Towards realistic covariance estimation of ICP-based Kinect V1 scan matching: The 1D case
Martin Barczyk, Silvere Bonnabel

To cite this version:

HAL Id: hal-01695782
https://hal-mines-paristech.archives-ouvertes.fr/hal-01695782
Submitted on 29 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Towards Realistic Covariance Estimation of ICP-based Kinect V1 Scan Matching: the 1D Case

Martin Barczyk and Silvère Bonnabel

Abstract—The Iterative Closest Point (ICP) algorithm is a classical approach to obtaining relative pose estimates of a robot by scan matching successive point clouds captured by an onboard depth camera such as the Kinect V1, which has enjoyed tremendous popularity for indoor robotics due to its low cost and good performance. Because the sensed 3D point clouds are noticeably corrupted by noise, it is useful to associate a covariance matrix to the relative pose estimates, either for diagnostics or for fusing them with other onboard sensors by means of a probabilistic sensor fusion method such as the Extended Kalman Filter (EKF). In this paper, we review the sensing characteristics of the Kinect camera, then present a novel approach to estimating the covariance of pose estimates obtained from ICP-based scan matching of point clouds from this sensor. Our key observation is that the prevailing source of error for ICP registration of Kinect-measured point clouds is quantization noise rather than white noise. We then derive a closed-form formula which can be computed in real time onboard the robot’s hardware, for the case where only 1D translations are considered. Experimental testing against a ground truth provided by an optical motion capture system validates the effectiveness of our proposed method.

I. INTRODUCTION

Scan matching [1] is the process of computing the relative roto-translation between successive point clouds measured by a scanning sensor onboard a robot moving through a structured environment. The resulting estimates are then used to incrementally compute the global pose of the vehicle. A classical approach to scan matching is the Iterative Closest-Point (ICP) algorithm [2], [3]. A large number of implementations of the ICP algorithm have been developed over the years [4], [5], which vary in the details of how point pairs are selected, matched or rejected, and the choice of the alignment error cost function: point-to-point [2] or point-to-plane [3].

The resulting pose estimates are typically fused with other measurements such as wheel odometry, visual landmark detection and/or GPS using probabilistic filters such as the Extended Kalman Filter (EKF) [6], EKF Variants [7], particle filters [8], or optimization-based smoothing methods as in GraphSLAM [9]. In order to do this, we require a covariance matrix which quantifies the uncertainty associated to the pose estimated by scan matching.

Current methods to associate a covariance to the ICP estimates are either computationally costly, such as the one proposed in [10], or applications of the pioneering work by A. Censi [11], [12]. This author proposes an explicit formula based on assuming Gaussian white sensor noise. But this assumption does not match the noise characteristics of a Kinect, for which the formula provides greatly over-optimistic covariance estimates. Following our earlier work [13], [14], we seek to develop a real-time method to associate a realistic covariance to pose estimates obtained from scan matching point clouds from a low-cost Kinect V1. Note that even for different sensing technology such as LIDAR, remarkably small covariance estimates resulting from applying the formula of [12] have been recently reported in [15].

The contribution of the present paper is to present a novel approach to estimating covariance values for ICP-based scan matching of point clouds from a Kinect V1, which focuses on quantization noise rather than Gaussian white noise. We demonstrate its validity through experimental hardware testing. The proposed method is restricted to translations along a single axis, but the approach may be generalized to the full 6 DoF case. This is deferred to future work.

The remainder of this paper is structured as follows. Section II describes the Kinect V1 hardware and its noise characteristics. Section III reviews the covariance of estimates obtained from ICP-based scan matching and explains why quantization errors dominate over Gaussian white noise errors in the Kinect V1. Section IV introduces a novel approach to accounting for depth camera quantization effects when calculating the covariance of ICP-based scan matching estimates. Section V provides experimental hardware results which validate the proposed approach. Finally Section VI offers conclusions and lays out future work.

II. KINECT V1 CAMERA

The Kinect V1 depth camera was originally sold as a gaming peripheral for Microsoft’s XBox 360 console. The Kinect’s sensing technology is well described in [16]. This product has found tremendous popularity for indoor robotics applications due to its low cost and good sensing performance. The Kinect employs an infrared laser to project a speckle pattern ahead of itself, whose image is read back using an infrared camera offset from the laser projector. By correlating the acquired image with a stored reference image corresponding to a known distance, the unit computes a depth map of the scene which is employed to construct a 3D point cloud as shown in the next paragraph. The depth maps computed by the Kinect, corresponding to individual pixels of the IR camera image, are available at a rate of
30 Hz. The IR camera has a total angular field of view of 57° horizontally and 43° vertically.

