Indoor Localization Algorithms for a Human-Operated Backpack System

George Chen, John Kua, Stephen Shum†, Nikhil Naikal, Matthew Carlberg, Avideh Zakhor
Video and Image Processing Lab, University of California, Berkeley
{gchen,jkua,nnaikal,carlberg,avz}@eecs.berkeley.edu, †sshum@csail.mit.edu

Abstract

Automated 3D modeling of building interiors is useful in applications such as virtual reality and entertainment. Using a human-operated backpack system equipped with 2D laser scanners and inertial measurement units, we develop four scan-matching-based algorithms to localize the backpack and compare their performance and tradeoffs. We present results for two datasets of a 30-meter-long indoor hallway and compare one of the best performing localization algorithms with a visual-odometry-based method. We find that our scan-matching-based approach results in comparable or higher accuracy.

1. Introduction

Three-dimensional modeling of indoor and outdoor environments has a variety of applications such as training and simulation for disaster management, virtual heritage conservation, and mapping of hazardous sites. Manual construction of these models can be time consuming, and as such, automated 3D site modeling has garnered much interest in recent years. Interior modeling in particular poses significant challenges, the primary one being indoor localization in the absence of GPS.

Localization has been studied by robotics and computer vision communities in the context of the simultaneous localization and mapping problem (SLAM) where a vehicle's location within an environment and a map of the environment are estimated simultaneously [13]. The vehicle is typically equipped with a combination of laser range scanners, cameras, and inertial measurement units (IMU's). Recent work in this area has shifted toward solving SLAM with six degrees of freedom (DOF) [1, 12], namely position and orientation. SLAM approaches with laser scanners often use scan matching algorithms such as Iterative Closest Point (ICP) [9] to align scans from two poses in order to recover the transformation between the poses. Meanwhile, advances in visual odometry algorithms have led to camera-based SLAM approaches [5, 12]. With a single camera, pose can be estimated only up to an unknown scale factor. This scale is generally determined using GPS waypoints, making it inapplicable to indoor environments unless objects of known size are placed in the scene. To resolve this scale ambiguity, stereo camera setups have gained popularity, as the extrinsic calibration between the cameras can be used to recover absolute translation parameters [11, 12].

Localizing a vehicle using scan matching, visual odometry, and wheel odometry can result in significant drifts in navigation estimates over time. The error becomes apparent when the vehicle encounters a previously visited location, at which point it has traversed a loop. However, the estimated trajectory from localization algorithms may not form a loop. Such inconsistencies can be remedied by detecting when these loop closure events occur and solving optimization problems to close the loop [4, 6, 7, 12].

In this paper, we develop localization algorithms for a human-operated backpack system equipped with laser scanners and IMU's in order to capture the 3D geometry of building interiors. Localizing the backpack over time is a key step for indoor modeling as it is allows us to place all collected laser scans into the same 3D coordinate frame, forming a point cloud of the environment traversed by the backpack operator. Geometry of the environment can then be generated from this 3D point cloud.

Our work most closely resembles the 6-DOF laser scan-matching-based graphical SLAM approach by Borrmann *et al*. [1]. Typically 6-DOF scan-matching-based SLAM approaches use 3D laser scanners on a wheeled robot. Our approach is different in that we use orthogonally positioned, lightweight 2D short-range laser scanners and an orientation-only IMU to resolve all six degrees of freedom. Furthermore, our sensors are not attached to a wheeled robot and are instead mounted on a human-operated backpack system. Our ultimate goal is to use our localization algorithms to create accurate 3D indoor models automatically. As such, localizing the backpack in a 2D plane is insufficient, and 6-DOF localization is needed in order to generate accurate 3D point clouds of the environment.

The architecture of our backpack system is described in Section 2. We present four scan-matching-based localization algorithms in Section 3 and assess their accuracies in

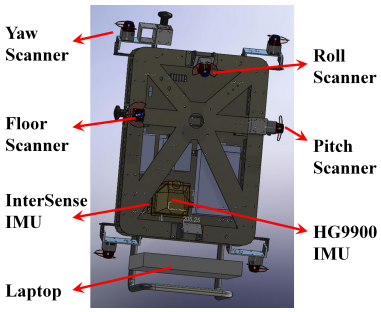


Figure 1: *CAD model of the backpack system.*

comparison with a visual-odometry-based method ICP-VO [10] in Section 4. Conclusions are in Section 5.

2. Architecture

We mount three 2D laser range scanners and two IMU's onto our backpack rig, which is carried by a human operator. Figure 1 shows the CAD model of our backpack system. We use 2D Hokuyo URG-04LX laser scanners to capture data at the rate of 10Hz. Each scanner has a range of approximately 5 meters and a field of view of 240 degrees. These scanners are positioned orthogonal to each other so as to recover all six pose parameters. One IMU is a strap down navigation-grade Honeywell HG9900 IMU, which combines three ring laser gyros with bias stability of less than 0.003 degrees/hour and three precision accelerometers with bias of less than 0.245 mm/sec^2. The HG9900 provides highly accurate measurements of all six DOF at 200Hz and serves as our ground truth. The other IMU, an InterSense InertiaCube3, provides orientation parameters at the rate of 180Hz. Our overall approach is to localize the backpack as a function of time using only the laser scanners and the InterSense IMU. In particular, we wish to estimate the backpack's pose at a rate of 10Hz, the same rate as the laser scanners.

