Automatic Loop Closure Detection Using Multiple Cameras for 3D Indoor Localization

John Kua, Nicholas Corso, and Avideh Zakhor

Video and Image Processing Lab, University of California, Berkeley, Berkeley, CA 94720

ABSTRACT

Automated 3D modeling of building interiors is useful in applications such as virtual reality and environment mapping. We have developed a human operated backpack data acquisition system equipped with a variety of sensors such as cameras, laser scanners, and orientation measurement sensors to generate 3D models of building interiors, including uneven surfaces and stairwells. An important intermediate step in any 3D modeling system, including ours, is accurate 6 degrees of freedom localization over time. In this paper, we propose two approaches to improve localization accuracy over our previously proposed methods. First, we develop an adaptive localization algorithm which takes advantage of the environment's floor planarity whenever possible. Secondly, we show that by including all the loop closures resulting from two cameras facing away from each other, it is possible to reduce localization error in scenarios where parts of the acquisition path is retraced. We experimentally characterize the performance gains due to both schemes.

Keywords: indoor localization, loop closure detection, indoor modeling

1. INTRODUCTION

In recent years, three-dimensional modeling has attracted much interest due to its wide range of applications such as virtual reality, disaster management, virtual heritage conservation, and mapping of potentially hazardous sites. Manual construction of these models is labor intensive and time consuming; as such, methods for automated 3D site modeling have garnered much interest.

An important component of any 3D modeling system is localization of the data acquisition system over time and space. Localization has been studied by the robotics and computer vision communities in the context of the simultaneous localization and mapping problem (SLAM). Recently much work has been done toward solving the SLAM problem with six degrees of freedom (DOF),[1–3] i.e. position and orientation. SLAM approaches with laser scanners typically rely on scan matching algorithms such as Iterative Closest Point (ICP)[4] to align scans from two poses in order to recover the transformation. In addition, recent advances in visual odometry algorithms have led to camera-based SLAM approaches.[2,5]

Localization in indoor environments is particularly challenging since GPS is unavailable inside buildings. In addition, 3D modeling of complex environments such as stairwells precludes the use of wheeled acquisition systems. To overcome this, a human operated backpack system equipped with a number of laser scanners, cameras, and an orientation measurement system has been developed to both localize the system and to construct geometry and texture for 3D indoor modeling.[6,7] In previous work, we proposed a set of localization algorithms for recovering all 6 DOF over time and characterized its accuracy over a 60 meter loop on a 30 meter hallway using manually detected loop closure (LC) events.[6] In doing so, we empirically found that localization error is significantly reduced in situations where (a) the floor is planar, and (b) localization algorithms are designed to take advantage of the planarity. While such a localization algorithm is inapplicable to scenarios with stairwells or uneven surfaces, it can be applied to portions of the data acquisition path in which the planarity assumption

Further author information: (Send correspondence to Avideh Zakhor)
E-mail: {jkua, ncorso, avz}@eecs.berkeley.edu
Project website: http://www-video.eecs.berkeley.edu/research/indoor

This material is based upon work supported by, or in part by, the U. S.
Army Research Laboratory and the U. S.
Army Research Office under contract/grant number W911NF1110088.

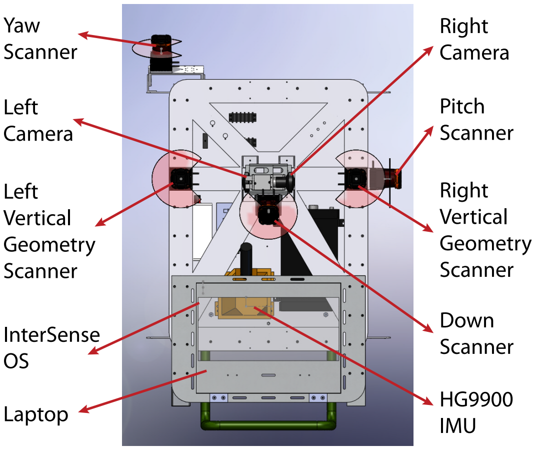


Figure 1. CAD model of the backpack system.

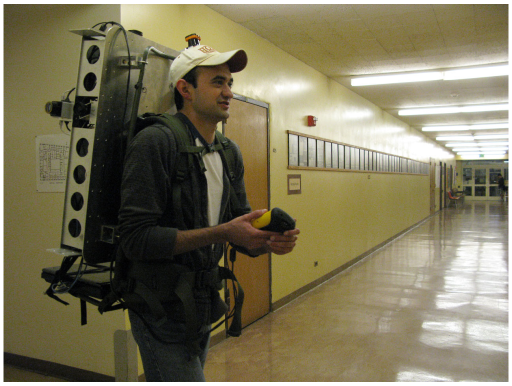


Figure 2. Photo of the backpack system in action.

does hold true. Thus, the challenge lies in classifying the acquisition path into planar and non-planar segments and to apply the appropriate localization algorithm to each portion. In this paper we develop such an adaptive localization algorithm and show that it can improve localization error in complex mixed environments made of both planar floors, e.g. hallways, and non-planar floors, e.g. staircases.

Due to various process errors and sensor biases, localization based on any combination of scan matching, visual odometry, and wheel odometry can result in significant drifts in navigation estimates over time. Often this error becomes apparent when the acquisition system visits a landmark or traverses a loop. In the case of revisiting a previous location, the estimated trajectory from a localization algorithm may not form a perfect loop. This type of inconsistency can be remedied by detecting LC events and solving an optimization problem to reduce the error.[2,8–10]

In general, finding LC events is a non-trivial task; accumulated localization error causes naïve detection schemes to miss them due to large errors in position estimates. Image data can be used to detect LCs independent of the current position estimate. Recently, we proposed a two step algorithm for automatic image based LC detection from a single camera for an indoor modeling system;[7] the first step, which is based on FAB-MAP,[8] results in a rank ordered list of candidate image pairs. The list of image pairs is processed in the second step using keypoint matching[11] to filter out the erroneous candidates.

