means the ray OU lands on the surface of the sphere), the cost $Err(\mathbf{C})$ is the distance from $\mathbf{U}$ to the surface (Figure 3(a)); when $q \geq R$ (which means the ray misses the sphere), the cost is the distance from $\mathbf{U}$ to the point on the sphere closest to the ray (Figure 3(b)).

Finding the center of the sphere given a point cloud $\mathcal{P}$ can be treated as solving the following nonlinear optimization problem:

$$\mathbf{C}_{(X,Y,Z)} = \arg\min_{\mathbf{C}} \sum_{\mathbf{U} \in \mathcal{P}} Err(\mathbf{C}, \mathbf{U}) \tag{8}$$

3.3. Solving for the Extrinsic Parameters

Once the 3D coordinate $\mathbf{C_i^j}$ of the j^{th} sphere center in the i^{th} sensor's frame is extracted for all i and j, each sensor's orientation $\mathbf{R}_i$ and position $\mathbf{t}_i$ with respect to a common coordinate frame can be calculated. For simplicity, set $i = 1$ as the reference frame, namely $\mathbf{R}_1 = \mathbf{I}$, $\mathbf{t}_1 = \mathbf{0}$. Therefore the rigid body transforms $[\mathbf{R}_i|\mathbf{t}_i]$ that aligns $\mathbf{C}_i^j, \forall j, i \neq 1$ with $\mathbf{C}_1^j, \forall j$ can be found by minimizing the following cost function:

$$[\mathbf{R}_i, \mathbf{t}_i] = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{j=2}^{M} \|(\mathbf{R} \cdot \mathbf{C}_i^j + \mathbf{t}) - \mathbf{C}_1^j\|^2 \tag{9}$$

where M is the total number of sphere images. This cost function has a closed-form linear solution based on SVD. The estimated true location of the centers $\tilde{\mathbf{C}}_i$ is then set to the centroid of the sphere centers transformed into this common coordinate frame.

$$\tilde{\mathbf{C}}^j = \frac{1}{N} \sum_{j=1}^{N} \mathbf{R}_i^T \cdot (\mathbf{C}_i^j - \mathbf{t}_j) \tag{10}$$

where N is the total number of sensors. The solution to Equation 9 and the output from Equation 10 are fed as initial guesses into a bundle adjustment solver which minimizes the Euclidean error in 3D space:

$$\left[\{\mathbf{R},\mathbf{t}\}_i, \hat{\mathbf{C}}^j\right] = \arg\min_{\mathbf{R},\mathbf{t},\hat{\mathbf{C}}} \sum_{i=1}^{N} \sum_{j=1}^{M} \|(\mathbf{R} \cdot \mathbf{C}_i^j + \mathbf{t}) - \hat{\mathbf{C}}^j\|^2 \tag{11}$$

3.4. Simultaneous Optimization of the Sphere Centers and Extrinsic Parameters

Though solving this bundle adjustment problem often yields a visually pleasing result, it is not the optimal solution in the Maximum Likelihood Estimation (MLE) sense. Given the true location of the spheres, a sensor's observation and its position/orientation are no longer independent. Bundle adjustment essentially allows $\hat{\mathbf{C}}^j$ to move freely, which is contradictory to the fact that the measurement error is not an isotropic 3D Gaussian distribution. Hence we propose a novel cost function that simultaneously refines the sphere detection and estimates the sensor extrinsic parameters.

$$Err_{MLE}(\{\mathbf{R},\mathbf{t}\}_i, \hat{\mathbf{C}}^j) = \sum_{i=1}^{N} \sum_{j=1}^{M} \Bigg(\|\tilde{\mathbf{C}}_i^j - \mathbf{C}_i^j\|^2 + \frac{\lambda_i}{K_i^j} \sum_{k=1}^{K_i^j} Err(\tilde{\mathbf{C}}_i^j, \mathbf{U}_k) \Bigg) \tag{12}$$

where k indexes over the K_i^j points on sphere j seen by sensor i, $Err(\mathbf{C}_i^j, \mathbf{U}_k)$ is the directional cost defined in Equation 4, λ_i is a user-specified factor to adjust how much each sensor's measurements can be trusted, and $\tilde{\mathbf{C}}_i^j$ is defined as the transformation of the j^{th} estimated center to the i^{th} sensor's frame:

$$\tilde{\mathbf{C}}_i^j = \mathbf{R}_i^T \cdot (\hat{\mathbf{C}}^j - \mathbf{t}_i) \tag{13}$$

The first half of Equation 12 is identical to the Euclidean distance minimized in the bundle adjustment; whereas the second half constrains every step taken in the parameter space. The MLE interpretation of Equation 12 is thus to simultaneously estimate the locations of the sphere that optimally explain the observed point clouds under the current sensor model as well as the sensors' poses with respect to a common reference frame.