We use a right-handed local coordinate system. With the backpack worn upright x is forward, y is leftward, and z is upward. Referring to Figure 1, the yaw scanner scans the x-y plane, the pitch scanner scans the x-z plane, and both the roll and floor scanners scan the y-z plane. Thus, the yaw scanner can resolve yaw rotations about the z axis, the pitch scanner about the y axis, and both the roll and floor scanners about the x axis.

3. Localization

We denote roll, pitch, and yaw with symbols ϕ, θ, and ψ respectively. For two backpack poses $\mathbf{p} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^\top$ and $\mathbf{p}' = \begin{bmatrix} x' & y' & z' & \phi' & \theta' & \psi' \end{bmatrix}^\top$, the 6-DOF transformation $\begin{bmatrix} t_x & t_y & t_z & \Delta\phi & \Delta\theta & \Delta\psi \end{bmatrix}^\top$ that takes pose $\mathbf{p}'$ and transforms it to pose $\mathbf{p}$ satisfies:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{R}(\phi, \theta, \psi) \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (1)$$

$$\mathbf{R}(\phi', \theta', \psi') = \mathbf{R}(\phi, \theta, \psi)\, \mathbf{R}(\Delta\phi, \Delta\theta, \Delta\psi) \quad (2)$$

where $\mathbf{R}(\cdot, \cdot, \cdot)$ is used throughout this paper to denote the direction cosine matrix for given roll, pitch, and yaw angles. t_x, t_y, and t_z refer to incremental position parameters, and $\Delta\phi$, $\Delta\theta$, and $\Delta\psi$ refer to rotation parameters. We present four algorithms in Sections 3.1, 3.2, 3.3, and 3.4 to estimate such 6-DOF transformations between backpack poses at different times. These transformation estimates are then refined to enforce global and local consistency as described in Sections 3.5 and 3.6.

3.1. Scan Matching with Laser Scanners

To localize the backpack, we use the yaw, pitch, and roll scanners shown in Figure 1. We use the roll scanner to resolve roll ϕ since we have empirically found it to resolve roll more accurately than the floor scanner. Assuming that the yaw scanner roughly scans the same plane over time, we can apply scan matching on successive laser scans from the yaw scanner and integrate the translations and rotations obtained from scan matching to recover x, y, and yaw ψ. Likewise, assuming that the roll scanner roughly scans the same plane over time, then we can use it to recover y, z, and roll ϕ. Finally, with a similar assumption, we can use the pitch scanner to recover x, z, and pitch θ.

These assumptions are somewhat stringent, and in particular, the assumption of scanning the same plane typically does not hold for the roll scanner at all. Specifically, when the backpack is moving forward, at each time step, the roll scanner sees a cross section of the indoor environment that is different from the cross section seen at the previous time step. Depending on the geometry of the environment, these two cross sections may be drastically different and hence, their laser scans cannot be aligned with scan matching. This problem, however, does not arise frequently in indoor environments with fairly simple geometry.

Another issue is the choice of scanners for estimating the different pose parameters since there are some redundancies. For example, it is possible to estimate pose parameter x from both the yaw and pitch scanners. However, to find a translation in the x direction, it is desirable for the two laser scans used in scan matching to have distinguishable features in the x direction. We have empirically found that distinguishable features in the x direction are more common in the yaw scanner's laser scans than in the pitch scanner's laser scans. Specifically, the ceiling and floor seen by the pitch scanner tend to lack distinguishable geometric features as compared to the walls scanned by the yaw scanner. Thus, we choose to use the yaw scanner rather than the pitch

scanner for estimating pose parameter x. Using similar reasoning, we choose to estimate parameters x, y, and yaw ψ from the yaw scanner, roll ϕ from the roll scanner, and z and pitch θ from the pitch scanner.

We use Censi's PLICP algorithm for scan matching [3]. Given 2D laser scans A and B, PLICP applies iterative refinement to compute 2D rotation R and translation t minimizing the error metric:

$$E(R,t) = \sum_i \left(n_i^\top (Rq_i + t - p_i)\right)^2, \tag{3}$$

where q_i is the i-th laser point in scan B, p_i is the interpolated point in scan A corresponding to point q_i, and n_i is the unit vector normal to the surface of scan A at point p_i. If scans A and B are from the same plane and PLICP finds enough correct corresponding laser points between the two scans, then R and t align scan B with scan A.

In addition to applying PLICP to estimate the transformation between two scans, we need to quantify the uncertainty of the resulting transformations in order to refine pose estimates later. As such, we use Censi's ICP covariance estimate [2] to determine a covariance matrix for each transformation determined by PLICP.