For image based LC detection using one camera, both the position and orientation of the camera, and hence the acquisition system, need to be similar during a revisit with a given location. However, with two side-looking cameras pointing 180° away from each other, it is conceivable to detect LC events *across* two cameras, provided during the revisit the system is about 180° away from its initial yaw orientation. This condition is satisfied in situations where a system traverses up and down a hallway or a stairwell. Therefore in practice, it is possible to detect a large number of LCs by matching images from one side-looking camera while traversing up the hallway or stairwell, to images from the opposing side-looking camera while traveling in the opposite direction: the larger the amount of overlap in the retraced path, the larger the number of such LCs. In this paper, we apply the automatic LC detection algorithm[7] to images from two opposite facing side cameras on a human operated backpack system in order to detect such LCs. In doing so, we show that the increased number of LCs in these scenarios results in reduction in 6 DOF localization error.

The outline of the paper is as follows. The architecture and conventions of our backpack system is described in Section 2. In Section 3, the adaptive algorithm for mixed paths with both planar and non-planar floors is discussed and evaluated. In Section 4, we describe image based LCs for the two side-looking cameras and characterize its localization error. The conclusions are in Section 5.

2. ARCHITECTURE AND CONVENTIONS

We mount five 2D laser range scanners, two cameras, an orientation sensor, and an IMU onto a backpack system, which is carried by a human operator. Figure 1 shows the CAD model of such a system. The laser scanners are

40Hz Hokuyo UTM-30LX 2D laser scanners with a 30-meter range and a 270° field of view. These scanners are mounted orthogonally to one another. The two cameras are Point Grey Grasshopper GRAS-50S5C units equipped with fisheye lenses, resulting in a 180° field of view. The IMU, a Honeywell HG9900, is a strap-down navigation-grade sensor which combines three ring laser gyros with bias stability of less than 0.003°/hour and three precision accelerometers with bias of less than 0.245mm/sec^2. The HG9900 provides highly accurate measurements of all 6 DOF at 200Hz and thus serves as our ground truth. The orientation sensor (OS), an InterSense InertiaCube3, provides orientation parameters at a rate of 180Hz. As seen later, only the yaw and pitch scanners, and the InterSense OS are used to localize the backpack in all the localization algorithms discussed in this paper. In particular, the position of the system is estimated at a rate of 10Hz. The left and right cameras are used to detect LC events while the side-looking vertical geometry scanners are only used to construct geometry.

Throughout this paper we assume a right-handed coordinate system. With the backpack system worn upright the x axis is forward, the y axis is leftward, and the z axis is upward. As shown in Figure 1, the yaw scanner scans the x-y plane, the pitch scanner scans the x-z plane, and the vertical geometry scanners scan the y-z plane. Thus, the yaw scanner can resolve yaw rotations about the z axis.

3. ADAPTIVE LOCALIZATION

In this section, we begin by reviewing two of our previously proposed 6 DOF localization algorithms.[6] These algorithms combine scan matches from orthogonal scanners and OS data to recover the 6 DOF transformation from the pose at time t_1 to the pose at time t_2. Integrating the recovered transformations, an estimated trajectory for the backpack can be obtained. Due to process errors and sensor biases, the estimated transformation is somewhat erroneous as the error grows large over long trajectories. Once LC events are known, they are enforced using a nonlinear optimization technique, the Tree-based netwORk Optimizer (TORO), to reduce localization error.[6, 10] In Section 3.1, we review the 2×ICP+OS method[6] for transformation recovery without a priori knowledge about the scene environment. This algorithm is applicable to paths with planar or non-planar floors. In Section 3.2, we review the localization algorithm 1×ICP+OS+Planar,[6] which is based on the floor planarity assumption. In Section 3.3, we introduce a new algorithm which adaptively switches between the two localization algorithms by automatically segmenting the traversed path into planar and non-planar segments. Such an algorithm is useful in mixed environments with both planar and non-planar floors such as staircases. Specifically, it enjoys the low localization error of 1×ICP+OS+Planar in planar floor regions, while simultaneously able to handle non-planar floor regions the same way as 2×ICP+OS does. Thus, it can be thought of as a hybrid between 2×ICP+OS and 1×ICP+OS+Planar. Results for this adaptive localization algorithm are presented in Section 3.4.

3.1 Overview of 2×ICP+OS Localization

Given the input laser scans and OS data at t_1 and t_2, the 2×ICP+OS algorithm provides an estimate of the linear 6 DOF transformation from the pose at t_1 to the pose at t_2 by running ICP twice, once on the yaw scanner and once on the pitch scanner. The 6 DOF localization problem can be approximately decoupled into a series of 2D scan matching problems.[6] Only the yaw and pitch scanners are needed to recover the translation between successive poses. Specifically, we use Censi's PLICP algorithm for scan matching[12] on the yaw scanner data to obtain the change in x, y, and yaw, namely t_x, t_y, and $\Delta\psi$. Since accurate covariance measurements are needed for TORO optimization, we employ Censi's method[13] to estimate the covariances Σ_{t_x}, Σ_{t_y}, and $\Sigma_{\Delta\psi}$. Similarly, by applying scan matching to the pitch scanner data, the estimate of change in z, i.e. t_z, as well as its covariance measure, i.e. Σ_{t_z}, are obtained. The InterSense OS provides absolute orientation estimates, pitch and roll, which allows for construction of the incremental orientation from time t_1 to t_2. The covariance measures for the OS's orientation parameters are taken from the product specifications.