Each depth map reported by the Kinect is provided as a vector of 11-bit unsigned integers (values between 0 and 2047), describing the normalized depth values of the 640 x 480 IR camera image pixels from the top-left corner and proceeding right then wrapping around one line down. Let \( u \in [0, 639] \) and \( v \in [0, 479] \) denote the coordinates of a given pixel, with \( (u, v) = (0, 0) \) at the top-left corner, with corresponding normalized depth value \( w \). Each triplet \( (u, v, w) \) is converted to a 3D point \( (x, y, z) \) in a right-down-forward axes frame using the following formulas:

\[
\begin{align*}
    d &= K_x (b_x - w) \\
    z &= \frac{f_x (l_B)}{d} \\
    x &= \frac{z(u - c_x)}{f_x} \\
    y &= \frac{z(v - c_y)}{f_y}
\end{align*}
\]

where \( d \) is the disparity, \( K_x \), \( b_x \) are the shift (un-normalization) scale and offset, respectively, \( f_x, f_y \) are the camera focal lengths along the horizontal and vertical image axes, respectively, \( l_B \) is the baseline distance between laser emitter and IR camera on the front of the Kinect, and \( (c_x, c_y) \) are the coordinates of the principal point in the image plane. The set of identified values of these parameters and their units for our camera are listed in Table I. These were obtained by running a Kinect camera calibration module built into the well-known Robot Operating System (ROS).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (px)</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_x )</td>
<td>595.2</td>
<td></td>
</tr>
<tr>
<td>( f_y )</td>
<td>595.2</td>
<td></td>
</tr>
<tr>
<td>( l_B )</td>
<td>0.074</td>
<td>m</td>
</tr>
<tr>
<td>( K_x )</td>
<td>0.125</td>
<td></td>
</tr>
<tr>
<td>( b_x )</td>
<td>1090.8</td>
<td></td>
</tr>
<tr>
<td>( c_x )</td>
<td>328.4</td>
<td>px</td>
</tr>
<tr>
<td>( c_y )</td>
<td>251.8</td>
<td>px</td>
</tr>
</tbody>
</table>

Examining logged Kinect data, we see \( w \) is reported as integers ranging from 0 to approximately 1028, corresponding to \( z = 0.32 \) m up to \( z = 5.61 \) m by the above calculation. Since points too near or too far from the camera may not be reliable due to noise effects, the device manufacturer recommends a practical ranging limit between 0.8 m and 4.0 m. In addition, values of \( w = 2047 \) are used to signal that a depth value could not be computed at the current pixel, which may be due to lighting interference, light absorption by dark surfaces, surface tilt, or other factors. These points are simply omitted from the 3D computations.

The Kinect’s noise characteristics were experimentally analyzed in [17] by scanning a flat surface located at varying distances from the camera, then performing a RANSAC-based plane fitting and computing the standard deviation of the residual errors. The resulting \( \sigma \) was found to vary quadratically with distance, from a few millimeters at depth \( z = 0.5 \) m up to 4 cm at depth \( z = 5 \) m. In order to limit noise, [17] recommended a depth range of 1.0 m to 3.0 m. Within this depth interval, the standard deviation of residuals is at worse \( \sigma \approx 1 \) cm, and can be viewed as a random noise effect.

A second source of error, also noted in [17], is the quantization caused by \( (u, v, w) \) being integer values. For instance at a depth of 2 m, inverting (1b) yields \( w = 915 \) as the normalized depth. However \( w = 915 \) and \( w = 916 \) values correspond respectively to \( z = 199.97 \) cm and \( z = 201.11 \) cm and thus a depth resolution error of \( \delta_z = 1.14 \) cm, where \( \delta_z \) is proportional to \( z \). By inspection of (1c) and (1d) the computed \( x \) and \( y \) values exhibit the same effect with \( \delta_x \) or \( \delta_y \) (the two have identical values due to \( f_x = f_y \)) in Table I also proportional to \( z \). A plot of the \( \delta \) values versus \( z \) is shown in Figure 1, which matches closely with the experimental results presented in [17].