Algorithm 1 shows our proposed method for constructing a 6-DOF transformation between poses at two time instances and approximating the covariance for the transformation via a diagonal matrix. As shown, we combine scan matching results of the three scanners. Since we run PLICP three times to construct the transformation, we refer to this method as "3×ICP."

Algorithm 1: *3×ICP*

Input: laser scans for yaw, pitch, and roll scanners at times τ and τ'

Output: 6-DOF transformation T that takes the pose at time τ' to the pose at time τ; covariance matrix for T

1. Run PLICP to align the yaw scanner's laser scan from time τ' with the scan from time τ to estimate $t_x, t_y, \Delta\psi$; use Censi's method to estimate variances $\hat{\sigma}^2_{t_x}, \hat{\sigma}^2_{t_y}, \hat{\sigma}^2_{\Delta\psi}$ for $t_x, t_y, \Delta\psi$ respectively
2. Run PLICP to align the roll scanner's laser scan from time τ' with the scan from time τ to estimate $\Delta\phi$; use Censi's method to estimate variance $\hat{\sigma}^2_{\Delta\phi}$ for $\Delta\phi$
3. Run PLICP to align the pitch scanner's laser scan from time τ' with the scan from time τ to estimate $t_z, \Delta\theta$; use Censi's method to estimate variances $\hat{\sigma}^2_{t_z}, \hat{\sigma}^2_{\Delta\theta}$ for $t_z, \Delta\theta$ respectively
4. Construct 6-DOF transformation
$$T = \begin{bmatrix} t_x & t_y & t_z & \Delta\phi & \Delta\theta & \Delta\psi \end{bmatrix}^\top$$
and take its covariance to be the diagonal matrix with diagonal entries $\hat{\sigma}^2_{t_x}, \hat{\sigma}^2_{t_y}, \hat{\sigma}^2_{t_z}, \hat{\sigma}^2_{\Delta\phi}, \hat{\sigma}^2_{\Delta\theta}, \hat{\sigma}^2_{\Delta\psi}$

3.2. Integrating the InterSense IMU with Scan Matching

An alternative to the 3×ICP algorithm is to use the InterSense InertiaCube3 IMU, which directly estimates roll, pitch, and yaw. We have empirically found yaw values from the IMU, which are affected by local magnetic fields, to be less reliable than those obtained from running PLICP on scans from the yaw laser scanner. Thus, we alter our setup in the 3×ICP case above by obtaining roll and pitch values from the IMU instead. We then choose to omit the roll laser scanner since it is only used to estimate roll. The pitch scanner is still needed to estimate z even though the pitch values are obtained from the InterSense IMU. The main assumption here is that pitch and roll estimates from the InterSense IMU are more accurate than those obtained via scan matching with the pitch and roll scanners.

In summary, to construct a 6-DOF transformation from laser scans and IMU measurements from two time instances, we run PLICP twice, once on the yaw scanner's laser scans and once on the pitch scanner's laser scans. For the variance of the new roll and pitch estimates, we look up the IMU error specifications. We refer to the resulting method as "2×ICP+IMU." The pseudocode for this method is similar to that of 3×ICP.

3.3. Scan Matching with the Planar Floor Assumption

While integrating the InterSense IMU improves roll and pitch estimates, we have empirically found scan matching to drift significantly when estimating translation in the z direction regardless of which laser scanner is used. However, if we additionally assume that the floor is a plane, it is possible to improve the estimate for z, roll ϕ and pitch θ without using the IMU or scan matching. To make use of the planar floor assumption, we need to use scanners that sense a significant portion of the floor. As such, we opt to use the floor scanner rather than the roll scanner in Figure 1 for roll estimation, as the former more reliably senses the floor. For pitch and z estimation, we continue to use the same pitch laser scanner as before. We first discuss estimation of pitch and z using this planar floor assumption; estimating roll is similar to estimating pitch with an additional pre-processing step.

3.3.1 Pitch and z Estimation

With respect to the pitch laser scanner, we define a positive x-axis that points opposite the direction of motion, and a positive y-axis that points downwards toward the floor. A diagram of this setup is shown in Figure 2.

Given a scan from the pitch laser scanner, we begin by detecting the floor. Assuming that the backpack is worn upright on the human operator's back and that the floor is not

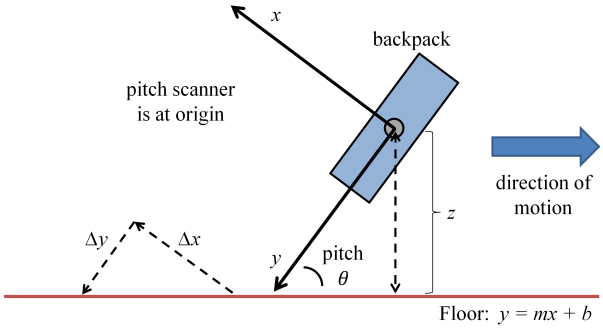


Figure 2: *Side view of the backpack system. The x and y axes show the coordinate system of the pitch laser scanner. The direction of motion assumes that the backpack is worn by a human operator moving forward.*