3.2 Overview of 1×ICP+OS+Planar Localization

With a priori knowledge that the backpack is moving through an environment with a planar floor, the 1×ICP+OS+Planar algorithm allows for a much more accurate estimate of the transformation between successive poses. Similar to the 2×ICP+OS algorithm, performing PLICP and Censi's method on the yaw scanner data allows for recovery of t_x, t_y, and $\Delta\psi$ as well as their covariance measures. In addition, the InterSense OS provides

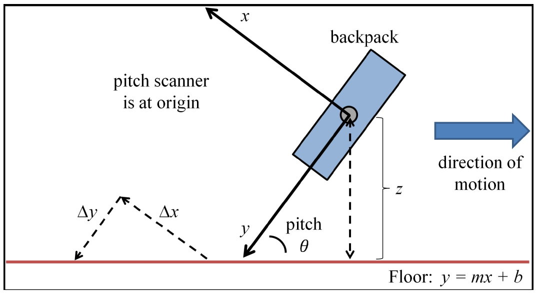


Figure 3. Overview of the floor planarity assumption; the axis are shown in the coordinate system of the pitch laser scanner; it is assumed that the backpack is worn upright by a human operator.

pitch and roll. However, as shown in Figure 3, if a line can be fit to the scan samples on the floor, absolute z can be estimated at every time instant. Successive estimates for z allow for the construction of t_z, with the covariance estimated using the method described in Ref. 6. This estimate of z has been empirically shown to be more accurate than the one obtained via 2×ICP+OS in regions where the planarity assumption holds. This is because whereas in 2×ICP+OS the change in z, i.e. t_z, is estimated via scan matching from one instant in time to another, in 1×ICP+OS+Planar, absolute z values are estimated from scratch at each time instant, and hence do not get a chance to drift over time.

3.3 Adaptive Localization Method

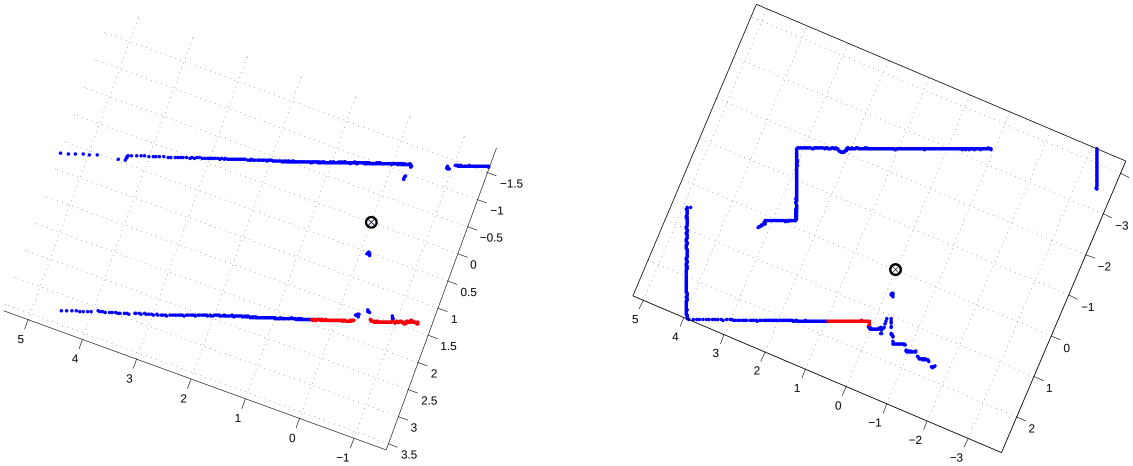
The main drawback of the 1×ICP+OS+Planar algorithm is that a priori knowledge of a planar floor is required for the entire acquisition path. The planarity assumption breaks down in more complex, mixed environments. In order to adaptively switch between 1×ICP+OS+Planar and 2×ICP+OS, the data must be segmented into planar and non-planar regions.

To identify planar portions of the acquisition path, we need to locate range data that coincides with the floor. We begin by detecting clusters of points in the pitch scan which form lines, limiting the search to the angular region of 30° on either side of the gravity downvector, as measured by the InterSense OS. Taking advantage of the ordered nature of the laser scan capture, we process the points sequentially, adding points to a cluster as long as the point satisfies two criteria: (1) the distance between the last point and the new point is less than a distance threshold; (2) the point does not increase the average residual error of the best fit line to the cluster above a specified error threshold. These thresholds should be chosen to be smaller than the minimum expected floor feature size, e.g. a stair step. We choose the distance threshold to be 5cm and the error threshold to be 10cm. When a point fails either or both criteria, a new cluster is started with the new point.

As lines corresponding to the floor should be perpendicular to the gravity downvector, the detected lines are then filtered by comparing their normals to the gravity downvector. Lines with normals within 10° of the downvector are declared as candidate floor lines. The fit error for the entire search region of the pitch scan is computed to each of the candidate lines and the line with the most points below the error threshold is chosen as the final floor plane.

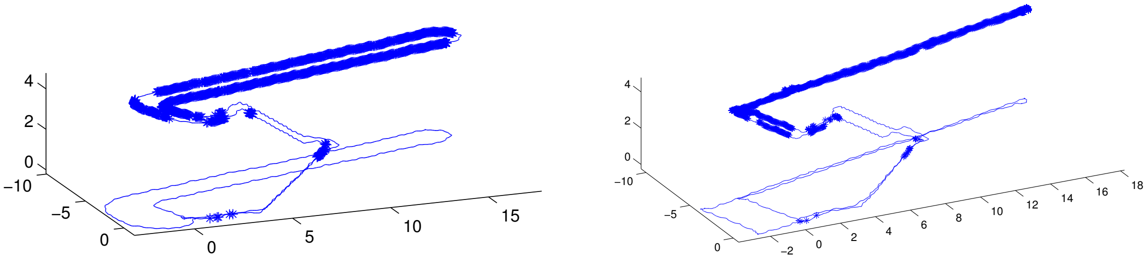
Finally, the algorithm checks to ensure that there are enough support points in the accepted line cluster on both sides of the gravity downvector such that there are support points both in front and behind the operator. This ensures that the operator is currently standing on the detected floor plane. The number of points is based on the angular resolution, the typical height of the pitch scanner, and the desired support length of floor. In our case, this is 30 points, which corresponds to a scanner height of approximately 1.2m and a floor support length of approximately 0.5m. If the final floor plane passes this final check, then that pose is marked as being in a planar environment. An example of a pitch scan in a planar environment is shown in Figure 4(a), and an example in a non-planar environment is shown in Figure 4(b).