![Quantization (resolution) errors for Kinect, 1 ≤ z ≤ 3 m range](image)

Quantization errors are very different from Gaussian white noise because they behave as a correlated noise, since the odds that nearby points fall into the same quanta are very large.

### III. Covariance of ICP-based Scan Matching

As we mathematically proved in [18], if two 3D point clouds \( \{a_i\} \in \mathbb{R}^3 \) and \( \{b_i\} \in \mathbb{R}^3 \) from successive scans are close to each other — either due to a fast scanning rate, or by pre-alignment using odometry-based dead reckoning — the covariance of their ICP-based scan matching pose estimate can be computed by modeling this algorithm as a linear least-squares problem, but only for the point-to-plane ICP variant.

Using the framework from [18], the linearized point-to-plane ICP cost function in the 3D case takes the form

\[
J(x) = \sum_i \|y_i - B_ix\|^2
\]

with \( y_i = n_i^T(a_i - b_i) \) and \( B_i = [-n_i \times] \) with \( (a_i, b_i) \) denoting the \( i \)th pair of corresponding points from successive scans and \( n_i \) the unit surface normal to point \( b_i \). Within this linearized context, \( x = [x_R \ x_T] \in \mathbb{R}^6 \)
parameterizes $X \in SE(3)$, the rigid-body transformation from $(a_i)$ to $(b_i)$, by

$$X = I + \begin{bmatrix} S(x_R) & x_T \\ 0 & 1 \end{bmatrix}$$

which is valid for $X$ close to identity and where $S(\cdot)$ is the $3 \times 3$ skew-symmetric matrix such that $S(a)b = a \times b, a, b \in \mathbb{R}^3$. The least-squares solution of (2) gives the estimated transformation

$$\hat{x} = \left( \sum_i B_i^T B_i \right)^{-1} \sum_i B_i^T y_i \quad \text{(3)}$$

where we define $A := \sum_i B_i^T B_i$. Remark $A^T = A$. Let $x^*$ denote the true transformation, and based on (2) assume the linear measurement model $y_i = B_i x^* + r_i$, where $r_i$ is the residual for the $i$th point pair. Solution (3) becomes

$$\hat{x} = A^{-1} \sum_i B_i^T (B_i x^* + r_i) = x^* + A^{-1} \sum_i B_i^T r_i \quad \text{(4)}$$

Substituting the definitions of $B_i$ and $y_i$ into $r_i = y_i - B_i x^*$ yields

$$r_i = n_i^T (a_i - b_i) + (a_i \times n_i)^T x_R^* + n_i^T x_T^*$$
$$= (x_R^* \times a_i)^T n_i + (x_T^* + (a_i - b_i))^T n_i$$
$$= [(x_R^* \times a_i) + x_T^* + a_i - b_i] \cdot n_i := w_i \cdot n_i$$

where $w_i$ represents the post-alignment error of the $i$th point pair due to sensor noise, and $n_i$ is its projection along the surface normal $n_i$. Assuming the ICP is an unbiased estimator such that $E(\hat{x}) = x^*$, the covariance of its estimates is given by

$$\text{cov}(\hat{x}) = E((\hat{x} - x^*)(\hat{x} - x^*)^T)$$
$$= E\left( \left( A^{-1} \sum_i B_i^T r_i \right) \left( A^{-1} \sum_i B_i^T r_i \right)^T \right)$$
$$\text{cov}(\hat{x}) = A^{-1} \sum_i \sum_j \left( B_i^T n_i^T E(w_i w_j^T) n_j B_j \right) A^{-1} \quad \text{(5)}$$

If we assume the post-alignment errors $w_i$ are independent and identically distributed as $w_i \sim \mathcal{N}(0, \sigma^2 I)$, then $E(w_i w_j^T) = E(w_i)E(w_j^T) = 0, i \neq j$ and the double sum in (5) reduces to a single sum ($n_i^T n_i = 1$ for unit normals):

$$\text{cov}(\hat{x}) = A^{-1} \sigma^2 \sum_i \left( B_i^T B_i \right) A^{-1} = \sigma^2 A^{-1}$$

This is precisely the covariance of a linear unbiased estimator using observations with additive Gaussian white noise [19, p. 85]. However, there is a catch: for a Kinect sensor with $N \approx 300,000$ points per cloud and an average standard deviation $\sigma \approx 1$ cm (as discussed in Section II), evaluating this last expression yields a covariance matrix with entries on the order of nanometers, a wildly over-optimistic result. This result is due to the independence assumption, which makes the estimator converge as $1/\sqrt{N}$ where $N$ is the (high) number of points. We conclude that the Kinect’s sensor noise cannot be adequately modeled as additive i.i.d. Gaussian white noise. The second type of noise — quantization, also discussed in Section II — is the dominant source of scan matching uncertainty, and needs to be accounted for in the calculation of $\hat{x}$.