obstructed, reflective, or transparent, there exists a region of the laser scan, i.e. a range of angles and distances, whose points always correspond to the floor. As such, we use least squares to fit a line $y = mx + b$ to the region of interest, where x and y are in the coordinate frame of the pitch scanner. Under the planar floor assumption, this line lies on the plane of the floor. Then, upon inspecting Figure 2, the pitch angle θ is given by

$$\theta = \arctan\left(\frac{\Delta x}{\Delta y}\right) = \arctan\left(\frac{1}{m}\right) \tag{4}$$

and z is the perpendicular distance from the floor to the origin in the pitch scanner's coordinates. Furthermore, we can obtain z in the coordinate frame of any other sensor on the backpack by a simple trigonometric transformation, as the sensors are in a rigid configuration.

In case of occasional aberrant pitch θ and z estimates that arise when the floor estimation does not successfully detect the floor, we ignore θ and z estimates if including them would violate either of the following constraints:

$$\frac{d\theta}{dt} \leq 30 \text{ deg/sec}, \qquad \frac{dz}{dt} \leq 0.5 \text{ m/sec}. \tag{5}$$

Variance for pitch estimates is set to a constant determined by comparing the estimates against ground truth provided by the HG9900. Meanwhile, we need the variance of translation t_z in the z direction and not z itself. We describe how we determine t_z and estimate its variance in Section 3.3.3.

3.3.2 Roll Estimation

In estimating pitch and z, we have made the implicit assumption that roll is negligible. This is reasonable for our datasets since we have empirically found that human walking motion does not have significant roll. However, human pitch can vary widely and must be taken into account when estimating roll. Thus, to estimate roll, we use the same procedure as for estimating pitch except we prepend a pitch compensation step. This pre-processing step simply involves applying the pitch found via pitch estimation to the floor scanner's laser scan and projecting the scan points back into the floor scanner's scan plane.

3.3.3 Estimating t_z and Its Variance

Solving Equation (1) for t_z results in

$$t_z = \sec(\phi)\sec(\theta)\,\Delta z + \sec(\phi)\tan(\theta)\,t_x - \tan(\phi)\,t_y \tag{6}$$

where $\Delta z = z' - z$. In particular, t_z is written in terms of known values: ϕ, θ, z' and z are estimated via the roll and pitch estimation methods described previously, and t_x and t_y are obtained from applying the PLICP algorithm to the yaw scanner's laser scans.

To estimate the variance of t_z, we first write t_z as a function of $\Theta = \begin{bmatrix}\phi & \theta & \Delta z & t_x & t_y\end{bmatrix}$; a first-order Taylor approximation about our current estimate $\hat{\Theta}$ of Θ at time τ results in

$$t_z(\Theta) \approx t_z(\hat{\Theta}) + J(\Theta - \hat{\Theta}), \tag{7}$$

where $J = \frac{dt_z}{d\Theta}\big|_{\Theta=\hat{\Theta}}$. Then the variance of t_z is approximately $J\Sigma J^T$ where Σ is the covariance matrix of Θ. For simplicity, we approximate Σ with a diagonal matrix where each variance term along the diagonal is estimated empirically from comparison against ground truth or, in the case of t_x and t_y, are obtained from ICP covariance estimates.

The resulting method, described in Algorithm 2, uses PLICP once for the yaw scanner and uses the above methods to estimate roll, pitch, and t_z; we refer to this algorithm as "1×ICP+planar."

3.4. Integrating the IMU and Scan Matching with the Planar Floor Assumption

Lastly we use roll and pitch estimates from the IMU and determine t_z via the method described in Section 3.3.3. As before, we use scan matching for the yaw scanner's laser scans to determine t_x, t_y, and $\Delta\psi$. We refer to the resulting method as "1×ICP+IMU+planar." The pseudocode for this method is similar to that of 1×ICP+planar.

3.5. Trajectory Estimation and Global Refinement

Any of the transformation estimation algorithms described above can be used for backpack localization. By estimating the transformation between poses at consecutive times, we can compose these transformations to determine the entire trajectory the backpack traverses. However, since each transformation estimate is somewhat erroneous, the error in the computed trajectory can become large over time resulting in loop closure errors. For simplicity, we assume loop closures to have already been detected with known methods [4, 6]. We then enforce loop closure using the Tree-based netwORk Optimizer (TORO) by Grisetti *et al.* [7].