The segmentation algorithm is tested on two separate datasets. Both consist of two long hallways connected by a stairwell. Unlike the upper floor, the lower floor has an extremely reflective floor with no returned laser points. The segmentation results are shown in Figure 5. The boldface regions correspond to planar floor detections. As



(a) (b)

Figure 4. Two sample pitch scans in (a) planar and (b) non-planar environments. The red points indicate the floor points detected by the adaptive segmentation algorithm and the circle is the position of the pitch scanner.



(a) (b)

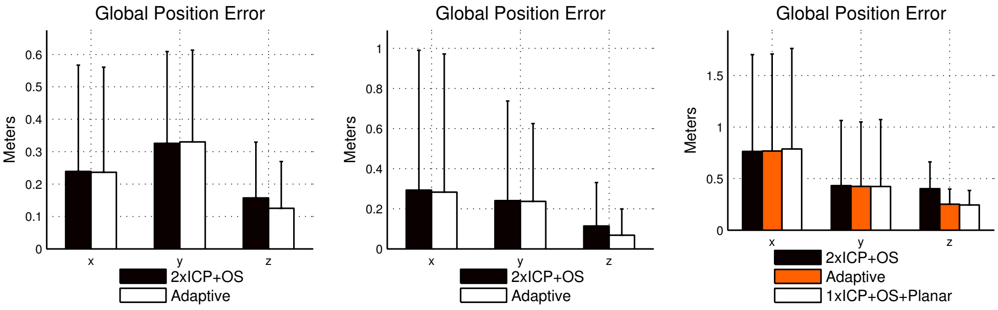
Figure 5. Segmentation results for (a) dataset 1; (b) dataset 2; the boldface markers indicate regions where planar floors are detected.

seen, our proposed segmentation algorithm correctly identifies planar regions on the top floor. Even though the lower floor is planar, it is not detected as planar since it is reflective and has no returns.

The output of the segmentation algorithm is a disjoint, binary partitioning of the transformations between successive poses. Specifically, T_p denotes all transformations that should be computed under the planar assumption and T_{np} denotes those without. The segmentation results are combined with the localization procedures of Sections 3.1 and 3.2 to generate an adaptive localization algorithm. At each time interval, the segmentation algorithm is run and the transition from the pose at t_{i-1} to the pose t_i is classified as planar, T_p, or non-planar, T_{np}. If the transition is a member of T_p, then the transformation is computed according to the 1×ICP+OS+Planar algorithm; otherwise, it is computed via 2×ICP+OS.

3.4 Adaptive Localization Results

The adaptive localization algorithm is characterized on three separate data sets, referred to as datasets 1, 2, and 3. The first two datasets consist of two long hallways connected by a stairwell. Dataset 1 is comprised of two roughly 20-meter hallways connected by a stairwell roughly 4.5-meters in height. Similarly, dataset 2 consists of two 20-meter hallways connected by a 4.5-meter stairwell. Since the complex environment does not allow for the planarity assumption for the entire path, the adaptive localization algorithm is compared to 2×ICP+OS for datasets 1 and 2. On the other hand, for dataset 3, which is entirely planar, we compare the adaptive algorithm to both 2×ICP+OS and 1×ICP+OS+Planar. A manually detected set of LCs are used for all results presented



(a) (b) (c)

Figure 6. Global RMS position error for 2×ICP+OS and adaptive; markers above each bar denote peak error; (a) dataset 1; (b) dataset 2; (c) dataset 3.

Dataset	Average Position Error		% Change
	2×ICP+OS	Adaptive	
1	0.401 m	0.396 m	-1.2%
2	0.343 m	0.321 m	-6.4%
3	0.903 m	0.847 m	-6.2%

Table 1. Average position error for each dataset for 2×ICP+OS and adaptive localization methods.

in this section. Comparison is made to the ground truth data collected by the Honeywell HG9900 IMU. Global errors are computed in a world reference frame such that x is east, y is north, and z is upwards.

It is important to clearly distinguish between what we call incremental and global errors. Incremental errors refer to the error in any of the 6 DOF parameters from one time instant to the next. Global error is computed by (a) successively applying incremental transformations from our proposed localization algorithms for all times to reconstruct the entire 6 DOF localization path, including position and orientation, and (b) comparing their values with those obtained via the ground truth. Thus, global errors result from accumulated incremental errors. As such, the magnitude of global error for each localization parameter is for the most part decoupled from that of its corresponding incremental error. In particular, various components of incremental localization 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.