IV. QUANTIZATION UNCERTAINTY

As established in Section III, the effect of quantization in Kinect point clouds needs to be accounted for when computing scan matching covariance through Equation (5). We now demonstrate how this is done through a progression of increasingly complex models.

A. Single Point Pair in 1D

We begin with the simplest possible case for scan matching: a single pair of points located along a 1D axis. Let $z_1$ and $z_2$ denote the true position of the two points, and $\Delta z = z_2 - z_1$ the true distance between them. We cannot directly measure $z_1$ or $z_2$; instead, we assume their measurements are quantized in uniform steps of $q > 0$. In this way, the true position $z_1$ of the first point can be represented as the (quantized) measurement $z_m$ plus a random variable $T$ which is uniformly distributed on the interval $[-q/2, q/2]$:

$$z_1 = z_m + T.$$  

We can thus represent $z_2$ as $z_1 + \Delta z = z_m + T + \Delta z$, where $\Delta z$ is the true scan matching result which cannot be directly obtained. We also define $\Delta y$ as the difference between the (quantized) measurements of $z_1$ and $z_2$, denoted as $Q(z_1)$ and $Q(z_2)$ respectively:

$$\Delta y = Q(z_m + T + \Delta z) - Q(z_m + T)$$
$$Q(z) = \left\lfloor \frac{z}{q} + \frac{1}{2} \right\rfloor$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Remark $\Delta y$ can be directly obtained by scan matching the measured points. Because $z_m$ is by a construction an integer multiple of $q$, the above simplifies to

$$\Delta y = Q(\Delta z + T) - Q(T)$$
$$= q \left( \left\lceil \frac{\Delta z + T}{q} + \frac{1}{2} \right\rceil - \left\lfloor \frac{T}{q} + \frac{1}{2} \right\rfloor \right)$$
$$= q \left\lceil \frac{\Delta z + T}{q} + \frac{1}{2} \right\rceil = Q(\Delta z + T)$$

Let $w := \Delta y - \Delta z$ denote the error in alignment of the two points by our scan-matching algorithm. Although we cannot obtain $w$ directly, we can calculate its variance using the probability density function associated to the random variable $T$:

$$\text{Var}_T(w) = \text{Var}_T(\Delta y - \Delta z)$$
$$= \text{Var}_T(\Delta y - \Delta z - E_T(\Delta y - \Delta z)) - \left( E_T(\Delta y - \Delta z) \right)^2$$
$$= E_T \left( (\Delta y - E_T(\Delta y))^2 \right)$$

(6)
Within expression (6), we calculate

\[ E_T(\Delta y) = E_T \left( q \left( \frac{\Delta z + T}{q} + \frac{1}{2} \right) \right) \]

\[ = q \int_{-q/2}^{q/2} \frac{1}{q} \left( \frac{\Delta z + t}{q} + \frac{1}{2} \right) dt \]

\[ = q \int_{\Delta z/q}^{\Delta z/q+1} [s] ds \quad \text{(by change of variables)} \]

\[ = q \left( \int_{\Delta z/q}^{\Delta z/q+1} [s] ds + \int_{\Delta z/q+1}^{\Delta z/q+1} [s] ds \right) \]

\[ = q \left( \left( \frac{\Delta z}{q} \right) + 1 - \frac{\Delta z}{q} \right) \left( \frac{\Delta z}{q} \right) \]

\[ + \left( \frac{\Delta z}{q} - \left[ \frac{\Delta z}{q} \right] \right) \left( \frac{\Delta z}{q} + 1 \right) \]

\[ = \Delta z \]

such that (6) becomes

\[ \text{Var}_T(w) = E_T \left( [\Delta y - \Delta z]^2 \right) \]

\[ = E_T((\Delta y)^2) - 2E_T(\Delta y)\Delta z + (\Delta z)^2 \]

\[ = E_T((\Delta y)^2) - (\Delta z)^2 \]