Algorithm 2: *1×ICP+planar*

Input: laser scans for yaw, pitch, and floor scanners at times τ and τ'; yaw ψ at time τ

Output: 6-DOF transformation T that takes the pose at time τ' to the pose at time τ; covariance matrix for T

Run PLICP to align the yaw scanner's laser scan from time τ' with the scan from time τ to estimate $t_x, t_y, \Delta\psi$; use Censi's method to estimate variances $\hat{\sigma}^2_{t_x}, \hat{\sigma}^2_{t_y}, \hat{\sigma}^2_{\Delta\psi}$ for $t_x, t_y, \Delta\psi$ respectively

Estimate roll ϕ, pitch θ, and z at time τ; and roll ϕ', pitch θ', and z' at time τ' via the methods described in Sections 3.3.1 and 3.3.2

Estimate t_z and its variance $\hat{\sigma}^2_{t_z}$ via the method described in Section 3.3.3

Extract $\Delta\phi$ and $\Delta\theta$ from the direction cosine matrix

$$[\mathbf{R}(\phi, \theta, \psi)]^{-1} \mathbf{R}(\phi', \theta', \psi + \Delta\psi)$$

Construct 6-DOF transformation

$$T = \begin{bmatrix} t_x & t_y & t_z & \Delta\phi & \Delta\theta & \Delta\psi \end{bmatrix}^\top$$

and take its covariance to be the diagonal matrix with diagonal entries $\hat{\sigma}^2_{t_x}, \hat{\sigma}^2_{t_y}, \hat{\sigma}^2_{t_z}, \hat{\sigma}^2_{\Delta\phi}, \hat{\sigma}^2_{\Delta\theta}, \hat{\sigma}^2_{\Delta\psi}$; values for $\hat{\sigma}^2_{\Delta\phi}, \hat{\sigma}^2_{\Delta\theta}$ are based on emperically determined error characteristics of our roll and pitch estimation methods

We supply TORO with a directed graph $\mathcal{G} = (V, \mathcal{E})$ with nodes V and edges $\mathcal{E}$ where each node X_i in V represents a 6-DOF pose, and each directed edge (i, j) in $\mathcal{E}$ represents a 6-DOF transformation $T_{i,j}$ that takes pose X_i to pose X_j. Each transformation $T_{i,j}$ needs to have a covariance matrix $\Sigma_{i,j}$ specifying its uncertainty. TORO refines pose estimates $\mathbf{X} = (X_1, X_2, \ldots, X_{|V|})$ using gradient descent to minimize the error metric

$$E(\mathbf{X}) = \sum_{(i,j)\in\mathcal{E}} \left(\widehat{T}_{i,j}(\mathbf{X}) - T_{i,j}\right)^\top \Sigma^{-1}_{i,j} \left(\widehat{T}_{i,j}(\mathbf{X}) - T_{i,j}\right) \tag{8}$$

where $T_{i,j}$ is obtained from any of the four transformation estimation methods described earlier, and $\widehat{T}_{i,j}(\mathbf{X})$ is the transformation between current pose estimates for X_i and X_j. To enforce loop closure, we supply a transformation in graph G that causes G to have a cycle. Algorithm 3 shows how to estimate and refine the backpack trajectory via TORO. This algorithm can use any of the four transformation estimation methods discussed earlier. In Section 4, we compare the accuracy of these four transformation estimation methods.

3.6. Additional Local Refinement

The above global refinement method adds an edge constraint to the directed graph input to TORO for each pair of poses that close a loop. It is also possible to add local edge constraints for use with TORO. In particular, a simple method is to consider a subset of poses consisting of every K-th pose and apply one of our four algorithms to estimate the transformation between these poses. As shown in Figure 3, these transformations add edge constraints to the directed graph input to TORO.

Algorithm 3: *Backpack localization*

Input: all laser scanner and InterSense IMU data for times $\tau_1, \tau_2, \ldots, \tau_N$; loop closures $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_M, \beta_M)$ i.e. pose α_i is roughly in the same location as the pose β_i

Output: 6-DOF pose estimates $X_1, X_2, \ldots, X_N$ for times $\tau_1, \tau_2, \ldots, \tau_N$ respectively

Set pose $X_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$

Set list of edges $\mathcal{E} = \emptyset$

for $i = 1, 2, \ldots, N-1$ **do**

  Compute the transformation $T_{i,i+1}$ between the pose at time τ_i and the pose at time τ_{i+1} and estimate the transformation's covariance $\Sigma_{i,i+1}$ using one of the four described transformation estimation methods

  Set pose X_{i+1} to be the result of applying $T_{i,i+1}$ to pose X_i

  Add edge $(i, i+1)$ to $\mathcal{E}$ with associated transformation $T_{i,i+1}$ and covariance $\Sigma_{i,i+1}$

end

for $i = 1, 2, \ldots, M$ **do**

  Compute the transformation T_{α_i,β_i} between the pose α_i and pose β_i and estimate the transformation's covariance $\Sigma_{\alpha_i,\beta_i}$ using one of the four described transformation estimation methods

  Add edge (α_i, β_i) to $\mathcal{E}$ with associated transformation T_{α_i,β_i} and covariance $\Sigma_{\alpha_i,\beta_i}$

end

Run TORO with an input directed graph $\mathcal{G} = (\{X_1, X_2, \ldots, X_N\}, \mathcal{E})$ to refine our pose estimates $X_1, X_2, \ldots, X_N$

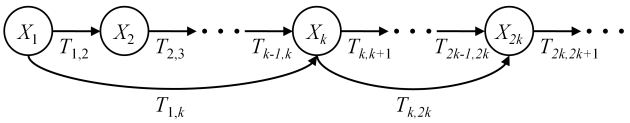


Figure 3: *Directed graph of poses X_i used as input to TORO. Edges correspond to transformations between poses. The bottom arcs are edges for additional local refinement.*