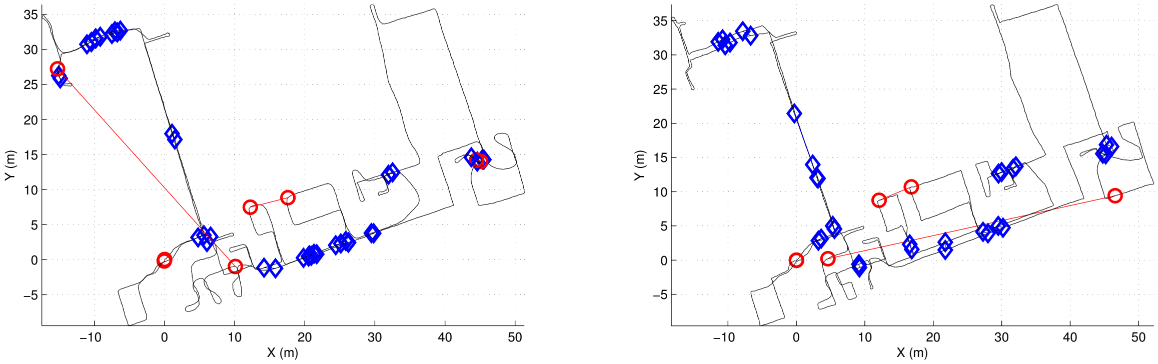
Figures 6(a) and 6(b) show the global RMS and peak position errors for datasets 1 and 2, respectively. As seen, the adaptive algorithm substantially lowers both the peak and RMS localization error for the z-axis. Specifically, for datasets 1 and 2, the reduction in RMS z-error is roughly 20% and 40%, respectively. The translation along other axes as well as global and incremental rotations remain largly unchanged. This is a simple consequence of the fact that both 2×ICP+OS and 1×ICP+OS+Planar estimate all other parameters in the same manner.

Figure 6(c) shows the global RMS and peak errors for dataset 3. Dataset 3 consists of a single T-shaped hallway with a non-glossy floor. The performance of the adaptive algorithm on dataset 3 is almost identical to that of 1×ICP+OS+Planar and roughly 40% better than 2×ICP+OS for z. This is to be expected because the adaptive algorithm classifies almost the entire dataset as planar, 91% in this case.

The average position errors for 2×ICP+OS and the adaptive algorithm for all three datasets are shown in Table 1. As seen, the adaptive algorithm outperforms 2×ICP+OS for all three datasets. To conclude, the proposed adaptive algorithm has been shown to be more accurate than 2×ICP+OS in mixed environments with planar and non-planar floors, at the same time as being as accurate as 1×ICP+OS+Planar in planar floor environments.

4. LOOP CLOSURES FROM MULTIPLE CAMERAS

Rather than using images from only a single camera to look for LCs, in this section we propose to use images from a pair of diametrically opposed side-looking cameras, where one points to the operator's left and the other to the



(a) (b)

Figure 7. The LCs automatically detected from camera images for (a) dataset 4 with 24 closures, (b) dataset 5 with 19 closures; LCs detected by a single (double) camera are shown with red circles (blue diamonds). Note that there are a few erroneous detections.

operator's right. In doing so, many more LC points are detected when the operator retraces the path in opposite directions. Figure 7 shows LCs for 2 datasets detected by a single camera in red circles, and the additional LCs detected by two diametrically opposed side-looking cameras in blue diamonds. The two camera LC detection increases the number of LCs by three-fold or better, allowing the localized path to be better constrained in the regions around each LC. In this localization process, a graph is constructed where each node represents a pose at a moment in time, and the edges connecting the nodes are the incremental transformations and the covariance matrices of those transformations. When an LC is identified, an extra edge is added to connect different parts of the graph together.

Once an LC is detected, the incremental transformation and covariance is computed and inserted as an edge to the pose graph. This is done by scan matching the laser scans associated with the LC in a process similar to the way incremental transformations are estimated for the algorithms described in Section 3. However, as the scan events for LCs are from different points in time and do not necessarily correspond to sufficiently similar poses, the PLICP scan matching may converge to a local minima,[12] especially since the initial estimate of the transform is not readily available. In addition, geometrically simple indoor environments, such as hallways where scans appear as parallel lines, can lead to degenerate cases with erroneous PLICP estimates of the translation for the yaw scanner. This is particularly problematic as the yaw scanner is used to recover 3 out of 6 localization variables, regardless of which specific localization algorithm we use. Concurrent to this issue is the problem that the LC detector outputs a small number of incorrect false LC detections, for which it is not possible for scan matching to estimate the incremental transformation as the two nodes are not in the same location. Both sources of erroneous transforms lead to large errors during the TORO optimization process, and therefore steps must be taken to prune LC candidates with poor/incorrect transformations.

To detect these erroneous transformations, we apply a set of criteria to all candidate LC transformations; two of these are based on statistics generated by the yaw scan matching process, as described in Section 4.1, and another three are based on the consistency of the transformation with the 3D surroundings, as described in Section 4.2. We evaluate these criteria with a random forest classifier, as described in Section 4.3. Section 4.4 compares the results of using single and multiple camera LCs.

4.1 Pruning Based on Scan Matching Statistics

The first two metrics for pruning erroneous LCs are computed based on the matched, or associated, points between the two laser scans during the scan matching process. Associated points are determined by taking into account the transformation between two successive yaw scans from the horizontal yaw scanner, and computing the distance from each point in the first scan to the nearest neighbor in the second scan. Points for which this distance is below a specified distance, or association gate, are considered to be associated, as shown in Figure 8. We then compute the percentage of associated points in each scan as well as the residual distance between those

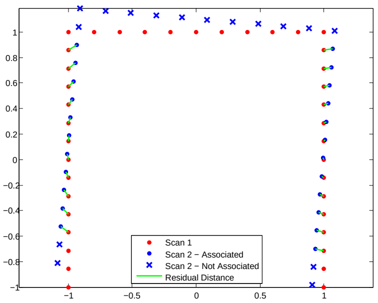


Figure 8. Illustration of scan point association. Red points are from scan 1, blue points are from scan 2, and the associated points are connected with green lines.