Within this expression we calculate

\[ E_T((\Delta y)^2) = E_T \left( q^2 \left( \frac{\Delta z + T}{q} + \frac{1}{2} \right)^2 \right) \]

\[ = q^2 \int_{-q/2}^{q/2} \frac{1}{q} \left( \frac{\Delta z + t}{q} + \frac{1}{2} \right)^2 dt \]

\[ = q^2 \int_{\Delta z/q}^{\Delta z/q+1} [s]^2 ds \quad \text{(by change of variables)} \]

\[ = q^2 \left( \int_{\Delta z/q}^{\Delta z/q+1} [s]^2 ds + \int_{\Delta z/q+1}^{\Delta z/q+1} [s]^2 ds \right) \]

\[ = q^2 \left( \left( \frac{\Delta z}{q} \right) + 1 - \frac{\Delta z}{q} \right) \left( \frac{\Delta z}{q} \right)^2 \]

\[ + \left( \frac{\Delta z}{q} - \left[ \frac{\Delta z}{q} \right] \right) \left( \frac{\Delta z}{q} + 1 \right) \]

\[ = q^2 \left( \frac{2\Delta z}{q} \left[ \frac{\Delta z}{q} \right] + \frac{\Delta z}{q} - \left[ \frac{\Delta z}{q} \right] - \left[ \frac{\Delta z}{q} \right]^2 \right) \]

and so the previous becomes

\[ \text{Var}_T(w) = E_T((\Delta y)^2) - (\Delta z)^2 \]

\[ = q^2 \left( \frac{\Delta z}{q} - \left[ \frac{\Delta z}{q} \right] \right)^2 + 2 \frac{\Delta z}{q} \left[ \frac{\Delta z}{q} \right] \]

\[ - \left[ \frac{\Delta z}{q} \right] - \left[ \frac{\Delta z}{q} \right]^2 \]

This expression can be rewritten as

\[ \text{Var}_T(w) = q^2 \left( \frac{\Delta z}{q} - \left[ \frac{\Delta z}{q} \right] \right) \left( 1 - \left( \frac{\Delta z}{q} - \left[ \frac{\Delta z}{q} \right] \right) \right) \]

Now, take the expectation over \( \Delta z \), i.e. over all the possible displacements. Observe that \( \forall \Delta z \in \mathbb{R} \), we have \( \Delta z/q - \left[ \Delta z/q \right] \in [0, 1] \). We then assume that \( Z := \Delta z/q - \left[ \Delta z/q \right] \) is a random variable uniformly distributed on \([0, 1]\), and calculate the expected value of the function \( \text{Var}_T(w) \) of random variable \( Z \) as

\[ E_Z(\text{Var}_T(w)) = \int_0^1 q^2 x(1-x) dx = \frac{q^2}{6} \]

This result is in accordance with the well established belief in the digital signal processing literature that quantization noise is uniformly distributed between plus and minus half a quanta, giving it zero mean and a variance of one-twelfth the square of a quanta. Here we are dealing with a difference of quantized values, leading to a discrepancy by a factor of two.

B. Several Point Pairs in 1D

Now consider the more complicated case of several point pairs, each located along a 1D axis. Assume we have a set of \( N \) pairs with true positions \( (z_{1i}, z_{2i}), 1 \leq i \leq N \), with a common distance \( \Delta z = z_{2i} - z_{1i} \) between all the point pairs. Physically, this setup represents two successive scans of a surface by a depth camera moving in a straight line towards it. Note we cannot directly know \( z_{1i}, z_{2i} \) nor \( \Delta z \) due to quantization of the measurements.

Analogously to Section IV-A, we write \( z_{1i} = z_{mi} + T_i \) where \( T_i \) is a random variable uniformly distributed on \([-q_i/2, q_i/2]\) where \( q_i \) is the quantization step size associated to the \( i \)-th point pair. We have \( z_{2i} = z_{1i} + \Delta z = z_{mi} + T_i + \Delta z \), and define \( \Delta y_i := Q_i(z_{2i}) - Q_i(z_{1i}) \) as the difference between the (quantized) measurements of the \( i \)-th point pair, and \( w_i := \Delta y_i - \Delta z \) as the error in its alignment. By Section IV-A we have \( E_T(\Delta y_i) = \Delta z \), such that \( E_T(w_i) = 0 \). Under this measurement model, for \( 1 \leq i \leq N \) and \( 1 \leq j \leq N \), if \( z_{1i} = z_{1j} \) then \( T_i = T_j \), otherwise \( T_i \) is independent from \( T_j \). Following Section IV-A, we show \( \Delta y_i = Q_i(\Delta z + T_i) \) such that \( w_i \) is a function of the random variable \( T_i \), \( w_j \) is a function of the random variable \( T_j \), and so \( E(w_i w_j) = E(w_i) E(w_j) = 0 \).