4. Results

We test our localization method in Algorithm 3 using each of our four transformation estimation methods on two

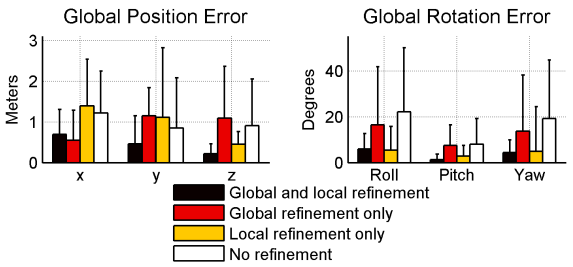


Figure 4: *Global RMS error characteristics for 3×ICP localization using various refinement techniques for dataset 1. Markers above each bar denote peak errors.*

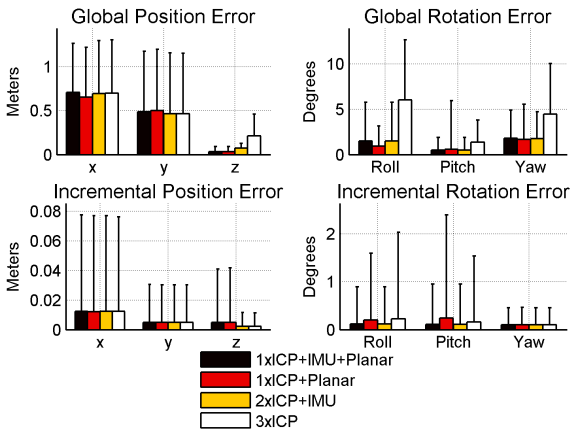


Figure 5: *Global and incremental RMS error characteristics using various transformation estimation methods for dataset 1. Markers above each bar denote peak errors.*

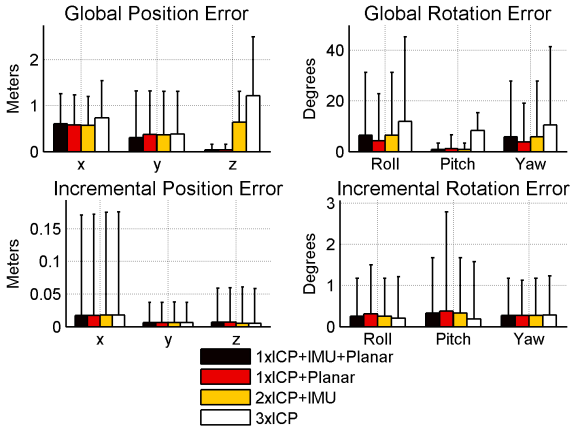


Figure 6: *Global and incremental RMS error characteristics using various transformation estimation methods for dataset 2. Markers above each bar denote peak errors.*

datasets from an interior hallway approximately 30 meters long in the north-south direction [10]. The two datasets were obtained by two different human operators with different gaits with each dataset corresponding to a 60-meter loop in the hallway. Error characteristics are determined by comparing estimated poses with the Honeywell HG9900 IMU, which serves as ground truth. Global position and orientation errors are computed in a frame where x is east, y is north, and z upward. Incremental position and rotation errors are computed in a local body frame where x is forward, y is leftward, and z is upward. Note that global errors result from accumulated local errors. As such, their magnitude is for the most part decoupled from the magnitude of local errors. In particular, local errors can either cancel each other out to result in lower global errors, or they can interact with each other in such a way so as to magnify global errors.

We first examine the effect of global and local refinement as described in Sections 3.5 and 3.6. Plots of global pose errors of 3×ICP localization for various refinement scenarios are shown in Figure 4. As seen, applying both global and local refinement results in the lowest error for most pose parameters. Local refinement by itself generally lowers rotation error compared to not using refinement but causes position errors in x and y to increase especially in the absence of global refinement. Plots for incremental pose errors are not shown since incremental pose errors are roughly the same across the methods.

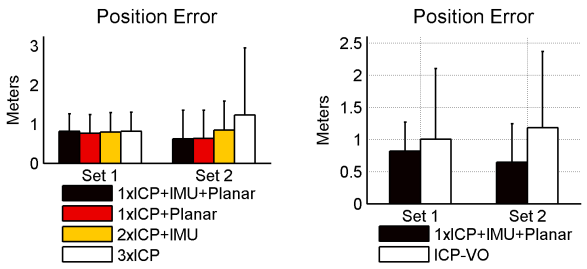
We do not include comparison plots for different refinement techniques for dataset 2 or for other transformation estimation methods since the trends are roughly the same as the 3×ICP dataset 1 case. For the remainder of the results presented, we use both global and local refinements.