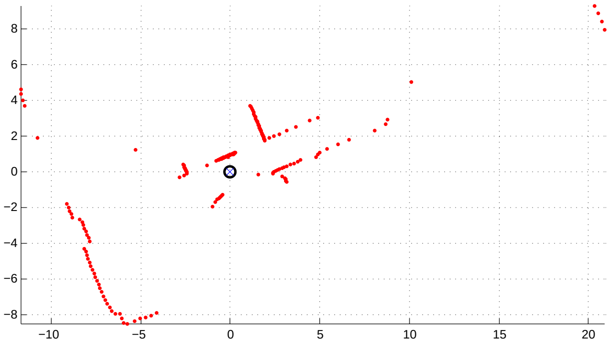


Figure 9. Sample laser scan showing the decrease in scan density as range increases. The scanner location is marked with a circle.

associated points. However, this percentage gives disproportionate weight to scene geometry near the scanner due to the polar nature of the laser scanner, whereby near scene geometry has a higher density of scan points compared to far scene geometry, as shown in Figure 9. In order to ameliorate this problem, we instead compute a weighted association percentage in which each associated point is weighted by its range, r_i to the scanner:

$$\text{Weighted Association \%} = \frac{\sum_{i \in A} r_i}{\sum r_i} \tag{1}$$

where A is the set of associated points.

The second metric for pruning LCs is simply the mean residual distance between the associated points. As the initial conditions to the scan matching process are not known, other than which side camera the images came from, the process needs to be repeated across various association gates in order to avoid converging to a local minima. Throughout this paper, the association gates are chosen to be 0.05m, 0.45m, and 1m. Each of these association gates provides a candidate LC transformation. Intuitively, the weighted association percentage indicates how similar the two scans are and the mean residual distance is a measure of the quality of the scan alignment.

4.2 Pruning Based on Image Features

We propose an error metric based on the transformation of the 3D location of features between matching LC images. We start with the two images, I_1 and I_2, corresponding to LC events, and based on their timestamps, we collect the corresponding 2D scans from the side-looking vertical geometry scanners which are approximately captured around the same time as the two images. In doing so, we ensure that laser scans and corresponding camera images both come from the same side of the backpack. Specifically, we choose 15 seconds[*] or 150 consecutive scans from the left (right) looking side scanner whereby the timestamp for the 75th scan is closest to that of the image from the left (right) camera. We then apply pre-TORO, unoptimized[†] open loop pose estimates from the algorithms in Section 3 to the scans in order to generate a small point cloud. This point cloud is then projected onto the LC images resulting in 3D depth values for the 2D pixel locations in the images. Figure 10 shows an example of such a depth map.

We then apply a feature extraction algorithm such as SIFT[11] to I_1 and I_2 and match their features to obtain a set of matching features with pixel locations y_1 and y_2. Depth values are assigned to the features according to the following methodology. For each feature, the laser points which project near each feature, i.e. within 25 pixels, are extracted and clustered by their distance to the camera center at the capture time of the image. This clustering is done by sorting the range values, computing the distance between neighboring values and looking for large gaps in the list which exceed 0.25m. The nearest cluster is selected as the candidate cluster. Then the

[*] We have empirically found 15 seconds of laser data to be sufficient to describe the entire field of view of the cameras.

[†] The use of the unoptimized poses is not problematic in this application because they tend to be locally accurate. Besides, prior to TORO optimization, open loop, dead reckoning poses are the only ones available to use.

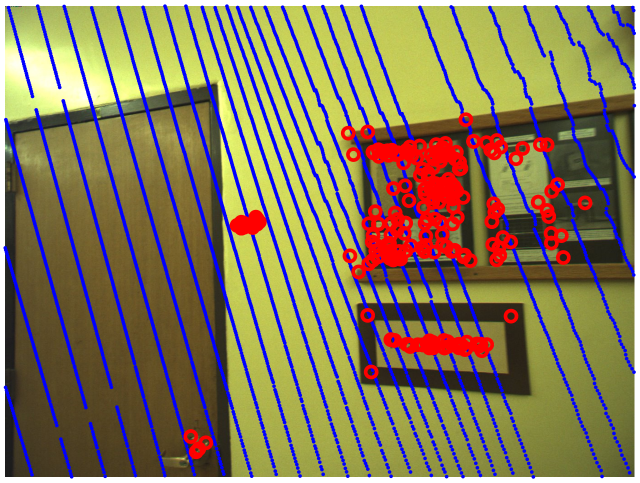


Figure 10. Example depth map using projected laser points to estimate depth. Projected laser points are shown in blue and SIFT features are shown as red circles.

features are individually assigned to the 3D position of the nearest projected laser point in the candidate cluster only if the distance is less than 15 pixels. The clustering process is performed in order to ensure that the final associated laser point corresponds to the same geometry seen by the camera and not geometry captured at a different time instant.

Matching features with assigned depth values are collected into the sets $\hat{y_1}$ and $\hat{y_2}$. Outlier rejection to remove erroneous feature to laser point correspondences is performed by computing the residual distance between the points in set $\hat{y_1}$, rototranslated by the candidate transform, and the points in set $\hat{y_2}$, and then discarding points which have residual distances beyond 3 times the standard deviation from the mean.

The candidate LC transformations are then scored according to:

$$e = \frac{1}{N} \sum_{i=1}^{N} \parallel \hat{y_2}(i) - \xi(\hat{y_1}(i), \hat{\boldsymbol{\theta}}, \hat{\mathbf{t}}) \parallel \tag{2}$$

where $\xi(\hat{y_1}(i), \hat{\boldsymbol{\theta}}, \hat{\mathbf{t}})$ is the 3D location of feature $\hat{y_1}(i)$ rototranslated by the candidate 6 DOF transformation $\xi(\cdot, \hat{\boldsymbol{\theta}}, \hat{\mathbf{t}})$, with $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{t}}$ denoting estimated rotations and translations respectively. In essence, this represents the mean distance between the 3D locations of features $\hat{y_1}$ and $\hat{y_2}$ under the candidate transformation. For an ideal depth map, e would be zero when using a perfect candidate transformation. However, since the depth map contains errors, e tends to be nonzero even with a perfect transformation.