It should be noted that Equation 12 is a high dimensional, non-convex problem. In practice, a regular bundle adjustment is first run without the regularization term. Points on the sphere from each frame are re-selected based on the projection of the estimated true location. The optimization is then run using the output from the bundle adjustment as the initial guess and is similarly solved using derivative-based methods.

4. Experimental Results

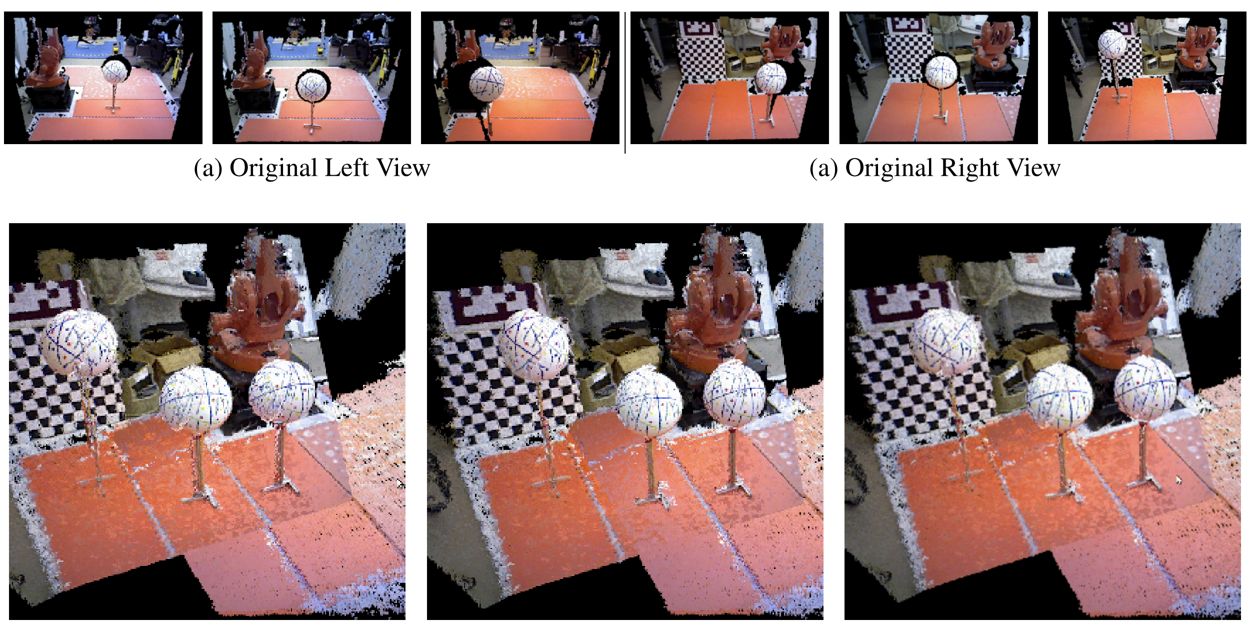
A qualitative example of the extrinsic calibration of two facing RGB-D cameras is shown in Figure 1. This type of setup is not easily handled by calibration methods that use planar targets. In addition to visually inspecting the fused point cloud, we also quantitatively evaluate our algorithm on two more common scenarios: one consists of two Xtion Pros and the other involves different range sensors. In both experiments, we use a spherical target of radius 20.4 cm mounted on a stand to avoid synchronization issues. The true locations of the sphere are measured with a high-precision Leica ScanStation2 laser scanner. The uncertainty of the laser scanner estimates of the sphere location and radius are on the order of 1 mm, whereas the sensors to be calibrated have accuracies on the order of 3 cm within the operating range.

To get a full picture of the accuracy of the algorithm, we compute the following errors for both experiments:

- The **3D Reprojection Error** measures the degree to which the sphere center estimates from each sensor agree with the corresponding estimates from the other sensors. The 3D reprojection error is computed as the RMS error between the estimated sphere centers ($\tilde{\mathbf{C}}_i^j$) and the corresponding detected sphere centers ($\mathbf{C}_i^j$). This error does not use the ground truth from the laser scanner, so it cannot measure certain types of errors (such as scaling).

- The **Global Registration Error** error measures the difference between the estimated sphere centers and the ground truth centers provided by the laser scanner. The global registration error is the RMS difference between all sphere center estimates ($\tilde{\mathbf{C}}_j$) and their corresponding ground truth positions from the laser scanner after optimally aligning them in a common coordinate reference frame.

- The **Individual Registration Error** measures the in-



(a) Original Left View (a) Original Right View

(b) (c) (d)

Figure 4. Fused view with transforms computed from the three different methods (3 out of 10 views overlaid) (b) Image corner method. The sphere are reasonably well aligned, but the sticks are slightly misaligned (c) Projected image corner method. All three spheres have some noticeable misalignment (d) Our method. The scene is noticeably better reconstructed. For example, the patterns on the spheres and sticks are clearly visible in the fused cloud.

Table 1. Results of the two Xtion Pros experiment. Units in cm.

Method	Reproj. Err.	Global Err.	Individual Err.	
			Xtion 1	Xtion 2
Image Corner	1.85 $\pm$1.48	2.30 $\pm$1.66	1.07 $\pm$0.54	1.88 $\pm$1.05
Proj. Corner	1.94 $\pm$1.51	2.41 $\pm$1.63		
Proposed	0.62 $\pm$0.42	1.47 $\pm$0.91		

trinsic calibration accuracy of a single sensor. It is identical to the 3D reprojection error, except that the estimated sphere centers for a single sensor ($\mathbf{C}_i^j$) are used instead of the globally estimated sphere centers ($\tilde{\mathbf{C}}_j$).

4.1. Calibration of Two RGB-D Cameras

Our first experiment is conducted with a pair of Asus Xtion Pro sensors, connected to two separate computers. We compare the two commonly used indirect calibration techniques, namely 1) using image corners only and 2) using image corners and their corresponding 3D points, with 3) our method using the spherical target. We denote the transforms found by the three methods $\{\mathbf{R}, \mathbf{t}\}_{1\cdots 3}$.

After the sensors are calibrated using the three methods, one way to evaluate the performance is to visually compare the fused point cloud as shown in Figure 4. Obtaining comparable quantitative results is more difficult because each algorithm optimizes a different objective function. The three error metrics are thus computed as follows:

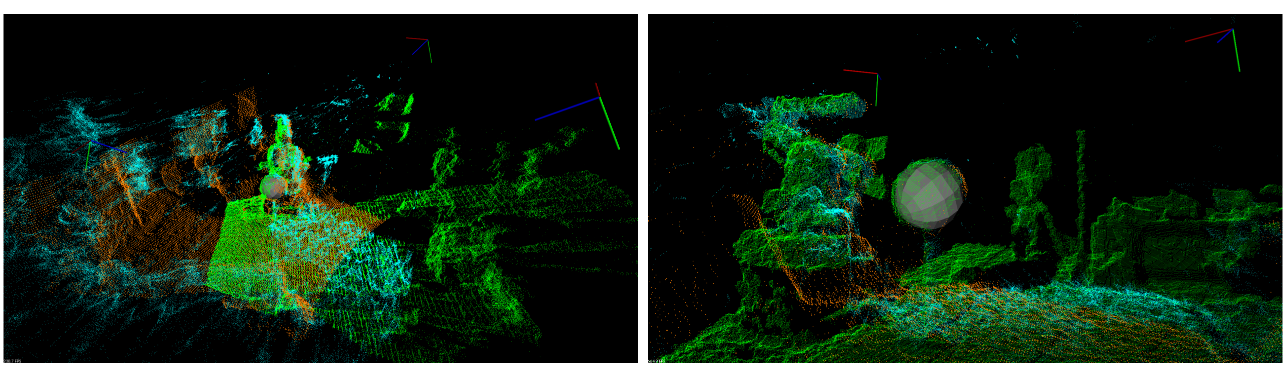
1. Place the spherical target at 10 locations spanning the overlapped view frustum. Collect point clouds with both sensors and scan the target with the laser scanner.
2. Each 3D point cloud of the spheres is automatically segmented using the same preprocessing algorithm and manually inspected to remove possible outliers.
3. The sphere is independently detected in each frame of both sensors.
4. The three error measures are computed as described above.

The quantitative results are summarized in Table 1. As expected, using back-projected corners yields the highest error, most likely caused by the noisy range measurement of individual points on the checkerboard surface. The effectiveness of the image corner method will quickly decrease as the angle between the two sensors goes above $90°$. Our method outperforms both indirect methods by driving down the global registration error close to the individual registration error.