Next, we compare the four transformation estimation techniques. Global and incremental pose errors for datasets 1 and 2 are shown in Figures 5 and 6 respectively. We find the IMU-based methods to have the best performance for incremental rotation. This is expected since they make use of the InterSense IMU, which provides more accurate roll and pitch measurements. In terms of global rotation, the four methods have comparable performance except for 3×ICP, which has much larger error.

Meanwhile, for incremental position errors, 3×ICP and 2×ICP+IMU have lowest z errors. For global position error, the "planar" methods have the best z performance since they inherently estimate z in the global frame at each time step. As for x and y global position errors, the four transformation estimation methods have similar performance. This is expected since the x and y directions are resolved by the yaw scanner, which is used across all four transformation estimation methods.

Overall, for incremental error, 2×ICP+IMU has the best performance, and for global error, the "planar" algorithms peform best. This trend for global error extends to the reconstructed path error shown in Figure 7(a) where we measure the average distance between estimated backpack positions and their ground truth positions at corresponding time instances: for dataset 1, the reconstructed path errors are roughly the same across algorithms, but for dataset 2, clearly the "planar" algorithms have lower error.

We now compare our scan-matching-based localization



(a) (b)

Figure 7: *Average reconstructed path error of: (a) various transformation estimation methods; (b) 1×ICP+IMU+planar localization vs. ICP-VO. Markers above each bar denote peak errors.*

algorithms with a visual odometry (VO) approach called ICP-VO [10]. As described in [10], ICP-VO estimates the 6-DOF transformation between two poses by applying 3×ICP to initalize rotations and then using VO to compute translations and refine rotations. ICP-VO further refines locally via local bundle adjustment but does not globally refine to enforce loop closure. To globally refine ICP-VO results, we detect loop closures using known methods [4, 6], estimate covariances for ICP-VO's transformations using methods described in Chaper 5 of [8], and input the transformations and their estimated covariances into TORO to obtain final refined pose estimates. Results reported for ICP-VO in this paper all include this global refinement.

Again, we use the HG9900 IMU as ground truth to derive error characteristics for comparison purposes. However, due to motion stereo baseline limitations, ICP-VO's estimated backpack poses are at a rate approximately 10 times lower than our scan-matching-based localization's rate of 10Hz. Therefore, we subsample our earlier scan-matching-based localization results so that the poses being compared correspond to approximately the same time instances as those of ICP-VO. Comparison plots of 1×ICP+IMU+planar localization vs. ICP-VO for datasets 1 and 2 are shown in Figures 8 and 9 respectively.

As seen, 1×ICP+IMU+planar has significantly lower global rotation and z errors than ICP-VO despite the former having comparable or lower corresponding RMS incremental errors. This can be explained in two ways: First, the z, pitch, and roll estimation techniques presented earlier inherently work in a global frame which ensures that global error is kept low and drift is prevented. Second, we have found the average error or the "bias" of ICP-VO's estimates for pose parameter z and all three rotation parameters to be three to four times larger than those of 1×ICP+IMU+planar. This bias in local error can result in a large global error.

Another observation across both datasets is that the RMS and peak incremental errors in x, or the direction of travel,

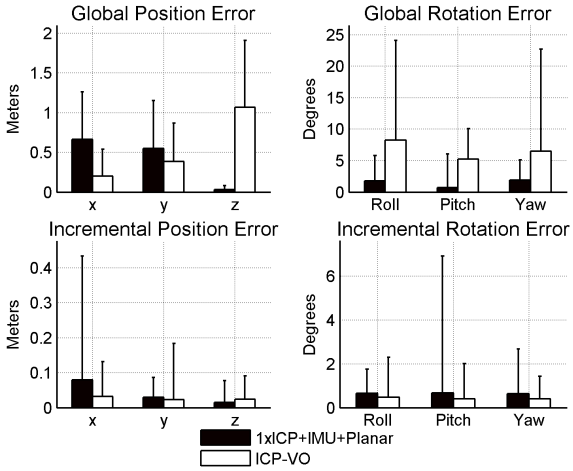


Figure 8: *Global and incremental RMS error characteristics of 1×ICP+IMU+planar localization and ICP-VO for dataset 1. Markers above each bar denote peak errors.*

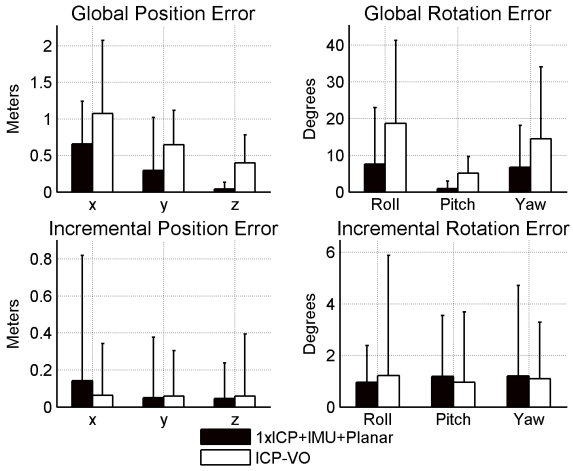


Figure 9: *Global and incremental RMS error characteristics of 1×ICP+IMU+planar localization and ICP-VO for dataset 2. Markers above each bar denote peak errors.*

is significantly higher for 1×ICP+IMU+planar. Since the ICP-VO results are obtained with a camera facing perpendicular to the direction of travel, this suggests that using such a camera setup can significantly reduce incremental localization error in the direction of travel compared to using the yaw scanner setup.