We estimate the quality of the error by examining the standard deviation of the residual distance between between the points in set $\hat{y_1}$, rototranslated by the candidate transform, and the points in set $\hat{y_2}$. If the standard deviation is high, greater than 0.1m for our data, this indicates either a high percentage of poor feature to laser point correspondences or a poor rotation estimate. Typically, scan matching provides reasonable rotation estimates and so a high standard deviation of the residuals is most often a sign of poor feature to laser point correspondences. Another important indicator of estimate quality is the number of matched features in sets $\hat{y_1}$ and $\hat{y_2}$; if this number is small, say below 10, e is likely to be invalid as there is not enough support for the error estimate.

4.3 Pruning With a Random Forest Classifier

Now that we have both laser and image-based metrics for the three candidate loop closure transformations, corresponding to the three association gates used in scan matching, we need a method to determine which of the transformations to use or to reject the loop closure entirely. Since it is difficult to define simple heuristics due to the somewhat noisy nature of the metrics, we choose to train a random forest classifier[14] to predict the best classification given the metrics. Random forests train a set or "forest" of randomly generated decision trees, each using a subset of the available features. By training each tree using a different subset of the data, random forests are quite resistant to overfitting.

We implement our classifier in a 2 layer structure. In the first layer, we train two separate random forests, both classifying the quality of each individual loop closure transformation. The first uses only the scan matching statistics from Section 4.1, namely, the weighted association percentage, the mean residual error of the associated points, and the association gate[‡] used, in addition to whether the loop closure has been detected from a single camera or opposing cameras. This last feature is used because for LC events from the same camera, translation and rotation is close to zero. In this case, convergence in scan matching is not problematic as it happens close to the zero initial condition. In contrast, LC images from opposing cameras often correspond to scans from slightly different portions of the scene; this is due to a large pitch difference when the system is worn facing left as compared to facing right.

The second classifier in the first layer uses all the same features as the first, in addition to the image feature based metrics described in Section 4.2, including the mean error, the standard deviation, and the number of features used. The reason for having two separate classifiers in this layer is that not all LC transformation estimates return a valid image feature based error. This is due to insufficient number of matched features or a high standard deviation of the residual error. For a given LC candidate, if the image based error is available, then we apply the classification from this second classifier which uses both laser and image based metrics. If not, we fall back to the classification from the first classifier which only uses laser based metrics. This results in an acceptable/not acceptable classification of each individual transformation estimate.

In the second layer, we use a third random forest classifier to select the best transformation among various association gates or reject the loop closure. All the previous features used in the first layer for all three association gate transformations estimated for each loop closure, in addition to the classification result from the first layer, are input as one sample to the classifier. The output is the selected transformation together with its association gate or a rejection of the loop closure.

In order to train the classifier, we use 356 loop closures from 7 different datasets. Truth classifications are manually generated due to inertial drift in our ground truth system accumulating to greater than 0.75m on some of our longer paths. We classify all loop closure transformations which are within 0.5m of the correct transformation as an acceptable loop closure and then manually select the best transformation/association gate for each loop closure. In doing so, we are able to correctly select 229/240 (94.5%) of the loop closures with a good transformation; however, 38 of those (15.8%) are sub-optimal transformations. We correctly reject 109/116 (94.0%) of the loop closures with unacceptable transformations. This means a false negative rate of 4.6% and a false positive rate of 6.0%.

We have tested the classifier on a dataset not used in the training set. This dataset has 18 detected loop closures; 7 loop closure transformations are returned after pruning with the classifier, 6 of which are acceptable and 1 which was unacceptable. The false positive occurs in two similarly structured but separate rooms, practically indistinguishable from the loop closure detection images with the exception of a different paint scheme on one wall, which is not captured by the image features.

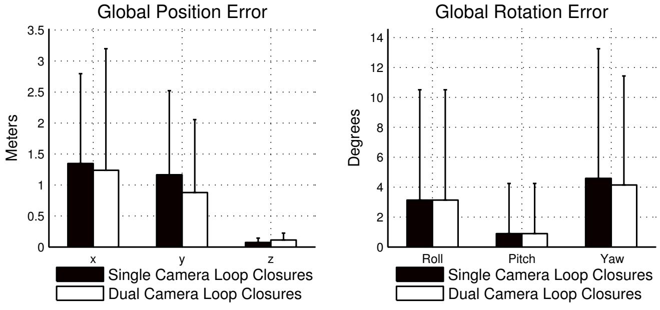
4.4 Results on Multiple Cameras LCs

We test the approach described in this section on two datasets 4 and 5, shown in Figures 7(a) and 7(b) respectively. The datasets are separate, approximately 500m traverses of a series of rooms and hallways on one floor, returning back to the beginning. We compare the localization error for two cases, both of which automatically detect LCs using our previous approach.[6] In the first case, which we refer to as "Single Camera Automatic," or SCA, one camera is used and the initial translation, $\Delta\mathbf{t} \approx 0$ and the initial yaw, $\Delta\psi \approx 0$. In the second case, which we refer to as "Dual Camera Automatic," or DCA, one or two cameras are used and $\Delta\mathbf{t} \approx 0$ and $\Delta\psi \approx 0$ or $180°$. Table 2 shows the number of loop closures used in each case with and without pruning. As expected, DCA has more LCs than SCA even after pruning.

[‡] Note that for each LC there are three transformation candidates, corresponding to three different association gates.

Dataset	Detected Loop Closures	
	SCA	DCA
4	4 (1)	20 (11)
5	3 (1)	16 (6)

Table 2. Number of LCs detected for each dataset and detection mode: single camera (SCA) and dual camera (DCA), with the number of LCs after pruning in parentheses.



(a) (b)

Figure 11. Global RMS (a) position and (b) rotation error characteristics using SCA and DCA LCs for dataset 4. Markers above each bar denote peak errors.