The uncertainty of \( w_i \) terms is quantified by their covariance:

\[ \text{cov}(w_i, w_j) = E((w_i - E(w_i))(w_j - E(w_j))) = E(w_i w_j) \]

By Section IV-A and the previous paragraph, we obtain

\[ E(w_i w_j) = \begin{cases} \frac{q_i^2}{6}, & i = j \\ 0, & i \neq j \end{cases} \]

This result can be used in conjunction with Equation (5) to compute the covariance of scan matching along the forward direction axis.
C. Multiple points in 3D case

The approach presented in Section IV-B forms the basis of scan matching in 3D. For 3 DoF motion consisting of pure translations along three orthogonal axes, the model can be extended by considering the quantizations $\delta_x$ and $\delta_y$ along the $x$ and $y$ camera axes, which vary identically with depth as discussed in Section II. For general 6 DoF motion, the calculations become longer due to the coupling between translation and rotation of the camera when scanning point clouds. These more complicated models are left as future work.

V. EXPERIMENTAL VALIDATION

A. Technical details

A picture of the experimental validation setup is shown in Figure 2. The wheeled rover is equipped with a forward-facing Kinect camera, supplying 3D point clouds at a 15 Hz rate. The robot is also equipped with high-resolution encoders on the left and right wheels, as well as a 2D scanning laser, although these sensors are not employed in the present case. The robot starts out stationary, then travels in a straight line with a constant velocity towards the end wall, where it stops. An Optitrack motion-capture system, whose cameras are mounted on tripods seen in Figure 2, employs optical markers fixed to the vehicle to provide sub-millimeter ground truth information, used to validate the results obtained from the onboard ICP-based scan matching algorithm.

We employ our own implementation of the point-to-plane ICP algorithm, which following the taxonomy in [4] uses random normal-space sampling of source points from both meshes, nearest-neighbour point matching with constant weighting, rejection of pairs based on excessive distance or containing points on mesh boundaries, and alternates matching pairs with minimizing the cost function by linearization. The surface normals $n_i$ are obtained using the PlanePCA method [20]. The input point clouds are decimated to a rate of 3 Hz for scan matching, meaning that the linearization assumptions in Section I are valid even without pre-alignment by odometry-based dead reckoning.

Since we are only considering 1D motions along the $z$-axis of the camera and body-fixed frame, we take the term $E(w_j w_j^T)$ in (5) as

$$E(w_j w_j^T) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_i^2/6 \end{bmatrix} \text{ if } i = j, \text{ else } = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where $q_i$ is taken as the depth quantization at the $z$ coordinate of the second point of the pair $(a_i, b_i)$. The value of $q_i$ is equal to $\delta_z$ from Section II and is computed algebraically. Only the square root of the bottom-right diagonal entry of the $6 \times 6$ covariance matrix of $\tilde{x} = [\tilde{x}_R \ \tilde{x}_T]$ is reported, which represents the standard deviation of the estimated translation along the $z$ axis as explained in Section III.

B. Results and Discussion

The results from the scan matching experiment described in Section V-A are summarized in Table II. A set of 18 scan matching pairs is presented, listed in order of decreasing (approximate) distance to the far wall seen in Figure 2. Each pair number gives the corresponding estimated forward displacement $\Delta z$ computed by the ICP, the true forward displacement $\Delta z$ measured by the optical motion capture system, and the calculated standard deviation $\sigma_z$ associated with the estimated displacement.