Finally, Figure 7(b) compares the reconstructed path error as compared to ground truth for ICP-VO and 1×ICP+IMU+planar. As seen, the latter outperforms the former in both RMS and peak errors. This is expected since the latter has superior overall global error performance than the latter.

We plot the resulting 1×ICP+IMU+planar estimated path vs. the ground truth path in Figure 10; using the estimated backpack poses, we place the laser points collected by the floor scanner into the same 3D space and superim-

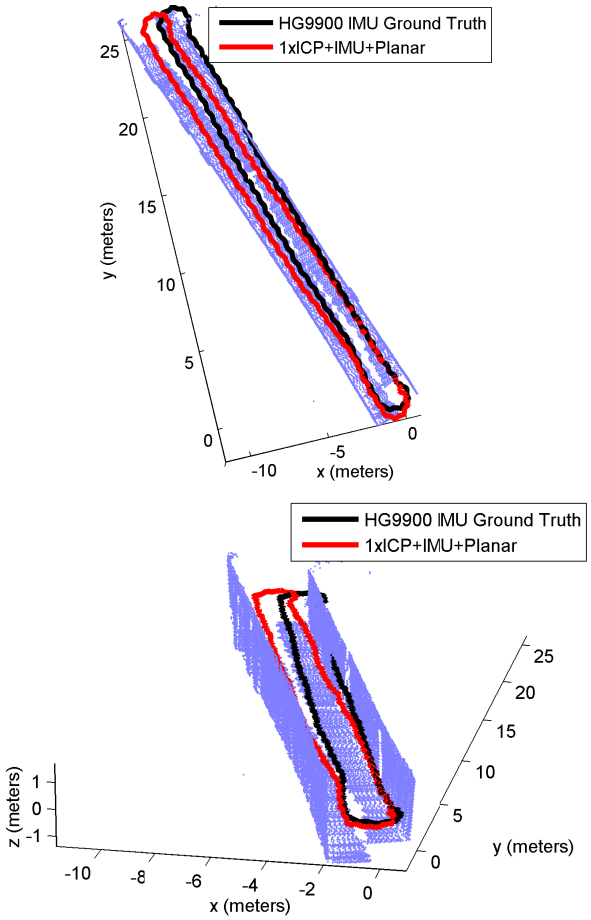


Figure 10: *Two views of the 3D laser point cloud using 1×ICP+IMU+planar localization results for dataset 1. The ground truth path is shown for comparison. Points from the floor scanner are superimposed in blue.*

pose all the laser points over the estimated path. As seen, there is about one degree of yaw difference between the ground truth and recovered path. The captured walls of the hallway are easily detected in the 3D point cloud.

5. Conclusions

We have presented a number of algorithms for scan-matching-based localization using portable 2D laser scanners. The different methods may be useful under different circumstances. For example, in environments with planar floors, the 1×ICP+planar or 1×ICP+IMU+planar methods are most accurate. Without the planar floor assumption, the 2×ICP method can be used. Comparing 1×ICP+IMU+planar localization with ICP-VO with global refinement, we find that the two methods have comparable incremental localization error performance while the former has better global error performance.

Even though we have not optimized the presented algorithms for speed, generally speaking ICP-VO is likely to be significantly more computation-intensive than the scan-matching-based localization algorithms in this paper. Another advantage of the scan-matching-based algorithms is the higher rate at which pose can be estimated, i.e. the scan rate of the laser scanners. For ICP-VO, the rate for pose estimates is dictated by the baseline selection algorithm used in motion stereo and, as such, is considerably lower than the scan rate of the laser scanners.

References

[1] D. Borrmann, J. Elseberg, K. Lingemann, A. Nuchter, and J. Hertzberg. Globally consistent 3d mapping with scan matching. *Robotics and Autonomous Systems*, 56:130–142, 2008. 1

[2] A. Censi. An accurate closed-form estimate of icp's covariance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007. 3

[3] A. Censi. An icp variant using a point-to-line metric. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008. 3

[4] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007. 1, 4, 7

[5] A. J. Davison and N. D. Molton. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, 2007. Member-Reid, Ian D. and Member-Stasse, Olivier. 1

[6] K. Granstrom, J. Callmer, F. Ramos, and J. Nieto. Learning to detect loop closure from range data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009. 1, 4, 7

[7] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3d. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007. 1, 4

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 7

[9] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1994. 1

[10] N. Naikal, J. Kua, G. Chen, and A. Zakhor. Image augmented laser scan matching for indoor dead reckoning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009. 2, 6, 7

[11] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 1

[12] V. Pradeep, G. Medioni, and J. Weiland. Visual loop closing using multi-resolution sift grids in metric-topological slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[13] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005. 1