4.2. Calibration of Multiple Heterogeneous Range Sensors

In the second experiment, we demonstrate our algorithm on an industrial work cell monitoring application, which consists of multiple sensors of different accuracies and resolutions. This setup consists of one Swiss Ranger 4000, one Tyzx stereo camera, and one Asus Xtion Pro. Since the sensors are mounted so widely apart, checkerboard-based calibration is infeasible for this experiment. The sphere is placed at 19 different locations within the area of interest. The same laser scanner is used to obtain the true locations of the spheres.

The three error statistics are summarized in Table 2, and the registration is visualized in Figure 5. Individual registration errors agree with each sensor's specified accuracy for the operating range. In particular, Asus Xtion Pro produces the largest error when the sphere is far away. The Swiss Ranger and Tyzx are mostly consistent throughout



(a) (b)

Figure 5. Fused point clouds from multiple sensors. SR4K is shown in orange, Xtion Pro is shown in green, and Tyzx is shown in cyan. (a) Bird's-eye view of the fused point clouds. The three sensors' poses are marked with colored axes. (b) Close-up view of the calibration target in front of a robot arm.

Table 2. Results of the multi-sensor experiment. Units in cm.

<table>
<tr><td rowspan="2"></td><td>Reproj.</td><td>Global</td><td colspan="3">Individual Err.</td></tr>
<tr><td>Error</td><td>Error</td><td>SR 4K</td><td>Xtion Pro</td><td>Tyzx</td></tr>
<tr><td>Bundle Adjustment</td><td>1.67 ±0.93</td><td>2.83 ±2.00</td><td rowspan="2">3.57
±1.87</td><td rowspan="2">4.31
±1.86</td><td rowspan="2">2.10
±1.30</td></tr>
<tr><td>MLE Optimization</td><td>1.05 ±0.64</td><td>2.74 ±0.92</td></tr>
</table>

the range. Though the difference of using the MLE optimization step is not always noticeable in the fused point cloud, the MLE optimization effectively reduces the reprojection error and also slightly improves the global registration errors.

5. Conclusions

We have proposed an easy-to-use and accurate technique for extrinsic calibration of generic range sensors using a spherical target. A spherical target offers the benefit that its image is invariant under rotation and thus can be imaged from any viewpoint. We demonstrated that the center of the sphere can be reliably extracted from noisy point cloud using the directional error metric, and the pose of each sensor can be accurately estimated. Although the current technique is designed to work exclusively with range sensors, the method could be extended to include image-based cameras if the sphere can be reliably detected in RGB or grayscale images.

6. Acknowledgments

This work was supported by the National Science Foundation under grant number #IIS-1208598. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The authors thank Dr. Reid Simmons for his constructive feedback, Christopher Okonkwo for his work on 3D morphological operations, and Chris Niessl for system support and his assistance with data collection.

References

[1] A. Al-Sharadqah and N. Chernov. Error analysis for circle fitting algorithms. *Electronic Journal of Statistics*, pages 886–911, 2009.

[2] P. Anderson-Sprecher. Intelligent monitoring of assembly operations. Master's thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2011.

[3] E. Auvinet, J. Meunier, and F. Multon. Multiple depth cameras calibration and body volume reconstruction for gait analysis. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 478–483, July 2012.

[4] J.-Y. Bouguet. Camera calibration toolbox for matlab, 2004.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(1):415–428, 2012.

[7] M. Franaszek, G. Cheok, K. Saidi, and C. Witzgall. Fitting spheres to range data from 3-d imaging systems. *Instrumentation and Measurement, IEEE Transactions on*, 58(10):3544–3553, Oct 2009.

[8] A. Geiger, F. Moosmann, O. Car, and B. Schuster. Automatic camera and range sensor calibration using a single shot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3936–3943, May 2012.

[9] L. Guan and M. Pollefeys. A unified approach to calibrate a network of camcorders and tof cameras. *In Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[10] Y. M. Kim, D. Chan, C. Theobalt, and S. Thrun. Design and calibration of a multi-view tof sensor fusion system. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–7, June 2008.

[11] S. Miller, A. Teichman, and S. Thrun. Unsupervised extrinsic calibration of depth sensors in dynamic scenes. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2695–2702, Nov 2013.

[12] P. Rybski, P. Anderson-Sprecher, D. Huber, C. Niessl, and R. Simmons. Sensor fusion for human safety in industrial workcells. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012.

[13] C. Witzgall, G. Cheok, and A. Kearsley. Recovering spheres from 3d point data. In *Applied Imagery and Pattern Recognition Workshop, 2006. AIPR 2006. 35th IEEE*, pages 8–8, Oct 2006.

[14] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000.