	Avg. Pos. Error	% Change
SCA LCs	1.579 m	—
DCA LCs	1.313 m	-16.8%

Table 3. Average position errors for dataset 4 using SCA and DCA LCs.

Figure 11(a) shows global position error for dataset 4. As seen, global position errors are generally improved with DCA; global x (y) errors for DCA are about 8% (25%) lower than SCA. Figure 11(b) shows global rotation error; as seen, global yaw errors for DCA are about 10% lower than SCA. Average position error for dataset 4 is shown in Table 3. As expected, DCA outperforms SCA by 16.8%. The reconstructed paths are shown in Figure 12(a) along with the ground truth path. Note how the SCA path drifts down and away from the ground truth path in the lower middle section of the path, shown circled. The DCA path does not suffer from this due to LCs in this area.

Ground truth data was not collected for dataset 5 and so position errors are not computed, however, visually, the DCA path looks more accurate than the SCA path, as shown in Figure 12(b).

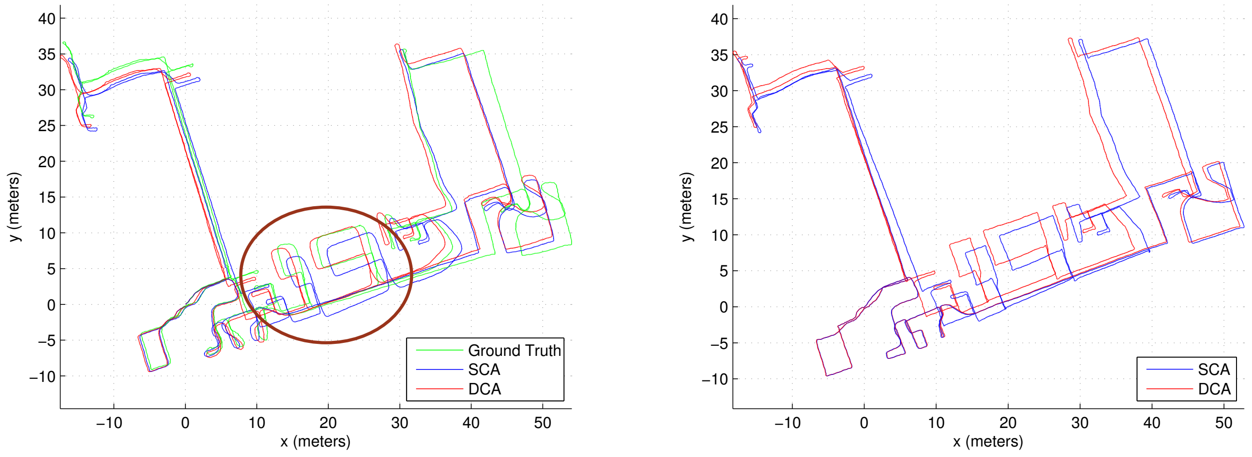
5. CONCLUSIONS

In this paper we have presented an adaptive localization algorithm which detects and exploits floor planarity in order to increase accuracy in mixed environments with planar and non-planar floors. We have presented a method for the integration of automatic image based LCs from diametrically opposing cameras for situations where data acquisition paths are re-traversed. We have shown that in general, after pruning the resultant erroneous LC transformations, the addition of the extra LCs resulting from using left and right cameras decreases localization error.

REFERENCES

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

[2] Pradeep, V., Medioni, G., and Weiland, J., "Visual loop closing using multi-resolution SIFT grids in metric-topological SLAM," in [*Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*], (2009). 1, 2



(a) (b)

Figure 12. Reconstructed paths for SCA and DCA LCs for datasets (a) 4 and (b) 5. Ground truth data is not available for dataset 5. Note how the SCA path in the circled area of dataset 4 drifts downwards away from the ground truth in comparison to the DCA path.

DCA	Dual Camera Automatic
DOF	Degree Of Freedom
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
LC	Loop Closure
OS	Orientation Sensor
PLICP	Point-Line Iterative Closest Point[12]
SCA	Single Camera Automatic

Table 4. Common abbreviations used in this paper.

[3] Bosse, M. and Zlot, R., "Continuous 3D scan-matching with a spinning 2D laser," in [*Proc. of the IEEE Int. Conference on Robotics and Automation*], 4244–4251 (2009). 1

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

[5] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1052–1067 (2007). 1

[6] Chen, G., Kua, J., Shum, S., Naikal, N., Carlberg, M., and Zakhor, A., "Indoor localization algorithms for a human-operated backpack," in [*3D Data Processing, Visualization, and Transmission*], (2010). 1, 3, 4, 10

[7] Liu, T., Carlberg, M., Chen, G., Chen, J., Kua, J., and Zakhor, A., "Indoor localization and visualization using a human-operated backpack system," in [*Int. Conference on Indoor Positioning and Indoor Navigation*], (2010). 1, 2

[8] Newman, P. and Cummins, M., "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The Int. J. of Robotics Research* **27**(6), 647–665 (2008). 2

[9] Granstrom, K., Callmer, J., Ramos, F., and Nieto, J., "Learning to detect loop closure from range data," in [*Proc. of the IEEE Int. Conference on Robotics and Automation*], (2009). 2

[10] Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., and Burgard, W., "Efficient estimation of accurate maximum likelihood maps in 3D," in [*Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems*], (2007). 2, 3

[11] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**(2), 91–110 (2004). 2, 8

[12] Censi, A., "An ICP variant using a point-to-line metric," in [*Proc. of the IEEE Int. Conference on Robotics and Automation*], (2008). 3, 7, 12

[13] Censi, A., "An accurate closed-form estimate of ICP's covariance," in [*Proc. of the IEEE Int. Conference on Robotics and Automation*], (2007). 3

[14] Breiman, L., "Random forests," *Machine Learning* **45**, 5–32 (2001). 10.1023/A:1010933404324. 9