<table>
<thead>
<tr>
<th>Scan pair number</th>
<th>Distance [m]</th>
<th>$\Delta z$ (ICP) [cm]</th>
<th>$\Delta z$ (true) [cm]</th>
<th>$\sigma_z$ [cm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.8</td>
<td>-0.4</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>3.8</td>
<td>5.0</td>
<td>3.1</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>3.75</td>
<td>-0.2</td>
<td>10.9</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>3.6</td>
<td>10.7</td>
<td>9.8</td>
<td>0.3</td>
</tr>
<tr>
<td>5</td>
<td>3.4</td>
<td>9.1</td>
<td>8.5</td>
<td>0.3</td>
</tr>
<tr>
<td>6</td>
<td>3.25</td>
<td>9.7</td>
<td>9.6</td>
<td>0.3</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>9.9</td>
<td>10.1</td>
<td>0.2</td>
</tr>
<tr>
<td>8</td>
<td>2.75</td>
<td>9.9</td>
<td>10.0</td>
<td>0.2</td>
</tr>
<tr>
<td>9</td>
<td>2.5</td>
<td>9.9</td>
<td>9.8</td>
<td>0.3</td>
</tr>
<tr>
<td>10</td>
<td>2.3</td>
<td>10.2</td>
<td>10.1</td>
<td>0.2</td>
</tr>
<tr>
<td>11</td>
<td>2.15</td>
<td>10.1</td>
<td>9.8</td>
<td>0.2</td>
</tr>
<tr>
<td>12</td>
<td>1.9</td>
<td>10.1</td>
<td>9.9</td>
<td>0.1</td>
</tr>
<tr>
<td>13</td>
<td>1.7</td>
<td>10.2</td>
<td>9.9</td>
<td>0.1</td>
</tr>
<tr>
<td>14</td>
<td>1.5</td>
<td>10.1</td>
<td>9.7</td>
<td>0.1</td>
</tr>
<tr>
<td>15</td>
<td>1.3</td>
<td>10.3</td>
<td>10.2</td>
<td>0.1</td>
</tr>
<tr>
<td>16</td>
<td>1.15</td>
<td>8.9</td>
<td>8.9</td>
<td>0.1</td>
</tr>
<tr>
<td>17</td>
<td>1.1</td>
<td>3.2</td>
<td>2.7</td>
<td>0.03</td>
</tr>
<tr>
<td>18</td>
<td>1.05</td>
<td>0.0</td>
<td>0.0</td>
<td>0.03</td>
</tr>
</tbody>
</table>

We based on the data in Table II, we make the following observations. For the majority of the cases (pairs 4 to 16, inclusive), the ICP-estimated $\Delta z$ lies inside the three-sigma interval of the ground truth displacement. Pairs 1 and 18 represent the start and end phases of the experiment, respectively, where the robot is stationary and the pair of scans should overlap. Note the ICP incorrectly estimates a negative (backwards) displacement of 4 mm at point 1, while the point 2 and 3 estimates are clearly incorrect since they lie outside the three-sigma interval. These errors are caused by the large distance between the Kinect and the end wall, giving rise to a sparse point cloud — caused by regions of
non-computed depth values, c.f. Section II — and hence poor accuracy of \( n_i \) and convergence to a non-global minimum by the ICP algorithm. Point 17 is also affected by an incorrect convergence, which can be diagnosed by monitoring the values of the ICP’s cost function during successive iterations. As discussed in [18], convergence to a wrong (non-global) minimum by the ICP is a type of error which cannot be accounted for by our method, since it breaks the assumption of the ICP being an unbiased estimator.

Also note that the value of \( \sigma_2 \) tends to decrease as the camera approaches the far wall, which agrees with the Kinect’s depth measurement precision increasing at smaller distances as shown in Section II. Data points 1 to 3 do not follow this trend, but this is likely caused by the poor accuracy of \( n_i \) values along the far wall mentioned in the previous paragraph, which in turn affects the \( z \)-axis covariance result due to the projection \( \mathbf{n}_i E(\mathbf{w}_i \mathbf{w}_i^T) \mathbf{n}_i \) in (5).

VI. CONCLUSION

In this paper, we have introduced a novel approach to computing the covariance of ICP-based scan matching estimates of point clouds obtained from a Kinect V1 depth camera. This was then successfully validated in hardware through a 1D motion experiment, demonstrating the effectiveness of the proposed method.

As stated in Section IV-C, the present calculations only apply to translations along the depth axis of the camera. The proposed method now needs to be extended to the full six degree-of-freedom case. We are in the process of expanding the calculations to handle this case, as well as validating them through hardware experiments.

ACKNOWLEDGMENTS

We thank Emilien Flayac for his help in this project. The work reported in this paper was partially funded by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES


